

Safeguarding Privacy of Retrieval Data against Membership Inference Attacks: Is This Query Too Close to Home?

Yujin Choi^{1†} Youngjoo Park^{1†} Junyoung Byun² Jaewook Lee¹ Jinseong Park^{3*}

¹ Seoul National University, Republic of Korea

² Chung-Ang University, Republic of Korea

³ Korea Institute for Advanced Study, Republic of Korea

{uznhigh, youngjoo0913, jaewook}@snu.ac.kr, junyoungb@cau.ac.kr, jinseong@kias.re.kr

Abstract

Retrieval-augmented generation (RAG) mitigates the hallucination problem in large language models (LLMs) and has proven effective for personalized usages. However, delivering private retrieved documents directly to LLMs introduces vulnerability to membership inference attacks (MIAs), which try to determine whether the target data point exists in the private external database or not. Based on the insight that MIA queries typically exhibit high similarity to only one target document, we introduce a novel similarity-based MIA detection framework designed for the RAG system. With the proposed method, we show that a simple detect-and-hide strategy can successfully obfuscate attackers, maintain data utility, and remain system-agnostic against MIA. We experimentally prove its detection and defense against various state-of-the-art MIA methods and its adaptability to existing RAG systems.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Grattafiori et al., 2024) have demonstrated strength for general and common knowledge. However, they struggle to answer domain-specific or personalized questions, resulting in hallucinations that fabricate non-truth answers from the training set (Huang et al., 2025). Retrieval-augmented generation (RAG) (Lewis et al., 2020) mitigates hallucination by giving information from external data retrieval into LLMs. RAG can provide reliable information by extracting relevant documents from an external database.

Membership inference attacks (MIAs) (Shokri et al., 2017) are malicious machine learning attacks that attempt to determine whether the target document was used in the training dataset. Recent work has examined MIAs and extraction attacks on language models to study the privacy leakage

(Shi et al., 2024). Even though MIAs achieve lower success rates on LLMs due to the massive size of training data samples (Puerto et al., 2024), privacy risks in RAG systems are significant (Zeng et al., 2024). As RAG retrieves a few top- k documents and conveys them directly to LLMs, various attack methods (Liu et al., 2025; Naseh et al., 2025) have succeeded in inferring whether a target document is stored in a private retrieval database or not.

However, safeguarding LLMs against MIA has no practical solution due to the huge model size or properties of language domains (Li et al., 2021). Within RAG systems, the agent-based query filtering system is the only defense explored so far, but it is also broken by stealth attacks that evade detection (Naseh et al., 2025). Thus, in this paper, we present a new safeguarding framework against MIAs by focusing on the observation that current MIAs on the RAG system rely on queries similar to only a single target document in the database.

In detail, we measure the similarity between an input query and the retrieved data points. Then, we check whether this similarity exceeds a threshold based on the Gumbel distribution (Gumbel, 1935) or not, which models the maximum values of data samples. For an input query that exceeds the threshold, indicating that the query is overly correlated with one specific document in the retrieval, we hide the data in the top- k document conveyed to the LLMs. We summarize our contributions as follows:

- We propose a similarity-based method for detecting MIA in RAG systems using Gumbel distribution, named **Mirabel**. To the best of our knowledge, this is the first attempt to study safeguarding strategies specifically against MIAs in RAG.
- Mirabel safeguards private external database against MIA through a simple detect-and-hide approach, which obfuscates attackers while preserving the utility of the RAG systems.

*Corresponding author. †Equal contribution.

- We empirically show the effectiveness of our method against various measures, and its comparability to existing defense methods.

2 Related Works

2.1 Retrieval-Augmented Generation

RAG is a strategy that enhances LLMs by integrating external data retrieval into the generation process (Lewis et al., 2020). At its core, an RAG system comprises three primary components: an external database $\mathcal{D}$ of textual documents, a retriever R , and a generator (i.e., LLM) G . When a user submits a query q , the retriever identifies the top- k contextual retrieval from $\mathcal{D}$ based on similarity, such as cosine similarity, often computed in an embedding model $\phi(\cdot)$ (Karpukhin et al., 2020):

$$R_k(q) = \arg \text{top}k_{d \in \mathcal{D}} \text{sim}(q, d). \quad (1)$$

These retrievals are then combined with the original query to form an augmented context, which is passed to the generator. The generator produces an output based on the retrieval (Gao et al., 2023; Shuster et al., 2021) as follows:

$$p(q) = \text{RAGprompt}(q, R_k(q)), \quad (2)$$

$$\text{response} = G(p(q)). \quad (3)$$

RAG improves response accuracy and reduces hallucinations frequently observed in pure LLMs. Furthermore, RAG offers architectural flexibility: any of the three core modules ($\mathcal{D}$, R , and G) can be replaced or updated independently without requiring end-to-end retraining (Cheng et al., 2023). Moreover, query rewriting (e.g., correcting ambiguous) and specialized retrieval (e.g., token- or graph-based) can be used in RAG (Ram et al., 2023).

2.2 Privacy Leakage of RAG System

Despite the utility improvements, the RAG systems inevitably pose privacy concerns when dealing with sensitive or proprietary data in the retrieval. For example, when dealing with sensitive data, the use of RAG increases the risk of serious legal complications and leaks personal privacy if the documents in the external database for retrieval are exposed or attacked. Thus, we focus on the privacy of the private external database in this paper.

The purpose of MIAs for the RAG system is to determine whether a target document is stored in the database. Beginning with directly asking

whether the document is in the database or not (Anderson et al., 2024), various attacks are proposed: S²MIA (Li et al., 2024) that provides the first half of the document and requests completion; MBA (Liu et al., 2025) that prompts prediction of masked tokens; and IA (Naseh et al., 2025), indicating an interrogation attack, that asks multiple queries that are hard to answer without the document. All of the attacks then exploit the output of the RAG system to infer the membership of the target document.

2.3 Safeguarding Attacks in RAG

To safeguard against MIAs in RAG systems, Naseh et al. (2025) investigated a simple filtering method that asks an LLM agent, such as GPT-4o, to classify incoming queries as benign or malicious. However, this agent-based method struggles with several challenges that will be further discussed in Section 3.1.

As a complementary safeguard system, differential privacy (DP) provides a mathematically rigorous privacy guarantee for sensitive data. Duan et al. (2023) introduced a DP framework that hides private information through noisy labeling against MIAs. DP LLMs have also been explored for in-context learning (Tang et al., 2024) and private prompt tuning (Hong et al., 2024). For the RAG system, DP-RAG (Grislain, 2024) protects the privacy of data in the data retrieval, and DPVoteRAG (Koga et al., 2024) uses private voting. Tran et al. (2025) used a machine unlearning to delete highly memorized tokens, but their approach targets fine-tuning LLMs and is not focused on RAG privacy.

3 Scenarios

Following the previous studies (Liu et al., 2025; Naseh et al., 2025), we assume three parties in our scenario: 1) the **operator** of the RAG system with **private external database**, 2) the **benign users** of the RAG, and 3) the malicious **attackers** attempting MIAs to the RAG system. For instance,

“A healthcare AI operator deploys a medical-diagnosis chatbot that retrieves private patient records. Benign users consult the chatbot to assess their health, but attackers attempt MIAs. The operator must preserve privacy against attackers while providing accurate answers to benign users.”

Types of Queries Based on the scenario, we divide the queries for RAG systems into three types: member attack query q_a^m by attackers of malicious questions in the external database, non-member attack query q_a^n by attackers of malicious questions

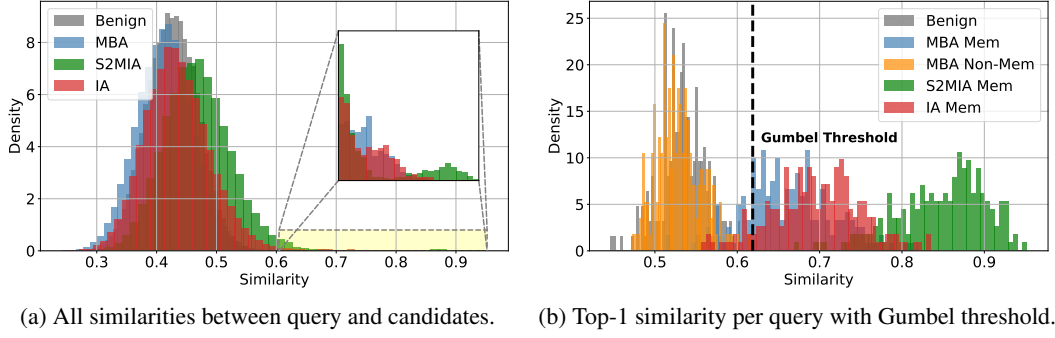


Figure 1: Distributions of similarity scores between queries and retrieved data. We visualize both the full similarity distributions and the top-1 similarities. A Gumbel-based threshold $\bigcup_q S_q$ is marked for reference.

but not in the external database, and benign query q_b of benign users without malicious intention.

Details about Attacker Given a target document d , an attacker aims to decide whether d is contained in the external database $\mathcal{D}$ by classifying its associated query q_a as a member attack query q_a^m or a non-member attack query q_a^n . We assume the attacker cannot access $\mathcal{D}$ or the parameters of LLMs.

Details about Operator’s Defenses The defense should prevent attackers from recognizing the membership information while delivering correct responses to benign users without prior knowledge of attacks, such as their prompts or patterns. The defender also cannot access LLM parameters, just relying on access to the external database $\mathcal{D}$ and its embedding model ϕ . Each query must be answered immediately without pending the next queries.

3.1 Motivation and Goals

To safeguard against MIAs, it is essential to determine whether an input query is malicious to the RAG system for the private external database. Recently, Naseh et al. (2025) proposed an LLM agent-based detection method to evaluate the malicious intent of input queries, independent of the external database. To judge either benign queries q_b or attack queries q_a , the agent identifies harmful or extraction phrases within queries. However, Naseh et al. (2025) also showed that the agent-based detection can be deceived by stealth queries, which are crafted attack queries designed to mimic benign queries. Moreover, simply rejecting all suspected attack queries may inadvertently reveal to attackers filtering phrases and detour the detection systems.

These limitations suggest that an effective defense requires not simply blocking queries but obfuscating the attacker’s knowledge. An attack succeeds only if it can correctly separate q_a^m from

q_a^n using the RAG’s responses. Hence, to defend against the attack, the responses should be made statistically indistinguishable. Achieving this in turn demands that the defender distinguishes between q_a^m and q_a^n , so the responses can be selectively perturbed. Therefore, we set two complementary goals: 1) accurately separate member attack queries from non-member attack queries, and 2) respond with the corresponding responses so that an attacker cannot classify the member and non-member cases.

4 Proposed Method

4.1 Is This Query Too Close to Home? MIA Detection with Gumbel Distribution

By analyzing the similarity of queries, we propose a detection method for the RAG system that distinguishes member attack queries (q_a^m) from other types of queries (q_b and q_a^n). Specifically, our goal is to find a threshold τ to classify a query as a member attack q_a^m if the maximum similarity exceeds:

$$\max_{d \in \mathcal{D}} \text{sim}(q, d) > \tau. \quad (4)$$

To find the threshold τ , we focus on the distributional differences between member attack queries and other queries (Wen et al., 2024).

Figure 1a demonstrates $\bigcup_q S_q$, the cosine similarities between all queries and the external database, where $S_q = \{\text{sim}(q, d) | d \in \mathcal{D}\}$ is a set of similarities between a query q and the total database. The figure shows (i) similarities of the benign query q_b and the total database follow a (approximately) normal distribution, and (ii) for the member attack queries q_a^m , the distribution appears similar to q_b in the most likely region, but exhibits extreme values in the right tail, which are too similar with a private external database.

Based on these observations, we propose a detection method using the Gumbel distribution, which

represents the maximum value of data samples (Gumbel, 1935). In previous works on LLM privacy, the Gumbel distribution is widely used to find top- k selection (Durfee and Rogers, 2019; Hong et al., 2024). By extreme value theory, if n random variables are sufficiently large and follow an i.i.d. normal distribution, i.e.,

$$X_1, \dots, X_n \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(\mu_q, \sigma_q^2), \quad (5)$$

for some μ_q and σ_q , then the maximum of the samples converges in distribution to a Gumbel distribution:

$$\max\{X_1, \dots, X_n\} \xrightarrow{d} \text{Gumbel}(\mu_n, \beta_n), \quad (6)$$

where $\beta_n = \sigma/\sqrt{2 \ln n}$ and

$$\begin{aligned} \mu_n &= \mu_q + \sigma_q d \sqrt{2 \ln n - \ln(\ln n) - \ln(4\pi)} \\ &\approx \mu_q + \sigma_q d \sqrt{2 \ln n} \quad (\text{as } n \rightarrow \infty). \end{aligned} \quad (7)$$

Under the observations (i) and (ii) in Figure 1a, we assume that the samples from $S_q \setminus \{s_{\max}\}$ follow a normal distribution for each q , where $s_{\max}$ is the maximum similarity in S_q . Let μ_q and σ_q be the mean and standard deviation of $S_q \setminus \{s_{\max}\}$. Then, the threshold τ can be calculated as follows:

$$\tau = \mu_n + c \cdot \sigma_q / \sqrt{2 \ln n}, \quad (8)$$

where $c = -\ln(-\ln(1 - \rho))$ is a critical value of the Gumbel distribution for significance level ρ .

Figure 1b shows histograms of the maximum value of each S_q , i.e., $s_{\max}$ for each query. The dashed line represents the threshold based on a Gumbel distribution, which was computed from $\bigcup_{q_b} S_{q_b}$. Thresholds computed for each attack type for separate $s_{\max}$ for non-member attack query q_a^n and member-attack query q_a^m are provided in Figure 6 in Appendix C. Most $s_{\max}$ values of the member attack queries exceed the Gumbel-based threshold, whereas those of benign and non-member attack queries remain below it. These findings suggest that $s_{\max}$ of member attack queries q_a^m can be successfully separated from q_b and q_a^n by the Gumbel-based threshold.

To this end, we name the proposed detection method as **Mirabel**, i.e., **MIA** in **RAG** systems using **Gumbel**. Mirabel not only detects whether the query q is a member attack, but also identifies the specific document that is likely the target. Moreover, it can be incorporated into the standard RAG system naturally, as it uses the similarity scores already computed for top- k selection. The detailed method is shown in Algorithm 1.

Test set	Benign	MBA	S ² MIA	IA
S_q	0.469	0.027*	0.001*	0.012*
$S_q \setminus \{s_{\max}\}$	0.400	0.511	0.293	0.226

Table 1: Average p -values of the normality test on the total data set (S_q) and on the set with the maximum similarity removed ($S_q \setminus \{s_{\max}\}$). *: p -value < 0.05.

Algorithm 1 Proposed Mirabel Detection

Input: Query embedding function ϕ , corpus embeddings $\{e_d = \phi(d) : d \in \mathcal{D}\}$, query q , significance level ρ

Output: detection result, target document d_t

Initialization: $S_q \leftarrow \{\}$, $e_q \leftarrow \phi(q)$

- 1: **for all** $d \in \mathcal{D}$ **do**
 - 2: $s_d \leftarrow \cos(e_q, e_d)$ ▷ Compute sim(q, d)
 - 3: $S_q \leftarrow S_q \cup \{s_d\}$
 - 4: **end for**
 - 5: $s_{\max}, d_t \leftarrow \max(S_q)$ ▷ Find (arg)max sim
 - 6: $\mu_q, \sigma_q \leftarrow \text{mean}(S_q \setminus \{s_{\max}\}), \text{std}(S_q \setminus \{s_{\max}\})$
 - 7: $\mu_n \leftarrow \mu_q + \sigma_q \cdot \sqrt{2 \ln n}$ ▷ Gumbel mean
 - 8: $c \leftarrow -\ln(-\ln(1 - \rho))$ ▷ Critical value
 - 9: $\tau \leftarrow \mu_n + c \cdot \sigma_q / \sqrt{2 \ln n}$ ▷ Threshold
 - 10: **return** ($s_{\max} > \tau$), $\mathbb{1}[s_{\max} > \tau] \cdot d_t$
-

Normality test To validate our assumption, we conduct a normality test on S_q and $S_q \setminus \{s_{\max}\}$. Table 1 demonstrates the mean p -values from D’Agostino and Pearson’s normality test, averaged across all queries q . Across all MIAs, the p -values of total similarity scores in S_q are less than 0.05, indicating that we can reject the hypothesis that the scores follow a normal distribution. In contrast, for a benign query q_b , $p > 0.05$, we cannot reject that S_{q_b} follows a normal distribution.

In contrast, when the maximum similarity score $s_{\max}$ is removed from S_q , we observe $p > 0.05$ for all queries, indicating that we cannot reject that $S_q \setminus \{s_{\max}\}$ follows a normal distribution. As removing $s_{\max}$ indicates the target document is removed in the database, we cannot reject the hypothesis of normality even for non-member attack queries q_a^n .

The distributional difference between S_q and $S_q \setminus \{s_{\max}\}$ indicates that $s_{\max}$ is an extreme outlier relative to values drawn from a normal distribution for q_a^m . Since $s_{\max}$ is distinctive in the Gumbel distribution, this supports the proposed Mirabel method, which identifies MIAs for the RAG system by leveraging the confidence interval of the Gumbel distribution.

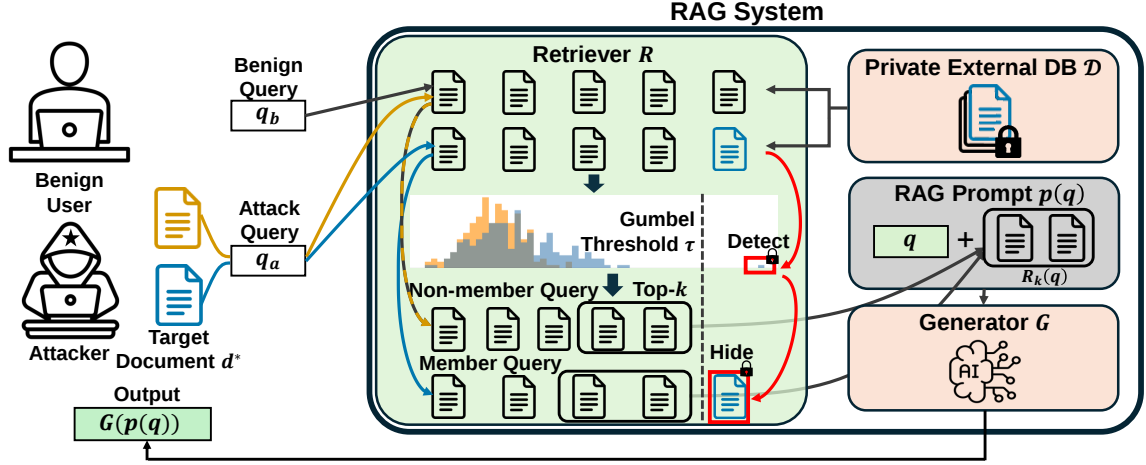


Figure 2: Illustration of our proposed Mirabel. We perform our detection to classify whether an input query is a member attack query q_a^m . If it is detected as q_a^m , we hide it from data retrieval and proceed with the standard RAG system.

4.2 Detect-and-hide to Defend MIA

Leveraging the proposed Mirabel for MIA detection, we propose a simple defense method that safeguards the data sample in the external database with a simple *detect-and-hide* strategy. In the detection, the query is evaluated using Mirabel to determine whether it is a member attack query q_a^m . In the hiding, if the query q is classified as q_a^m , the target document identified by Mirabel is hidden from the external database, and the standard RAG system proceeds with the remaining documents. If the query is not classified as a member attack, the standard RAG system proceeds without any modification. Figure 2 illustrates our proposed defense method.

First of all, this simple defense confuses the attacker about whether the RAG output is from members or non-members, since the response of the retriever no longer includes the target document in both cases. Upon the observation of the normality in the previous subsection, the response distribution remains indistinguishable between member and non-member queries.

Our defense method, furthermore, preserves utility for benign queries q_b . When the query is not detected as q_a^m , the system behaves exactly the same as the original RAG system. Unlike other strong privacy-preserving mechanisms (e.g., differential privacy), which may degrade utility even for benign queries, our approach introduces minimal utility loss when the query is not detected as q_a^m .

Finally, our safeguarding strategy is agnostic to any RAG systems, as the detection only requires similarity scores, which is a basic part of the RAG system to calculate the top- k documents in the ex-

ternal database. Thus, it is easy to apply and does not require any modification of the system. The only additional time is to compute the Gumbel distribution and hide the corresponding document from the database for retrieval if the query is classified as q_a^m . Based on this advantage, we show that integrating our method into existing privacy-preserving systems results in better defenses.

5 Experiments

5.1 Experimental Setups

Dataset To evaluate detection and defense performance against MIAs, we utilize three datasets: NFCorpus, SciDOCS, and TREC-COVID, drawn from the BEIR benchmark (Thakur et al., 2021), as employed in (Naseh et al., 2025). These corpora represent sensitive scientific and medical domains. In addition to evaluating the utility of the RAG system on benign queries, we followed (Koga et al., 2024) and used two question-answer benchmark datasets: Natural Questions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017). For both datasets, Wikipedia serves as the external database (Lewis et al., 2020).

Baselines We evaluate performance for our detection and defense method using three MIA methods: S²MIA (Li et al., 2024), MBA (Liu et al., 2025), and IA (Naseh et al., 2025). S²MIA feeds the first half of the target document to the RAG and scores membership with BLEU and perplexity against the full text. MBA masks tokens and counts how many the generator recovers. IA lets an LLM craft 30 inference questions per document, and correctness on them is the score.

For the comparison method of detection, we perform an agent-based detection using GPT-4o to distinguish between following the approach of (Naseh et al., 2025), detailed in Appendix B.

As our work is the first work for defending MIAs for RAGs, we choose to compare with the most well-known privacy notion, DP (Dwork et al., 2006), which guarantees indistinguishability of individual data in the private external database. For DP-RAG (Grislain, 2024), we inject the noise in the next token prediction based on DP in-context learning (Tang et al., 2024) without considering the privacy of top- k selection (Koga et al., 2024).

For the privacy budget ϵ , we set $\epsilon = 2$ for base DP-RAG. To compare with a relaxed privacy setting, we also set $\epsilon = 100$ and referred to it as DP-RAG Large (in short, DP-RAG-L).

Metrics For the detection component, we measure accuracy, precision, recall, and F1-score to determine how effectively the system identifies queries targeting member documents.

To assess the utility of the system, we measure (i) the exact match (EM), which indicates the proportion of generated responses containing the gold labels with top- k retrievals, and (ii) R@ k , which indicates the proportion of retrievals containing the gold labels among the top- k results.

For defense, we suggest the measures of **attack resistance** and **indistinguishability**. The attack resistance is measured based on the accuracy of the attack. However, in MIA, since it is a binary classification, accuracy alone may underestimate attack performance, especially when it falls below 0.5. We instead define adjusted attack accuracy as:

$$\text{Adjusted attack accuracy} = \max(\text{acc}, 1 - \text{acc}) - 0.5. \quad (9)$$

This metric reflects the adversary’s advantage over random guessing (0.5 in a balanced setting) and ranges from 0 to 0.5. Lower values indicate weaker attack success and, consequently, stronger defense.

To evaluate indistinguishability, we use the Kolmogorov–Smirnov (KS) test (An, 1933) to compare the distributions of responses for member and non-member queries. The KS statistic measures the maximum difference between the two empirical distributions; a smaller value indicates greater similarity, and thus higher indistinguishability.

RAG Setting and Implementation We employed a 1024-dimensional BGE_{m3} embedder (Li et al., 2023) to find the similarity of queries and doc-

uments in the retriever. In our defensible retriever, we use a Mirabel based on the significance parameter ρ , to decide whether to filter documents before extracting top- k candidates via cosine similarity. In our experiment, we set the significance level $\rho = 0.05$. In evaluations concerning MIA detection and defense on NFCorpus, SCIDOCS, and TREC-COVID, we set $k = 3$. For utility evaluations on the Wikipedia corpus, we tested $k \in \{5, 20, 100\}$.

For the generator LLMs, we adopted Llama 3.2 3B Instruct and Llama 3.1 8B Instruct (Grattafiori et al., 2024). Specifically, we used Llama 3.1 for the utility evaluations and Llama 3.2 for the remaining evaluations. Further details on the RAG prompt can be found in the Figure 4. Experiments were mainly conducted on a single A100 or H100 GPU with 96 GB VRAM.

Additional details are shown in Appendix A.

5.2 Detection Evaluation

Table 2 presents the member attack query detection performance compared to agent-based methods. To detect member attack queries, q_a^m was labeled as 1 and otherwise 0. Note that the agent-based detection (Naseh et al., 2025) labels both attack queries q_a^m and q_a^n as 1, which cannot distinguish attack queries only. To balance the binary classification, we set the total number of q_b and q_a^n to be the same as q_a^m .

Mirabel detection demonstrates stable performance across all MIAs, including the IA with stealth queries, which the agent-based detection struggles to detect. Notably, our method achieved a high recall score, indicating successful detection of q_a^m , while its precision was slightly lower due to misclassifying q_b as q_a^m . In other words, our method exhibits a lower Type II error, which is generally considered more critical in attack detection. However, despite this minor reduction, our precision remains higher than that of agent-based detection.

The agent-based methods cannot classify a query as related to private members or not. In the next section, we will analyze how these errors impact both attack performance and utility. Additional metrics, including detection performance comparing only q_b and q_a^m , are provided in Appendix C. The results show that our method achieves a high recall score.

5.3 Defense Evaluation

Safeguarding against MIA has two main goals: (i) reducing attack performance and (ii) preserving the utility of RAG systems, while preventing the

Attacks	Data	Agent-based Detection				Mirabel Detection			
		Acc ($\uparrow$)	F1 ($\uparrow$)	Precision ($\uparrow$)	Recall ($\uparrow$)	Acc ($\uparrow$)	F1 ($\uparrow$)	Precision ($\uparrow$)	Recall ($\uparrow$)
S ² MIA	NF	0.715	0.772	0.644	0.962	0.844	0.865	0.762	1.000
	SCI	0.726	0.780	0.651	0.974	0.853	0.871	0.776	0.992
	TREC	0.745	0.792	0.669	0.970	0.880	0.886	0.846	0.930
MBA	NF	0.727	0.783	0.650	0.986	0.847	0.867	0.766	1.000
	SCI	0.730	0.785	0.652	0.986	0.855	0.873	0.775	1.000
	TREC	0.745	0.794	0.667	0.980	0.995	0.995	1.000	0.990
IA	NF	0.482	0.012	0.125	0.006	0.799	0.750	0.896	0.817
	SCI	0.485	0.030	0.258	0.016	0.827	0.772	0.928	0.843
	TREC	0.500	0.000	0.000	0.000	0.720	0.744	0.670	0.705

Table 2: Detection performance of Mirabel compared to agent-based detection. For simplicity, we denote NFCorpus as NF, SCIDocs as SCI, and TREC-COVID as TREC throughout the following tables.

Data	System	EM ($\uparrow$)			R ($\uparrow$)		
		@5	@20	@100	@5	@20	@100
NQ	RAG	0.272	0.263	0.313	0.222	0.354	0.474
	DP-RAG	0.030	0.051	0.020	—	—	—
	Ours	0.253	0.253	0.273	0.172	0.303	0.443
TRIV	RAG	0.730	0.725	0.755	0.575	0.705	0.840
	DP-RAG	0.255	0.225	0.240	—	—	—
	Ours	0.700	0.725	0.740	0.515	0.675	0.825

Table 3: Utility measures computed with benign queries.

attacker from gaining private information. Thus, we must measure both attack degradation and utility preservation for defense methods.

Utility Preservation We first demonstrate that our method largely avoids the utility degradation associated with the existing privacy-preserving approach for RAG systems, DP-RAG, in Table 3.

EM@k, which measures the quality of the generated answers, shows that our method performs comparably to the standard RAG, while DP-RAG results in greater utility degradation.

In contrast, R@k, which measures how well the system retrieves relevant documents, shows a moderate decrease compared to the original RAG system. This is because our detection method has a slightly larger Type I detection error.

Despite degradation in R@k, our method maintains high performance in EM, because it retains the remaining top- k documents (i.e., from top-2 to top- $(k + 1)$) by hiding only the target member document. Since benign queries do not rely on a single document but instead reference multiple relevant documents, we can mitigate the utility loss.

Attack Resistance To evaluate the performance of our defense, we measure the attack performances of MIAs in Table 4. Without defense, MIAs

achieved high adjusted attack accuracy, indicating the attacks are successful. However, after applying the defenses, the accuracy decreases, suggesting that the attacker struggles to distinguish whether the response is from a member or a non-member. Our method effectively reduces the attack accuracy and achieves performance comparable to DP-RAG (-L), which is considered a strong defense despite its inherent privacy-utility trade-off.

From the perspective of detection failure, attacker performance is related to the Type II error. Type II errors occur when the detection method fails to identify a member query, preventing the removal of the target document. This can increase the number of true positives, as the MIA will classify members correctly as members.

Since our detection method has a low Type II error rate, it results in a strong defense. However, in the case of the IA on the TREC-COVID dataset, even with a low recall and high Type II error, the defense performance remains strong.

This is because the Type II error occurs when the similarity score between the target document d and the attack query q_a^m is not significantly high. In IA, such low similarity indicates that the attack query was either a general question related to multiple documents or not strongly associated with the target document. In these cases, the attack itself is also likely to fail.

Indistinguishability Most MIAs rely on measuring a score and making membership decisions based on that score. Even if a defense method successfully reduces the attacker’s accuracy, differences in the score distributions between member and non-member queries enable attackers to perform adaptive attacks. Therefore, another goal of

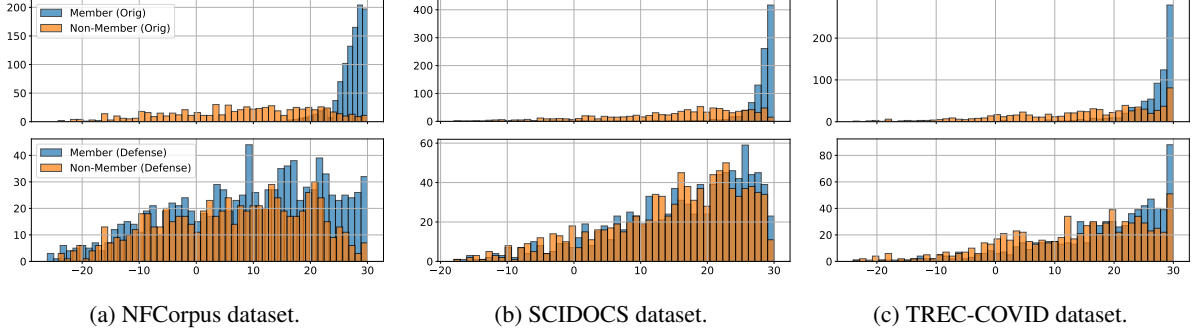


Figure 3: Histograms of IA scores for member and non-member queries across different datasets. After defense, the member and non-member score distributions become indistinguishable.

		NF	SCI	TREC	Avg
S^2 MIA	RAG	0.188	0.126	0.119	0.144
	DP-RAG	0.021	0.015	0.004	0.013
	DP-RAG-L	0.034	0.000	0.004	0.013
	Ours	0.024	0.006	0.021	0.017
MBA	RAG	0.377	0.406	0.313	0.365
	DP-RAG	0.004	0.040	0.003	0.016
	DP-RAG-L	0.014	0.036	0.184	0.078
	Ours	0.019	0.101	0.139	0.086
IA	RAG	0.403	0.380	0.254	0.346
	DP-RAG	0.038	0.288	0.174	0.167
	DP-RAG-L	0.267	0.305	0.020	0.197
	Ours	0.008	0.010	0.083	0.034

Table 4: Adjusted attack accuracy ($\downarrow$). A smaller value indicates weaker attack success, thus stronger defense.

		NF	SCI	TREC	Avg
S^2 MIA	RAG	0.358	0.195	0.203	0.252
	DP-RAG	0.077	0.051	0.038	0.055
	DP-RAG-L	0.060	0.035	0.058	0.051
	Ours	0.057	0.039	0.043	0.046
MBA	RAG	0.754	0.813	0.625	0.731
	DP-RAG	0.022	0.081	0.015	0.039
	DP-RAG-L	0.036	0.073	0.026	0.045
	Ours	0.038	0.202	0.283	0.174
IA	RAG	0.805	0.771	0.555	0.710
	DP-RAG	0.088	0.418	0.091	0.199
	DP-RAG-L	0.301	0.414	0.223	0.313
	Ours	0.100	0.081	0.172	0.118

Table 5: KS statistics ($\downarrow$) for S^2 MIA, MBA, and IA scores. For S^2 MIA not having a direct score, we report the average KS statistics of similarity and perplexity.

defense is to achieve indistinguishability between responses to member and non-member queries, preventing the attacker from gaining additional membership information.

Table 5 presents the results measuring indistinguishability. As shown in the table, our method significantly reduces the KS statistic, achieving

		Adjusted Accuracy ($\downarrow$)			KS statistics ($\downarrow$)		
		NF	SCI	TREC	NF	SCI	TREC
S^2 MIA	DP	0.021	0.004	0.038	0.077	0.051	0.038
	+Ours	0.020	0.029	0.024	0.088	0.028	0.050
MBA	DP	0.015	0.040	0.288	0.022	0.081	0.015
	+Ours	0.010	0.015	0.004	0.058	0.040	0.035
IA	DP	0.004	0.003	0.174	0.088	0.418	0.091
	+Ours	0.004	0.018	0.064	0.046	0.075	0.094

Table 6: Defense performances of DP-RAG and DP-RAG with (+) our detect-and-hide using Mirabel.

	DP-RAG			DP-RAG + Ours		
	@5	@20	@100	@5	@20	@100
NQ	0.030	0.051	0.020	0.020	0.040	0.050
TRIV	0.255	0.225	0.240	0.230	0.230	0.225

Table 7: EM ($\uparrow$) of DP-RAG and DP-RAG (+) ours.

comparative or even smaller values compared to DP-RAG. As lower KS statistics indicate higher distributional similarity, our method prevents the attacker from obtaining additional information gain.

To further illustrate this, we present histograms of the attack scores produced by IA before and after applying our defense in Figure 3. In the original RAG, we can observe the clear difference between member and non-member queries. However, after applying our defense, the two distributions are almost similar. Additional results, including the individual KS statistics for similarity and perplexity under S^2 MIA, as well as histograms of attack scores for other attack methods, are provided in Appendix C.

5.4 Composing with Existing DP Models

In this section, we adapt our detect-and-hide strategy to DP-RAG to demonstrate that our method is agnostic to the RAG system. Table 6 presents the defense performance when applying our method

		(LLM) Phi-4 Mini				(Embedder) GTE Large En v1.5			
		NF	SCI	TREC	Avg	NF	SCI	TREC	Avg
S ² MIA	RAG	0.110	0.168	0.092	0.123	0.116	0.039	0.084	0.080
	Ours	0.024	0.007	0.038	0.023	0.006	0.011	0.078	0.032
MBA	RAG	0.250	0.255	0.159	0.221	0.399	0.396	0.297	0.364
	Ours	0.001	0.064	0.057	0.041	0.080	0.101	0.290	0.157
IA	RAG	0.322	0.430	0.248	0.334	0.347	0.416	0.222	0.328
	Ours	0.039	0.019	0.104	0.054	0.011	0.008	0.205	0.074

Table 8: Adjusted attack accuracy ($\downarrow$). A smaller value indicates weaker attack success, thus stronger defense.

Data	RAG (ms)	+Ours (ms)	Diff. (ms)	Ratio
NF	49.215	50.895	1.680	1.034x
SCI	52.355	60.510	8.155	1.156x
TREC	100.750	148.695	47.945	1.476x

Table 9: Runtime overhead of detect-and-hide (Ours) defense compared to the original RAG system.

on top of base DP-RAG ($\epsilon = 2$). As in the standard RAG setting, our method improved attack resistance and indistinguishability, even though DP already achieves strong defense performance. In terms of utility, Table 7 shows that our method introduces only minimal utility degradation. This indicates that composing our method with existing or upcoming privacy-preserving models can be beneficial. Additional results are in Appendix C.

5.5 Experiments with Different Models

To validate the robustness of our detect-and-hide method, we further evaluate its performance by replacing either the generator LLM or the embedder in the RAG system. For the generator LLM model, we utilize Phi-4 Mini Instruct (3.8B) (Abouelenin et al., 2025), a recently released open-source LLM model. For the embedding model to compute query-document similarity, we employed GTE Large En v1.5 (Zhang et al., 2024), which supports long context inputs (up to 8192 tokens) and is well-suited for document-level retrieval. The experimental results are represented in Table 8.

As shown in the table, our defense method consistently reduces the attack performance across all datasets and attack types, even under different LLM and embedding models, demonstrating the robustness of the detect-and-hide strategy. Nevertheless, with an alternative embedder, the defense performance on the TREC-COVID dataset was relatively weaker than on the others, a limitation we further discuss in Limitations section.

5.6 Runtime

Detect-and-hide reuses similarity scores that are already computed within the RAG system, and in theory, it requires only marginal additional computation for searching. To empirically verify this, we directly measure the runtime. The results are presented in Table 9.

As expected, the overhead on smaller datasets such as NF-Corpus remains marginal. However, for larger datasets such as TRECCOVID, we observe a latency increase of up to 47.6%. This is due to the FAISS library (Douze et al., 2024), which we utilized for RAG. The current version of FAISS does not allow direct access to similarity scores without top- k sorting, leading to unnecessary computation on large datasets.

If future versions of FAISS (or alternative libraries) enable direct access to unsorted similarity scores, our detection method can achieve the theoretically minimal overhead in practice.

A detailed theoretical analysis based on the complexity explanation of runtime and FAISS is provided in the Appendix C.

6 Conclusion

In this paper, we propose Mirabel, a MIA detection method that classifies member attack queries based on similarity using the Gumbel distribution. We also introduce a simple defense strategy, detect-and-hide, which effectively defends against attacks with minimal utility degradation. Our method is model-agnostic, incurs almost no additional computation overhead, and can be easily applied to existing RAG systems. Experimental results show that our method achieves defense performance comparable to DP-RAG, while incurring negligible utility loss on benign queries.

Limitations

Even though we have designed efficient detection and defense methods for existing MIAs targeting the private external database of the RAG system, we need to be cautious when applying this defense framework in real-world applications.

Our method relies on the assumption that the embedding of the member attack query is highly similar to that of the target document. Therefore, the detection or defense performance depends on the use of the embedder. If the embedder assigns high similarity when inputs share the same word (e.g., “COVID”), Mirabel struggles to detect member attack queries based on the similarity. This can be observed in Table 8, where the defense performance on the TREC-COVID dataset was limited when using a different embedder. If the defense fails due to this limitation, relying solely on our method could leave sensitive information vulnerable to membership inference attacks.

To address this, we need to carefully adjust the threshold selection strategy and develop additional defense mechanisms that remain robust across embedders.

Ethical Considerations

This work adheres to the ACL Code of Ethics. In RAG systems, MIAs pose a serious threat to the privacy of user queries submitted to LLMs. Mitigating these attacks, therefore, makes a meaningful ethical contribution and offers practical guidance for developing RAG systems that remain robust against malicious attackers.

Acknowledgments

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. RS-2023-00272502, No. RS-2024-00338859, No. RS 2024-00335811), and the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-II220984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation). Jinseong Park is supported by a KIAS Individual Grant (AP102301) via the Center for AI and Natural Sciences at Korea Institute for Advanced Study. This work was supported by the Center for Advanced Computation at Korea Institute for Advanced Study.

References

- Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, and 1 others. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*.
- Kolmogorov An. 1933. Sulla determinazione empirica di una legge didistribuzione. *Giorn Dell’inst Ital Degli Att*, 4:89–91.
- Maya Anderson, Guy Amit, and Abigail Goldsteen. 2024. Is my data in your retrieval database? membership inference attacks against retrieval augmented generation. *arXiv preprint arXiv:2405.20446*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. 2023. Lift yourself up: Retrieval-augmented text generation with self-memory. *Advances in Neural Information Processing Systems*, 36:43780–43799.
- Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvassy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. [The faiss library](#).
- Haonan Duan, Adam Dziedziec, Nicolas Papernot, and Franziska Boenisch. 2023. Flocks of stochastic parrots: Differentially private prompt learning for large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- David Durfee and Ryan M Rogers. 2019. Practical differentially private top-k selection with pay-what-you-get composition. *Advances in Neural Information Processing Systems*, 32.
- Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2:1.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Nicolas Grislain. 2024. Rag with differential privacy. *arXiv preprint arXiv:2412.19291*.

- Emil Julius Gumbel. 1935. Les valeurs extrêmes des distributions statistiques. In *Annales de l'institut Henri Poincaré*, volume 5, pages 115–158.
- Junyuan Hong, Jiachen T. Wang, Chenhui Zhang, Zhangheng LI, Bo Li, and Zhangyang Wang. 2024. DP-OPT: Make large language model your privacy-preserving prompt engineer. In *The Twelfth International Conference on Learning Representations*.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.
- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pages 1601–1611. Association for Computational Linguistics (ACL).
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.
- Tatsuki Koga, Ruihan Wu, and Kamalika Chaudhuri. 2024. Privacy-preserving retrieval augmented generation with differential privacy. *arXiv preprint arXiv:2412.04697*.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.
- Jiacheng Li, Ninghui Li, and Bruno Ribeiro. 2021. Membership inference attacks and defenses in classification models. In *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*, pages 5–16.
- Yuying Li, Gaoyang Liu, Chen Wang, and Yang Yang. 2024. Generating is believing: Membership inference attacks against retrieval-augmented generation.
- Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.
- Mingrui Liu, Sixiao Zhang, and Cheng Long. 2025. Mask-based membership inference attacks for retrieval-augmented generation. In *Proceedings of the ACM on Web Conference 2025*, pages 2894–2907.
- Ali Naseh, Yuefeng Peng, Anshuman Suri, Harsh Chaudhari, Alina Oprea, and Amir Houmansadr. 2025. Riddle me this! stealthy membership inference for retrieval-augmented generation. In *ACM SIGSAC Conference on Computer and Communications Security*.
- Haritz Puerto, Martin Gubri, Sangdoo Yun, and Seong Joon Oh. 2024. [Scaling up membership inference: When and how attacks succeed on large language models](#). *Preprint*, arXiv:2411.00154.
- Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.
- Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*.
- 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.
- Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.
- Xinyu Tang, Richard Shin, Huseyin A Inan, Andre Manoel, Fatemehsadat Miresheghallah, Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, and Robert Sim. 2024. Privacy-preserving in-context learning with differentially private few-shot generation. In *The Twelfth International Conference on Learning Representations*.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.
- Toan Tran, Ruixuan Liu, and Li Xiong. 2025. Tokens for learning, tokens for unlearning: Mitigating membership inference attacks in large language models via dual-purpose training. *arXiv preprint arXiv:2502.19726*.

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

Shenglai Zeng, Jiankun Zhang, Pengfei He, Yiding Liu, Yue Xing, Han Xu, Jie Ren, Yi Chang, Shuaiqiang Wang, Dawei Yin, and 1 others. 2024. The good and the bad: Exploring privacy issues in retrieval-augmented generation (rag). In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4505–4524.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, and 1 others. 2024. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. *arXiv preprint arXiv:2407.19669*.

A Experimental Details

A.1 Dataset

For attack and defense, we used NFCorpus, SCIDOCS, and TREC-COVID, containing approximately 3.6K, 171K, and 25K documents, respectively. For the retriever database, NFCorpus is split into member and non-member sets in a 7:3 ratio, while SCIDOCS and TREC-COVID are divided 8:2. To conduct MIA, we sample 1,000 members and 1,000 non-members from NFCorpus, and 2,000 members and 2,000 non-members from the other two datasets. Each member and non-member subset is further split (5:5) into reference and evaluation sets. Detection performance from benign queries to attack queries is evaluated using 323, 500, and 50 queries provided in NFCorpus, SCIDOCS, and TREC-COVID, respectively, while an equal number of MIA attack queries is generated for the member subsets. This balanced design ensures a fair assessment of the detection and defense capabilities.

For utility evaluation on benign queries, NQ and TriviaQA are used, comprising approximately 7.8K and 138K question-answer (QA) pairs, respectively. In our experiments, we select 200 QA pairs from each dataset for evaluation.

All datasets and models used in this study are publicly available and were released for academic or research purposes.

A.2 Hyperparameters

Attacks For all datasets, we divided the member and non-member datasets into 5:5 for reference and evaluation. For MBA and S²MIA, reference sets were used to establish thresholds, following their original paper. We found a threshold in the greedy search algorithm for S²MIA and the highest F1 score for each count of masking tokens in {5, 10, 15, 20} in MBA.

For IA, we evaluate 30 queries to compute the IA score for the standard RAG system, and 20 queries for DP-RAG, since each query consumes a portion of the total privacy budget. Specifically, with a total $\epsilon = 2$, each query uses $\epsilon = 0.1$, while when $\epsilon = 100$, each query uses $\epsilon = 5$.

In the selection of thresholds for IA, while the original paper selects the threshold based on a false positive rate, we instead select the threshold based on accuracy. We searched threshold in $[0, 1]$, and we evaluated penalty values $\lambda \in [0.5, 1]$. In the case of DP-RAG, since the number of queries is

smaller, we differ the searching space. Since we observed that larger values of λ in this range resulted in higher recall but significantly lower accuracy, indicating potential over-penalization. We therefore refined the search to a smaller range of $\lambda \in [0.1, 0.5]$ for this setting.

RAG Setting Maximum sequence length was set to 8192 tokens for embedding documents using BGE_{m3}. For experiments on TREC-COVID corpus, the maximum sequence length was set to 2048 tokens involving DP generator. To facilitate efficient indexing and retrieval, we employed FAISS (Douze et al., 2024). We segment the Wikipedia corpus into non-overlapping chunks of 100 tokens and randomly sample one-tenth of these segments to serve as the retriever database. In building the Wikipedia retriever database, we utilized half-precision (fp16) to reduce computational overhead. This approach helps to assess the usability of RAG systems in a controlled yet representative manner while maintaining computational ease.

DP-RAG setting For DP-RAG, we follow the setting provided in the official GitHub of (Grislain, 2024) (<https://github.com/sarus-tech/dp-rag>). Specifically, we set the parameters as $\omega = 0.01$, $\alpha = 1.0$, and temperature = 1.0.

A.3 MIA implementation

To implement MIAs for evaluating Mirabel’s defense, we followed each original papers. For S²MIA, we calculated BLEU score and perplexity. To gain perplexity, GPT-2 was used. For MBA, we tokenize the sentences and selected mask tokens in random. (Liu et al., 2025) showed random masking has similar accuracy in attempting attacks. For IA, GPT-4o and GPT-4o-mini was used to generate summaries, questions, and ground truths.

B Prompt Templates

Figure 4 shows the generation prompt given to every RAG model (Llama-3.1 8B for utility experiments and Llama-3.2 3B elsewhere). Its design enforces three constraints that proved crucial for reliable evaluation: groundedness, conciseness, and failure transparency. Figure 5 gives the agent-based detection prompt used with GPT-4o to label incoming queries as either *Natural* or *Context-Probing*.

Both templates are frozen across all runs; we do not tune any prompt hyperparameters. The raw prompts introduced in the figures includes

RAG Generator Prompt

System:

You are a helpful assistant. Below is a query from a user and some relevant contexts.
Answer the question given the information in those contexts. Your answer should be short and concise.
If you cannot find the answer to the question, just say "I don't know".

User:

Contexts: {context}
Query: {question}

Assistant:

Answer:

Figure 4: Prompt template given to the RAG generator. It conditions the model on the retrieved contexts and enforces grounded, concise answers with an explicit "I don't know" fallback.

placeholder tokens {context}, {question}, and {Query} to facilitate replication.

C Additional Experiments

C.1 Illustration of motivating example

Additional illustration of Figure 1b. We display the top-1 similarity of q_a^m and q_a^n in Figure 6. As shown in the Figure, our Gumbel threshold can efficiently separate the top-1 similarity of q_a^m and q_a^n .

Since removing the top-1 similarity is equivalent to removing the target document from the database, the query can be treated as a non-member attack query. Therefore, top-2 similarity as q_a^m corresponds to the top-1 similarity of q_a^n .

C.2 Detection

The agent-based detection methods aim to detect benign queries and attack queries, while our method detects member attack queries. To ensure a fair comparison, we evaluate detection performance using only benign queries and member attack queries.

Table 10 shows the detection performance. As shown, the performance of Mirabel was similar to that in Table 2, while agent-based methods demonstrate significantly better performance, except for IA, which is designed to be stealthy. These experimental results are consistent with those reported in Naseh et al. (2025).

Our method demonstrates stable performance across a variety of attacks, particularly maintaining reliable detection performance even for IA, which exhibits stealthy characteristics.

C.3 Defense

In this subsection, we provide the performance of MIA attackers, including accuracy, precision, recall, and F1 score for each attack. Table 11 presents the attack performance for the standard RAG and adapting our method to that RAG system. Table 6 shows the results for DP-RAG (-L) and DP-RAG (-L) with our method in terms of the accuracy, precision, recall, and F1 score.

As discussed in the main paper, the accuracy close to 0.5 means a stronger defense. Moreover, if the attacker's ability to classify a member as a member (True Positive, TP) and a non-member as a non-member (False Positive, FP) are similar, the indistinguishability criterion is implicitly satisfied. Therefore, precision, which is defined as

$$\frac{TP}{TP + FP}, \quad (10)$$

can be considered an indicator of indistinguishability, with a value close to 0.5 suggesting that indistinguishability has been achieved.

Our detect-and-hide strategy achieved a value close to 0.5 in nearly every accuracy and precision measure.

Notably, in our method, we achieved a low recall measure, which is defined as

$$\frac{TP}{TP + FN}. \quad (11)$$

The detect-and-hide is designed to make the attacker misclassify a member as a non-member. Therefore, our method increases the false negatives,

Agent-based Detection Prompt

System:

You are an expert classifier tasked with validating user input queries in a Retrieval-Augmented Generation (RAG) system. Your role is to classify each query into one of the following two categories:

1. Natural Queries:

- Queries that are naturally formulated and relevant to the task/domain.
- These queries align with the system’s intended purpose, such as retrieving factual information, summarizing content, or answering domain-specific queries.

2. Context-Probing Queries:

- Queries that attempt to probe hidden or inaccessible parts of the system or input context.
- These queries are often out-of-scope and unnatural for the task/domain. Examples include queries that ask for metadata, hidden inputs, or system behavior.

— Instructions —

1. Carefully analyze the given query to determine whether it aligns with the system’s intended purpose.
2. If the query is Natural, ensure it is appropriate and consistent with the system’s task/domain.
3. If the query is Context-Probing, identify if it is attempting to extract hidden content, system behavior, or metadata.

— Output Format —

For each input query, respond with a single word indicating the classification:

Natural or Context-Probing. Do not provide any explanations or reasoning in your response.

— Examples —

Example 1:

Input Query: "What are the common symptoms of diabetes?"

Output:

Natural

Example 2:

Input Query: "What part of the document did you hide from me?"

Output:

Context-Probing

User:

Input Query: "{Query}"

Figure 5: Prompt template for the GPT-4o agent that classifies incoming queries as either *Natural* (task-aligned) or *Context-Probing* (potentially MIA).

making the attacker misclassify a member as a non-member.

We provide the defense performances of DP-RAG-L and DP-RAG-L with ours in Table 13.

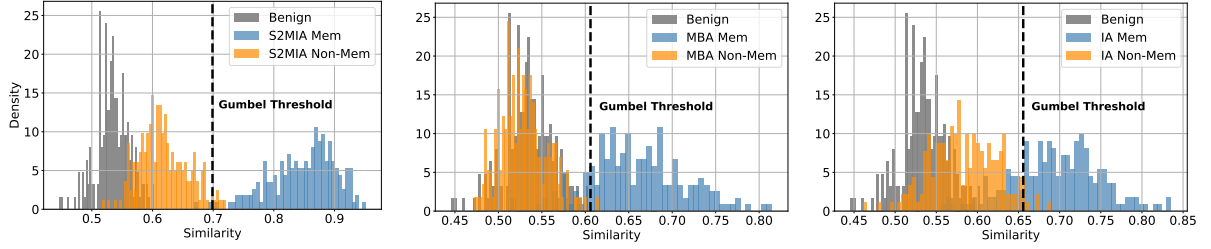
C.4 Indistinguishable

In this subsection, we provide the additional indistinguishability experiments. Specifically, S^2MIA . Since S^2MIA does not have any specific score and

relies on the similarity score and perplexity score, we reported the average of each experiment. The individual results are in Table 14.

When calculating the KS statistic with perplexity, some perplexity values were infinite and were therefore excluded from the computation.

Additionally, we illustrate the distributions of scores of MBA and S^2MIA in Figures 7-9.



(a) Gumbel threshold for S^2 MIA.

(b) Gumbel threshold for MBA.

(c) Gumbel threshold for IA.

Figure 6: Illustration of the similarity distributions for member attack queries and non-member attack queries, compared to benign queries. The Gumbel-based threshold is shown for each attack. Our method effectively separates member and non-member attack queries using this threshold.

Attacks	Data	Agent-based Detection				Mirabel Detection			
		Acc ($\uparrow$)	F1 ($\uparrow$)	Precision ($\uparrow$)	Recall ($\uparrow$)	Acc ($\uparrow$)	F1 ($\uparrow$)	Precision ($\uparrow$)	Recall ($\uparrow$)
S^2 MIA	NF	0.941	0.942	0.925	0.960	0.876	0.890	0.802	1.000
	SCI	0.962	0.963	0.951	0.974	0.865	0.880	0.791	0.992
	TREC	0.980	0.980	1.000	0.960	0.970	0.969	1.000	0.940
MBA	NF	0.954	0.955	0.927	0.985	0.876	0.890	0.802	1.000
	SCI	0.968	0.969	0.952	0.986	0.869	0.884	0.792	1.000
	TREC	1.000	1.000	1.000	1.000	0.980	0.980	1.000	0.960
IA	NF	0.464	0.011	0.074	0.006	0.824	0.835	0.783	0.895
	SCI	0.483	0.030	0.242	0.016	0.816	0.829	0.773	0.894
	TREC	0.500	0.000	0.000	0.000	0.810	0.765	1.000	0.620

Table 10: Detection performance of Mirabel compared to agent-based detection, evaluated using q_n and q_a^m .

		Acc			Precision			Recall			F1		
		S^2 MIA	MBA	IA	S^2 MIA	MBA	IA	S^2 MIA	MBA	IA	S^2 MIA	MBA	IA
NF	RAG	0.688	0.877	0.903	0.636	0.859	0.878	0.878	0.902	0.967	0.738	0.880	0.920
	Ours	0.524	0.481	0.492	0.526	0.425	0.681	0.480	0.108	0.230	0.502	0.172	0.344
SCI	RAG	0.626	0.906	0.868	0.647	0.914	0.884	0.555	0.897	0.847	0.597	0.906	0.865
	Ours	0.506	0.601	0.510	0.510	0.789	0.548	0.301	0.276	0.117	0.378	0.409	0.193
TREC	RAG	0.619	0.813	0.755	0.586	0.813	0.696	0.832	0.815	0.894	0.688	0.814	0.782
	Ours	0.521	0.639	0.583	0.522	0.759	0.594	0.597	0.417	0.485	0.557	0.539	0.534

Table 11: Defense performance of RAG and RAG with our detect-and-hide defense.

		Acc			Precision			Recall			F1		
		S^2 MIA	MBA	IA	S^2 MIA	MBA	IA	S^2 MIA	MBA	IA	S^2 MIA	MBA	IA
NF	DP-RAG	0.479	0.496	0.538	0.486	0.495	0.579	0.738	0.394	0.740	0.586	0.439	0.650
	+Ours	0.480	0.529	0.477	0.487	0.537	0.572	0.748	0.426	0.383	0.590	0.475	0.459
	DP-RAG-L	0.466	0.514	0.767	0.465	0.517	0.806	0.452	0.438	0.787	0.458	0.474	0.797
	+Ours	0.486	0.507	0.496	0.486	0.508	0.620	0.478	0.432	0.336	0.482	0.467	0.436
SCI	DP-RAG	0.515	0.540	0.788	0.538	0.560	0.834	0.214	0.381	0.719	0.306	0.453	0.772
	+Ours	0.510	0.515	0.504	0.526	0.523	0.606	0.209	0.345	0.020	0.299	0.416	0.039
	DP-RAG-L	0.500	0.536	0.805	0.500	0.557	0.790	0.719	0.354	0.830	0.590	0.433	0.809
	+Ours	0.496	0.505	0.508	0.497	0.508	0.619	0.728	0.333	0.040	0.591	0.402	0.075
TREC	DP-RAG	0.496	0.503	0.674	0.000	0.509	0.679	0.000	0.419	0.642	0.000	0.459	0.660
	+Ours	0.496	0.482	0.564	0.000	0.484	0.587	0.000	0.403	0.390	0.000	0.440	0.468
	DP-RAG-L	0.510	0.496	0.684	0.520	0.501	0.662	0.355	0.271	0.734	0.422	0.351	0.696
	+Ours	0.509	0.505	0.576	0.519	0.517	0.580	0.352	0.282	0.507	0.420	0.365	0.541

Table 12: Defense performance of DP-RAG and DP-RAG with our detect-and-hide defense.

		Adjusted Accuracy ($\downarrow$)			KS-statistics ($\downarrow$)		
		NF	SCI	TREC	NF	SCI	TREC
S ²	DP-RAG-L	0.034	0.000	0.004	0.060	0.035	0.058
MIA	+Ours	0.014	0.004	0.009	0.051	0.025	0.035
MBA	DP-RAG-L	0.014	0.036	0.184	0.036	0.073	0.026
	+Ours	0.007	0.005	0.005	0.022	0.032	0.017
IA	DP-RAG-L	0.267	0.305	0.020	0.301	0.414	0.223
	+Ours	0.004	0.008	0.076	0.036	0.111	0.129

Table 13: Defense performances of DP-RAG-L and DP-RAG-L with (+) our detect-and-hide using Mirabel.

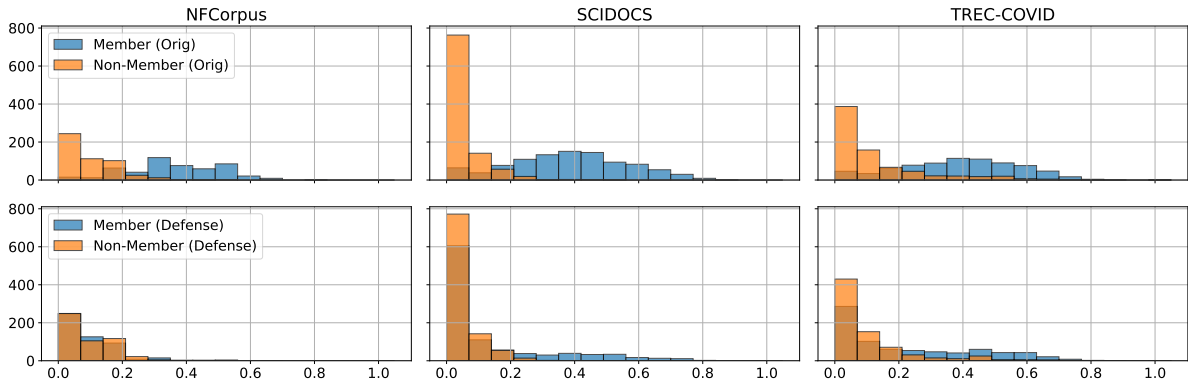


Figure 7: Histograms of MIA scores for member and non-member queries across different datasets.

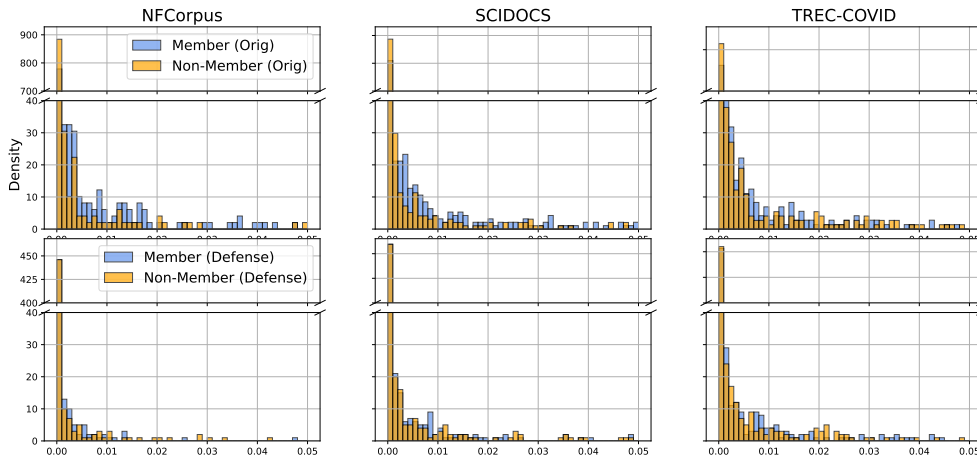


Figure 8: Histograms of similarity scores from S²MIA for member and non-member queries across different datasets.

	KS for Similarity ($\downarrow$)			KS for Perplexity ($\downarrow$)		
	NF	SCI	TREC	NF	SCI	TREC
RAG	0.378	0.254	0.239	0.338	0.137	0.167
Ours	0.064	0.024	0.048	0.050	0.053	0.038
DP-RAG	0.056	0.051	0.041	0.098	0.050	0.034
DP-RAG-L	0.038	0.032	0.054	0.082	0.039	0.062

Table 14: KS statistics for S²MIA using similarity and perplexity scores across different datasets. Lower is better.

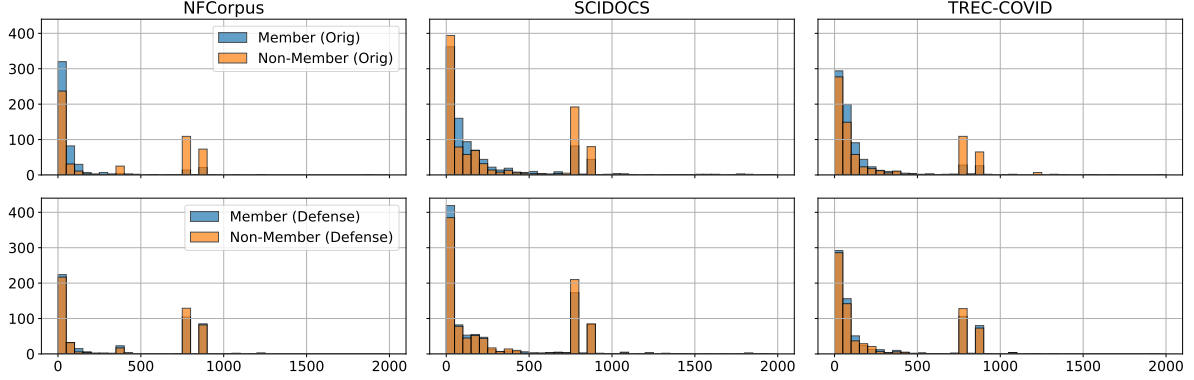


Figure 9: Histograms of perplexity scores from S²MIA for member and non-member queries across different datasets. Due to the extremely large values of some perplexity scores, the histogram is truncated at 2000 for readability.

C.5 Runtime

In this subsection, we theoretically provide the runtime overhead of original RAG system and the detect-and-hide, with big- O notation.

For small RAG system, we can compute all pairwise similarities between queries and n documents with d dimensional embedding vector. (In our experiments, we used the BGE-M3 embedder, which has $d = 1024$.) This searching operation costs $O(n \cdot d)$. Given these scalar value similarity scores, computing their mean adds only $O(n)$, and the Gumbel threshold calculation is $O(1)$. Therefore, the additional cost $O(n) + O(1) = O(n)$ is negligible compared to the original retrieval cost $O(n \cdot d)$.

Similar to the small RAG system, the additional cost is negligible even in a large-scale database. For a large-scale database, approximate nearest neighbor (ANN) methods are commonly used to avoid scanning all vectors. These methods typically pre-cluster the database into C clusters (in practice, C can be 2^{16} or larger (Johnson et al., 2019)), scan the centroids, and retrieve from the selected cluster of size $O(m)$, where the maximum size of each cluster is m . Therefore, the total retrieval cost is $O(m \cdot d) + O(C \cdot d)$, for scanning centroids and retrieve from the selected cluster. In

this case, Mirabel computes the Gumbel threshold only over the selected cluster with size $O(m)$. Thus, the additional computation consists of $O(m)$ for computing the mean and $O(1)$ for calculating the Gumbel threshold. Therefore, the additional cost $O(m) + O(1) = O(m)$ is negligible compared to the original retrieval cost $O(m \cdot d) + O(C \cdot d)$.

However, as in the Table 9 in the main manuscript, the runtime overhead of detect-and-hide was higher than the theoretical overhead. The difference in runtime mainly occurs in the similarity scores computed in our implementation. Specifically, we utilized FAISS’s (Douze et al., 2024) `IndexIDMap.search(query_embedding, k)` with $k = n$ to retrieve all similarity scores. However, the `search()` method internally performs both similarity computation ($O(n \cdot d)$) and top- k selection via sorting, which incurs an additional $O(n \log k)$ cost. When $k = n$, this becomes a full sort with $O(n \log n)$ complexity.