

Towards Unsupervised Adversarial Document Detection in Retrieval Augmented Generation Systems

Patrick Levi 

Department of Electrical Engineering, Media, and Computer Science
Ostbayerische Technische Hochschule Amberg-Weiden
Amberg, Germany
e-mail: p.levi@oth-aw.de

Abstract—Retrieval augmented generation systems have become an integral part of everyday life. Whether in internet search engines, email systems, or service chatbots, these systems are based on context retrieval and answer generation with large language models. With their spread, also the security vulnerabilities increase. Attackers become increasingly focused on these systems and various hacking approaches are developed. Manipulating the context documents is a way to persist attacks and make them affect all users. Therefore, detecting compromised, adversarial context documents early is crucial for security. While supervised approaches require a large amount of labeled adversarial contexts, we propose an unsupervised approach, being able to detect also zero day attacks. We conduct a preliminary study to show appropriate indicators for adversarial contexts. For that purpose generator activations, output embeddings, and an entropy-based uncertainty measure turn out as suitable, complementary quantities. With an elementary statistical outlier detection, we propose and compare their detection abilities. Furthermore, we show that the target prompt, which the attacker wants to manipulate, is not required for a successful detection. Moreover, our results indicate that a simple context summary generation might even be superior in finding manipulated contexts.

Keywords—RAG security; chatbot security; intrusion detection; adversarial attack.

I. INTRODUCTION

Over the last few years, Retrieval Augmented Generation (RAG) systems [1] have become a valuable and indispensable tool in our lives, in particular as a support for knowledge intensive activities. Almost every current chatbot is conceptualized as a RAG system, ranging from internet search engines over email bots to customer service chatbots. RAG systems combine the user query (prompt) with a document database, which contains potential supporting information. A retriever component finds documents that provide information regarding the prompt (context), while the generator Large Language Model (LLM) creates an answer to the prompt based on these contexts. Thus, RAG systems efficiently combine the power of LLMs with large information collections. Due to their powers, they also attract attackers, offering potentially high returns: The information database may contain private or confidential information which can be extract, the RAG system can be tricked into spreading biased or false information, or it can be sabotaged into shutdown. To reach their goal, hackers could attack the prompt directly, but usually they sneak Adversarial Contexts (ACs) into the context document database. This way, the attack becomes persistent and potentially effects all users. Often, the context

database is at least partly accessible to attackers, e.g., for email bots: An attacker can add a document by sending an email to the victim. Therefore, ACs are a serious security threat for RAG systems. While defenses exist, these defenses might imply a huge effort and thus a downgrade of the system usability (e.g., human in the loop). Adversarial mechanism are basically those to jailbreak LLMs in general [2][3], which are continuously adapted and improved. Therefore, the best strategy against ACs is detecting them as early as possible and preventing the attack entirely. Detection has to account for continuously new (zero day) attacks. Many approaches exist, to differentiate between adversarial and non-adversarial documents, e.g., leveraging the power of supervised learning [4]. However, supervised learning requires large amounts of labeled documents and usually only detects attacks similar to those learned from the data. These effects are well known from network Intrusion Detection (ID) [5]. Similar to ID applications, we want to switch from supervised to unsupervised approaches, designing an Adversarial Context Detector (ACD), which is flexible, robust and does not require knowledge about the specific attack type, nor much labeled AC examples.

The development of our approach is currently work in progress, however, we present a promising initial study answering two main questions. First, which quantities indicate ACs reliably enough for an ACD. Second, is it necessary to know the target question (prompt), which the attacker intended to manipulate. We use a simple statistical outlier detection to answer these questions, thus showing the overall feasibility of unsupervised ACD. Our paper is organized as follows: We summarize related work in Section II, present our statistical approach in Section III, and propose experiments on an adversarial dataset in Section IV. We evaluate our results on this dataset in Section V. In Section VI we conclude on the feasibility of ACD with our indicator variables and outline future work.

II. RELATED WORK

Various attacks on RAG system have been investigated recently by researchers and ethical hackers alike. Some attacks target private or confidential information in the context documents by either performing membership inference attacks [6] or even extract whole documents from the database [7] using adversarial prompts. Other approaches use adversarial documents to cause the RAG system to reveal whole documents.

In the case of an email agent with the permission to send emails, this has been shown to extend even into a self-replicating worm-like attack [8]. Attackers who want the RAG system to spread false information usually add poisoned context documents [9]. Dedicated context poisoning can also trick safety alignment in LLMs to cause refusing answers, thus generating a denial-of-service attack [10], which is not easy to detect. The techniques used in the prompt as well as in adversarial documents are the same as for LLM jailbreaking [4]. ACs are usually created, using oracle based text manipulation, targeted whitebox optimizations [11] or heuristic optimization methods [12]. Often, attacks need a smart combination of tricking the retriever, as well as the generator components to achieve their goal efficiently [13]. Recently, various works have been published to detect adversarial attacks against LLMs. Layer activations have been shown to be successful indicators for adversarial attacks and even be useful for classification of attack types [14]. Smart supervised approaches have been applied to detect adversarial attacks, dealing with the problem of few examples and small amounts of labeled data [15]. We plan to extend this research to unsupervised approaches using anomaly detection to find adversarial contexts, as has been done in adversarial image detection [16].

III. METHOD

Our ACD feasibility study is based on the following threat model. The hacker intends to poison a specific question or a group of questions with targeted adversarial documents. The defender is not aware of the target question, but needs to detect the attack by screening the contexts. The attacker can add context documents, which is realistic for common applications using internet contexts or emails. These contexts might be found by the retriever and thus find their way into the generator’s context. The defender must detect the attack only based on the contexts. We assume they have three potential knowledge levels: They can query the RAG generator arbitrarily and have access to the text output, eventually also to the corresponding logits, or even to the generator LLM layer activations. We assess the capability of our ACD depending on these knowledge levels. Generator activations are used directly as an adversarial attack indicator, where we restrict ourselves to the last layer activations. From the logits, we compute the TokenSAR score [17], an entropy based measure [18], which has been proven to be successful for jailbreak detection [19]. The generator output text is encoded into a 768-dimensional embedding vectors using a variant of the MPNet model [20][21]. We chose this model since it is small and thus cheap to deploy and operate. To realize our ACD, we want to work without the target prompt. Therefore, we use a simple summary prompt asking the generator model to summarize available context information (“Summarize the following context documents: <contexts>. Consider every important aspect in your summary.”). We evaluate activations, TokenSAR, and embeddings for the related answer. We consider these quantities obtained for N combinations of non-adversarial contexts as a reference and compute the mean value and standard deviation for the N TokenSAR values. For the N

activations and embeddings, respectively, we first compute their center and the Euclidean distance of each individual vector to it. Then, we consider the mean distance and its standard deviation. We repeat the summary prompt, replacing at least one context with an AC, and extract the same quantities from the generator LLM outputs. For activations and embeddings, we consider Euclidean distances from the previously computed center. Subsequently, we apply Grubb’s test at $\alpha = 0.1$ to determine whether any of the quantities q_{adv} (TokenSAR, embeddings, activations) obtained with the ACs is an outlier with respect to the reference values:

$$q_{adv} \notin [\mu - G_{crit}; \mu + G_{crit}] \quad (1)$$

$$G_{crit} = \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2}{N-t^2}} \quad (2)$$

with μ, s being the empirical mean and standard deviation of the reference values and $t = t_{1-\frac{\alpha}{2N}, N-2}$ being the two-sided quantile of Student’s t-distribution. We implicitly assume normal distribution of the quantities, being aware of the interdependence due to the center distances. This way, we have a semi-supervised method, requiring only valid contexts to obtain the reference values.

IV. EXPERIMENTS

Our experiments are performed based on 100 questions from the HotpotQA dataset [22] with adversarial contexts from PoisonedRAG [4][23]. Every HotpotQA question consists of, among other things, a question, related context documents from Wikipedia, and a correct answer. PoisonedRAG adds adversarial contexts and a target incorrect answer. For each question in the dataset, we conduct a small study using $N = 10$ combinations of valid contexts. To simulate the retriever component, we select $k = 5$ valid contexts from the original set. The RAG generator is realized with a Llama-3.1 8B model [24]. Using the summary prompt introduced in Section III, each context combination is summarized. The output is processed to obtain its embeddings, TokenSAR values and the activations (see Figure 1). In addition, the generator LLM is prompted with the question and the contexts (question prompt: “Answer the following question based on the provided context: Question: <question>. Contexts: <contexts>.”) to verify that the contexts are valid and informative enough for the generator to be able to answer the question correctly. The answer is verified to be semantically equivalent to the documented correct answer from the dataset using Mistral-7B [25] as a judge. If the answer cannot be verified as correct, it is discarded, leading eventually to less than $N = 10$ valid contexts. In case there are too few valid contexts, the question is discarded.

In the same way as for the summaries, also the answers to the question prompt are processed to obtain embeddings, TokenSAR values and the last layer activations are extracted. A comparison between question and summary will show the impact of not knowing the target prompt. Afterwards, each question is evaluated using the same prompts but introducing AC to the contexts. The $k = 5$ valid contexts are successively replaced by AC, starting with the first one

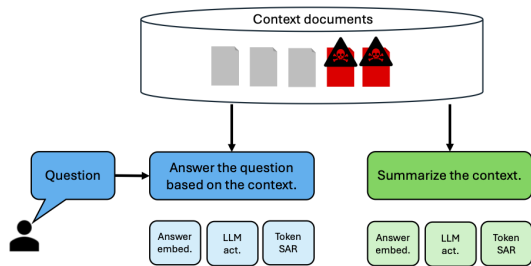


Figure 1. Schematic diagram of our approach: The question prompt requires the user question and the contexts, while the simpler summary prompt only requires the contexts.

until more than half the contexts are AC. This way, we cover various degrees of manipulations in our study.

V. RESULTS AND DISCUSSION

A. Results

We ran one experiment on HotpotQA dataset, using both, the summary prompt as well as the question prompt. Comparing both results we will see the effect of knowing the targeted question or instruction. Table I summarizes how many cases using at least one AC can be detected by our ACD. There are up to 30 such context combinations per question containing at least one AC.

TABLE I. QUANTITATIVE RESULTS FOR THE DIFFERENT INDICATORS AND PROMPT TYPES (QUESTION OR SUMMARY)

Quantity	Summary	Question
Invalid	11	11
Undetected	486	486
<i>Detected by</i>		
TokenSAR + Emb. only	60	60
TokenSAR + Act. only	38	22
TokenSAR + Emb. + Act.	1912	1928
TokenSAR total	2010	2010
Emb. total	1972	1988
Act. total	1950	1950

Questions which did not reveal a sufficient number of correct answers for valid contexts are marked as invalid. For all others we count, how many AC cases are exclusively detected by any combination of our indicators. Moreover, we count how often each indicator is successful. The number of detected AC cases were counted, independent of attack success.

TokenSARs turn out to have the highest predictive capability, followed by activations and embeddings. Clearly, the summary is equally suited for ACD as the question prompt. None of the indicator appears to be dominant over the others. Every attack is detectable by at least two indicators, most attacks even by all three of them. For both prompts, about 19.5 % (486 of 2496) of attacks remain undetected.

B. Discussion

The results are promising towards ACD for two reasons: First, it turned out that it is not necessary for the defender

to know the question the attacker tried to manipulate. Using a summary prompt turned out sufficient to detect AC. We interpret this as follows: When we increased the number of AC, we conclude that that with few AC contradictions to valid contexts might be found in our indicators, while for increasing number of AC, the summary is different enough to be an outlier compared to the valid context summary. These effects could well be reflected in the (last layer) activations of the generator. Further, they influence the summary text, since a specific statement either does not occur in it or it is weakened to resolve the controversy. In the TokenSAR, AC are visible due to a higher uncertainty in the answer logits of the generator LLM, resulting in a higher entropy.

Second, our results show that several indicators are able to detect manipulations. Embeddings and TokenSAR values are the most successful ones. While activations require no extra computation overhead, they might not be available, especially for proprietary models and APIs. Since the TokenSARs are based on the output logits, they are easy to compute and often available. Output embeddings are always available. The computational effort can be reduced by using a small embedding model.

VI. CONCLUSION AND FUTURE WORK

Detection of adversarial attacks against RAG systems is highly relevant for the future of chatbots and other systems. Attackers have two attack paths, the prompt and the context documents. The latter are especially exposed in important applications, like email bots. Attacks against the context documents need to be reliably prevented, since they lead to persistent misbehavior of the system. Therefore, reliable ACD methods are needed. In contrast to existing supervised approaches, we conducted a preliminary study to look into unsupervised detection methods. Using a simple statistical approach, we successfully identified three indicator quantities that are well suited for ACD: the generator LLM activations, output embeddings, and entropy-based TokenSAR. Systematic comparison of these indicators revealed that they show a high redundancy, but also complement each other in some cases. Furthermore, we found that ACD can be realized without knowing the specific prompt (question, instruction, etc.) that is targeted by the attacker. A simple summary instruction is sufficient. Therefore, the goal of this small, preliminary study, to elucidate promising approaches for unsupervised ACD has been achieved. Future research shall focus on the following key aspects. First, increase the experimental runs to obtain more normal contexts, while this study used a relatively small number. With more samples, a proper quantitative evaluation of false positives (valid contexts incorrectly detected as AC) will be possible, too. Furthermore, the approach needs to be extended to more RAG attacks beyond PoisonedRAG [4]. In particular, corpus poisoning [9], BADRAG [13], and GARAG [12] shall be considered. Finally, more sophisticated anomaly detection methods shall be used, following previous work from network intrusion detection [5].

REFERENCES

- [1] P. Lewis *et al.*, “Retrieval-augmented generation for knowledge-intensive nlp tasks,” in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS ’20, Vancouver, BC, Canada: Curran Associates Inc., 2020, ISBN: 9781713829546.
- [2] Y. Liu *et al.*, *Jailbreaking ChatGPT via prompt engineering: An empirical study*, version 1, 2023. DOI: 10.48550/arXiv.2305.13860. arXiv: 2305.13860 [cs.SE].
- [3] P. Levi and C. Neumann, “Goal hijacking using adversarial vocabulary for attacking vulnerabilities of large language model applications,” vol. 17, pp. 214–225, Dec. 2024. DOI: 10.5281/zenodo.14680185.
- [4] W. Zou, R. Geng, B. Wang, and J. Jia, “PoisonedRAG: Knowledge corruption attacks to retrieval-augmented generation of large language models,” in *Proceedings of the 34th USENIX Conference on Security Symposium*, ser. SEC ’25, Seattle, WA, USA: USENIX Association, 2025.
- [5] E. M. Cabeza-Lopez, R. Ruiz-Gonzalez, A. Merino-Gomez, L. E. Curiel-Herrera, and J. A. Rincon, “A Comparison of AI-Enabled Techniques for the Detection of Attacks in IoT Devices,” in *International Joint Conferences*, H. Quintián *et al.*, Eds., vol. 957, Series Title: Lecture Notes in Networks and Systems, Cham: Springer Nature Switzerland, 2024, pp. 227–236. DOI: 10.1007/978-3-031-75016-8_21. [Online]. Available: https://link.springer.com/10.1007/978-3-031-75016-8_21.
- [6] Y. Choi, Y. Park, J. Byun, J. Lee, and J. Park, “Safeguarding privacy of retrieval data against membership inference attacks: Is this query too close to home?” In *Findings of the Association for Computational Linguistics: EMNLP 2025*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds., Suzhou, China: Association for Computational Linguistics, Nov. 2025, pp. 8241–8258. DOI: 10.18653/v1/2025.findings-emnlp.438. [Online]. Available: <https://aclanthology.org/2025.findings-emnlp.438/>.
- [7] S. Zeng *et al.*, “The good and the bad: Exploring privacy issues in retrieval-augmented generation (RAG),” in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 4505–4524. DOI: 10.18653/v1/2024.findings-acl.267. [Online]. Available: <https://aclanthology.org/2024.findings-acl.267/>.
- [8] S. Cohen, R. Bitton, and B. Nassi, “Here comes the AI worm: Preventing the propagation of adversarial self-replicating prompts within GenAI ecosystems,” in *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’25, Taipei, Taiwan: Association for Computing Machinery, 2025, pp. 3975–3989. DOI: 10.1145/3719027.3765196. [Online]. Available: <https://doi.org/10.1145/3719027.3765196>.
- [9] Z. Zhong, Z. Huang, A. Wettig, and D. Chen, “Poisoning retrieval corpora by injecting adversarial passages,” in *Empirical Methods in Natural Language Processing (EMNLP)*, 2023.
- [10] A. Shafraan, R. Schuster, and V. Shmatikov, *Machine against the RAG: Jamming retrieval-augmented generation with blocker documents*, version 4, 2025. arXiv: 2406.05870 [cs.CR]. [Online]. Available: <https://arxiv.org/abs/2406.05870>.
- [11] A. Zou *et al.*, *Universal and transferable adversarial attacks on aligned language models*, version 2, 2023. DOI: <https://doi.org/10.48550/arXiv.2307.15043>. arXiv: 2307.15043 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2307.15043>.
- [12] S. Cho, S. Jeong, J. Seo, T. Hwang, and J. C. Park, *Typos that broke the RAG’s back: Genetic attack on RAG pipeline by simulating documents in the wild via low-level perturbations*, version 1, 2024. DOI: <https://doi.org/10.48550/arXiv.2404.13948>. arXiv: 2404.13948 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2404.13948>.
- [13] J. Xue *et al.*, *Badrag: Identifying vulnerabilities in retrieval augmented generation of large language models*, version 2, 2024. DOI: <https://doi.org/10.48550/arXiv.2406.00083>. arXiv: 2406.00083 [cs.CR]. [Online]. Available: <https://arxiv.org/abs/2406.00083>.
- [14] S. Ball, F. Kreuter, and N. Panickssery, *Understanding Jailbreak Success: A Study of Latent Space Dynamics in Large Language Models*, Oct. 2024. DOI: 10.48550/arXiv.2406.09289. [Online]. Available: <http://arxiv.org/abs/2406.09289>.
- [15] X. Tan *et al.*, “RevPRAG: Revealing poisoning attacks in retrieval-augmented generation through LLM activation analysis,” in *Findings of the Association for Computational Linguistics: EMNLP 2025*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds., Suzhou, China: Association for Computational Linguistics, Nov. 2025, pp. 12999–13011. DOI: 10.18653/v1/2025.findings-emnlp.698. [Online]. Available: <https://aclanthology.org/2025.findings-emnlp.698/>.
- [16] H. Liu, B. Zhao, J. Guo, K. Zhang, and P. Liu, “A lightweight unsupervised adversarial detector based on autoencoder and isolation forest,” *Pattern Recognition*, vol. 147, p. 110127, 2024, ISSN: 0031-3203. DOI: <https://doi.org/10.1016/j.patcog.2023.110127>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320323008245>.
- [17] J. Duan *et al.*, “Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 5050–5063. DOI: 10.18653/v1/2024.acl-long.276. [Online]. Available: <https://aclanthology.org/2024.acl-long.276/>.
- [18] S. Steindl, U. Schäfer, B. Ludwig, and P. Levi, “Linguistic obfuscation attacks and large language model uncertainty,” in *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertainNLP 2024)*, R. Vázquez *et al.*, Eds., St Julians, Malta: Association for Computational Linguistics, Mar. 2024, pp. 35–40. [Online]. Available: <https://aclanthology.org/2024.uncertainlp-1.4>.
- [19] F. Rubenbauer, S. Steindl, P. Levi, D. Loebenberger, and U. Schäfer, “Detection of adversarial prompts with model predictive entropy,” in *Findings of the Association for Computational Linguistics: EACL 2026*, Accepted for publication, Rabat, Morocco: Association for Computational Linguistics, 2026.
- [20] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, *Mpnet: Masked and permuted pre-training for language understanding*, version 2, 2020. arXiv: 2004.09297 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2004.09297>.
- [21] Huggingface, *All-mpnet-base-v2*, 2026.03.10. [Online]. Available: <https://huggingface.co/sentence-transformers/all-mpnet-base-v2>.
- [22] Z. Yang *et al.*, “HotpotQA: A dataset for diverse, explainable multi-hop question answering,” in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
- [23] W. Zou, R. Geng, B. Wang, and J. Jia, *PoisonedRAG*, 2026.03.10. [Online]. Available: https://github.com/sleepeer/PoisonedRAG/blob/main/results/adv_targeted_results/hotpotqa.json.
- [24] A. Grattafiori *et al.*, *The Llama 3 herd of models*, version 3, 2024. arXiv: 2407.21783 [cs.AI]. [Online]. Available: <https://arxiv.org/abs/2407.21783>.
- [25] A. Q. Jiang *et al.*, *Mistral 7B*, version 1, 2023. arXiv: 2310.06825 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2310.06825>.