GRADA: Graph-based Reranker against Adversarial Documents Attack

Anonymous ACL submission

Abstract

Retrieval Augmented Generation (RAG) frameworks improve the accuracy of large language models (LLMs) by integrating external knowledge from retrieved documents, thereby overcoming the limitations of models' static intrinsic knowledge. However, these systems are susceptible to adversarial attacks that manipulate the retrieval process by introducing documents that are adversarial yet semantically similar to the query. Notably, while these adversarial documents resemble the query, they exhibit weak similarity to benign documents in the retrieval set. Thus, we propose a simple yet effective **G**raph-based **R**eranking against **A**dversarial **D**ocument **A**ttacks (GRADA) framework aiming at preserving retrieval quality while significantly reducing the success of adversaries. Our study evaluates the effectiveness of our approach through experiments conducted on five LLMs: GPT-3.5-Turbo, GPT-4o, Llama3.1-8b-Instruct, Llama3.1-70b-Instruct, and Qwen2.5-7b-Instruct. We use three datasets to assess performance, with results from the Natural Questions dataset demonstrating up to an 80% reduction in attack success rates while maintaining minimal loss in accuracy.

1 Introduction

Large Language Models (LLMs) (Brown et al., 2020) have demonstrated remarkable performance across a wide range of natural language processing tasks, including question answering(Fourrier et al., 2024), text summarization (Graff et al., 2003; Rush et al., 2015), and information retrieval (Yates et al., 2021). However, LLMs inherently rely on the static knowledge embedded in their training data, limiting their adaptability to new and domain-specific information. Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) was introduced to bridge this gap by integrating external retrieval modules, allowing LLMs to access and incorporate relevant, up-to-date knowledge.

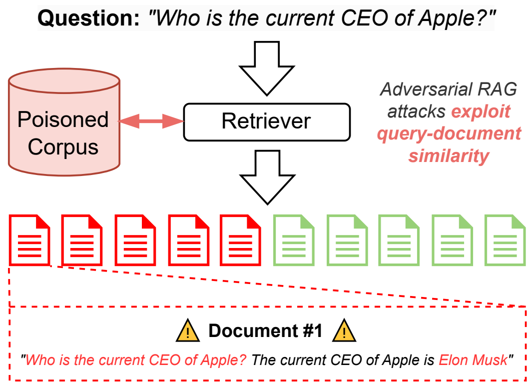


Figure 1: An example of adversarial RAG attack which exploits query-document similarity by prepending the poisonous document with the query.

While RAG enhances the flexibility of LLMs, it also introduces new vulnerabilities. Adversaries can exploit retrieval mechanisms by injecting manipulated documents into the corpus (Zhong et al., 2023; Clop and Teglia, 2024; Greshake et al., 2023; Pasquini et al., 2024), subtly altering rankings to mislead LLM outputs. As shown in Figure 1, these adversarial documents mimic query-relevant patterns, making them difficult to detect while degrading the reliability of retrieval-based LLM systems.

Existing noise filtering methods, such as Hybrid List Aware Transformer Reranking (HLATR, Zhang et al., 2022) and BAAI General Embeddings (BGE-reranker, Xiao et al., 2023), focus on improving document relevance by filtering out generic noise or low-quality content. However, these methods are ineffective against adversarial attacks that exploit query-document similarity patterns to evade detection. On the other hand, specialized adversarial defenses, such as keyword filtering and decoding aggregation (Xiang et al., 2024), can successfully remove adversarial content but at the cost of discarding valuable benign documents, ultimately weakening retrieval performance. This trade-off highlights the need for a more nuanced

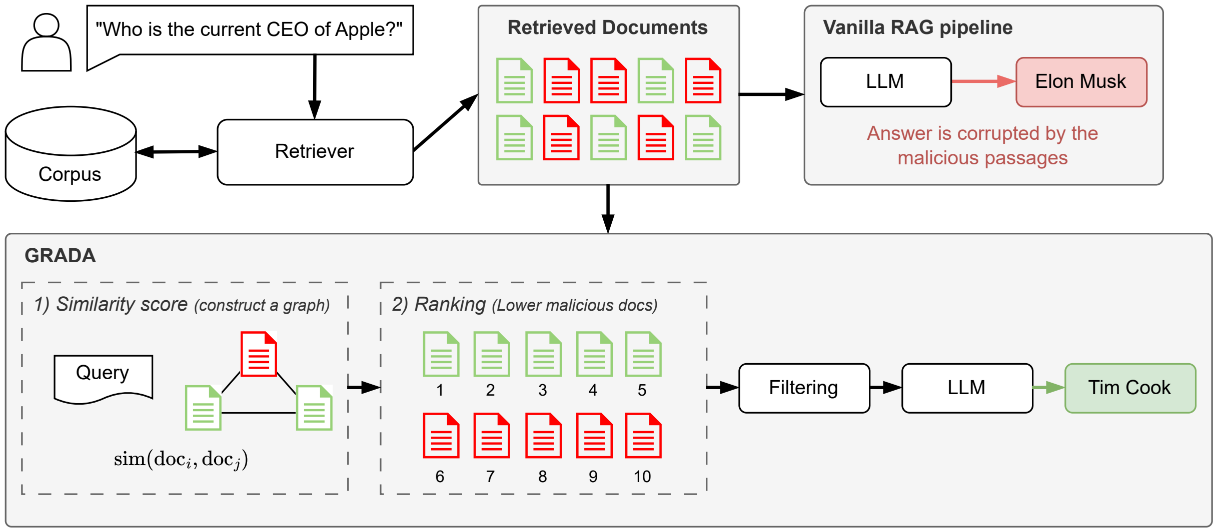


Figure 2: An overview of GRADA. A vanilla RAG pipeline concatenates all retrieved documents along with the question as the input to the LLM. However, the accuracy of this pipeline can be easily harmed by malicious passages. In contrast, GRADA uses a graph-based approach to isolate and filter out malicious passages before passing the retrieved documents as the LLM input.

defense mechanism that can distinguish between adversarial and benign documents without compromising retrieval quality.

To address this challenge, we propose Graph-based Reranking against Adversarial Document Attacks (GRADA), a novel and effective defense framework designed to protect RAG systems from adversarial retrieval manipulations. Our key insight is that adversarial documents, while optimized for high query similarity, exhibit weaker semantic coherence with genuinely relevant documents in the retrieval set. Leveraging this property, we construct a graph where each retrieved document is represented as a node, and edges capture document-document similarity relationships. By propagating ranking scores through this graph structure, our approach prioritizes clusters of semantically consistent documents while suppressing adversarially crafted outliers. As illustrated in Figure 2, our method significantly enhances the robustness of RAG-based LLMs, mitigating adversarial influences while preserving the integrity of benign retrieval results.

We conducted comprehensive experiments on Natural Questions (NQ), MS-MARCO, and HotpotQA across five different models. Our method has shown at least a 30% decrease in reducing the Attack Success Rate (ASR), with improvements of up to 80% across various adversarial attack strategies.

We summarize our contributions as follows:

- We introduce GRADA, a weighted similarity graph among retrieved documents iteratively propagates scores to mitigate the impacts of adversarial passages.
- We introduce a novel scoring function that simultaneously captures both query-document and document-document correlations, thereby improving robustness against adversarial attempts to mimic the query.
- Conducted comprehensive experiments across three different datasets against four chosen attacks. Showing GRADA's advantages over the current defense baselines.

2 Related Work

Corpus poisoning attacks (Zhong et al., 2023) show a possible new attack surface on LLMs. However, this method does not directly affect the accuracy of the LLM; instead, it focuses on the retriever. Later, prompt injection attacks were introduced to bypass the retriever and affect the generator successfully (Greshake et al., 2023; Pasquini et al., 2024). However, compared to the prior work, these methods are unstable in ensuring the retriever retrieves the adversarial passage every time.

More recently, PoisonedRAG (Zou et al., 2024) was proposed as a more stable attack. It uses two passages concatenated together, with one of them appended to guarantee the retrieval of the adversarial passage and one to achieve a given adver-

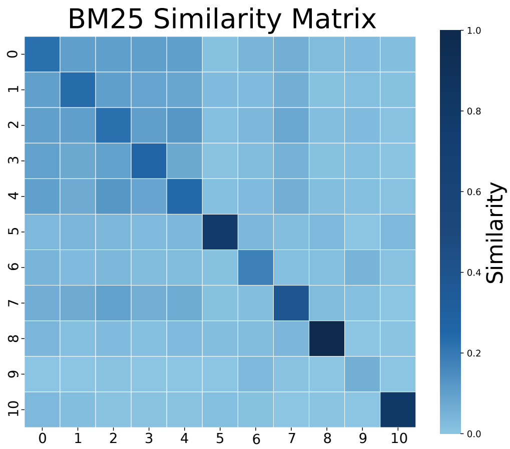


Figure 3: BM25 similarity matrix among retrieved documents, where D0-D4 are poisoned, and D5-D10 are clean.

sarial goal on the generator. The goal is to let the LLM output the answer the attacker wants. PoisonedRAG inspired a lot of the new attacks. Phantom (Chaudhari et al., 2024), which introduces a trigger to the question and achieves the adversarial goal only when the trigger is shown in the query. Another Prompt Injection Attacks (PIA, Clop and Teglia, 2024) makes use of the passage that guarantees the retrieve in PoisonedRAG and focuses on the adversarial goal beyond misinformation.

A recent study proposed a defense mechanism that generates responses independently and produces an output based on the majority vote (Xiang et al., 2024). However, this method initiates its defense at the generator stage, which can impact the accuracy of the system, especially when multiple documents are required. GRADA addresses this issue by focusing on the stage before generation, specifically the reranking process.

3 GRADA

A defining characteristic of recent poisoning attacks on RAG (Zou et al., 2024) is their exclusive emphasis on ensuring semantic similarity to the query while introducing anomalous similarities among poisoned documents. However, these attacks overlook the relationships among benign retrieved documents, as illustrated in Figures 2 and 3. Leveraging these abnormal similarity patterns, we propose a graph-based reranking method that utilizes document-document similarity to enhance retrieval robustness. In Section 3.1, we detail the graph construction process, followed by a description of our reranking system in Section 3.2.

3.1 Graph Construction

We construct a weighted, undirected graph $G = (V, E)$, where each node $v_i \in V$ corresponds to a document doc_i, and each edge $e_{ij} \in E$ is an undirected edge connecting node v_i and v_j. Each edge is assigned a weight $w_{ij} \in \mathbb{R}^+$, which quantifies the similarity between the corresponding documents, *i.e.,* $\text{sim}(v_i, v_j)$. The graph is undirected because document relationships are not inherently directional; rather, the connectivity structure defines their associations. The edge weight w_{ij} can be computed using two different approaches:

- **Doc-to-Doc Similarity (D2DSIM):** The weight is directly determined by the similarity between documents.
- **Hybrid Relevance Similarity (HRSIM):** A function f that integrates both document-document similarity and query-document relevance:

$$w_{ij} = f\big(\text{sim}(v_i, v_j), \text{sim}(v_i, q), \text{sim}(v_j, q)\big)$$

The second approach assigns edge weights that not only reflect direct document-to-document similarity but also incorporate each document's relevance to an external query. This dual consideration leads to a more nuanced representation of document relationships.

To mitigate the influence of adversarial passages—documents that mimic the query q to gain higher rankings—we introduce a function f, which adjusts the similarity score by applying a penalty based on the document-to-query similarities. First, we define the combined query relevance for a pair of documents v_i and v_j as follows:

$$\text{sim}_{\text{sum}} = \text{sim}(v_i, q) + \text{sim}(v_j, q)$$

Then, the edge weight w_{ij} between v_i and v_j is computed by subtracting a penalty term from their direct similarity, ensuring that the weight remains non-negative:

$$w_{ij} = \max\left(\text{sim}(v_i, v_j) - \alpha \cdot \text{sim}_{\text{sum}}, 0\right)$$

Here, α is a penalty coefficient that controls the influence of query similarity. If $\text{sim}(v_i, v_j) < \alpha \cdot [\text{sim}\,(v_i, q) + \text{sim}\,(v_i, q)]$, the edge weight is set to zero, effectively removing the connection between v_i and v_j.

Regarding the similarity function, we explore two popular methods:

- **BM25:** we use BM25 (Robertson and Zaragoza, 2009) to calculate $\text{sim}(v_i, v_j)$. Since BM25 is an asymmetric metric, we adopt the following approach to compute the similarity score, ensuring symmetry in the process:

$$w_{ij} = \frac{1}{2}\left(\text{BM25}\left(v_i, v_j\right) + \text{BM25}\left(v_j, v_i\right)\right)$$

- **Embedding-based Distance (EBD):** we transform the documents $\mathbf{x}_i$ and $\mathbf{x}_j$ into dense vectors v_i and v_j and compute their cosine distance:

$$w_{ij} = \text{sim}(v_i, v_j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\|\|\mathbf{x}_j\|}$$

3.2 Reranking

Inspired by PageRank (Page et al., 1999), we refine document rankings through an iterative score propagation process after constructing the graph. This approach prioritizes well-connected nodes while mitigating the influence of adversarial documents, ensuring a more robust and reliable ranking.

Initially, each node v_i is assigned a score s_i^*, forming the initial score vector $\mathbf{s}^* = [s_1^*, s_2^*, \ldots, s_n^*]^\top$. The scores are then iteratively updated at each step t via:

$$s_i^{(t)} = (1-d)s_j^* + d \sum_{v_j \in \mathcal{N}(i)} \frac{w_{ij}}{\sum_{v_k \in \mathcal{N}(j)} w_{jk}} s_j^{(t-1)} \tag{1}$$

where $N(i)$ represents the set of neighbor nodes connected by v_i and d is the damping factor, typically set to 0.85. The initial score vector s^* is set by uniform initialization $s^* = \left[\frac{1}{|V|}, \frac{1}{|V|}, \ldots, \frac{1}{|V|}\right]$. For experiments comparing different initialization methods, please refer to Appendix D.

The framework works as follows: The retriever identifies M documents most similar to the query, with n being the number of documents originally intended for retrieval and $M \geq n$. We retrieve additional documents to maintain consistency in the number of documents in the non-defended scenario. By ensuring that poisoned documents do not form the majority in the retrieved set (with $M \geq 2n$), we prevent adversarial documents, which may exploit the query for high relevance scores, from dominating. For example, if the original set of n documents contains all poisoned ones (*e.g.*, $n = 5$), adding $\geq n$ benign documents ensures the majority is non-poisoned. This strategy guarantees that non-poisoned documents remain a significant portion of the final selection, enhancing the system's robustness against adversarial manipulation.

After the algorithm reaches a stationary score distribution, the top n documents are retained, while the remaining documents are discarded. These top n documents are then provided as the context of the model.

4 Experiments

This section begins by detailing the experimental setup, followed by a comparison of our approach with multiple baseline methods. Finally, we analyze the effectiveness of our approach across different settings.

4.1 Experimental Setup

Attack setup. We conduct experiments on three widely used english datasets: **Natural Question** (Kwiatkowski et al., 2019), **MS-MARCO** (Nguyen et al., 2016) and **HotpotQA** (Yang et al., 2018). The victim models chosen for this study are **GPT-3.5-Turbo (version 0125)** (Brown et al., 2020), **GPT-4o (version 2024-08-06)** (OpenAI et al., 2024), **Qwen2.5** (Qwen et al., 2025) and **LLaMA-3** (Grattafiori et al., 2024). The prompts used to generate answers are detailed in Appendix A. Contriever (Izacard et al., 2021), is a dense retriever model used to find relevant documents by calculating similarity scores between the query and the knowledge base. It was selected for this study due to its efficiency and ability to handle large datasets. In this work, we investigate four distinct attack strategies on RAG. Two of them are Black-box attacks that have no knowledge about the retriever: PoisonedRAG (Zou et al., 2024) and PIA (Greshake et al., 2023; Pasquini et al., 2024; Perez and Ribeiro, 2022). The remaining two are white-box attacks, in which the attacker has access to the victim's retriever: PoisonedRAG(Hotflip) (Zou et al., 2024) and Phantom (Chaudhari et al., 2024)

Under default settings without any defense, as in Zou et al. (2024), we retrieve the five most similar documents from the knowledge database to serve as the context for each question. We select 10 close-ended questions from each dataset, repeated 10 times and excluding questions that have already been used in previous iterations, totaling 100 questions for the attack experiments.

However, in contrast to Zou et al. (2024), where five poisoned texts are generated and injected into the knowledge base, To provide a more realistic assessment of the attack's effectiveness, we modify the experiment to inject only a single poisoned document into the database. The original setup, which retrieved only poisoned documents, resulted in a 100% Attack Success Rate (ASR), making it impractical to evaluate the true impact of the attack. As shown in Figure 3, a similarity cluster of poisoned documents appears in the top-left corner. By applying a clustering algorithm, we can identify and merge redundant information, effectively removing repetitive poisoned entries. This adjustment ensures that only one poisoned document is retrieved, allowing for a more meaningful evaluation of the attack's success.

Defense setup. We explore three similarity score combinations for GRADA: Embedding-based Distance, BM25, and Hybrid Relevance Similarity with BM25 as the similarity function.[1] Here, we utilize Contriever to encode both documents and queries, while for BM25, we adopt the implementation provided by Lù (2024). We compare GRADA against two reranking models and one defense method: HLATR (Zhang et al., 2022), which achieved first place in the MS-MARCO Passage Ranking Leaderboard, BGE-reranker (Xiao et al., 2023), which achieves a high precision score in ranking tasks, and Keyword Aggregation (Xiang et al., 2024), the only existing defense specifically designed for RAG-based adversarial attacks, as a baseline.

We evaluate the effectiveness of these defense methods by integrating them into our two-stage retrieval system described in Section 3. We initially retrieve $M = 10$ documents, which are then reranked using the aforementioned methods (except for Keyword Aggregation). The top five ranked documents are subsequently provided as the context for the model to answer the query. This ensures that, regardless of the defense configuration, the model always receives a fixed number of five context documents to respond to the question. For Keyword Aggregation, which does not perform reranking, the model directly generates the output based on the algorithm's selection.

[1]We examine other similarity functions in Section 4.3

Evaluation metrics. In our experiments, we employ Attack Success Rate (ASR) and Exact Match (EM) as metrics. ASR is defined as the ratio of successful attacks to the total number of attacks conducted. An attack is considered successful if the intended poisoned answer appears as a substring within the generated response from the model. This definition accommodates attack strategies like PIA, which aim to introduce harmful links into the output of the model, allowing for some tolerance to semantically equivalent responses. A higher ASR indicates a more successful attack. This evaluation methodology follows the approach used in previous work (Zou et al., 2024).

To assess the question-answering accuracy of the models, we adopt EM score. EM requires that the predicted answer of the model matches the ground truth answer exactly. This strict criterion ensures that the response of the model is precise and follows the need for exact wording specified in the query, as outlined in Appendix A.

4.2 Results and Discussions

Attacking without defense. As shown in Table 1, including a single poisoned document in the retrieval process results in a high ASR score. For instance, PoisonedRAG achieves an ASR of 50% across three datasets on both GPT-3.5-Turbo and Llama3.1-8b-Instruct. PIA achieves at least 69% ASR on Llama3.1-8b-Instruct and up to 100% ASR in GPT-3.5-Turbo. These findings emphasize that even minimal adversarial input can achieve very high ASR and degrade the model's accuracy.

Effectiveness of GRADA. The impact of GRADA on mitigating adversarial attacks is demonstrated in Tables 1 and 2. As shown in Table 1, on the NQ and MS-MARCO datasets using GPT-3.5-Turbo, the ASR for PIA decreases from 98.0% and 88.0% to 2.0% and 3.0% by using D2DSIM-EBD. With D2DSIM-EBD, GRADA is also effective against PoisonedRAG, effectively reducing the ASRs from 56.0% and 48.0% to 27.0% and 28.0%. However, the reduction of ASR against PoisonedRAG is more modest than against the other attacks. On this attack, D2DSIM-BM25 and HRSIM led to significant improvements compared to D2DSIM-EBD, where D2DSIM-BM25 achieved an extra 13% decrease in ASR to 14% and 15%. Beyond that, HRSIM which introduces penalties for excessive similarity to the query, finalizes the ASR to 3% and 9%.

Defense	PoisonedRAG			PIA			Phantom		
	HotpotQA	NQ	MS-MARCO	HotpotQA	NQ	MS-MARCO	HotpotQA	NQ	MS-MARCO
	ASR ↓ / EM ↑	ASR ↓ / EM ↑	ASR ↓ / EM ↑	ASR ↓ / EM ↑	ASR ↓ / EM ↑	ASR ↓ / EM ↑	ASR ↓ / EM ↑	ASR ↓ / EM ↑	ASR ↓ / EM ↑
					GPT-3.5-Turbo				
None	59.0 / 32.0	56.0 / 34.0	48.0 / 41.0	100.0 / 0.0	98.0 / 2.0	88.0 / 7.0	80.0 / 18.0	79.0 / 11.0	65.0 / 28.0
HLATR	64.0 / 29.0	51.0 / 37.0	34.0 / 51.0	100.0 / 0.0	92.0 / 4.0	84.0 / 9.0	74.0 / 22.0	84.0 / 12.0	51.0 / 39.0
BGE-reranker	54.0 / 38.0	46.0 / 44.0	31.0 / 59.0	98.0 / 2.0	37.0 / 43.0	43.0 / 43.0	78.0 / 16.0	54.0 / 24.0	44.0 / 41.0
Keyword Aggregation	**13.0** / 63.0	**2.0** / 48.0	**3.0** / 62.0	**0.0** / 65.0	**0.0** / 51.0	**0.0** / 58.0	**0.0** / 53.0	**0.0** / 47.0	**0.0** / 58.0
GRADA (D2DSIM-EBD)	49.0 / 39.0	27.0 / 51.0	28.0 / 57.0	33.0 / 43.0	2.0 / 59.0	3.0 / 72.0	56.0 / 29.0	12.0 / 49.0	13.0 / 61.0
GRADA (D2DSIM-BM25)	45.0 / 40.0	14.0 / 57.0	15.0 / 65.0	42.0 / 33.0	12.0 / 55.0	2.0 / 69.0	27.0 / 32.0	6.0 / 51.0	1.0 / 68.0
GRADA (HRSIM)	**10.0** / 52.0	**3.0** / 59.0	**9.0** / 72.0	**27.0** / 42.0	**2.0** / 59.0	**1.0** / 74.0	**23.0** / 38.0	**0.0** / 51.0	**0.0** / 70.0
					Llama3.1-8b-Instruct				
None	51.0 / 37.0	50.0 / 32.0	41.0 / 39.0	88.0 / 3.0	82.0 / 8.0	69.0 / 14.0	54.0 / 19.0	50.0 / 28.0	17.0 / 55.0
HLATR	52.0 / 36.0	39.0 / 42.0	35.0 / 44.0	91.0 / 3.0	69.0 / 17.0	50.0 / 20.0	48.0 / 32.0	47.0 / 30.0	16.0 / 50.0
BGE-reranker	50.0 / 38.0	41.0 / 40.0	33.0 / 43.0	81.0 / 9.0	29.0 / 41.0	21.0 / 44.0	32.0 / 39.0	24.0 / 41.0	8.0 / 61.0
Keyword Aggregation	**4.0** / 35.0	**3.0** / 39.0	**6.0** / 38.0	**0.0** / 33.0	**0.0** / 42.0	0.0 / 41.0	**0.0** / 33.0	**0.0** / 36.0	0.0 / 39.0
GRADA (D2DSIM-EBD)	41.0 / 37.0	23.0 / 46.0	32.0 / 41.0	31.0 / 35.0	**1.0** / 55.0	2.0 / 55.0	18.0 / 40.0	5.0 / 51.0	1.0 / 50.0
GRADA (D2DSIM-BM25)	31.0 / 40.0	8.0 / 53.0	20.0 / 49.0	39.0 / 29.0	8.0 / 48.0	**0.0** / 55.0	27.0 / 37.0	5.0 / 53.0	**0.0** / 54.0
GRADA (HRSIM)	**7.0** / 43.0	**2.0** / 57.0	**11.0** / 53.0	**23.0** / 37.0	2.0 / 56.0	**0.0** / 58.0	**14.0** / 40.0	**0.0** / 54.0	**0.0** / 60.0

Table 1: ASR and EM (%) for various defense methods on the three attack methods (PoisonedRAG, PIA, Phantom) on GPT-3.5-Turbo and Llama3.1-8b-Instruct. The results of other models can be found in Tables 8 to 12. We highlight the top-2 lowest ASR results in **blue** cells.

Defense	HotpotQA	NQ	MS-MARCO
	ASR ↓ / EM ↑	ASR ↓ / EM ↑	ASR ↓ / EM ↑
		GPT-3.5-Turbo	
None	64.0 / 30.0	54.0 / 29.0	39.0 / 51.0
HLATR	56.0 / 34.0	49.0 / 36.0	34.0 / 52.0
BGE-reranker	56.0 / 35.0	43.0 / 40.0	27.0 / 60.0
Keyword Aggregation	**8.0** / 59.0	**2.0** / 48.0	**5.0** / 59.0
GRADA (D2DSIM-EBD)	44.0 / 37.0	9.0 / 56.0	9.0 / 69.0
GRADA (D2DSIM-BM25)	40.0 / 43.0	9.0 / 60.0	8.0 / 70.0
GRADA (HRSIM)	**7.0** / 54.0	**4.0** / 60.0	**7.0** / 71.0
		Llama3.1-8b-Instruct	
None	49.0 / 31.0	51.0 / 29.0	53.0 / 31.0
HLATR	46.0 / 36.0	41.0 / 38.0	36.0 / 39.0
BGE-reranker	48.0 / 32.0	43.0 / 36.0	37.0 / 34.0
Keyword Aggregation	**4.0** / 33.0	**3.0** / 41.0	**7.0** / 35.0
GRADA (D2DSIM-EBD)	37.0 / 37.0	10.0 / 51.0	17.0 / 52.0
GRADA (D2DSIM-BM25)	24.0 / 45.0	11.0 / 53.0	17.0 / 49.0
GRADA (HRSIM)	**7.0** / 43.0	**5.0** / 54.0	**10.0** / 53.0

Table 2: ASR and EM (%) for various defense methods on PoisonedRAG (Hotflip).

The defense methods demonstrate consistent effectiveness across the NQ and MS-MARCO datasets, achieving ASR reductions of over 30% in most cases. However, performance on HotpotQA is less stable, particularly for D2DSIM-EBD and D2DSIM-BM25, which achieve only around a 10% reduction in ASR against PoisonedRAG attacks. In contrast, HRSIM maintains its effectiveness, delivering ASR reductions exceeding 30%, comparable to its performance on other datasets. This discrepancy likely stems from HotpotQA's multi-hop reasoning requirements, which pose challenges for single-document similarity metrics.

In Table 1, HLATR and BGE-reranker exhibit limited ability to filter poisoned documents, with ASR remaining largely unchanged compared to scenarios without any defense mechanisms. Although BGE-reranker occasionally outperforms HLATR, its overall performance remains inferior to GRADA in handling adversarial cases. This discrepancy underscores a critical limitation in contemporary reranking systems, which are primarily optimized for question relevance but insufficiently equipped to address adversarial attacks with high question relevance.

Keyword Aggregation is able to reduce ASR significantly, especially for attacks like PIA and Phantom. Keyword Aggregation works by extracting keywords from the answers of each passage to generate the final response, effectively neutralizing attack payloads designed to manipulate or deny answers, such as producing advertisements. However, while it reduces ASR effectively, its EM scores are lower than those of GRADA. For example, on Llama3.1-8b-Instruct in Table 1, GRADA's EM scores dominate Keyword Aggregation with at most 21% difference as some critical information may be lost during keyword extraction. This shows the ability of GRADA to perform well on normal answers even after mitigating adversarial contents.

Similar results to those presented in Table 1 can also be observed in Table 2. Notably, GRADA combined with HRSIM consistently outperforms all other approaches, demonstrating that HRSIM is a strong similarity scoring function compared to the alternatives used in GRADA.

Table 3 highlights the impact of different defense mechanisms on benign inputs. On GPT-3.5-Turbo, both HLATR and BGE-reranker demonstrate strong performance, outperforming GRADA

Defense	HotpotQA	NQ	MS-MARCO
GPT-3.5-Turbo			
None	65.0	58.0	76.0
HLATR	69.0	62.0	78.0
BGE-reranker	**70.0**	66.0	**78.0**
Keyword Aggregation	56.0	49.0	58.0
GRADA (D2DSIM-EBD)	64.0	60.0	75.0
GRADA (D2DSIM-BM25)	58.0	**66.0**	77.0
GRADA (HRSIM)	54.0	63.0	77.0
Llama3.1-8b-Instruct			
None	52.0	50.0	54.0
HLATR	55.0	51.0	57.0
BGE-reranker	**58.0**	54.0	**59.0**
Keyword Aggregation	33.0	41.0	40.0
GRADA (D2DSIM-EBD)	51.0	**56.0**	58.0
GRADA (D2DSIM-BM25)	48.0	51.0	54.0
GRADA (HRSIM)	44.0	54.0	57.0

Table 3: EM scores of defense methods when presented with benign inputs.

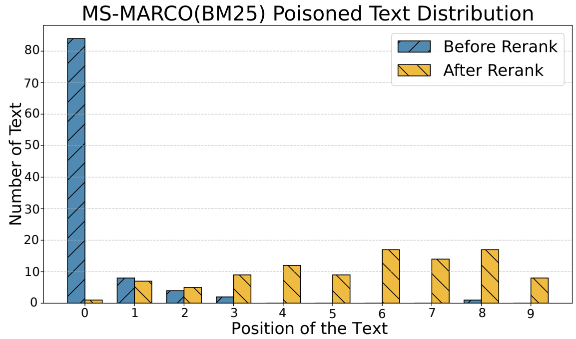


Figure 4: Distribution of poisoned document positions after applying GRADA (D2DSIM-BM25) in the MS-MARCO dataset. Documents positioned below rank 5 are effectively mitigated by the ranking algorithm. Other results are showed in Figure 11 and Tables 5 to 7

and enhancing the model's overall accuracy. these reranking systems yield at least a 2% improvement in EM scores, suggesting their effectiveness in mitigating noise unrelated to the posed questions.

GRADA with D2DSIM-EBD effectively preserves model performance on benign inputs across all datasets, with EM score deviations remaining within 2%. Notably, the use of D2DSIM-BM25 leads to an 8% improvement in EM scores on NQ, matching the performance of BGE-reranker, which achieves the highest EM overall. However, on HotpotQA, HRSIM resulted in an 11% reduction in EM scores when handling benign inputs. While this trade-off is significant, it corresponds to HRSIM's remarkable ASR reduction. Striking a balance between retrieval quality and defense robustness remains a crucial challenge for future research.

Keyword Aggregation has a much lower performance also in EM scores on benign input compared to GRADA. For example, in MS-MARCO, it results in 40% compared to 57% on Llama3.1-8b-Instruct and 58% compared to 77% on GPT-3.5-Turbo. Indeed showing the cost of discarding valuable information when facing benign documents.

Using GRADA, we demonstrate that it is possible to defend against the chosen attacks without compromising the model's overall performance on EM. While reranking methods such as HLATR and BGE-reranker show promise in reducing noise, their limited effectiveness in countering adversarial attack noise highlights a critical gap in existing defenses. Similarly, Keyword Aggregation presents a valuable strategy for mitigating attack payloads but comes with trade-offs in EM scores.

For the attack to be effective, the attackers must ensure that the retriever selects the poisoned documents. To achieve this, they primarily focus on making these documents resemble the queries, as most retrieval models prioritize query-document similarity when selecting relevant results. Additionally, poisoned documents typically exhibit only weak similarity to other documents in the corpus. This characteristic makes them less susceptible to detection by defense mechanisms that compare retrieved documents against one another.

4.3 Additional Studies

Ranking distribution. We have demonstrated the effectiveness of our approach in enhancing defense performance. To gain a deeper understanding of its impact, we further analyze how our method systematically lowers the ranking of poisoned documents. As illustrated in Figure 4, the position distribution of poisoned documents within the retrieval set shifts significantly after applying GRADA with D2DSIM-BM25. Notably, over 70% of poisoned documents are relegated beyond the top five positions, substantially reducing their influence. These findings confirm that GRADA is both robust and effective in mitigating adversarial attacks.

Selections of HRSIM. Thus far, our focus has primarily been on utilizing BM25 for HRSIM. In this section, we explore other similarity functions for HRSIM. As shown in Figure 5, we extend our analysis by incorporating SBERT (Reimers and Gurevych, 2019), alongside the three previously discussed methods, to better capture document-to-document similarity. Our results indicate that both EBD and SBERT exhibit strong overall performance against PIA and PoisonedRAG attacks.

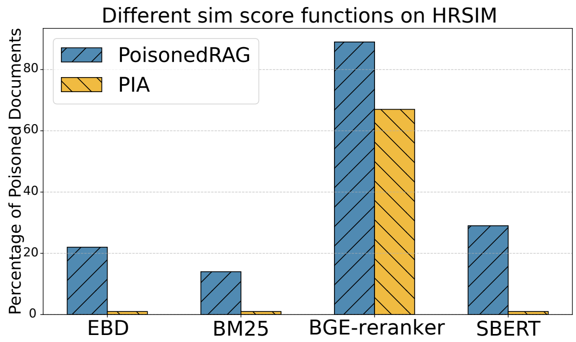


Figure 5: HRSIM performance with different similarity functions selection on MSMARCO dataset. The figure illustrates the proportion of test instances in which poisoned documents remain among the top five retrieved results.

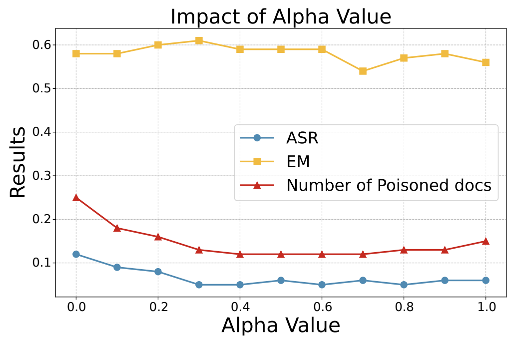


Figure 6: Impact of the α value as it increases with three metrics (ASR, number of poisoned documents, and EM) on NQ dataset with GPT-3.5-Turbo.

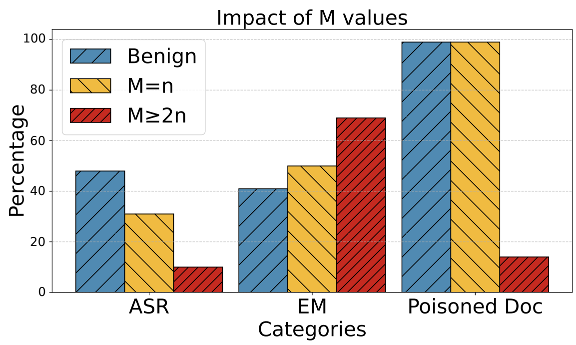


Figure 7: Impact of the M value as it changes with three metrics (ASR, number of poisoned documents, and EM) on MSMARCO dataset with GPT-3.5-Turbo.

In contrast, BGE-Reranker struggles to effectively filter out poisoned documents, likely due to its primary training objective of computing query-to-document similarities rather than document-to-document relationships. HRSIM, when combined with BM25, effectively minimize the presence of poisoned documents, reducing them to just 14 out of 100 test instances. This outcome underscores its remarkable effectiveness in filtering malicious content.

Impact of α and M. As shown in Figure 6, the number of poisoned documents in the context decreases as α increases, reaching a minimum at $\alpha = 0.3$ before starting to rise again after $\alpha = 0.8$. The ASR follows a similar trend to the number of poisoned documents after $\alpha = 0.3$. Conversely, the EM score exhibits a minimum at $\alpha = 0.7$. We selected $\alpha = 0.4$ because it strikes a balance, avoiding excessive penalization for query similarity, which could otherwise result in fewer query-related documents. When $\alpha = 0.4$, all three metrics (ASR, number of poisoned documents, and EM) are within an acceptable range, approaching the optimal performance values for α.

Figure 7 illustrates the effect of selecting $M = n$. It shows that, regardless of how documents are reranked, poisoned documents can still remain within the context provided to the model. However, this approach results in a 17% decrease in ASR and a 9% increase in EM, indicating that simply adjusting document positions can significantly impact model performance. This aligns with our observations in Table 4, and the specific positions of the documents are detailed in Figure 4. By including additional documents for reranking and then retrieving only the top n results, the ASR is further reduced from 21% to 10%, with only 14% of poisoned documents remaining in the context provided to the model. This demonstrates the importance of including extra documents during reranking to remove poisoned content and achieve better overall performance effectively.

5 Conclusion

The study examines the robustness challenges faced by RAG systems. We identify a critical vulnerability in current adversarial attacks, which focus on increasing semantic similarity to the query without accounting for the relationships between the retrieved documents. Our proposed graph-based filtering framework, GRADA, enhances the robustness of RAG systems by leveraging document similarities and effectively mitigating adversarial impacts through information flow. Experimental results on datasets such as MS-MARCO and NQ, demonstrate at least 30% reductions in ASR across various adversarial strategies. Overall, this work presents a promising direction for developing more secure and reliable RAG systems.

Limitations

Despite its effectiveness, our approach has limitations. First, it struggles with multi-hop reasoning tasks, facing attacks like PIA and Phantom. As the number of poisoned documents increases, system robustness deteriorates. Second, our method assumes poisoned documents are a minority. When they form the majority, their effectiveness declines, and future work should explore adaptive retrieval strategies to counter adversarial dominance.

Ethics Statement

Our study focuses on improving the robustness of RAG systems, thereby enhancing their reliability and minimizing harmful manipulations. We evaluated our proposed method, GRADA, using publicly available datasets as detailed in Appendix F. We do not engage in harmful data practices.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

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. *Preprint*, arXiv:2405.20485.

Cody Clop and Yannick Teglia. 2024. Backdoored retrievers for prompt injection attacks on retrieval augmented generation of large language models. *Preprint*, arXiv:2410.14479.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. 2024. Open llm leaderboard v2. `https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard`.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

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, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, AISec '23, page 79–90, New York, NY, USA. Association for Computing Machinery.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning.

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.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Xing Han Lù. 2024. Bm25s: Orders of magnitude faster lexical search via eager sparse scoring. *Preprint*, arXiv:2407.03618.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

Dario Pasquini, Martin Strohmeier, and Carmela Troncoso. 2024. Neural exec: Learning (and learning from) execution triggers for prompt injection attacks. In *Proceedings of the 2024 Workshop on Artificial Intelligence and Security*, AISec '24, page 89–100, New York, NY, USA. Association for Computing Machinery.

Fábio Perez and Ian Ribeiro. 2022. Ignore previous prompt: Attack techniques for language models. *Preprint*, arXiv:2211.09527.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Chong Xiang, Tong Wu, Zexuan Zhong, David Wagner, Danqi Chen, and Prateek Mittal. 2024. Certifiably robust rag against retrieval corruption. *Preprint*, arXiv:2405.15556.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained transformers for text ranking: BERT and beyond. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*, pages 1–4, Online. Association for Computational Linguistics.

Yanzhao Zhang, Dingkun Long, Guangwei Xu, and Pengjun Xie. 2022. Hlatr: Enhance multi-stage text retrieval with hybrid list aware transformer reranking. *ArXiv*, abs/2205.10569.

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.

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. 2024. Poisonedrag: Knowledge poisoning attacks to retrieval-augmented generation of large language models. *Preprint*, arXiv:2402.07867.

A Prompts to LLM

Prompts given to LLMs

Prompt 1: You are a helpful assistant, below is a query from a user and some relevant contexts. Answer the question given the information in those contexts / keywords. Your answer should be short ,concise and without explainations ,just answer with one or two words like yes or no. If you cannot find the answer to the question, just say "I don't know". Contexts / Keywords: [context / keywords] Query: [question] Answer:
Prompt 2: You are a helpful assistant, below is a query from a user and some relevant contexts. Answer the question given the information in those contexts. If you cannot find the answer to the question, just say "I don't know". Contexts: [context] Query: [question] Answer:

Figure 8: Example of prompts given to LLM, Prompt 1 is used in all of the defence methods to generate the final output. Prompt 2 is only used in the phase to generate keywords.

B Deconstructing PoisonedRAG

PoisonedRAG is an adversarial attack on RAG systems that operates in two distinct stages. The first part enhances the relevance of the adversarial passage to the query, ensuring semantic similarity to the query and increasing the likelihood of its retrieval. The second part introduces adversarial content to achieve the attack's intended goal, which is typically to mislead the model into generating a specific incorrect response. This two-part structure has become a foundational template for several subsequent attack strategies targeting RAG systems.

While the approach used to achieve the first part of the attack is effective, it is also relatively simple and naive. Specifically, the adversarial passage is constructed by using the query itself as the first part, a method that can easily be identified and filtered by humans. Moreover, as demonstrated in Figure 3 and Figure 9, the attacks injected into the database often exhibit considerable similarity to one another. This redundancy presents an opportunity for improvement: by employing a clustering algorithm, we can detect and merge these repetitive entries, effectively removing redundant information and weakening the attack's overall impact.

Attack Method	HotpotQA	NQ	MS-MARCO
Normal retrieved	59.0	56.0	48.0
w/o question	66.0	61.0	51.0
Poisoned in the middle	59.0	54.0	37.0
w/o question	63.0	51.0	34.0

Table 4: PoisonedRAG Attack Success Rate (%) where the retrieval part is removed, and the poisoned documents are placed in the middle.

Despite the simplicity of this approach, PoisonedRAG still manages to degrade model performance significantly. As shown in Table 4 (first row), even with just one adversarial passage, the attack achieves an attack success rate (ASR) of approximately 50% across three different datasets. This underscores the effectiveness of the adversarial strategy, despite its seemingly straightforward nature, in misleading the model and causing substantial degradation in accuracy.

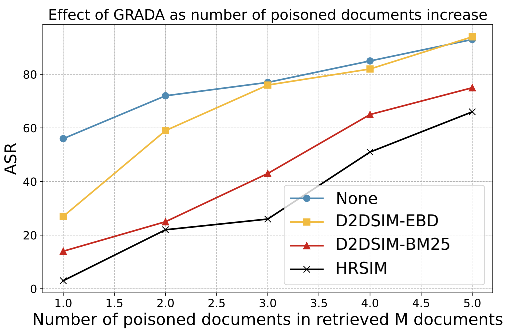
Interestingly, our analysis reveals that the first part of the adversarial passage—composed of the query itself—does not significantly affect the ASR. In fact, as shown in Table 4, removing this part actually leads to an increase in the ASR. This suggests that the primary role of the query in the context is not to contribute to the ASR directly but to ensure that the retriever selects the adversarial document. Furthermore, Table 4 shows another significant finding: the position of the poisoned document within the retrieved set plays a crucial role in determining the overall ASR. Specifically, the adversarial document's position can influence the effectiveness of the attack, with a potential decrease in ASR of up to 17%. This phenomenon aligns with the lost-in-the-middle effect (Liu et al., 2024), where the adversarial document's position among other retrieved documents impacts its effectiveness in influencing the model's output.

A key characteristic of the PoisonedRAG attack is its exclusive focus on ensuring semantic similarity to the query without considering the relationships among the retrieved documents. This limitation can be leveraged to defend against such attacks. By focusing on the similarities between the documents in the retrieved set, we can filter out adversarial passages and mitigate the attack's effectiveness.

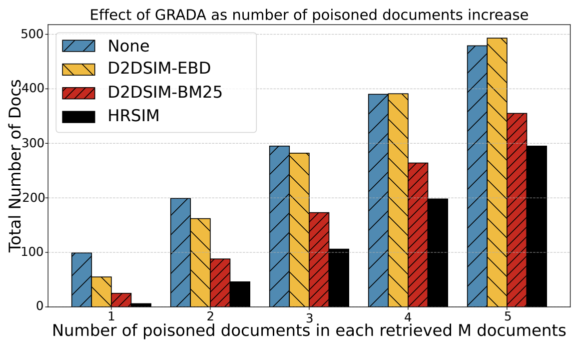
PoisonedRAG Example

Question: "When did the apple iphone se come out?"
Benign documents 1: "March 21, 2017 iPhone SE shipped with iOS 9.3, supporting iPhone 6 and 6S-exclusive features such as Apple Pay, Live Photos, Retina Flash and always-on Siri voice activation.[6][10][16]"
Benign documents 2: "On launch, it was released in models with either 16 or 64 GB of internal storage; on March 21, 2017, Apple announced that these models would be replaced by new improved models at the same launch price points, releasing on March 24, 2017.[11][12]"
Poisoned Document 1: "When did the apple iphone se come out. The Apple archive, housing important launch events, mentions a significant one on April 15, 2015. On this day, the compact, budget-friendly iPhone SE hit the market, impressing many gadget enthusiasts."
Poisoned Document 2: "When did the apple iphone se come out. The iPhone series has released several models. Notable amongst them is the iPhone SE, introduced to the market on April 15, 2015, attracting considerable consumer attention."

Figure 9: Example of PoisonedRAG attacks. Poisoned documents injected into the database are all very similar to each other and focus solely on ensuring similarity to the query, the similarities among the retrieved documents are never considered.



(a) ASR of GRADA as poisoned documents increase.



(b) Total poisoned documents remain after filtering.

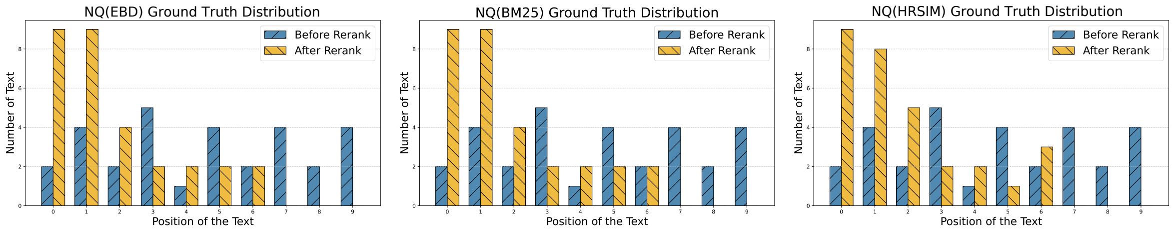
Figure 10: Impact of increasing poisoned documents on GRADA's performance in NQ dataset (GPT-3.5-Turbo, $M = 10$).

C Number of poisoned documents increase

As shown in Figure 10a, GRADA's effectiveness diminishes as the proportion of poisoned documents increases. When using D2DSIM-EBD, its performance converges with that of an undefended system. However, HRSIM remains effective, achieving a 27% reduction in ASR even when half of the retrieved documents are adversarial. This is further supported by Figure 10b, which shows that 38% of poisoned documents are still successfully filtered.

D Different initial score vector

Different initial score vectors can have a significant impact on the final distribution of documents in certain cases. For instance, we experimented with initializing the score vector with query-document similarity $s^* = \left[\frac{sim(q,v_0)}{\sum_{j=0}^{n} sim(q,v_j)}, \frac{sim(q,v_1)}{\sum_{j=0}^{n} sim(q,v_j)}, \ldots, \frac{sim(q,v_n)}{\sum_{j=0}^{n} sim(q,v_j)}\right]$. As illustrated in Figure 12a, using a query-document initialization results in a greater number of documents confined to positions 5 through 8, rather than being ranked at the lower end of the rankings. This issue arises because adversarial documents may receive disproportionately high initial scores compared to benign documents. Such an imbalance gives adversarial documents a substantial advantage, particularly when the edge weights between documents are relatively small. In these scenarios, the graph-based reranking process may struggle to compensate for this initial disparity, as demonstrated in Figure 13. From



(a) D2DSIM-EBD (b) D2DSIM-BM25 (c) HRSIM

Figure 11: Distribution of Ground Truth document positions after applying GRADA in the NQ dataset with different ranking methods.

the analysis in Figure 12b, we observe that this phenomenon is more prevalent in datasets like HotpotQA.

E Computational Resources

The estimated cost of a single defense run on GPT-3.5-Turbo is $0.50, identical to a standard query since the method does not introduce additional API calls. Experiments for LLaMA-3 and Qwen2.5 were conducted on an A100 80GB GPU, with each defense run taking approximately one hour to complete.

F License and Distribution Terms

The dataset used in our experiments is publicly available under Creative Commons Attribution 4.0 International (MS-MARCO) and Apache License 2.0 (NQ, HotpotQA). The code used in our experiments is publicly available under MIT License (BM25s, PoisonedRAG).

Defense Method	PoisonedRAG		PoisonedRAG(Hotflip)		PIA		Phantom	
	Before	After	Before	After	Before	After	Before	After
HLATR	99.0	100.0	99.0	100.0	96.0	93.0	94.0	89.0
BGE-reranker	99.0	100.0	99.0	98.0	96.0	47.0	94.0	58.0
GRADA (D2DSIM-EBD)	99.0	55.0	99.0	20.0	96.0	6.0	94.0	5.0
GRADA (D2DSIM-BM25)	99.0	25.0	99.0	16.0	96.0	6.0	94.0	4.0
GRADA (HRSIM)	99.0	13.0	99.0	8.0	96.0	7.0	94.0	2.0

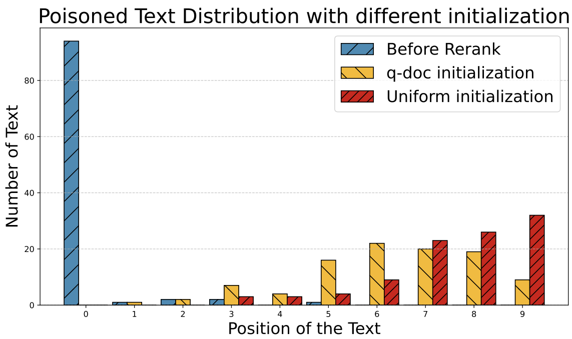
Table 5: The percentage of poisoned documents in the given context to LLM before and after different defense methods on NQ dataset. Method Keyword not included as it is not reranking anything.

Defense Method	PoisonedRAG		PoisonedRAG(Hotflip)		PIA		Phantom	
	Before	After	Before	After	Before	After	Before	After
HLATR	98.0	98.0	99.0	96.0	89.0	85.0	65.0	70.0
BGE-reranker	98.0	98.0	99.0	98.0	89.0	48.0	65.0	53.0
GRADA (D2DSIM-EBD)	98.0	69.0	99.0	22.0	89.0	10.0	65.0	10.0
GRADA (D2DSIM-BM25)	98.0	34.0	99.0	15.0	89.0	2.0	65.0	2.0
GRADA (HRSIM)	98.0	19.0	99.0	8.0	89.0	1.0	65.0	2.0

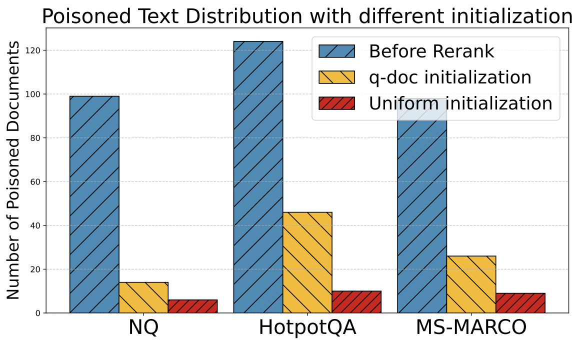
Table 6: The percentage of poisoned documents in the given context to LLM before and after different defense methods on MS-MARCO dataset.

Defense Method	PoisonedRAG		PoisonedRAG(Hotflip)		PIA		Phantom	
	Before	After	Before	After	Before	After	Before	After
HLATR	100.0	100.0	100.0	100.0	100.0	100.0	100.0	99.0
BGE-reranker	100.0	98.0	100.0	100.0	100.0	98.0	100.0	98.0
GRADA (D2DSIM-EBD)	100.0	84.0	100.0	66.0	100.0	52.0	100.0	49.0
GRADA (D2DSIM-BM25)	100.0	64.0	100.0	53.0	100.0	35.0	100.0	32.0
GRADA (HRSIM)	100.0	19.0	100.0	18.0	100.0	26.0	100.0	20.0

Table 7: The percentage of poisoned documents in the given context to LLM before and after different defense methods on HotpotQA dataset.



(a) Distribution of Poisoned document positions after applying GRADA (HRSIM) with different initialization in the NQ dataset.



(b) Total number of poisoned documents after applying GRADA (HRSIM) with different initialization in the NQ dataset.

Figure 12: Impact of different initialization score vectors on GRADA's performance ($M = 10$).

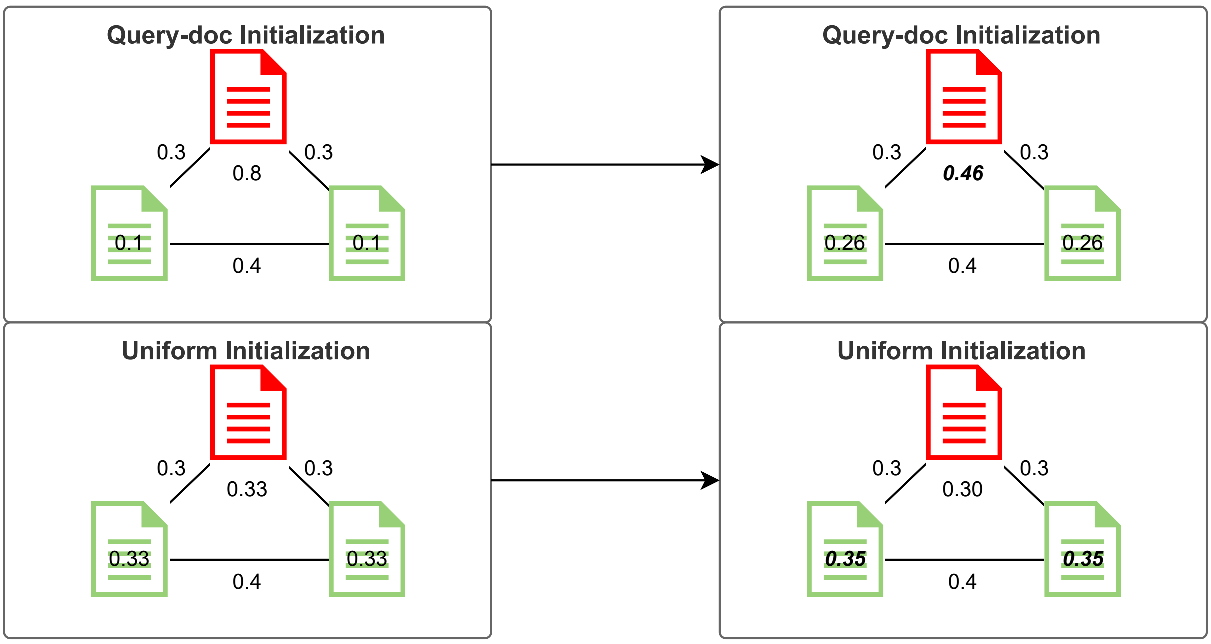


Figure 13: A demonstration on different initial score vector and their results when the adversarial documents receive significantly higher initial scores compared to benign documents.

Model	**Defense**	**HotpotQA**	**NQ**	**MS-MARCO**
GPT-4o	None	62.0	59.0	65.0
	HLATR	67.0	62.0	68.0
	BGE-reranker	63.0	65.0	73.0
	Keyword Aggregation	63.0	47.0	47.0
	GRADA (D2DSIM-EBD)	58.0	54.0	62.0
	GRADA (D2DSIM-BM25)	57.0	59.0	66.0
	GRADA (HRSIM)	52.0	62.0	65.0
Llama3.1-70b-Instruct	None	57.0	59.0	54.0
	HLATR	63.0	60.0	55.0
	BGE-reranker	58.0	66.0	55.0
	Keyword (Xiang et al., 2024)	27.0	35.0	61.0
	GRADA (D2DSIM-EBD)	48.0	56.0	47.0
	GRADA (D2DSIM-BM25)	41.0	58.0	52.0
	GRADA (HRSIM)	38.0	54.0	55.0
Qwen2.5-7b-Instruct	None	44.0	46.0	50.0
	HLATR	48.0	48.0	43.0
	BGE-reranker	45.0	49.0	47.0
	Keyword	12.0	17.0	23.0
	GRADA (D2DSIM-EBD)	41.0	46.0	45.0
	GRADA (D2DSIM-BM25)	38.0	50.0	44.0
	GRADA (HRSIM)	33.0	46.0	51.0

Table 8: Defence methods performance on benign inputs.

Model	Defense	HotpotQA ASR ↓ / EM ↑	NQ ASR ↓ / EM ↑	MS-MARCO ASR ↓ / EM ↑
GPT-4o	None	42.0 / 41.0	28.0 / 40.0	24.0 / 46.0
	HLATR	37.0 / 47.0	26.0 / 50.0	21.0 / 53.0
	BGE-reranker	39.0 / 44.0	24.0 / 55.0	20.0 / 54.0
	Keyword Aggregation	6.0 / 61.0	1.0 / 46.0	5.0 / 45.0
	GRADA (D2DSIM-EBD)	37.0 / 41.0	10.0 / 47.0	19.0 / 51.0
	GRADA (D2DSIM-BM25)	24.0 / 44.0	5.0 / 60.0	10.0 / 64.0
	GRADA (HRSIM)	5.0 / 49.0	1.0 / 65.0	4.0 / 67.0
Llama3.1-70b-Instruct	None	58.0 / 37.0	56.0 / 30.0	55.0 / 28.0
	HLATR	54.0 / 43.0	50.0 / 38.0	41.0 / 37.0
	BGE-reranker	53.0 / 42.0	49.0 / 38.0	38.0 / 38.0
	Keyword (Xiang et al., 2024)	2.0 / 26.0	2.0 / 39.0	0.0 / 59.0
	GRADA (D2DSIM-EBD)	45.0 / 37.0	26.0 / 44.0	35.0 / 38.0
	GRADA (D2DSIM-BM25)	36.0 / 38.0	12.0 / 57.0	15.0 / 51.0
	GRADA (HRSIM)	9.0 / 38.0	3.0 / 53.0	9.0 / 52.0
Qwen2.5-7b-Instruct	None	62.0 / 24.0	50.0 / 27.0	49.0 / 29.0
	HLATR	60.0 / 29.0	44.0 / 30.0	40.0 / 29.0
	BGE-reranker	60.0 / 30.0	48.0 / 29.0	42.0 / 30.0
	Keyword	4.0 / 15.0	0.0 / 17.0	9.0 / 24.0
	GRADA (D2DSIM-EBD)	57.0 / 24.0	24.0 / 36.0	38.0 / 31.0
	GRADA (D2DSIM-BM25)	43.0 / 27.0	12.0 / 45.0	23.0 / 39.0
	GRADA (HRSIM)	7.0 / 34.0	6.0 / 41.0	12.0 / 40.0

Table 9: ASR and EM (%) for various defense methods on PoisonedRAG.

Model	Defense	HotpotQA ASR ↓ / EM ↑	NQ ASR ↓ / EM ↑	MS-MARCO ASR ↓ / EM ↑
GPT-4o	None	46.0 / 41.0	32.0 / 41.0	26.0 / 48.0
	HLATR	43.0 / 44.0	29.0 / 46.0	23.0 / 51.0
	BGE-reranker	40.0 / 42.0	27.0 / 49.0	20.0 / 55.0
	Keyword Aggregation	8.0 / 61.0	1.0 / 46.0	4.0 / 46.0
	GRADA (D2DSIM-EBD)	31.0 / 47.0	6.0 / 54.0	12.0 / 58.0
	GRADA (D2DSIM-BM25)	21.0 / 47.0	5.0 / 61.0	8.0 / 65.0
	GRADA (HRSIM)	5.0 / 48.0	1.0 / 65.0	4.0 / 67.0
Llama3.1-70b-Instruct	None	59.0 / 33.0	54.0 / 28.0	53.0 / 29.0
	HLATR	51.0 / 38.0	46.0 / 35.0	34.0 / 33.0
	BGE-reranker	47.0 / 45.0	43.0 / 38.0	37.0 / 32.0
	Keyword (Xiang et al., 2024)	2.0 / 30.0	2.0 / 39.0	4.0 / 60.0
	GRADA (D2DSIM-EBD)	35.0 / 39.0	12.0 / 52.0	18.0 / 46.0
	GRADA (D2DSIM-BM25)	28.0 / 41.0	8.0 / 57.0	12.0 / 49.0
	GRADA (HRSIM)	8.0 / 38.0	3.0 / 54.0	7.0 / 51.0
Qwen2.5-7b-Instruct	None	58.0 / 28.0	59.0 / 20.0	53.0 / 28.0
	HLATR	59.0 / 31.0	53.0 / 29.0	38.0 / 34.0
	BGE-reranker	57.0 / 34.0	49.0 / 31.0	43.0 / 31.0
	Keyword	3.0 / 15.0	0.0 / 20.0	10.0 / 22.0
	GRADA (D2DSIM-EBD)	41.0 / 33.0	16.0 / 46.0	19.0 / 35.0
	GRADA (D2DSIM-BM25)	35.0 / 35.0	11.0 / 46.0	17.0 / 37.0
	GRADA (HRSIM)	6.0 / 33.0	7.0 / 43.0	12.0 / 39.0

Table 10: ASR and EM (%) for various defense methods on PoisonedRAG(Hotflip).

Model	Defense	HotpotQA ASR ↓ / EM ↑	NQ ASR ↓ / EM ↑	MS-MARCO ASR ↓ / EM ↑
GPT-4o	None	99.0 / 0.0	96.0 / 4.0	80.0 / 11.0
	HLATR	97.0 / 2.0	78.0 / 15.0	53.0 / 32.0
	BGE-reranker	87.0 / 8.0	35.0 / 39.0	24.0 / 51.0
	Keyword Aggregation	0.0 / 57.0	0.0 / 44.0	0.0 / 45.0
	GRADA (D2DSIM-EBD)	31.0 / 43.0	2.0 / 57.0	2.0 / 60.0
	GRADA (D2DSIM-BM25)	41.0 / 37.0	10.0 / 58.0	0.0 / 68.0
	GRADA (HRSIM)	25.0 / 43.0	1.0 / 63.0	0.0 / 68.0
Llama3.1-70b-Instruct	None	100.0 / 0.0	98.0 / 2.0	88.0 / 8.0
	HLATR	100.0 / 0.0	92.0 / 5.0	84.0 / 9.0
	BGE-reranker	98.0 / 2.0	42.0 / 39.0	42.0 / 32.0
	Keyword (Xiang et al., 2024)	0.0 / 25.0	0.0 / 35.0	0.0 / 60.0
	GRADA (D2DSIM-EBD)	33.0 / 29.0	2.0 / 56.0	3.0 / 48.0
	GRADA (D2DSIM-BM25)	42.0 / 25.0	12.0 / 52.0	2.0 / 53.0
	GRADA (HRSIM)	26.0 / 32.0	1.0 / 56.0	1.0 / 54.0
Qwen2.5-7b-Instruct	None	4.0 / 23.0	6.0 / 16.0	5.0 / 27.0
	HLATR	15.0 / 24.0	18.0 / 13.0	18.0 / 21.0
	BGE-reranker	23.0 / 17.0	23.0 / 32.0	20.0 / 32.0
	Keyword (Xiang et al., 2024)	0.0 / 14.0	0.0 / 19.0	0.0 / 24.0
	GRADA (D2DSIM-EBD)	12.0 / 36.0	2.0 / 46.0	3.0 / 41.0
	GRADA (D2DSIM-BM25)	15.0 / 28.0	8.0 / 42.0	1.0 / 44.0
	GRADA (HRSIM)	8.0 / 36.0	2.0 / 47.0	1.0 / 44.0

Table 11: ASR and EM (%) for various defense methods on PIA.

Model	Defense	HotpotQA ASR ↓ / EM ↑	NQ ASR ↓ / EM ↑	MS-MARCO ASR ↓ / EM ↑
GPT-4o	None	68.0 / 4.0	37.0 / 21.0	44.0 / 37.0
	HLATR	64.0 / 9.0	39.0 / 25.0	38.0 / 42.0
	BGE-reranker	31.0 / 39.0	23.0 / 38.0	41.0 / 35.0
	Keyword Aggregation	0.0 / 45.0	0.0 / 43.0	0.0 / 43.0
	GRADA (D2DSIM-EBD)	27.0 / 32.0	1.0 / 48.0	10.0 / 45.0
	GRADA (D2DSIM-BM25)	8.0 / 41.0	2.0 / 51.0	1.0 / 63.0
	GRADA (HRSIM)	4.0 / 42.0	0.0 / 49.0	0.0 / 63.0
Llama3.1-70b-Instruct	None	8.0 / 46.0	6.0 / 39.0	11.0 / 48.0
	HLATR	9.0 / 50.0	15.0 / 38.0	11.0 / 51.0
	BGE-reranker	20.0 / 45.0	22.0 / 39.0	12.0 / 53.0
	Keyword (Xiang et al., 2024)	0.0 / 23.0	0.0 / 29.0	0.0 / 53.0
	GRADA (D2DSIM-EBD)	6.0 / 40.0	3.0 / 45.0	2.0 / 49.0
	GRADA (D2DSIM-BM25)	28.0 / 24.0	6.0 / 49.0	1.0 / 55.0
	GRADA (HRSIM)	17.0 / 25.0	0.0 / 51.0	0.0 / 54.0
Qwen2.5-7b-Instruct	None	1.0 / 26.0	1.0 / 20.0	3.0 / 38.0
	HLATR	1.0 / 31.0	6.0 / 13.0	3.0 / 35.0
	BGE-reranker	7.0 / 29.0	19.0 / 27.0	9.0 / 39.0
	Keyword (Xiang et al., 2024)	0.0 / 4.0	0.0 / 5.0	0.0 / 5.0
	GRADA (D2DSIM-EBD)	9.0 / 23.0	6.0 / 32.0	4.0 / 46.0
	GRADA (D2DSIM-BM25)	25.0 / 27.0	6.0 / 35.0	0.0 / 45.0
	GRADA (HRSIM)	18.0 /27.0	0.0 / 37.0	0.0 / 50.0

Table 12: ASR and EM (%) for various defense methods on Phantom.