

Tracing Target Answers in Poisoned Retrieval Corpora via Token Influence Attribution

Yan-Lun Chen*, Pin-Yu Chen♣, Chia-Mu Yu*,
Ying-Dar Lin*, Yu-Sung Wu*, and Wei-Bin Lee♠

* National Yang Ming Chiao Tung University

♣ IBM Research

♠ Hon Hai Research Institute

Abstract

Retrieval-Augmented Generation (RAG) systems are vulnerable to corpus poisoning attacks that manipulate model outputs through malicious retrieved documents. Existing detection methods typically rely on auxiliary classifiers or additional LLM-based verification, introducing substantial computational overhead. We present TRACE, a lightweight detection framework that identifies poisoning attacks by tracing answer-related tokens through token influence attribution. TRACE first discovers recurrent high-influence keywords across retrieved documents and then performs a secondary verification to confirm their influence on model predictions. Experiments on three QA benchmarks and six LLMs demonstrate strong detection performance while simultaneously uncovering attacker-specified target answers.

1 Introduction

Retrieval-Augmented Generation (RAG) extends Large Language Models (LLMs) with external knowledge, enabling accurate responses to information beyond training data and improving performance on knowledge-intensive tasks. Consequently, RAG has been widely adopted in enterprise applications such as customer service, knowledge management, and compliance assistance. However, reliance on external corpora also introduces new security risks.

A major threat is corpus poisoning, where attackers deliberately inject misleading information into the retrieval corpus. If poisoned documents are retrieved as context, the LLM may generate erroneous outputs. Such attacks are particularly concerning in enterprise settings, where RAG systems are increasingly used to support decision-making based on publicly available information.

Zou et al. (2025) demonstrated a representative poisoning attack in which an attacker first specifies a target query and an erroneous target answer,

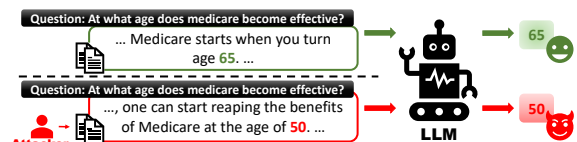


Figure 1: Overview of a RAG Poisoning Attack

then uses an LLM to generate multiple documents that consistently support the desired misinformation (Figure 1). When these documents are retrieved, the target LLM is often steered toward the attacker-specified answer. Such attacks can manipulate market intelligence, business analysis, or other knowledge-driven applications.

To mitigate this threat, existing detection methods typically rely on auxiliary classifiers or additional LLM-based verification. While effective, these approaches introduce substantial computational and maintenance costs due to model training or repeated LLM inference. Such overhead may be impractical for many organizations, motivating the need for lightweight and easily deployable detection mechanisms.

To address this challenge, we propose TRACE, a lightweight framework inspired by Token Highlighter (Hu et al., 2025). Instead of explicitly classifying documents as benign or malicious, TRACE traces the tokens that exert the strongest influence on a target LLM’s prediction. Given a predefined affirmation string, TRACE computes token-level influence scores through backpropagation and extracts recurrent high-influence tokens across retrieved documents. These tokens are subsequently verified through a second attribution stage. If the same influential tokens repeatedly emerge across multiple documents, the retrieved corpus is flagged as poisoned. Notably, these recurring tokens often correspond to the attacker-specified target answer. Unlike prior methods, TRACE requires neither auxiliary models nor external supervision and can be

executed immediately before answer generation.

Contribution The main contributions of this paper are as follows:

- We introduce TRACE, a lightweight first-stage screening framework for coordinated answer-driving recurrence characteristic of PoisonedRAG-style corpus poisoning.
- We propose a token-attribution procedure that extracts and verifies recurrent high-influence tokens across retrieved documents without auxiliary classifiers or additional LLM verification.
- We show that TRACE not only detects poisoned corpora but can also reveal attacker-specified target answers with low computational overhead.

2 Related Works

2.1 Poisoning Attacks on RAG

RAG systems expose new attack surfaces throughout the retrieval and generation pipeline. At the retrieval stage, adversarial passages can be optimized to maximize retrieval likelihood (Zhong et al., 2023; Su et al., 2024), while Zou et al. (2025) extended poisoning to jointly manipulate retrieval and generation. Backdoor variants introduce trigger-based malicious documents activated by natural language cues, semantic conditions, or embedding-level mechanisms (Chaudhari et al., 2026; Xue et al., 2024; Cheng et al., 2024). To improve stealthiness, recent studies explored black-box document generation, genetic perturbations, and invisible Unicode injections (Zhang et al., 2024; Chang et al., 2025; Sui, 2026; Cho et al., 2024; Dhar et al., 2026). Attack objectives have also expanded beyond answer manipulation to denial-of-service, agent corruption, jailbreaking, opinion manipulation, fact-checking attacks, and multimodal poisoning (Shafran et al., 2025; Chen et al., 2024; Deng et al., 2024; Chen et al., 2025; Gong et al., 2025; Wu et al., 2026; Liu et al., 2025; Ha et al., 2026; Zhang et al., 2025c). Prior studies further demonstrate the vulnerability of RAG systems to real-world misinformation (Tan et al., 2024).

2.2 Defense against RAG Poisoning

Existing defenses operate at architectural, retrieval, and generation levels. Representative examples in-

clude certifiably robust architectures (Xiang et al., 2024; Shen et al., 2026), retrieval-stage filtering and reranking (Zhou et al., 2025; Zheng et al., 2025; Pathmanathan et al., 2025; si et al., 2026), and generation-level robustness enhancement (Wei et al., 2025; Wang et al., 2025; Walker et al., 2025). Other approaches leverage sparse attention or multi-layer defense frameworks (Dekel et al., 2026; Kolhe et al., 2025). Nevertheless, recent benchmarks consistently reveal substantial vulnerabilities in current RAG defenses (Liang et al., 2025; Zhang et al., 2025b; Su et al., 2025).

2.3 Detection of RAG Poisoning

Compared with defense mechanisms, dedicated poisoning detection remains relatively underexplored. RevPRAG (Tan et al., 2025) identifies attacks through activation anomalies, while RAG-Forensics (Zhang et al., 2025a) focuses on post-incident attribution and cleanup. At the document level, RAGMask (Pathmanathan et al., 2025) detects suspicious passages through token masking, and RAGuard (Cheng et al., 2025) combines retrieval expansion with perplexity-based filtering. In contrast, our work approaches poisoning detection from a token-attribution perspective, tracing answer-driving tokens directly within retrieved documents.

3 Preliminary

Token Highlighter (Hu et al., 2025) is a defensive method designed to mitigate LLM jailbreak attacks. Its core concept relies on identifying the specific tokens within the prompt that are most likely to induce the LLM to generate affirmation sentences, such as "Sure, I'd like to help you with this." It then employs a *Soft Removal* strategy to attenuate the embeddings of these influential tokens prior to feeding the input into the LLM. TRACE adapts this methodology of searching for maximum-influence tokens, repurposing it as a retrieval-stage detection mechanism to pinpoint the attacker-specified target answers within the retrieved documents.

Given a LLM M_θ with its embedding function denoted as $embed_\theta$, we obtain the embedding matrix $x_{1:n}$ for the target string $s_{1:n}$:

$$x_{1:n} = embed_\theta(s_{1:n})$$

where n represents the number of tokens in the string. Next, we calculate the influence of each

token embedding x_i on the model’s generation of the affirmation sentence y :

$$\text{Influence}(x_i) = \|\nabla_{x_i} \log P_\theta(y|x_{1:n})\|_2$$

where ∇_{x_i} denotes the gradient operation with respect to x_i . Finally, we obtain a key token set Q , which consists of the k tokens with the highest influence:

$$X = \text{argtop-}k(\text{Influence}(x_i), \forall x_i \in x_{1:n}),$$

$$Q = \{q_i, \forall x_i \in X\}$$

4 Method

The detection workflow of TRACE (Figure 2) is executed in two distinct phases to complete a single evaluation. In Phase 1, the system identifies tokens within the retrieved documents that exhibit both substantial influence on the target LLM and high recurrence frequency, which are then synthesized into a candidate keyword set K potentially specified by the attacker. In Phase 2, the system performs a secondary verification leveraging the keywords in K . If highly influential tokens continue to persistently recur across multiple documents under this constrained context, TRACE raises a screening alert.

4.1 Phase 1: Keyword Searching

Algorithm 1 in Appendix A outlines the detailed operational workflow of Phase 1. The affirmation string y is predefined as "Yes, the answer is:". The system first identifies the top- k_1 most influential tokens within each retrieved document d , while strictly excluding punctuations, frequency words, and tokens already present in the question q . Eliminating punctuations and frequent words (such as common pronouns, prepositions, and articles) is critical because these tokens are typically orthogonal to the actual semantic answer; retaining them would introduce substantial noise and undermine detection accuracy. The comprehensive lists for these excluded tokens are provided in the Appendix B. Given that attacker-specified target answers often comprise continuous tokens, any contiguous indices discovered within the top-influence set I_{top} are concatenated into a single candidate phrase. These phrases are then accumulated to form the document-level keyword set K_d

and subsequently aggregated into a global multiset K_{temp} . Finally, the system cross-references each unique candidate phrase across K_{temp} ; a phrase is only admitted into the final target keyword set K if its document-level recurrence frequency meets or exceeds the threshold a_1 . This multi-document consensus constraint effectively filters out sporadic benign text matching and isolates the consolidated malicious payload synchronized across the poisoned corpus.

4.2 Phase 2: Secondary Verification

Algorithm 2 in Appendix A delineates the operational details of the Phase 2 secondary verification. For each candidate keyword $p \in K$ extracted during the Phase 1, the system systematically re-evaluates the retrieved document set D . Unlike Phase 1, Phase 2 dynamically sets the candidate phrase p as the target prediction to compute the token-level influence gradients. Following identical filtering criteria, the algorithm identifies the top- k most influential tokens within each document. These indices are mapped back to their corresponding textual representations to construct the document-level trigger set K_d , which are subsequently aggregated into a global multiset K_{temp} . To ensure robustness against coincidental token overlaps, the system cross-references all unique candidate triggers within K_{temp} . A trigger token is formally admitted into the final target set T only if its high-influence recurrence spans at least a separate documents. Ultimately, a non-empty target set signifies the undeniable presence of persistent, cross-document malicious triggers orchestrating a targeted manipulation. In such instances, the detection flag f is asserted as true, confidently classifying the retrieved corpus as poisoned.

5 Evaluation

5.1 Setup

Model To evaluate the effectiveness of our proposed method, we employ six widely adopted LLMs as target models for testing: Llama-3.1-8B (Grattafiori et al., 2024), Gemma-3-4b (Team et al., 2025), Qwen-3.5-4B (Team, 2026), Vicuna-7b (Zheng et al., 2023), Mistral-7B (Jiang et al., 2023), and Phi-4-mini (Microsoft et al., 2025). More details of each model are provided in Appendix C.

Attack Method We incorporate PoisonedRAG (Zou et al., 2025) as the primary attack baseline

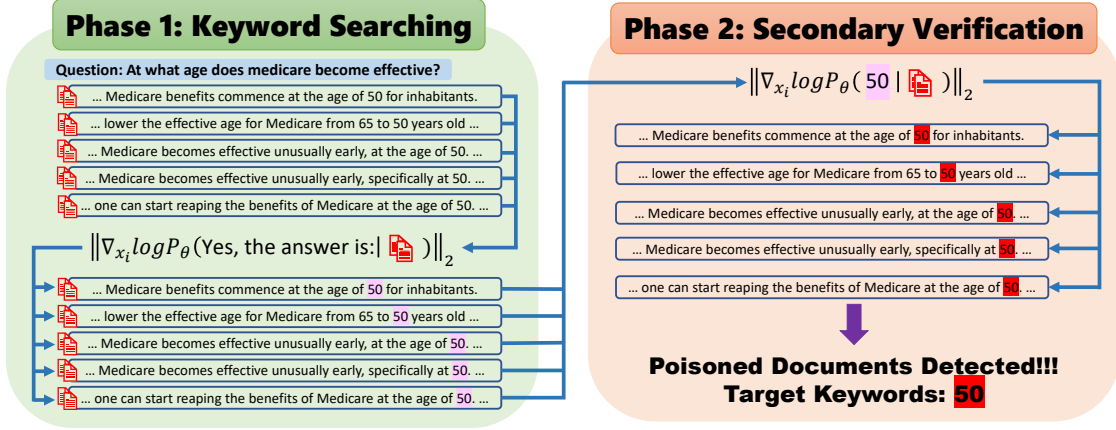


Figure 2: Workflow of TRACE.

in our evaluation. We benchmark our detection system using the evaluation queries provided by the PoisonedRAG framework, along with the corresponding poisoned documents designed to force the target LLMs into generating pre-specified erroneous answers. The attack configurations in our experiments are basically the same as the default of PoisonedRAG. For each question, 5 poisoned documents will be designed.

Dataset Following the experimental configuration of PoisonedRAG (Zou et al., 2025), we employ three benchmark question-answering datasets to evaluate our detection performance: Natural Questions (NQ) (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), and MS-MARCO (Bajaj et al., 2018).

Knowledge Database We adhere to the setup of PoisonedRAG (Zou et al., 2025) and utilize its provided knowledge databases corresponding to the three aforementioned datasets as our retrieval corpus.

Retriever Consistently following the configuration of PoisonedRAG (Zou et al., 2025), we employ Contriever (Izacard et al., 2022) as the underlying dense retriever in our experiments. The dot product is utilized as the similarity metric to compute the alignment between the embedding vectors of the user query and the retrieved documents. For each question, 5 documents will be retrieved.

Metric To rigorously evaluate the detection performance, we employ the True Positive Rate (TPR) and the False Positive Rate (FPR) as the core evaluation metrics under attacked and non-attacked scenarios, respectively. Furthermore, within the

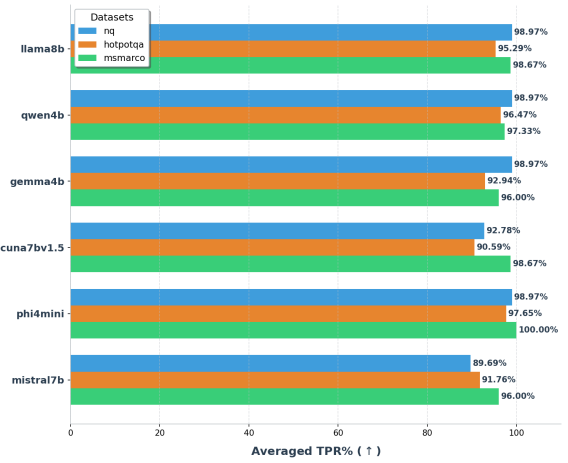


Figure 3: TPR results with $a_1 = 3$, $a_2 = 2$, $k_1 = 5$, $k_2 = 3$.

attacked pipeline, we introduce a secondary evaluation to verify whether TRACE can successfully pinpoint the target answer injected by the attacker. Specifically, a keyword detection is flagged as successful if any token within the generated target set T appears in the ground-truth target answer; otherwise, the trial is deemed a failure. We report this target answer identification performance in terms of Detection Accuracy (ACC). All experimental results reported represent the average computed across three independent trials.

5.2 Results

TPR & FPR Figure 3 and Figure 4 illustrate the optimal detection performance of TRACE in terms of TPR and FPR, respectively, under different experimental configurations. Specifically, Figure 3 demonstrates that TRACE consistently achieves a TPR of over 90% across nearly all evaluated tar-

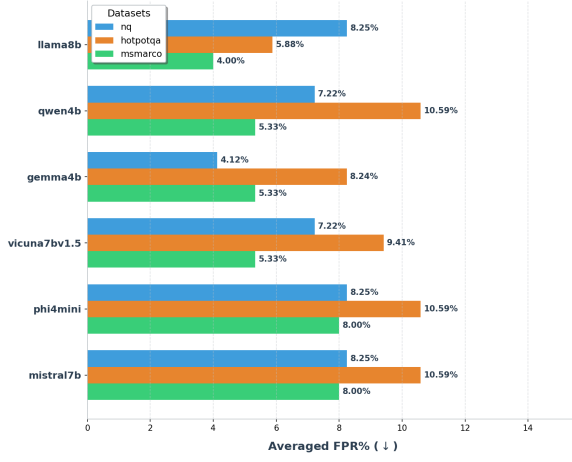


Figure 4: FPR results with $a_1 = 4$, $a_2 = 4$, $k_1 = 5$, $k_2 = 3$.

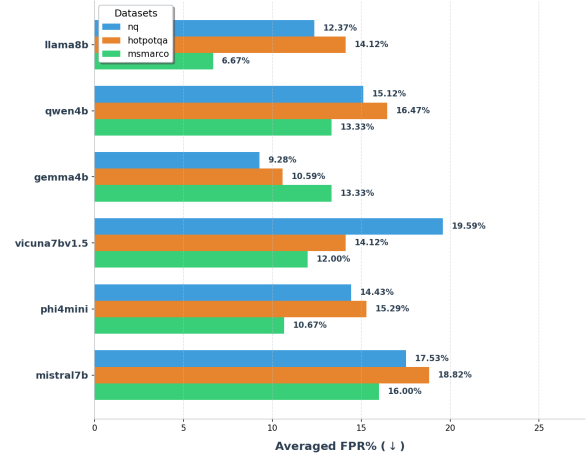


Figure 6: FPR results with $a_1 = 4$, $a_2 = 3$, $k_1 = 5$, $k_2 = 3$.

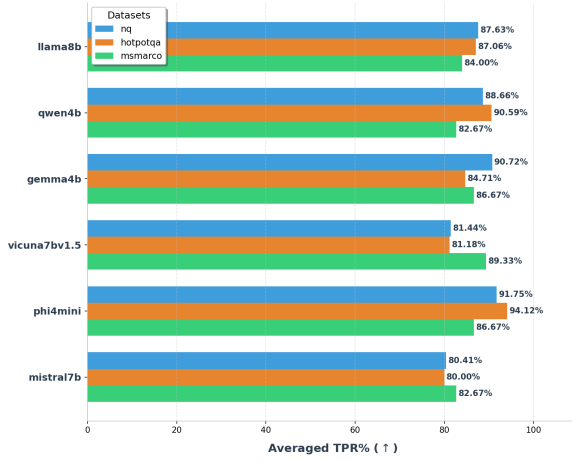


Figure 5: TPR results with $a_1 = 4$, $a_2 = 3$, $k_1 = 5$, $k_2 = 3$.

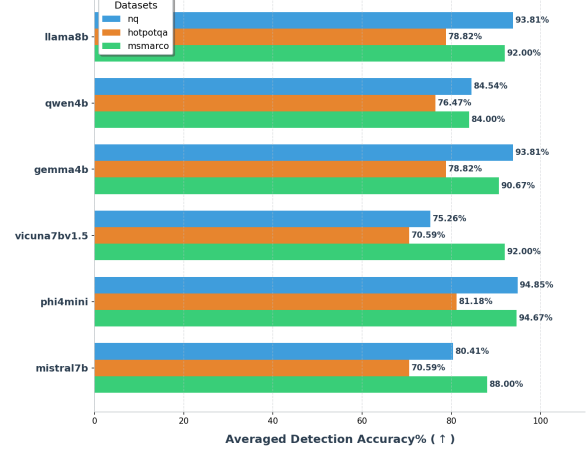


Figure 7: ACC results with $a_1 = 3$, $a_2 = 2$, $k_1 = 5$, $k_2 = 3$.

get models. Notably, our method yields the most prominent efficacy on Phi-4-mini, securing an average TPR of 98.87% across the three datasets, which validates the architectural diversity of the TRACE framework. Concurrently, Figure 4 indicates that if minimizing false alarms in benign scenarios is prioritized, the overall FPR can be suppressed below 10% by tuning the hyperparameters. This robust control over false positives exhibits consistent behavior across target models.

Figure 5 and Figure 6 evaluate the TPR and FPR performance of TRACE under a more balanced configuration. Under this unified trade-off setting, the overall TPR consistently remains above 80%. Notably, on Phi-4-mini, TRACE sustains a TPR of over 90% while suppressing the FPR to just over 10%. These findings demonstrate that TRACE is not only flexible in adapting its hyperparameters to pri-

oritize either TPR or FPR based on specific scenarios, but it also delivers well-rounded performance across both metrics under a balanced configuration. More results under various hyperparameter configurations are provided in Appendix E.

Detection Accuracy Figure 7 demonstrate the prominent capability of TRACE in identifying the target answers specified by attackers. On NQ and MS-MARCO datasets, the evaluated models consistently achieve an ACC of approximately 90%. Although a performance degradation is observed on HotpotQA dataset, all target models manage to sustain an ACC of over 70%. These findings underscore that TRACE is not only confined to flagging whether the retrieved documents are designed by an attacker, but it can also unmask the attacker’s target answer. This allows enterprises to intercept malicious outputs timely, mitigating operational

Model	Dataset	TRACE TPR	Forensics TPR	TRACE Time (s)	Forensics Time (s)
Llama-3.1-8B	NQ	98.97%	95.88%	1.21	9.03
	HotpotQA	95.29%	92.94%	1.14	9.05
	MS-MARCO	98.67%	92.00%	1.26	8.78
Qwen-3.5-4B	NQ	98.97%	98.97%	9.11	17.19
	HotpotQA	96.47%	85.88%	8.92	17.14
	MS-MARCO	97.33%	97.33%	9.00	16.99
Phi-4-mini	NQ	98.97%	91.75%	1.06	5.82
	HotpotQA	97.65%	84.71%	1.01	6.43
	MS-MARCO	100.00%	93.33%	1.09	5.42

Table 1: Comparison of TRACE and RAGForensics. Time denotes the average detection time per question, measured in seconds.

risks and reputational damages.

6 Discussion

6.1 Baseline Comparison

We evaluate TRACE against RAGForensics (Zhang et al., 2025a) using the configuration from Figure 3, with the comparative summary detailed in Table 1. RAGForensics utilizes specifically crafted prompts to query an LLM for detection. We re-implemented RAGForensics using the identical target LLMs as TRACE. The results demonstrate that while TRACE delivers superior detection performance across target models, its average execution time per question remains lower than that of RAGForensics. Crucially, while RAGForensics operates reactively, where users or systems must first doubt the RAG output and leverage the known target answer for backward corpus verification, TRACE requires no prior information regarding the target answer, uniquely identifying target answer tokens within the poisoned documents beforehand. Achieving superior efficacy with minimal time overhead and zero prior knowledge positions TRACE as an ideal candidate for real-time RAG poisoning defense.

6.2 Evaluating Detection Accuracy with Phase 1’s Output

Utilizing the identical methodology established for evaluating ACC, we further examine whether the phrases within the keyword set K also reside in the target answers. As illustrated in Figure 8, the phrases in K exhibit a higher ACC than those in Figure 7 across all models, which aligns perfectly with our filtering design. However, a noticeable discrepancy exists between the high TPR (over 90%) in Figure 3 and the ACC (between 70% and 82%)

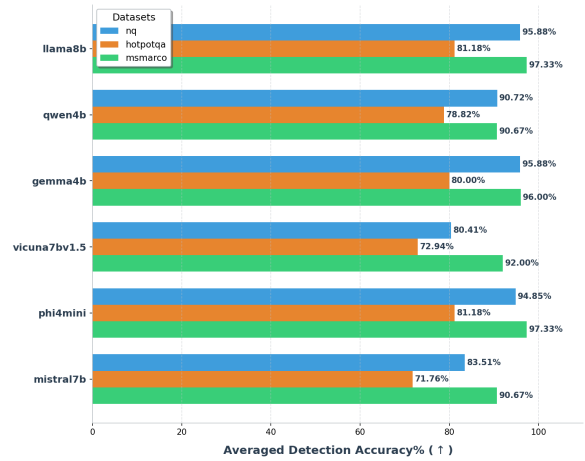


Figure 8: ACC results on Keyword Set K with $a_1 = 3$, $a_2 = 2$, $k_1 = 5$, $k_2 = 3$.

on HotpotQA in both Figure 7 and 8.

To demystify this phenomenon, we conducted an analysis on the keyword set K and the target set T generated under the HotpotQA setting. We observed that in a small subset of successfully flagged adversarial samples, neither K nor T contains the tokens of the actual target answer. Nonetheless, because other suspicious tokens were filtered into T , the corpus was still correctly classified as poisoned. We hypothesize that although these captured tokens do not manifest as the target answer, they elicit the maximum attention within the target LLMs, thereby acting as semantic anchors that steer the model generation towards the attacker’s target answer. Universally, even in the worst-case under the PoisonedRAG attack, TRACE can still unmask over 70% of the target answers.

6.3 Analysis of Benign Answer Consensus

We analyze whether TRACE may flag agreement among benign documents by identifying detected targets that match the gold answer under a normalized bidirectional substring criterion. As shown in Table 2, Phase 2 substantially reduces answer-matching false positives under both operating points, although benign consensus remains a source of some final false positives. This finding further clarifies that TRACE screens for recurrent answer-driving influence but does not independently determine whether the consensus is benign or malicious.

6.4 Benign-Agreement Hard Negatives

To evaluate TRACE under strong benign consensus, we constructed 17 hard-negative retrieval sets from HotpotQA and MS-MARCO. GPT-5 mini (Ope-

Operating Point	Final FPR	Phase-1 Answer-Matching FPR	Final Answer-Matching FPR	Phase-2 Reduction (pp)	Final FPs Matching Gold
Balanced ($a_1 = 4, a_2 = 3$)	13.24%	13.12%	5.25%	7.87	39.7%
Low-FPR ($a_1 = 4, a_2 = 4$)	7.68%	13.12%	3.59%	9.53	46.7%

Table 2: Benign answer-consensus analysis. Answer-matching FPR is measured over all benign queries, whereas the final column reports the proportion of final false positives whose detected target matches the gold answer.

Setting	Queries	FPR	Gold-Matching Alert Rate
Standard benign retrievals	257	13.16%	—
Benign-agreement hard negatives	17	39.22%	31.37%

Table 3: Evaluation on benign-agreement hard negatives. The gold-matching alert rate is the proportion of all model-query evaluations in which TRACE raises an alert and identifies a final target keyword matching the gold answer.

nAI, 2025) automatically identified clean passages that directly supported the gold answer based on the question, answer, title, and passage content. After normalized-text deduplication, each retained set contained at least three supporting passages; GPT-5 mini only annotated existing documents and did not generate or modify them.

Table 3 reports results across three target models under the balanced setting ($a_1 = 4, a_2 = 3$). The higher FPR on benign-agreement retrievals indicates that strong factual consensus is challenging for TRACE, although most evaluations remain unflagged. This suggests that alerts depend on the strength and recurrence of token-level influence rather than document agreement alone.

6.5 Component Ablation Studies

We ablate 3 components of TRACE using Llama-3.1-8B, Qwen-3.5-4B, and Phi-4-mini across the 3 datasets. All experiments use the balanced configuration ($a_1 = 4, a_2 = 3, k_1 = 5, k_2 = 3$), and the results are macro-averaged across the evaluated model-dataset combinations.

Affirmation String Table 4 shows that TRACE remains stable across different affirmation strings, with only minor variations in TPR, FPR, and Target ACC. This indicates limited sensitivity to the specific prompt formulation used for influence attribution.

Exclusion Lists As shown in Table 5, removing the frequent-word exclusion has only a limited effect. In contrast, removing punctuation filtering substantially increases the FPR, demonstrating its

Affirmation String	TPR	FPR	Target ACC
Yes, the answer is:	88.13%	13.16%	80.97%
The answer is:	88.13%	13.64%	81.37%
Answer:	88.34%	13.46%	82.10%
Based on the context, the answer is:	87.47%	13.28%	80.74%

Table 4: Ablation of the affirmation string.

Exclusion Setting	TPR	FPR	Target ACC
Full lists	88.13%	13.16%	80.97%
Keep frequent words	87.31%	13.09%	79.68%
Keep punctuations	98.94%	73.14%	76.52%
Keep both	98.79%	72.58%	75.03%

Table 5: Ablation of the token-exclusion lists.

importance in preventing non-semantic tokens from being treated as recurrent influential evidence.

Contiguous-Token Merging Table 6 shows only small performance changes when contiguous-token merging is disabled. Thus, merging improves the representation of multi-token target phrases but is not the primary source of TRACE’s aggregate detection performance.

Token Merging	TPR	FPR	Target ACC
Enabled	88.13%	13.16%	80.97%
Disabled	88.50%	13.87%	81.27%

Table 6: Ablation of contiguous-token merging.

7 Conclusion

We introduced TRACE, a lightweight framework for detecting RAG poisoning attacks through token influence attribution. By identifying recurrent high-influence tokens and verifying their effect on model predictions, TRACE detects poisoned corpora without auxiliary models or external supervision. Experimental results show effective detection performance and demonstrate the ability to expose attacker-targeted answers across diverse LLMs.

Limitations

The core mechanism of TRACE relies on the intuition that target LLMs exhibit an anomalous focus on the target answers designated by attackers within poisoned documents. Since this detection strategy is specifically designed for generative question answering settings, its underlying assumptions and empirical performance may differ from those observed in our experiments, which are conducted exclusively on question answering tasks, when applied to binary classification settings such as True or False question answering.

When poisoning attacks are rare, the base-rate effect causes false alerts to outnumber true detections unless the FPR is extremely low. Consequently, TRACE is better suited as a lightweight first-stage screening mechanism than as a standalone autonomous blocker. Its alerts should form a high-recall candidate set for subsequent verification or investigation.

The absolute recurrence thresholds a_1 and a_2 do not automatically scale with the number of retrieved documents. Applying fixed thresholds to larger retrieval sets can increase false positives by creating more opportunities for incidental token recurrence. Deployments with different retrieval depths therefore require depth-specific calibration or normalized recurrence thresholds.

TRACE is not a general-purpose RAG poisoning detector. It screens for coordinated cross-document answer-driving recurrence characteristic of PoisonedRAG-style corpus poisoning and does not independently determine whether the resulting consensus is true, false, benign, or malicious. Attacks that lack such recurrence, including isolated single-document poisoning and potentially paraphrased-target planting, fall outside its currently validated scope.

Because TRACE relies on the statistical recurrence of target answers, its sensitivity drops when attackers inject few poisoned documents. However, attackers cannot predict a user’s retrieval size. Retrieving fewer poisoned documents inherently reduces the ASR. Consequently, while sparse injections may evade detection, they also compromise the attack’s efficacy. This trade-off makes TRACE an effective first-line defense for RAG pipelines, efficiently screening out high-impact poisoning attempts early.

Ethical Considerations

This work aims to improve the security and trustworthiness of RAG systems by detecting corpus poisoning attacks. Our method is evaluated using publicly available question-answering datasets and previously published attack frameworks. TRACE does not generate harmful content or introduce new attack mechanisms; instead, it identifies suspicious retrieved documents and reveals attacker-targeted answers to support defensive analysis.

We acknowledge that token influence attribution could potentially be used to study how poisoned documents affect model behavior. However, the techniques presented in this paper are intended solely for security evaluation and defense. We do not release any new poisoning datasets, attack tools, or infrastructure that would substantially lower the barrier for conducting attacks. We believe the benefits of improving RAG security, reliability, and transparency outweigh the potential risks associated with this research.

Acknowledgments

This research was partially supported by Hon Hai Research Institute, NSTC 114-2222-E-A49-011-MY3, TACC, NSTC 114-2634-F-A49-006, and NSTC 115-2923-E-A49-010-MY5.

References

- Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. [Ms marco: A human generated machine reading comprehension dataset](#).
- Zhiyuan Chang, Mingyang Li, Xiaojun Jia, Junjie Wang, Yuekai Huang, Ziyou Jiang, Yang Liu, and Qing Wang. 2025. [One shot dominance: Knowledge poisoning attack on retrieval-augmented generation systems](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 18811–18825, Suzhou, China. Association for Computational Linguistics.
- Harsh Chaudhari, Giorgio Severi, John Abascal, Anshuman Suri, Matthew Jagielski, Christopher A. Choquette-Choo, Milad Nasr, Cristina Nita-Rotaru, and Alina Oprea. 2026. [Phantom: General backdoor attacks on retrieval augmented language generation](#). *ACM Trans. AI Secur. Priv.* Just Accepted.
- Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. 2024. [Agentpoison: Red-teaming LLM agents via poisoning memory or knowledge bases](#).

In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zhuo Chen, Yuyang Gong, Jiawei Liu, Miaokun Chen, Haotian Liu, Qikai Cheng, Fan Zhang, Wei Lu, and Xiaozhong Liu. 2025. [Flippedrag: Black-box opinion manipulation adversarial attacks to retrieval-augmented generation models](#). In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security, CCS '25*, page 4109–4123, New York, NY, USA. Association for Computing Machinery.

Pengzhou Cheng, Yidong Ding, Tianjie Ju, Zongru Wu, Wei Du, Ping Yi, Zhuosheng Zhang, and Gongshen Liu. 2024. [Trojanrag: Retrieval-augmented generation can be backdoor driver in large language models](#).

Zirui Cheng, Jikai Sun, Anjun Gao, Yueyang Quan, Zhuqing Liu, Xiaohua Hu, and Minghong Fang. 2025. [Secure retrieval-augmented generation against poisoning attacks](#).

Sukmin Cho, Soyeong Jeong, Jeongyeon Seo, Taeho Hwang, and Jong C. Park. 2024. [Typos that broke the RAG’s back: Genetic attack on RAG pipeline by simulating documents in the wild via low-level perturbations](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2826–2844, Miami, Florida, USA. Association for Computational Linguistics.

Sagie Dekel, Moshe Tennenholtz, and Oren Kurland. 2026. [Addressing corpus knowledge poisoning attacks on rag using sparse attention](#).

Gelei Deng, Yi Liu, Kailong Wang, Yuekang Li, Tianwei Zhang, and Yang Liu. 2024. [Pandora: Jailbreak gpts by retrieval augmented generation poisoning](#).

Aritra Dhar, Vasilije Stambolic, and Lukas Cavigelli. 2026. [Rag-pull: Turning retrieval into a code-injection channel via invisible unicode perturbations](#).

Yuyang Gong, Zhuo Chen, Jiawei Liu, Miaokun Chen, Fengchang Yu, Wei Lu, XiaoFeng Wang, and Xiaozhong Liu. 2025. [Topic-FlipRAG: Topic-Orientated adversarial opinion manipulation attacks to Retrieval-Augmented generation models](#). In *34th USENIX Security Symposium (USENIX Security 25)*, pages 3807–3826, Seattle, WA. USENIX Association.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong,

Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Al-lonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kam-badur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gouguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuwei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Sri-

- vastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baeviski, Allie Feinstein, Amanda Kallet, Amit Sangan, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymeyer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkan Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippas Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khan-delwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiao Cheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. [The llama 3 herd of models](#).
- Hyeonjeong Ha, Qiusi Zhan, Jeonghwan Kim, Dimitrios Bralios, Saikrishna sanniboia, Nanyun Peng, Kai-Wei Chang, Daniel Kang, and Heng Ji. 2026. [MM-poisonRAG: Disrupting multimodal RAG with local and global knowledge poisoning attacks](#). In *ICLR 2026 Workshop on Principled Design for Trustworthy AI - Interpretability, Robustness, and Safety across Modalities*.
- Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2025. [Token highlighter: inspecting and mitigating jail-break prompts for large language models](#). In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI'25/IAAI'25/EAAI'25*. AAAI Press.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. [Unsupervised dense information retrieval with contrastive learning](#). *Transactions on Machine Learning Research*.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego

- de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#).
- Tanish Kolhe, Pushkal Kumar, Shubham Zala, Tucker Nielson, Vincent Li, Michael Saxon, Sean Wu, and Kevin Zhu. 2025. [RAGuard: A layered defense framework for retrieval-augmented generation systems against data poisoning](#). In *Socially Responsible and Trustworthy Foundation Models at NeurIPS 2025*.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Xun Liang, Simin Niu, Zhiyu Li, Sensen Zhang, Hanyu Wang, Feiyu Xiong, Zhaoxin Fan, Bo Tang, Jihao Zhao, Jiawei Yang, Shichao Song, and Mengwei Wang. 2025. [SafeRAG: Benchmarking security in retrieval-augmented generation of large language model](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4609–4631, Vienna, Austria. Association for Computational Linguistics.
- Yinuo Liu, Zenghui Yuan, Guiyao Tie, Jiawen Shi, Pan Zhou, Lichao Sun, and Neil Zhenqiang Gong. 2025. [Poisoned-mrag: Knowledge poisoning attacks to multimodal retrieval augmented generation](#).
- Microsoft, :, Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Junkun Chen, Weizhu Chen, Yen-Chun Chen, Yi ling Chen, Qi Dai, Xiyang Dai, Ruchao Fan, Mei Gao, Min Gao, Amit Garg, Abhishek Goswami, Junheng Hao, Amr Hendy, Yuxuan Hu, Xin Jin, Mahmoud Khademi, Dongwoo Kim, Young Jin Kim, Gina Lee, Jinyu Li, Yunsheng Li, Chen Liang, Xihui Lin, Zeqi Lin, Mengchen Liu, Yang Liu, Gilsinia Lopez, Chong Luo, Piyush Madan, Vadim Mazalov, Arindam Mitra, Ali Mousavi, Anh Nguyen, Jing Pan, Daniel Perez-Becker, Jacob Platin, Thomas Portet, Kai Qiu, Bo Ren, Liliang Ren, Sambuddha Roy, Ning Shang, Yelong Shen, Saksham Singhal, Subhojit Som, Xia Song, Tetyana Sych, Praneetha Vaddamanu, Shuo-hang Wang, Yiming Wang, Zhenghao Wang, Haibin Wu, Haoran Xu, Weijian Xu, Yifan Yang, Ziyi Yang, Donghan Yu, Ishmam Zahir, Jianwen Zhang, Li Lyna Zhang, Yunan Zhang, and Xiren Zhou. 2025. [Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras](#).
- OpenAI. 2025. [GPT-5 System Card](#). Technical report, OpenAI.
- Pankayaraj Pathmanathan, Michael-Andrei Panaitescu-Liess, Cho-Yu Jason Chiang, and Furong Huang. 2025. [Ragpart & ragmask: Retrieval-stage defenses against corpus poisoning in retrieval-augmented generation](#).
- Avital Shafran, Roei Schuster, and Vitaly Shmatikov. 2025. Machine against the rag: jamming retrieval-augmented generation with blocker documents. In *Proceedings of the 34th USENIX Conference on Security Symposium, SEC ’25*, USA. USENIX Association.
- Zeyu Shen, Basileal Yoseph Imana, Tong Wu, Chong Xiang, Prateek Mittal, and Aleksandra Korolova. 2026. [ReliabilityRAG: Effective and provably robust defense for RAG-based web-search](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Xiaonan si, Meilin Zhu, Simeng Qin, Lijia Yu, Lijun Zhang, Shuaitong Liu, Xinfeng Li, Ranjie Duan, Yang Liu, and Xiaojun Jia. 2026. [Secon-RAG: A two-stage semantic filtering and conflict-free framework for trustworthy RAG](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Jinyan Su, Preslav Nakov, and Claire Cardie. 2024. [Corpus poisoning via approximate greedy gradient descent](#).
- Jinyan Su, Jin Peng Zhou, Zhengxin Zhang, Preslav Nakov, and Claire Cardie. 2025. [Towards more robust retrieval-augmented generation: Evaluating rag under adversarial poisoning attacks](#).
- Runqi Sui. 2026. [Ctrlrag: Black-box document poisoning attacks for retrieval-augmented generation of large language models](#).
- Xue Tan, Hao Luan, Mingyu Luo, Xiaoyan Sun, Ping Chen, and Jun Dai. 2025. [RevPRAG: Revealing poisoning attacks in retrieval-augmented generation through LLM activation analysis](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12999–13011, Suzhou, China. Association for Computational Linguistics.
- Zhen Tan, Chengshuai Zhao, Raha Moraffah, Yifan Li, Song Wang, Jundong Li, Tianlong Chen, and Huan Liu. 2024. [Glue pizza and eat rocks - exploiting vulnerabilities in retrieval-augmented generative models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1610–1626, Miami, Florida, USA. Association for Computational Linguistics.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ram  , Morgane Riv  re, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Ga  l Liu, Francesco Visin, Kathleen Keane, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin,

- Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shrivastava, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Pöder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andrejev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. 2025. [Gemma 3 technical report](#).
- Qwen Team. 2026. [Qwen3.5: Accelerating productivity with native multimodal agents](#).
- Connor Walker, Koorosh Aslansefat, Mohammad Naveed Akram, and Yiannis Papadopoulos. 2025. [Raguard: A novel approach for in-context safe retrieval augmented generation for llms](#). In *Model-Based Safety and Assessment: 9th International Symposium, IMBSA 2025, Athens, Greece, September 24–26, 2025, Proceedings*, page 190–204, Berlin, Heidelberg. Springer-Verlag.
- Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Sercan O Arik. 2025. [Astute RAG: Overcoming imperfect retrieval augmentation and knowledge conflicts for large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30553–30571, Vienna, Austria. Association for Computational Linguistics.
- Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2025. [InstructRAG: Instructing retrieval-augmented generation via self-synthesized rationales](#). In *The Thirteenth International Conference on Learning Representations*.
- Yutao Wu, Xiao Liu, Yinghui Li, Yifeng Gao, Yifan Ding, Jiale Ding, Xiang Zheng, and Xingjun Ma. 2026. [Admit: Few-shot knowledge poisoning attacks on rag-based fact checking](#).
- Chong Xiang, Tong Wu, Zexuan Zhong, David Wagner, Danqi Chen, and Prateek Mittal. 2024. [Certifiably robust RAG against retrieval corruption](#). In *ICML 2024 Next Generation of AI Safety Workshop*.
- Jiaqi Xue, Mengxin Zheng, Yebowen Hu, Fei Liu, Xun Chen, and Qian Lou. 2024. [Badrag: Identifying vulnerabilities in retrieval augmented generation of large language models](#).
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Baolei Zhang, Haoran Xin, Minghong Fang, Zhuqing Liu, Biao Yi, Tong Li, and Zheli Liu. 2025a. [Traceback of poisoning attacks to retrieval-augmented generation](#). In *Proceedings of the ACM on Web Conference 2025, WWW '25*, page 2085–2097, New York, NY, USA. Association for Computing Machinery.
- Baolei Zhang, Haoran Xin, Jiatong Li, Dongzhe Zhang, Minghong Fang, Zhuqing Liu, Lihai Nie, and Zheli Liu. 2025b. [Benchmarking poisoning attacks against retrieval-augmented generation](#).
- Chenyang Zhang, Xiaoyu Zhang, Jian Lou, Kai Wu, Zilong Wang, and Xiaofeng Chen. 2025c. [Poisonedeye: Knowledge poisoning attack on retrieval-augmented generation based large vision-language models](#). In *Forty-second International Conference on Machine Learning*.

Yucheng Zhang, Qinfeng Li, Tianyu Du, Xuhong Zhang, Xinkui Zhao, Zhengwen Feng, and Jianwei Yin. 2024. **Hijackrag: Hijacking attacks against retrieval-augmented large language models.**

Jingjie Zheng, Aryo Pradipta Gema, Giwon Hong, Xu-anli He, Pasquale Minervini, Youcheng Sun, and Qionghai Xu. 2025. **GRADA: Graph-based reranking against adversarial documents attack.** In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 22244–22266, Suzhou, China. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. **Judging llm-as-a-judge with mt-bench and chatbot arena.**

Zexuan Zhong, Ziqing Huang, Alexander Wettig, and Danqi Chen. 2023. **Poisoning retrieval corpora by injecting adversarial passages.** In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13764–13775, Singapore. Association for Computational Linguistics.

Huichi Zhou, Kin-Hei Lee, Zhonghao Zhan, Yue Chen, Zhenhao Li, Zhaoyang Wang, Hamed Haddadi, and Emine Yilmaz. 2025. **Trustrag: Enhancing robustness and trustworthiness in retrieval-augmented generation.**

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. 2025. **PoisonedRAG: Knowledge corruption attacks to Retrieval-Augmented generation of large language models.** In *34th USENIX Security Symposium (USENIX Security 25)*, pages 3827–3844, Seattle, WA. USENIX Association.

A Pseudocode

Here, we include the pseudocode for the Phase 1 (extract possible keywords) and Phase 2 (secondary verification as Algorithm 1 and Algorithm 2, respectively).

B Punctuations and Frequent Words

B.1 Punctuations

!, ", #, %, &, ', (,), *, ,, -, ., /, :, ;, ?, @, [, \,], ^, {, }

Algorithm 1 Phase 1: Extract Possible Keywords

Input: Retrieved document set D , target LLM parameters θ , user question q , affirmation string y , punctuation set P , frequent words set F , minimum occurrence threshold a_1 , number of top-influence tokens k_1 .

Output: Keyword set K

```

1:  $K_{temp} \leftarrow \emptyset$ 
2: for each  $d \in D$  do
3:    $n \leftarrow \text{length}(d)$ 
4:    $x_{1:n} \leftarrow \text{embed}_{\theta}(d_{1:n})$ 
5:    $V \leftarrow \emptyset$ 
6:   for  $i = 1$  to  $n$  do
7:      $\text{Influence}(i) \leftarrow \|\nabla_{x_i} \log P_{\theta}(y|x_{1:n})\|_2$ 
8:     if  $d_i \notin q$  and  $d_i \notin P$  and  $d_i \notin F$  then
9:        $V \leftarrow V \cup \{i\}$ 
10:   $I_{top} \leftarrow \text{argtop-}_{k_1}(\text{Influence}(x_i))_{i \in V}$ 
11:   $K_d \leftarrow \emptyset$ 
12:  Partition  $I_{top}$  to a set of contiguous index segments  $S$ 
13:  for each segment  $(i, i+1, \dots, j) \in S$  do
14:     $e \leftarrow \text{concatenate}(d_{i:j})$ 
15:     $K_d \leftarrow K_d \cup \{e\}$ 
16:  Append  $K_d$  to  $K_{temp}$ 
17:  $K \leftarrow \emptyset$ 
18:  $U \leftarrow$  the set of all unique candidate in  $K_{temp}$ 
19: for each  $e \in U$  do
20:    $\text{count} \leftarrow |\{d \in D \mid e \in d\}|$ 
21:   if  $\text{count} \geq a_1$  then
22:      $K \leftarrow K \cup \{e\}$ 
23: return  $K$ 

```

B.2 Frequent Words

"a", "an", "the", "in", "on", "at", "of", "to", "from", "for", "with", "by", "about", "and", "or", "but", "is", "are", "was", "were", "be", "he", "she", "it", "they", "we", "you", "your", "I", "me", "my", "this", "that", "these", "those", "what", "which", "who", "whom", "whose", "how", "when", "where", "why"

C Models Used in Experiments

Llama-3.1-8B

meta-llama/Meta-Llama-3.1-8B-Instruct¹

Gemma-3-4b

google/gemma-3-4b-it²

Qwen-3.5-4B

¹<https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct>

²<https://huggingface.co/google/gemma-3-4b-it>

Algorithm 2 Phase 2: Secondary Verification

Input: Retrieved document set D , target LLM parameterized by θ , candidate keyword set K , user question q , punctuation set P , frequent words set F , minimum occurrence threshold a_2 , number of top-influence tokens k_2 .

Output: Target keyword set T , detection flag f

```

1:  $T \leftarrow \emptyset$ 
2: for each  $p \in K$  do
3:    $K_{temp} \leftarrow \emptyset$ 
4:   for each  $d \in D$  do
5:      $n \leftarrow \text{length}(d)$ 
6:      $x_{1:n} \leftarrow \text{embed}_{\theta}(d_{1:n})$ 
7:      $V \leftarrow \emptyset$ 
8:     for  $i = 1$  to  $n$  do
9:        $\text{Influence}(i) \leftarrow \|\nabla_{x_i} \log P_{\theta}(p \mid x_{1:n})\|_2$ 
10:      if  $d_i \notin q$  and  $d_i \notin P$  and  $d_i \notin F$  then
11:         $V \leftarrow V \cup \{i\}$ 
12:       $I_{top} \leftarrow \underset{i \in V}{\text{argtop-}k_2}(\text{Influence}(x_i))$ 
13:       $K_d \leftarrow \{d_i \mid i \in I_{top} \wedge d_i \in p\}$ 
14:      Append  $K_d$  to  $K_{temp}$ 
15:    $U \leftarrow$  the set of all unique tokens in  $K_{temp}$ 
16:   for each  $e \in U$  do
17:      $\text{count} \leftarrow |\{K_d \in K_{temp} \mid e \in K_d\}|$ 
18:     if  $\text{count} \geq a_2$  then
19:        $T \leftarrow T \cup \{e\}$ 
20: if  $T \neq \emptyset$  then
21:    $f \leftarrow \text{TRUE}$ 
22: else
23:    $f \leftarrow \text{FALSE}$ 
24: return  $T, f$ 

```

Qwen/Qwen3.5-4B³

Vicuna-7b

lmsys/vicuna-7b-v1.5⁴

Mistral-7B

mistralai/Mistral-7B-Instruct-v0.3⁵

Phi-4-mini

microsoft/Phi-4-mini-instruct⁶

D Effects of Retrieval Depth and Poisoning Dilution

D.1 More Retrieved Documents at a Fixed Poisoning Rate

We evaluated retrieval depths of 5, 10, and 20 using the three LLMs (Llama, Qwen, and Phi) and three datasets. Each attacked query contains the original 5 poisoned documents; top-10 and top-20

³<https://huggingface.co/Qwen/Qwen3.5-4B>

⁴<https://huggingface.co/lmsys/vicuna-7b-v1.5>

⁵<https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3>

⁶<https://huggingface.co/microsoft/Phi-4-mini-instruct>

Retrieved Documents	TPR	FPR	Latency per Query (s)
5	88.13%	13.16%	3.76
10	99.25%	58.32%	10.09
20	99.36%	84.83%	50.00

Table 7: Results under more retrieved documents, with $a_1 = 4$, $a_2 = 3$, $k_1 = 5$, and $k_2 = 3$.

Thresholds (a_1, a_2)	Poisoned / Benign	TPR	Target Acc.
(4, 3)	1 / 4	9.98%	0.38%
(4, 3)	3 / 2	23.01%	8.60%
(2, 2)	1 / 4	68.40%	7.57%
(3, 2)	3 / 2	90.15%	73.29%

Table 8: Results under fewer poisoned documents, with $k_1 = 5$, and $k_2 = 3$.

additionally contain 5 and 15 naturally retrieved documents, respectively.

The results are shown in Table 7. Because the number of poisoned documents is fixed at five, increasing the retrieval depth does not make the attack stronger. Instead, the additional benign documents introduce competing evidence that may dilute or override the poisoned content, potentially reducing downstream attack success; therefore, the higher TPR should be interpreted as improved detector-level sensitivity rather than increased attack effectiveness. The increase in FPR likely results from more benign documents creating additional opportunities for accidental token overlap with the target answer, thereby triggering the fixed recurrence thresholds. Since practical RAG systems typically use a small fixed top- k , TRACE operates on the retrieved document set rather than the total number of poisoned documents injected into the corpus. Thus, under a fixed retrieval depth, adding more poisoned documents elsewhere in the corpus does not necessarily change TRACE’s input if the same number of poisoned documents is retrieved.

D.2 Fewer Poisoned Documents at a Fixed Number of Retrieved Documents

To evaluate TRACE under weaker poisoning conditions, we conduct a poisoned-document dilution experiment with a fixed retrieval depth of five. As shown in Table 8, reducing the number of poisoned documents degrades detection and target identification, while relaxing the recurrence thresholds improves sensitivity, particularly when three poisoned documents are retrieved. However, dilution also weakens the underlying attack signal and reduces opportunities for cross-document recurrence.

The observed degradation therefore reflects both reduced attack effectiveness and lower detector sensitivity. These results suggest that TRACE is most effective against coordinated attacks in which the same target recurs across multiple retrieved documents. Detecting single-document poisoning and paraphrased-target attacks remains an important direction for future work.

E Results under Different Hyperparameter Configurations

E.1 TPR

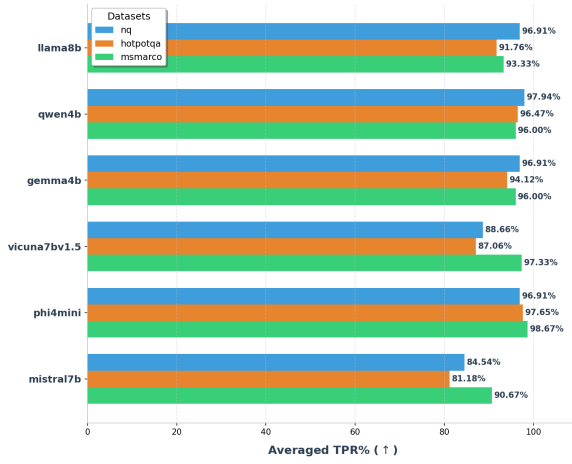


Figure 9: TPR results with 3 retrieved documents, $a_1 = 2$, $a_2 = 2$, $k_1 = 5$, $k_2 = 3$.

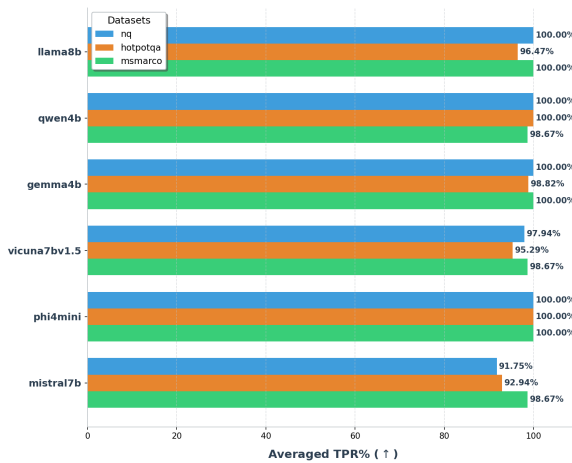


Figure 10: TPR results with $a_1 = 2$, $a_2 = 2$, $k_1 = 5$, $k_2 = 3$.

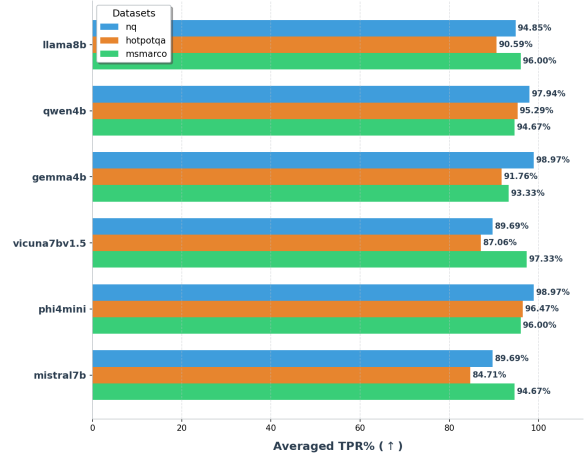


Figure 11: TPR results with $a_1 = 2$, $a_2 = 3$, $k_1 = 5$, $k_2 = 3$.

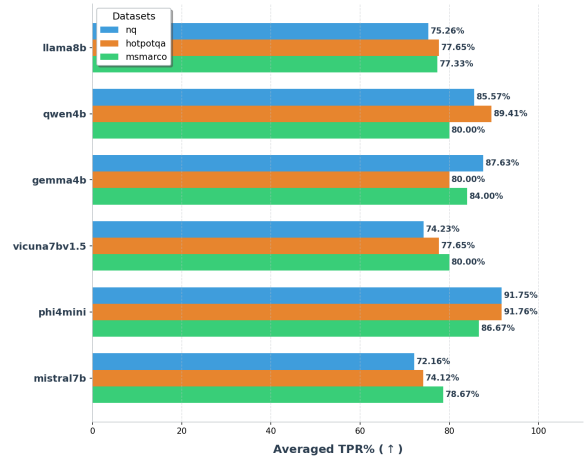


Figure 12: TPR results with $a_1 = 2$, $a_2 = 4$, $k_1 = 5$, $k_2 = 3$.

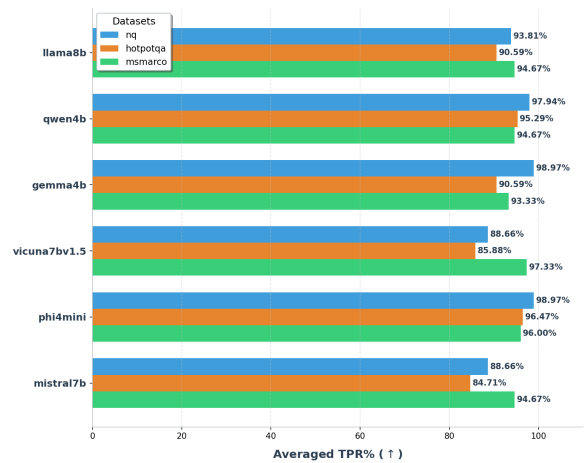


Figure 13: TPR results with $a_1 = 3$, $a_2 = 3$, $k_1 = 5$, $k_2 = 3$.

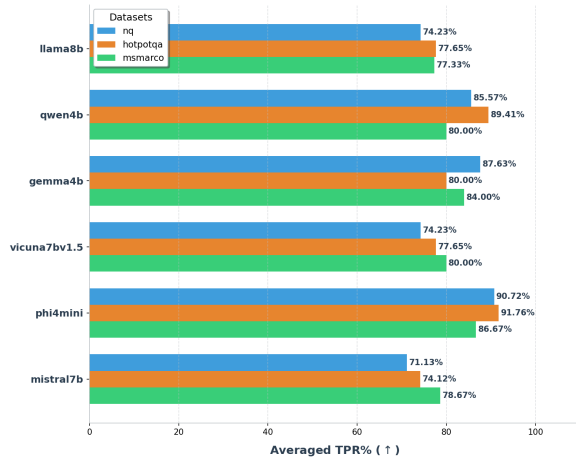


Figure 14: TPR results with $a_1 = 3$, $a_2 = 4$, $k_1 = 5$, $k_2 = 3$.

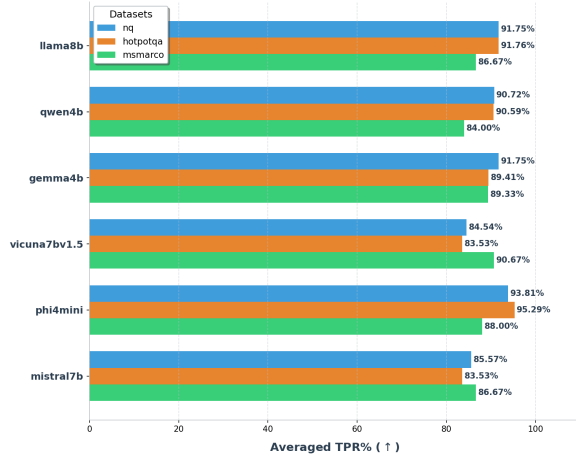


Figure 15: TPR results with $a_1 = 4$, $a_2 = 2$, $k_1 = 5$, $k_2 = 3$.

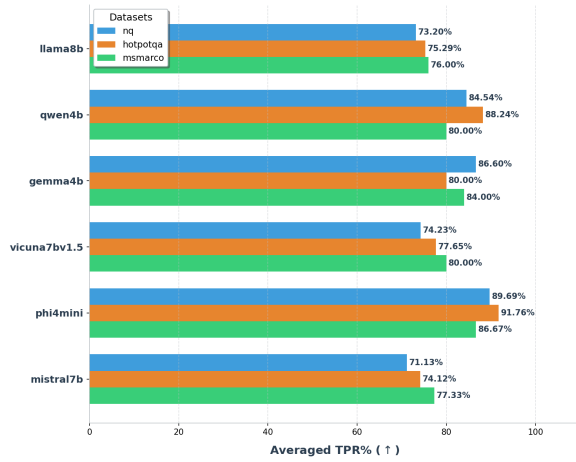


Figure 16: TPR results with $a_1 = 4$, $a_2 = 4$, $k_1 = 5$, $k_2 = 3$.

E.2 FPR

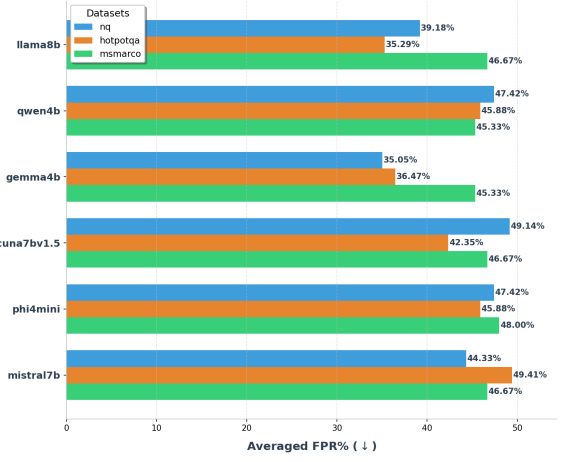


Figure 17: FPR results with 3 retrieved documents, $a_1 = 2$, $a_2 = 2$, $k_1 = 5$, $k_2 = 3$.

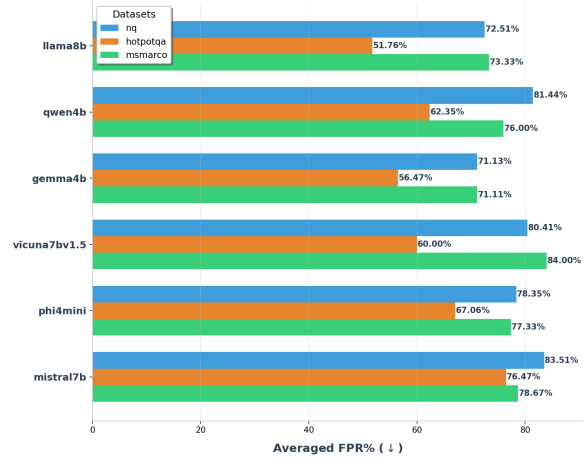


Figure 18: FPR results with $a_1 = 2$, $a_2 = 2$, $k_1 = 5$, $k_2 = 3$.

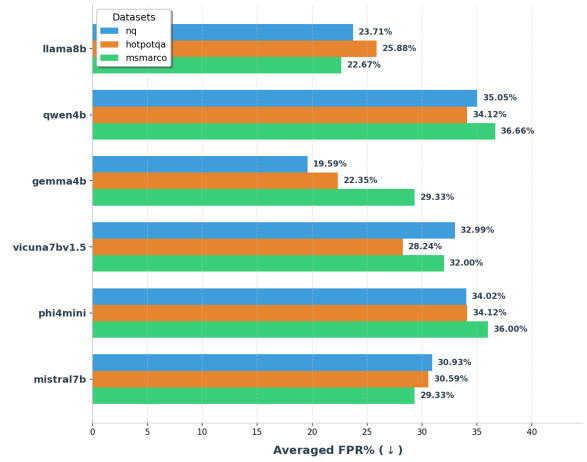


Figure 19: FPR results with $a_1 = 2$, $a_2 = 3$, $k_1 = 5$, $k_2 = 3$.

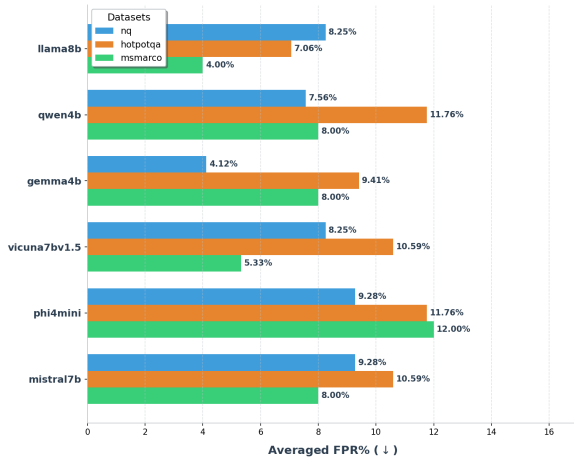


Figure 20: FPR results with $a_1 = 2$, $a_2 = 4$, $k_1 = 5$, $k_2 = 3$.

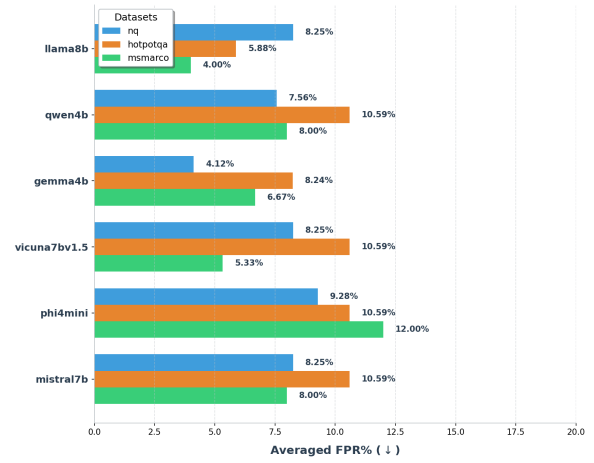


Figure 23: FPR results with $a_1 = 3$, $a_2 = 4$, $k_1 = 5$, $k_2 = 3$.

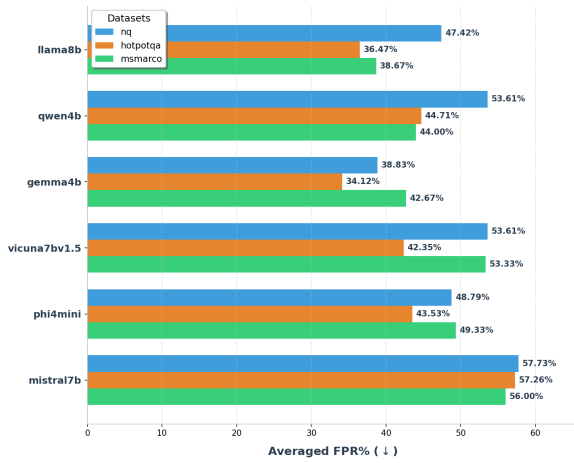


Figure 21: FPR results with $a_1 = 3$, $a_2 = 2$, $k_1 = 5$, $k_2 = 3$.

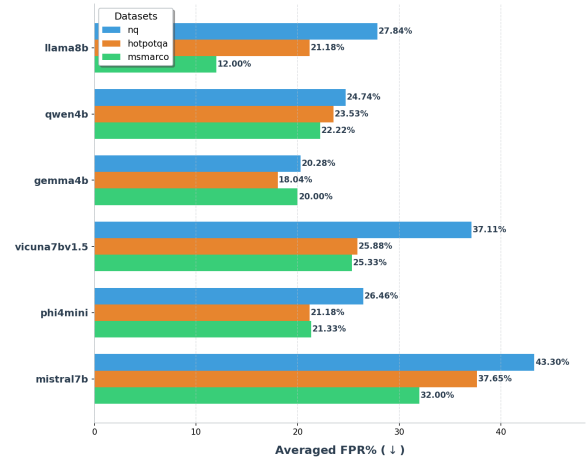


Figure 24: TPR results with $a_1 = 4$, $a_2 = 2$, $k_1 = 5$, $k_2 = 3$.

E.3 ACC

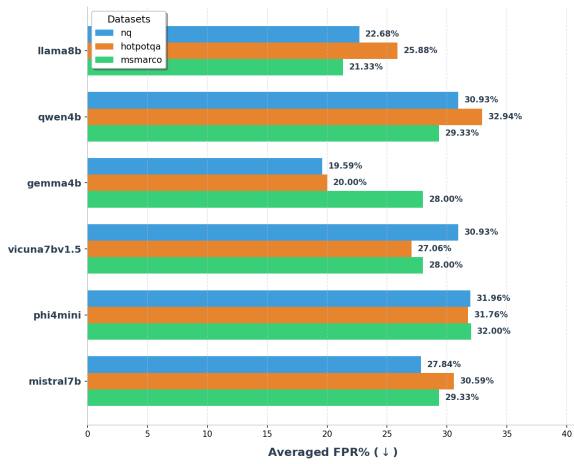


Figure 22: FPR results with $a_1 = 3$, $a_2 = 3$, $k_1 = 5$, $k_2 = 3$.

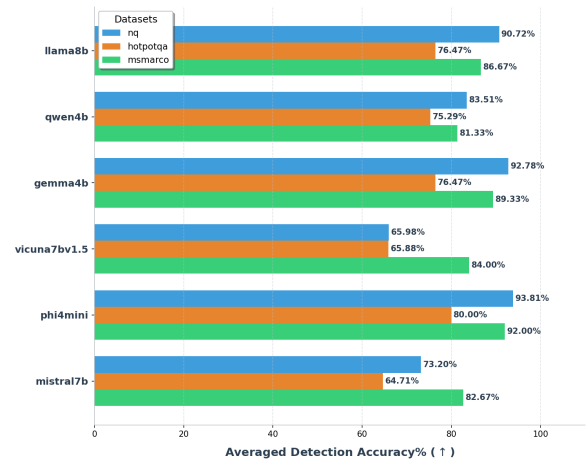


Figure 25: ACC results with 3 retrieved documents, $a_1 = 2$, $a_2 = 2$, $k_1 = 5$, $k_2 = 3$.

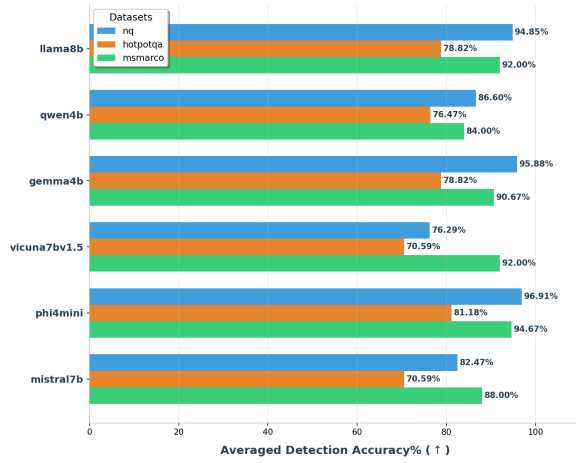


Figure 26: ACC results with $a_1 = 2, a_2 = 2, k_1 = 5, k_2 = 3$.

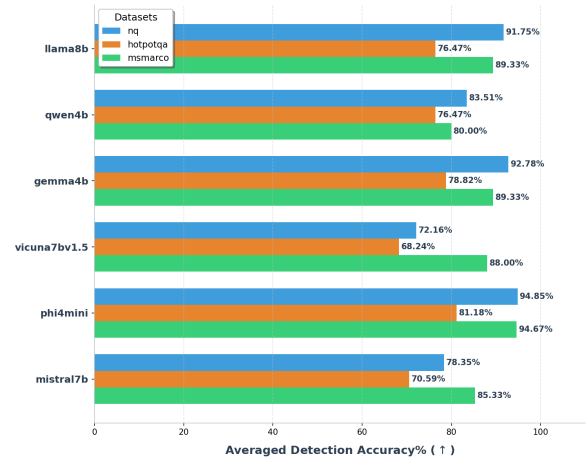


Figure 29: ACC results with $a_1 = 3, a_2 = 3, k_1 = 5, k_2 = 3$.

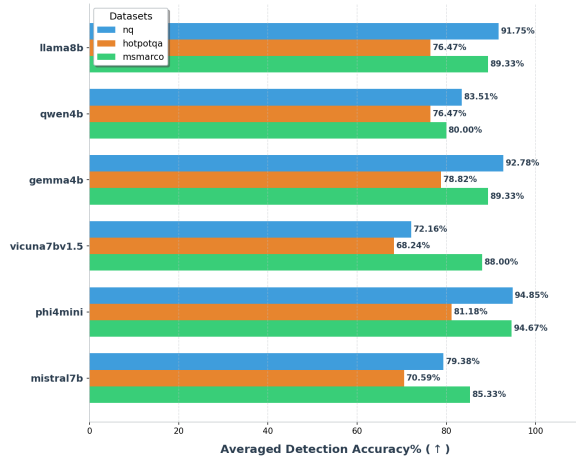


Figure 27: ACC results with $a_1 = 2, a_2 = 3, k_1 = 5, k_2 = 3$.

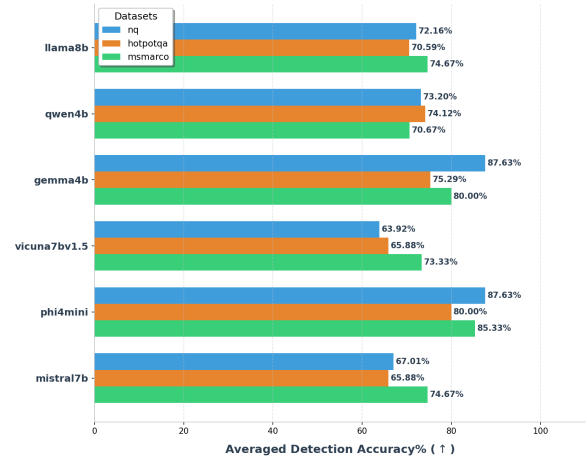


Figure 30: ACC results with $a_1 = 3, a_2 = 4, k_1 = 5, k_2 = 3$.

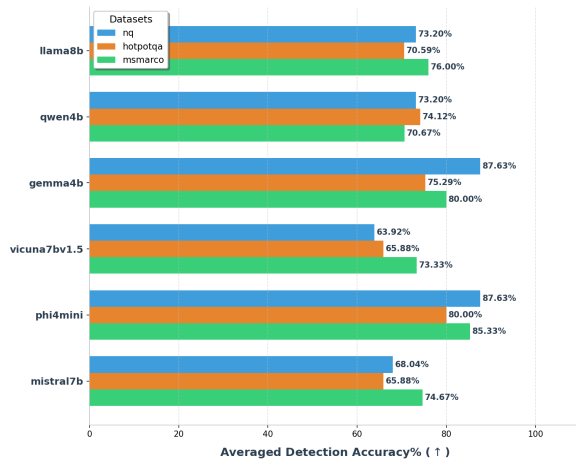


Figure 28: ACC results with $a_1 = 2, a_2 = 4, k_1 = 5, k_2 = 3$.

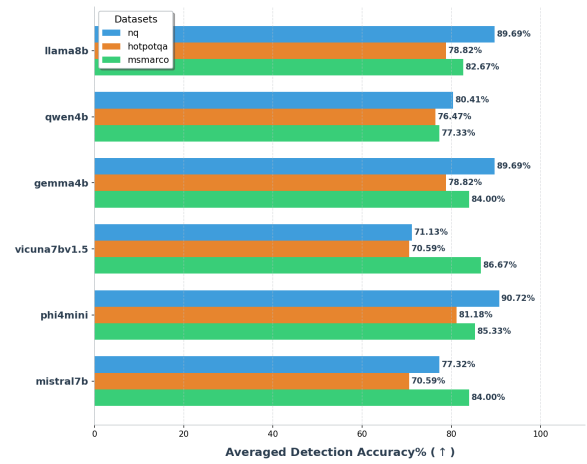


Figure 31: ACC results with $a_1 = 4, a_2 = 2, k_1 = 5, k_2 = 3$.

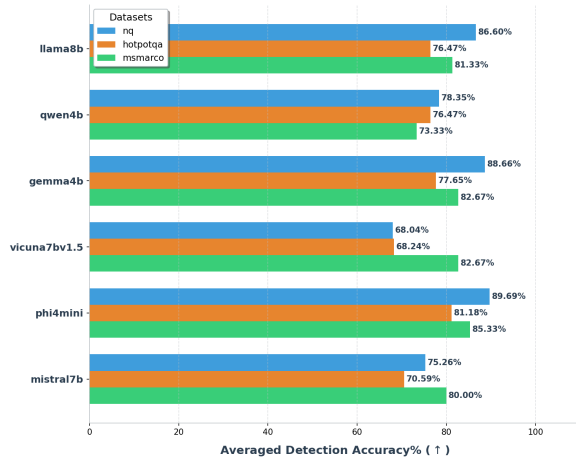


Figure 32: ACC results with $a_1 = 4$, $a_2 = 3$, $k_1 = 5$, $k_2 = 3$.

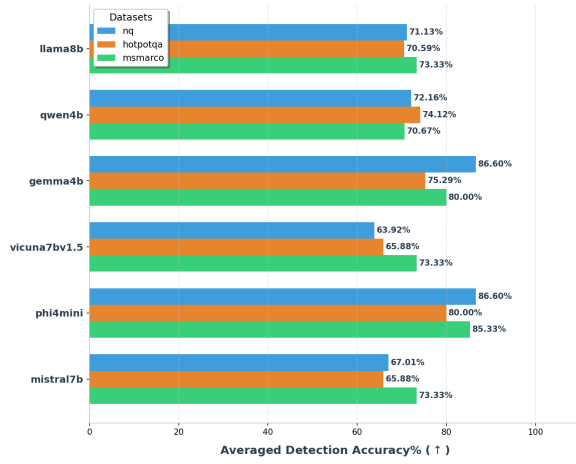


Figure 33: ACC results with $a_1 = 4$, $a_2 = 4$, $k_1 = 5$, $k_2 = 3$.