Sampling-based Pseudo-Likelihood for Membership Inference Attacks

Masahiro Kaneko[1,2*] **Youmi Ma**[2*] **Yuki Wata**[3] **Naoaki Okazaki**[2]
[1]MBZUAI [2]Institute of Science Tokyo [3]The University of Tokyo
Masahiro.Kaneko@mbzuai.ac.ae ma.y@comp.isct.ac.jp
uk1@is.s.u-tokyo.ac.jp okazaki@comp.isct.ac.jp

Abstract

Large Language Models (LLMs) are trained on large-scale web data, which makes it difficult to grasp the contribution of each text. This poses the risk of leaking inappropriate data such as benchmarks, personal information, and copyrighted texts in the training data. Membership Inference Attacks (MIA), which determine whether a given text is included in the model's training data, have been attracting attention. Previous studies of MIAs revealed that likelihood-based classification is effective for detecting leaks in LLMs. However, the existing likelihood-based methods cannot be applied to some proprietary models like ChatGPT or Claude 3 because the likelihood for input text is unavailable to the user. In this study, we propose a Sampling-based Pseudo-Likelihood (**SPL**) method for MIA (**SaMIA**) that calculates SPL using only the text generated by an LLM to detect leaks. The SaMIA treats the target text as the reference text and multiple outputs from the LLM as text samples, calculates the degree of n-gram match as SPL, and determines the membership of the text in the training data. Even without likelihoods, SaMIA performed on par with existing likelihood-based methods[1].

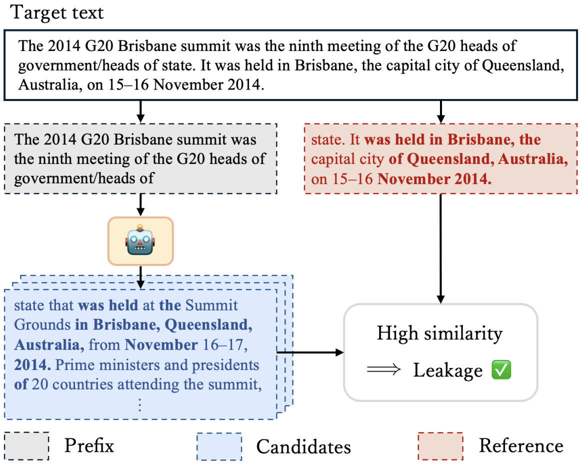

Target text
The 2014 G20 Brisbane summit was the ninth meeting of the G20 heads of government/heads of state. It was held in Brisbane, the capital city of Queensland, Australia, on 15–16 November 2014.
The 2014 G20 Brisbane summit was the ninth meeting of the G20 heads of government/heads of
state. It **was held in Brisbane, the** capital city **of Queensland, Australia,** on 15–16 **November 2014.**
state that **was held** at **the** Summit Grounds **in Brisbane, Queensland, Australia,** from **November** 16–17, **2014.** Prime ministers and presidents **of** 20 countries attending the summit,
⋮
High similarity
⟹ Leakage ✅
Prefix
Candidates
Reference

Figure 1: MIA using SPL based on the degree of n-gram between sampled candidate texts and a reference text.

1 Introduction

Large Language Models (LLMs) bring about a game-changing transformation in various services used on a daily basis (Brown et al., 2020; Touvron et al., 2023). The pre-training of LLMs relies on massive-scale web data of mixed quality (Zhao et al., 2023). While pre-processing such as filtering is applied to construct as clean datasets as possible, it is unrealistic to remove everything undesired (Almazrouei et al., 2023). There is a risk of unintentionally leaking benchmark data, copyrighted texts, or personal information into the pre-training data (McCoy et al., 2023; Ippolito et al., 2023; Kaneko and Baldwin, 2024). The leakage of benchmark data can lead to an overestimation of LLMs' capabilities, preventing an appropriate assessment of their true performance (Yu et al., 2023; Zhou et al., 2023). Additionally, LLM generation based on copyrighted texts or personal information can result in serious violations of the law (Yeom et al., 2017; Eldan and Russinovich, 2023).

Membership Inference Attacks (MIA) consider the task of determining whether a given target text is included in the training data of a model (Shokri et al., 2016). Generally, because models are trained to fit the data, a text included in the training data tends to exhibit a higher likelihood compared to ones unseen in the training data (Yeom et al., 2017). Existing MIA studies rely on this idea and thus require the likelihood of a input text computed by the model (Carlini et al., 2021; Kandpal et al., 2022; Ye et al., 2022; Mattern et al., 2023; Shi et al., 2023). It is impossible to apply the existing studies to the models that do not provide a likelihood.

When the training data of a model is accessible,

[*] These authors contributed equally to this work.
[1] Our code is available at: https://github.com/nlp-titech/samia

Findings of the Association for Computational Linguistics: ACL 2025, pages 8894–8907
July 27 - August 1, 2025

it is straightforward to check the membership by directly searching for a given text in the training data (Zhang et al., 2022; Biderman et al., 2023; Wang and Komatsuzaki, 2021). Therefore, the primary targets for MIA are LLMs without the training data disclosed (Zhang et al., 2022; Biderman et al., 2023; Touvron et al., 2023). Such LLMs mostly provide no access to likelihoods, severely limiting the practical applicability of likelihood-based MIA. Moreover, as the commercial value of LLMs is increasing, developers of recent powerful models like ChatGPT[2], Gemini[3], and Claude 3[4] are more reluctant to disclose the details of their training data.

In this paper, we propose a Sampling-based Pseudo Likelihood (**SPL**) method for MIA (**SaMIA**) that calculates SPL using the match ratio of n-grams between the output texts sampled from an LLM and the target text. We provide proof that the match rate of sampled outputs serves as an approximation of likelihood. Figure 1 shows the detection of a leaked text using SPL, based on the n-gram similarity between text samples and the reference text. Providing the initial part of the target text to the LLM, SaMIA generates multiple continuations of the text by sampling. Treating the generated sequences as candidate texts and the remaining part of the target text (after the given part) as the reference text, we calculate the degree of n-gram overlaps between the candidate texts and the reference text. If the degree of n-gram overlaps exceeds a specified threshold, we regard that the LLM has been trained on the text. SaMIA can detect leakage even under settings where the target model is a black box (Ishihara, 2023).

We conducted experiments on MIA with OPT-6.7B (Zhang et al., 2022) and LLaMA-2-7B (Touvron et al., 2023) whose training data is publicly available. The experimental results on the Wikipedia articles (Shi et al., 2023) and book (Duarte et al., 2024) datasets showed that SaMIA achieves performance comparable to existing methods based on likelihood or loss values. In addition, we introduce a leakage detection method that combines information content and SPL, which achieves the highest average score. We also report the analysis results of the impacts of the number n of n-gram, the number of text samples, and the length of the target text on the performance of SaMIA. The performance of SaMIA is highest when using unigrams. We observed the improved performance by increasing the number of text samples and length of the target text.

2 SaMIA

2.1 Definition of the MIA Task

MIA is a binary classification task to determine whether a target text x is included in the training dataset $\mathcal{D}_{\text{train}}$ of a model f_θ. The attacker's goal is to design an appropriate attack function $A_{f_\theta} : \mathcal{X} \rightarrow \{0, 1\}$ and to determine the truth value of $x \in \mathcal{D}_{\text{train}}$ for an instance x in the text space $\mathcal{X}$. Motivated by detecting copyrighted text and privacy information (Zhao et al., 2022), this paper targets verbatim text memorization.

2.2 SPL

Our method is applicable under harder conditions than the previous studies, which requires the loss $\mathcal{L}$ or token likelihood P_θ of the model f_θ. In other words, our method can be applied to any LLMs because it uses only generated texts without the loss or likelihood. The proposed method is formalized as follows. For a text $x = (w_1, w_2, \ldots, w_T)$ of length T to be detected, we divide it into a prefix $x_{\text{prefix}} = (w_1, w_2, \ldots, w_{\lfloor T/2 \rfloor})$ and a reference text $x_{\text{ref}} = (w_{\lfloor T/2 \rfloor+1}, w_{\lfloor T/2 \rfloor+2}, \ldots, w_T)$ based on the number of words T. The LLM then generates m text samples (*candidates* hereafter) $x^j_{\text{cand}} (j = 1, \ldots, m)$ that continue from the prefix x_{prefix}. We use these candidates for MIA.

SaMIA judges that a text x is included in the training data of the LLM if the candidate text x^j_{cand} generated by the LLM has a high surface-level similarity to the reference text x_{ref}. We use ROUGE-N (Lin, 2004) as the similarity metric, which measures the recall of n-grams in the reference text. Theoretical proofs that ROUGE-N approximates the likelihood yielded are shown in Appendix E[5].

Given a candidate text x_{cand} generated by the LLM and a reference text x_{ref}, we calculate the ROUGE-N score (ranging from 0 to 1):

$$\text{ROUGE-N}(x_{\text{cand}}, x_{\text{ref}}) = \frac{\sum_{\text{gram}_n \in x_{\text{ref}}} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{\text{gram}_n \in x_{\text{ref}}} \text{Count}(\text{gram}_n)} \tag{1}$$

Here, n is the length of the n-grams. The denominator is the total number of n-grams in the

[2] https://chat.openai.com/

[3] https://gemini.google.com/app

[4] https://claude.ai/chats

[5] Semantic similarity metrics like BERTScore (Zhang et al., 2019) are inconsistent with this objective, and our pilot studies showed they underperform n-gram-based metrics.

Algorithm 1 Sampling-based Membership Inference Attacks

```
Input: target text $x = (w_1, w_2, \dots, w_T)$, language model $f_\theta$, number of samples $m$, length of $n$-gram $N$, threshold $\tau$, zlib compression flag $z$
Output: is text $x$ included in the training data of $f_\theta$? (1 or 0)
$x_{\text{prefix}} \leftarrow (w_1, w_2, \dots, w_{\lfloor T/2 \rfloor})$
$x_{\text{ref}} \leftarrow (w_{\lfloor T/2 \rfloor + 1}, w_{\lfloor T/2 \rfloor + 2}, \dots, w_T)$
for $j = 1$ to $m$ do
    $x^j_{\text{cand}} \leftarrow f_\theta(x_{\text{prefix}})$    ▷ The $j$-th candidate text $x^j_{\text{cand}}$ generated from $f_\theta$ using $x_{\text{prefix}}$ as the prompt
end for
if $z = 1$ then
    $\overline{\mathrm{R}}_m \leftarrow \frac{1}{m} \sum_{j=1}^{m} \text{ROUGE-N}(x^j_{\text{cand}}, x_{\text{ref}}) \cdot \text{zlib}(x^j_{\text{cand}})$    ▷ Combine with zlib compression
else
    $\overline{\mathrm{R}}_m \leftarrow \frac{1}{m} \sum_{j=1}^{m} \text{ROUGE-N}(x^j_{\text{cand}}, x_{\text{ref}})$
end if
if $\overline{\mathrm{R}}_m > \tau$ then
    return 1
else
    return 0
end if
```

reference text, and the numerator is the total number of n-grams that overlap between the candidate text and the reference text. For example, a ROUGE-1 score is high when the words generated by the LLM appear in the reference text.

We expect that SPL can appropriately capture the distribution of LLM's generations by empirically sampling texts from the LLM. Let W be a random variable defined by the LLM, and let $P(W = x)$ be the probability that W takes the text x. Let $X_x^{(j)}$ be a random variable such that $X_x^{(j)} = 1$ if the j-th sampled sequence from the language model is x, and $X_x^{(j)} = 0$ otherwise.

$$X_x^{(j)} = \begin{cases} 1 & \text{if the } j\text{-th sampled sequence is } x \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The expected value of $X_x^{(j)}$ is calculated as follows:

$$\begin{aligned} \mathbb{E}[X_x^{(j)}] &= 1 \cdot P(X_x^{(j)} = 1) + 0 \cdot P(X_x^{(j)} = 0) \\ &= P(X_x^{(j)} = 1) \end{aligned} \tag{3}$$

Here, the probability that $X_x^{(j)} = 1$ equals the probability that the language model generates the text x, i.e., $P(W = x)$. When the sample size N is sufficiently large, the average of $X_x^{(j)}$, which is the relative frequency of the text x in the j-th sample, will converge to the expected value by the law of large numbers, and become equivalent to $P(W = x)$.

$$\lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} X_x^{(i)} = \mathbb{E}[X_x^{(i)}] = P(W = x) \tag{4}$$

Therefore, by sampling the texts generated by the LLM, the appearance frequency of sampled texts approximates the probability that the language model generates a text. Thus, calculating the similarity between sampled texts and a target text can reflect the output tendencies of the LLM.

2.3 Detection by SaMIA Based on SPL

SaMIA calculates the average ROUGE-N score between each candidate text x^j_{cand} generated by the LLM and the reference text x_{ref}. If this average exceeds a threshold τ, the text x is considered a member of the LLM's training data (Equation 5).

$$A_{f_\theta}(x) = \mathbb{1}\left[\frac{1}{m} \sum_{j=1}^{m} \text{ROUGE-N}(x^j_{\text{cand}}, x_{\text{ref}}) > \tau\right] \tag{5}$$

Here, $\mathbb{1}[\cdot]$ is an indicator function, and m is the sample size. We set the threshold τ based on the development data. The interpretation of this detection metric is straightforward: if a text x were used in training, the LLM would generate many of the n-grams present in the reference text x_{ref}.

2.4 Improving SaMIA With Zlib Compression

Existing methods such as PPL/zlib use the information content computed by zlib compression to evaluate the redundancy characteristics of the generated text (Carlini et al., 2021). Samples from unseen data in training tend to contain repetitive generation (e.g., *"I love you. I love you..."*), and the information content of such samples after zlib compression is expected to be lower. PPL/zlib uses the ratio of the perplexity of the sample x and the bit length of x after zlib compression, $\text{zlib}(x)$, as the detection metric (Equation 6).

$$A_{f_\theta}(x) = \mathbb{1}\left[\frac{\prod_{i=1}^{n} P_\theta(x_i|x_{1:i-1})^{-\frac{1}{n}}}{\text{zlib}(x)} < \tau\right] \quad (6)$$

Here, $P_\theta(x_i|x_{1:i-1})$ is the model's likelihood of generating token x_i given $x_{1:i-1}$ and $\text{zlib}(x)$ represents the entropy of x in bits after zlib compression.

Since $\text{zlib}(x)$ depends only on the character information of the text x, it can also be applied to SaMIA. Additionally, SaMIA does not consider the presence of repetitive generation in the candidate texts x^j_{cand}. Therefore, combining it with zlib can be expected to improve performance:

$$A_{f_\theta}(x) = \mathbb{1}\left[\frac{1}{m}\sum_{j=1}^{m}\text{ROUGE-N}(x^j_{\text{cand}}, x_{\text{ref}})\cdot\text{zlib}(x^j_{\text{cand}}) > \tau\right] \quad (7)$$

Algorithm 1 presents the detail of the whole process of the proposed method.

3 Experiments

3.1 Existing Methods

We compare SaMIA with several existing methods to evaluate the effectiveness of our proposed method. Our goal is to propose a theoretically-grounded detection approach using sampling-based approximation that achieves comparable performance under limited information access. The competitors range from basic to edge-cutting MIA approaches as follows.

We compared SaMIA with several existing methods to evaluate the effectiveness of our proposed approach. The comparisons range from basic to cutting-edge MIA approaches that require access to likelihoods, as listed below. Note that our goal is to propose a theoretically-grounded detection method using sampling-based approximation that achieves comparable performance under limited information access compared to existing methods, rather than outperforming them.

LOSS (Yeom et al., 2018) The most straightforward MIA method, attacking by thresholding the Negative Log-Likelihood (NLL) loss. Given a target text x, if model f_θ yields an NLL loss lower than a threshold τ, then x is considered present in the training data of model f_θ.

Lowercase (Carlini et al., 2021) The method extends **LOSS** by thresholding the difference between a target text x and its lowercase version x_{lower}.

PPL/zlib (Carlini et al., 2021) The method alleviates the effect of repeated substrings via zlib compression[6]. PPL/zlib decides if a target text x is included in the training data of model f_θ by thresholding the ratio of the log of the perplexity and the zlib entropy (Equation 6).

Min-$k\%$ (Shi et al., 2023) The method utilizes only a subset of each target text x during an attack. Specifically, tokens in x are sorted in the ascending order of the log-likelihood, and the average log-likelihood of the first $k\%$ tokens, composing a token set $\mathcal{E}$, are used for detection:

$$A_{f_\theta}(x) = \mathbb{1}\left[\frac{1}{|\mathcal{E}|}\sum_{x_i\in\mathcal{E}}\log P_\theta(x_i|x_{1:i-1}) > \tau\right]. \quad (8)$$

Following findings in Shi et al. (2023), we set $k = 20$ for Min-$k\%$ in our experiments. **Min-$k\%$++ (Zhang et al., 2024b)**, an improved version of Min-$k\%$, is also employed as the strongest and most recent likelihood-dependent baseline.

3.2 Settings

Benchmarks Following existing work, we adopt WikiMIA (Shi et al., 2023) and BookTection (Duarte et al., 2024) as the benchmarks. WikiMIA contains texts from Wikipedia event pages. Texts from pages created after 2023 are considered unleaked data (i.e., text excluded from the training data of LLMs), and those from pages created before 2017 are considered leaked data (i.e., text included in the training data of LLMs). BookTection consists of 165 books, operating on the principle that books published post-2023 are definitively unleaked data, whereas those published before or during 2021 may potentially be leaked data. Books from 2022 are not considered due to the ambiguity surrounding some models' exposure to content from that year.

Models To properly evaluate MIA methods, knowledge of the training data is essential. Since training data is not accessible for blackbox LLMs such as GPT-4 and Gemini, we conduct experiments with whitebox LLMs. We evaluate the performance of SaMIA as well as existing MIA methods against LLMs including **OPT-6.7B** (Zhang et al., 2022), and **LLaMA-2-7B** (Touvron et al., 2023). Their checkpoints are publicly available on Huggingface[7]. We present an exposition regarding the training data for the LLMs to show the

[6] https://www.zlib.net/

[7] https://huggingface.co/

	WikiMIA		BookTection	
	OPT	LLaMA	OPT	LLaMA
Existing Methods (likelihood-dependent)				
LOSS	8.38	7.23	10.83	**17.97**
PPL/zlib	8.18	9.83	10.27	15.30
Lowercase	11.38	5.58	13.50	6.13
Min-K%	15.20	5.18	10.87	14.87
Min-K%++	10.90	7.93	9.10	13.40
Proposed Methods (likelihood-independent)				
SaMIA	20.38	9.25	11.03	8.20
SaMIA*zlib	**26.28**	**12.18**	**14.13**	15.27

Table 1: Macro-averages of TPR@5%FPR of the proposed and existing methods when performing MIA on various LLMs using WikiMIA and BookTection. The best score of each column is **bolded**.

	OPT-6.7B				LLaMA-2-7B			
Length	32	64	128	256	32	64	128	256
Existing Methods (likelihood-dependent)								
LOSS	0.61	0.57	0.62	0.64	0.55	0.50	0.56	0.59
PPL/zlib	0.61	0.58	0.64	0.65	0.55	0.51	0.57	0.59
Lowercase	0.58	0.57	0.57	0.59	0.49	0.50	0.49	0.59
Min-K%	0.62	0.60	0.67	0.67	0.51	0.50	0.56	0.58
Min-K%++	**0.65**	0.65	0.69	0.64	**0.62**	**0.60**	0.60	0.57
Proposed Methods (likelihood-independent)								
SaMIA	0.55	0.63	0.66	**0.80**	0.53	0.55	0.60	**0.71**
SaMIA*zlib	0.62	**0.68**	**0.70**	**0.80**	0.55	0.59	**0.63**	0.69

Table 2: AUC scores of the proposed and existing methods when performing MIA on various LLMs using WikiMIA. The best score of each column is **bolded**.

properness of WikiMIA as a benchmark for leakage detection. OPT (Zhang et al., 2022) is trained on the dataset containing the Pile[8], a dataset published in late 2020 comprising 800GB of text data[9]. Moreover, PushShift.io Reddit (Baumgartner et al., 2020) is used for pre-training. LLaMA-2 (Touvron et al., 2023) employs English CommonCrawl, C4, Github, Wikipedia, Books, ArXiv, and StackExchange as pre-training datasets. The model was released in 2023, but its pre-training data has a cutoff date of September 2022.

All models evaluated in this work are pre-trained on data ealier than 2022. Therefore, it is proper to treat Wikipedia pages created after 2023 as being excluded from the training data.

Implementation Details For both SaMIA and SaMIA*zlib, we use ROUGE-1 and set the number of candidate texts to 10, using different seeds. Analyses of the influence of these factors are detailed in Section 4. During generation, the hyper-parameters of all models are fixed as `temperature=1.0`, `max_length=1024`, `top_k=50`, `top_p=1.0`, following the default hyperparameters of Huggingface. For existing approaches, we report the scores obtained from our experiments.

Evaluation Metrics Both benchmarks group texts by length: 32, 64, 128, and 256 for WikiMIA and 64, 128, 256 for BookTection. On each group, we evaluate the performance of detection methods via the True Positive Rate (TPR) and the False Positive Rate (FPR).[10] We plot the ROC curve to measure the trade-off between TPR and FPR at each threshold τ and report the AUC score, i.e., the area under the ROC curve, and TPR@5%FPR, i.e., TPR when FPR is 5% (Mattern et al., 2023; Shi et al., 2023). TPRs at a lower and higher false positive rate are reported in Appendix B.

Computational Environment All experiments are carried out on a single Tesla V100 16GB GPU. We first generate candidate texts using target LLMs and then conduct MIA with existing and proposed methods. For each LLM, the inference process for the entire WikiMIA dataset takes approximately 48 GPU hours. Due to the computation cost of running inferences for multiple LLMs, we report the results based on a single run for each model[11].

3.3 Main Results

The macro average of TPR@5%FPR among all groups of WikiMIA and BookTection are reported in Table 1. AUC scores of the existing and proposed detection methods for each text length group are reported in Table 2. Additionally, the detection performance of SaMIA and existing methods are provided in Appendix 3.5.

On average, SaMIA*zlib, without accessing the likelihood, performs on-par with current state-of-the-art methods. As shown in Table 1, SaMIA*zlib performs on par or even better than Min-K%++, the method with state-of-the-art performance on MIA. The high performance is achieved without access to likelihoods computed

[8] https://huggingface.co/datasets/EleutherAI/pile

[9] Pile aggregates content from 22 sources, including Wikipedia, books, websites, and GitHub repositories.

[10] Because we use white-box LLMs with publicly available training data, we can directly verify whether a given instance was included in the training set. This ground-truth verification removes the need to constrain the FPR, unlike in black-box settings (Zhang et al., 2024a), where the training data is unknown.

[11] For the same reason, we downsample BookTection to keep the same size as WikiMIA.

from LLMs, highlighting the superiority of our proposed method. While SaMIA*zlib can be applied to any LLMs, existing methods are limited to LLMs with accessible likelihoods. Thus, the superiority of our proposed method includes not only better performance but also wider usage.

SaMIA outperforms existing methods when the target text is long. From Table 2, when evaluating the group with a text length of 256, we observe SaMIA outperforming all existing methods. For groups with shorter text lengths, SaMIA still performs comparably or better than its competitors. Notably, while all existing methods base their detection on the likelihood yielded from LLMs, SaMIA does not rely on the likelihood. SaMIA, therefore, has its superiority over other methods in achieving competitive or even better performance using less information.

SaMIA*zlib further improves SaMIA regardless of the text length. Incorporating SaMIA with zlib compression entropy further boosts the detection performance for all models, regardless of the length of target texts. The observation indicates that zlib compression benefits leakage detection by compressing the influence of repeated substrings.

3.4 Scalability w.r.t Parameter Size

Section 3.3 reports the effectiveness of our proposed methods on LLMs with approximately 7B parameters. In this section, we investigate how the effectiveness scales to larger models. Specifically, we experiment with LLaMA-2-13B and compare the detection performance against that of LLaMA-2-7B. The results are reported in Table 3. Consistent with the results observed on LLaMA-2-7B, SaMIA continues to demonstrate comparable or superior performance relative to likelihood-based methods. This underscores the robustness of SaMIA across different model sizes.

3.5 Detection Performance

This section compares the detection accuracy of the existing and proposed methods during inference. The testbed is WikiMIA following a 5-fold cross-validation setting, with the average over 5 splits reported in Table 4. The threshold that maximizes the gap between the true and false positive rates is selected as the threshold during inference. As in the table, **SaMIA*zlib exhibits comparable or even performance in inference with those likelihood-based methods**.

	LLaMA-2-7B		LLaMA-2-13B	
	TPR@5%FPR	AUC	TPR@5%FPR	AUC
Existing Methods (likelihood-dependent)				
LOSS	7.23	0.55	10.20	0.55
PPL/zlib	9.83	0.56	10.55	0.56
Lowercase	5.58	0.52	5.73	0.56
Min-K%	5.18	0.54	5.20	0.55
Min-K%++	7.93	0.60	7.60	0.61
Proposed Methods (likelihood-independent)				
SaMIA	9.25	0.60	9.83	0.61
SaMIA*zlib	**12.18**	**0.62**	**11.15**	**0.62**

Table 3: TPR@5%FPR and macro averages of AUC scores of the proposed and existing methods, when performing MIA on LLMs of different sizes using WikiMIA. The best score of each column is **bolded**.

	OPT-6.7B	LLaMA-2-7B
Random	0.49	
LOSS	0.57	0.55
PPL/zlib	0.56	0.52
Lowercase	0.54	0.52
Min-K%	0.58	0.52
Min-K%++	0.62	**0.58**
SaMIA	0.58	0.54
SaMIA*zlib	**0.63**	0.55

Table 4: The detection accuracy of the existing and proposed methods on WikiMIA. Micro averages over all different sentence lengths are reported. **Random** represents the baseline where leaked/unleaked of each instance are randomly predicted.

3.6 Comparison with Other Likelihood-Independent Approaches

We compare the performance of SaMIA against DE-COP (Duarte et al., 2024), a method for leakage detection with no reliance on likelihood. DE-COP conducts MIA by providing an LLM with a sentence extracted from a book along with three paraphrased sentences and enhancing it to identify which one is the original using a prompt (e.g., “Which Passage is True Verbatim from The Lord of The Rings?”). As it relies on prompts specific to the book domain requiring a specific book title, it cannot be easily applied to other datasets such as WikiMIA. As a result, we only conducted a comparison on BookTection, as shown in Table 5.

From the results, **SaMIA (*zlib) consistently outperforms DE-COP across all text-length groups**. Note that DE-COP initially reports document-level results, but here we aligned their method with our instance-level evaluation protocol. We used a subset of the original test set for our experiments due to limited computational resources.

	Length		
	64	128	256
DE-COP	0.51	0.54	0.51
SaMIA	0.62	0.63	**0.63**
SaMIA*zlib	**0.64**	**0.65**	0.57

Table 5: AUC score of DE-COP and our proposed method when performing MIA on BookTection. The LLM is LlaMA-2-7B. The best scores of each column is **bolded**.

Length	32	64	128	256	**Avg.**
OPT-6.7B					
SaMIA$_{Rec.}$	**0.55**	**0.63**	**0.66**	**0.80**	**0.66**
SaMIA$_{Prec.}$	0.52	0.51	0.54	0.45	0.51
LLaMA-2-7B					
SaMIA$_{Rec.}$	**0.53**	**0.55**	**0.60**	**0.71**	**0.60**
SaMIA$_{Pre.}$	0.52	0.48	0.56	0.58	0.54

Table 6: AUC scores of SaMIA on WikiMIA when surface-level similarity is evaluated by n-gram recall and precision. SaMIA$_{Rec.}$ is the default SaMIA based on recall and SaMIA$_{Prec.}$ is the variant based on precision.

Due to the abovementioned differences in experimental settings, the results reported here can not be directly compared to those in Duarte et al. (2024).

4 Analysis

This section investigates how each design will affect the performance of SaMIA(*zlib).

4.1 ROUGE-1 v.s. ROUGE-2

We have reported the performance of SaMIA based on ROUGE-1, i.e., unigram overlaps between generated texts and the reference text, in Table 1 and 2. Here, we investigate how varying the size of units used to measure surface-level similarity affects our leakage detection. Specifically, we evaluate the performance of SaMIA based on ROUGE-2 by setting n to 2 in Equation 5.

ROUGE-2 differs from ROUGE-1 in modeling bi-gram overlaps (Lin, 2004). Intuitively, it is easier for an LLM to output the same word in the reference text than two continuous ones. Therefore, we presume that a bi-gram match between the generated texts and the reference text is less likely to happen than a unigram match, making ROUGE-2 a less sensitive metric for detection.

The AUC scores of SaMIA based on ROUGE-1 and ROUGE-2 are shown in Figure 2. On most text length groups, the AUC score of SaMIA based on ROUGE-1 surpasses that of ROUGE-2. For both LLMs, the performance gap between ROUGE-1 and ROUGE-2 becomes evident for the text group with a length of 256. We thus confirmed that measuring surface-level similarity with unigram overlap is a better choice than bi-gram for SaMIA.

4.2 Recall-Based Similarity v.s. Precision-Based Similarity

To measure the surface-level similarity between a generated candidate text x_{cand} and the reference text x_{ref}, SaMIA utilizes a recall-based metric, computing the ratio of n-grams correctly recalled from x_{cand}. Here, we conduct experiments to test the effectiveness of the recall-based metric against a precision-based one. For each candidate text x_{cand}, the precision-based metric evaluates how many n-grams within x_{cand} overlaps with x_{ref} among all generated n-grams. Specifically, we replace the denominator in Equation 1 with the total number of n-grams in x_{cand}:

$$\text{Precison-N}(x_{\text{cand}}, x_{\text{ref}}) = \frac{\sum_{\text{gram}_n \in x_{\text{ref}}} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{\text{gram}_n \in x_{\text{cand}}} \text{Count}(\text{gram}_n)}. \tag{9}$$

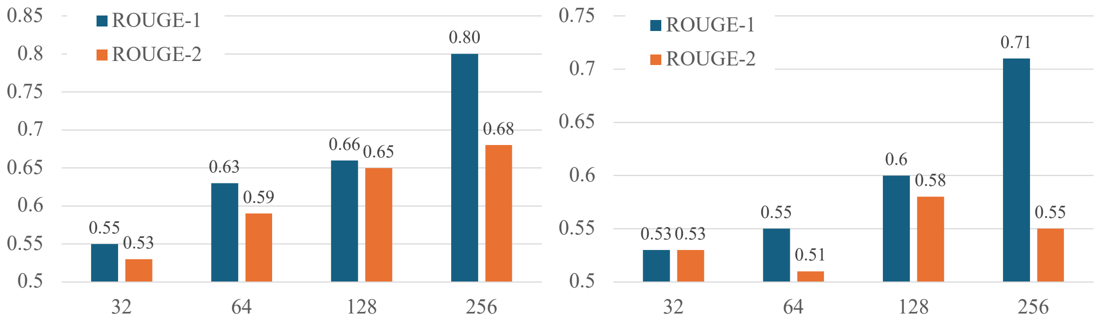
The leakage detection is thus performed by thresholding the averaged Precision-N score among all candidate texts and the reference text.

Table 6 demonstrates the experiment results. We observe that for both LLMs, leakage detection based on precision yields nearly random performance (AUC≈0.50). The observation suggests that it is more beneficial to focus on the quantity of information recalled from the reference text than how precisely the reference text is reproduced. One possible explanation for this can be the difficulty of reproducing the text exactly without introducing any redundant information.

4.3 Number of Samples

Here, we investigate the effect of sampling size, i.e., the number of generated candidate texts, on SaMIA. Intuitively, generating more samples provides more information about the target LLM, thus helping better simulate the "real" distribution. The pseudo-likelihood computed from a larger number of samples should, therefore, be closer to the true likelihood yielded by LLMs. We thus hypothesize that the performance of SaMIA is positively related to the sample size.

Table 7 shows how the performance of SaMIA varies with the sampling size. To analyze the trend more effectively, we report the AUC score for each



(a) OPT-6.7B.

(b) LLaMA-2-7B.

Figure 2: AUC scores of SaMIA on WikiMIA when using ROUGE-1 and ROUGE-2.

Samples	1	2	3	4	5	6	7	8	9	10
OPT-6.7B	0.61	0.64	0.65	0.65	0.65	0.66	0.66	0.66	0.66	0.66
LLaMA-2-7B	0.59	0.59	0.59	0.58	0.60	0.60	0.60	0.60	0.60	0.60

Table 7: AUC scores of SaMIA with different sampling sizes on WikiMIA. Values reported are macro-averages among all target length groups.

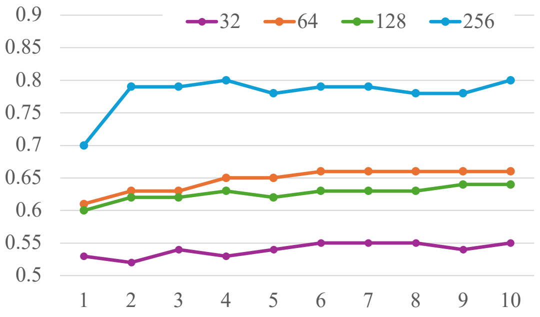


Figure 3: AUC scores of SaMIA on WikiMIA with different sampling sizes. The target LLM is OPT-6.7B.

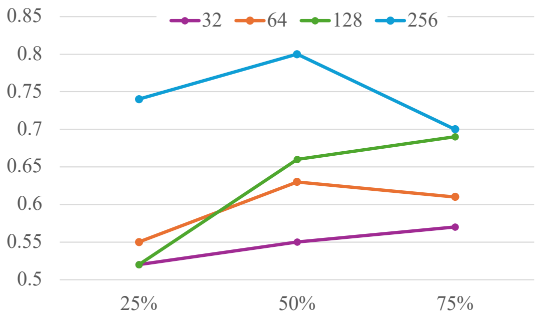


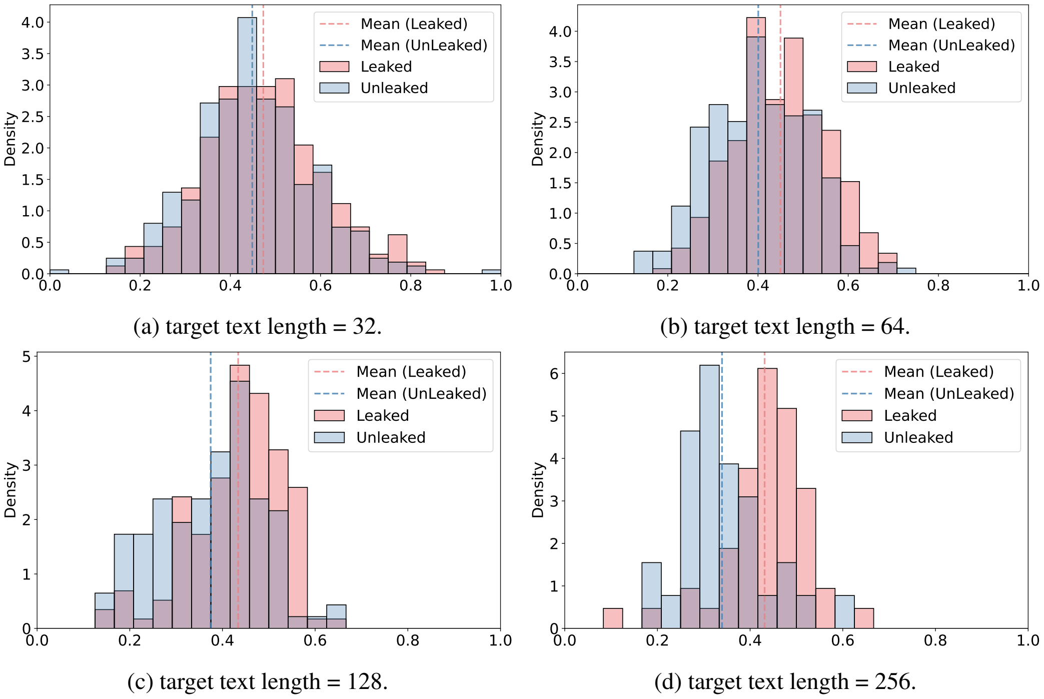
Figure 4: AUC scores of SaMIA on WikiMIA with different prefix ratios. The target LLM is OPT-6.7B.

text length group alongside the changes in sampling size in Figure 3. On OPT-6.7B, while the performance of SaMIA based on a single sample is limited, adding one or two more samples improves the performance for all text-length groups. However, the improvement fades out when the sampling size exceeds 5. The observation suggests that our hypothesis is partially correct, but maintaining many samples is unnecessary. A similar phenomenon can be observed on LLaMA-2-7B, where the performance of SaMIA stabilizes with multiple samples. Therefore, we conclude that a hyper-parameter search should be performed to discover a balanced sampling size that optimizes both detection accuracy (stability) and inference cost.

4.4 Length of Prefix

As introduced in Section 2.2, for each target text x, we divide it into two halves, where the first half serves as the prefix (i.e., prompt) to generate the second half. Here, we investigate how SaMIA behaves if provided with a shorter or longer prefix. A shorter prefix limits the information accessible for the LLM, making it more challenging to recall the preceding context accurately. Consequently, this increases the difficulty of leakage detection.

Specifically, we conduct experiments using 25%, 50%, and 75% of the target text as the prefix, with results shown in Figure 4. In general, compared with the setting where half of the whole text is provided as the prefix, limiting the prefix ratio to 25% causes performance drops. However, increasing the prefix ratio from 50% to 75% does not consistently improve performance; instead, we observe a notable performance decline in the text group with 256 words. The result shows that an optimal prefix length falls within the middle range, consistent with the trend described in Carlini et al. (2022); Huang et al. (2022); Shi et al. (2023); Yang et al. (2024),



(a) target text length = 32.

(b) target text length = 64.

(c) target text length = 128.

(d) target text length = 256.

Figure 5: ROUGE-1 scores of texts generated from OPT-6.7B, using the original texts in WikiMIA as references. Red bars show the distribution of leaked texts and blue bars show that of unleaked ones.

where the prefix-generated texts from LLMs are used to distinguish between text written by the LLMs and humans. The observation indicates that, while prefixes provide important hints for leakage detection, a longer prefix does not guarantee better performance. For text included in the training data, there may exist a soft threshold at which LLMs can effectively recall the context, rendering additional information unnecessary. We left the exploration of such a soft threshold to future work.

4.5 Length of Target Text

Table 2 has demonstrated that as the target text length increases, the performance of SaMIA also improves. Here, we conduct experiments to help understand the reason. Given that SaMIA detects leakage based on ROUGE-N (N=1), we investigate how ROUGE-1 scores of leaked and unleaked texts differ for each text length group.

The distribution of ROUGE-1 scores of leaked and unleaked texts for different lengths is shown in Figure 5. With the text length ranging from 32 to 256, the distance between the distributions of leaked and unleaked texts increases. As the distributions of leaked and unleaked texts move far from each other, they become more separable, resulting in more precise leakage detection via thresholding. As described in Section 2.2, the first half of each target text serves as the prompt for generating the second half. Therefore, one possible explanation can be that longer prompts reduce ambiguity, aiding LLMs in better recalling the memory and generating subsequent contents.

5 Conclusion

We propose SaMIA, a pseudo-likelihood-based approach for leakage detection. We have proven that the token distribution of outputs sampled from the model approximates the likelihood. In experiments, SaMIA demonstrated performance comparable to existing likelihood-dependent methods, despite having limited access to information.

Limitations

Typically, LLMs with non-public training data like ChatGPT and Gemini would be the target of the MIA. However, since it is not possible to prepare both trained and untrained data, in this study we targeted LLMs with known training data. In our experiment, we only evaluated a limited number of datasets (Shi et al., 2023; Duarte et al., 2024), and investigating the robustness of SaMIA across different tasks is a future challenge.

Our primary objective is to demonstrate that our likelihood-free approach can achieve performance comparable to existing likelihood-based methods.

In line with this goal, we have chosen baselines that focus on likelihood-based approaches to test our hypothesis directly. Furthermore, our method does not require access to additional information or extra training, which is why we do not compare it with MIA techniques (Zhang et al., 2024c; Liu et al., 2024; Wen et al., 2024) that rely on such requirements. Due to these fundamental differences, including those approaches in our analysis would be less meaningful for validating our core claims.

Ethical Considerations

LLMs are known to have issues regarding fairness, toxicity, and social bias (Oba et al., 2023; Anantaprayoon et al., 2023; Kaneko et al., 2024b,a,c). In this paper, the experiments were conducted using existing data and existing models, so there are no new ethical concerns from a resource perspective. On the other hand, when sampling the output of the LLM to compute the SPL, there is a possibility of generating text that is related to those concerns. However, since our method ultimately outputs a score rather than text, this does not pose a problem.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra-Aimée Cojocaru, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *ArXiv*, abs/2311.16867.

Panatchakorn Anantaprayoon, Masahiro Kaneko, and Naoaki Okazaki. 2023. Evaluating gender bias of pre-trained language models in natural language inference by considering all labels. *arXiv preprint arXiv:2309.09697*.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. *ArXiv*, abs/2001.08435.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 2633–2650. USENIX Association.

André V. Duarte, Xuandong Zhao, Arlindo L. Oliveira, and Lei Li. 2024. DE-COP: Detecting Copyrighted Content in Language Models Training Data. *Preprint*, arXiv:2402.09910.

Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? approximate unlearning in llms. *ArXiv*, abs/2310.02238.

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. 2022. Are large pre-trained language models leaking your personal information? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2038–2047, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Daphne Ippolito, Florian Tramer, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher Choquette Choo, and Nicholas Carlini. 2023. Preventing generation of verbatim memorization in language models gives a false sense of privacy. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 28–53, Prague, Czechia. Association for Computational Linguistics.

Shotaro Ishihara. 2023. Training data extraction from pre-trained language models: A survey. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 260–275, Toronto, Canada. Association for Computational Linguistics.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR.

Masahiro Kaneko and Timothy Baldwin. 2024. A little leak will sink a great ship: Survey of transparency for large language models from start to finish. *ArXiv*, abs/2403.16139.

Masahiro Kaneko, Danushka Bollegala, and Timothy Baldwin. 2024a. Eagle: Ethical dataset given from real interactions. *arXiv preprint arXiv:2402.14258*.

Masahiro Kaneko, Danushka Bollegala, and Timothy Baldwin. 2024b. The gaps between pre-train and downstream settings in bias evaluation and debiasing. *arXiv preprint arXiv:2401.08511*.

Masahiro Kaneko, Danushka Bollegala, Naoaki Okazaki, and Timothy Baldwin. 2024c. Evaluating gender bias in large language models via chain-of-thought prompting. *arXiv preprint arXiv:2401.15585*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zhenhua Liu, Tong Zhu, Chuanyuan Tan, Bing Liu, Haonan Lu, and Wenliang Chen. 2024. Probing language models for pre-training data detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1576–1587, Bangkok, Thailand. Association for Computational Linguistics.

Mengsay Loem, Sho Takase, Masahiro Kaneko, and Naoaki Okazaki. 2022. Are neighbors enough? multi-head neural n-gram can be alternative to self-attention. *arXiv preprint arXiv:2207.13354*.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11330–11343, Toronto, Canada. Association for Computational Linguistics.

R. Thomas McCoy, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. 2023. How much do language models copy from their training data? evaluating linguistic novelty in text generation using RAVEN. *Transactions of the Association for Computational Linguistics*, 11:652–670.

Daisuke Oba, Masahiro Kaneko, and Danushka Bollegala. 2023. In-contextual bias suppression for large language models. *arXiv preprint arXiv:2309.07251*.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*.

R. Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2016. Membership inference attacks against machine learning models. *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18.

Simeng Sun and Mohit Iyyer. 2021. Revisiting simple neural probabilistic language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5181–5188, Online. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Rui Wen, Zheng Li, Michael Backes, and Yang Zhang. 2024. Membership inference attacks against in-context learning. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 3481–3495.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. 2024. DNA-GPT: Divergent n-gram analysis for training-free detection of GPT-generated text. In *The Twelfth International Conference on Learning Representations*.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. 2022. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3093–3106.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2017. Privacy risk in machine learning: Analyzing the connection to overfitting. *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *31st*

IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018, pages 268–282. IEEE Computer Society.

Weichen Yu, Tianyu Pang, Qian Liu, Chao Du, Bingyi Kang, Yan Huang, Min Lin, and Shuicheng Yan. 2023. Bag of tricks for training data extraction from language models. *ArXiv*, abs/2302.04460.

Jie Zhang, Debeshee Das, Gautam Kamath, and Florian Tramèr. 2024a. Membership inference attacks cannot prove that a model was trained on your data. *arXiv preprint arXiv:2409.19798*.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. 2024b. Min-k%++: Improved baseline for detecting pre-training data from large language models. *Preprint*, arXiv:2404.02936.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *ArXiv*, abs/1904.09675.

Weichao Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024c. Pretraining data detection for large language models: A divergence-based calibration method. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5263–5274, Miami, Florida, USA. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Z. Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jianyun Nie, and Ji rong Wen. 2023. A survey of large language models. *ArXiv*, abs/2303.18223.

Xuandong Zhao, Lei Li, and Yu-Xiang Wang. 2022. Provably confidential language modelling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 943–955, Seattle, United States. Association for Computational Linguistics.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. 2023. Don't make your llm an evaluation benchmark cheater. *ArXiv*, abs/2311.01964.

Length	Leaked	Unleaked	Total
32	387	389	776
64	284	258	542
128	139	111	250
256	51	31	82
Total	861	789	1,650

Table 8: Number of leaked/unleaked instances in each text length group of WikiMIA.

A WikiMIA: Statistics

Experiments in this work are mainly based on WikiMIA (Shi et al., 2023). The dataset is a benchmark for evaluating membership inference attack methods and comprises Wikipedia event pages from 2017 to 2023. The texts are divided into 4 groups with different lengths, namely 32, 64, 128, and 256. We regard texts with a timestamp earlier than 2023 as data included in the training data for LLMs (i.e., leaked data) and those with a timestamp later than 2023 as data excluded from the training data (i.e., unleaked data). Table 8 details the statistics of WikiMIA.

B TPR at other FPRs

Section 3 reports the true positive rates of different leakage detection methods at a false positive rate as low as 5% (TPR@5%FPR). Here, we detail the true positive rates at a higher or lower false positive rate, i.e., 10% and 1%. As shown in Table 9, SaMIA*zlib robustly outperforms the existing likelihood-dependent baselines, with the only exception being LLaMA-2-7B at a 1% false positive rate.

C Effect of N-gram

We adopted ROUGE-N for leakage detection, where we finally chose $N = 1$ and conducted MIA based on unigrams. Here, we clarify our design choice of using unigrams by reporting the results of our preliminary experiments. As in Table 10, increasing the gram size beyond 2 does not result in further performance improvements. We, therefore, focus on unigrams. This means that token-level information is the most beneficial for approximating the likelihood.

D Incoporating Zlib

This work incorporated zlib into ROUGE via multiplication. However, combining these two measure-

	OPT-6.7B		LLaMA-2-7B	
	10%	1%	10%	1%
Existing Methods (likelihood-dependent)				
LOSS	16.8	0.68	13.8	**3.38**
PPL/zlib	17.5	16.1	9.83	2.00
Lowercase	16.0	3.68	15.5	0.00
Min-K%	25.8	2.88	10.7	0.73
Min-K%++	26.6	2.25	15.1	2.60
Proposed Methods (likelihood-independent)				
SaMIA	31.8	1.40	16.4	0.38
SaMIA*zlib	**37.3**	**4.55**	**20.0**	1.65

Table 9: Macro-averages of TPR at low FPR (10% and 1%) of the proposed and existing methods, when performing MIA on various LLMs using WikiMIA. The best score of each column is **bolded**.

	n=1	n=2	n=3	n=4	n=5
WikiMIA					
SaMIA	0.59	0.54	0.51	0.50	0.50
SaMIA*zlib	0.62	0.57	0.53	0.50	0.50
BookTection					
SaMIA	0.63	0.60	0.57	0.55	0.53
SaMIA*zlib	0.62	0.60	0.57	0.55	0.53

Table 10: Macro-average of AUC of our proposed methods when varying the size of n-grams.

ments with other methods is also possible. This section reports the detection performance of ROUGE-N + zlib in AUC as in Table 11 For comparison purposes, we include the performance of detecting only with ROUGE and with ROUGE * zlib. As shown in the table, compared to adding zlib, multiplying zlib yields more consistent performance improvement. The observation validates our design choice of multiplying ROUGE with zlib as the detection metric.

E Likelihood Estimation Using ROUGE-N

We prove that the ROUGE-N score between a reference text x_{ref} and candidates x_{cand} generated from the LLM approximates the likelihood of an n-gram LM for x_{ref}. We start with the definition of ROUGE-N:

$$\text{ROUGE-N}(x_{\text{cand}}, x_{\text{ref}}) = \frac{\sum_{\text{gram}_n \in x_{\text{ref}}} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{\text{gram}_n \in x_{\text{ref}}} \text{Count}(\text{gram}_n)} \tag{10}$$

	OPT-6.7B	LLaMA-2-7B
ROUGE	0.66	0.60
ROUGE*zlib	**0.71**	**0.62**
ROUGE+zlib	0.64	0.60

Table 11: Macro-average of AUC of the existing and proposed methods when performing MIA on various LLMs using WikiMIA. The best scores of each column is **bolded**.

We can express $\text{Count}_{\text{match}}(\text{gram}_n)$ using indicator functions$I_i(\cdot)$:

$$\text{Count}_{\text{match}}(\text{gram}_n) = \sum_{i=n}^{|x_{\text{cand}}|} I_i(\text{gram}_n) \tag{11}$$

where I_i is an indicator function defined as:

$$I_i(\text{gram}_n) = \begin{cases} 1 & \text{if } (x_{\text{cand,i-n+1}}, \ldots, x_{\text{cand,i}}) = \text{gram}_n \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Computing the expectation of ROUGE-N:

$$\mathbb{E}[\text{ROUGE-N}(x_{\text{cand}}, x_{\text{ref}})] = \frac{\sum_{\text{gram}_n \in x_{\text{ref}}} \mathbb{E}[\sum_{i=n}^{|x_{\text{cand}}|} I_i(\text{gram}_n)]}{\sum_{\text{gram}_n \in x_{\text{ref}}} \text{Count}(\text{gram}_n)} \tag{13}$$

Note that since the denominator $\sum_{\text{gram}_n \in x_{\text{ref}}} \text{Count}(\text{gram}_n)$ depends only on the reference text and is therefore constant, we can move it outside the expectation. The expectation of the indicator function $I_i(\text{gram}_n)$ is:

$$\begin{aligned} \mathbb{E}[I_i(\text{gram}_n)] &= P(I_i(\text{gram}_n) = 1) \\ &= P_\theta(\text{gram}_n | x_{\text{cand,i-n+1}}, \ldots, x_{\text{cand,i-1}}) \end{aligned} \tag{14}$$

Substituting this expectation back into the ROUGE-N expectation and applying the linearity of expectation:

$$\begin{aligned} &\mathbb{E}[\text{ROUGE-N}(x_{\text{cand}}, x_{\text{ref}})] \\ &= \frac{1}{Z} \sum_{\text{gram}_n \in x_{\text{ref}}} \sum_{i=n}^{|x_{\text{cand}}|} P_\theta(\text{gram}_n | x_{\text{cand,i-n+1}}, \ldots, x_{\text{cand,i-1}}) \end{aligned} \tag{15}$$

where $Z = \sum_{\text{gram}n \in x\text{ref}} \text{Count}(\text{gram}_n)$ is a normalization term. When the model f_θ is well-trained, the conditional probability distribution of generated candidates approaches that of the reference:

$$\begin{aligned}&\mathbb{E}[\text{ROUGE-N}(x_{\text{cand}}, x_{\text{ref}})]\\ &\approx \frac{1}{Z}\sum_{i=n}^{|x_{\text{ref}}|} P_{f_\theta}(x_{\text{ref,i}}|x_{\text{ref,i-n+1}}, ..., x_{\text{ref,i-1}})\end{aligned} \tag{16}$$

This right-hand side relates to the normalized log-likelihood:

$$\begin{aligned}\log P_\theta(x_{\text{ref}}) &= \sum_{i=n}^{|x_{\text{ref}}|} \log P_\theta(x_{\text{ref,i}}|x_{\text{ref,i-n+1}}, ..., x_{\text{ref,i-1}})\\ &\approx c \cdot Z \cdot \mathbb{E}[\text{ROUGE-N}(x_{\text{cand}}, x_{\text{ref}})]\end{aligned} \tag{17}$$

where c is a positive constant. Therefore, we have proven that the expectation of the ROUGE-N score approximates the likelihood of the n-gram LM. Furthermore, it has been experimentally shown that n-gram LMs can approximate neural LMs (Sun and Iyyer, 2021; Loem et al., 2022), and using ROUGE-L is beneficial for approximating the likelihood of an LM.