Typos that Broke the RAG's Back: Genetic Attack on RAG Pipeline by Simulating Documents in the Wild via Low-level Perturbations

Sukmin Cho[1] **Soyeong Jeong**[2] **Jeongyeon Seo**[1] **Taeho Hwang**[1] **Jong C. Park**[1*]
School of Computing[1] Graduate School of AI[2]
Korea Advanced Institute of Science and Technology[1,2]
{nelllpic,starsuzi,yena.seo,doubleyyh,jongpark}@kaist.ac.kr

Abstract

The robustness of recent Large Language Models (LLMs) has become increasingly crucial as their applicability expands across various domains and real-world applications. Retrieval-Augmented Generation (RAG) is a promising solution for addressing the limitations of LLMs, yet existing studies on the robustness of RAG often overlook the interconnected relationships between RAG components or the potential threats prevalent in real-world databases, such as minor textual errors. In this work, we investigate two underexplored aspects when assessing the robustness of RAG: 1) vulnerability to noisy documents through low-level perturbations and 2) a holistic evaluation of RAG robustness. Furthermore, we introduce a novel attack method, the Genetic Attack on RAG (*GARAG*), which targets these aspects. Specifically, *GARAG* is designed to reveal vulnerabilities within each component and test the overall system functionality against noisy documents. We validate RAG robustness by applying our *GARAG* to standard QA datasets, incorporating diverse retrievers and LLMs. The experimental results show that *GARAG* consistently achieves high attack success rates. Also, it significantly devastates the performance of each component and their synergy, highlighting the substantial risk that minor textual inaccuracies pose in disrupting RAG systems in the real world. Code is available at https://github.com/zomss/GARAG.

1 Introduction

Large Language Models (LLMs) (Brown et al., 2020; OpenAI, 2023b) have enabled remarkable advances in diverse Natural Language Processing (NLP) tasks, especially in Question-Answering (QA) tasks (Joshi et al., 2017; Kwiatkowski et al., 2019). Despite these advances, however, LLMs face challenges in having to adapt to ever-evolving

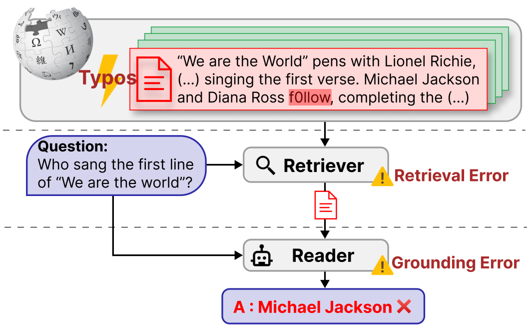

Figure 1: Impact of noisy documents in real-world databases on the RAG system: The retriever selects a noisy document, causing the reader to produce incorrect answers.

or long-tailed knowledge due to their limited parametric memory (Kasai et al., 2023; Mallen et al., 2023), resulting in a hallucination where the models generate convincing yet factually incorrect text (Li et al., 2023a). Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) has emerged as a promising solution by utilizing a retriever to fetch enriched knowledge from external databases, thus enabling accurate, relevant, and up-to-date response generation. Specifically, RAG has shown its superior performance across diverse knowledge-intensive tasks (Lewis et al., 2020; Lazaridou et al., 2022; Jeong et al., 2024), leading to its integration as a core component in various real-world APIs (Qin et al., 2024; Chase, 2022; OpenAI, 2023a). Given its extensive applications, ensuring robustness under diverse conditions of real-world scenarios becomes critical for safe deployment. Thus, assessing potential vulnerabilities within the overall RAG system is vital, particularly by assessing its components: the retriever and the reader.

However, existing studies on assessing the robustness of RAG often focus solely on either retrievers (Zhong et al., 2023; Zou et al., 2024; Long et al., 2024) or readers (Li et al., 2023b; Wang et al., 2023; Zhu et al., 2023). The robustness of a single component might only partially capture the complexities of RAG systems, where the retriever and reader work together in a sequential flow, which is

* Corresponding author

Findings of the Association for Computational Linguistics: EACL 2024, pages 2826–2844

crucial for optimal performance. In other words, the reader's ability to accurately ground information significantly depends on the retriever's capability of sourcing query-relevant documents (Baek et al., 2023; Lee et al., 2023). Thus, it is important to consider both components simultaneously when evaluating the robustness of an RAG system.

While concurrent work has shed light on the sequential interaction between two components, they have primarily evaluated the performance of the reader component given the high-level perturbed errors within retrieved documents, such as context relevance or counterfactual information (Thakur et al., 2023; Chen et al., 2024; Cuconasu et al., 2024). However, they have overlooked the impact of low-level errors, such as textual typos due to human mistakes or preprocessing inaccuracies in retrieval corpora, which often occur in real-world scenarios (Piktus et al., 2021; Le et al., 2023). Additionally, LLMs, commonly used as readers, struggle to produce accurate predictions when confronted with textual errors (Zhu et al., 2023; Wang et al., 2023). Note that these are the practical issues that can affect the performance of any RAG system in real-world scenarios, as illustrated in Figure 1. Therefore, to deploy a more realistic RAG system, we should consider: "*Can minor document typos comprehensively disrupt both the retriever and reader components in RAG systems?*"

In this paper, we evaluate the RAG system's robustness against textual typos in the database by generating a perturbed counterpart of the clean document retrieved for a given query. Initially, we establish two attack objectives to qualitatively measure the negative impact of the adversarial document on the RAG system's retrieval and grounding capabilities. To comprehensively assess system resilience under these objectives, we propose a novel black-box adversarial attack method, *GARAG*, which uses a genetic algorithm to search for the most adversarial document with low values for both loss objectives among the perturbed documents. The method begins by generating an initial population of adversarial documents by injecting minor textual errors into the original document while ensuring that answer tokens remain unaltered. Through an iterative process of mutation, crossover, and selection to refine the population, the method searches for the most adversarial document for a given query by effectively exploring the vast search space of typos space and exploiting the most adversarial documents. To sum up, *GARAG* assesses the holistic robustness of an RAG system against minor textual errors, offering insights into the system's resilience through iterative adversarial refinement.

We validate our method on three standard QA datasets (Joshi et al., 2017; Kwiatkowski et al., 2019; Rajpurkar et al., 2016), with diverse retrievers (Karpukhin et al., 2020; Izacard et al., 2022) and LLMs (Touvron et al., 2023; Chiang et al., 2023; Jiang et al., 2023). The experimental results reveal that adversarial documents with low-level perturbation generated by *GARAG* significantly induce retrieval and grounding errors, achieving a high attack success rate of approximately 70%, along with a significant reduction in the performance of each component and the overall system. Our analyses also highlight that lower perturbation rates pose a greater threat to the RAG system, emphasizing the challenges of mitigating such inconspicuous yet critical vulnerabilities.

Our contributions in this paper are threefold:

- We point out that the RAG system is vulnerable to minor but frequent textual errors within the documents, prevalent in real-world scenarios.
- We propose a black-box adversarial attack method, *GARAG*, based on a genetic algorithm searching for adversarial documents targeting both components within RAG simultaneously.
- We experimentally show that *GARAG* effectively attacks the RAG system with significant performance degradation, validating the vulnerability to textual typos.

2 Related Work

2.1 Robustness in RAG

The robustness of RAG, characterized by its ability to fetch and incorporate external information dynamically, has gained much attention for its critical role in real-world applications (Chase, 2022; Liu, 2022; OpenAI, 2023a). However, previous studies concentrated on the robustness of individual components within RAG systems, either retriever or reader. The vulnerability of the retriever is captured by injecting adversarial documents, specially designed to disrupt the retrieval capability, into retrieval corpora (Zhong et al., 2023; Zou et al., 2024; Long et al., 2024). Additionally, the robustness of LLMs, often employed as readers, has been critically examined for their resistance to out-of-distribution data and adversarial attacks (Wang et al., 2021; Li et al., 2023b; Wang et al., 2023;

Zhu et al., 2023). However, these studies overlook the sequential interaction between the retriever and reader components, thus not fully addressing the overall robustness of RAG systems.

In response, there is an emerging consensus on the need to assess the holistic robustness of RAG, with a particular emphasis on the sequential interaction of the retriever and reader (Thakur et al., 2023; Chen et al., 2024). They point out that RAG's vulnerabilities stem from retrieval inaccuracies and inconsistencies in how the reader interprets retrieved documents. Specifically, the reader generates incorrect responses if the retriever fetches partially (or entirely) irrelevant or counterfactual documents within the retrieved set. The solutions to these challenges range from prompt design (Cho et al., 2023; Press et al., 2023) and plug-in models (Baek et al., 2023) to specialized language models for enhancing RAG's performance (Yoran et al., 2024; Asai et al., 2024). However, they focus on the high-level errors within retrieved documents, which may overlook more subtle yet realistic low-level errors frequently encountered in the real world.

In this study, we spotlight a novel vulnerability in RAG systems related to low-level textual errors found in retrieval corpora, often originating from human mistakes or preprocessing inaccuracies (Thakur et al., 2021; Piktus et al., 2021; Le et al., 2023). Specifically, Faruqui et al. (2018) pointed out that Wikipedia, a widely used retrieval corpus, frequently contains minor errors within its contents. Therefore, we focus on a holistic evaluation of the RAG system's robustness against pervasive low-level text perturbations, emphasizing the critical need for systems that can maintain comprehensive effectiveness for real-world data.

2.2 Adversarial Attacks in NLP

Adversarial attacks involve generating adversarial samples designed to meet specific objectives to measure the robustness of models (Zhang et al., 2020). In NLP, such attacks use a transformation function to inject perturbations into text, accompanied by a search algorithm that identifies the most effective adversarial sample.

The operations of the transformation function can be categorized into high-level and low-level perturbations. High-level perturbations leverage semantic understanding (Alzantot et al., 2018; Ribeiro et al., 2018; Jin et al., 2020), while low-level perturbations are based on word or character-level changes, simulating frequently occurring errors (Eger et al., 2019; Eger and Benz, 2020; Le et al., 2022; Formento et al., 2023).

Search algorithms aim to find optimal adversarial samples by identifying victim tokens in the original document, chosen based on their word importance as calculated by a single target model. For instance, deletion-based scoring (Gao et al., 2018) identifies important tokens by assessing increases in attack objectives when a token is deleted, while gradient-based scoring (Yoo and Qi, 2021a) uses the gradient of the attack objective for each token. Since these methods are unsuitable for multi-objective scenarios, a genetic algorithm that randomly selects tokens with elaborate exploitation is more effective (Alzantot et al., 2018; Zang et al., 2020; Williams and Li, 2023). To evaluate the robustness of the overall RAG system, which has non-differentiable and dual objectives for a retriever and a reader, we propose a novel attack algorithm incorporating a genetic algorithm.

3 Method

Here, we introduce our problem formulation and a novel attack method, *GARAG*. Further details of the proposed method are described in Appendix A.

3.1 Problem Formulation

Pipeline of RAG. Let $\boldsymbol{q}$ be a query the user requests. In a RAG system, the retriever first fetches the query-relevant document $\boldsymbol{d}$, then the reader generates the answer grounded on document-query pair $(\boldsymbol{d}, \boldsymbol{q})$. The retriever, parameterized with $\phi = (\phi_d, \phi_q)$, identifies the most relevant document in the database. The relevance score r is computed by the dot product of the embeddings for document $\boldsymbol{d}$ and query $\boldsymbol{q}$, as $r_\phi(\boldsymbol{d}, \boldsymbol{q}) = \texttt{Enc}(\boldsymbol{d}; \phi_d) \cdot \texttt{Enc}(\boldsymbol{q}; \phi_q)$. Finally, the reader, using an LLM parameterized with θ, generates the answer $\boldsymbol{a}$ from the document-query pair $(\boldsymbol{d}, \boldsymbol{q})$, as $\boldsymbol{a} = \texttt{LLM}(\boldsymbol{d}, \boldsymbol{q}; \theta)$.

Adversarial Document Generation. To simulate the adversarial document having typical noise encountered in real-world scenarios, we introduce low-level perturbations to mimic these conditions. We generate an adversarial document $\boldsymbol{d}'$ by transforming the clean document $\boldsymbol{d}$ using a function f that alters each token d into a perturbed version d'. The function f randomly applies one of several operations — inner-shuffling, truncation, keyboard errors, or natural typos — to each token, then outputs the perturbed token: $d' = f(d)$. This randomness reflects the unpredictable nature of textual typos. Therefore, we explore a broad search space of potential adversarial documents generated from $\boldsymbol{d}$ using f to identify the adversarial document for the RAG system,

Attack Objective on RAG. To identify an adversarial document $\boldsymbol{d'}$ that challenges the capabilities of the RAG, we compare its negative impact against the original document $\boldsymbol{d}$ for a given query $\boldsymbol{q}$. The goal is for $\boldsymbol{d'}$ to divert attention from $\boldsymbol{d}$, ensuring that $\boldsymbol{d}$ no longer appears as the top result for $\boldsymbol{q}$. Additionally, $\boldsymbol{d'}$ should mislead LLM into generating an incorrect answer $\boldsymbol{a'}$ when paired with $(\boldsymbol{d^*}, \boldsymbol{q})$. To measure this negative impact, we use two loss objectives: the Relevance Score Ratio (RSR) and the Generation Probability Ratio (GPR) for retrieval and grounding, respectively.

The RSR calculates the ratio of the relevance score[1] from the adversarial document $\boldsymbol{d'}$ to the score from the original document $\boldsymbol{d}$ for the given query $\boldsymbol{q}$. Conversely, the GPR calculates the ratio of the generation probability[2] of the correct answer $\boldsymbol{a}$ from the original pair $(\boldsymbol{d}, \boldsymbol{q})$ to the probability from the adversarial pair $(\boldsymbol{d'}, \boldsymbol{q})$. These two metrics are formally represented as:

$$\mathcal{L}_{\text{RSR}}(\boldsymbol{d'}) = \frac{e^{r_\phi(\boldsymbol{d},\boldsymbol{q})}}{e^{r_\phi(\boldsymbol{d'},\boldsymbol{q})}}, \mathcal{L}_{\text{GPR}}(\boldsymbol{d'}) = \frac{p_\theta(\boldsymbol{a}|\boldsymbol{d'},\boldsymbol{q})}{p_\theta(\boldsymbol{a}|\boldsymbol{d},\boldsymbol{q})}. \tag{1}$$

The values below 1 signify that a noisy document $\boldsymbol{d'}$ generated from the adversarial attack successfully satisfies the attack objectives of distracting the retriever and misleading LLM. Note that, as these objectives are designed for adversarial attacks, they don't directly align with each module's performance measured by conventional metrics.

Consequently, the search for an optimal adversarial document within the RAG system is defined as a dual-objective optimization problem, aiming to minimize both the RSR and GPR simultaneously:

$$\boldsymbol{d^*} = \underset{\boldsymbol{d'} \in D'}{\arg\min}(\mathcal{L}_{\text{RSR}}(\boldsymbol{d'}), \mathcal{L}_{\text{GPR}}(\boldsymbol{d'})) \tag{2}$$

This optimization problem involves dual-model environments, resulting in non-differentiable conditions. To design effective adversarial attack methods targeting the RAG system through noisy document simulation, these methods must address the challenges of dual-objective and dual-model optimization within a vast search space characterized by unpredictable and diverse textual typos.

[1]Given the potential for relevance scores to be negative, we have structured the term to guarantee positivity.

[2]The generation probability represents the joint probabilities over the answer tokens given a single document and a single question.

3.2 GARAG: Genetic Attack on RAG

In this work, we introduce a novel black-box adversarial attack method called *GARAG*, employing a genetic algorithm to address the dual-objective and dual-model optimization problem in a large search space. Initially, as shown in Figure 2, we divide the search space into four zones based on the attack objectives: safety, retrieval error, grounding error, and holistic error. The adversarial document should ideally be in a holistic error zone, where retrieval and grounding errors intersect, and should be closer to the origin, indicating a more significant negative impact on the RAG system. Then, our proposed method, *GARAG*, iteratively refines a population of adversarial documents, methodically moving them closer to the origin. This process involves exploring the search space to discover new adversarial documents and exploit the most adversarial ones with crossover, mutation, and selection steps.

Formally, given the query-document pair $(\boldsymbol{q}, \boldsymbol{d})$ where the document $\boldsymbol{d} = \{d_i\}_{i=1}^N$ is retrieved for the query $\boldsymbol{q}$, our objective is to generate the adversarial counterpart $\boldsymbol{d'}$ with $N \cdot pr_{pert}$ perturbed tokens, where pr_{pert} is a pre-defined hyperparameter and N is the number of tokens in $\boldsymbol{d}$. The steps, including crossover, mutation, and selection, are repeated N_{iter} times after initialization.

Initialization. Our attack begins with the initialization step. We first construct the initial population P_0, consisting of adversarial documents $\boldsymbol{d'_i}$, formalized as $P = \{\boldsymbol{d'_i}\}_{i=1}^S$, where S is the total number of documents in the population. In detail, generating the adversarial document $\boldsymbol{d'_i}$ involves selecting tokens for the attack, applying perturbations, and assembling the modified document. Initially, to determine which tokens to alter, a subset of indices I' containing $N \cdot pr_{\text{pert.}}$ indices is randomly selected from the complete set of token indices $I = \{1, \ldots, N\}$, where N represents the total number of tokens in the document $\boldsymbol{d}$. This selection is designed to exclude any indices that correspond to the correct answer $\boldsymbol{a}$ within the document, thus ensuring that the perturbations focus exclusively on assessing the impact of noise. Each selected token d_i is then transformed using the function f, yielding a perturbed version d'_i, for $i \in I' \subset I$. The final document $\boldsymbol{d'}$ merges the set of unaltered

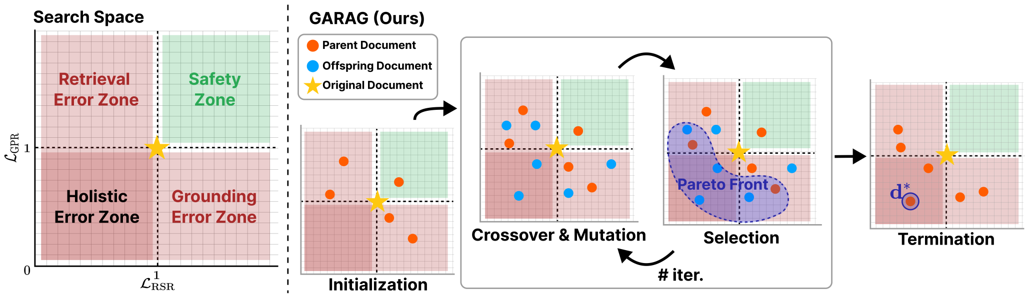

Figure 2: (Left) The search space formulated by our proposed attack objectives, $\mathcal{L}_{\text{RSR}}$ and $\mathcal{L}_{\text{GPR}}$. (Right) An overview of the iterative process implemented by our proposed method, *GARAG*.

tokens $T = \{d_i | i \notin I \setminus I'\}$ with the set of modified tokens, represented by $T' = \{d'_j | j \in I'\}$, forming $\boldsymbol{d'} = T \cup T'$. In Figure 2, the figure on the right shows the initialization step where the initial (parent) documents are represented as orange-colored dots, given the star-shaped original document.

Crossover & Mutation. Then, through the crossover and mutation steps, the adversarial documents are generated by balancing the exploitation of existing knowledge within the current population (parent documents) and the exploration of new documents (offspring documents). In detail, the crossover step generates offspring documents by recombining tokens from pairs of parent documents, incorporating their most effective adversarial features. Subsequently, the mutation step introduces new perturbations to some tokens in the offspring, aiming to explore genetic variations that are not present in the parent documents.

Formally, the crossover step selects N_{parents} pairs of parent documents from the population P. Let $\boldsymbol{d'_0}$ and $\boldsymbol{d'_1}$ be the selected parent documents along with their perturbed token sets T'_0 and T'_1, respectively. Then, the swapping tokens perturbed in each parent document generate the offspring documents, excluding those in the shared set $T'_0 \cap T'_1$. The number of swapping tokens is determined by the predefined crossover rate pr_{cross}, applied to the number of unique perturbed tokens in each document.

The mutation step selects two corresponding subsets of tokens, M from the original token set T and M' from the perturbed token set T', ensuring that both subsets are of equal size $|M| = |M'|$. The size of these subsets is determined by the predefined mutation probability $pr_{\text{mut.}}$, which is applied to $pr_{\text{pert.}} \cdot N$. Tokens $d_i \in M$ are altered using a perturbation function f, whereas tokens $d'_j \in M'$ are reverted to their original states d_j. Following this, the sets of unperturbed and perturbed tokens, T_{new} and T'_{new}, respectively, are updated to incorporate these modifications: $T_{\text{new}} = (T \setminus M) \cup M'$ and $T'_{\text{new}} = (T' \setminus M') \cup M$. The newly mutated document, $\boldsymbol{d'_{\text{new}}}$, is composed of the updated sets T_{new} and T'_{new}, and the offspring set O is then formed, comprising these mutated documents. The offspring documents are represented by blue-colored dots in the figure on the right in Figure 2.

Selection. The remaining step is to select the most optimal adversarial documents from the combined set $\hat{P} = P \cup O$, which includes both parent and offspring documents. Specifically, each document within $\hat{P}$ is evaluated against the two attack objectives, $\mathcal{L}_{\text{RSR}}$ and $\mathcal{L}_{\text{GPR}}$, to assess their effectiveness in the adversarial context. Therefore, we incorporate a non-dominated sorting strategy (Deb et al., 2002) to identify the optimal set of documents, known as the Pareto front. In this front, each document is characterized by having all objective values lower than those in any other set, as shown in the right of Figure 2. Then, the documents in the Pareto front will be located in a holistic error zone closer to the origin. Additionally, to help preserve diversity within the document population, we further utilize the crowding distance sorting strategy to identify adversarial documents that possess unique knowledge by measuring how isolated each document is relative to others. Then, the most adversarial document $\boldsymbol{d}^*$ is selected from a less crowded region of the Pareto front. Details of a non-dominated sorting algorithm are described in Appendix A.4.

Note that this process, including crossover, mutation, and selection steps, continues iteratively until a successful attack is achieved, where the selected adversarial document $\boldsymbol{d}^*$ prompts an incorrect answer a', as illustrated in the figure on the right in Figure 2. If the process fails to produce a successful attack, it persists through the predefined number of iterations, $N_{\text{iter.}}$.

Table 1: Results of adversarial attacks using *GARAG*, averaged across three datasets, NQ, TQA, and SQuAD. The most vulnerable results are in **bold**.

Retriever	LLM	Attack Success Ratio (↑)			End-to-End (↓)	
		ASR_R	ASR_L	ASR_T	EM	Acc
DPR	**Llama2-7b**	79.2	90.5	70.1	77.1	81.3
	Llama2-13b	78.4	**92.0**	70.8	81.9	87.3
	Vicuna-7b	88.7	80.7	69.8	57.2	79.3
	Vicuna-13b	88.8	81.6	70.8	58.4	83.2
	Mistral-7b	83.7	85.5	69.5	66.7	96.5
Contriever	**Llama2-7b**	85.3	91.0	**76.6**	75.0	79.6
	Llama2-13b	82.0	**92.0**	74.2	80.7	87.3
	Vicuna-7b	**92.1**	81.5	73.9	55.1	**76.9**
	Vicuna-13b	91.3	83.2	74.7	**53.5**	79.5
	Mistral-7b	89.2	86.6	75.9	63.1	95.3
w/o *GARAG*		-	-	-	100	100

Table 2: Retrieval performance under RAG system using Llama-7b when the adversarial documents generated by *GARAG* are injected into the retrieval corpus.

Dataset	Attacked	DPR			Contriever		
		MAP@100	NDCG@100	ASR_R	MAP@100	NDCG@100	ASR_R
NQ	✗	.417	.633	-	.248	.489	-
	✓	.356	.593	75.4	.219	.462	85.9
TQA	✗	.532	.740	-	.337	.696	-
	✓	.471	.696	78.2	.298	.559	84.9
SQD	✗	.321	.540	-	.267	.498	-
	✓	.279	.513	80.0	.223	.468	86.1

4 Experimental Setup

In this section, we describe the experimental setup.

4.1 Model

Retriever. We use two recent dense retrievers: **DPR** (Karpukhin et al., 2020), a supervised one trained on query-document pairs, and **Contriever** (Izacard et al., 2022), an unsupervised one.

Reader. Following concurrent work (Asai et al., 2024; Wang et al., 2024) that utilizes LLMs as readers for the RAG system, with parameters ranging from 7B to 13B, we have selected open-source LLMs of similar capacities: **Llama2** (Touvron et al., 2023), **Vicuna** (Chiang et al., 2023), and **Mistral** (Jiang et al., 2023). Each model has been either chat-versioned or instruction-tuned. To adapt these models for open-domain QA tasks, we employ a zero-shot prompting template for exact match QA derived from Wang et al. (2024).

4.2 Dataset

We leverage three representative QA datasets: **Natural Questions (NQ)** (Kwiatkowski et al., 2019), **TriviaQA (TQA)** (Joshi et al., 2017), and **SQuAD (SQD)** (Rajpurkar et al., 2016), following the setups of Karpukhin et al. (2020). To assess the robustness of the RAG system, we randomly extract 1,000 instances of the triple $(\boldsymbol{q}, \boldsymbol{d}, \boldsymbol{a})$. In each triple, $\boldsymbol{q}$ is a question from the datasets, $\boldsymbol{d}$ is a document from the top-100 documents retrieved from the Wikipedia corpus corresponding to $\boldsymbol{q}$, and $\boldsymbol{a}$ is the answer generated by the LLM, which is considered as correct for the specific question-document pair.

4.3 Evaluation Metric

To measure the effectiveness of *GARAG* and the actual impact of generated adversarial documents on RAG systems, we incorporate two types of metrics to show the effectiveness of the adversarial attacks and the end-to-end QA performance measuring the actual impact on the RAG system.

Attack Success Ratio (ASR). Attack Success Ratio (ASR) is the ratio of the generated documents from the adversarial attack, located in the holistic error zone (i.e., the values below 1 for $\mathcal{L}_{\text{RSR}}$ and $\mathcal{L}_{\text{GPR}}$). Specifically, ASR is for measuring the effectiveness of the proposed method addressing dual-objective optimization problems.

End-to-End Performance (E2E). To evaluate the impact of the adversarial document on RAG systems, we report it with standard QA metrics: **Exact Match (EM)** and **Accuracy (Acc)**. EM evaluates if a prediction precisely matches the correct answer, while Acc checks if the answer span is included in the predicted response. If the attack fails (i.e., either value for $\mathcal{L}_{\text{RSR}}$ or $\mathcal{L}_{\text{GPR}}$ exceeds 1), we transmit the original document $\boldsymbol{d}$ to LLM instead of the adversarial one $\boldsymbol{d'}$ during prediction.

4.4 Implementation Details

The proposed method, *GARAG*, was configured with hyperparameters: N_{iter} was set to 25, N_{parents} to 10, and S to 25. pr_{pert}, pr_{cross}, and pr_{mut} were set to 0.2, 0.2, and 0.4, respectively. The operations of perturbation function f in *GARAG* consist of the inner swap, truncate, keyboard typo, and natural typo, following Eger and Benz (2020)[3]. For computing resources, we use A100 GPU clusters.

5 Results

In this section, we show our experimental results with an in-depth analysis of the adversarial attack.

5.1 Main Result.

Table 1 shows our main results averaged over three datasets using *GARAG* with two metrics: attack

[3] https://github.com/yannikbenz/zeroe

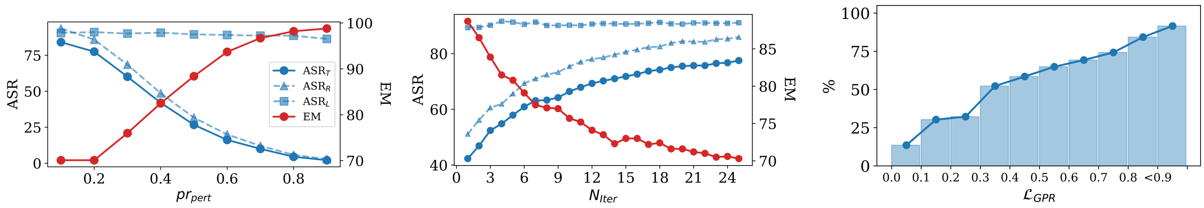

Figure 3: Adversarial attack analysis on the NQ dataset using Contriever and Llama2-7b: (Left) Variations in ASR and EM scores as the pr_{pert} increases from 0 to 0.9, with ASR shown in blue and EM in red. (Center) Variations in ASR and EM scores across increasing iterations (N_{iter}), also indicated in blue and red respectively. (Right) Distribution of correctness among predictions depending on $\mathcal{L}_{\text{GPR}}$.

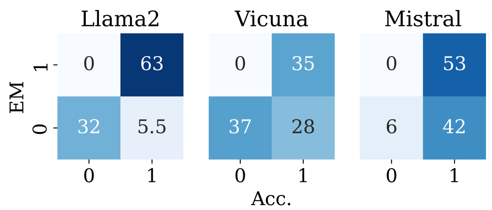

Figure 4: Confusion matrices of prediction from $\boldsymbol{d}^*$ across EM and Acc. on NQ with Contriever.

success ratio (ASR) and end-to-end performance (E2E). First, a notable success rate of over 70% across all scenarios indicates that *GARAG* effectively locates adversarial documents within the holistic error zone by simultaneously considering retrieval and reader errors. Additionally, we analyze the E2E performance to assess how adversarial attacks impact overall QA performance. Based on the EM metric, the performance of RAG systems decreased by an average of 30% and a maximum of close to 50% in all cases. These findings imply that noisy documents with minor errors, frequently found in the real world, can pose significant risks to downstream tasks using RAG.

Impact on Retrieval Ability. We qualitatively explored the impact of adversarial documents on the RAG system's retrieval ability. After injecting these documents into the original retrieval corpus, we evaluated the results using conventional IR metrics like MAP and NDCG. As shown in Table 2, the adversarial documents degrade retrieval performance across all scenarios, despite being assessed solely by the $\mathcal{L}_{\text{RSR}}$ in the *GARAG* process without considering the entire retriever corpus. Additionally, as DPR achieves better retrieval performance both before and after the attack, these results suggest that retrievers with superior retrieval performance tend to be more robust against typos.

Impact on Grounding Ability. We further analyze the response patterns of LLM to adversarial documents, categorizing the results based on EM and Acc as shown in Figure 4. For instance, an EM of 0 and Acc of 1 indicates that the response includes the correct answer along with irrelevant tokens, whereas an EM and Acc of 0 means that the response is entirely incorrect, likely a hallucination. First, Llama2 tends to produce exact matches more frequently, as evidenced by a high rate of (1,1) outcomes. but struggles with completely incorrect responses under adversarial conditions, indicated by a lower proportion of (0,1). By contrast, Mistral, despite fewer exact matches, consistently includes the correct answer span in its responses. These insights are vital for understanding how different models perform in realistic scenarios, especially when handling noisy or adversarially altered documents, highlighting the varied impacts of such conditions on LLMs.

Impact of pr_{pert} and N_{iter} Then, we further explore how varying the perturbation probability pr_{pert} or the number of iterations N_{iter} affects the attack outcomes. As the left and center figures of Figure 3 illustrate, there is an apparent correlation between the attack success rates for the retriever (ASR_R) and the entire pipeline (ASR_T). Moreover, the consistently high success rate for the LLM (ASR_L) across all cases highlights a significant vulnerability in the reader against typos. These findings highlight the critical role of the retriever as a first line of defense in the RAG system. Interestingly, in the left figure of Figure 3, the results indicate that a lower proportion of perturbation within a document leads to a more disruptive impact on the RAG system. These experimental results suggests that documents with a few typos, which are common in the wild, could have a more detrimental effect on performance.

This phenomenon is counter-intuitive, as other attack approaches typically show that more attack vectors lead to stronger adversarial effects. We speculate that this occurs because the training data for neural retrievers generally consists of clean documents without typos, making it easier for the retriever to identify and reject documents with many errors. In contrast, LLMs, which are also trained

Table 3: Ablation study of *GARAG* on NQ with Contriever and Llama-7b.

	ASR			E2E
	ASR_R	ASR_L	ASR_T	EM
GARAG	85.9	91.1	77.5	70.1
Low-level Perturbations included f				
Natural Typo	88.8	90.0	78.8	75.4
Keyboard Typo	84.6	91.4	76.2	71.2
Truncate	89.2	90.2	79.4	71.4
Inner Swap	83.4	87.8	71.4	78.0
Low-level Perturbations not included f				
Punc.	93.0	93.7	86.7	68.9
Phonetic.	84.7	92.1	76.8	70.0
Visual.	77.7	90.5	68.8	72.5

Table 4: Adversarial attack on paraphrased query on NQ with Contriever and Llama-7b.

Paraphrased	Attacked	ASR_R	ASR_L	ASR_T	EM
✗	✗	-	-	-	100
✗	✓	85.9	91.1	77.5	70.1
✓	✗	-	-	-	79.1
✓	✓	72.8	62.5	44.1	75.1

on clean texts, struggle to generate correct answers when typos are present. The errors cause the document to lose key information or clarity, making it difficult for the model to infer the correct answer.

Impact of Lowering $\mathcal{L}_{\text{GPR}}$. Since the value of $\mathcal{L}_{\text{GPR}}$ does not directly indicate the likelihood of generating incorrect answers with auto-regressive models, we analyze the correlation between the likelihood of generating incorrect answers and $\mathcal{L}_{\text{GPR}}$. As illustrated in the right panel of Figure 3, we categorize predictions into buckets based on their $\mathcal{L}_{\text{GPR}}$ ranges and calculate the proportion of incorrect answers within each bucket. The results validate our objective design, demonstrating that a lower $\mathcal{L}_{\text{GPR}}$ value is associated with a higher likelihood of incorrect responses.

5.2 Analysis

Evaluation on Paraphrased Query. To create a more realistic scenario, we tested the effect of noisy documents with paraphrased queries that were not used in the adversarial attack. After generating an adversarial document for a given document-query pair, we paraphrased the query using GPT-3.5 (Brown et al., 2020). These paraphrased queries, while not part of the adversarial document generation, still seek the same answers as the original ones. As shown in Table 4, our results demonstrate the robustness of adversarial documents generated by *GARAG*. While these documents are less effective against paraphrased queries, resulting in lower ASR and higher EM scores, they still degrade RAG system performance after attacks. The paraphrased queries also destabilize RAG systems, underscoring their vulnerability in dynamic, real-world settings like human-RAG system interactions.

Table 5: Comparison with other search methods on NQ with Contriever and Llama-7b.

	ASR			E2E
	ASR_R	ASR_L	ASR_T	EM
GARAG	85.9	91.1	77.5	70.1
***GARAG* on Retriever**	96.6	18.0	18.0	94.4
***GARAG* on LLM**	33.2	100.0	33.2	85.2
DS on Retriever	94.8	56.6	53.8	89.2
DS on LLM	16.0	100.0	16.0	90.4
GS on Retriever	26.5	75.0	4.6	93.2
GS on LLM	4.9	96.2	17.8	97.2

Types of Low-level Perturbation. Table 3 presents the results of an ablation study on the operations included and excluded in the transformation function f. Using multiple operations in f as the default setup consistently outperformed all single operations included in f, highlighting *GARAG*'s ability to exploit promising areas in a vast search space. Furthermore, the other types of low-level perturbations not initially included in f—such as punctuation insertion, phonetic similarity, and visual similarity—successfully comprise the RAG system with a significant performance drop. Notably, punctuation insertion alone compromised the system in 86% of the attacks, demonstrating *GARAG*'s effectiveness in leveraging diverse perturbations for attacks.

Comparison with Other Search Methods. We validated the effectiveness of our proposed method, *GARAG*, by comparing it with two search methods based on word importance calculated through deletion scoring (DS) and gradient scoring (GS). Note that both methods can target only a single module. As shown in Table 5, these single-targeted methods fail to comprehensively search for adversarial documents across all modules. Even when implemented for single-module attacks, *GARAG* achieves significantly higher ASR and lower E2E than other methods, demonstrating the genetic algorithm's effectiveness. This underscores the importance of attacking both retriever and reader rather than targeting a single module.

Defense Strategy. Various defense mechanisms against adversarial attacks in NLP have been proposed. Adversarial training, fine-tuning the model

Table 6: Case study with Contriever and Llama-7b, where perturbed texts are in red and correct answers are in blue .

Question	Who sang the first line of 'We Are The World'?
Noisy Document	We Are the World lines in the sing's repetitive chorus proclaim, "We are the world, we are the onss who make a brighger day, so letś start giving". "We Are the World" pens with Lionel Richie , Stevie Wonder , Paul Simon , Kenny Rogers , James Ingram , Tina Turner , and Billy Joel singing the first verse. Michael Jackson and Diana Ross f0llow , completing the first choruc together. Dionne Warwick, Willif Nelson, and Al Jarreau singe the second vers4 , before Bruce Springsteen, Kenny Loggins, Steve Perry, and Daryl Hall go through the second chorus.
Answer	Stevie Wonder, Tina Turner, Billy Joel, James Ingram, Kenny Rogers, Paul Simon, Lionel Richie
Prediction	Michael Jackson

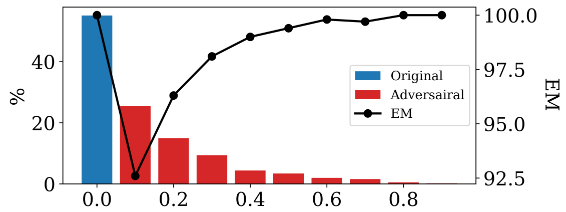

Figure 5: Distribution of grammatically correct documents among d^* on NQ with the Contriever and Llama2-7b.

on adversarial samples, is a popular approach (Yoo and Qi, 2021b). However, this strategy is not practically viable for RAG systems, given the prohibitive training costs associated with models exceeding a billion parameters. Alternatively, a grammar checker is an effective defense against low-level perturbations within documents (Formento et al., 2023). Our analysis, depicted in Figure 5, compares the grammatical correctness of original and adversarial documents via grammar checker model [4] presented in Dehghan et al. (2022). It reveals that approximately 50% of the original and clean samples are determined to be the nosiy documents containing grammatical errors. Also, even within the adversarial set, about 25% of the samples maintain grammatical correctness at a low perturbation level. This observation highlights a critical limitation: relying solely on a grammar checker would result in dismissing many original documents and accepting some adversarial ones. Consequently, this underscores the limitations of grammar checkers as a standalone defense and points to more sophisticated and tailored defense strategies.

Case Study. We further qualitatively assess the impact of low-level textual perturbations within a document in Table 6. Note that since we ensure that the answer spans remain unperturbed, LLMs should ideally generate correct answers. However, interestingly, an LLM fails to identify the correct answers, which are mentioned in the document, but instead generates an incorrect answer, "Michael Jackson," included in the document. To this end, we would like to emphasize that addressing typographical errors is a complex challenge that requires many considerations in defense against the threat of typos, which seems relatively trivial. In our all experiments, we didn't perturb the tokens included in the correct answer span, as shown in Table 6, and we empirically validated that RAG systems often can't generate correct answers from the document, even including the correct answers. This poses a critical question: *Should we discard such documents because of typographical errors or find ways to use this information effectively within them?* These considerations highlight the need for comprehensive and sophisticated defense strategies, underscoring the ongoing vulnerability within RAG systems.

In Appendix B, we provide detailed results of adversarial attacks for each dataset and analysis including comparing high-level perturbation attacks and attacking closed-source models.

6 Conclusion

In this work, we highlighted the importance of assessing the overall robustness of the retriever and reader components within the RAG system, particularly against noisy documents containing minor typos that are common in real-world databases. Specifically, we proposed two objectives to evaluate the resilience of each component, focusing on their sequential dependencies. Furthermore, to simulate real-world noises with low-level perturbations, we introduced a novel adversarial attack method, *GARAG*, incorporating a genetic algorithm. Our findings indicate that noisy documents critically hurt the RAG system, significantly degrading its performance. Although the retriever serves as a protective barrier for the reader, it still remains susceptible to minor disruptions. Our *GARAG* shows promise as an adversarial attack strategy when assessing the holistic robustness of RAG systems against various low-level perturbations.

[4] https://huggingface.co/imohammad12/GRS-Grammar-Checker-DeBerta

Acknowledgement

This work was supported by Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea government (No. 2018-0-00582, Prediction and augmentation of the credibility distribution via linguistic analysis and automated evidence document collection) and the Artificial intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT (MSIT, Korea) & Gwangju Metropolitan City.

Limitation

In this work, we explored the robustness of the RAG system by using various recent open-source LLMs of different sizes, which are widely used as reader components in this system. However, due to our limited academic budget, we could not include much larger black-box LLMs such as the GPT series models, which have a hundred billion parameters. We believe that exploring the robustness of these LLMs as reader components would be a valuable line of future work. Furthermore, *GARAG* aims for the optimal adversarial document to be located within a holistic error zone, by simultaneously considering both retrieval and grounding errors. However, we would like to note that even though the adversarial document is located within the holistic error zone, this does not necessarily mean that the reader will always generate incorrect answers for every query, due to the auto-regressive nature of how reader models generate tokens. Nevertheless, as shown in the right figure of Figure 3 and discussed in its analysis, we would like to emphasize that there is a clear correlation: a lower $\mathcal{L}_{\text{GPR}}$ value is associated with a higher likelihood of incorrect responses.

Ethics Statement

We designed a novel attack strategy for the purpose of building robust and safe RAG systems when deployed in the real world. However, given the potential for malicious users to exploit our *GARAG* and deliberately attack the system, it is crucial to consider these scenarios. Therefore, to prevent such incidents, we also present a defense strategy, detailed in Figure 5 and its analysis. Additionally, we believe that developing a range of defense strategies remains a critical area for future work.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani B. Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2890–2896. Association for Computational Linguistics.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Jinheon Baek, Soyeong Jeong, Minki Kang, Jong C. Park, and Sung Ju Hwang. 2023. Knowledge-augmented language model verification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1720–1736. Association for Computational Linguistics.

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, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Harrison Chase. 2022. LangChain.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 17754–17762. AAAI Press.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Sukmin Cho, Jeongyeon Seo, Soyeong Jeong, and Jong C. Park. 2023. Improving zero-shot reader by reducing distractions from irrelevant documents in open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP*

2023, Singapore, December 6-10, 2023, pages 3145–3157. Association for Computational Linguistics.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for RAG systems. *arXiv preprint arXiv:2401.14887*, abs/2401.14887.

Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.*, 6(2):182–197.

Mohammad Dehghan, Dhruv Kumar, and Lukasz Golab. 2022. GRS: combining generation and revision in unsupervised sentence simplification. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 949–960. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 31–36. Association for Computational Linguistics.

Steffen Eger and Yannik Benz. 2020. From hero to zéroe: A benchmark of low-level adversarial attacks. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020*, pages 786–803. Association for Computational Linguistics.

Steffen Eger, Gözde Gül Sahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1634–1647. Association for Computational Linguistics.

Manaal Faruqui, Ellie Pavlick, Ian Tenney, and Dipanjan Das. 2018. Wikiatomicedits: A multilingual corpus of wikipedia edits for modeling language and discourse. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 305–315. Association for Computational Linguistics.

Brian Formento, Chuan-Sheng Foo, Anh Tuan Luu, and See-Kiong Ng. 2023. Using punctuation as an adversarial attack on deep learning-based NLP systems: An empirical study. In *Findings of the Association for Computational Linguistics: EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 1–34. Association for Computational Linguistics.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops, SP Workshops 2018, San Francisco, CA, USA, May 24, 2018*, pages 50–56. IEEE Computer Society.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. 2024. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*, abs/2310.06825.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui.

2023. Realtime QA: what's the answer right now? In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. 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. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*, abs/2203.05115.

Thai Le, Jooyoung Lee, Kevin Yen, Yifan Hu, and Dongwon Lee. 2022. Perturbations in the wild: Leveraging human-written text perturbations for realistic adversarial attack and defense. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2953–2965. Association for Computational Linguistics.

Thai Le, Yiran Ye, Yifan Hu, and Dongwon Lee. 2023. Cryptext: Database and interactive toolkit of human-written text perturbations in the wild. In *39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, CA, USA, April 3-7, 2023*, pages 3639–3642. IEEE.

Hyunji Lee, Se June Joo, Chaeeun Kim, Joel Jang, Doyoung Kim, Kyoung-Woon On, and Minjoon Seo. 2023. How well do large language models truly ground? *arXiv preprint arXiv:2311.09069*, abs/2311.09069.

Patrick S. H. 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 *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. Halueval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6449–6464. Association for Computational Linguistics.

Xinzhe Li, Ming Liu, Shang Gao, and Wray L. Buntine. 2023b. A survey on out-of-distribution evaluation of neural NLP models. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 6683–6691. ijcai.org.

Jerry Liu. 2022. LlamaIndex.

Quanyu Long, Yue Deng, Leilei Gan, Wenya Wang, and Sinno Jialin Pan. 2024. Backdoor attacks on dense passage retrievers for disseminating misinformation. *arXiv preprint arXiv:2402.13532*, abs/2402.13532.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9802–9822. Association for Computational Linguistics.

OpenAI. 2023a. Chatgpt plugins.

OpenAI. 2023b. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, abs/2303.08774.

Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Dmytro Okhonko, Samuel Broscheit, Gautier Izacard, Patrick S. H. Lewis, Barlas Oguz, Edouard Grave, Wen-tau Yih, and Sebastian Riedel. 2021. The web is your oyster - knowledge-intensive NLP against a very large web corpus. *arXiv preprint arXiv:2112.09924*, abs/2112.09924.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5687–5711. Association for Computational Linguistics.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, dahai li, Zhiyuan Liu, and Maosong Sun. 2024. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *The Twelfth International Conference on Learning Representations*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 856–865. Association for Computational Linguistics.

Nandan Thakur, Luiz Bonifacio, Xinyu Zhang, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Boxing Chen, Mehdi Rezagholizadeh, and Jimmy Lin. 2023. Nomiracl: Knowing when you don't know for robust multilingual retrieval-augmented generation. *arXiv preprint arXiv:2312.11361*, abs/2312.11361.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, abs/2307.09288.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, Sang T. Truong, Simran Arora, Mantas Mazeika, Dan Hendrycks, Zinan Lin, Yu Cheng, Sanmi Koyejo, Dawn Song, and Bo Li. 2023. Decodingtrust: A comprehensive assessment of trustworthiness in GPT models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Yuhao Wang, Ruiyang Ren, Junyi Li, Wayne Xin Zhao, Jing Liu, and Ji-Rong Wen. 2024. REAR: A relevance-aware retrieval-augmented framework for open-domain question answering. *arXiv preprint arXiv:2402.17497*, abs/2402.17497.

Phoenix Neale Williams and Ke Li. 2023. Black-box sparse adversarial attack via multi-objective optimisation CVPR proceedings. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 12291–12301. IEEE.

Jin Yong Yoo and Yanjun Qi. 2021a. Towards improving adversarial training of NLP models. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 945–956. Association for Computational Linguistics.

Jin Yong Yoo and Yanjun Qi. 2021b. Towards improving adversarial training of NLP models. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 945–956. Association for Computational Linguistics.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6066–6080. Association for Computational Linguistics.

Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol.*, 11(3):24:1–24:41.

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, EMNLP 2023, Singapore, December 6-10, 2023*, pages 13764–13775. Association for Computational Linguistics.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, and Xing Xie. 2023. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv preprint arXiv:2306.04528*, abs/2306.04528.

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

A Implementation Detail

A.1 Operations

We explore four types of low-level perturbations, capturing the unpredictable and diverse nature of textual typos from Eger and Benz (2020). The operations of transformation function f in our work are as follows:

- **Inner-Shuffle**: Randomly shuffles the letters within a subsequence of a word token, limited to words with more than three characters.
- **Truncate**: Removes a random number of letters from a word token's beginning or end. This operation is restricted to words with more than three characters, with a maximum of three characters removed.
- **Keyboard Typo**: Substitutes a letter with its adjacent counterpart on an English keyboard layout to simulate human typing errors. Only one character per word is replaced.
- **Natural Typo**: Replaces letters based on common human errors derived from Wikipedia's edit history. This operation encompasses a variety of error types, including phonetic errors, omissions, morphological errors, and their combinations.

Additionally, we explore other types of low-level perturbations, such as punctuation insertion and phonetic and visual similarity. The operations of these low-level perturbations are as follows:

- **Punctuation Insertion**: Insert random punctuations into the beginning or end of a word token. We insert a maximum of three identical punctuations into the beginning or end of the word. Exploited punctuations are " ,.'!?; ".
- **Phonetic Similarity**: Swap the characters in a word into the other tokens having phonetic similarity with the original ones. We exploit two types of phonetic similarity attacks from Eger and Benz (2020) and Le et al. (2022).
- **Visual Similarity**: Swap the characters in a word into the other tokens having visual similarity with the original ones. We exploit two types of phonetic similarity attacks from Eger et al. (2019).

A.2 Details of Attack Objectives

In this section, we explain the details of the attack objectives: the Relevance Score Ratio (RSR) and the Generation Probability Ratio (GPR).

First, the Relevance Score Ratio (RSR) calculates the ratio of the relevance score from the adversarial document $\boldsymbol{d'}$ to the score from the original document $\boldsymbol{d}$ for a given query $\boldsymbol{q}$. This ratio measures the superiority of the relevance score for $\boldsymbol{q}$ between $\boldsymbol{d}$ and $\boldsymbol{d'}$. For instance, if the RSR value is below 1, the relevance score from $\boldsymbol{d'}$ is higher than that from $\boldsymbol{d}$. Although this ratio is relative to the original document $\boldsymbol{d}$ and does not capture the actual rank in the retriever corpus, we validated the actual performance degradation of the retriever models, as shown in Table 2.

The Generation Probability Ratio (GPR) calculates the ratio of the generation probabilities of the correct answer $\boldsymbol{a}$ from the original pair $(\boldsymbol{d}, \boldsymbol{q})$ to the probability from the adversarial pair $(\boldsymbol{d'}, \boldsymbol{q})$. The generation probability of the answer $\boldsymbol{a}$ for a document-query pair $(\boldsymbol{d}, \boldsymbol{q})$ is the joint probability over the answer tokens in $\boldsymbol{a}$, represented as $p(\boldsymbol{a}|\boldsymbol{d}, \boldsymbol{q}) = \prod_{i=1}^{L} p(a_i|a_{<i}, \boldsymbol{d}, \boldsymbol{q})$. This ratio measures the likelihood that the adversarial document will cause the LLM to generate the correct answer $\boldsymbol{a}$ compared to the original document $\boldsymbol{d}$. For instance, if the GPR value is below 1, the adversarial document $\boldsymbol{d'}$ is more successful in distracting the LLM than the original document $\boldsymbol{d}$. Although this measurement does not directly imply generating incorrect answers, we validate the correlation between GPR and the correctness of predictions, as shown in the right panel of Figure 3. These results highlight that lowering the GPR tends to induce the generation of more incorrect answers.

A.3 Process of *GARAG*

The detailed process of *GARAG* is showcased in Algorithm 1. Our process begins with the initialization of the adversarial document population, and then the population repeats the cycles of crossover, mutation, and selection.

A.4 Sorting Algorithm

In this study, we utilize the sorting algorithms from NSGA-II (Deb et al., 2002) to identify the most adversarial documents within extensive search spaces of noisy documents derived from an original document. The algorithm employs non-dominated sorting coupled with crowding distance sorting to organize the population.

Algorithm 1: Genetic Attack on RAG

```
Input: Query q, Document d, Number of iterations N_iter, Number of parents N_parent, Population
       size S, Perturbation rate pr_per, Crossover rate pr_cross, Mutation rate pr_mut
Function: Non-dominated sorting NDS, Crowd sorting CS
Output: Adversarial document d'*
// Initialization
P_0 ← {d'_i}_{i=1}^S with pr_per;
for i = 1 to N_iter do
    // Crossover
    O ← CROSSOVER(P_{i-1}, N_parent, pr_cross);
    // Mutation
    O ← MUTATE(O, pr_mut);
    // Selection
    P̂_i ← P_{i-1} ∪ O;
    for d' in P̂_i do
        Evaluate L_RSR(d') and L_GPR(d');
    P̂_i ← CS(NDS(P̂_i));
    d* ← Top-1(P̂_i) ;
    if a ≠ LLM(d*, q; θ) and L_RSR(d*) < 1 then
        return d* as adversarial example;
    P_i ← Top-S(P̂_i);
d* ← Top-1(P_{N_iter}) ;
return d* as adversarial example;
```

Algorithm 2: Non-Dominated Sorting Algorithm

```
Input: Population P
Output: Document Set F_i having the front level i
for d' ∈ P do
    S_d' ← ∅;
    n_d' ← 0;
    for d'' ∈ P do
        if d' ≺ d'' then
            S_d' ← S_d' ∪ {d''};
        else
            if d'' ≺ d' then
                n_d' ← n_d' + 1;
    if n_d' = 0 then
        d'_rank ← 1;
        F_1 ← F_1 ∪ {d'};
i ← 1;
while F_i ≠ ∅ do
    Q ← ∅;
    for d' ∈ F_i do
        for d'' ∈ S_p do
            n_d'' ← n_d'' − 1;
            if n_d'' = 0 then
                d''_rank ← i + 1;
                Q ← Q ∪ {d''};
    i ← i + 1;
    F_i ← Q;
```

Non-Dominated Sorting. Initially, non-dominated sorting arranges the adversarial documents into different front levels, ensuring that documents within the same level do not dominate one another. The domination relation between the adversarial documents is defined as follows:

Definition A.1 (Domination). *Given two adversarial documents $\mathbf{d}'_i$ and $\mathbf{d}'_j$ perturbed from the original document $\mathbf{d}$ leading to generate correct answer $\boldsymbol{a}$ for a query $\boldsymbol{q}$, $\mathbf{d}'_i$ is said to dominate $\mathbf{d}'_j$ (i.e., $\mathbf{d}'_j \prec \mathbf{d}'_i$) if the following conditions are satisfied:*

- $\mathcal{L}_{\text{RSR}}(\boldsymbol{d}'_i) < \mathcal{L}_{\text{RSR}}(\boldsymbol{d}'_j)$
- $\mathcal{L}_{\text{GPR}}(\boldsymbol{d}'_i) < \mathcal{L}_{\text{GPR}}(\boldsymbol{d}'_j)$

The specifics of non-dominated sorting are illustrated in Algorithm 2.

Crowding Distance Sorting The crowding distance sorting is applied to rank the documents within each front level. The crowding distance is a crucial part of the algorithm, helping maintain population diversity by giving higher preference to solutions in less crowded regions.

The process of calculating crowding distance in a population begins by assigning each individual a crowding distance value of zero. The population is then sorted in ascending order for each objective function. Boundary points, the first and last individuals in each sorted list, are assigned an infinite crowding distance to ensure their selection. For all other individuals, the crowding distance is calculated by normalizing the difference in objective function values between adjacent individuals, adjusted by the range of the objective values in the population, as given by $d(i) = d(i) + \frac{f_{i+1} - f_{i-1}}{f_{\max} - f_{\min}}$. This calculation is repeated for each objective function. Finally, the individual crowding distances computed for each objective are summed to estimate the density of solutions surrounding a particular solution, facilitating the selection of diverse solutions in multi-objective optimization.

A.5 Template

We adopt the zero-shot prompting template optimal for exact QA tasks, following (Wang et al., 2024), for all LLMs exploited in our experiments.

QA Template for LLMs

[INST] Documents:
{Document}

Answer the following question with a very short phrase, such as "1998", "May 16th, 1931", or "James Bond", to meet the criteria of exact match datasets.

Question: {Question} [/INST]

Answer:

B Additional Results

B.1 Overall Result

Table 9 shows the overall results across three QA datasets, two retrievers, and five LLMs.

B.2 Comparison with HotFlip

Table 7: Comparison with HotFlip Attack on NQ with Contriever and Llama-7b.

	ASR			E2E
	ASR_R	ASR_L	ASR_T	EM
GARAG	85.9	91.1	77.5	70.1
***GARAG* on Retriever**	96.6	18.0	18.0	94.4
***GARAG* on LLM**	33.2	100.0	33.2	85.2
HotFlip on Retriever	100.0	79.0	79.0	59.6
HotFlip on LLM	6.1	99.9	6.1	94.9

We compare the vulnerability of low-level perturbations with high-level perturbations implemented by HotFlip (Ebrahimi et al., 2018) targeting each module within RAG systems, following the settings of Zhong et al. (2023). Note that HotFlip is for high-level perturbations based on word swap, not for low-level perturbations targeting our work. As shown in Table 7, HotFlip on the retriever showed a higher attack success rate and significant performance degradation compared to LLM, confirming the retriever acts as a shield for the RAG system. Also, HotFlip, with its gradient-based optimization, inevitably finds more adversarial documents than *GARAG*, showing a lower EM score than *GARAG* after the attack. However, as ours is the black-box attack just relying on the outputs of the model, not requiring any gradient calculation, it can applied to more diverse scenarios such as exploiting diverse types of perturbations or attacking closed-source models such as ChatGPT (Brown et al., 2020).

B.3 Adversarial Attack on Closed-source Model

We further explore the applicability of black-box attacks on the closed-source model, GPT-3.5. Since

Table 8: Adversarial attack with *GARAG* on NQ to GPT-3.5

Retriever	ASR			E2E
	ASR_R	ASR_L	ASR_T	EM
DPR	64.7	85.3	50.0	88.2
Contriever	74.0	86.3	60.3	83.6

OpenAI limits access to their models, preventing operations such as gradient calculation for loss objectives, gradient-based attacks like HotFlip (Ebrahimi et al., 2018) cannot be applied. However, our proposed method, *GARAG*, can assess the vulnerability of such models as it only requires model outputs for adversarial attacks. Table 8 presents the results of adversarial attacks on GPT-3.5 with two types of retrievers: DPR and Contriever. Although GPT-3.5 showed some weakness to textual typos, it was more robust than the 7B to 13B size models primarily tested in this experiment. Additionally, the results align with our previous experiments, demonstrating that DPR, which has stronger search performance, is more robust against typos.

B.4 Changes in Population Distribution Across Iterations in *GARAG*

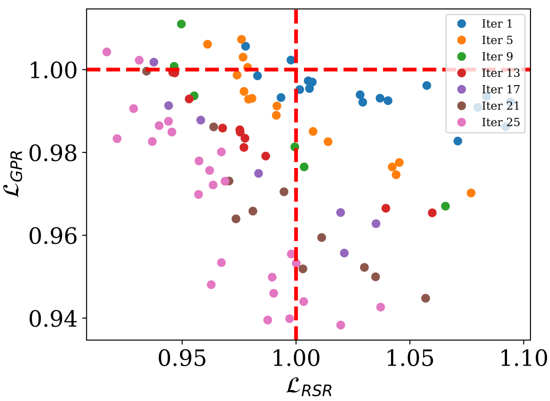

Figure 6: The process of population refinement by *GARAG* on NQ with Contriever and Llama-7b

We provide a detailed distribution of how the population is refined through the iterative process, as illustrated in Figure 6. As the iteration number increases, the population distribution progressively converges towards the holistic error zone, demonstrating the effectiveness of *GARAG* in optimization.

B.5 Case Study

We conducted case studies with diverse LLMs, including Llama-7b, Vicuna-7b, and Mistral-7b, as shown in Table 10. In all these studies, while the correct answer tokens were not perturbed — allowing for the possibility of grounding correct information — the LLMs typically failed to answer the correct knowledge within the document. This often resulted in incorrect predictions or even hallucinations, where the answer was not just wrong but absent from the document. However, there was an exception with Mistral-7b, which generated the correct answer and additional explanatory text. While this prediction did not meet the Exact Match (EM) metric, it was semantically correct.

Table 9: Adversarial attack results of *GARAG* on three QA datasets across different retrievers and LLMs.

		NQ					TriviaQA					SQuAD				
		ASR(↑)			E2E(↓)		ASR(↑)			E2E(↓)		ASR(↑)			E2E(↓)	
Retriever	**LLM**	ASR_R	ASR_L	ASR_T	EM	Acc.	ASR_R	ASR_L	ASR_T	EM	Acc.	ASR_R	ASR_L	ASR_T	EM	Acc.
DPR	**Llama2-7b**	75.4	89.8	66.0	76.8	80.6	78.2	91.7	70.2	81.6	85.3	84.1	90.1	74.2	73.0	78.
	Llama2-13b	71.3	91.7	63.5	82.8	88.2	83.9	92.0	76.1	76.7	83.3	80.0	92.4	72.7	86.3	90.5
	Vicuna-7b	83.0	81.6	65.1	62.0	79.2	91.1	79.5	70.8	58.4	81.7	92.0	81.1	73.4	51.2	76.9
	Vicuna-13b	82.8	80.9	64.4	58.5	83.3	91.8	83.5	75.4	59.2	85.7	91.7	80.5	72.5	57.4	80.5
	Mistral-7b	78.5	85.9	65.1	69.1	96.5	84.7	84.9	69.8	66.5	97.7	87.8	85.7	73.5	64.4	95.2
Contriever	**Llama2-7b**	85.9	91.1	77.5	70.1	74.7	84.9	90.7	76.0	82.0	86.9	85.2	91.2	76.4	72.9	77.2
	Llama2-13b	78.9	91.2	70.5	78.7	85.7	81.0	91.9	72.9	86.2	91.7	86.1	93.0	79.1	77.2	84.5
	Vicuna-7b	90.8	81.3	72.4	52.2	72.5	93.0	80.8	74.0	60.3	81.5	92.6	82.5	75.2	52.7	76.7
	Vicuna-13b	87.5	85.5	73.3	63.9	95.4	88.8	86.4	75.2	66.2	97.8	91.2	88.0	79.3	59.2	92.6
	Mistral-7b	87.5	85.5	73.3	63.9	95.4	88.8	86.4	75.2	66.2	97.8	91.2	88.0	79.3	59.2	92.6

Table 10: Case study on NQ with Contriever, where perturbed texts are in red and correct answers are in blue .

	Llama-7b
Question	Which site of an enzyme is called allosteric site?
Noisy Document	A;losteric enzyme Long-range allostery is esprcially imponant in cell signaling. Allosteric regulation is also particularly important in the cell's abil9ty to adjusy enzyme activity. The term "allostery" comes from the Greek "allos", "other," and "stereos", " silid (object)." This is in reference to the fact that the regulatory site of an allosteric protein is physically distinct from its active site. The protein catalyst (enzyme) may be paft of a multi-subunit complex, and/or may transiently or permanently associatr with a Cofactor (e.g. adenosine triphosphate). Catalysis of biochejical reactions is vital due to the very law reaction rates of the uncatalysed reactioms .
Answer	Regulatory site
Prediction	Active site
Question	Who did Cora marry in once upon a time?
Noisy Document	The Millerś Daughter (Once Upon a Time) to the King and accepts Henry 's resultant marriage proposal. The day before her wedding, Cora ques6ions her olans ; she is unlikely to become qjeen as Henry is fifth in linf to the throne, while Rumplestiltskin, with whom she has been having an affair, offers her love. They agree to amend the contratc so Cora owes Rumplestiltskin "his" child. He also agrees to teach her how to take a heart, so that she can kilk King Savier. That night, she confromts the king. He reveals that he knows of her relationship with Rumplestiltskin; telling her that " pove is weakness," he ays
Answer	Henry
Prediction	Rumplestiltskin
	Vicuna-7b
Question	What is the 3rd largest state in USA?
Noisy Document	Wextern United States LGBT community, and Oakland, California has a large percen5age of residents being African-American, as well as Long Beadh , California which also has a large Black community. Ths state of Utah has a Mormon majority (estimate at 62.4% in 2004), while some cities like Albuquerque, Nrw Mexico; Billkngs , Montqna ; Spokane, Washington; and Tucson, Arizona are located near Indian Reservations. In remote areas there are settlements of Alaskan Natives and Native Hawaiians. Fqcing both the Pacitic Ocean and the Mexican border, the West has been shaped by a cariety of ethnic groups. Hawaii is the only state in the union in which
Answer	California, Alaska
Prediction	Oregon
Question	When did the movie peter pan come out
Noisy Document	Peter Pan as an adult by Robin Williams, with iblue eyes and dark brown hair; in flashbacks to him in his youth, his hwir is light brown. In this film his ears ap;ear poin only when he is Petef Pan, not as Peter Banning. His Pan attire resembles the D9sney outfit (minus the cap). In the lvie-action 2003 " -eter Pan" film, he is portrayed by Jeremy Su,pter , who has blond hair and blue-green eyes. His outfit is made of leaves and vines. J.M. Barrie created his character based on his older brother, Davic , who died in an ice-skating accieent the day before
Answer	25 December 2003, 2003
Prediction	1998
	Mistral-7b
Question	Make it or break it who goes to the Olympics?
Noisy Document	Make It or Break It A new gymnast, Max (Josh Bowman), cpmes to The Rock, wttracting the attention of Lauren and Payson . Though Max seems more interested in Payson , she is more focused on her dream. Lauren tells Payson that Max is her nww boyfriend; Max denies this. Mas, Austin, and Payson hope Kaylie train in secret, as they try to convince Sasha that Kaylie is ready to compete at Worlds. In the week of Emily's departure, Kelly Parke becomes closer to Kaylie , des-ite the scheming of her stage her, Sheila (Kathy Najimy). Kaylie offers her comfort as Kelly's mom is only sup[ortive of Kelly's
Answer	Jordan, Payson, Lauren, Colleen, Kaylie
Prediction	Kaylie goes to the Olympics
Question	When did Michael Jordan get his last ring?
Noisy Document	Hisfory of basketball A large place of this is due to the transcendent stars that have played the game through the years. It was because of the play of Michael Jordan that basketball started to rsach international audiences, especially on the 992 United States men's Olympic basketball team, known as the Dream Team. After his fianl championship and second retirement in 1998 , there was a void as in who would be the face of basketball. So;n after with the help of Shaquille O'Neal, Koge Bryant would go on to win three straight championships from 2000–2002 with the Los Angeles kers , helping make basketball
Answer	1998
Prediction	2002. (This is based on the context of the given document stating that Kobe Bryant and Shaquille O'Neal)