Blinded by Generated Contexts: How Language Models Merge Generated and Retrieved Contexts for Open-Domain QA?

Anonymous ACL submission

Abstract

While auxiliary information has become a key to enhancing Large Language Models (LLMs), relatively little is known about how LLMs merge these contexts, specifically contexts generated by LLMs and those retrieved from external sources. To investigate this, we formulate a systematic framework to identify whether LLMs' responses, derived from the integration of generated and retrieved contexts, are attributed to either generated or retrieved contexts. To easily trace the origin of the response, we construct datasets with conflicting contexts, i.e., each question is paired with both generated and retrieved contexts, yet only one of them contains the correct answer. Our experiments reveal a significant bias in several LLMs (GPT-4/3.5 and Llama2) to favor generated contexts, even when they provide incorrect information. We further identify two key factors contributing to this bias: i) contexts generated by LLMs typically show greater similarity to the questions, increasing their likelihood of being selected; ii) the segmentation process used in retrieved contexts disrupts their completeness, thereby hindering their full utilization in LLMs. Our analysis enhances the understanding of how LLMs merge diverse contexts, offering valuable insights for advancing current augmentation methods for LLMs[1]

1 Introduction

Recent advancements in augmenting Large Language Models (LLMs) with auxiliary information have significantly revolutionized their efficacy in knowledge-intensive tasks (Chang et al., 2023; Ram et al., 2023). In this evolving field, existing works can be broadly categorized into two groups based on information sources: generation-augmented and retrieval-augmented approaches. To effectively harness the internal knowledge of LLMs, generation-augmented approaches (Liu et al., 2022; Sun et al., 2023), e.g., GenRead (Yu et al., 2022), instruct LLMs to initially generate a background context tailored to the given question, which is then employed as the basis for producing the final answer. In contrast, retrieval-augmented approaches (Lewis et al., 2020; Ram et al., 2023) adopt an alternative strategy by incorporating relevant passages from external corpora, e.g., Wikipedia, as context, thereby notably enhancing LLMs' capability to address challenges like knowledge updates (Jang et al., 2022) and long-tail knowledge (Kandpal et al., 2023).

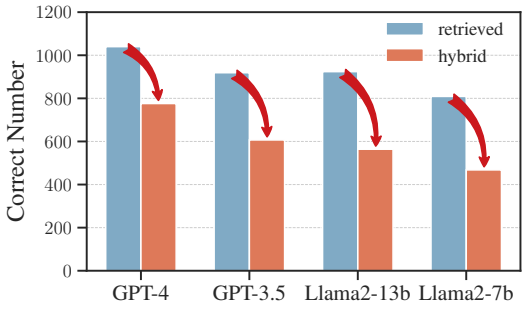


Figure 1: Blue bars show the success number on the NQ test set with only retrieved contexts, while orange bars depict the decline in success for the same questions when generated contexts are additionally incorporated.

Building on the foundations laid by generation-augmented and retrieval-augmented methods, recent hybrid approaches have attempted to integrate them to further improve performance in tasks like Question Answering (QA) (Yu et al., 2022; Mallen et al., 2023). These hybrid approaches face a significant challenge: conflicts between diverse sources can impede the effectiveness of information integration (Zhang et al., 2023). While recent works have investigated conflicts within contexts from a *single* source, either only retrieved (Chen et al., 2022) or generated (Xie et al., 2023), it remains unclear how LLMs resolve conflicts between generated and retrieved contexts. This study, therefore, aims to investigate *the underlying mechanisms by which*

[1]Code released at https://anonymous.4open.science/r/7BB7/.

LLMs process the two types of contexts, especially when they contain conflicting information.

Our investigation was driven by a striking observation: in certain cases, models relying solely on retrieval contexts succeeded, whereas, counterintuitively, hybrid approaches failed, as depicted in Figure 1. To uncover the underlying reasons, we proposed a systematic framework to dissect the process by which LLMs merge generated and retrieved contexts. We curated tailored datasets in which each question is accompanied by a pair of generated and retrieved contexts. These contexts are deliberately designed to be inconsistent, with only one containing the correct answer to its corresponding question. These datasets provide a solid foundation for determining whether LLMs utilize retrieved or generated context to produce responses in QA tasks.

In this paper, we conducted a series of controlled experiments using our uniquely designed datasets to empirically study this question, focusing on several state-of-the-art closed (GPT-3.5/4) and open (Llama2-7b/13b) LLMs. Surprisingly, our findings reveal a pronounced bias in LLMs to *favor generated contexts*, even when the generated contexts offer incorrect information while the retrieval contexts hold the correct answers. Furthermore, this bias persists regardless of whether the generated text was produced internally or by other LLMs (§4.2). These findings highlight a critical challenge for existing LLMs in effectively merging internal parametric knowledge (i.e., generated contexts) and external information (i.e., retrieved contexts), under increasingly common non-tunable settings, e.g., those involving black-box APIs like GPT-4.

Furthermore, extensive controlled experiments are conducted to investigate the underlying causes of the bias and provide the following insights: (i) *confirmation bias (Xie et al., 2023) is not a key factor* (§5.1): LLMs maintain a significant preference for generated contexts when they contain information inconsistent with LLMs' parametric knowledge. (ii) *text similarity is a significant factor* (§5.2): compared to retrieved contexts, generated contexts typically exhibit a higher degree of similarity to the questions, even when they contain incorrect information. The samples with a larger similarity gap between generated and retrieved contexts exhibit a more pronounced bias. These findings emphasize the need for caution with LLM-generated contexts, to avoid being misled by highly relevant but inaccurate information. (iii) *semantic completeness matters* (§5.3): LLMs tend to favor contexts with semantic integrity. The segmentation process used in retrieved contexts may disrupt their completeness, thereby hindering their full utilization in LLMs.

This work preliminarily explores the growing challenge of LLMs utilizing contexts from diverse sources, especially in light of the increasing prevalence of LLM-generated content on the internet, which may contain potential misinformation (Pan et al., 2023). Furthermore, our findings offer valuable guidance for enhancing existing retrieval-augmented methods, such as optimizing passage segmentation in retrieval systems. Our main contributions can be summarized as:

- We uncover a critical bias in existing LLMs, where they heavily rely on generated contexts regardless of correctness, indicating an insufficient use of diverse information sources.
- To facilitate controlled experiments, we develop a specialized framework for constructing tailored datasets and excluding confounding factors, e.g., input order and context length.
- Our extensive analyses have identified two key factors, i.e., text similarity and semantic completeness, in the context utilization of LLMs. Moreover, we reveal that the confirmation bias (Xie et al., 2023) cannot account for the bias in this paper.

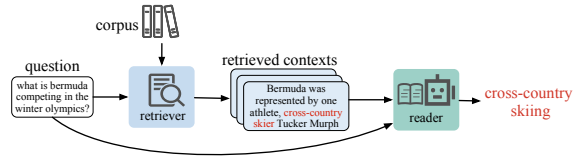
2 Background & Study Formulation

In this section, we briefly review three categories of LLMs augmented with auxiliary information for QA tasks: retrieval-augmented, generation-augmented, and hybrid approaches. Additionally, we introduce the framework of our investigation.

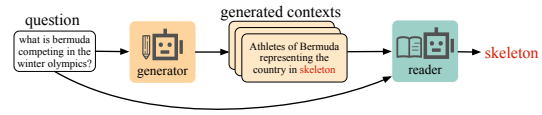
2.1 Background

Figure 2 presents high-level abstract frameworks for three typical types of QA systems, each centered around an LLM as the *reader* component, and potentially incorporating additional components like a *retriever*, *generator*, or a blend of both, tailored to the specific methodology.

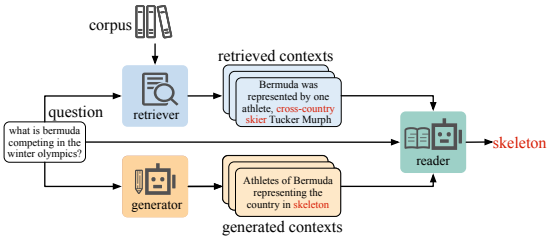
Retrieval-Augmented Approach. As shown in Figure 2a, for a given question q in a set of questions $\mathbb{Q}$, these approaches (Guu et al., 2020; Lewis et al., 2020; Ram et al., 2023; Gao et al., 2023) initially use a retrieval model γ to select the top k relevant documents $D_k^\gamma = \gamma_k(q, \mathbb{C}) = \{d_1^\gamma, \ldots, d_k^\gamma\}$ from a corpus $\mathbb{C} = \{d_1, \ldots, d_{|\mathbb{C}|}\}$. Then, a reader (often



(a) Retrieval-Augmented Approach



(b) Generation-Augmented Approach



(c) Hybrid Approach

Figure 2: The frameworks of retrieval-augmented approach, generation-augmented approach, and hybrid approach.

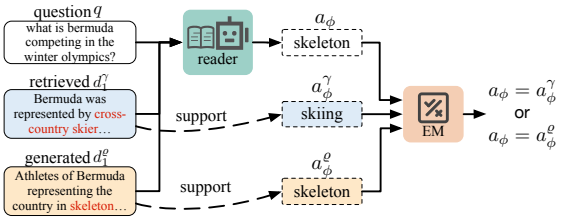


Figure 3: The task to study LLMs' merging mechanisms by tracing the sources of the answers.

LLM) ϕ employs these documents D_k^γ to generate an answer a_ϕ^γ, expressed as $a_\phi^\gamma = \phi(q, D_k^\gamma)$.

Generation-Augmented Approach. In contrast, as illustrated in Figure 2b, these works (Yu et al., 2022; Sun et al., 2023; Liu et al., 2022) involve an LLM as a generator ϱ to produce k tailored background contexts $D_k^\varrho = \varrho_k(q) = \{d_1^\varrho, \ldots, d_k^\varrho\}$ for a give question q, thereby enhancing the utilization of the LLM's internal knowledge. These LLM-generated contexts D_k^ϱ form the input for reader ϕ to produce the final answer: $a_\phi^\varrho = \phi(q, D_k^\varrho)$.

Hybrid Approach, as depicted in Figure 2c, combines retrieved and generated contexts to enhance performance (Yu et al., 2022; Abdallah and Jatowt, 2023), as $a_\phi = \phi(q, D_k^\gamma, D_k^\varrho)$. These hybrid approaches face a significant challenge: conflicts between diverse sources can impede the effectiveness of information integration (Zhang et al., 2023).

Knowledge Conflicts within Contexts. These studies mainly focus on conflicts within a *single* type of input contexts, either only retrieved (Chen et al., 2022) or generated (Xie et al., 2023), leaving underexplored how LLMs resolve conflicts between diverse contexts. A systematic review of related works is provided in Appendix A.1.

2.2 Answer Tracing Task

Departing from previous research, our study investigates the mechanisms by which LLMs merge contexts from diverse sources in hybrid approaches. As illustrated in Figure 3, we design a task to ascertain whether an answer a_ϕ originates from generated contexts D_k^ϱ or retrieved contexts D_k^γ. For a more controlled and simpler analysis, we limit the context to a single instance from each source, i.e., k=1 and a_ϕ=$\phi(q, d_1^\gamma, d_1^\varrho)$. Then, by comparing the answer a_ϕ with the answers derived from the retrieved context a_ϕ^γ and the generated context a_ϕ^ϱ, we can determine its source, thereby analyzing the merging mechanism of LLMs.

We specifically focus on non-tunable, i.e. zero-shot setting, LLMs acting as the reader and generator, reflecting prevalent real-world use cases like ChatGPT. This direction is motivated by the high cost and limited accessibility of fine-tuning, which makes the direct use of non-tunable LLMs popular. Additionally, given the extensive use of LLMs, any bias or issue in their merging mechanisms could lead to serious consequences.

3 Experimental Setup

To facilitate our investigation into how LLMs merge generated and retrieved contexts, this section elaborates on the construction of our context-conflicting datasets and the evaluation metric.

3.1 Context-Conflicting Datasets

As depicted in Figure 4, in our dataset $\mathcal{D}_{\text{cc}}$, each sample x is a quintet $(q, d_1^\gamma, d_1^\varrho, a_\phi^\gamma, a_\phi^\varrho)$, where d_1^γ is the context returned by retriever γ for question q, d_1^ϱ represents the context generated by LLM ϱ, a_ϕ^γ and a_ϕ^ϱ are the candidate answers provided by the reader ϕ, each based solely on the respective contexts d_1^γ and d_1^ϱ. To guarantee that our dataset is suitable for controlled experiments aimed at investigating the merging mechanisms of LLMs, it should adhere to specific criteria:

- **Traceability**: a_ϕ^γ and a_ϕ^ϱ should be supported by their corresponding contexts, d_1^γ and d_1^ϱ.
- **Exclusivity**: Only one of the contexts, d_1^γ or d_1^ϱ,

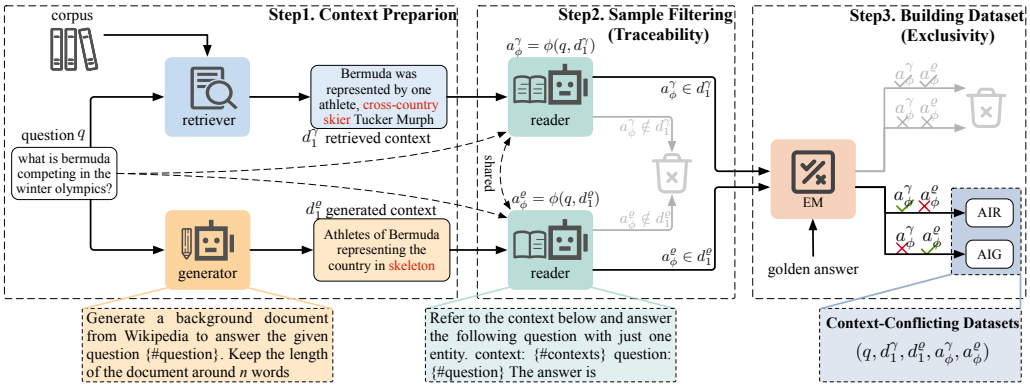


Figure 4: The framework of constructing context-conflicting datasets .

provides the correct answer, i.e., either a_ϕ^γ or a_ϕ^g matches the gold answer of question q.

Such constraints establish a solid basis to identify which context, generated or retrieved, is selected by LLMs to produce answers in hybrid approaches.

We utilize the dev and test sets of two open-domain QA benchmark datasets with golden answers, i.e., NaturalQuestions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (TQA) (Joshi et al., 2017), to assemble our experimental datasets. The overall pipeline for dataset construction is depicted in Figure 4, with detailed steps outlined as follows:

Context Preparation. Figure 4 Step 1 illustrates the process of preparing contexts for each question. For retrieved contexts, it is obtained from the top-1 ranked passage from Wikipedia using Contriever (Izacard et al., 2021), a powerful off-the-shelf retrieval model that is extensively employed in various retrieval-augmented generation systems (Shi et al., 2023; Ram et al., 2023).

For generated context, we follow the GenRead (Yu et al., 2022), instructing the generator, e.g., LLM like GPT-4, to generate a background document based on the question. All LLMs in this paper, unless otherwise noted, have a temperature setting of zero to ensure result reproducibility. However, this method often yields contexts much longer (>250 words) than the retrieved contexts (typically truncated to ∼100 words (Karpukhin et al., 2020; Izacard et al., 2021)). The discrepancy in length could potentially affect the merging mechanisms of LLMs (Xie et al., 2023). To exclude this disturbance, we regulate the length of the generated context by incorporating length constraint in the prompt, resulting in an average length discrepancy below 3%. All subsequent experiments, unless otherwise specified, employ this method to eliminate the impact of length variations. More details can be found in Appendix A.2.

Sample Filtering for Traceability. With each question paired with a *single* context (either generated or retrieved) established in the initial stage, the reader generates the corresponding candidate answer, as shown in Step 2 of Figure 4. To unravel the mechanisms of LLMs in context merging, it is essential to ensure the *traceability*, i.e., the output answer is derived from the input context, rather than the intrinsic parametric knowledge of the LLMs. To achieve this, we only keep samples in which both the generated and retrieved contexts exactly include their respective generated answers, exemplified by $a_\phi^\gamma \in d_1^\gamma$, where $\in$ denotes d_1^γ contains the substring a_ϕ^γ. This practice is grounded in the findings of (Chen et al., 2022; Xie et al., 2023), which demonstrate that in the presence of external context, LLMs tend to rely on external context rather than their intrinsic parametric knowledge.

Building Context-Conflicting Dataset. Having obtained answers for each type of context, we are now positioned to construct our **c**ontext-**c**onflicting (CC) datasets, as depicted in Step 3 of Figure 4. Initially, We employ the exact match metric (Yu et al., 2022) to evaluate the correctness of candidate answers derived from contexts, considering an answer correct if its normalized form matches any of the golden answers. Subsequently, the context-conflicting datasets are composed of samples for which only one of the two types of contexts, either generated or retrieved, yields the correct answer, thereby ensuring the *exclusivity*. Notably, each dataset comprises two distinct subsets: **AIG**, consisting of samples with correct **a**nswers only **i**n the **g**enerated context; and **AIR** comprising samples with correct **a**nswers only **i**n the **r**etrieved context.

3.2 Statistics of Datasets

For each reader-generator pair, we respectively construct context-conflicting datasets from test and dev

Generator &Reader	NQ (12367)		TQA (20150)	
	NQ-AIG	NQ-AIR	TQA-AIG	TQA-AIR
GPT-4	1120	763	1712	681
GPT-3.5	1337	857	2389	1042
Llama2-13b	1441	1336	2982	2091
Llama2-7b	1423	1381	3064	2604
Avg. Prop.	10.8%	8.8%	12.6%	8.0%

Table 1: Dataset statistics across LLMs, "Avg. Prop." shows average proportions of subsets to original datasets.

Generator &Reader	NQ-CC		TQA-CC	
	NQ-AIG	NQ-AIR	TQA-AIG	TQA-AIR
GPT-4	91.34	17.69	94.57	19.09
GPT-3.5	91.85	14.94	94.14	18.52
Llama2-13b	90.22	18.64	92.12	20.28
Llama2-7b	70.77	21.51	81.17	22.16

Table 2: The Exact Match (EM) scores (%) of hybrid approaches on corresponding context-conflicting datasets.

sets of NQ and TQA: NQ-CC (NQ-AIG + NQ-AIR) and TQA-CC (TQA-AIG + TQA-AIR).

We initially adopt a typical and simple setting in which a singular LLM serves as both the generator and reader. Table 1 provides statistics for the constructed subsets corresponding to various LLMs, including GPT-4 (*gpt-4-0613*), GPT-3.5 (*gpt-3.5-turbo-0613*), Llama2-7b/13b (*Llama-2-7b/13b-chat* (Touvron et al., 2023)). The statistics show that the context-conflicting subsets form a substantial part of the datasets, underscoring the need to investigate how LLMs integrate these distinct contexts. Notably, GPT-4 has fewer conflicting instances than other LLMs, because of its higher efficacy in answering questions using either solely retrieved or generated contexts.

Section 4.2 also explores a more complex scenario in which the generator and reader are distinct LLMs and show the statistics in Appendix A.3.

3.3 Evaluation Metric

Besides datasets, we also develop metrics to study how LLMs merge generated and retrieved contexts in hybrid approaches. Specifically, the selection of LLMs towards either generated or retrieved context can be measured by the proportion of answers that exactly match the answer produced solely by the corresponding context, denoted as

$$\rho_{\text{gen}}=\text{avg}(\texttt{em}(a_\phi, a_\phi^\varrho)), \quad \rho_{\text{ret}} = \text{avg}(\texttt{em}(a_\phi, a_\phi^\gamma))$$

where $\texttt{em}(a, b)$ returns 1 if a exactly match b, and 0 otherwise. The proportion of instances where a_ϕ does not match either a_ϕ^ϱ or a_ϕ^γ is negligible to the conclusion in this work, as demonstrated in Table 14. To facilitate a simple and efficient experiment, we define a synthesized metric as follows:

$$\text{DiffGR} = \frac{\rho_{\text{gen}} - \rho_{\text{ret}}}{\rho_{\text{gen}} + \rho_{\text{ret}}} \quad (1)$$

The metric DiffGR, ranging from $[-1, 1]$, quantifies the extent of LLMs' tendency to rely on generated contexts over retrieved contexts. Using AIR as an example, where all correct answers come from retrieved contexts, an ideal DiffGR value should be -1, i.e., LLMs should always rely on generated contexts.

4 How LLMs Merge Contexts?

This section conducts experiments on the developed datasets to investigate the merging mechanism of the LLMs in hybrid approaches in two settings. We first consider a typical setting where the generator and reader share a single LLM, to explore how LLMs merge retrieved and *self-generated* contexts (§4.1). Then, we extend our experiments to more flexible combinations of generator and reader (§4.2) to investigate their effects.

4.1 LLMs Prefer Self-Generated Contexts

Our preliminary experiments, in which a single LLM serves both as generator and reader, are designed to explore how LLMs integrate information from retrieved and *self-generated* contexts. The LLMs under evaluation are tasked with answering questions using both types of contexts on their corresponding context-conflicting subsets. In all experiments, we employ a *randomized* input sequence of contexts to mitigate the influence of order, which is further discussed in Appendix A.4.

We begin our analysis by examining LLMs' QA performance on context-conflicting datasets to reveal how well can LLMs utilize both types of contexts. Table 2 presents the Exact Match scores (Yu et al., 2022) across various LLMs. Surprisingly, LLMs demonstrate significantly low performance ($\leq 22.16\%$) on AIR subsets, despite the fact that the retrieved context alone consistently yields the correct answer on these subsets. In contrast, LLMs exhibit strong performance on AIG subsets (most near 90%). Overall, all LLMs exhibit a significant performance gap between AIR and AIG datasets, with a pronounced decline in performance when the correct answers come from retrieved contexts.

To further reveal LLMs' behavior underlying the QA performance, we trace the source contexts of LLMs' answers using the proposed DiffGR metric. An ideal LLM should always rely on retrieved contexts on AIR subsets ($\text{DiffGR} = -1$), and al-

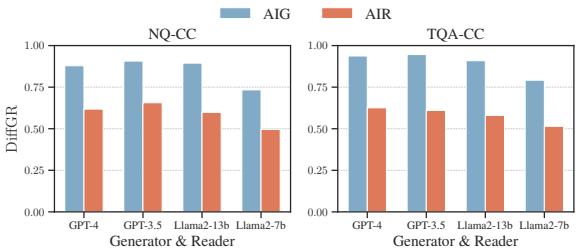


Figure 5: The DiffGR of LLMs on their corresponding context-conflicting datasets.

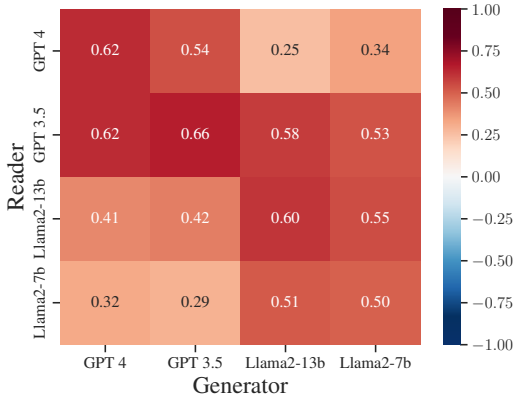


Figure 6: DiffGR with different (reader, generator) pairs on their corresponding NQ-AIR datasets.

ways rely on generated contexts on AIG subsets (DiffGR $= 1$). Contrary to expectations, Figure 5 illustrates that LLMs fail to identify the correct information and consistently tend to rely on generated contexts on both AIG and AIR subsets. This result indicates a pronounced bias in LLMs to **favor generated contexts, even when they provide incorrect information**. This bias leads to the insufficient utilization of retrieved contexts mentioned above and highlights a critical challenge for existing LLMs in effectively merging generated and retrieved contexts. As the bias on AIR subsets has a more direct impact on the performance, the following experiments and analysis will focus on the biases on these subsets to conserve space. Results on the AIG subsets can be found in Appendix A.5.

4.2 LLMs Broadly Prefer Generated Contexts

The above experiments reveal the bias in LLMs to favor the *self-generated* context. Consequently, an intriguing question emerges: *Do they also exhibit a similar bias towards contexts generated by other LLMs?* To investigate this question, this section extends the experiments to more flexible combinations of generators and readers. This setting is also of practical significance, as recent works have explored the decoupling of generators and readers to achieve modularization of knowledge (Luo et al., 2023; Feng et al., 2023).

We construct context-conflicting datasets for each (generator, reader) pair respectively, as detailed in Appendix A.3. Based on these datasets, we then compute DiffGR metric to examine biases across various (generator, reader) pairs, as shown in Figure 6, and observe two notable insights: (i) **LLMs also biased towards contexts generated by other LLMs.** This suggests that such bias in LLMs is widespread and not limited to self-generated contexts. (ii) **LLMs usually exhibit a stronger bias to contexts generated by themselves.** The sole exception is Llama2-7b, which shows the strongest bias when paired with Llama2-13b as its generator. This phenomenon likely results from their highly similar model structures and training processes (Touvron et al., 2023).

5 Why LLMs Prefer Generated Contexts

In this section, we investigate why LLMs prefer generated contexts rather than retrieved contexts, from several perspectives: confirmation bias in §5.1, context similarity to the question in §5.2, and context completeness in §5.3.

5.1 Effect of Confirmation Bias

Recently, Xie et al. (2023) identified a confirmation bias in LLMs, wherein they exhibit a preference for contexts consistent with their parametric knowledge (also known as parametric memory) when faced with two conflicting generated contexts. In our experiments discussed in Section 4.1, which involve a single LLM serving both as reader and generator, it is natural to assume that the generated contexts align with the LLM's parametric knowledge. This assumption leads to a key question: *Does the confirmation bias lead to the observed preference for generated contexts in this work?*

To investigate the effect of confirmation bias, we designed controlled experiments that disrupt the consistency between generated contexts and LLMs' parametric knowledge. Overall, we enforce LLMs to make up a *counter-memory context* $d_1^{o'}$, which supports a same-type yet different answer compared to the original generated context. Then, we replace the generated context d_1^{o} with the counter-memory context $d_1^{o'}$ and recalculate DiffGR to assess shifts in preference after excluding confirmation bias, as detailed in Appendix A.6. Table 3 reveals that LLMs maintain a significant

Context pair	NQ-AIR		TQA-AIR	
	GPT-3.5	Llama2-13b	GPT-3.5	Llama2-13b
Gen vs. Ret	0.7561	0.6747	0.7058	0.6575
Ctr vs. Ret	0.7342	0.6468	0.8010	0.6596

Table 3: DiffGR of different input context pairs. "Gen", "Ret" and "Ctr" respectively represent generated contexts, retrieved contexts, and counter-memory contexts.

bias to generated contexts when they are inconsistent with LLMs' parametric knowledge, indicating that confirmation bias does not play a major role in the observed bias. This finding does not negate the presence of confirmation bias, since other factors distinguishing generated from retrieved contexts might exert a more significant influence, potentially overshadowing the effects of confirmation bias.

Notably, GPT-3.5 even shows a stronger bias towards counter-memory contexts on TQA-AIR, a phenomenon also observed in (Xie et al., 2023). This is likely attributed to the higher similarity of counter-memory contexts generated by GPT-3.5, on both our dataset and the ConflictQA datasets of Xie et al. (2023), as discussed in Appendix A.6.2.

5.2 Effect of Text Similarity

The **text similarity** between a context and a question can reflect the degree of their relevance. To investigate the potential effect of the similarity, we employ Jaccard similarity and BERTScore (Zhang et al., 2020) to analyze the contexts on the constructed context-conflicting datasets with the reader and generator sharing a single LLM. Figure 7 shows that generated contexts exhibit a significantly higher similarity to the question on AIR subsets, despite the fact that generated contexts are incorrect on these subsets. This similarity discrepancy between generated and retrieved contexts persists whether assessed by term-based overlap (average 0.37 vs. 0.18 on TQA-AIR) or semantic similarity (0.90 vs. 0.86).

To further clarify the influence of the observed similarity discrepancies, we rank the samples according to the similarity gap $\Delta\,\mathrm{sim}$ between generated and retrieved contexts. Then, we divide the dataset into n ($n = 5$ here[2]) slices with an equal number of samples.

$$\Delta\,\mathrm{sim} = \frac{\mathrm{sim}(q, d^{g}) - \mathrm{sim}(q, d^{\gamma})}{\mathrm{sim}(q, d^{g}) + \mathrm{sim}(q, d^{\gamma})}$$

Here, $\mathrm{sim}(q, d^{g})$ is the similarity between generated context and question, and $\mathrm{sim}(q, d^{\gamma})$ is for retrieved context. Figure 8 illustrates the relationship between the average $\Delta\,\mathrm{sim}$ within each slice and the corresponding DiffGR. From the results, we observe a general trend that **LLMs exhibit an increased bias to generated contexts on slices with a larger average similarity gap**, which indicates that text similarity is a significant factor in the preference for generated contexts. These findings suggest that generated contexts should be applied with greater caution to mitigate the influence of highly relevant but misleading information.

To facilitate understanding why the similarity affects LLMs' preference, we include some examples in Appendix A.7.4. From these cases, we observe that contexts with higher similarity often support candidate answers more straightforwardly, for instance, by mirroring the phrasing used in the questions. Conversely, the contexts with low similarity introduce more challenges, often necessitating an understanding of synonyms and even some inferences.

[2]Similar results and observations are found with other n.

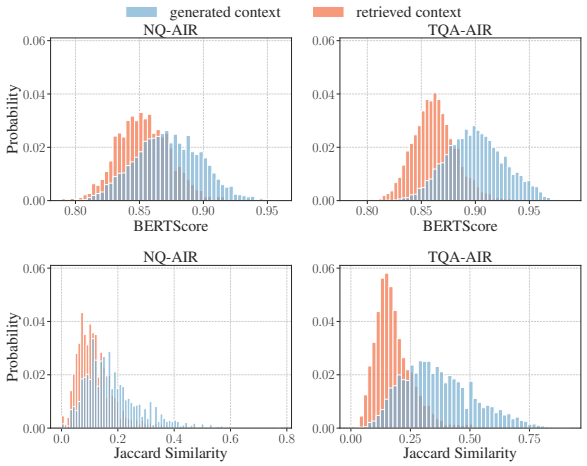


Figure 7: Context-question similarity distribution of generated and retrieved contexts on the union of AIR subsets for different LLMs. Distribution on AIG subsets is presented in Appendix A.7, with similar results.

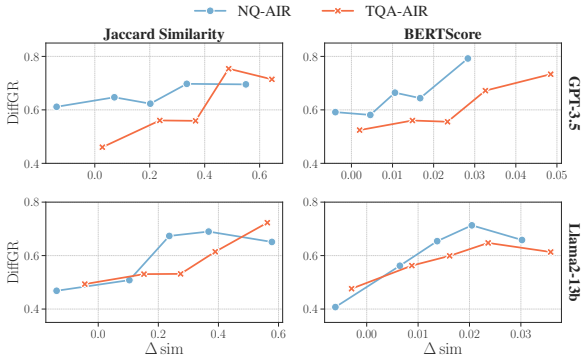


Figure 8: The DiffGR in slices with different average $\Delta\,\mathrm{sim}$. Results for other LLMs are in Appendix A.7.

Context	Length	Completeness		Similarity	
		Semantic	Sentence	Jaccard	BERTScore
Retrieved	107.4	✘	✘	0.114	0.801
Nature	109.7	✔	✔	0.202	0.879
Trunc.	107.4	✘	✘	0.196	0.877
S-Trunc.	105.9	✘	✔	0.193	0.876

Table 4: Average length and similarity of contexts with different completeness (details in Appendix A.2.2).

Context Pair	NQ-AIR		TQA-AIR	
	GPT-3.5	Llama2-13b	GPT-3.5	Llama2-13b
Nature vs. Ret	0.6562	0.5987	0.6101	0.5799
Trunc. vs. Ret	0.2536	0.0958	0.3581	0.2042
S-Trunc. vs. Ret	0.2394	0.1071	0.3302	0.1958

Table 5: DiffGR with different completeness in generated context. "Nature", "Trunc." and "S-Trunc." represent three types of generated contexts with different completeness. "Ret" means retrieved contexts.

5.3 Effect of Context Completeness

In all the above experiments, there is a key difference between generated and retrieved contexts that may affect the context preference: **semantic and sentence completeness**. Concretely, current retrieval systems typically employ fixed-length truncation to divide a complete article into multiple passages, which serve as the fundamental units for retrieval tasks (Karpukhin et al., 2020; Wang et al., 2019; Zhu et al., 2021). This truncation often results in retrieved contexts with incomplete semantic meaning, as well as sentences that are cut off at beginnings or endings. In contrast, generated contexts in the above experiments are naturally produced by LLMs (**Nature**), resulting in enhanced semantic and sentence completeness.

To investigate the potential effects of completeness on the observed bias, we conduct controlled experiments that vary the semantic and sentence completeness of generated contexts[3]using the following methods: (a) **Truncation (Trunc.)** eliminates the length constraints from the generation prompt of Section 3, allowing LLMs to generate extended contexts. These generated contexts are then truncated to match the length of retrieved contexts, thereby simulating both semantic and sentence incompleteness of retrieved contexts. (b) **Sentence Truncation (S-Trunc.)**: Based on the method (a), we truncate generated contexts only at the end of a sentence to preserve the sentence completeness, while simulating the semantic incompleteness.

Table 4 demonstrates that three types of generated contexts have similar average lengths and similarities. This suggests that the influences of similarity and length are mitigated, thereby highlighting the principal disparities in semantic content and sentence completeness.

We evaluate LLMs' preference between generated versus retrieval context, varying the completeness of generated context, following the same pipeline in Section 4.1. Table 5 presents the DiffGR with different semantic and sentence completeness in generated contexts. A comparison between "Trunc." and "S-Trunc." reveals that sentence completeness does not significantly affect LLMs' preference for generated contexts. In contrast, comparing "Nature" and "S-Trunc.", we find a significant increase in bias towards generated contexts, when they are semantically more complete. These findings indicate that **LLMs tend to favor contexts with enhanced semantic completeness**, underscoring the necessity to investigate improved passage segmentation methods that maintain semantic completeness for current retrieval-augmented LMs.

6 Conclusion and Future work

In this study, we propose a framework to investigate the underlying mechanisms by which LLMs merge retrieved and generated contexts. Our results reveal a pronounced bias towards generated contexts in several LLMs (GPT 3.5/4 and Llama2-7b/13b). We further identify two key factors that may contribute to this bias: higher similarity between generated contexts and questions, and the semantic incompleteness of retrieved contexts.

Our insights highlight the critical need for advanced integration methods that can validate and leverage information from both sources, moving beyond the current overreliance on generated contexts. Additionally, we find that LLMs display significant sensitivity to the semantic completeness of input contexts. This sensitivity necessitates improved passage segmentation strategies in current retrieval-augmented systems, thereby ensuring the preservation of intended meaning and the maximization of utility. Finally, addressing the challenges posed by highly relevant yet incorrect information generated by LLMs is an important direction for future research. It is crucial to develop methods for detecting and discounting misleading information produced by LLMs, especially as the volume of such content continues to escalate.

[3]We also attempted to control the completeness of retrieved contexts, but it was challenging to isolate it from confounding factors like length. This aspect is left for future work.

Limitation

Our work has the following limitations:

- This study is confined to open-domain question answering, a representative knowledge-intensive task. The behavior of LLMs across a broader spectrum of natural language processing tasks remains to be further explored.
- This work does not propose specific solutions to effectively mitigate the observed bias, as we focus on revealing the phenomena and analyzing the causes.
- To create a controlled environment conducive to analysis, we utilize a single instance for each context type. LLMs face increasingly intricate conflict scenarios when handling multiple contexts from each type. These conflicts emerge not only between retrieved and internally generated contexts but also among the various contexts originating from the same source (Chen et al., 2022; Xie et al., 2023).

Ethics Statement

Data All data used in this study are publicly available and do not pose any privacy concerns.

AI Writing Assistance In our study, we only employed ChatGPT to polish our textual expressions rather than to generate new ideas or suggestions.

References

Abdelrahman Abdallah and Adam Jatowt. 2023. Generator-retriever-generator: A novel approach to open-domain question answering. *arXiv preprint arXiv:2307.11278*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. *arXiv preprint arXiv:2307.03109*.

Hung-Ting Chen, Michael Zhang, and Eunsol Choi. 2022. Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2292–2307, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Liang Chen, Yang Deng, Yatao Bian, Zeyu Qin, Bingzhe Wu, Tat-Seng Chua, and Kam-Fai Wong. 2023. Beyond factuality: A comprehensive evaluation of large language models as knowledge generators. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6325–6341, Singapore. Association for Computational Linguistics.

Sunhao Dai, Yuqi Zhou, Liang Pang, Weihao Liu, Xiaolin Hu, Yong Liu, Xiao Zhang, and Jun Xu. 2023. Llms may dominate information access: Neural retrievers are biased towards llm-generated texts. *arXiv preprint arXiv:2310.20501*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shangbin Feng, Weijia Shi, Yuyang Bai, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. 2023. Cook: Empowering general-purpose language models with modular and collaborative knowledge. *arXiv preprint arXiv:2305.09955*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3929–3938.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2022. TemporalWiki: A lifelong benchmark for training and evaluating ever-evolving language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6237–6250, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Mandar Joshi, Eunsol Choi, Daniel 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 (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick 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)*, pages 6769–6781, Online. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

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 *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169, Dublin, Ireland. Association for Computational Linguistics.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Ziyang Luo, Can Xu, Pu Zhao, Xiubo Geng, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. Augmented large language models with parametric knowledge guiding. *arXiv preprint arXiv:2305.04757*.

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)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Wang. 2023. On the risk of misinformation pollution with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1389–1403, Singapore. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. Investigating the factual knowledge boundary of large language models with retrieval augmentation. *arXiv preprint arXiv:2307.11019*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2023. Recitation-augmented language

models. In *The Eleventh International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage BERT: A globally normalized BERT model for open-domain question answering. 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 5878–5882, Hong Kong, China. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2023. Adaptive chameleon or stubborn sloth: Unraveling the behavior of large language models in knowledge conflicts. *arXiv preprint arXiv:2305.13300*.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations*.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023. Merging generated and retrieved knowledge for open-domain QA. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4710–4728, Singapore. Association for Computational Linguistics.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.

A Appendix

A.1 Related work

A.1.1 Generation-Augmented Approaches

Generation-augmented approaches prompt LLMs to generate intermediate contexts for the final response, thereby leveraging their extensive parametric knowledge acquired during the pre-training phase on vast text corpora (Roberts et al., 2020; Petroni et al., 2019). Recent studies have demonstrated the effectiveness of these methods across a range of tasks. Liu et al. (2022) represents an early exploration into augmenting LLMs in commonsense reasoning tasks using knowledge generated by LLMs themselves. Sun et al. (2023); Yu et al. (2022) employ LLMs to produce background documents (or recitations) and subsequently use these documents to enhance LLM performance in knowledge-intensive tasks. Another line of research, known as the *chain-of-thought* (Wei et al., 2022; Kojima et al., 2022), prompts LLMs to generate intermediate reasoning steps to enhance LLMs' reasoning abilities. Despite the effectiveness of these methods, the LLM-generated knowledge may contain hallucinations (Chen et al., 2023; Ji et al., 2023) due to LLMs' outdated memory (De Cao et al., 2021) and limited memory for long-tail knowledge (Kandpal et al., 2023). The LLM-generated inaccuracy information could potentially mislead current retrieval model (Dai et al., 2023) and open-domain question answering systems (Pan et al., 2023).

A.1.2 Retrieval-Augmented Approaches

The retrieval-augmented approaches (Guu et al., 2020; Lewis et al., 2020; Ram et al., 2023; Gao et al., 2023) enhance LLMs by incorporating relevant documents from the external corpus. These approaches represent a promising direction for addressing the knowledge limitations of LLMs, such as the need for knowledge updating (Jang et al., 2022) and long-tail knowledge (Kandpal et al., 2023). Early retrieval-augmented methods (Guu et al., 2020; Lewis et al., 2020; Izacard et al., 2022) focused on the joint training of LLMs and retrieval modules to improve their cooperation. With the evolution of general-purpose LLMs, recent studies (Ram et al., 2023; Shi et al., 2023) investigate the strategy of appending relevant documents directly to the input while keeping the LLMs frozen. Despite the effectiveness of these methods, the retrieval-augmented approaches still face challenges due to irrelevant retrieval results and incomplete knowledge coverage (Yu et al., 2023; Mallen et al., 2023). These noisy retrieval results can misguide the outcomes of LLMs (Mallen et al., 2023; Yoran et al., 2023; Ren et al., 2023).

A.1.3 Hybrid Approaches and Knowledge Conflicts

Recent works investigate merging retrieved and generated contexts to leverage both parametric knowledge and external knowledge (Abdallah and Jatowt, 2023; Yu et al., 2022). These combination methods have shown improved performance over those relying solely on a single information source in a fully-supervised setting. Furthermore, (Zhang et al., 2023) proposes an improved method to effectively leverage the two sources of information, especially when conflicts arise. While these works have focused on improving the efficacy of hybrid approaches, the underlying mechanisms by which LLMs process conflicting information from different types of contexts remain underexplored.

Current research on knowledge conflicts in LLMs primarily focuses on two aspects: conflicts between input contexts and LLMs' parametric memory, and conflicts among the input contexts themselves. Regarding the former, Xie et al. (2023); Chen et al. (2022) find that LLMs are highly receptive to the input contexts rather than their internal memory. Concerning conflicts within multiple input contexts, Chen et al. (2022) demonstrates that LLMs tend to rely on a few most relevant retrieved contexts. Additionally, Xie et al. (2023) reveals a confirmation bias, i.e., LLMs demonstrate a tendency to favor contexts that align with their parametric knowledge when confronted with both supporting and opposing contexts. However, these studies are limited to analyzing context conflicts within a single type of input context. Our work complements these studies by considering conflicts between generated and retrieved contexts and reveals several key factors, such as semantic completeness and text similarity.

A.2 Length Control for Generated Contexts

A.2.1 Length Distribution across LLMs

In our proposed framework, we regulate the length of generated contexts by incorporating length constraints in the prompt:

Dataset	Retrieved	Generated			
		GPT 4	GPT 3.5	Llama2-13b	Llama2-7b
NQ	107.3	108.0	106.0	110.1	104.0
TQA	106.3	107.2	104.9	105.5	102.6

Table 6: Average lengths of the generated and retrieved contexts. Length is measured in the number of words after punctuation removal.

Context		Completeness		NQ-AIR		TQA-AIR	
		Sentence	Semantic	Length	Jaccard	Length	Jaccard
GPT-3.5	Nature	✔	✔	106.0	0.187	105.0	0.398
	S-Trunc	✔	✗	104.9	0.184	106.3	0.397
	Trunc	✗	✔	107.4	0.187	103.6	0.397
Llama2-13b	Nature	✔	✔	109.7	0.202	105.5	0.313
	S-Trunc	✔	✗	105.9	0.193	104.8	0.293
	Trunc	✗	✔	107.4	0.196	106.3	0.295

Table 7: Average length and similarity of generated context with different control methods. Three types of methods create contexts with comparable average length and similarity. The core difference lies in the completeness.

Generate a background context from Wikipedia to answer the given question {#question} Keep the length of the document around n words

We observed that GPT 4 effectively controls the output length, whereas other models struggle with this aspect. To address this issue in the latter, we employ multiple values of n and select the one that best matches the retrieved context.

As a result, Figure 9 shows the length distribution of retrieved contexts and contexts generated by various LLMs. The length distribution of retrieved contexts is more concentrated as they consist of text limited to precisely 100 words, along with their titles (Karpukhin et al., 2020). The variation in the length of different retrieved contexts is solely due to the differences in title lengths.

A.2.2 Length Distribution with Different Control Methods

In Section 5.3, we employ three methods, "Nature", "Trunc." and "S-Trunc.", to vary the completeness of generated contexts, while controlling the length at the same time. Figure 10 illustrates the length distribution for generated contexts corresponding to these methods. From the results, we can observe that the contexts generated by original GenRead (Yu et al., 2022) are significantly longer compared to the retrieved contexts.

Table 7 illustrates the average similarity and completeness of three types of generated contexts. "Nature", "Trunc." and "S-Trunc." result in contexts with similar average length and similarity, with preliminary differences in completeness.

A.3 Dataset Size

Table 8 presents the data size of context-conflicting datasets corresponding to various generator-reader pairs. The statistics indicate that conflicting data comprise a substantial proportion across all combinations of generators and readers.

A.4 Effect of Context Order

In the above experiments, retrieved and generated contexts are presented in random order. Previous studies (Xie et al., 2023; Liu et al., 2023; Lu et al., 2022) have found that the model may be sensitive to the order of the input contexts. In their experiments, the input context was either all retrieved (Liu et al., 2023) or all generated (Xie et al., 2023). We conducted experiments to investigate whether the context order impacts the preference for the generated context. The generated and retrieved contexts are concatenated with three different orders: generated-first, retrieved-first, and random. To control the cost of API, this section conducts experiments on the context-conflicting datasets from only the test sets of NQ and TQA. We compute the DiffGR with different context orders respectively.

As shown in Table 9, across all context orders, LLMs consistently show a strong tendency to favor generated contexts. When the retrieved context is positioned first, there is a slight reduction in DiffGR. This reduction may result from the LLMs' preference for generated contexts being partially offset by their bias towards the top context (Liu et al., 2023; Xie et al., 2023).

A.5 More Results on AIG Datasets

Figure 11 shows the DiffGR with different (reader, generator) pairs on their corresponding NQ-AIG datasets. It can be observed that LLMs show a strong tendency to rely on generated contexts across various (reader, generator) pairs.

A.6 More Details about Confirmation Bias

A.6.1 Experiment Setting

We construct counter-memory contexts for each instance on the original AIR context-conflicting subsets, as outlined below:

Counter-Memory Answers Preparation. For each question on the AIR subsets, we substitute the original memory answer (e.g., "Canada") with a same-type yet distinct entity (e.g., "United States"), which serves as the counter-memory answer. Concretely, we employ ChatGPT to associate a differ-

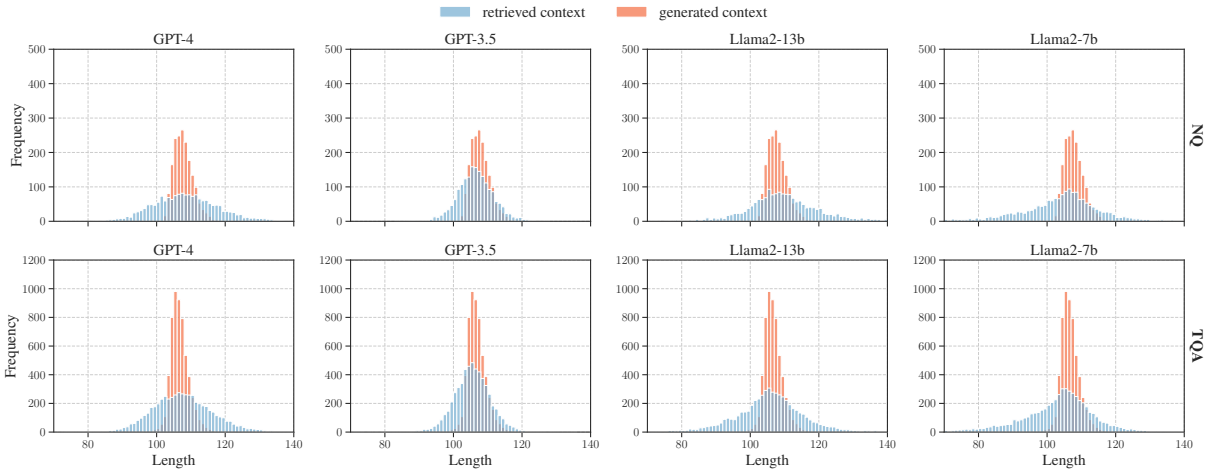


Figure 9: Length distribution of generated and retrieved contexts on different datasets with different generator models.

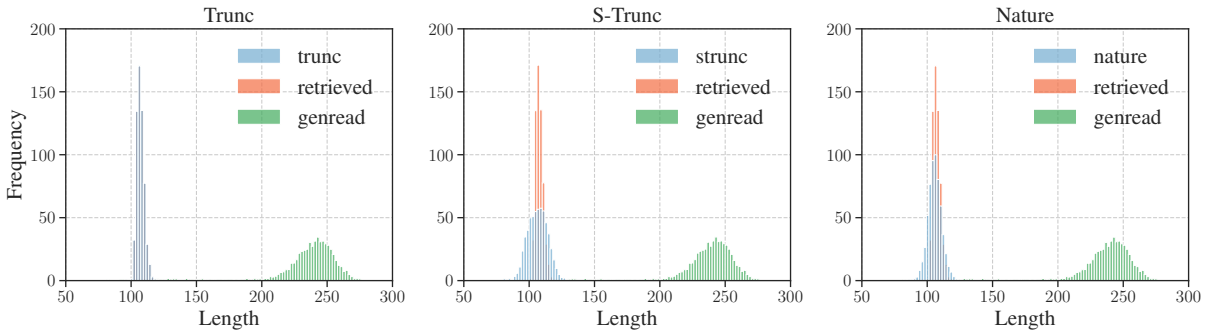


Figure 10: Length distribution of generated and retrieved contexts on the NQ dataset with GPT-3.5 as the generator. "genread" represents the contexts generated by the original GenRead method (Yu et al., 2022). "trunc", "strunc", and "nature" are the generated contexts using three different methods to control the length.

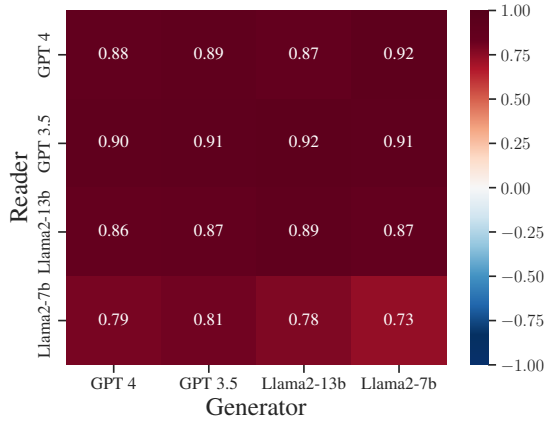


Figure 11: DiffGR with different (reader, generator) pairs on their corresponding NQ-AIG datasets.

ent entity of the same type. It is noteworthy that *the prompt does not incorporate the question*, ensuring that the LLM's counter-memory answer is merely a categorical equivalent rather than a memory related to the question.

> *Give you a reference word, transform it into an analogous word.*
> *Reference: Missouri River*
> *Analogous Word: Mississippi River*
> *Reference: {memory answer}*
> *Analogous Word:*

Furthermore, to facilitate the calculation of DiffGR, we only retain those samples where the counter-memory answer also diverges from a_ϕ^γ, the answer provided by the retrieved context. Formally, the counter-memory answer satisfies $a_\phi^{\varrho'} \neq a_\phi^{\varrho}$ and $a_\phi^{\varrho'} \neq a_\phi^{\gamma}$.

Counter-Memory Contexts Generation. We enforce LLMs to generate a counter-memory context that supports the counter-memory answer. This approach is inspired by the success of using LLMs to generate misinformation (Pan et al., 2023).

Reader	Generator	NQ (3610)		TQA (11313)	
		NQ-AIG	NQ-AIR	TQA-AIG	TQA-AIR
GPT 4	GPT 4	1120	763	1712	681
GPT 4	GPT 3.5	1017	922	-	-
GPT 4	Llama2-13b	730	1461	-	-
GPT 4	Llama2-7b	600	1627	-	-
GPT 3.5	GPT 4	1514	769	2701	794
GPT 3.5	GPT 3.5	1337	857	2389	1042
GPT 3.5	Llama2-13b	875	1318	1781	2119
GPT 3.5	Llama2-7b	701	1502	1471	2641
Llama2-13b	GPT 4	2501	767	4769	741
Llama2-13b	GPT 3.5	2211	899	4210	1038
Llama2-13b	Llama2-13b	1441	1336	2982	2091
Llama2-13b	Llama2-7b	1221	1583	2567	2773
Llama2-7b	GPT 4	2699	668	5370	830
Llama2-7b	GPT 3.5	2435	785	4813	1120
Llama2-7b	Llama2-13b	1569	1220	3526	2051
Llama2-7b	Llama2-7b	1423	1381	3064	2604

Table 8: The data quantities of the constructed subsets for different (Generator, Reader) pairs. NQ and TQA refer to the original datasets (dev+test).

Order	NQ-AIR	TQA-AIR
generated-first	0.699	0.682
retrieved-first	0.665	0.556
random	0.691	0.586

Table 9: DiffGR with different context order on NQ-AIR and TQA-AIR datasets. GPT-3.5 serves as the generator and reader.

Generate a background document in support of the given opinion to the question. Keep the length of the document around n words. Question: {question} Opinion: {counter memory answer} Document:

Following Section 3, the prompt also incorporates a length constraint to mitigate the influence of length, as evidenced by the length statistics presented in Table 10.

Answer Consistency Checking. To verify that the counter-memory context actually supports the counter-memory answer, we retain only those instances where the predicted answer, derived exclusively from the counter-memory context, exactly matches the counter-memory answer:

$$a_{\phi}^{o'} = \phi(q, d_1^{o'})$$

Following the above procedures, we obtain subsets from original AIR datasets, with each instance encompassing a question q, a generated context d_1^o, a retrieved context d_1^γ, a counter-memory context $d_1^{o'}$ and the associated answers for these contexts. Table 11 shows the size of developed context-conflicting subsets with counter-memory contexts.

Context		NQ-AIR		TQA-AIR		ConflictQA(PopQA)	
		Length	Jaccard	Length	Jaccard	Length	Jaccard
Retrieved		107.5	0.111	106.3	0.184	-	-
GPT-3.5	Gen	105.9	0.194	104.9	0.385	62.14	0.169
	Ctr	105.7	0.251	103.8	0.499	100.4	0.236
Llama2-13b	Gen	107.4	0.197	104.3	0.319	-	-
	Ctr	106.7	0.165	103.5	0.310	-	-

Table 10: Average length and Jaccard similarity of different contexts. Gen, Ret, and Ctr respectively represent generated contexts, retrieved contexts, and counter-memory contexts. "-" means the context is not included in this dataset. Detailed discussion about Jaccard similarity is shown in Section 5.2.

Reader & Generator	NQ-AIR	TQA-AIR
GPT-3.5	504	596
Llama2-13b	883	1376

Table 11: The size of context-conflicting subsets with counter-memory contexts.

A.6.2 Discussion

In Table 3, we observe a phenomenon: GPT-3.5 exhibits a stronger bias towards counter-memory contexts compared to the original generated contexts, which does not exist in Llama2-13b. This phenomenon is also observed in Xie et al. (2023) (Table 5 in their work). Upon further investigation, we discover that this may be due to the relatively higher similarity between the counter-memory context generated by GPT-3.5 and the question, as shown in Table 10. In contrast, the generated context and counter-memory context for Llama2-13b exhibit approximately equal levels of similarity. Furthermore, we analyze the original ConflictQA

(PopQA) datasets introduced by Xie et al. (2023) and identify a similar disparity in context similarities, as shown in Table 10. This means that previous works may ignore some significant factors to LLMs' preference, such as similarity (investigated in Section 5.2).

A.7 More Details and Results about Similarity

A.7.1 Similarity Metric

We employ Jaccard similarity to assess the term-based overlap, and BERTScore (Zhang et al., 2020) for evaluating the semantic similarity between contexts and questions. To mitigate the effect of length discrepancies between contexts and questions, we calculate the similarity at the sentence level and then aggregate them to derive the overall context-question similarity. In this work, we adopt a maximum aggregation strategy due to the single-hop nature of the NQ and TQA datasets, where the majority of questions can be answered using a small subset of sentences. We also try the average aggregation strategy and observe similar results.

Figure 12 illustrates the distribution of similarity when employing maximum and average aggregation methods. It is observable that the generated contexts exhibit a markedly higher degree of similarity regardless of the aggregation method used. Furthermore, this disparity in similarity is more pronounced with maximum aggregation, as contexts typically contain sentences that are irrelevant, which dilute the similarity scores when an average aggregation is applied.

A.7.2 Similarity Distribution

Figure 13 and 14 show the similarity distribution of retrieved and generated contexts across various generators. All LLM-generated contexts exhibit a higher similarity over retrieved contexts.

A.7.3 Effect of Similarity.

Figure 15 demonstrates a general trend that on slices with a smaller average similarity gap, LLMs exhibit a reduced preference for generated context.

A.7.4 Cases about Similarity

Table 12 shows examples that contain contexts with different similarities to the question. The contexts with high similarity typically directly support answering by repeating the phrasing in the question. Conversely, the contexts with low similarity introduce more challenges, often necessitating an understanding of synonyms and even some inferences. These observations indicate that text similarity can partly reflect the relevance between a question and a context, as well as the difficulty the LLM encounters in identifying potential answers.

A.8 Cases about completeness

Table 13 provides some examples to facilitate the understanding of completeness. From the cases, we observe that retrieved contexts and "Trunc." often contain incomplete sentences. Additionally, compared to "S-Trunc", "Nature" typically exhibits greater semantic completeness. Specifically, "Nature" often encompasses a full logical structure of an article, including an introduction, discussion, and conclusion, whereas "S-Trunc" may terminate abruptly.

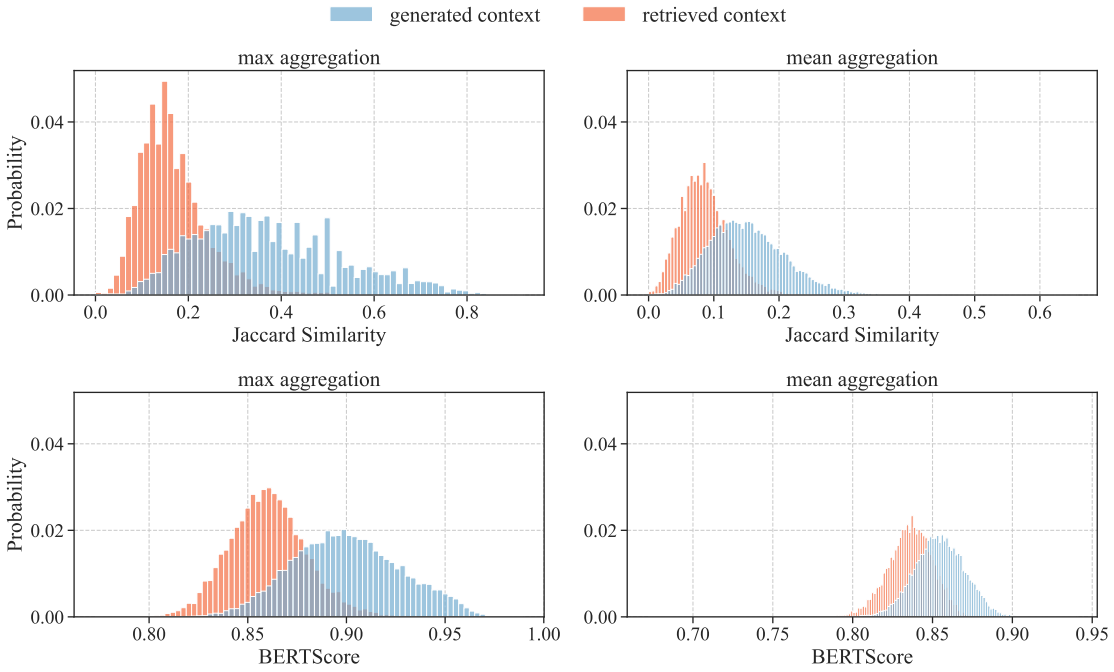


Figure 12: Similarity distribution with maximum or mean aggregation strategies. Generated contexts consistently exhibit higher similarity across two aggregation strategies.

	TQA-AIR Example	TQA-AIG Example
Question	Between 1959 and 1967 which city was the capital of Pakistan (Islamabad was being built)?	Who is the most successful UK solo artist in the USA?
Golden Answer	Rawalpindi	Elton John
Generated Context	Between 1959 and 1967, the capital of Pakistan was Karachi. Karachi is the largest city in Pakistan and is located on the southern coast of the country... **Jaccard Similarity**: 0.47 **BertScore**: 0.93	Elton John is the most successful UK solo artist in the USA. Born Reginald Kenneth Dwight in 1947, he adopted the stage name Elton John in the late 1960s... **Jaccard Similarity**: 0.69 **BertScore**: 0.93
Retrieved Context	...was first shifted temporarily to Rawalpindi in the early 60s, and then to Islamabad when essential development work was completed in 1966... **Jaccard Similarity**: 0.16 **BertScore**: 0.85	...In 2009, Jay Sean's single "Down" reached the number one spot on the "Billboard" Hot 100 and sold millions in the United States, making him the most successful male UK urban artist in US chart historyät the time... **Jaccard Similarity**: 0.14 **BertScore**: 0.86
Model output	Karachi	Elton John

Table 12: Some examples where both the generator and reader are GPT-3.5. We highlight the incorrect candidate answers in the context in pink, and the correct answers in the context in green.

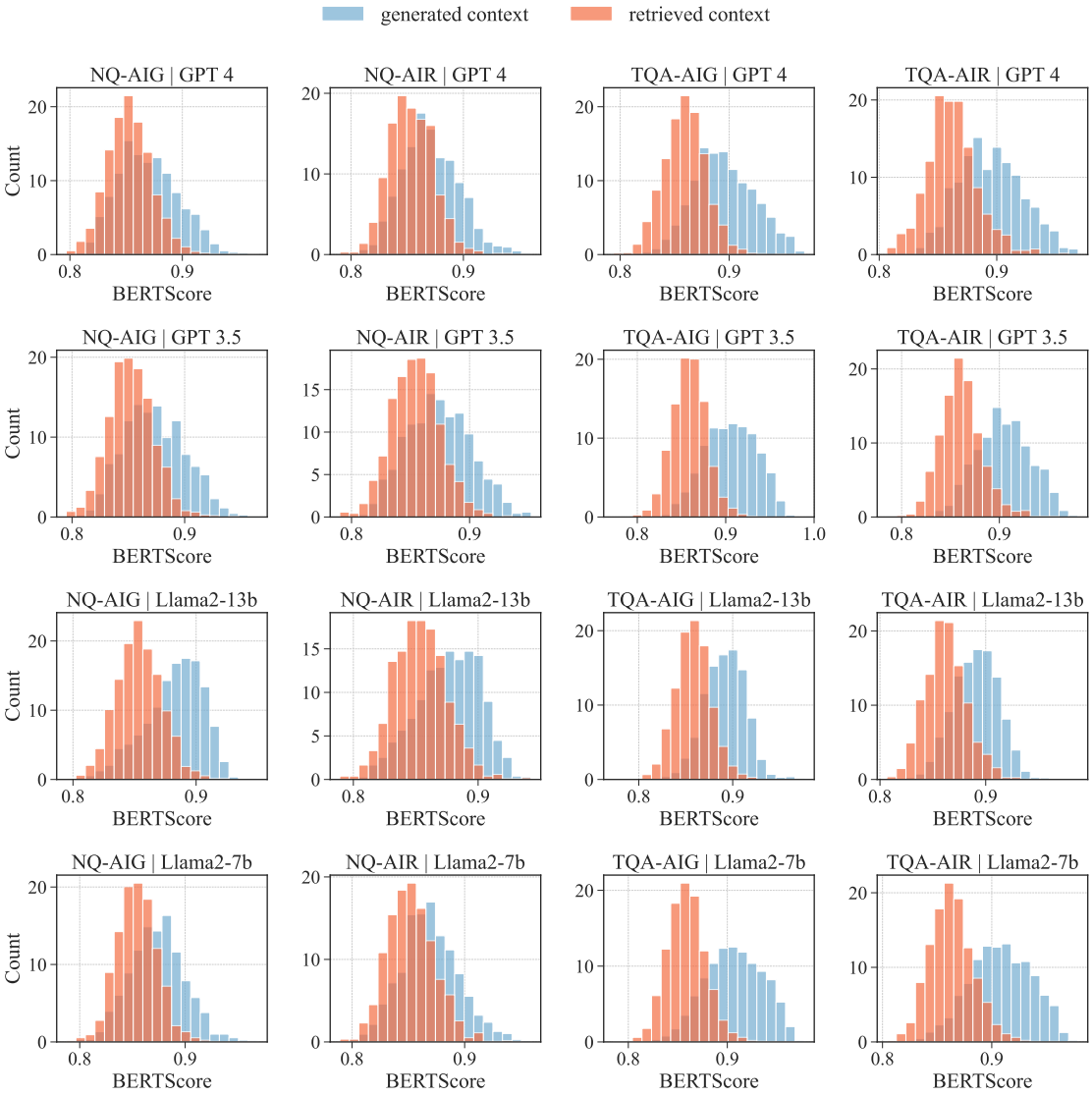


Figure 13: BERTScore distribution of retrieved contexts and contexts generated by different LLMs. All LLM-generated contexts exhibit a higher similarity over retrieved contexts.

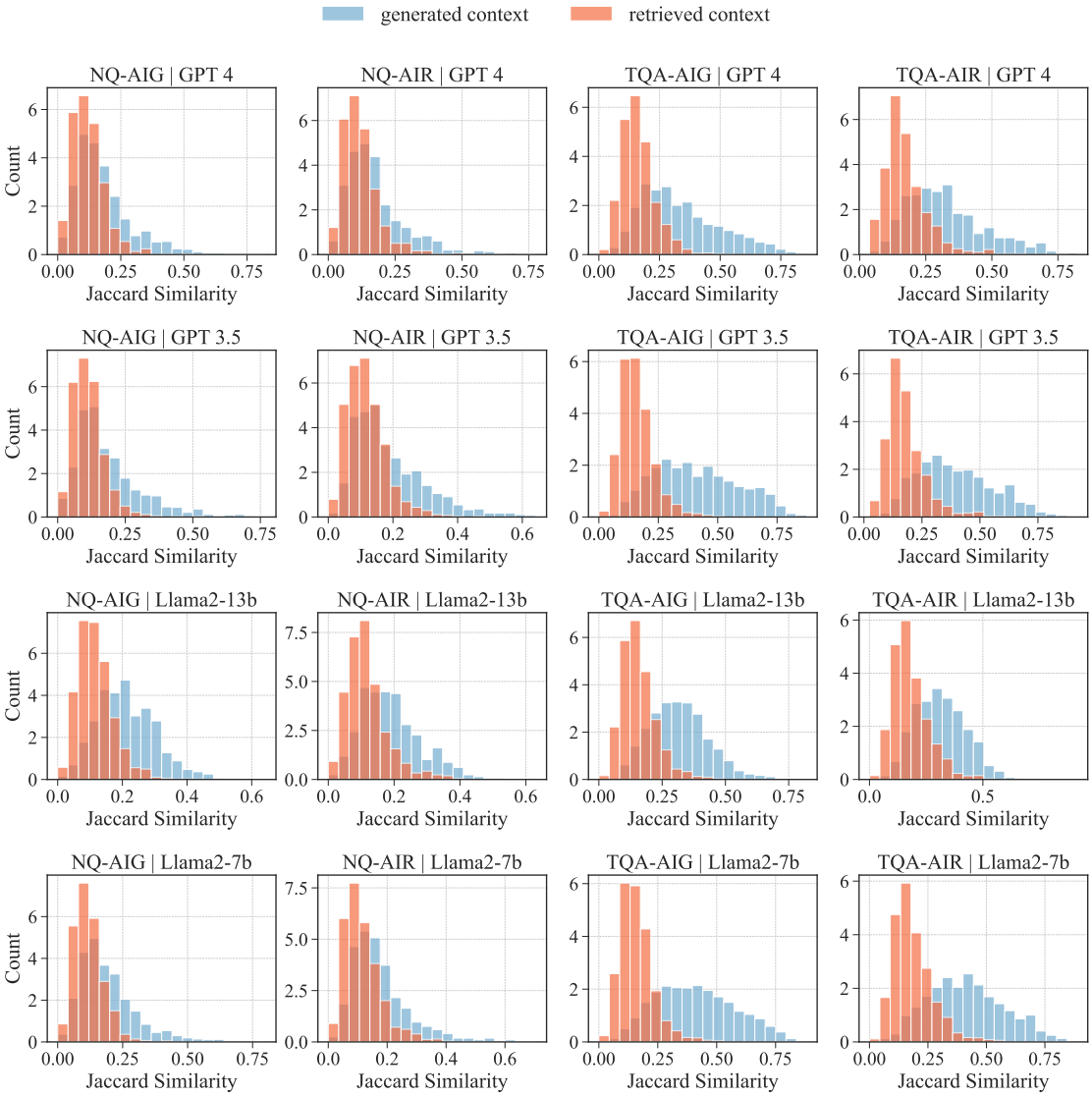
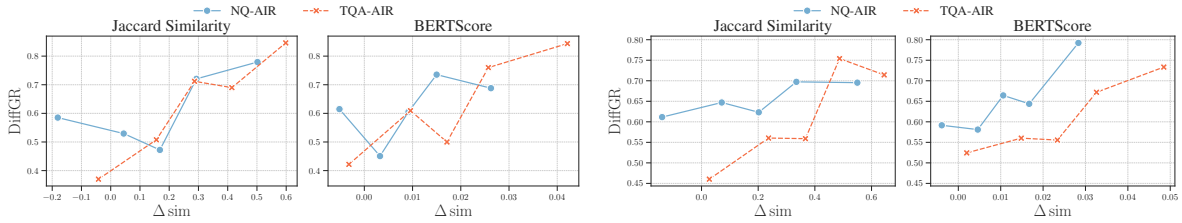
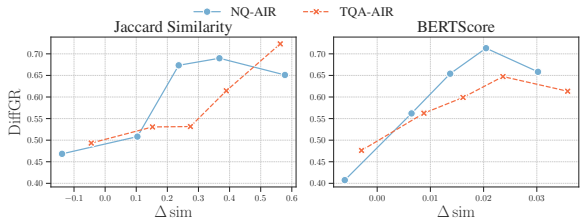


Figure 14: Jaccard Similarity distribution of retrieved contexts and contexts generated by different LLMs. All LLM-generated contexts exhibit a higher similarity over retrieved contexts.

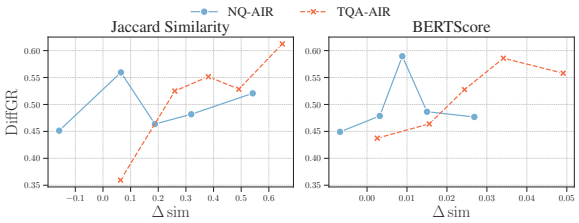


(a) GPT 4

(b) GPT 3.5



(c) Llama2-13b



(d) Llama2-7b

Figure 15: The generation ratio in slices with different average Δ sim. Δ sim is the difference in similarity between the generated context and the retrieved context.

Example
Question: In World War I, what did the Allies call the defence line that was known as the 'Siegfried Line' to the Germans? **Answer:** ['Hindenberg Line', 'Hindenburg line', 'Battles of the Hindenburg Line', "'HINDENBURG LINE'", 'Advance to the Hindenburg Line', 'Hindenburg Line'] **Retrieved context:** Title: Siegfried Line Content: Siegfried Line The term Siegfried Line refers to two different German defensive lines, one during the First World War and the other during the Second World War. The original was a line of forts and tank defences built by Germany in northern France during 1916–17, known in German as the Siegfriedstellung ("Siegfried Position"). The British referred to it as the "Hindenburg Line", a name subsequently adopted by the rest of the Allies. After WW1 the German Fortifications East of the Rhein were required to be demolished due to the Treaty of Versailles. The second, commonly known in German as the **Generated context (Nature):** The Allies referred to the German defence line known as the 'Siegfried Line' as the 'Westwall' during World War I. Constructed between 1936 and 1940, the Westwall was a system of fortifications built along Germany's western border. It stretched from the Swiss border to the Dutch border, covering approximately 630 kilometers. The line consisted of concrete bunkers, tank traps, and anti-tank ditches, aiming to deter any potential invasion from the west. The Allies recognized the formidable nature of this defensive barrier and thus referred to it as the Westwall, while the Germans named it the Siegfried Line, after the mythical hero Siegfried from German folklore. **Generated context (Trunc.):** The Allies referred to the German defence line known as the "Siegfried Line" as the "Westwall" during World War I. The Westwall was a series of fortifications constructed by Germany along its western border, stretching from the Swiss border to the Netherlands. It was named after the German Emperor, Wilhelm II, who was commonly referred to as "Kaiser Wilhelm" or "Siegfried" by the Allies. The construction of the Westwall began in the 1930s as a response to the growing tensions in Europe and the threat of war. The line consisted of a network of bunkers, pillboxes, tank traps, and other defensive structures, strategically positioned **Generated context (S-Trunc.):** The Allies referred to the German defence line known as the "Siegfried Line" as the "Westwall" during World War I. The Westwall was a series of fortifications constructed by Germany along its western border, stretching from the Swiss border to the Netherlands. It was named after the German Emperor, Wilhelm II, who was commonly referred to as "Kaiser Wilhelm" or "Siegfried" by the Allies. The construction of the Westwall began in the 1930s as a response to the growing tensions in Europe and the threat of war. The line consisted of a network of bunkers, pillboxes, tank traps, and other defensive structures, strategically positioned to impede any potential invasion from the west.

Table 13: Examples with retrieved contexts and generated contexts. "Nature", "Trunc." and "S-Trunc." represent three types of generated contexts with different completeness. Retrieved contexts often contain incomplete sentences.

Reader	Generator	NQ-AIG			NQ-AIR			TQA-AIG			TQA-AIR		
		Gen	Ret	Others	Gen	Ret	Others	Gen	Ret	Others	Gen	Ret	Others
GPT 4	GPT 4	0.9125	0.0589	0.0286	0.7379	0.1743	0.0878	0.9387	0.0304	0.031	0.7651	0.1762	0.0587
GPT 3.5	GPT 3.5	0.9177	0.0449	0.0374	0.7083	0.1470	0.1447	0.9347	0.026	0.0393	0.7332	0.1775	0.0893
Llama2-13b	Llama2-13b	0.8966	0.0500	0.0534	0.7216	0.1811	0.0973	0.9071	0.0433	0.0496	0.7212	0.1918	0.0870
Llama2-7b	Llama2-7b	0.7041	0.1082	0.1876	0.6148	0.2071	0.1781	0.7973	0.0927	0.1100	0.6555	0.2101	0.1344

Table 14: "Gen" denotes the proportion of responses that match the candidate answer within generated contexts, whereas "Ret" refers to the proportion of matching the candidate answer within retrieved contexts. "Others" encompasses the proportion of responses that do not align with either category. *Given that the proportion of "Others" is significantly lower relative to the disparities between "Gen" and "Ret", its impact on the conclusions of this paper is negligible.*