

A Prior-Aware Metric for Efficiently Distinguishing Memorization from Generalization in Large Language Models

Trishita Tiwari
Cornell University

Ari Trachtenberg
Boston University

G. Edward Suh
NVIDIA, Cornell University

Abstract

Training data leakage from Large Language Models (LLMs) raises serious concerns related to privacy, security, and copyright compliance. A central challenge in assessing this risk is distinguishing prefix-specific memorization of training data from the generation of statistically common sequences. Existing approaches to measuring memorization often conflate these phenomena, labeling outputs as memorized even when they arise from generalization over common patterns. Counterfactual memorization and other related metrics (Zhang et al., 2023; Wang et al., 2025; Lesci et al., 2024) provide principled solutions, however, their reliance on retraining multiple baseline models or parsing through the training data makes them computationally impractical at scale. This work introduces *Prior-Aware memorization*, a theoretically grounded, lightweight and training-free criterion for identifying prefix-specific memorization in LLMs. The key idea is to evaluate whether a candidate suffix is strongly associated with its specific training prefix or whether it appears with high probability across many IID sampled sequences from the training data distribution due to statistical commonality. We correlate our metric with counterfactual memorization, and also evaluate it on the training corpora of two pre-trained models, LLaMA and OPT. Our results show that between 55% and 90% of sequences previously labeled as memorized fail our criterion and are consistent with statistical commonality

1 Introduction

The growing adoption of Large Language Models (LLMs) has amplified concerns over training-data leakage. Two prominent issues include (a) copyright and licensing violations, which have been the subject of several lawsuits and prior literature (Karamolegkou et al., 2023; Tremblay v. OpenAI, Inc.; Kadrey v. Meta Platforms, Inc.; The New York Times Company v. Microsoft Corporation; Chang et al., 2023), and (b) exposure of sensitive data, such as Personally Identifiable Information (PII) (Carlini et al., 2021; Mozes et al., 2023). As a result, several studies have attempted to quantify LLM training data leakage, often using various metrics to benchmark the extraction of training data in different training and inference scenarios (Carlini et al., 2021; 2022; Karamolegkou et al., 2023; Yu et al., 2023; Biderman et al., 2024).

Prior approaches to quantifying memorization in LLMs often overlook the models’ capacity to generalize, conflating prefix-specific memorization with the generation of statistically common sequences (Carlini et al., 2022; Yu et al., 2023). For example, a prompt like “The murder was committed by” may yield “John Doe” with high probability not because the model memorized this sequence from training. Even if the full sequence appears in the training corpus, the high-probability of the output may simply reflect the statistical prevalence of ‘John Doe’ as a common placeholder name. Recent work confirms that LLMs can reproduce training text verbatim by generalizing from related patterns rather than recalling specific examples (Liu et al., 2025). Thus, statistically likely sequences should not necessarily be classified as *memorized*.

Counterfactual Memorization (Zhang et al., 2023) seeks to more robustly quantify such generalization by comparing the likelihood of leaking a training sequence from models trained with and without that sequence. In this approach, a sequence has a high probability of leakage in both models is considered unlikely to be counterfactually memorized. Unfortunately, explicitly computing this difference may be very expensive and impractical for production language models, as it requires training several “counterfactual” or “baseline” models for every training sequence, all within tightly controlled settings. For this reason, prior studies on Counterfactual memorization have been limited in scale (Zhang et al., 2023; Ghosh et al., 2025). While there have been other metrics proposed to solve this issue (Lesci et al., 2024; Wang et al., 2025), they remain fundamentally constrained by the same computational intractability at the scale of modern production models as Counterfactual memorization. This motivates the need for a cheaper method for filtering out statistically common sequences.

Toward that goal, we introduce *Prior-Aware memorization* (PA memorization), a theoretically grounded and efficient, inference-only alternative to Counterfactual memorization (and related metrics). Our approach filters out statistically likely outputs (suffixes) depending on whether they are strongly associated *only* with their specific training prompts (prefixes). It distinguishes prefix-specific memorization from statistical likelihood by testing whether a suffix retains high generation probability when the model is prompted with prefixes sampled IID from the training data (according to the same distribution). If a suffix appears with high confidence across many independent prefixes, its high-likelihood reflects generalization from statistical commonality rather than from memorization of a specific prefix–suffix pair.

We validate our approach empirically against Counterfactual memorization (Zhang et al., 2023), and evaluate our methodology on text from the training corpus of two pre-trained models: Llama, and OPT. We find that using the largest OPT and LLama models, 55 – 90% of the sequences that would have been labeled as memorized are in fact statistically common sequences. More surprisingly, we note that a similar results occur even with the SATML training data extraction challenge dataset (Yu et al., 2023), where around 40% of sequences are “common” in nature; this is despite the fact that each sequence in the challenge dataset occurs only once in the entire training data (1-eidetic). *Our findings highlight the significance of looking beyond verbatim production of training data when labeling a sequence as memorized, urging us to rethink some of our previous notions about memorization in LLMs.*

Note that we audit known, naturally occurring prefix-suffix pairs using likelihood access and target-independent prefixes sampled from the training distribution; synthetic injections are used only for validation. We do not study adaptive prompt search or discovery of unknown records, meaning that “prior-aware” is a post-hoc auditing metric, not an adversarial-extraction bound.

The remainder of this work is organized as follows: in Section 2, we outline the research gaps motivating our work. Section 3 includes the definitions of the metrics we use for our experimental results. We present our findings in Section 4 and highlight our novel conclusions. We conclude this work in Section 5.

2 Generalization vs Memorization: Issues with Leakage Measures

Prior work generally considers training data that can be produced verbatim with high probability to be “memorized” (Carlini et al., 2021; 2022; Biderman et al., 2024; Yu et al., 2023; Shi et al., 2024). However, as noted in our Introduction, such methods may misclassify statistically common sequences as memorized. To make this limitation precise, consider a prefix p and suffix s , with $p||s \in D$ indicating that their concatenation occurs in the dataset D . One popular instantiation of existing memorization metrics typically deem the sequence $p||s$ to be *memorized* if the conditional probability $P(s | p)$ is large (Carlini et al., 2022; Yu et al., 2023). Following prior work, we refer to this definition as *extractable memorization*, because the memorized content can be elicited from the model by supplying prefix p . However, we argue that memorization should not be defined this way. This is because there are multiple reasons that $P(s | p)$ can be high, as can be seen through Bayes’ rule:

$$P(s | p) = \frac{P(p | s) \cdot P(s)}{P(p)} \quad (1)$$

Indeed, $P(s \mid p)$ can be high if:

1. If $P(s)$ is very large, suggesting that s may be statistically popular (generic). For instance, in the sequence if $p = \text{"The murder was committed by"}$ and $s = \text{"John Doe"}$, $P(s)$ may be large because s is a popular occurrence in the dataset.
2. If the relative belief ratio $\frac{P(p|s)}{P(p)}$ is large. This would happen if $P(p)$ is small (e.g., the prefix is a rare prefix in the dataset) relative to $P(p \mid s)$, suggesting that the suffix is a strong indicator of the prefix, as we would expect for memorization.

We argue that if $P(s|p)$ is high due to the former reason, then sequence $p||s$ is *not memorized* by the model. Indeed, variations of this observation have been made by other works (Zhang et al., 2023; Lesci et al., 2024; Prashanth et al., 2024), and have led to alternative definitions of memorization that we discuss below.

A more rigorous metric that captures this argument is *Counterfactual Memorization* (Zhang et al., 2023), which quantifies the performance differential of a model trained with versus without a specific target sequence. Under this definition, a sequence is considered memorized only if it exhibits low performance on models from which it was excluded; conversely, high performance across both model variants indicates generalization, as the output is not contingent on explicit training exposure. While robust, this approach is often computationally prohibitive. One may consider cheaper estimates to this calculation, such as training approximate baselines with a different data distribution; however, it is fundamentally difficult to make these estimates accurate (Zhang et al., 2024).

Alternative metrics, such as *Distributional Memorization* (Wang et al., 2025), attempt to account for generalization by correlating output probabilities with pretraining frequencies via a "task-gram". However, constructing such a task-gram at the scale of modern production corpora remains largely infeasible. Similarly, Lesci et al. (2024) estimate causal memorization profiles from training checkpoints. Contextual Memorization (Ghosh et al., 2025) is instead grounded in local overfitting: it compares training recollection with the best leave-one-out recollection across epochs and is stricter than counterfactual memorization. Exact computation requires per-target leave-one-out retraining and access to the training trajectory. PA uses a Bayes-motivated within-model likelihood ratio, trading that counterfactual, training-dynamics interpretation for inference-only scalability.

There have also been other definitions of memorization where it is unclear if they can practically distinguish sequences $p||s$ with statistically popular targets s from their truly memorized counterparts. One such definition is by Schwarzschild et al. (2024), which classifies a sequence as memorized if it can be generated with a compressed prompt. Similarly, concurrent work by Morris et al. (2025) defines a compression-based memorization metric using Kolmogorov complexity (which is not computable in general).

Failing to account for generalization capabilities may greatly overestimate memorization, as indicated by a recent study (Liu et al., 2025). This emphasizes the clear need for an inexpensive, but accurate, method for identifying whether a particular training sequence is truly memorized, or just statistically common.

3 Our Metric: Prior-Aware (PA) Memorization

We next propose and describe our approach to quantifying *Prior-Aware* [PA] memorization, starting with a definition and proceeding to discuss how it can be feasibly computed. Throughout this section, we presuppose a language model M , defined as follows.

Definition 1. The language model M maps an input token sequence $p = (t_1, t_2, \dots, t_j)$ to a distribution for the next token t_{j+1} :

$$M : p \mapsto P(\cdot \mid p).$$

We assume that the model M is trained on a dataset D , consisting of sequences of prefix and suffix pairs $p||s$ (where $||$ is the concatenation operator). We will use the notation $P(x|y; M)$

to denote the probability of generating sequence x from such an application of M to an initial sequence y , and $P(x; M)$ will denote the total probability of seeing x from any starting sequence. We also assume that text generation proceeds auto-regressively from a starting sequence y , and each time generating (and incorporating into y) a subsequent token. Where the choice of model is clear, we may omit it for brevity, and write simply $P(x|y)$ or $P(x)$.

Definition 2 (Prior-Aware memorization). *For tuning thresholds $0 \leq m \leq 1$ and $n \geq 0$ we say that a sequence $p||s \in D$ is Prior-Aware memorized by a model M if $P(s | p; M) > m$ and $\frac{P(s|p;M)}{P(s;M)} > n$.*

Extractability and Relative Belief. The definition imposes two requirements: First, that $P(s | p; M) > m$ (see Section 3.1), which ensures that the suffix s has a high probability of being generated verbatim by model M when prompted with p . This has been adopted from prior work (Carlini et al., 2022). Second, the definition insists that the relative belief ratio, $\frac{P(s|p;M)}{P(s;M)} > n$ (see Section 3.2), implying that s is a strong indicator of p . A large ratio ($n \gg 1$) suggests that the presence of p predicts the presence of s , or, in other words, s much more likely following p than in general.

Rationale for the Dual Criteria. Although the relative belief ratio incorporates the conditional probability $P(s | p; M)$, we explicitly maintain both terms for practical reasons. First, evaluating the conditional probability is significantly more tractable than estimating the marginal probability $P(s; M)$ (see Sections 3.1 and 3.2). Thus, in practice, the first condition serves as an efficient filter: if a sequence fails the m -threshold for extractability, it is discarded before performing the more expensive marginal density estimation required for the second term. Conceptually, if $p||s$ lacks a sufficient probability of verbatim generation, its status as a statistically common sequence is moot, as it does not pose a measurable risk of extractability.

To compute our Prior-Aware metric for any given pair $p||s \in D$, we need to compute two terms: $P(s | p; M)$ and $P(s; M)$. We explain how to do this in the next two subsections.

3.1 $P(s | p; M)$: verbatim generation of s given p

For notational convenience, we write $t_{1:j}$ to denote the sequence of tokens $(t_1, t_2, \dots, t_j)$. Using this notation, the probability of generating the k -token suffix $s = t_{j+1:j+k}$ when prompted with the j -token prefix $p = t_{1:j}$ derives from the chain rule:

$$P(s | p) = \prod_{i=j+1}^{j+k} P(t_i | t_{1:i-1}). \quad (2)$$

Equation 2 thus provides a closed form for the probability of leaking our target sequence verbatim from a given prefix. In practice, this term can be efficiently computed in a single forward-pass per token (Tiware & Suh, 2024; Hayes et al., 2024), and we can thus use this to compute whether $P(s | p) > m$ for a given $p||s$.

3.2 $P(s; M)$: generality of s

The second component of our definition is $\frac{P(s|p)}{P(s)}$. To compute the denominator, we would like to calculate the total probability over all prefixes V^* in the model’s vocabulary V . To do this, let $\pi_{V^*}^D$ denote the empirical distribution over valid prefix occurrences induced by the sampling procedure on D . Then,

$$P(s; M) \triangleq v_s = \mathbb{E}_{p \sim \pi_{V^*}^D} [P(s | p; M)] = \sum_{p \in V^*} \pi_{V^*}^D(p) P(s | p; M) \quad (3)$$

However, this calculation is computationally prohibitive. Instead, we use an unbiased estimator $\hat{v}_s$ (inspired by Monte-Carlo Integration (Binder et al., 1992)) that converges to the true probability $P(s)$ as the number samples c grows to infinity.

Definition 3. Given samples $q_1, q_2, \dots, q_c \stackrel{\text{i.i.d.}}{\sim} \pi_{\mathcal{V}^*}^D$, our total probability estimator is given by

$$\hat{v}_s = \frac{1}{c} \sum_{i=1}^c P(s \mid q_i) \quad (4)$$

Note this estimator requires samples q_i to match the distribution of prefixes $p_i \sim \pi_{\mathcal{V}^*}^D$ and be statistically (but not necessarily semantically) independent, and it is thus computed for a specific model M ; for brevity, we omit M from the notation where it can be straightforwardly inferred. The next two theorems show that $\hat{v}_s$ is unbiased.

Theorem 1. $\mathbb{E}[\hat{v}_s] = v_s$.

Proof. By linearity of expectation, $\mathbb{E}[\hat{v}_s] = \frac{1}{c} \sum_{i=1}^c \mathbb{E}[P(s \mid q_i)]$. Since q_i are chosen i.i.d, we have that $\mathbb{E}[P(s \mid q_i)] = \mathbb{E}[P(s \mid q_1)]$. This means that $\mathbb{E}[\hat{v}_s] = \frac{1}{c} \cdot (c \cdot \mathbb{E}[P(s \mid q_1)]) = \mathbb{E}[P(s \mid p_1)] = v_s$. Recall that q_1 and p_1 are identically distributed to p_i . $\square$

Indeed, the variance $\lim_{c \rightarrow \infty} \text{Var}[\hat{v}_s] = 0$, as demonstrated in the following theorem (whose proof is relegated to Appendix A due to space constraints):

Theorem 2. $\text{Var}[\hat{v}_s] = \frac{1}{c} \cdot \text{Var}[P(s \mid p_i)] \leq \frac{1}{4c}$

In other words, over a large number of prefixes sampled from the dataset, the error of the estimator tends to 0.

4 Evaluation

In this section, we start by empirically evaluating the correlation of Prior Aware memorization with Counterfactual memorization (Zhang et al., 2023), an existing metric that aims to distinguish memorized $p||s$ from those where s is statistically common. Due to the prohibitive cost of computing Counterfactual memorization, we first conduct two small-scale correlation experiments with real and synthetic data. We discuss the experiments with real data in Section 4.1, but relegate a discussion of the latter to Appendix B. We then evaluate PA memorization alone at scale with pretrained models and note any trends in Section 4.2.

4.1 Correlation with Counterfactual Memorization

We first empirically show that PA memorization can indeed distinguish $p||s$ where s is statistically popular by measuring its correlation with Counterfactual memorization (Zhang et al., 2023). To do so, we train several models in a controlled setting, and measure both Counterfactual memorization and our Prior-Aware memorization $\frac{P(s|p)}{\hat{v}_s}$. We then show that these quantities are positively correlated with a Pearson Correlation Coefficient of 0.71.

4.1.1 Counterfactual Memorization

For completeness, we restate the definition of Counterfactual Memorization from (Zhang et al., 2023).

Definition 4 (Counterfactual Memorization (Zhang et al., 2023)). Given a training algorithm A that maps a training dataset D to a trained model f , and a measure $L(f, x)$ of the accuracy of f on a specific example x , the Counterfactual Memorization of a training example x in D is given by

$$\text{mem}(x) \triangleq \mathbb{E}_{S \subset D, x \in S} [L(A(S), x)] - \mathbb{E}_{S' \subset D, x \notin S'} [L(A(S'), x)] \quad (5)$$

where S and S' are subsets of training examples sampled from D . The expectation is taken with respect to the random sampling of S and S' , as well as the randomness in the training algorithm A . Generally speaking, the first term measures expected accuracy over models that contain the target sequence x , and the second over models without x . A sequence is only considered Counterfactually memorized if this difference is large, as it indicates that the high performance on x by the former models is due to explicit training exposure.

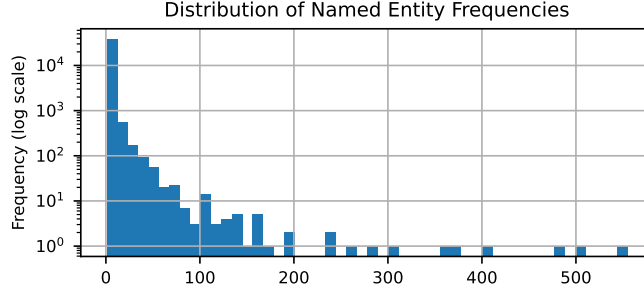


Figure 1: Distribution of Named Entities across Wikitext. The x-axis shows the frequency of each Named-Entity, and the y-axis shows how many named entities fit in each bucket.

4.1.2 Metric measurements.

We describe how we measure Definitions (4) and (2) in our experiments.

Counterfactual Memorization. In our setup, we interpret x in (4) as $p||s$, where p and s are of equal length. To simplify notation, we use M and M' to refer to $A(S)$ and $A(S')$ respectively, which we refer to as the target (M) and baseline (M') models (described below). To measure the accuracy of the model $M(x = p||s)$, we utilize the measure $\log(P(s | p; M))$, following the definition of extractable memorization based on (Carlini et al., 2022).

Thus, we measure the following:

$$\mathbb{E}_M[\log(P(s | p; M))] - \mathbb{E}_{M'}[\log(P(s | p; M'))] \quad (6)$$

PA Memorization. While PA memorization (Definition 2) is formally defined as a binary indicator, Counterfactual memorization yields a continuous-valued score. To facilitate a direct comparison between these definitions, we employ the continuous relative belief ratio, $\frac{P(s|p;M)}{\hat{\vartheta}_{s;M}}$, omitting the discrete threshold n . Furthermore, although Definition 2 includes a preliminary conditional probability threshold, we waive this filtering step in our small-scale analysis as its primary function is to optimize large-scale experiments. This simplification allows for a direct correlation analysis between the singular PA score and the corresponding Counterfactual score for every sequence. For numerical stability and alignment with common evaluation practices, we compute this ratio in log-space:

$$\mathbb{E}_M[\log\left(\frac{P(s | p; M)}{\hat{\vartheta}_s}\right)] = \mathbb{E}_M[\log(P(s | p; M))] - \mathbb{E}_M[\log(\hat{\vartheta}_{s;M})] \quad (7)$$

Note that PA memorization only relies on M , thus obviating the need for training baseline models M' like with Counterfactual memorization.

4.1.3 Target and Baseline Datasets.

Dataset Here, we train 124M-parameter GPT-2 models (Radford et al., 2019) on WikiText (Merity et al., 2016). Crucially, we select naturally occurring $p||s$ pairs from WikiText in which the suffix s spans a wide range of frequencies. Specifically, we focus on pairs where s is a named entity, following prior work on memorization (Lukas et al., 2023). Figure 1 presents a frequency histogram of named entities in WikiText: for example, “United States of America” appears approximately 500 times, whereas “Starlicide” occurs only once.

We sample a total of 50 $p||s$ pairs, with s drawn uniformly across the frequency buckets shown in Figure 1. Each target requires a separately retrained leave-one-out baseline (50 baseline runs per seed here, plus the shared target model), so the cost scales linearly with the audit set. We therefore treat this experiment as a proof of concept motivating PA’s inference-only scalability, not as a corpus-level prevalence estimate.

Target Model Here, we train the model as-is on Wikitext without modifying the dataset.

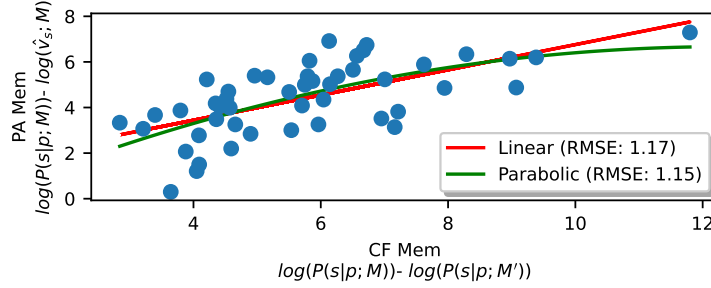


Figure 2: Counterfactual Memorization (x-axis) vs PA Memorization (y-axis). Each data point represents the scores of a single $p||s$ in WikiText, and the PCC between the two quantities is 0.71, and the Spearman coefficient is 0.65 with a p-value of $3.53 \cdot 10^{-7}$.

Baseline Model In this setting, we retrain the model after removing the target sequence $p||s$ from the WikiText training data. For example, if $p||s$ corresponds to “I live in the” || “United States of America,” this exact prefix-suffix pair is excluded prior to training. Importantly, we do not remove other occurrences of the suffix s (e.g., “United States of America”) that appear with different prefixes, since our goal is to measure memorization of the specific sequence $p||s$ rather than the suffix in isolation.

4.1.4 Correlation Results

Figure 2 shows a positive association across the 50 pairs sampled from WikiText (Pearson $r = 0.71$; Spearman $\rho = 0.65$, $p = 3.53 \times 10^{-7}$) between Counterfactual Memorization metric (Equation 6) on the x-axis, and $\frac{P(s|p)}{\hat{v}_s}$ (Equation 7) on the y-axis. The mild flattening at high counterfactual scores is consistent with exposure increasing both $P(s|p; M)$ and $\hat{v}_s$, thereby compressing their log-ratio. PA is therefore best viewed as a ranking proxy rather than a linearly calibrated replacement for counterfactual memorization.

4.2 Large Scale Evaluation

Now that we have shown PA memorization correlates with Counterfactual memorization, we now evaluate PA memorization by itself on 2 recent generative pre-trained models with open source datasets: Llama (Touvron et al., 2023) and OPT (Zhang et al., 2022). These are respectively pre-trained on the Common Crawl and The Pile. We experiment with all available sizes of the models, namely 3B, 7B, and 13B for Llama, and 125M, 350M, 1.3B, 2.7B, 6.7B and 13B for OPT. Unless otherwise stated, the default model size for which results are displayed is OPT 6.7B and Llama 7B.

4.3 Target Sequences for Extraction

To compute $P(s)$ for each sequence we test, we prompt the model with 5000 randomly sampled sequences from their respective datasets and measure the likelihood of generating s in each case. We repeat this for 5 trials. Our evaluation is performed in 3 different settings:

Named Entities. Following Lukas et al. (2023), we randomly sample $\approx 5000 - 8000$ sequences that contain Named Entities (NEs) from each of the datasets. We do this since NEs can simulate sensitive data that might be targets for extraction since they are typically names of individuals, organizations, places, etc. We must note that this is simply to emulate a realistic use-case; in reality, model developers would repeat the measurements with sequences that they consider sensitive. For consistency, each of our sequences consists of a prefix of 50 tokens followed by a 4-token long Named Entity. While we experiment with longer prefix lengths of upto 400 tokens, the default prefix length is 50 tokens.

Long Sequences. We also simulate extraction of longer sequences, which could be useful to measure the risk of leaking copyrighted data. For this, we randomly sample 5000 sequences with 50-token prefixes and 50-token suffixes from each dataset, similar to prior work (Biderman et al., 2024; Carlini et al., 2022). Here too, we experiment with longer prefix lengths of upto 400 tokens, but unless otherwise stated, the default prefix length is 50 tokens.

SATML Challenge. Lastly, we also perform a smaller-scale, additional analysis on the dataset in Yu et al. (2023), which consists of 1-eidetic sequences (sequences where each $p||s$ is known to occur only once in The Pile). This dataset was released for the 2023 SATML training data extraction challenge and consists of 15,000 sequences, each with a prefix of 50 tokens followed by a suffix of 50 tokens. We present results on 1000 of the 15,000 sequences.

4.4 Hyper-parameters

The only parameters that need to be tuned in Definition 2 are m and n :

m — threshold for $P(s|p)$: This parameter allows us to pick which $p||s$ have a high risk of leakage, and thus can be adjusted based on the risk tolerance. Since m is a threshold for $P(s|p)$, $\frac{1}{m}$ indicates, on average, the number of times a user would need to prompt the model with p to leak the target s . We set $m = 0.01$ for 4-token s , and $m = 0.0001$ for 50-token s . We deliberately pick small values of m to be conservative in our estimation of memorization.

n — threshold for $\frac{P(s|p)}{\hat{\theta}_s}$: The second parameter, n , allows us to differentiate $p||s$ where s has a large prior from those that do not. To estimate this, we compute $\frac{P(s|p)}{\hat{\theta}_s}$ over s that are known to be easy-to-predict for LLMs (examples provided in Appendix C). We set n to be the average of $\frac{P(s|p)}{\hat{\theta}_s}$ over these sequences. Note that since $\frac{P(s|p)}{\hat{\theta}_s}$ is a model specific quantity, the threshold is recomputed for each model tested; it does not guarantee a fixed false-positive rate. We also perform an ablation over different n in Appendix D.

4.5 Effect of Model Size on PA Memorization

In Figures 3 and 5, we plot both the number of *extractably memorized* (simply high $P(s|p)$), as defined by Carlini et al. (2022) and in Section 2) and *PA memorized* sequences as a function of model size. We also plot PA memorized sequences as a proportion of the total extractably memorized sequences (labeled “Ratio” in the legend). There are two interesting observations to be made:

Gap in extractable and PA memorized sequences. Consistent with prior work (Hayes et al., 2024; Carlini et al., 2022; Schwarzschild et al., 2024), both extractable and PA memorization increase as the model size increases. However, there is a difference between number of extractably memorized and PA memorized sequences. Notably, this difference is quite large when the target suffixes are 4 token Named Entities, where as many as 90% of extractable memorized samples are *not* PA memorized. This suggests that most of the suffixes of extractably memorized sequences are indeed statistically common rather than truly memorized. As shown in Table 1, many of these s are indeed popular entities such as politicians, celebrities, countries, etc. More surprisingly, we note a similar result in the SATML challenge dataset in Figure 5, where around 40% of sequences are “common” in nature. This is despite the fact that each sequence in the challenge dataset occurs only once in the entire training data.

Decreasing proportion of PA memorized sequences. Secondly, Figures 3 and 5 show that as model size increases, the proportion of PA memorized sequences generally *decreases*. This suggests that larger models are increasingly able to reproduce verbatim text by generalizing from common or near-duplicate data, rather than through true memorization—a finding consistent with recent work (Liu et al., 2025).

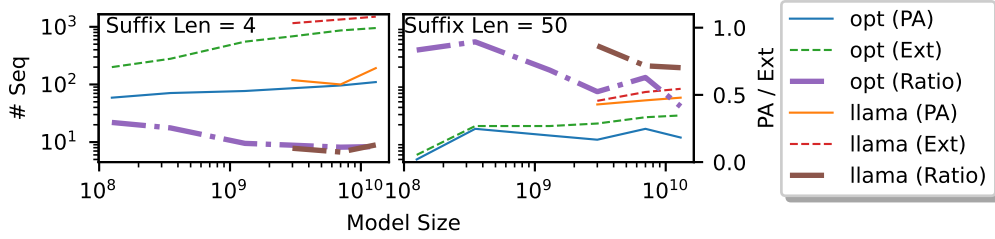


Figure 3: The primary y-axis reports the number of extractable and PA memorized sequences as a function of model size for 4- and 50-token suffixes. The thicker lines denote the proportion of extractable-memorized sequences that are also PA memorized, with their scale indicated on the secondary y-axis.

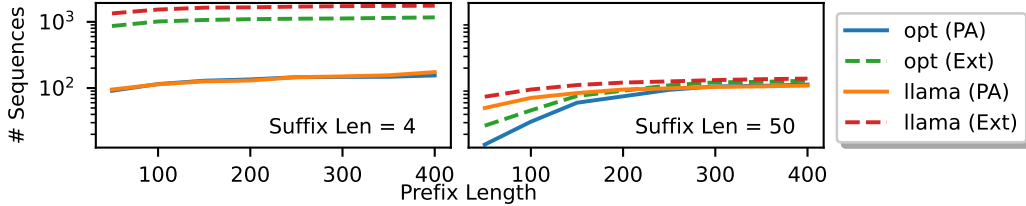


Figure 4: Number of a) Extractable, and b) PA Memorized Sequences as a function of model prefix length for two types of suffixes.

4.6 Effect of Prefix Length on Discovering PA Memorization

In Figure 4 we plot the number of samples labeled as PA memorized and extractable memorized as a function of the length of prompt p provided to the model. Unsurprisingly, we observe that longer p 's can discover more PA memorized sequences. However, note that the PA memorized of 4 token Named Entity suffixes does not benefit as much from longer prefixes as the 50 token suffixes. Again, this is likely due to the fact that many of these Named Entities (*e.g.*, United States of America) are popular occurrences in web-text, and likely do not need specific prompts to be produced correctly.

4.7 Qualitative Analysis

Table 1 shows a few sequences with 4-token Named Entity suffixes that have high and low PA memorization scores (i.e., $\frac{P(s|p)}{P(s)}$). We present more such examples in Appendix C.1. We note that all of the sequences with low scores have suffixes that are places, political figures, etc. We also attempt to verify how “rare” the high-scoring sequences are. While it is infeasible to determine rarity precisely within a dataset with trillions of tokens, we perform a web-search and note that the high-scoring sequences have very few search results. In fact, the highest scoring sequence in Table 1 has just a single search result, which is precisely the sequence that is included in the dataset. We also note that high-scoring suffixes do appear to be more “artifact-like” than their low-scoring counterparts, which appear to be more natural text. This is more apparent in longer sequences, such as the examples presented in Appendix C.1.

Score (Low)	Sequence ($p s$)	Score (High)	Sequence ($p s$)
2.9	...and is a tributary to Saginaw Bay	4052	... t1=“Sea Zone” t2=“ South Atlantic Sea Zone
3.0	...special prosecutor Leon Jaworski	3358	... misguided members of the Autonomie Club
3.7	...Jack Germond and Jules Witcover	2544	...I’m watching Gore’s Warmista-Fest
4.0	...Hospital had received Hill-Burton	1560	... PLUS Gold certification.- Corsair Gold AX850

Table 1: Examples of low and high scoring $p||s$. s in each sequence is in bold text.

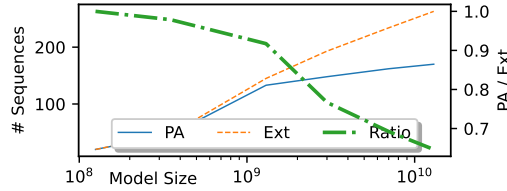


Figure 5: Memorization as a function of model size for 1K sequences from the SATML challenge dataset.

4.8 Limitations

In this section, we outline the limitations of PA-memorization.

Sensitivity to Threshold Calibration. The accuracy of PA memorization depends on the calibration of the relative belief ratio threshold, n . A model’s capacity and the entropy of its training dataset directly influence baseline probabilities. Firstly, smaller, higher-perplexity models tend to yield “flatter” marginal distributions, which can compress the relative belief ratio and make it more difficult to cleanly separate rote memorization from generalization compared to larger, more performant models. Secondly, in high-entropy (highly diverse) datasets, specific sequences are globally rarer, lowering the average prior $P(s)$ and inherently inflating the baseline relative belief ratio $\frac{P(s|p)}{P(s)}$. Conversely, low-entropy (highly structured) datasets contain frequent boilerplate, increasing $P(s)$ and suppressing the ratio.

Given this, setting n too high or low would lead to under- or over-flagging of memorization, respectively. To mitigate this, we dynamically compute $\frac{P(s|p)}{\hat{P}(s)}$ over ground-truth s that are known to be easy-to-predict for LLMs for every model size. We also provide a sensitivity analysis of the number of PA-memorized sequences to the threshold n in Appendix D

Extensions to Post-Trained Models. Our current Monte Carlo estimator for $P(s)$ relies on sampling prefixes IID according to the original training data distribution. Consequently, our large-scale empirical validations focus on base models pre-trained directly on web text. Post-training processes, such as instruction tuning or RLHF, introduce significant distributional shifts. Applying our current sampling strategy to these post-trained models without accounting for this structural shift could bias the prior estimation. Adapting the PA framework to robustly sample and estimate priors across shifted post-training distributions remains an important direction for future work.

4.9 Discussion: Assessing True Downstream Risks

PA-memorization provides a quantitative lens for *why* a given sequence is easily extractable. From a privacy perspective, it contextualizes statistically common PII that lacks the uniqueness required for meaningful violations, aligning with plausible deniability. In the context of copyright, PA memorization leverages the global prior $P(s)$ to filter highly generic, uncopyrightable boilerplate from true infringement. Finally, understanding this root cause is vital for technical mitigation: targeted data scrubbing or machine unlearning will largely fail if an output stems from broad statistical prevalence. By isolating rote recall from general popularity, PA memorization ensures that unlearning defenses are directed toward genuinely memorized sequences where they can actually succeed.

5 Conclusion

In this work, we propose *Prior-Aware memorization*, an efficient, inference-only metric for distinguishing statistically common continuations from their memorized counterparts. Across evaluated models, we find that 55 – 90% of sequences labeled as memorized by prior metrics are, in fact, statistically likely sequences that could be produced without specifically training on said sequences. This highlights how traditional metrics may overstate memorization in LLMs, and may have ramifications for privacy, copyright, and regulation.

References

- Stella Biderman, Usvsn Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. Emergent and predictable memorization in large language models. *Advances in Neural Information Processing Systems*, 36, 2024.
- Kurt Binder, Dieter W Heermann, and K Binder. *Monte Carlo simulation in statistical physics*, volume 8. Springer, 1992.
- Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.
- Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2022.
- Kent Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. Speak, memory: An archaeology of books known to chatgpt/gpt-4. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7312–7327, 2023.
- Bishwamittra Ghosh, Soumi Das, Qinyuan Wu, Mohammad Aflah Khan, Krishna P Gum-madi, Evimaria Terzi, and Deepak Garg. Rethinking memorization measures in llms: Recollection vs. counterfactual vs. contextual memorization. In *The Impact of Memorization on Trustworthy Foundation Models: ICML 2025 Workshop*, 2025.
- Jamie Hayes, Marika Swanberg, Harsh Chaudhari, Itay Yona, and Ilia Shumailov. Measuring memorization through probabilistic discoverable extraction. *arXiv preprint arXiv:2410.19482*, 2024.
- Kadrey v. Meta Platforms, Inc., 2023. URL <https://www.courtlistener.com/docket/67569326/kadrey-v-meta-platforms-inc/>. Kadrey v. Meta Platforms, Inc., 3:23-cv-03417, (N.D. Cal.).
- Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright violations and large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7403–7412, 2023.
- Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8424–8445, 2022.
- Pietro Lesci, Clara Meister, Thomas Hofmann, Andreas Vlachos, and Tiago Pimentel. Causal estimation of memorisation profiles. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15616–15635, 2024.
- Ken Ziyu Liu, Christopher A Choquette-Choo, Matthew Jagielski, Peter Kairouz, Sanmi Koyejo, Percy Liang, and Nicolas Papernot. Language models may verbatim complete text they were not explicitly trained on. *arXiv preprint arXiv:2503.17514*, 2025.
- Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 346–363. IEEE, 2023.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.
- John X Morris, Chawin Sitawarin, Chuan Guo, Narine Kokhlikyan, G Edward Suh, Alexander M Rush, Kamalika Chaudhuri, and Saeed Mahloujifar. How much do language models memorize? *arXiv preprint arXiv:2505.24832*, 2025.

- Maximilian Mozes, Xuanli He, Bennett Kleinberg, and Lewis D Griffin. Use of llms for illicit purposes: Threats, prevention measures, and vulnerabilities. *arXiv preprint arXiv:2308.12833*, 2023.
- USVSN Sai Prashanth, Alvin Deng, Kyle O’Brien, Jyothir SV, Mohammad Aflah Khan, Jaydeep Borkar, Christopher A Choquette-Choo, Jacob Ray Fuehne, Stella Biderman, Tracy Ke, et al. Recite, reconstruct, recollect: Memorization in lms as a multifaceted phenomenon. *arXiv preprint arXiv:2406.17746*, 2024.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zachary Lipton, and J Zico Kolter. Rethinking llm memorization through the lens of adversarial compression. *Advances in Neural Information Processing Systems*, 37:56244–56267, 2024.
- Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*, 2024.
- The New York Times Company v. Microsoft Corporation. The new york times company v. microsoft corporation, 2023. URL <https://www.courtlistener.com/docket/68117049/the-new-york-times-company-v-microsoft-corporation/>. The New York Times Company v. Microsoft Corporation, 1:23-cv-11195, (S.D.N.Y.).
- Trishita Tiwari and G Edward Suh. Sequence-level leakage risk of training data in large language models. *arXiv preprint arXiv:2412.11302*, 2024.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- Tremblay v. OpenAI, Inc. Tremblay v. openai, inc., 2023. URL <https://www.courtlistener.com/docket/67538258/tremblay-v-openai-inc/>. 3:23-cv-03223, (N.D. Cal.).
- Xinyi Wang, Antonis Antoniadis, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, Kexun Zhang, and William Yang Wang. Generalization vs memorization: Tracing language models’ capabilities back to pretraining data. In *The Thirteenth International Conference on Learning Representations*, 2025.
- Weichen Yu, Tianyu Pang, Qian Liu, Chao Du, Bingyi Kang, Yan Huang, Min Lin, and Shuicheng Yan. Bag of tricks for training data extraction from language models. In *International Conference on Machine Learning*, pp. 40306–40320. PMLR, 2023.
- 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.
- Jie Zhang, Debeshee Das, Gautam Kamath, and Florian Tramèr. Membership inference attacks cannot prove that a model was trained on your data. *arXiv preprint arXiv:2409.19798*, 2024.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Exact Copy	Near-Dup
Quantum doughnuts might not exist, but theoretical bakers remain hopeful.	majesticum Nantonuts might Conradavery 258 texted theoretical imperialistmlicks Shim.

Table 2: A sequence and one possible 20% near-duplicate. Matching tokens are highlighted

A Proof for Theorem 2

Theorem 2. $Var[\hat{v}_s] = \frac{1}{c} \cdot Var[P(s | p_i)] \leq \frac{1}{4c}$

Proof. Since p_i are independent, we can reduce:

$$\begin{aligned}
Var[\hat{v}_s] &= Var\left[\frac{1}{c} \sum_{i=1}^c P(s | q_i)\right] \\
&= \frac{1}{c^2} Var\left[\sum_{i=1}^c P(s | q_i)\right] \\
&= \frac{1}{c^2} \sum_{i=1}^c Var[P(s | q_i)] \\
&= \frac{1}{c^2} \cdot c \cdot Var[P(s | q_1)] \\
&= \frac{1}{c} \cdot Var[P(s | p_i)]
\end{aligned}$$

From Popoviciu’s inequality on variances,

$$Var[\hat{v}_s] = \frac{1}{c} \cdot Var[P(s | p_i)] \leq \frac{1}{4c}$$

As in Theorem 1, the transition from conditioning on q_1 to p_1 follows from the equality of their distributions. $\square$

B Correlation Experiments with Synthetic Data

Synthetic Data We train 124M parameter GPT-2 models (Radford et al., 2019) with 1000 Wikitext (Merity et al., 2016) documents in two settings, as described below. We repeat the experiments in each setting several times with different seeds and target sequences, training over 350 different models.

Target Model. In this setting, we inject “near-duplicate” sequences into the dataset, a strategy commonly used to study memorization versus generalization in prior work (Zhang et al., 2023; Liu et al., 2025). For our purposes, near-duplicates are defined as sequences sharing 20% token overlap, as illustrated in Table 2. Unlike earlier studies, which often used duplicates with much higher overlap, we intentionally adopt a lower threshold to be conservative. This choice is motivated by recent findings that removing high-overlap sequences improves model performance (Lee et al., 2022). Thus, injecting highly overlapping sequences may not realistically reflect modern training regimes, where such examples are typically removed (Zhang et al., 2022; Touvron et al., 2023). Our goal is to demonstrate that even with relatively low overlap, models are capable of generalizing to a given sequence without requiring many (or any) exact copies in the training data.

To simulate varying degrees of generality, we vary the ratio of exact copies to near-duplicates of a target sequence injected into the training data. For example, an initial dataset may contain 180 near-duplicates of a target sequence and no exact copies. In the next dataset, 30 of these near-duplicates are replaced with 10 exact copies, yielding 150 near-duplicates and 10 exact matches. The rationale is that as more exact copies are included, the model is more likely to exhibit Counterfactual memorization, predicting the target sequence because

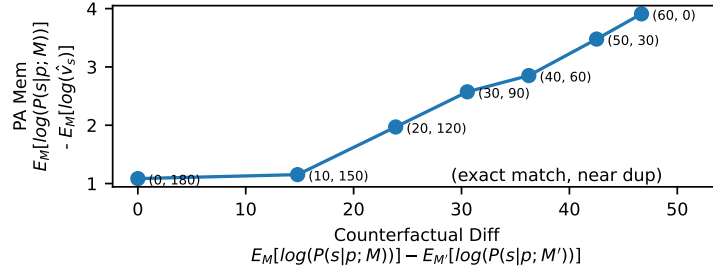


Figure 6: Counterfactual Memorization (x-axis) vs PA Memorization (y-axis). Each data point averages over models with the same frequency of exact matches and near duplicates.

it has seen it verbatim during training. Conversely, when the dataset consists mostly of near-duplicates, the model must rely on generalization from many similar—but not identical—examples, simulating the case where a sequence is generated with high probability simply because there is a lot of similar data in the training set (Liu et al., 2025). By carefully controlling the ratio of exact copies to near-duplicates, we can precisely modulate the extent to which the model relies on memorization versus generalization in order to reproduce the target sequence.

We use the following number of (exact copies, near-duplicates) to create 7 distinct types of datasets: (0, 180), (10, 150), (20, 120), (30, 90), (40, 60), (50, 30), (60, 0). In every case, the total training set size remains fixed at 1000 sequences, with only the composition of exact versus near-duplicate instances varying. We repeat this process for 25 different target sequences, training $25 \times 7 = 175$ different target models.

Baseline Model. The Counterfactual memorization metric relies on a comparison between the target model and a baseline model. In accordance with the metric’s definition of a baseline model, we remove only the exact copies of the target sequence, but keep the near duplicates in the training data. So the baseline model for a target model trained with 150 near-duplicates and 10 exact copies would have only the 150 near duplicates. This is to simulate the scenario where the exact target data might be removed, but other data that can generalize to the target sequence is still present in the training dataset.

Results Figure 6 shows the Counterfactual Memorization metric (Equation 6) on the x-axis, and $\frac{P(s|p)}{\hat{\vartheta}_s}$, the component of PA memorization that measures the commonality of s (Equation 7) on the y-axis. Each data point represents the average over 25 different models trained. Figure 6 illustrates that as the target sequence becomes less generic due to fewer near-duplicates, the Counterfactual Memorization metric (Equation (4)) increases correspondingly. As hypothesized, we also observe a strong correlation between PA memorization and Counterfactual Memorization, suggesting that our metric indeed measures generalization from similar data.

C Known Statistically Common Sequences for Threshold Calculation

In this section, we provide a list of some short and long sequences that are known to be statistically common. We use these to calculate the threshold n in Definition 2. The short sequences are used to calculate the threshold for the short, 4-token suffixes, while the long sequences are used to calculate the threshold for the longer 50-token suffixes. In each case, each sequence is divided into equal tokens of prefix and suffix used to calculate $\frac{P(s|p)}{P(s)}$. Recall that n is simply an average of $\frac{P(s|p)}{P(s)}$

- Please let me know if you have any questions,
- Thank you for your time and consideration,
- I look forward to hearing from you soon,

- If you have any concerns, please don't hesitate to contact me,
- Don't miss out on this limited-time offer,
- Our mission is to provide the best service,
- Learn more about our features and pricing options,
- Thank you for taking the time to read this message. I am writing to provide an update regarding our project and to ensure that you have all the information needed to move forward. At this stage, we have completed the initial steps and are preparing to begin the next phase. Please review the attached document, which includes a summary of progress, outstanding tasks, and anticipated challenges. If you have any questions or concerns, feel free to reach out at your earliest convenience.,
- This email is intended to confirm the details of our upcoming meeting. The session is scheduled for next week at the agreed time, and we will be covering several important items on the agenda. Please make sure to review the notes shared earlier so that we can make efficient use of our time together. If you are unable to attend, kindly let me know as soon as possible so we can reschedule or provide you with the necessary updates.
- I hope this message finds you well. I wanted to follow up on the status of the work assigned last month and check whether everything is on track for completion by the deadline. If you are facing any obstacles or require additional support, please do not hesitate to reach out. It is important for us to address any potential challenges early so that the overall timeline remains achievable.,
- In this article, we will discuss the basics of this topic and explain why it is important to understand this concept in the broader context of the field. We will begin with a brief introduction, followed by a step-by-step explanation of the main ideas, and finally provide some examples to illustrate how these principles can be applied in practice. Whether you are completely new to this topic or simply looking for a refresher, this guide is designed to help you gain a clear and practical understanding.,
- The purpose of this guide is to help you get started with the software in a straightforward and beginner-friendly way. First, we will walk through the installation process and highlight common issues that may arise during setup. Then we will cover the essential features you need to know to be productive right away. By the end of this guide, you should have a working environment and a solid foundation that will allow you to explore more advanced aspects of the software at your own pace.,

C.1 PA Memorization Examples

In this section, we provide examples of $p||s$ that have statistically likely s and those that do not. Each sequence is preceded by its value of $\frac{P(s|p)}{P(s)}$, presented in colored text. The suffix s is presented in bold text. We note that low-scoring sequences have more natural-appearing english text, which is common in the training data for both OPT and Llama models. However, high-scoring sequences tend to have excessive formatting, non-english characters, URLs, are about niche topics, or are boiler-plate text (e.g., terms and conditions from companies) that appear verbatim several times on the web.

C.2 Low $\frac{P(s|p)}{P(s)}$ sequences

- 1.48 ining the ship after this crossing." "That's not a problem," said Emma without explanation. The purser still didn't look convinced. "Can you read and write?" Emma would like to have told him that **she'd won a scholarship to Oxford, but simply said, "Yes, sir."** Without another word, he pulled open a drawer and extracted a long form, passed her a fountain pen and said, "Fill this in." As Emma began

- **1.48** “pageset”: “S53 476 (Tenn. 1973). Our supreme court stated the rationale for this rule: This well-settled rule rests **on a sound foundation. The trial judge and the jury see the witnesses face to face, hear their testimony and observe their demeanor on the stand. Thus the trial judge and jury are**
- **1.55** obvious) mistake in being unacceptably slow getting back into the play after the puck was cleared over his head into the neutral zone. He rather glided through the neutral zone as Moore passed him by to get in position to collect the loose puck. **In a cruel twist, the Caps would get one back with 2:16 left when – finally – someone was able to collect a loose puck in close and do something with it. Brooks Laich lifted a backhand over Halak almost from**
- **1.68** “Pay her off easy, easy,” he screamed to the sailors struggling at the wheel, his voice barely audible over the wind. He squinted into the blinding snow in an attempt to time the turn in a smooth, a patch of **waves smaller than the average. The seas came in groups, with every seventh to ninth attaining the most prodigious heights. Sometimes these larger groups merged with a cross sea reared up from the swift tidal currents, or married together to form a rogue**
- **2.05** fluffy, and great). When cooking first sparked my interest in a real way, I decided I needed to master the biscuit. Not coming from a family of biscuit makers, or from a place known for excellent biscuits, I wasn’t **even sure what made a great biscuit, if we are being honest. But I was on a quest for perfection.**
- **2.06** The people of Thebes acclaimed Oedipus as their new king. Monster Mash The Sphinx, the Greeks said, had the head of a woman, the body of a lion, the wings of an eagle, and the **tail of a serpent. In fact, this is how the Greeks dreamed up most of their mythical creatures: by mixing and matching body parts from real animals, like a hideous, nightmarish version of Build-A-Bear Workshop. Let’s take**
- **2.21** up sinks: cold water Washing machine(s) Laundry sinks: cold water Laundry sinks: hot water Dryers Ironing facilities Food, drink and groceries **Fresh bread available at the camp site Groceries: limited selection Restaurant (with ample choice) Snack bar Takeaway meals Bar Communal barbecue area Freezing for cooling**

C.3 High $\frac{P(s|p)}{P(s)}$ sequences

- **10.32** u7684 u6B65 u9AA4u3002 u5FAE u4FE1 u5C0Fu7A0B u5E8F u5BF9 u4E8Eu767B u5F55 u7684 u8BBEu8BA1 uFF0C u66F4 u662Fu7528 u4E4B u4E8E u65E0u5F62 **uFF0C u5728 u6574 u4E2A u7528 u6237 u4F7F u7528 u8FC7 u7A0B u4E2D u90FD u662F u65E0 u611F u77E5 u7684 uFF0C u4E00 u8FDB u5C0F u7A0B u5E8F u5176 u5B9E**
- **7.47** money, now banks are closed...” “Now I don’t have cash with me...” “Anto’, how much have you got?” “Not a single lira, love.” “You see...” “What can we do now?” “Don’t you worry, it does not matter.” “We’ll do as we have always done.” “Here is a million for you!” “It’s too much, madam.” “Thanks, madam.” “No, you must take it.” “
- **6.43** /1332) :star: - Learning Simple Algorithms from Examples. [‘pdf’](http://proceedings.mlr.press/v48/zaremba16.pdf) - Learning to Trans **duce with Unbounded Memory.** [‘arxiv’](https://arxiv.org/pdf/1506.02516.pdf) - Listen, Attend and Spell. [‘arxiv’](https://
- **4.59** be issued to the original card that was used. The refund amount will include only the amount paid by you after any discount or reward was applied to the returned item(s) and it will not include any shipping charge paid by you unless you are returning **a damaged, defective, or the wrong item was sent to you. The wedding dress is perhaps the most carefully chosen dress a woman will ever wear, and an amazing wedding ceremony could be the most unforgettable moment in a woman’s life.**

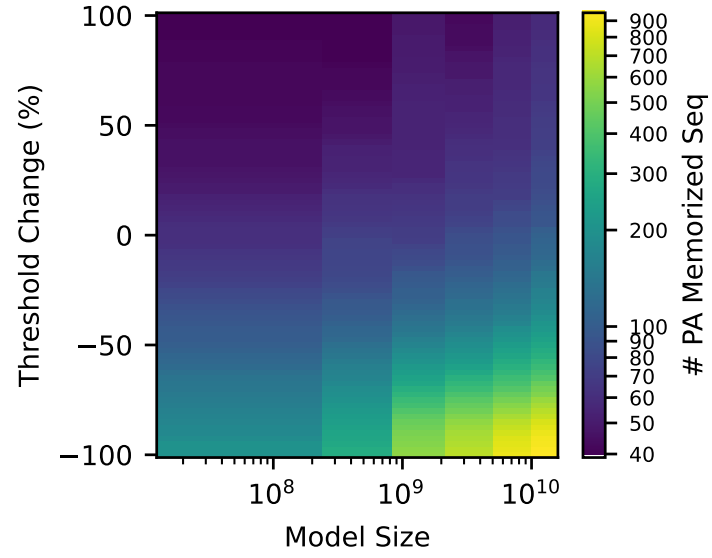
- 3.99 surface and prevent the possibility of sustaining burns unnecessarily while shaving. How can I prevent skin burns? Yes, you can. You have to adopt and adhere to the following best practices to be able to accomplish this particular feat: **Shave in the right Direction: In most instances, this problem arises when the hair is shaved against the direction of the strands thereof. To avoid this problem, it is advisable that you shave in the direction of the hair growth. This is**
- 3.24 ,006 (u5468u5B50) u3061u3087 u3063 u3068 uFF61— u3069 u3046 u3057 u305F u306E uFF1F— u3046 u3093 uFF1F—,011 u4F1A u793E u306B u884C u304B u306A u304D u3083 uFF61 (u5468 u5B50) u3044 u3044 u304B u3052 u3093 u306B u3057 u306A u3055u3044uFF61 u27A1241 00:
- 2.78 s costing to fill the car and get groceries! <http://no-apologies-round2.blogspot.com/> AmericanborninCanada lol wolfie. Are you a foghorn like I am? [http:// tinyurl.com/wwsotu](http://tinyurl.com/wwsotu) Wolfie ON my good days! If I went on American Idol, they'd just pay me the top prize to never come back again! However, my wife is a lovely singer and
- 2.67 , offices, and hotel). The mine-site and township were heavily polluted with asbestos and tailings from the mine were distributed around the town (5). These women have been followed up since 1973 at national and state death and cancer registries as well as frequent questionnaires to ascertain smoking histories and demographic information. Their mortality and cancer incidence have been reported elsewhere (6-8). The studies have ethics approval from the University of Western Australia Human Research Ethics Committee. A disease cluster is an occurrence
- 2.65 "parent": "z9iGIu1Pt", "text": "", "mid": "z9iZtx1BP", "date": "2012-12-11 22:35:08"}, {"kids": [], "uid": "1927311555", "parent": "z9iJ1w3Q7", "text": "", "mid": "z9iZjcvK8", "date": "2012-
- 2.52 that $N^{\{\lambda\}} = \emptyset$ and let $n \in \mathbb{N}_{>0}$ be minimal such that $N^{\{\lambda - n\alpha\}} \neq \emptyset$. Let $w \in N^{\lambda - n\alpha}$ be a non-zero element. Using the PBW Theorem, we may write $w = \sum_{i=0}^{\infty} c_i \overline{f}^i$
- 2.47 , pacing around a long chart unrolled like a hound's tongue across the floor. "We've backed away from the pedigrees in the past year." Spencer's shoe brushes the edge of the pedigree chart. It is a long genetic octopus, a family tree with arms and legs that tangle and cross, as do those of most degenerate families. Spotted throughout are symbols, keyed on the side. A dark black circle signifies Insane. A hollow

D Sensitivity Analysis of Threshold n

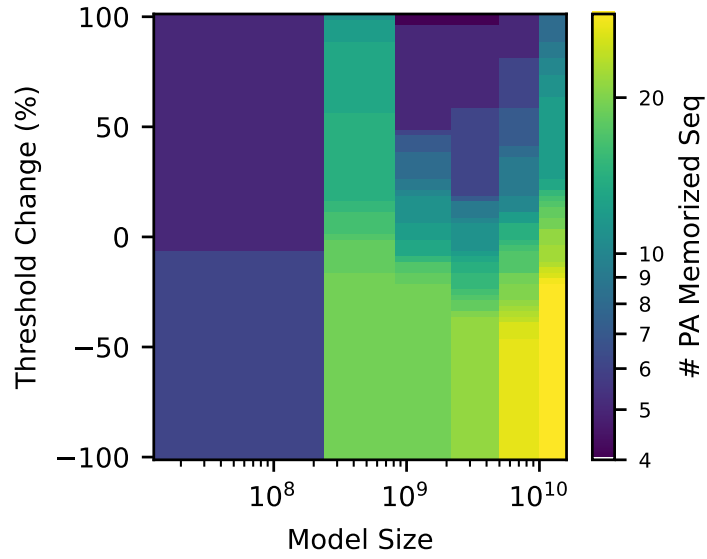
Figure 7 illustrates how the number of PA-memorized sequences varies with the relative belief ratio threshold, n . To ensure a robust baseline, recall that we dynamically calibrate n for each model size by computing $\frac{P(s|p)}{\delta_s}$ over ground-truth sequences known to be highly predictable (refer to Appendix C). We observe that for 4-token suffixes, the number of flagged sequences remains stable until n is reduced by more than 50% from its calibrated value. For 50-token suffixes, this stability extends across all tested values of n , with the absolute number of PA-memorized sequences remaining uniformly low (varying only between 4 and 29).

E LLM Usage

LLMs were used for fundamental background research and polishing wording for grammar and clarity.



(a) 4 token suffixes



(b) 50 token suffixes

Figure 7: Sensitivity analysis of the threshold n across various OPT model sizes for both 4 token suffixes (a), and 50 token suffixes (b). The heatmap displays the PA-memorization rate as the per-model threshold n is varied from -100% to 100% of its original value.