CtrlRAG: Black-box Document Poisoning Attacks for Retrieval-Augmented Generation of Large Language Models

Extended Abstract

Runqi Sui
Beijing University of Posts and Telecommunications
Beijing, China
srq1111@bupt.edu.cn

Xuejing Yuan
Beijing University of Posts and Telecommunications
Beijing, China
yuanxuejing@bupt.edu.cn

Di Tang
Sun Yat-sen University
Guangdong, China
tangd9@mail.sysu.edu.cn

Baojing Cui
Beijing University of Posts and Telecommunications
Beijing, China
cuibj@bupt.edu.cn

ABSTRACT

Retrieval-Augmented Generation (RAG) systems enhance response credibility and traceability by displaying reference contexts, but this transparency simultaneously introduces a novel black-box attack vector. Existing document poisoning attacks, where adversaries inject malicious documents into the knowledge base to manipulate RAG outputs, rely primarily on unrealistic white-box or gray-box assumptions, limiting their practical applicability. To address this gap, we propose CtrlRAG, a two-stage black-box attack that (1) constructs malicious documents containing misinformation or emotion-inducing content and injects them into the knowledge base, and (2) iteratively optimizes them using a localization algorithm and Masked Language Model (MLM) guided on reference context feedback, ensuring their retrieval priority while preserving linguistic naturalness.

KEYWORDS

Retrieval-Augmented Generation; LLMs; Black-box RAG Attacks

ACM Reference Format:
Runqi Sui, Xuejing Yuan, Di Tang, and Baojing Cui. 2026. CtrlRAG: Black-box Document Poisoning Attacks for Retrieval-Augmented Generation of Large Language Models: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/FMEO2393

1 INTRODUCTION

Retrieval-Augmented Generation (RAG) [2, 5, 10, 11] represents a significant advancement in natural language processing [8, 9, 12], combining the strengths of Large Language Models (LLMs) with external knowledge retrieval mechanisms. Despite this remarkable potential, as the scope of RAG application expands, researchers have increasingly focused on vulnerabilities within these systems, particularly document poisoning attacks [13, 14].

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/FMEO2393

Current research on attack methods primarily focuses on white-box or gray-box settings, where adversaries have access to model parameters or internal processes to optimize malicious documents [1, 3, 13]. However, realistic attack scenarios often involve attackers without access to the internal workings of the RAG system. The black-box attack method proposed by PoisonedRAG [14] generates static, one-shot malicious content by concatenating user queries to malicious documents to increase retrieval similarity. However, this approach lacks the ability to continuously optimize based on system feedback, causing a significant ceiling effect on the enhancement of malicious document retrieval similarity. Moreover, existing attack methods often produce malicious documents with detectable anomalies, such as unnatural language patterns or repetitive structures, which can be mitigated by basic filtering methods such as perplexity (PPL) [7] or pattern matching [6].

To develop practical and effective RAG attacks, we propose CtrlRAG, a black-box document poisoning attack that dynamically optimizes poisoning documents. Our approach begins by constructing an initial malicious document containing attack payloads such as misinformation or manipulation instructions tailored to the target question. The substitution localization algorithm is then applied, which analyzes system feedback to identify substitutable words in the document, strategically replacing them to increase the retrieval priority of the malicious document. Through multiple iterations, our method systematically extends the contextual coverage of malicious documents, guiding the RAG system to generate attacker-desired responses. Notably, we introduce a contextual replacement approach based on the Masked Language Model (MLM) [4], ensuring linguistic naturalness and eliminating anomalies in the poisoned document.

2 METHODOLOGY

We implement our attack by addressing three technical challenges: (1) Constructing high-quality initial malicious documents that can be effectively embedded into the reference context, (2) Designing a continuous optimization mechanism based on the reference context to overcome bottlenecks in retrieval similarity of crafted malicious content, and (3) Maintaining the linguistic naturalness of optimized malicious documents and ensuring that they do not exhibit obvious

anomalous features. To address these challenges, we propose a three-step attack approach which is shown in Algorithm 1.

Algorithm 1 *CtrlRAG (black-box)*

Input: Target question Q, adversarial objective O
Parameter: Top-k_p MLM predictions
Output: Malicious document W_m

1: **while** $rank(W)$ is *miss* **do**
2: $W \leftarrow$ Initialize(Q, O).
3: **end while**
4: $W = \{w_1, ..., w_n\}$
5: **for** w_i in W **do**
6: **if** $rank(W_{\setminus w_i}) \geq rank(W)$ **then**
7: $W_{mask} \leftarrow \{w_1, w_2, ..., [MASK]_i, ..., w_n\}$
8: $C_i \leftarrow MLM(i, W_{mask}, k_p)$
9: **end if**
10: **end for**
11: $C \leftarrow \prod_{i=1}^{n} C_i$
12: **for** c in C **do**
13: **if** O in c and $rank(c)$ is Current_Best **then**
14: $W_m \leftarrow c$
15: **end if**
16: **end for**
17: **return** $W_m \leftarrow None$

***Step* I**: *Initialize a malicious document based on the user query and attack target.* We utilize the transparent reference context as the attack vector. However, when the injection of malicious content does not lead to changes in the reference context, the system lacks measurable metrics for subsequent optimization. Therefore, the initial malicious document must meet the condition that it possesses sufficient retrieval similarity to be embedded in the reference context, serving as the baseline document for subsequent optimization. Notably, the ranking of the initial document in the reference context does not affect the overall effectiveness of the attack.

In this phase (lines 1-3 in Algorithm 1), we use prompt engineering to establish automated generation methods for different attack goals: *Hallucination Amplification* and *Emotion Manipulation*. Prompt templates are shown at *https://arxiv.org/abs/2503.06950*.

***Step* II**: *Identify substitutable words by analyzing document rankings after injection into the knowledge base.* Based on the initial malicious document, we introduce the *Feedback-driven Substitution Localization* approach (lines 4-6 in Algorithm 1) to determine the substitutable features of each word in a document.

Specifically, given a sentence with n words, $W = \{w_1, w_2, ..., w_n\}$, certain words may adversely affect the retrieval similarity, making them viable candidates for substitution. We define a binary substitutability metric S_{w_i} to quantify the impact of a word w_i on the similarity score. Let $W_{\setminus w_i}$ denote the sentence after removing w_i, i.e., $W_{\setminus w_i} = \{w_1, ..., w_{i-1}, w_{i+1}, ..., w_n\}$. We compute the similarity score between the modified sentence $W_{\setminus w_i}$ and the query Q, then compare it with the original similarity score $Sim(Q, W)$. The substitutability of w_i is formally defined as:

$$S_{w_i} = \begin{cases} 1, & \text{if } Sim(Q, W_{\setminus w_i}) \geq Sim(Q, W), \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Here, $S_{w_i} = 1$ indicates that w_i is substitutable, while $S_{w_i} = 0$ means that w_i is non-substitutable.

Black-box Setting. In black-box settings where similarity scores and model parameters are inaccessible as attack vectors, we utilize the reference context as an attack vector to indirectly compute S_{w_i} in Equation 1. We simultaneously inject both the original sentence W and the modified sentence $W_{\setminus w_i}$ into the knowledge base. By analyzing their relative rankings within the retrieved context, we infer the substitutability of each word. If removing w_i does not degrade the ranking of $W_{\setminus w_i}$ compared to W, then w_i is considered substitutable ($S_{w_i} = 1$). Otherwise, it is retained ($S_{w_i} = 0$). This can be translated as follows:

$$S_{w_i} = \begin{cases} 1, & \text{if } Rank(W_{\setminus w_i}) \geq Rank(W), \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

***Step* III**: *Perform word replacement using MLM-based perturbation methodology.* Following substitution localization, we propose an MLM-based perturbation method (lines 7-11 in Algorithm 1). Specifically, for each word $w_i \in W$ identified as substitutable, we apply a masked word substitution mechanism which masks w_i, $W_{mask} = \{w_1, ..., [MASK]_i, ..., w_n\}$, and employs a pre-trained MLM, such as BERT [4], to predict suitable replacements for the **[MASK]** token. For each masked position, we extract the top-k_p predictions from the MLM and construct a candidate pool C by computing the Cartesian product of these predictions across all substitutable positions. This can be formalized as:

$$C = \prod_{i=1}^{n} MLM(i, W_{mask}, k_p), \quad \text{s.t.} \quad S_{w_i} = 1. \tag{3}$$

The optimal substitution scheme is one that maximizes the ranking of the perturbed document within the reference context. Additionally, we impose an additional constraint (lines 12-17 in Algorithm 1): the modified document must maintain the original misinformation or emotion manipulation instruction, ensuring the effectiveness of the attack.

Extreme black-box scenario: *The adversary observes only the reference context and has no access to retrieval priority information.* In this setting, we replace the ranking-based criterion with a *hit-driven* principle while preserving the overall pipeline of *Feedback-driven Substitution Localization* and *MLM-based Contextual Perturbations*. Let the hit indicator be:

$$H(W) = \begin{cases} 1, & \text{if } W \text{ is retrieved by the RAG system,} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The attacker's objective becomes:

$$\underset{W'}{\text{minimize}} |W' \ominus W| \quad \text{s.t.} \quad H(W') = 1, \tag{5}$$

where $W' \ominus W$ denotes the symmetric difference between W' and W. More details are shown at *https://arxiv.org/abs/2503.06950*.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. 62202064) and the Doctoral Student Innovation Foundation of Beijing University of Posts and Telecommunications (Grant No. CX20241056).

REFERENCES

[1] Harsh Chaudhari, Giorgio Severi, John Abascal, Matthew Jagielski, Christopher A. Choquette-Choo, Milad Nasr, Cristina Nita-Rotaru, and Alina Oprea. 2024. Phantom: General Trigger Attacks on Retrieval Augmented Language Generation. arXiv:2405.20485 [cs.CR] https://arxiv.org/abs/2405.20485

[2] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 17754–17762.

[3] 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. arXiv:2405.13401 [cs.CR] https://arxiv.org/abs/2405.13401

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics*. 4171–4186.

[5] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997 [cs.CL] https://arxiv.org/abs/2312.10997

[6] Tony Hak and Jan Dul. 2009. *Pattern matching*. Technical Report.

[7] Frederick Jelinek. 1980. Interpolated estimation of Markov source parameters from sparse data. In *Proc. Workshop on Pattern Recognition in Practice, 1980*.

[8] Mohammed Khaliq, Paul Chang, Mingyang Ma, Bernhard Pflugfelder, and Filip Miletić. 2024. RAGAR, Your Falsehood Radar: RAG-Augmented Reasoning for Political Fact-Checking using Multimodal Large Language Models. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*. 280–296.

[9] Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-Augmented Dialogue Generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 8460–8478. https://doi.org/10.18653/v1/2022.acl-long.579

[10] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[11] Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. 2025. From matching to generation: A survey on generative information retrieval. *ACM Transactions on Information Systems* 43, 3 (2025), 1–62.

[12] Michael H Prince, Henry Chan, Aikaterini Vriza, Tao Zhou, Varuni K Sastry, Yanqi Luo, Matthew T Dearing, Ross J Harder, Rama K Vasudevan, and Mathew J Cherukara. 2024. Opportunities for retrieval and tool augmented large language models in scientific facilities. *npj Computational Materials* 10, 1 (2024), 251.

[13] 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*. 1610–1626.

[14] Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. 2024. PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models. arXiv:2402.07867 [cs.CR] https://arxiv.org/abs/2402.07867