Tag&Tab: Pretraining Data Detection in Large Language Models Using Keyword-Based Membership Inference Attack

Anonymous ACL submission

Abstract

Large language models (LLMs) have become essential tools for digital task assistance. Their training relies heavily on the collection of vast amounts of data, which may include copyright-protected or sensitive information. Recent studies on detecting pretraining data in LLMs have primarily focused on sentence- or paragraph-level membership inference attacks (MIAs), usually involving probability analysis of the target model's predicted tokens. However, these methods often exhibit poor accuracy, failing to account for the semantic importance of textual content and word significance. To address these shortcomings, we propose Tag&Tab, a novel approach for detecting data used in LLM pretraining. Our method leverages advanced natural language processing (NLP) techniques to tag keywords in the input text—a process we term *Tagging*. Then, the LLM is used to obtain probabilities for these keywords and calculate their average log-likelihood to determine input text membership, a process we refer to as *Tabbing*. Our experiments on four benchmark datasets (BookMIA, MIMIR, PatentMIA, and the Pile) and several open-source LLMs of varying sizes demonstrate an average increase in AUC scores ranging from 5.3% to 17.6% over state-of-the-art methods. Tag&Tab not only sets a new standard for data leakage detection in LLMs, but its outstanding performance is a testament to the importance of words in MIAs on LLMs.

1 Introduction

The rapid advancement of generative artificial intelligence (GenAI) in recent years has significantly shifted the tech industry's focus toward the development of powerful tools such as large language models (LLMs).

LLMs are now widely used for tasks such as conversational AI, content generation, and scientific research (Hoang et al., 2019; Nakano et al., 2021; OpenAI, 2022; Touvron et al., 2023). Their adoption reflects a broader shift in AI towards large-scale language understanding.

The widespread use of LLMs has intensified competition to improve model performance, which relies on collecting vast amounts of data (Wang et al., 2023).

To achieve these improvements, LLMs are primarily trained on open-source datasets obtained from various sources using methods such as synthetic data generation and web scraping (Nikolenko, 2021; Khder, 2021). The type of data collected may include books, code, academic papers, and medical records (Axon, 2024; Gao et al., 2020; Achiam et al., 2023; Touvron et al., 2023). The methods employed for data collection raise significant privacy and ethical concerns (Neel and Chang, 2023; Yao et al., 2024), primarily regarding the inclusion of personally identifiable information (PII) (Lukas et al., 2023) and copyright-protected contents. (Rahman and Santacana, 2023; Wu et al., 2024; Axon, 2024).

High-profile lawsuits, such as The New York Times vs. OpenAI (Times, 2023), highlight the need for tools that can detect unauthorized use of data in LLM training (Maini et al., 2024).

Membership inference attacks (MIAs) aim to identify whether a given text was part of a model's training data by exploiting behavioral differences in how LLMs process seen versus unseen data (e.g., higher prediction confidence or lower loss) (Hu et al., 2022; Carlini et al., 2022a). However, existing MIAs face several key limitations. First, most methods rely solely on token-level probabilities, neglecting the semantic importance of words within the broader context (Yeom et al., 2018; Carlini et al., 2021). Second, their performance varies widely across different models and datasets, often lacking consistent generalization (Duan et al., 2024; Maini et al., 2024). Lastly, MIAs are often evaluated on data that is not independently and identically distributed (IID), which can lead to the

detection of distribution shifts rather than genuine membership inference, thereby undermining the attacks' reliability (Zhou et al., 2023).

To address the limitations of existing methods, we introduce Tag&Tab, a novel approach based on common natural language processing (NLP) methods that is designed to efficiently and effectively detect LLMs' pretraining data. Specifically, our method aims to determine whether an LLM was trained on a given text sample, given black-box access to the target LLM (i.e., can only query the model).

Building on the work of Lukas et al. (Lukas et al., 2023), who highlighted the role of named entities in PII leakage detection, Tag&Tab prioritizes informative keywords using entropy-based selection.

Tag&Tab is designed to address the three key limitations of prior MIA methods. First, rather than relying solely on token-level perplexity, our method introduces semantic awareness by prioritizing meaningful content through keyword selection. Second, our results demonstrate strong generalization across models and datasets, addressing the issue of model inconsistency. Finally, while no MIA is entirely immune to distribution shifts, our focus on rare and informative keywords, rather than shallow statistical artifacts, provides better resilience to distributional variations.

Our method consists of the following steps:

1. Preprocessing - Constructing a word entropy map and filtering certain sentences to ensure optimal keyword selection.

2. Tagging - Identifying the high-entropy words in the text and selecting the K words with the highest entropy value, referred to as *keywords*.

3. Tabbing - Passing the entire text to the target LLM and calculating the average log-likelihood of these K keywords.

4. Inference - Comparing the average log-likelihood to a threshold to determine the text's membership (i.e., whether it was in the inspected model training set).

Our method is based on the intuition that a higher log-likelihood for challenging-to-predict high-entropy keywords suggests the model encountered the text during training (Carlini et al., 2022b) Based on this intuition, we hypothesize that these rare keywords are more likely to be memorized by the model and thus serve as effective indicators of the text's membership in the pretraining dataset (Thakkar et al., 2021; Carlini et al., 2019). By selecting a small number of high-entropy keywords, our method captures the most informative text elements while minimizing noise from other word probabilities.

We evaluated our method on ten LLMs of varying sizes and across four datasets containing 9 types of textual data. Our results show that Tag&Tab outperforms state-of-the-art (SOTA) MIAs, achieving an average increase in AUC scores ranging from 5.3% to 17.6% compared to the best-performing SOTA method across multiple textual data types. The contributions of our paper are as follows:

- We propose Tag&Tab, a novel approach for the detection of LLMs' pretraining data that focuses on the contextual and semantic relevance of the words in a text, and opens the door to additional research on MIAs against LLMs.

- To our knowledge, this is the first robust reference-free MIA method to achieve high and consistent performance across multiple textual data types and LLMs, outperforming SOTA methods.

- Our approach is both resource- and time-efficient. Unlike reference-based attacks that require training a separate model or reference-free methods that depend on auxiliary models (e.g., the Neighbor attack (Mattern et al., 2023)), Tag&Tab operates without any additional model training or inference. This minimizes computational overhead and simplifies deployment in real-world scenarios. .

2 Related Work

Membership inference (MI) (Shokri et al., 2017) is a classification task that determines whether a data sample x was part of a model's training D_{train} of a model f. An attacker receives a sample x and a model f, and applies an attack model A to classify x as a member $x \in D_{\text{train}}$ if $A(f(x)) = 1$; otherwise, x is classified as a non-member $x \notin D_{\text{train}}$.

Large language model membership inference is a subdomain of membership inference that has gained increasing research attention. Within this subdomain, detecting pretraining data has been

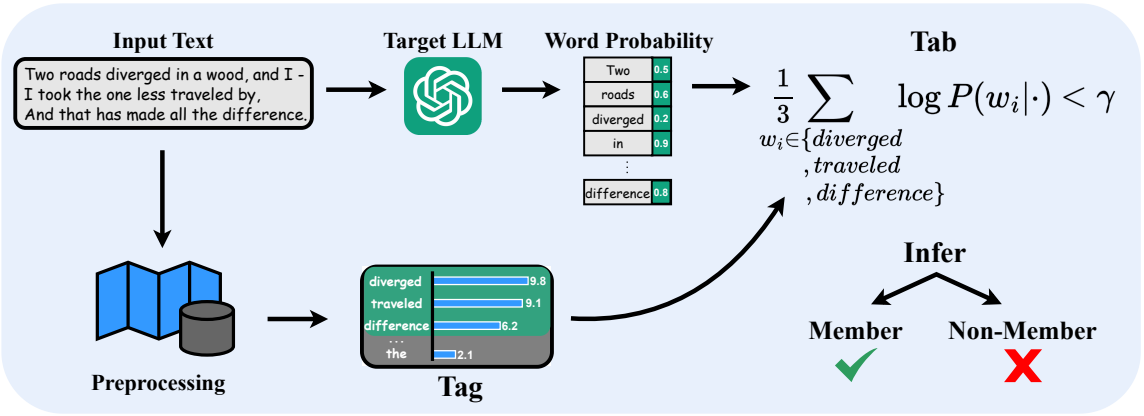

Figure 1: Illustration of the Tag&Tab method - The process starts by inputting a text (in this example, the input is the conclusion of the well-known poem “The Road Not Taken" (Frost, 1916)) in the target LLM to obtain its word probability distribution (word probability). In the tag step, the keywords are selected based on the words’ entropy value (created in the preprocessing phase). In the tab step, the log-likelihood of the selected keywords is calculated. Finally, in the infer step, the average log-likelihood of the chosen keywords is compared against a threshold γ to determine if the text was part of the target LLM’s pretraining data.

the focus of numerous studies exploring different methodologies for determining whether specific texts were included in an LLM’s training dataset. Existing MIAs for LLMs fall into two categories: reference-based and reference-free.

Reference-based (Shokri et al., 2017) methods compare a target model’s outputs to those of reference models, which are typically trained on the same data distribution. One such method is *LiRA* (Carlini et al., 2022a), which estimates the likelihood ratio of a target example’s loss under models trained with and without the example, using Gaussian distributions to simplify the computation.

In contrast, **reference-free methods** aim to determine membership by applying different probability-based calculations on token predictions. One such method, the *LOSS Attack* (Yeom et al., 2018), uses model loss values, which in language models correspond to text perplexity. Perplexity measures how well a probability model predicts a sample and is calculated as the exponentiation of the negative average log-likelihood per token:

$$\text{Perplexity}(P) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(t_i \mid t_1, \ldots, t_{i-1})\right)$$

where N is the number of tokens, and $P(t_i \mid t_1, \ldots, t_{i-1})$ is the conditional probability of the i-th token given its preceding tokens. The attack assumes that lower perplexity indicates a text is more familiar to the model, suggesting it was part of the training set. The *Zlib attack*, as presented by Carlini et al. (Carlini et al., 2021), infers membership by calculating the ratio of a text’s log-likelihood to its Zlib compression length. Newer attacks, such as the *Neighbor attack* (Mattern et al., 2023), modify selected words in a given text using a different language model to generate ’neighbor’ sentences, then compare the original text’s perplexity to that of its neighbors. Although the Neighbor attack showed some success, its computational cost is very high compared to other known methods. More computationally efficient attacks that outperform the Neighbor attack include *Min-K%* (Shi et al., 2023), and *Min-K%++* (Zhang et al., 2024a), which focus on the least confident model predictions. Min-K% calculates the average of the lowest k% probabilities from the model’s output, and Min-K%++ extends this by normalizing token log probabilities using the mean and variance. Lastly, two recently published attacks are *RECALL* (Xie et al., 2024), which measures the relative change in log-likelihood when conditioning the target text on non-member prefixes, and *DC-PDD* (Zhang et al., 2024b), which calibrates token probabilities using divergence from a reference corpus, effectively mitigating the impact of high-frequency tokens. While each of these attacks has demonstrated success on some datasets and models, their performance remains inconsistent across different studies. However, recent research performed by Maini et al. (Maini et al., 2024) showed that aggregating the results of multiple MIAs improves the accuracy of dataset membership inference. While promising,

these findings suggest that the success of this aggregated approach in real-world scenarios depends on improved MIAs, meaning attacks achieving AUC scores above 0.5.

Common models for evaluating MIAs on LLMs include open-source models with known pretraining data. One such model is LLaMA 1 (Touvron et al., 2023), created by Meta, which was trained on a mixture of publicly available datasets, including certain subsets of the Pile such as C4, GitHub, and many more, according to Meta's original paper and the MIMIR dataset (Duan et al., 2024).

The open-source Pythia model suite (Biderman et al., 2023), which includes eight LLMs ranging from 70M to 12B parameters. These models were trained on data including the Pile dataset, with all models processing the public data in the same order during training.

3 Method

We introduce *Tag&Tab*, a novel resource- and time-efficient method for identifying data used to pretrain LLMs. *Tag&Tab* applies common NLP techniques to tag keywords in the pretraining data and predict their occurrence using the target LLM.

Our method strategically selects words from the input that should be challenging for the LLM to predict. Successful prediction may indicate that the model previously learned the input's content during pretraining. In Tag&Tab, words are selected according to their entropy values. A high entropy value indicates that a word is less common in the input text compared to other words. Since LLMs tend to memorize rare or unique pretraining data, high-entropy words are more likely to be memorized (Carlini et al., 2022b), particularly in the context of their preceding words. Thus, the LLM is likely to assign higher probabilities to these words if it was trained on them, compared to high-entropy words in unfamiliar contexts.

Tag&Tab selects K words with the highest entropy value in a sentence, which are referred to as *keywords*. By selecting a small number of informative keywords, we aim to capture the semantic importance of the pretraining content. This approach hinges on a hypothesis that only K selected keywords are needed to accurately predict the membership of the entire input text in the pretraining data, minimizing noise from other words in the input text (Shi et al., 2023).

Tag&Tab operates under black-box constraints, meaning we can observe token probabilities from a given input but lack access to the model's weights, which is standard practice in MIAs (Truex et al., 2019; Hu et al., 2022; Mattern et al., 2023; Zhang et al., 2024a).

The *Tag&Tab* method consists of four stages, which are illustrated in Figure 1:

1. **Preprocessing** - First, the **word entropy map** is constructed using the Python package *wordfreq* (Speer, 2022), which provides frequency estimates for words in a specified language. The entropy for each word is calculated using the formula:

$$E(w_i) = p(w_i) \cdot \log_2 p(w_i)$$

   In the preprocessing stage, the text is also split into individual sentences, using segmentation tools (e.g., the NLTK package (Bird et al., 2009)). To avoid selecting less informative keywords due to insufficient sentence length, sentences with fewer than a specified number of words are filtered out.

2. **Tagging** - From each sentence S in the text file T, K keywords are selected, targeting **named entities** (e.g., people, organizations, locations) and **words with high entropy values**. This selection is based on the word entropy map for high-entropy words, and named entities are identified during this process. Named entities are identified using tools such as spaCy (Honnibal and Montani, 2017). The final set of keywords consists of the union of the named entities and the high-entropy words.

3. **Tabbing** - This stage mimics the auto-completion feature found in interfaces like a command line, where it predicts and fills in the rest of the command based on the context of the preceding input. Using the target model M, we compute the log-likelihood of the entire text, then focus on the previously identified keywords. For each sentence $S \in T$ consisting of n words $w_1, w_2, \ldots, w_n$, where each word w_i is decomposed into tokens, denoted as $w_i = t_{i_1}, t_{i_2}, \ldots, t_{i_m}$,token t_{i_j}, given its preceding tokens, is calculated as $\log p_M(t_{i_j}|t_{i_1}, \ldots, t_{i_{j-1}})$. We define the log-likelihood of a word w_i using the log-likelihood of its first token t_{i_1}

given its preceding tokens, expressed as $\log p_M(t_{i_1}|t_{1_1}, t_{1_2}, \ldots, t_{i-1_j})$. As a result, we obtain:

$$\log p_M(w_i \mid \cdot) = \log p_M(t_{i_1} \mid \cdot)$$

The method selects the K keywords from S and computes the average log-likelihood of the keywords:

$$\text{Keywords' Prob}(S) = \frac{1}{K} \sum_{w_i \in \text{Keywords}(S)} \log p_M(w_i|\cdot)$$

4. **Inferring** - In this stage, the method calculates the average probability of the keywords across all sentences in text T and compares it to a predetermined threshold γ, to determine membership.

4 Evaluation

This section presents a detailed evaluation of Tag&Tab's effectiveness. The experiments were conducted on a single NVIDIA RTX 6000 GPU, running for nearly three days in total across all models and datasets. We used the default parameter settings of widely adopted libraries, including spaCy and NLTK.

4.1 Model Comparison

To compare *Tag&Tab* with other reference-free baseline detection methods, we examined various open-source LLMs, including LLaMA 1 (7B, 13B, 30B) (Touvron et al., 2023), Pythia (160M, 1.4B, 2.8B, 6.9B, 12B) (Biderman et al., 2023), and Qwen-1.5-14B (Cloud, 2024). Additionally, we included GPT-3.5 Turbo[1] (trained on data up to September 2021[2]), given partial knowledge of its training on known books, as discussed in previous studies (Shi et al., 2023; Chang et al., 2023). Our black-box assumption still holds because the OpenAI API exposes token-level log probabilities for this model. LLaMA 1 and Pythia are well-suited for MIA evaluation due to their transparency regarding pretraining datasets, unlike newer models such as LLaMA 2 and 3, which lack such transparency.

[1]https://platform.openai.com/docs/models/gpt-3-5-turbo

[2]https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/models

4.2 Dataset Comparison

The experiments were conducted on the BookMIA[3] (Shi et al., 2023), The Pile[4] (Gao et al., 2020), and MIMIR[5] (Duan et al., 2024) datasets, meeting the requirement that MIA evaluation datasets should be as comprehensive and diverse as possible (Duan et al., 2024), covering diverse text types while maintaining consistent distribution between training and test sets.

The Pile is a collection of diverse textual sources designed to train and evaluate LLMs using open-source data, from which we used 10,000 samples each for training and testing from each domain. BookMIA evaluates MIAs on books known to have been memorized by GPT-3.5 Turbo, along with newly published books from 2023, using 5,000 samples each for the member and non-member sets. Notably, 34 of the 50 'member' books in BookMIA overlap with the Gutenberg dataset, which was also part of the training corpora for models like Pythia and LLaMA 1. MIMIR is a dataset built from The Pile, designed to evaluate memorization in LLMs. It contains data known to have been used in training across all Pythia model sizes, offering a unified benchmark for assessing membership inference. In our evaluation, we utilized around 2,000 samples per domain, combining approximately 1,000 'member' and 1,000 'non-member' samples.

Finally, we ran a dedicated non-Latin evaluation on the Chinese-language PatentMIA (Zhang et al., 2024b) corpus with the Qwen1.5-14B model. The setup and results are discussed in Appendix A.4.

It is important to note that we opted not to evaluate our method on the WikiMIA dataset[6] (Shi et al., 2023), as recent publications (e.g., (Maini et al., 2024)) questioned the reliability of the data, due to temporal shifts in writing styles and an insufficient number of samples.

4.3 Evaluation Approach

To assess our method's performance, we followed a systematic process that begins with the input dataset. Each text file in the dataset is processed, with every sentence truncated to a maximum of 2,048 tokens to ensure a consistent input size. Sentences with fewer than seven words are excluded. We identify and save the top-K keywords for every

[3]www.huggingface.co/datasets/swj0419/BookMIA

[4]www.huggingface.co/datasets/monology/pile-uncopyrighted/viewer/default/validation

[5]www.huggingface.co/datasets/iamgroot42/mimir

[6]www.huggingface.co/datasets/swj0419/WikiMIA

sentence based on their entropy values. We documented the outcomes of selecting 1 to 10 keywords per sentence.

After selecting the top-k keywords, the entire text is processed by the target model, which outputs probability distributions for each token. Then we average the log-likelihoods of all tokens in the keywords, conditioned on their preceding tokens, assessing the model's familiarity with the entire keyword.

Following Carlini et al.'s evaluation process (Carlini et al., 2022a), we set a threshold to assess attack performance, focusing on TPR at low FPR, denoted as T@F.

We also report the area under the ROC curve (AUC score) to provide a clearer measure of detection performance. The AUC score quantifies the overall performance of a classification method by considering TPRs and FPRs at all classification thresholds. Since AUC offers a comprehensive, threshold-independent evaluation metric, we do not need to determine a specific threshold γ for our method.

To simulate a real-world application, Appendix A.3 details how we pick a working threshold γ when only non–member data from the same domain are available. The appendix further shows that applying this book-derived threshold to a different domain (mathematics) degrades performance, so γ must be recalibrated for every model–dataset pair.

4.4 Comparison with Baseline Methods

To benchmark Tag&Tab's performance, we compared it with SOTA reference-free methods for detecting pretraining data. The baseline methods included LOSS Attack, Zlib Attack, Neighbor Attack, Min-K% Prob, Min-K%++ Prob, Max-K%Prob, RECALL, DC-PDD. A detailed description of these attacks can be found in the Related Work section.

5 Results

This section presents two case studies using Tag&Tab, each examining a different aspect of pretraining data detection in LLMs. Each case study is evaluated using AUC and TPR at a low FPR of 5% (T@F=5%).

Throughout this section, we report results for Tag&Tab using $K = 4$ and $K = 10$, as these values yielded the most consistent and high-performing outcomes across models and datasets. This choice is supported by the analysis presented later in Figure 2, which shows that Tag&Tab performs robustly for values of K between 4 and 10, with minimal performance variation—making the method resilient to non-optimal keyword selections.

The reported results are based on a single run, as we observed minimal variation across multiple runs.

5.1 Case Study 1: Detecting Specific Pretraining Data in LLMs

This case study focuses on the precision of detecting specific pretraining data in LLMs. We designed a targeted attack to infer whether copyrighted data was part of the model's pretraining. Unlike Case Study 2, member and non-member data come from different sources. Using the BookMIA dataset, we simulate partial knowledge of a model's pretraining data to infer specific text files suspected of being included in the target model's training set. For validation, we selected non-members from books published after the target model's release, thus ensuring they were not part of its pretraining data.

To determine the optimal number of keywords to select, we evaluate the results by selecting between 1 and 10 keywords from each sentence The results shown in Figure 2 demonstrate that the optimal number of keywords required to ensure effective detection depends on the model architecture. As observed, for different sizes of the LLaMa 1 models, the optimal number of keywords ranged from 2 to 3, while for the Pythia models and GPT-3.5 turbo, the optimal number of tagged keywords was 7. The best results across all models were achieved with $K = 4$, yielding an average AUC score of 0.8.

Table 1 summarizes the results of Tag&Tab and reference-free baseline attacks.

The main insights from these results are as follows:

- Tag&Tab outperforms all other attacks in AUC, with an average improvement of 5.3%–17.6% over baseline methods when $K = 4$.

- Tag&Tab ($K = 4$) outperforms other attacks in T@F=5% on LLaMa-7B and LLaMa-13B models. However, for most other models, the DC-DPP attack performs better, with Tag&Tab consistently ranking second.

Method	LLaMa-7b		LLaMa-13b		LLaMa-30b		Pythia-6.9b		Pythia-12b		GPT-3.5		Average	
	AUC	*T@F5*	*AUC*	*T@F5*	*AUC*	*T@F5*	*AUC*	*T@F5*	*AUC*	*T@F5*	*AUC*	*T@F5*	*AUC*	*T@F5*
Neighbor	0.65	0.27	0.71	0.38	0.90	0.73	0.65	0.26	0.71	0.36	0.96	0.88	0.76	0.48
Loss	0.59	0.25	0.70	0.43	0.89	0.74	0.62	0.24	0.69	0.32	**0.97**	0.90	0.74	0.48
Zlib	0.53	0.22	0.67	0.42	0.89	0.74	0.55	0.19	0.61	0.25	0.96	0.88	0.70	0.45
Min-20.0% Prob	0.61	0.24	0.70	0.42	0.87	0.70	0.65	0.25	0.70	0.34	0.95	0.86	0.75	0.47
MinK++-20.0% Prob	0.60	0.23	0.68	0.38	0.78	0.60	0.59	0.20	0.56	0.20	0.95	0.86	0.69	0.41
Max-20.0% Prob	0.51	0.15	0.66	0.34	0.87	0.69	0.51	0.13	0.59	0.20	0.96	**0.91**	0.68	0.40
ReCaLL	0.58	0.22	0.70	0.42	0.84	0.64	0.66	0.29	0.72	0.37	0.74	0.50	0.70	0.41
DC-PDD	0.61	0.27	0.71	0.47	0.88	**0.77**	0.68	**0.34**	0.74	**0.44**	0.95	0.89	0.76	**0.53**
Ours (Tag&Tab K=4)	**0.69**	**0.28**	**0.78**	**0.48**	**0.91**	0.76	**0.72**	0.30	0.75	0.36	**0.97**	0.90	**0.80**	0.51
Ours (Tag&Tab K=10)	0.67	0.26	0.77	0.46	**0.91**	**0.77**	**0.72**	0.30	**0.76**	0.36	0.96	0.87	**0.80**	0.50

Table 1: Detection of data from the BookMIA dataset used in pretraining seven models using Tag&Tab and six baseline MIAs, evaluated in terms of AUC and T@F=5%. All results are reported as decimal fractions. The last two rows compare the Tag&Tab method when selecting four and ten keywords. The best results are bolded, and second-best are underscored.

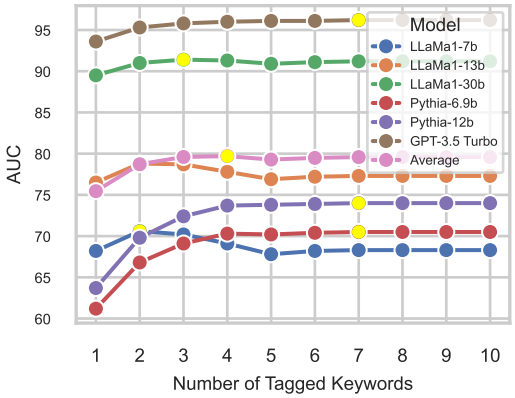

Figure 2: AUC scores as a function of the number of tagged keywords for the examined models on the BookMIA dataset. Yellow dots indicate optimal performance: 2-3 keywords for LLaMa 1, 7 for Pythia and GPT-3.5 turbo, and 4 on average (AVG).

- As the size of the tested LLM increases, the AUC scores of the MIAs also increase due to the model's memorization capacity (Carlini et al., 2022b). This can be seen in the results for Tag&Tab which achieved very high AUC scores: (1) 0.91 on LLaMa-30b compared to 0.69 on LLaMa-7b, (2) 0.75 on Pythia-12b compared to 0.72 Pythia-6.9b, and (3) 0.97 on GPT-3.5 Turbo, the largest model tested.

5.2 Case Study 2: Detecting Various Types of Pretraining Data in LLMs

This case study evaluates Tag&Tab's effectiveness and robustness in detecting different types of pretraining data in LLMs. We evaluate our method on various sizes of the Pythia model and compare its effectiveness against baseline attacks on seven text types in the Pile dataset. Table 2 summarizes the results obtained when targeting five Pythia model sizes ranging from 160M to 12B parameters, tested with two configurations of tagged keywords: 4 and 10.

The main insights from these results are as follows:

- Tag&Tab ($K = 4$) outperforms all baseline methods on average across the evaluated models, establishing itself as the most effective approach overall. Tag&Tab ($K = 10$) ranks as the second-best method, demonstrating strong performance but falling short of the results achieved with $K = 4$.

- Notably, Tag&Tab ($K = 4$) achieves either the best or second-best results in the majority of textual data types that were tested. Even when it does not lead, its performance remains competitive, offering a robust alternative to the leading method.

- While Tag&Tab ($K = 10$) selects more keywords, increasing the number of probabilities considered for text membership inference, its results are consistently lower than those of Tag&Tab when $K = 4$. This supports the hypothesis that selecting a smaller number of keywords allows the method to extract noise-free information from the model.

We also observe that certain formal texts, such as mathematical proofs, may contain fewer named entities or conventional keywords. However, they often feature domain-specific terminology or symbolic expressions that carry strong membership signals. This is evidenced by our results on the

Table 2: Comparison of AUC results for Tag&Tab and baseline methods on the MIMIR and Pile benchmarks. The upper table presents the best results from Tag&Tab and baseline methods across four MIMIR datasets, while the lower table shows the best results for four Pile datasets. The best results for each dataset and model size are highlighted in bold, and the second-best AUC is underscored.

Method	DM Mathematics					Github					Pile CC					C4				
	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
Loss	0.85	0.76	0.84	0.68	0.86	0.80	0.85	0.86	0.88	0.88	0.53	0.54	0.54	0.55	0.55	0.50	0.51	0.51	0.51	**0.51**
Zlib	0.68	0.59	0.66	0.55	0.69	0.84	0.88	0.89	0.90	0.90	0.51	0.53	0.53	0.54	0.54	0.51	0.51	0.51	0.51	**0.51**
Min-20% Prob	0.61	0.53	0.70	0.50	0.82	0.80	0.85	0.86	0.88	0.88	0.52	0.53	0.54	0.55	0.55	0.51	0.51	0.51	0.51	0.50
Max-20% Prob	0.63	0.67	0.61	0.58	0.51	0.78	0.85	0.85	0.87	0.86	0.52	0.53	0.53	0.53	0.54	0.51	0.50	0.50	0.50	0.50
Min-K++-20% Prob	0.81	0.79	0.66	0.81	0.73	0.57	0.57	0.61	0.63	0.66	0.51	0.50	0.52	0.53	0.53	0.52	0.51	0.51	0.50	0.50
RECALL	0.80	0.73	0.78	0.64	0.86	0.79	0.76	0.74	0.71	0.72	0.53	0.54	0.54	0.55	0.55	0.51	0.51	0.51	0.51	**0.51**
DC-PDD	0.90	0.86	0.86	0.85	0.86	**0.87**	**0.91**	**0.92**	**0.93**	**0.93**	0.54	0.55	**0.56**	**0.57**	**0.57**	0.51	0.51	0.51	0.51	**0.51**
Ours (Tag&Tab K=4)	**0.96**	**0.96**	**0.96**	**0.95**	**0.95**	0.78	0.82	0.83	0.84	0.85	0.54	**0.56**	**0.56**	**0.57**	**0.57**	**0.53**	**0.52**	**0.52**	**0.52**	**0.51**
Ours (Tag&Tab K=10)	0.92	0.92	0.93	0.92	**0.95**	0.79	0.83	0.84	0.85	0.86	**0.55**	**0.56**	**0.56**	**0.57**	0.56	**0.53**	**0.52**	**0.52**	**0.52**	**0.51**

Method	Ubuntu IRC					Gutenberg					EuroParl					Average				
	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
Loss	0.63	0.59	0.60	0.58	0.58	0.53	0.53	0.53	0.53	0.53	0.52	0.52	0.50	0.52	0.51	0.67	0.67	0.69	0.66	0.70
Zlib	0.52	0.52	0.53	0.54	0.54	0.53	0.60	0.53	0.53	0.53	0.51	0.51	0.50	0.51	0.51	0.63	0.63	0.65	0.62	0.66
Min-20% Prob	0.58	0.57	0.52	0.51	0.52	0.53	0.53	0.53	0.53	0.60	0.53	**0.54**	0.52	0.50	0.51	0.61	0.61	0.65	0.61	0.69
Max-20% Prob	0.69	**0.69**	**0.71**	**0.68**	**0.67**	**0.67**	0.73	0.60	0.67	0.67	0.53	**0.54**	**0.55**	0.53	0.55	0.61	0.64	0.62	0.62	0.60
Min-K++-20% Prob	0.52	0.51	0.52	0.54	0.61	**0.67**	0.60	0.60	0.60	0.60	0.54	0.53	0.51	0.51	0.51	0.60	0.59	0.57	0.62	0.61
RECALL	**0.72**	0.64	0.69	0.64	0.60	0.53	**0.80**	**0.67**	**0.73**	**0.80**	0.51	0.51	0.51	0.55	**0.57**	0.67	0.64	0.65	0.62	0.68
DC-PDD	0.58	0.53	0.53	0.53	0.53	0.53	0.60	0.60	0.53	0.53	0.51	0.52	0.50	0.51	0.54	**0.70**	0.70	0.72	0.71	0.70
Ours (Tag&Tab K=4)	0.64	0.65	0.64	0.66	0.64	**0.67**	0.67	**0.67**	0.67	0.67	0.55	**0.54**	**0.55**	**0.54**	0.56	**0.70**	**0.72**	**0.73**	**0.72**	**0.73**
Ours (Tag&Tab K=10)	0.61	0.63	0.62	0.61	0.62	0.60	0.67	**0.67**	0.67	0.67	**0.56**	**0.54**	**0.55**	**0.54**	0.55	**0.70**	0.71	**0.73**	**0.72**	0.72

DM Mathematics subset (Table 2), where Tag&Tab maintained SOTA performance, achieving an AUC between 0.95 and 0.96.

Despite outperforming baseline MIAs, the AUC achieved by Tag&Tab can still be relatively low in certain cases, hovering around 0.55. However, recent research by Maini et al. (Maini et al., 2024) shows that aggregating multiple MIAs improves dataset membership inference accuracy, emphasizing the need for better attacks that achieve AUC over 0.5 for improved aggregated attack performance. Tag&Tab meets this criterion, making it a valuable component in an ensemble of MIAs for enhanced inference accuracy.

To better understand our method's performance, in Appendix A.1, we examined the impact of our method's tagging stage by comparing the selection of the highest K entropy words with a random token selection, observing that prioritizing the highest K entropy words significantly enhances performance across all models, resulting in superior AUC scores.

Finally, Appendix A.2 reports extra experiments on our tagging choices. Replacing our keywords with random tokens drops accuracy, confirming the value of informed selection. Using only named-entity tokens or only high-entropy tokens each improves on baselines, but combining the two as Tag&Tab gives the strongest results. We also tried ranking tokens with TF-IDF instead of entropy, which led to noticeably lower performance.

6 Conclusion

We present Tag&Tab, a novel black-box method for detecting pretraining data in LLMs. By focusing on the semantic and contextual relevance of words, the method enhances detection capabilities. Tag&Tab outperforms SOTA attacks and consistently achieves high performance across diverse textual data types. Our comprehensive evaluation spans eight textual data types from three datasets (the Pile, MIMIR, and BookMIA) and six LLMs of varying sizes and architectures (LLaMa 1-7b, 13b, 30b, Pythia-160m, 1.4b, 2.4b, 6.9b, 12b, GPT-3.5 Turbo). Our study confirms that the selection of high-entropy keywords improves membership inference attack results, further validating our approach. Future work could extend Tag&Tab by considering keyword context and placement within documents. Additionally, developing new MIAs that leverage advanced NLP techniques to assess word significance could further improve the detection of pretraining data in LLMs.

7 Limitations

While Tag&Tab demonstrates strong performance in detecting pretraining data in LLMs, it has several limitations:

1. **Effectiveness Against Fine-Tuning**: Tag&Tab is less effective in benchmark settings where MIAs are evaluated on models fine-tuned specifically on entire documents. This fine-tuning process involves targeted adaptation to a small set of documents, amplifying memorization across the entire input. In these evaluation scenarios, methods that consider a larger portion of the input text by aggregating probabilities from many tokens tend to achieve better inference performance, as they can capture memorization signals spread throughout the entire document. In contrast, Tag&Tab focuses on a small set of informative keywords, making it less suited for scenarios where fine-tuning amplifies memorization uniformly across all tokens. It is important to note that this evaluation setup does not reflect realistic pretraining data detection, where memorization patterns are typically more localized and sparse.

2. **Language Generalization**: As shown in our experiments, Tag&Tab's effectiveness is currently limited to English texts. When applied to other languages, such as Chinese, the method's performance degrades. Extending Tag&Tab to multilingual settings requires further adaptation of the tagging process.

3. **Black-Box and Training Data Transparency**: Tag&Tab assumes black-box access to LLMs that provide token-level probability outputs. While this is common for many commercial APIs, it is not guaranteed for all models. Furthermore, newer open-source LLMs often do not disclose their exact pretraining datasets, making it challenging to construct reliable member and non-member sets for evaluation. This lack of transparency affects the applicability and benchmarking of MIAs, including ours, on recent models.

8 Ethical Considerations

Our primary goal is to improve the detection of sensitive pretraining data in LLMs, addressing critical issues like copyright violations and data misuse. However, we acknowledge that as an MIA technique, this method could be misused to compromise privacy or extract sensitive information from models.

To mitigate these risks, we carefully selected the datasets used in our evaluations, ensuring they are publicly available and free of personally identifiable information (PII) or other private data. Additionally, our research follows ethical guidelines and emphasizes the importance of transparency in model training and evaluation.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Samuel Axon. 2024. Youtube creators surprised to find apple and others trained ai on their videos. Ars Technica. https://arstechnica.com/ai/2024/07/apple-was-among-the-companies-that-trained-its-ai-on-yo URL https://arstechnica.com/ai/2024/07/apple-was-among-the-companies-that-trained-its-ai-on-yo

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.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022a. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE.

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

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

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from

large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Kent K Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. 2023. Speak, memory: An archaeology of books known to chatgpt/gpt-4. *arXiv preprint arXiv:2305.00118*.

Alibaba Cloud. 2024. Qwen1.5: An enhanced open-source language model series. Accessed: 2024-04-22.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*.

Robert Frost. 1916. The road not taken. Published in 'Mountain Interval'.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Google. 2006. Google patents. https://patents.google.com/. Accessed: 2024-04-22.

Andrew Hoang, Antoine Bosselut, Asli Celikyilmaz, and Yejin Choi. 2019. Efficient adaptation of pretrained transformers for abstractive summarization. *arXiv preprint arXiv:1906.00138*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. 2022. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s):1–37.

Moaiad Ahmad Khder. 2021. Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing & Its Applications*, 13(3).

Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. 2023. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 346–363. IEEE.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. 2024. Llm dataset inference: Did you train on my dataset? *arXiv preprint arXiv:2406.06443*.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison. *arXiv preprint arXiv:2305.18462*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Seth Neel and Peter Chang. 2023. Privacy issues in large language models: A survey. *arXiv preprint arXiv:2312.06717*.

Sergey I Nikolenko. 2021. *Synthetic data for deep learning*, volume 174. Springer.

OpenAI. 2022. Chatgpt. OpenAI. https://platform.openai.com/docs/models. URL https://platform.openai.com/docs/models.

Noorjahan Rahman and Eduardo Santacana. 2023. Beyond fair use: Legal risk evaluation for training llms on copyrighted text. In *ICML Workshop on Generative AI and Law*.

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. *arXiv preprint arXiv:2310.16789*.

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

Robyn Speer. 2022. rspeer/wordfreq: v3.0.

Om Dipakbhai Thakkar, Swaroop Ramaswamy, Rajiv Mathews, and Francoise Beaufays. 2021. Understanding unintended memorization in language models under federated learning. In *Proceedings of the Third Workshop on Privacy in Natural Language Processing*, pages 1–10, Online. Association for Computational Linguistics.

New York Times. 2023. New york times openai microsoft lawsuit. The New York Times. https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html. URL https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. 2019. Demystifying membership inference attacks in machine learning as a service. *IEEE transactions on services computing*, 14(6):2073–2089.

Zige Wang, Wanjun Zhong, Yufei Wang, Qi Zhu, Fei Mi, Baojun Wang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Data management for large language models: A survey. *arXiv preprint arXiv:2312.01700*.

Xiaodong Wu, Ran Duan, and Jianbing Ni. 2024. Unveiling security, privacy, and ethical concerns of chatgpt. *Journal of Information and Intelligence*, 2(2):102–115.

Roy Xie, Junlin Wang, Ruomin Huang, Minxing Zhang, Rong Ge, Jian Pei, Neil Zhenqiang Gong, and Bhuwan Dhingra. 2024. ReCaLL: Membership inference via relative conditional log-likelihoods. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8671–8689, Miami, Florida, USA. Association for Computational Linguistics.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, page 100211.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE.

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

Weichao Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024b. 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.

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 preprint arXiv:2311.01964*.

A Appendix

Our code is available in our anonymous GitHub repository [7].

A.1 The Impact of our Word Selection Method

In Figure 3 we demonstrate the impact of our word selection method, Tag, which selects the highest K entropy words, compared to a random selection of words using the same Tab algorithm. The dataset used for this evaluation was BookMIA. For each model, we presented 10 results using the Tag method (Blue) and 10 results using a random selection of words (Orange). The results indicate that selecting the highest K entropy words improves performance across all models. The Tag method achieved an average AUC of 0.8, compared to an average AUC of 0.64 with a random selection of K words. This demonstrates the effectiveness of the Tag method in enhancing model performance by focusing on high-entropy words.

A.2 Ablation Study: Alternative Word Selection Strategies

To further analyze the contribution of individual word selection strategies, we conducted an ablation study comparing several variants of our method on the BookMIA dataset. Specifically, we evaluated the following alternatives to the full Tag&Tab method, all using $K = 4$ selected keywords:

- **Entropy-Only**: selecting the 4 highest entropy tokens per sentence.
- **NER-Only**: selecting the first 4 named entities identified via spaCy per sentence.
- **TF-IDF**: selecting the top 4 tokens with the highest TF-IDF scores per sentence.

We evaluated these variants across all tested models except GPT-3.5 Turbo, including LLaMa-7B, LLaMa-13B, LLaMa-30B, Pythia-6.9B, and Pythia-12B.

The results, summarized in Table 3, show that both *Entropy-Only* and *NER-Only* achieve competitive performance, slightly below the combined Tag&Tab method. The *TF-IDF* variant performed notably worse, confirming that high-entropy and named-entity tokens are more effective indicators of membership when used together. These findings validate our design choice of combining both strategies for optimal pretraining data detection.

A.3 Threshold Calibration

A.3.1 Calibrating γ with No Member Labels

In a realistic "zero-knowledge" setting, we assume no access to the model's training data, but we can still collect data that were definitely not seen during training. This includes texts published after the model's release or synthetically generated samples.

[7] https://anonymous.4open.science/r/Tag-Tab-0E80/README.md

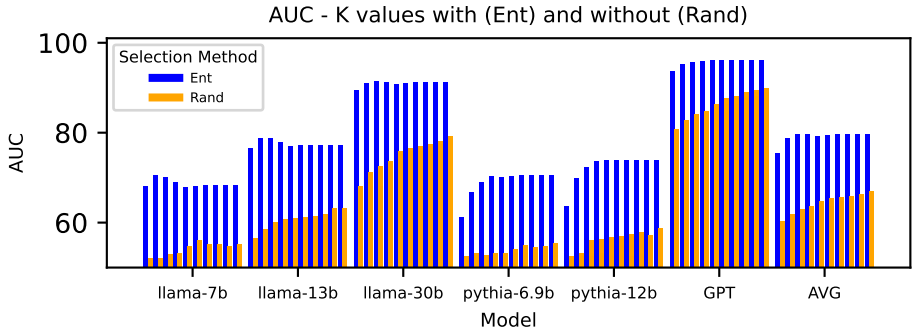

Figure 3: Comparison of Tag&Tab's keyword selection against random selection across different K values. For each model, we report AUC scores when selecting between 1 and 10 keywords per text using the Tag&Tab entropy-based method (blue) and a random selection baseline (orange). Each bar represents the AUC achieved for a specific K value. Results are shown for all evaluated models on the BookMIA dataset, including GPT-3.5 Turbo, as well as the overall average (AVG). The plot highlights the consistent advantage of high-entropy keyword selection across varying K values and model sizes.

Table 3: AUC scores of Tag&Tab and its ablated variants on the BookMIA dataset, all with $K = 4$ selected tokens.

Method	LLaMa-7B	LLaMa-13B	LLaMa-30B	Pythia-6.9B	Pythia-12B
Tag&Tab	**0.69**	**0.78**	**0.91**	**0.72**	**0.75**
Entropy-Only	0.67	0.76	0.89	0.70	0.73
NER-Only	0.62	0.72	0.85	0.68	0.73
TF-IDF	0.60	0.68	0.81	0.65	0.67

To calibrate the decision threshold γ, we compute Tag&Tab scores on a set of known non-member samples and select γ based on the upper tail of the score distribution (e.g., the 95$^{\text{th}}$ percentile). This threshold is then used to flag samples that appear "member-like" relative to the known non-members. Using BookMIA with LLaMA-30B and K=4, this approach yielded $\gamma \approx 0.0392$, resulting in 0.85 AUC on the test split, a modest drop from the 0.91 AUC achieved when member labels were available. Although performance was slightly lower, the method still achieved strong results, demonstrating its potential for practical real-world applications even under limited knowledge conditions.

A.3.2 Domain Mismatch Test

To check the transferability of the threshold, we applied the same $\gamma = 0.0392$ to a maths corpus from MIMIR (DM Mathematics). Performance dropped sharply (AUC < 0.60), showing that score distributions differ across domains. The same effect appears when switching models: each model–dataset pair needs its own calibrated threshold.

A.3.3 Recommendation

Thresholds should be recalibrated whenever the target *model* or *data domain* changes. A small, trustworthy non-member dev set from the intended domain is sufficient; no labelled members are required.

A.4 Generalization to Chinese Texts

To test whether Tag&Tab can generalize to a non-Latin language with a fundamentally different structure, we evaluated it on a Chinese text using the PatentMIA dataset (Zhang et al., 2024b), which contains patents sourced from Google Patents (Google, 2006). This evaluation was performed using the Qwen1.5-14B model[8], an open-source LLM developed by Alibaba Cloud, optimized for Chinese and multilingual understanding (Cloud, 2024). The method still achieved a meaningful signal (AUC = 0.6), but it did not surpass the strongest baseline (DC-PDD, 0.69 AUC). We attribute this gap to structural differences between Chinese and English, such as the absence of explicit word boundaries and different entropy

[8] https://huggingface.co/Qwen/Qwen1.5-14B

distributions, which diminish the effectiveness of our current keyword–entropy heuristic.

A.5 Robustness to Adversarial and Distributional Perturbations

To evaluate the robustness of Tag&Tab against minor textual modifications, we conducted an experiment where 2–5 words per sample in the BookMIA dataset were replaced with suitable synonyms. These changes preserved the original meaning while altering the lexical form, simulating both adversarial-style perturbations and natural distribution shifts.

We evaluated the impact of these modifications across LLaMA-7B, LLaMA-13B, LLaMA-30B, Pythia-6.9B, and Pythia-12B models. In all cases, we used Tag&Tab with $K = 4$ keywords.

The results, presented in Table 4, show that Tag&Tab exhibits only a minor performance drop of 1–2% in AUC across all models. These findings confirm that the method remains effective even when the exact data distribution is unknown, demonstrating resilience to small-scale semantic-preserving shifts.

Table 4: Robustness of Tag&Tab (K=4) under synonym-based perturbations on the BookMIA dataset.

Model	**Original AUC**	**Perturbed AUC**
LLaMa-7B	0.69	0.68
LLaMa-13B	0.78	0.76
LLaMa-30B	0.91	0.89
Pythia-6.9B	0.72	0.71
Pythia-12B	0.75	0.73