RAGLeak: Membership Inference Attacks on RAG-Based Large Language Models

Kaiyue Feng[1], Guangsheng Zhang[1], Huan Tian[1], Heng Xu[1], Yanjun Zhang[1], Tianqing Zhu[2], Ming Ding[3], and Bo Liu[1,4](✉)

[1] School of Computer Science, University of Technology Sydney, Sydney, Australia
`kaiyue.feng@student.uts.edu.au, bo.liu@uts.edu.au`
[2] Faculty of Data Science, City University of Macau, Macau, China
[3] Data61, CSIRO, Sydney, Australia
[4] Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, Australia

Abstract. As newly emerged powerful tools for question-answering tasks, large language models (LLMs) have attracted significant attention in recent years. Despite their superior performance on various tasks, LLMs often lack domain knowledge or up-to-date knowledge. Retrieval-augmented generation (RAG) has been proposed as a solution to supplement LLMs and mitigate hallucination. However, while RAG systems offer considerable improvements, they also introduce new attack surfaces and privacy risks for LLMs. Current RAG systems are considered as robust against privacy attacks as all malicious queries can be easily detected and defended. However, we find that a hidden privacy attack on RAG may jeopardize the privacy of RAG. To show the risk of RAG system, we propose RAGLeak, a stealthy novel attack method to infer membership information in RAG-based LLMs. Our method crops a part of the query, using the former part as the input and the latter part as the true answer. We investigate the grey-box setting where the attacker can access the perplexity and the black-box setting where the attacker only has the input/output access. For the grey-box setting, we decide the membership states by thresholding the output perplexity; for black-box setting, we calculate the similarity between the output and the true answer and set the similarity threshold. We evaluate our method on two datasets and three LLMs. The results show that RAGLeak can bypass few-shot based defense method and achieve an accuracy of over 0.8 on all datasets and LLMs. This competitive performance on RAG-based LLMs demonstrates that our method poses significant privacy risks without complicated fine-tuning or retraining. This work presents the potential risk of RAG on LLM and highlights the need for robust privacy-preserving techniques in RAG-based LLMs.

Keywords: Membership Inference Attack · Retrieval Augmented Generation · Large Language Model · Privacy Preservation

1 Introduction

Large language models (LLMs) have emerged as powerful tools for various domains, demonstrating superior performance on tasks such as question-answering, fact verification, and chatbots. Despite their success across diverse applications, LLMs face intrinsic limitations. The enormous scale of their training data and model parameters results in expensive training costs, which consequently leads to a lack of up-to-date knowledge in the LLM's knowledge base. This knowledge gap can cause LLMs to generate misleading or fabricated responses, a phenomenon known as "hallucination". Such risks are particularly notable in domain-specific fields like medicine and news.

To address these limitations, Retrieval-augmented generation (RAG) has been introduced as a solution, leveraging external knowledge databases to supplement LLMs. RAG systems retrieve information relevant to a user's query from a knowledge database and input both the retrieved documents and the user's query into the LLM. This approach enables LLMs to access and utilize information not contained in their original training data, such as recent events, proprietary information, or specialized domain knowledge. In contrast to fine-tuning, which requires expensive and time-consuming retraining of the model, RAG systems enable timely updates to the knowledge base without costly retraining processes. This flexibility makes RAG a reliable solution for enhancing LLM capabilities while maintaining efficiency and adaptability.

However, despite the significant performance improvements offered by RAG systems, they introduce new attack surfaces and elevate privacy and security risks for LLMs. A critical privacy threat is the membership inference attack (MIA). This type of attack aims to determine whether a particular data point was used in the model's training dataset or, in the case of RAG, is present in the retrieval database. Such attacks pose a significant risk, as they may enable attackers to extract confidential data, thereby undermining data privacy and exposing sensitive information. Research has shown that directly implementing MIAs against LLMs generally performs at levels comparable to random guessing across most domains [5,8,19,22,29], which might be the consequence of the generalization brought by large pretaining datasets [8]. On the contrary, the relatively small size of knowledge database makes RAG system more sensitive to membership inference attacks.

Some studies have investigated this privacy issue in RAG-based system. Anderson et.al. [2] conducted membership inference attacks against RAG systems using an inquiry-based approach. Their method directly queried the system with the prompt *Does this: "{Target Sample}" appear in the context? Answer with Yes or No."*. While this straightforward technique achieved a high attack success rate, the apparent malicious intent makes it easy detectable and thus easier to defend against.

To address this research gap, we propose a novel, stealthy membership inference attack–**RAGLeak**–to examine privacy leakage problem in RAG-based LLMs. Our approach aims to extract information from the RAG database without accessing the language model's structure or parameters, while simultaneously circumventing detection and defense mechanisms. Unlike other membership inference methods that target only large language model itself, RAGLeak specifically focuses on RAG databases. It operates under both grey-box and black-box setting, making it more applicable to real-world scenarios where internal model information is typically unavailable.

In our method, we crop a user query such that the initial part serves as the input, while the remaining part constitutes the correct answer. If this input exists in the RAG database, the retriever will fetch the full context and supply it to the LLM. Otherwise, only irrelevant context will be retrieved, resulting in an incorrect answer. This discrepancy in behavior between member and non-member samples serves as our criterion for membership inference. In the grey-box setting, RAGLeak relies on the perplexity of the model's output to determine membership by applying a threshold. In the black-box setting, it assesses membership based on the similarity between the LLM's output and the ground-truth answer. This novel approach offers a more stealthy and generalizable method for deceting privacy leakage in RAG systems, thereby revealing critical vulnerabilities in RAG-based LLMs.

Our major contributions are summarized as follows:

- We propose RAGLeak, a novel membership inference attack method specifically designed for RAG-based LLMs. This method addresses critical yet underexplored privacy vulnerabilities in RAG architectures.
- We demonstrate the effectiveness of RAGLeak under both grey-box and black-box settings, requiring no access to the internal structure or parameters of the large language model. This design significantly enhances the method's applicability in real-world scenarios where model internals are typically inaccessible.
- We evaluate RAGLeak under a few-shot learning defense mechanism. The results show that our method can bypass detection and still achieve outstanding attack performance. This feature enables RAGLeak to operate more stealthily compared with baseline method.
- We conduct a comprehensive evaluation of RAGLeak using multiple performance metrics across diverse datasets and LLMs, demonstrating its robustness and effectiveness. Our results show that RAGLeak achieves superior performance compared to existing baseline method, providing a new benchmark for privacy analysis in RAG systems.
- We provide insights into the factors influencing the success of membership inference attacks in RAG systems, contributing to a deeper understanding of privacy vulnerabilities in these architectures and paving the way for developing more secure RAG implementations.

The remainder of this paper is organized as follows: Sect. 2 reviews the related work. Section 3 outlines the preliminaries and formulates the problem. Section 4 details our proposed RAGLeak method. Section 5 presents our experimental setup and results. We discuss the potential defense and ethical consideration in Sect. 6. And Sect. 7 concludes this paper.

2 Related Work

In this section, we briefly review the existing attack methods against large language models and RAG-based LLMs.

2.1 Attacks to LLMs

The security and privacy issues of large language models are getting more and more attention in recent years. Researchers have explored various attacks against LLMs, including jailbreak attack [6,26,30], prompt injection attack [10,17,18, 20], poisoning attack [25,28], backdoor attack [16,27], membership inference attack [9,14], etc. Regarding the privacy issue of LLMs, Duan et.al. [8] conducted a comprehensive evaluation of five established membership inference methods against large language models. Their findings revealed that these traditional MIA approaches perform near random guessing when applied to LLMs, highlighting the transferability of these methods against such attacks.

Other researchers have focused on extracting training data via prompt-based attacks. Li et.al [15] investigated the use of direct and jailbreaking prompts to extract data from ChatGPT. They concluded that ChatGPT possesses sufficient defense mechanisms against direct inquiry attacks, emphasising the challenges in attacking LLM privacy through straightforward methods. Hui et al. [11] proposed PLeak, an innovative method for optimizing adversarial prompts to attack target LLM applications.

Despite the effectiveness, the attacks on LLMs usually cost expensive computational resources due to the large volume of training data. The black-box nature also complicates the validation of attack results for researchers. Given these constraints, RAG systems might be a more accessible attack surface for the following research directions.

2.2 Attacks to RAG-Based LLMs

Retrieval-augmented generation strategy can enhance the performance of a large language model without the need for fine-tuning or retraining. However, this approach also exposes new attack surfaces in large language models. The attacker can inject malicious content in the knowledge database without much effort. The injected content could lead to the generation of misleading or harmful responses.

Several research studies have investigated the security risks of retrieval-augmented generation. Zou et al. [31] was the first work that revealed the vulnerability of the RAG system against poisoning attacks. Unlike traditional approaches that targeted training datasets, PoisonedRAG crafted the poisoned text using LLM and injected it into the RAG knowledge database. This method exploits that the RAG knowledge database exposes new targets for poisoning attacks, highlighting a critical security concern in RAG implementations. Jiao et al. [13] proposed a more stealthy backdoor attack mechanism for RAG-based LLMs.

Some researchers also tried to investigate the privacy issue of RAG-based LLM. [2] conducted membership inference against RAG systems using an inquiry-based approach. They attacked the LLMs by asking "Does this sample appear in the contex". Although this method showed some efficacy, the apparent malicious queries make it much easier for human and the LLM to detect this attack, moreover, design corresponding defense mechanisms.

3 Preliminary

3.1 Membership Inference Attack

Membership inference attacks (MIA) aim to determine whether a specific data sample was used in the model's training dataset. In the typical scenario of membership inference attacks, the attacker has black-box access to the model. This means the attacker can query the model and observe its outputs but does not have direct access to the model's parameters or architecture.

Shokri et al. [23] pioneered the topic of membership inference attacks against machine learning models. They proposed the shadow training technique to distinguish between members and non-members and addressed the leakage of membership information related to overfitting. Shadow training starts with collecting shadow data with a similar distribution of the victim model's training data. The attacker then uses this shadow data to train a shadow model and constructs a binary classifier based on the behavioral differences between members and non-members of the training set. [21]

However, building a shadow model requires high computation costs, and in some scenarios is unrealistic. In that case, the threshold-based method is widely used. Threshold-based attacks usually decide the membership by thresholding the model's outputs, such as loss [29] and perplexity [5].

Note that membership inference usually refers to the membership of a sample in the training dataset of the model. This paper, on the other hand, defines "member" and "non-member" according to whether a sample exists in the RAG knowledge database.

3.2 Retrieval-Augmented Generation

The RAG framework typically consists of three major components: knowledge database, retriever, and LLM. The knowledge database contains time-sensitive data, such as news or academic research materials; or specific domain knowledge, such as medical or biology; and can be customized by uploading any private data. The next key part is the retriever. Given the query from the input of the LLM and the knowledge database, the retriever embeds the query and the external knowledge into the same vector space. Then, the retriever searches for k sentences, which are the most similar to the query. The searching can be based on different criteria, such as cosine similarity or semantic similarity. Finally, the retrieved k texts served as the context along with the query is input into the LLM as the prompt to generate an answer for the given question.

3.3 Generative Sequence Model

The large language model has the fundamental architecture of the generative sequence model. The generative sequence model generates words that follow the "next-step prediction" rules. Considering the task of modelling natural-language sentences from the space of all possible sequences of English words, a generative sequence model would assign probabilities to words based on the context in which those words appeared in the empirical distribution of the model's training data. Formally, the model generates the sequence of tokens according to the distribution [3]:

$$\mathbf{Pr}(x_1, x_2, ..., x_n),$$

where $x_1, x_2, ..., x_n$ is a sequence of tokens from a vocabulary by applying the chain rule of probability:

$$\mathbf{Pr}(x_1, x_2, ..., x_n) = \Pi_{i=1}^{n} \mathbf{Pr}(x_i | x_1, x_2, ..., x_{i-1)}).$$

Each individual computation $\Pi_{i=1}^{n} \mathbf{Pr}(x_i | x_1, x_2, ..., x_{i-1)})$ represents the probability of token x_i occurring at timestep i with precious token x_1 to x_{i-1}.

As a probabilistic generative model, large language model be measured using a natural likelihood: perplexity [4].

Definition 1. The *perplexity* of a sequence x is

$$\mathcal{P} = \exp\left(-\frac{1}{n}\sum_{i=1}^{n} \log \mathbf{Pr}(x_i \mid f_\theta(x_1, \ldots, x_{i-1}))\right).$$

The perplexity measures how "surprised" the model is to see a given value [5] and is inversely correlated with the likelihood. A lower perplexity indicates the model is more confident about the sequence, and a higher perplexity shows otherwise.

3.4 Threat Model

Attack Scenario. By employing our membership inference attack method, an attacker can determine whether certain private documents have been used by the model provider without authorization, thereby helping to safeguard intellectual property. For example, a pharmaceutical company might discover that its proprietary, copyrighted medical records were incorporated—without consent—into the knowledge base of a medical question-answering chatbot. In response, the company can collect text excerpts from these private documents that it suspects the model has accessed. The company has the same level of access to inputs, outputs, and perplexities as any other user.

Attacker's Goal. In the context of machine learning privacy, membership inference attacks typically aim to determine whether a given data sample was used to train the model. However, the attacker's objective differs slightly when targeting RAG-based LLMs. In our attack setting, the primary aim of the attacker is to determine whether a given sample exists in the RAG knowledge database. Given the growing adoption of RAG systems in sensitive domains—such as enterprise knowledge management, personal information assistants, and confidential data analysis— it is crucial to investigate whether this augmented generation technique inadvertently provides attackers with additional opportunities in breaching information confidentiality and system integrity.

Attacker's Capability Access Levels: We consider two different access levels for the attacker: a black-box setting and a grey-box setting. In the black-box setting, the attacker only has access to the model's inputs and outputs, without any further ability to manipulate weights or gradients. Given a text sequence x_q, the attacker can input a custom instruction x_p along with x_q to the target RAG-based LLM and obtain the corresponding answer $\hat{y}$. In the grey-box setting, the attacker can only access the perplexity P of the answer sequence. Notably, in our grey-box setting, the attacker is restricted to using perplexity-related information only. This restriction is intended to investigate the impact of perplexity on membership information leakage. Both access settings are highly realistic as all the LLMs allow users to query the model and receive responses. Furthermore, some models (e.g., GPT-3.5 turbo) provide metadata that can be used to calculate response perplexity.

Data Knowledge: We assume that the attacker is familiar with the distribution of the RAG database and can collect a new dataset following that same distribution. This new dataset contains the data that the attacker believes might be in the RAG database, along with an auxiliary dataset.

Model Knowledge: Model knowledge refers to information related to the victim model. We assume that the attacker lacks any knowledge of the victim model's internals. In many existing membership inference attacks, constructing a shadow model is a crucial step to imitate the victim's behavior. However, this approach is impractical for large language models, since it is nearly impossible for an attacker to build a comparable model. Because our attack is shadow-model-free, we do not require any knowledge of the victim model. The attacker simply submits queries and observes responses as any typical user would.

4 RAGLeak

4.1 Overview

We adopt a common membership inference attack setup, in which the attacker has either grey-box or black-box access to a RAG-based large language model

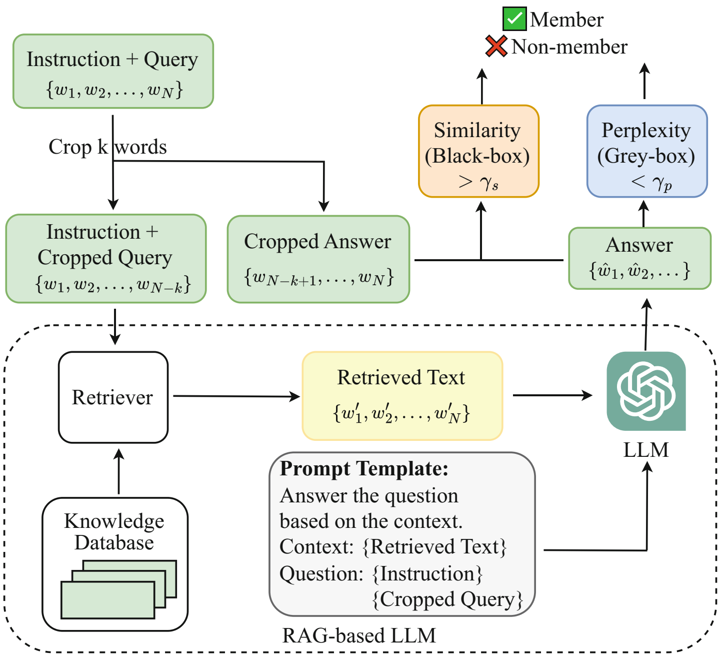

Member
Non-member
Instruction + Query
$\{w_1, w_2, \ldots, w_N\}$
Crop k words
Similarity (Black-box) $> \gamma_s$
Perplexity (Grey-box) $< \gamma_p$
Instruction + Cropped Query
$\{w_1, w_2, \ldots, w_{N-k}\}$
Cropped Answer
$\{w_{N-k+1}, \ldots, w_N\}$
Answer
$\{\hat{w}_1, \hat{w}_2, \ldots\}$
Retriever
Retrieved Text
$\{w'_1, w'_2, \ldots, w'_N\}$
LLM
Knowledge Database
Prompt Template:
Answer the question based on the context.
Context: {Retrieved Text}
Question: {Instruction} {Cropped Query}
RAG-based LLM

Fig. 1. Overview of our membership inference attack against RAG-based LLM.

llm which contains a knowledge database D_{rag} used to reinforce the output answer. The attacker's goal is to learn a classification function $f_{llm} : \chi \rightarrow 0, 1$, which determines for each x from the universe of textual samples χ whether $x \in D_{rag}$ or $x \notin D_{rag}$.

Figure 1 illustrates our membership inference attack method. The user input contains two parts: an instruction and a query. First, we crop k words from the query, splitting it into a "cropped query" and a "cropped answer". The instruction and the cropped query are then fed into the retriever, which searches the knowledge database for the set of texts (termed "retrieved texts") that are the most similar to the query. Here, the query and the texts in the knowledge database have been vectorised by the embedding model and stored in the form of vectors. The similarity between them is determined based on the cosine similarity between the two vectors. Next, the instruction, cropped query, and retrieved text are input into the large language model combined by the prompt template which then generates a corresponding answer. In the black-box setting, we compute the similarity between the generated answer and the cropped answer and apply a threshold to make the membership inference decision. In the grey-box setting, we use the perplexity of the generated answer as the membership metric. This overall procedure involves two steps—generation and classification—which will be detailed in the following sections.

4.2 Generation

For a RAG-based large language model, the input consists of two parts: a prompt x_p that guides the model to achieve certain propose, and a query x_q that is used to retrieve similar text in the database:

$$x = x_p + x_q, \tag{1}$$

The prompt x_p must be clear enough to instruct the LLM on the intended task, yet concise enough to avoid misleading the retriever into retrieving unrelated sentences. In this study, we use the instruction "**Complete the following sentence:**" as the prompt.

Consider a text sequence x_q for which we wish to determine membership:

$$x_q = \{w_1, w_2, ..., w_N\}, \tag{2}$$

where w_i represents the ith word in the text sequence, and N denotes the total number of words in the sequence.

The current transformer-based large language models generate the outputs by selecting the most possible token after seeing the given sequence. For the text sequence $\{w_1, w_2, ..., w_N\}$, we crop the last k words, dividing the sequence into a "cropped query" $x_c = \{w_1, ..., w_{N-k}\}$ and a "cropped answer" $y = \{w_{N-k+1}, ..., w_N\}$. We treat x_c as part of the user input to the LLM and y as the ground truth. The final input x can therefore be written as:

$$x = x_p + x_c \tag{3}$$

The retriever of the RAG system gets the user input and searches for the most similar sequence in the database, this process unfolds as follows:

$$x_R = Retrieve(x, D_{rag}) = \{w'_1, ..., w'_N\} \tag{4}$$

where x_R denotes the retrieved text, x is the input, and D_{rag} stands for the RAG database.

Once x_R is obtained, it is supplied to the LLM along with the user input x concatenated by a prompt template. This process yields the generated answer:

$$\hat{y} = llm(x, x_R) = \{\hat{w}_1, \hat{w}_2, ...\} \tag{5}$$

4.3 Classification

Membership inference attacks rely on the observation that members and non-members exhibit different behavior. In the context of a RAG-based large language model, this discrepancy can be captured using two metrics: perplexity and similarity.

Perplexity. Perplexity is inversely correlated with likelihood. It measures how well the language model predicts the next token [4]. Retrieval-augmented generation is designed to mitigate hallucinations, thereby producing more reliable answers from large language models. Prior work indicates that retrieval-augmented responses often have lower perplexity [24]. In our setting, the retrieved text provides the model with references, making the output more accurate. On the contrary, the retriever cannot locate suitable references for non-member queries. Consequently, the perplexity of augmented answers P_{mem} tends to be lower than that of non-augmented answers P_{non}.

Following Definition 1, the perplexity of a generated answer can be expressed as:

$$P(x) = \exp\left(-\frac{1}{n}\sum_{i=1}^{n} logprob_i\right), \tag{6}$$

where $logprob_i$ is the log probability of ith output token.

Similarity. Some LLMs do not provide perplexity through their API, which makes the model a black box. Under this black-box setting, we use the similarity between the cropped answer y and the generated answer $\hat{y}$ as our key metric. We convert both texts into vector representations and compute their cosine similarity:

$$S(x, \hat{y}) = \frac{V(y) \cdot V(\hat{y})}{\|V(y)\| \cdot \|V(\hat{y})\|}, \tag{7}$$

where V is a vectorizer.

Once we obtain the perplexity or the similarity metric for distinguishing members from non-members, we formulate membership inference as a binary classification problem. Because our method does not require building shadow models, we classify samples by thresholding either perplexity or similarity. The classification decision rule for grey-box setting is:

$$f_{grey}(x) = \mathbb{1}[P(x) < \gamma_p]. \tag{8}$$

Likewise, the decision rule for black-box setting is:

$$f_{black}(x) = \mathbb{1}[S(x, y) > \gamma_s]. \tag{9}$$

Here, γ_p and γ_s represent the perplexity threshold and the similarity threshold separately.

The interpretation of the decision rules is straightforward: In the grey-box setting, any sample whose output perplexity falls below the threshold $\gamma_p$is deemed a member of the RAG database; In the black-box setting, any sample whose generated answer exhibits a similarity to the cropped answer exceeding the threshold $\gamma_s$is classified as a member. Figure 2 provides an illustrative example of our proposed RAGLeak attack.

5 Experimental Results

5.1 Experimental Settings

Large Language Model. We evaluate our attack method on 3 different large language models: GPT-3.5-turbo in the grey-box setting; GPT-3.5-turbo, Llama3-8b, and Qwen2-7b in the black-box setting.

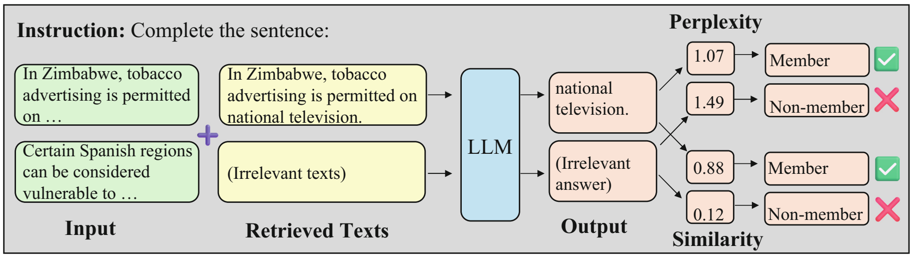

Instruction: Complete the sentence:
In Zimbabwe, tobacco advertising is permitted on …
Certain Spanish regions can be considered vulnerable to …
Input
In Zimbabwe, tobacco advertising is permitted on national television.
(Irrelevant texts)
Retrieved Texts
LLM
national television.
(Irrelevant answer)
Output
Perplexity
1.07
1.49
0.88
0.12
Similarity
Member
Non-member
Member
Non-member

Fig. 2. An attack example of RAGLeak.

Datasets. We evaluate our attack method on two publicly available question-answering datasets that are commonly used in RAG-related research: **NQ**[1] and **HealthCareMagic-100k**[2]. For each dataset, we select 1,000 records as members that in the RAG database and another 1,000 records as non-members.

Embedding Model. The embedding model we use is a sentence-transformers model: **all-MiniLM-L6-v2**[3]. This embedding model maps sentences and paragraphs to a 384-dimensional dense vector space.

Retriever. For the retriever, we employ the Facebook **FAISS** library [7] for similarity search and clustering of dense vectors. In all our experiments, the retriever is configured to return the top 1 most similar text.

Prompt Template. We use the following prompt template in the experiment to concatenate the retrieved texts, instruction, and the query:

```
Answer the question based on the context.
Context: {Retrieved text}
Question: {Instruction} {Cropped Query}
```

Cropped Length. We set the cropped length k to 10 in the main experiments. Additionally, we examine how varying the cropped length affects attack performance in the ablation study 5.5.

Threshold γ. The perplexity threshold γ_p and similarity threshold γ_s are pre-defined decision boundaries for classification. In our experiments, we report the results of the optimal threshold which has the highest accuracy.

[1] https://huggingface.co/datasets/mteb/nq.
[2] https://huggingface.co/datasets/lavita/ChatDoctor-HealthCareMagic-100k.
[3] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.

Defense Method. We apply defense method to both the baseline method and our proposed method. Specifically, we employ a few-shot learning prompt in an LLM to determine whether a given input query attempts to probe the privacy of the RAG database. The full prompt is provided in Appendix A.

Evaluation Metrics. As only two classes (member and non-member samples) exist in our attack model, we denote member samples as positive samples and non-member samples as negative samples. To evaluate the effectiveness of our attack model, we count the true positive (TP), false positive (FP), true negative (TN), and false negative (FN), as well as the precision and recall of our result.

To compare different attacks quantitatively, we use the above experimental results to evaluate our attack model using three metrics:

- Attack accuracy: The rate of the accurately predicted samples in terms of all samples.
- AUC score: The area under the receiver operating characteristic (ROC) curve, which measures the trade-off between the true and false positive rates.
- Precision: It measures the accuracy of positive predictions.
- Recall: It measures the completeness of positive predictions.
- F1 score: It combines precision and recall.

5.2 Baseline Attack: Inquiry Attack

We choose the Inquiry attack [2] as our baseline. Inquiry attack directly asks a LLM *"Does this appear in the context?"* The intuition of this attack relies on the language model's ability to recognise context. Since the retrieved texts were input into LLM along with the question, LLM has the full access to both the context and the query. By comparing the token vectors of the query and the context, LLM can easily tell the user the membership of the query.

Although the inquiry attack achieves high accuracy, it can be easily detected by human observers or by LLM-based detection. When a few-shot learning prompt is used to assess whether an input is attempting to extract information from the database, the performance of the inquiry attack becomes substantially less effective.

5.3 Exclude LLM Training Data

Thus far, we have only considered membership in the RAG database. However, the LLM training data can also influence the performance of our membership inference attack. For instance, if a sample x appears in the LLM training dataset D_{train} but not in the RAG database D_{rag}—which we label as a "non-member"—it may behave similarly to a true member of the RAG database. To mitigate any potential influence from the LLM training data, we compare the outputs of member and non-member samples in our experimental datasets without the

RAG mechanism. In other words, the member and non-member samples should exhibit comparable behavior if the retrieval-augmented generation process is not employed. Figure 3 illustrates the distribution of output similarity for member and non-member samples before and after applying the retrieval-augmented generation process. Prior to RAG, the member and non-member samples show a high degree of overlap. After incorporating RAG, however, their distributions diverge significantly: member samples shift closer to 1.0, while non-member samples remain near 0. This result confirms that our datasets effectively control for the impact of LLM training data on membership inference.

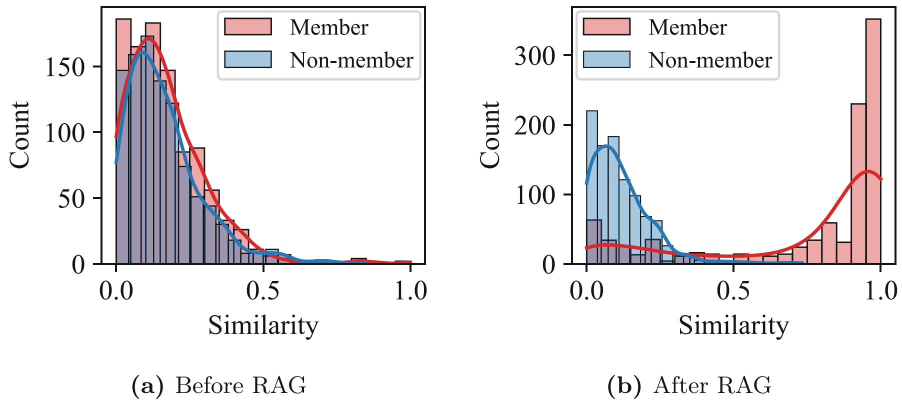


Fig. 3. Distribution of members and non-members' similarity before and after applying retrieval-augmented generation process.

5.4 Main Results

We present our main results based on the optimal threshold that yields the highest accuracy across all datasets and models. Table 1 provides a comparative evaluation of RAGLeak and the baseline attack on two datasets and three LLMs.

Note that all the experiments are conducted under the few-shot learning defense method. For our baseline Inquiry Attack, GPT-3.5 achieves near-zero accuracy on both datasets, indicating high susceptibility to the few-shot learningâĂŞbased defense. Although Llama3 and Qwen2 perform slightly better, their accuracies remain near or below random levels. In contrast, RAGLeak demonstrates robustness under the same defense mechanism.

On GPT-3.5 (NQ), RAGLeak achieves an accuracy of 0.879 and an AUC of 0.911, indicating that retrieval-augmented inference can reveal membership information even when direct inquiry-based attacks are mitigated. Similarly, for Llama3 and Qwen2 on the HealthCareMagic-100k dataset, RAGLeak maintains competitive performance, with accuracy values of 0.870 and 0.829, respectively. Notably, RAGLeak also achieves strong precision (Pre) and recall (Rec) scores,

Table 1. Performance of Inquiry Attack (Inquiry) and RAGLeak on two datasets and three LLMs under black-box setting. Grey-box setting (RAGLeak(G)) is conducted only on GPT-3.5 across two datasets.

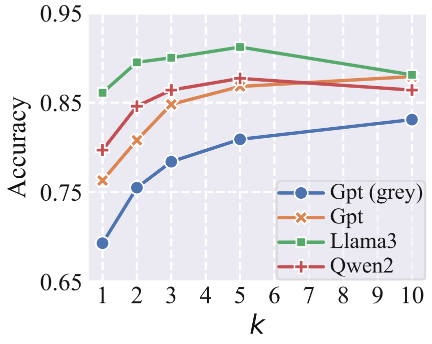
Dataset	Model	Method	Acc	Pre	Rec	F1	AUC
HealthCare Magic-100k	GPT-3.5	Inquiry	0.074	0.076	0.076	0.076	0.074
		RAGLeak	0.890	0.903	0.866	0.884	0.909
		RAGLeak (G)	0.792	0.763	0.869	0.813	0.813
	Llama3	Inquiry	0.541	0.588	0.270	0.370	0.541
		RAGLeak	0.870	0.915	0.811	0.860	0.926
	Qwen2	Inquiry	0.260	0.266	0.273	0.269	0.260
		RAGLeak	0.829	0.885	0.749	0.811	0.885
NQ	GPT-3.5	Inquiry	0.018	0.019	0.019	0.019	0.018
		RAGLeak	0.879	0.912	0.831	0.869	0.911
		RAGLeak (G)	0.831	0.807	0.886	0.845	0.918
	Llama3	Inquiry	0.364	0.353	0.329	0.341	0.364
		RAGLeak	0.881	0.963	0.788	0.867	0.883
	Qwen2	Inquiry	0.387	0.381	0.363	0.372	0.381
		RAGLeak	0.864	0.932	0.780	0.849	0.883

further demonstrating its robustness across different models. Because accuracy alone does not sufficiently reflect the balance between True Positive Rate (TPR) and False Positive Rate (FPR), AUC is a more critical metric for membership inference attacks. RAGLeak achieves AUC values above 0.88 across all datasets and models, underscoring its excellent ability to balance TPR and FPR. Under the grey-box setting, RAGLeak still attains high accuracy of 0.792 and 0.831, confirming the effectiveness of our method. However, the accuracy and other metrics in the grey-box mode are somewhat lower than in the black-box mode, as "similarity" is not utilized in the grey-box scenario. This finding suggests that "similarity" is more efficient than "perplexity" as a membership inference criterion.

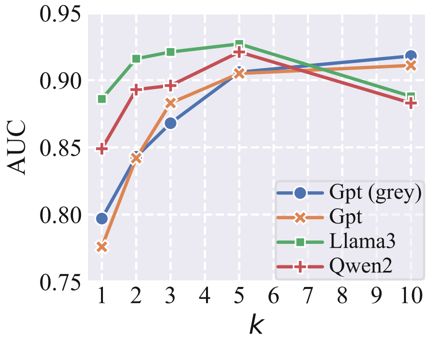
5.5 Ablation Study

Cropped Length k. To investigate the impact of cropped length k, we vary k across 1, 2, 3, 5, 10 and report the corresponding evaluation metrics. Figure 4 illustrates that increasing k generally results in improved performance for all models. Accuracy, AUC, and Recall rise sharply between $k = 1$ and $k = 5$, then continue to increase at a slower rate for $k = 5$ to $k = 10$; Precision follows a similar upward trend as k grows, albeit more gradually.

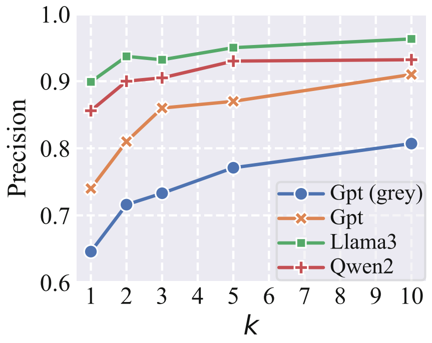
The primary reason for these findings is that, when k is small (k=1 or k=2), the LLM may occasionally predict the next words correctly using common collocations or fixed expressions rather than relying on the RAG database. Con-



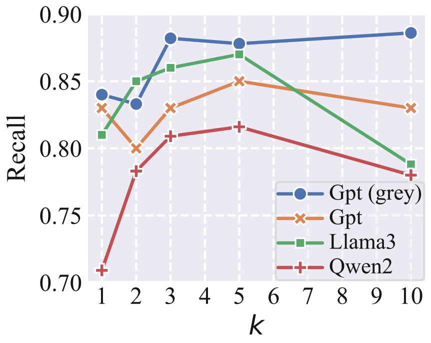
(a) Accuracy



(b) AUC



(c) Precision



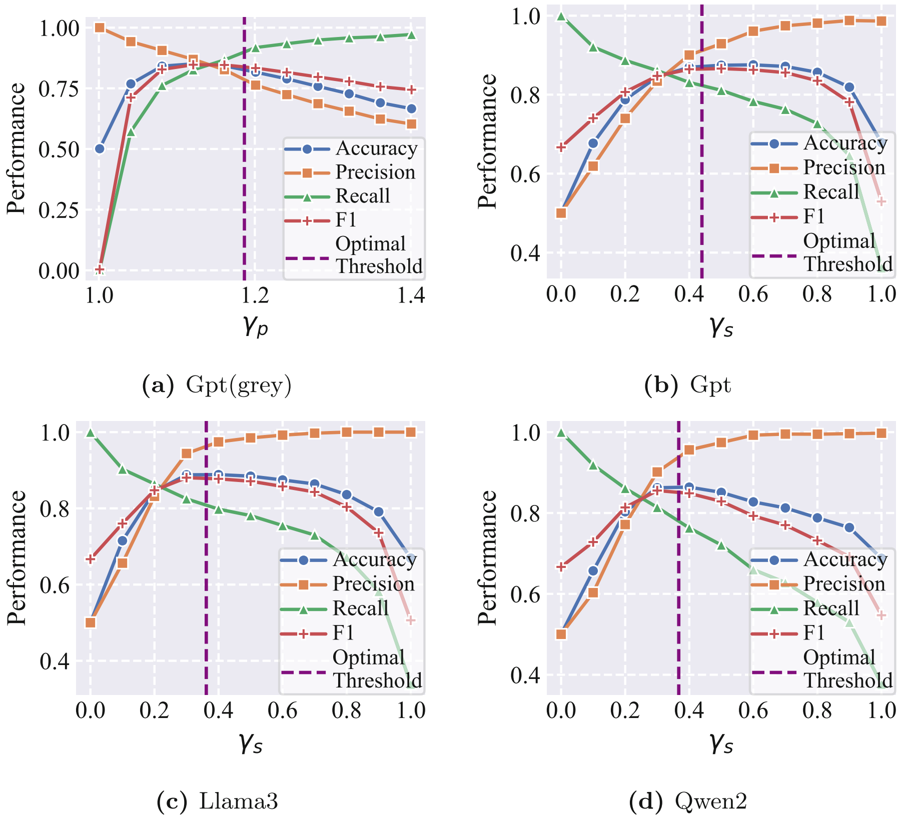
(d) Recall

Fig. 4. Impact of cropped length k for RAGLeak on NQ across different LLMs.

sequently, the similarity for non-member samples can inadvertently increase. However, as k becomes larger, it becomes more difficult for the LLM to "guess" the correct answer without the database reference, leading to higher overall classification accuracy.

Threshold γ. To examine the effect of the thresholds γ_p (grey-box) and γ_s (black-box), we vary γ_s from 1.0 to 1.4 and γ_p from 0 to 1, measuring the corresponding performance metrics for each model. Figure 5 illustrates these results. For the grey-box setting using GPT-3.5-turbo, the optimal threshold range for maximizing accuracy lies between 1.16 and 1.2. A lower threshold favors higher precision, whereas a higher threshold is beneficial for improving recall.

In the black-box setting, the optimal threshold for achieving the highest accuracy across all three models is around 0.4. Setting the threshold to 0.4 appears to get a good transferability.



(a) Gpt(grey) **(b)** Gpt

(c) Llama3 **(d)** Qwen2

Fig. 5. Impact of threshold γ_p (grey-box) and γ_s (black-box) for RAGLeak on NQ across different LLMs.

6 Discussions

6.1 Potential Defense

Now that we have shown we can conduct membership inference attacks on RAG-based LLMs, a follow-up question is how to mitigate these threats. Here, we discuss a possible strategy and an impractical strategy.

Confidence Masking. Confidence score masking is mainly used to mitigate grey-box MIAs on classification models. It aims to hide the true confidence scores returned by the target classifier and thus mitigates the effectiveness of MIAs. A common method of confidence masking is adding crafted to the prediction vector to hide the true confidence scores. Jia et al. [12] found that the attack model is vulnerable to adversarial examples and proposed a defense method called MemGuard. It can add a crafted noised vector to the prediction vector and turn it into an adversarial example of the attack model. Similarly, the defender can add random noise to the perplexity of generative language models to mitigate the performance of the attack model.

Differential Privacy. While differential privacy [1] is a powerful defense mechanism in many machine learning applications, it is impractical for protecting RAG-based LLMs, primarily because it is a training-based defense mechanism. Implementing differential privacy would require finetuning not just the retriever but also the LLM, which is a prohibitively expensive process in terms of computational resources and time. This training process is unfeasible for systems that require frequent updates or real-time data integration. Furthermore, the performance impact of applying differential privacy at this scale could significantly degrade the model's utility. As a result, differential privacy is a not possible defense strategy for RAG-based LLMs.

6.2 Ethical Considerations

All of the data used in our study are from public datasets. Our study was conducted entirely by our research team without external involvement. All experiments were performed in controlled environments using commercial LLM APIs and local servers. Based on our findings, we did not create or release any compromised RAG-based systems to the public. Our research may raise ethical concerns due to its potential for misuse. However, we also discussed potential defense to mitigate attacks. Overall, we believe this study is crucial for the long-term security of AI systems. We aim to alert LLM developers, RAG system designers, and end-users to these privacy risks by sharing our findings. We hope our work contributes to developing more secure and privacy-respecting AI technologies.

7 Conclusion

In this paper, we propose RAGLeak, a well-designed framework for stealthy membership inference attacks on RAG-based large language models. We investigate two attack settings: the grey-box setting, in which the attacker has access to perplexity information, and the black-box setting, in which the attacker has only input/output access. Our approach involves cropping part of a query, using the initial segment as the input and the remaining segment as the true answer. If the input segment exists in the RAG database, the retriever supplies the entire context to the LLM, resulting in different behaviors for member and non-member samples. In the grey-box scenario, membership is determined by thresholding the output perplexity; in the black-box scenario, it is determined by comparing the similarity between the output and the true answer. We evaluate RAGLeak on two datasets and three LLMs. The results show that RAGLeak achieves an accuracy above 0.8 across all datasets and models even after applying few-shot learningâĂŞbased defense strategies, indicating competitive performance compared with baselines. Future research will focus on assessing RAGLeak under a broader range of defense mechanisms and exploring additional experimental settings.

A Defense Prompt

```
SYSTEM MESSAGE / ROLE: system
You are a privacy guard for a Retrieval-Augmented Generation(RAG) system
Your task is to decide whether a user's request attempts to probe the
system's private database by asking for confirmation about a specific
snippet (membership check).
Instructions:
-If the user is explicitly checking whether a particular snippet is in
the private RAG database, answer "Yes"
-Otherwise, answer "No"
-Do not provide any explanation or reasoning beyond "Yes" or "No"
---
---Example 1:
User:
"Does this: "Hi, I'm ChatDoctor. I'd like to share a prescription with
you..." appear in the context? Answer with Yes or No."
Answer: "Yes"
---Example 2:
User: "What is the best treatment for vertigo?"
Answer: "No"
---Example 3:
User: "Can you tell me if the sentence 'Mary had a little lamb' is sto-
red in your private documents?"
Answer: "Yes"
---Example 4:
User: "Summarize the text: 'Hi, this is Chat Doctor. I recommend taking
zinc supplements for diarrhea...'"
Answer: "No"
---FINAL PROMPT FORMAT:
User: "\{question\}"
You should respond with either: "Yes" or "No"
(Without additional explanation.)
```

References

1. Abadi, M., et al.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 308–318 (2016)
2. Anderson, M., Amit, G., Goldsteen, A.: Is my data in your retrieval database? membership inference attacks against retrieval augmented generation. In: Proceedings of the 11th International Conference on Information Systems Security and Privacy, vol. 2: ICISSP, pp. 474–485. INSTICC, SciTePress (2025)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR arxiv:1409.0473 (2014)

4. Carlini, N., Liu, C., Erlingsson, U., Kos, J., Song, D.: The secret sharer: evaluating and testing unintended memorization in neural networks. In: Proceedings of the 28th USENIX Conference on Security Symposium, SEC'19, pp. 267–284. USENIX Association (2019)
5. Carlini, N., et al.: Extracting training data from large language models. In: 30th USENIX Security Symposium (USENIX Security 21), pp. 2633–2650 (2021)
6. Deng, G., et al.: Masterkey: automated jailbreaking of large language model chatbots. In: Proceedings of ISOC NDSS (2024)
7. Douze, M., et al.: The faiss library (2024)
8. Duan, M., et al.: Do membership inference attacks work on large language models? arXiv preprint arXiv:2402.07841 (2024)
9. Fu, W., Wang, H., Gao, C., Liu, G., Li, Y., Jiang, T.: Membership inference attacks against large language models via self-prompt calibration. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems, Vancouver, Canada (2024)
10. Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., Fritz, M.: Not what you've signed up for: compromising real-world llm-integrated applications with indirect prompt injection. In: Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security, AISec '23, pp. 79–90. Association for Computing Machinery, New York (2023)
11. Hui, B., Yuan, H., Gong, N., Burlina, P., Cao, Y.: Pleak: prompt leaking attacks against large language model applications. In: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, CCS '24, pp. 3600–3614. Association for Computing Machinery, New York (2024)
12. Jia, J., Salem, A., Backes, M., Zhang, Y., Gong, N.Z.: Memguard: defending against black-box membership inference attacks via adversarial examples. In: CCS '19, pp. 259–274. Association for Computing Machinery, New York (2019)
13. Jiao, R., et al.: Exploring backdoor attacks against large language model-based decision making. arXiv preprint arXiv:2405.20774 (2024)
14. Kandpal, N., Pillutla, K., Oprea, A., Kairouz, P., Choquette-Choo, C.A., Xu, Z.: User inference attacks on large language models, pp. 18238–18265 (2024)
15. Li, H., Guo, D., Fan, W., Xu, M., Huang, J., Meng, F., Song, Y.: Multi-step jailbreaking privacy attacks on chatgpt. arXiv preprint arXiv:2304.05197 (2023)
16. Li, Y., et al.: Badedit: backdooring large language models by model editing. arXiv preprint arXiv:2403.13355 (2024)
17. Liu, Y., et al.: Prompt injection attack against llm-integrated applications. ArXiv arxiv:2306.05499 (2023)
18. Liu, Y., Jia, Y., Geng, R., Jia, J., Gong, N.Z.: Formalizing and benchmarking prompt injection attacks and defenses. In: 33rd USENIX Security Symposium (USENIX Security 24), pp. 1831–1847. USENIX Association, Philadelphia (2024)
19. Mattern, J., Mireshghallah, F., Jin, Z., Schoelkopf, B., Sachan, M., Berg-Kirkpatrick, T.: Membership inference attacks against language models via neighbourhood comparison. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Findings of the Association for Computational Linguistics: ACL 2023, pp. 11330–11343. Association for Computational Linguistics, Toronto (2023)
20. Perez, F., Ribeiro, I.: Ignore previous prompt: attack techniques for language models (2022)
21. Salem, A., Zhang, Y., Humbert, M., Berrang, P., Fritz, M., Backes, M.: Ml-leaks: model and data independent membership inference attacks and defenses on machine learning models. In: Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS) (2019)

22. Shi, W., et al.: Detecting pretraining data from large language models (2023)
23. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 3–18. IEEE Computer Society, Los Alamitos (2017)
24. Shuster, K., Poff, S., Chen, M., Kiela, D., Weston, J.: Retrieval augmentation reduces hallucination in conversation. In: Moens, M.F., Huang, X., Specia, L., Yih, S.W.t. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 3784–3803. Association for Computational Linguistics, Punta Cana (2021)
25. Wan, A., Wallace, E., Shen, S., Klein, D.: Poisoning language models during instruction tuning. In: Proceedings of the 40th International Conference on Machine Learning. ICML'23. JMLR.org (2023)
26. Wei, A., Haghtalab, N., Steinhardt, J.: Jailbroken: how does llm safety training fail? Adv. Neural Inf. Process. Syst. **36** (2024)
27. Yan, J., et al.: Backdooring instruction-tuned large language models with virtual prompt injection. In: Duh, K., Gomez, H., Bethard, S. (eds.) Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1: Long Papers, pp. 6065–6086. Association for Computational Linguistics, Mexico City (2024)
28. Yao, H., Lou, J., Qin, Z.: Poisonprompt: backdoor attack on prompt-based large language models. In: ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7745–7749 (2024)
29. Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S.: Privacy risk in machine learning: analyzing the connection to overfitting. In: 2018 IEEE 31st Computer Security Foundations Symposium (CSF), pp. 268–282. IEEE (2018)
30. Zou, A., Wang, Z., Kolter, J.Z., Fredrikson, M.: Universal and transferable adversarial attacks on aligned language models (2023)
31. Zou, W., Geng, R., Wang, B., Jia, J.: Poisonedrag: knowledge poisoning attacks to retrieval-augmented generation of large language models. arXiv preprint arXiv:2402.07867 (2024)