RAC: Efficient LLM Factuality Correction with Retrieval Augmentation

Changmao Li
University of California, Santa Cruz
changmao.li@ucsc.edu

Jeffrey Flanigan
University of California, Santa Cruz
jmflanig@ucsc.edu

Abstract

Large Language Models (LLMs) exhibit impressive results across a wide range of natural language processing (NLP) tasks, yet they can often produce factually incorrect outputs. This paper introduces a simple but effective low-latency post-correction method, **Retrieval Augmented Correction (RAC)**, aimed at enhancing the factual performance of LLMs without requiring additional fine-tuning. Our method is general and can be used with any instruction-tuned LLM, and has greatly reduced latency compared to prior approaches. RAC decomposes the LLM's output into atomic facts and applies a fine-grained verification and correction process with retrieved content to verify and correct the LLM-generated output. Our extensive experiments show that RAC yields up to 30% improvements over the LLM baselines across two popular factuality evaluation datasets, validating its efficacy and robustness with and without the integration of Retrieval-Augmented Generation (RAG) across different LLMs. Notably, our method has reduced latency up to 40x and reduced token consumption up to 7x compared to previous state-of-the-art post-correction approaches with similar or better performance.[1]

1 Introduction

Recently Large Language Models (LLMs) have markedly changed the world of natural language processing (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023; OpenAI, 2022, 2023). Although LLMs can achieve superior performance on many NLP tasks, hallucination is a known issue for LLMs (Rawte et al., 2023; Ji et al., 2023; Zhang et al., 2023; Ye et al., 2023a; Huang et al., 2023). In particular, factually incorrect content generated by LLMs can be explicitly harmful to the application of LLMs (Li et al., 2024), such as providing incorrect medical suggestions or wrong information for educational purposes. Misinformation can cause unpredictable harm to humans when LLMs are used ubiquitously. Enhancing LLMs with better factuality can improve LLMs performance (Lee et al., 2022) and be less harmful to users.

To alleviate this factuality problem, previous research has investigated incorporating retrieved knowledge from a collection of documents into the LLM's context, called retrieval augmented generation (RAG) (Chen et al., 2017; Guu et al., 2020; Lewis et al., 2020; Izacard et al., 2022). RAG first retrieves from a document set to acquire information related to the input task. Retrieval can be done with a search engine such as Google or a from a corpus such as Wikipedia. The retrieved information is then input to the LLM with the task instructions. This predisposes the LLM to generate content that is faithful to the retrieved content, achieving improved factual performance (Lewis et al., 2020; Asai et al., 2024b). However, RAG does not guarantee factual content; even with entirely correct retrieved content in the context, the LLMs can still generate factually incorrect output (Wu et al., 2024). Possible reasons LLMs may still generate incorrect output is due to constraints and uncertainty in their internal states (Neeman et al., 2022; Mallen et al., 2023) or fine-tuning on new knowledge (Gekhman et al., 2024).

Prior work has attempted to improve RAG by improving the quality of retrieval (Asai et al., 2024a) or attempting to correct retrieved content (Yan et al., 2024). We find these steps are unnecessary if we use RAG with Google search[2] (Wei et al., 2024) , it produces results that are over ten points higher than previous baselines. Therefore, we focus on correcting generated output using the retrieved content. While we are not the first to use retrieved content for factuality correction (Gao et al., 2023; Gou

[1] Our code is at https://github.com/jlab-nlp/Retrieval-Augmented-Correction

[2] See Table 3, baseline results w/ RAG.

November 4-9, 2025 ©2025 Association for Computational Linguistics

	# of generation API calls	# of search queries for each sentence or iteration	# of correction iterations	total # of retrieval calls
RAG (Lewis et al., 2020)	1	1	0	1
RARR (Gao et al., 2023)	1	n_q	1	$n_s * n_q$
CRITIC (Gou et al., 2024)	1	1	3	3
EVER (Kang et al., 2024)	n_s	3	2	3*n_s
RAC (ours)	**1**	**1**	**1**	**1**

Table 1: Number of API calls and retriever calls compared to previous post-correction with retrieval methods. n_s is the number of generated sentences, and n_q is the number of generated questions per sentence (for RARR only). For experiments measuring latency, see §7.

	One-time Retrieval	**Fact-level Correction**	**Paragraph-wise Correction**	**Single Round Correction**	**Fact-level Verification**
RARR	✗	✗	✗	✗	✗
CRITIC	✗	✗	✓	✗	✗
EVER	✗	✗	✗	✗	✗
RAC	✓	✓	✓	✓	✓

Table 2: RAC compared to previous approaches.

et al., 2024; Kang et al., 2024), we greatly improve upon it for efficiency (Table 1) and performance (Table 3).

We propose a method we call **Retrieval Augmented Correction (RAC)**. RAC verifies and corrects LLM-generated content using retrieved knowledge to ensure factuality. Specifically, RAC breaks down LLM-generated content into atomic facts, leverages retrieved knowledge to verify or correct these atomic facts, and then revises the LLM output accordingly.

Lightweight and general, our approach can be viewed as a post-processing component of RAG to improve factuality performance further. Experiments show that our approach provides substantial improvements over all prior results across two popular factuality evaluation datasets, with improvements up to 30% compared to the baseline LLM.

By only correcting once at the atomic fact level, our approach has greatly reduced latency compared to similar prior methods with a similar or better performance. Additionally, our approach demonstrates that in some cases, the performance of our method without RAG can surpass that with RAG, indicating the robustness and effectiveness of our method even in the absence of retrieval augmented generation.

In summary, our contributions are the following:

- We proposed a plug-and-play post-processing component to RAG which improves the factuality performance of RAG-based LLMs. No additional retrieval beyond the retrieval step of RAG is necessary.
- The proposed verification, correction, and revision modules are fine-tuning free and can be applied to real applications with RAG or without RAG.
- Experimental results show the approach can improve factuality by up to 30% compared to the baseline LLM, depending on the application.
- Our approach exhibits greatly reduced latency while achieving similar or better performance compared to prior correction by retrieval methods (Table 6).

2 Related Work

Hallucination has been a known issue for generation tasks, especially when using LLMs (Maynez et al., 2020; Rawte et al., 2023; Ji et al., 2023; Zhang et al., 2023; Ye et al., 2023a; Huang et al., 2023). Our work focuses on one of the hallucination types for LLMs, factual incorrectness. There are four lines of work regarding reducing factual incorrectness: 1) from the LLM decoding perspective (Li et al., 2023; Chuang et al., 2024; Das et al., 2024), 2) from the factual enhancement perspective using retrieval augmentation (Chen et al., 2017;

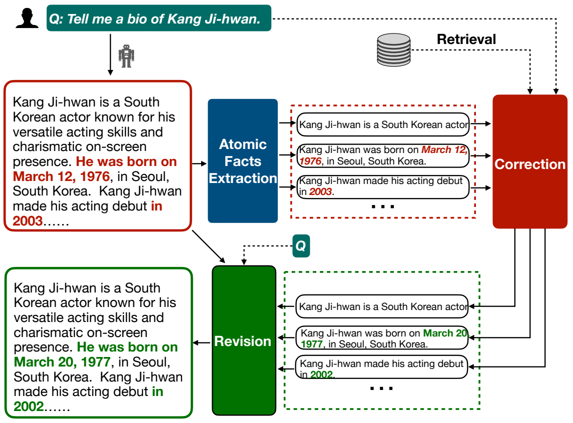
Q: Tell me a bio of Kang Ji-hwan.
Retrieval
Kang Ji-hwan is a South Korean actor known for his versatile acting skills and charismatic on-screen presence. He was born on March 12, 1976, in Seoul, South Korea. Kang Ji-hwan made his acting debut in 2003......
Atomic Facts Extraction
Kang Ji-hwan is a South Korean actor
Kang Ji-hwan was born on March 12, 1976, in Seoul, South Korea.
Kang Ji-hwan made his acting debut in 2003.
...
Correction
Q
Kang Ji-hwan is a South Korean actor known for his versatile acting skills and charismatic on-screen presence. He was born on March 20, 1977, in Seoul, South Korea. Kang Ji-hwan made his acting debut in 2002......
Revision
Kang Ji-hwan is a South Korean actor
Kang Ji-hwan was born on March 20 1977, in Seoul, South Korea.
Kang Ji-hwan made his acting debut in 2002.
...

Figure 1: Overview of RAC without retrieval augmented generation (RAG). Note we do not use a verification stage (see Fig. 2 below) when not using RAG, since we find that many sentences need to be corrected anyway.

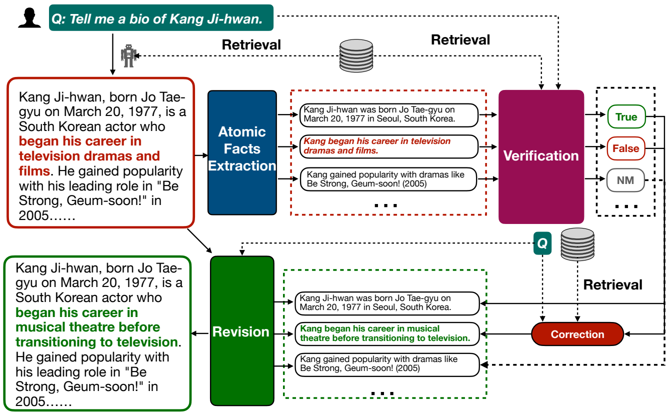
Q: Tell me a bio of Kang Ji-hwan.
Retrieval
Retrieval
Kang Ji-hwan, born Jo Tae-gyu on March 20, 1977, is a South Korean actor who began his career in television dramas and films. He gained popularity with his leading role in "Be Strong, Geum-soon!" in 2005......
Atomic Facts Extraction
Kang Ji-hwan was born Jo Tae-gyu on March 20, 1977 in Seoul, South Korea.
Kang began his career in television dramas and films.
Kang gained popularity with dramas like Be Strong, Geum-soon! (2005)
...
Verification
True
False
NM
...
Q
Retrieval
Kang Ji-hwan, born Jo Tae-gyu on March 20, 1977, is a South Korean actor who began his career in musical theatre before transitioning to television. He gained popularity with his leading role in "Be Strong, Geum-soon!" in 2005......
Revision
Kang Ji-hwan was born Jo Tae-gyu on March 20, 1977 in Seoul, South Korea.
Kang began his career in musical theatre before transitioning to television.
Kang gained popularity with dramas like Be Strong, Geum-soon! (2005)
...
Correction

Figure 2: Overview of RAC with RAG. The verification and correction stages use the same retrieved documents as RAG. NM means fact not mentioned in the retrieved documents. When using RAC with RAG, we add a verification stage since most LLM generated content is correct; we only need to correct false content.

Lewis et al., 2020; Izacard et al., 2022; Asai et al., 2023; Yoran et al., 2024; Luo et al., 2023; Jiang et al., 2023b; Yan et al., 2024) or fine-tuning (Lee et al., 2022; Yang et al., 2023; Tian et al., 2024), 3) from a self-correction or self-alignment perspective (Zhang et al., 2024; Wang et al., 2024), and 4) from a post-correction using retrieved content perspective (Gao et al., 2023; Gou et al., 2024; Kang et al., 2024).

The area we focus on is post-correction using retrieved content. All of these works, including ours, are LLM-based post-correction pipelines, which means LLMs are leveraged to post-correct themselves using retrieved content. Gao et al. (2023) introduces RARR, which generates several questions for each output, retrieves Bing Search for each question one by one, and repeatedly revises the output based on the retrieved content by iterating questions. Two major drawbacks to this method are that the retrieval process is expensive due to many calls to the retriever, and the correction for this method is cascaded, which can introduce errors. Our approach instead works by breaking down LLM outputs into atomic facts, verifying these facts against retrieved relevant documents, and then revising the output accordingly. Gou et al. (2024) introduces CRITC, which iteratively conducts correction using LLMs with retrieved knowledge, where for each iteration, they apply the LLM to generate a query to retrieve the knowledge base and revise the output based on the history of revising and decide the most possible answer. Kang et al. (2024) introduces EVER, which conducts sentence-by-sentence generation and correction in real-time generation, and for each sentence correction, they retrieve using different features multiple times. Their approach largely increases the latency and possibly introduces more hallucinations. Compared to previous approaches, our approach conducts the correction in a more fine-grained manner all at once, reduces the burden of generating questions and correcting several iterations, avoids more hallucinations by not correcting already corrected atomic facts, and only needs to retrieve the knowledge base once for each output based on their task instructions. Our method improves both efficiency and effectiveness compared to the previous approaches. Table 1 shows the number of API calls and retriever calls for previous correction methods with retrieved content. Table 2 shows the differences between our proposed RAC and all previous related work.

3 Retrieval Augmented Correction (RAC)

Our approach to improving factuality is to retrieve documents relevant to the input and use these documents to revise the output to be factual effectively with few LLMs API Calls. Our approach can be used with or without retrieval augmented generation (RAG). If used with RAG, then the retrieved documents for RAG are also used for RAC.

3.1 Overview

Figures 1 and 2 show the approach overview with or without RAG. We first break down the original output from the LLM into atomic facts (Min et al., 2023), which allows our method to do a fine-grained correction of individual facts.

For LLM generation without RAG (Fig. 1), we propose to add correction and revision stages. The correction stage directly corrects the extracted incorrect statements and keeps the correct statements based on the retrieved document sets for the task input. The statements are then fed into the revision module to revise the original LLM output.

For LLMs with RAG (Fig. 2), we find that adding a verification stage improves performance. RAG performs well enough that many of the statements do not need to be corrected; simply correcting all statements will introduce more hallucinations, which harm the performance rather than benefit. Considering this, we add a verification component to first to verify the statements and then only correct the false statements. This reduces the hallucinations during RAC since it ensures that truth statements are kept without passing into the correction stage.

Both our proposed correction and verification stages use the same retriever used in RAG. The retriever retrieves related factual documents from the input. We apply post-processing for the retrieved documents to make them related, faithful, and concise.

The paper is organized as follows. We introduce our retrieval method and post-processing (§3.2), describe atomic statement extraction (§3.3), and introduce the correction (§3.4) and revision (§3.5) stages for LLMs without RAG. We review the basics of RAG (§4.1) and discuss the additional verification stage (§4.2), and the corresponding correction (§4.3) and revision (§4.4) stages when used with RAG. Finally, we present experiments (§5), results (§6), ablations (§7), and a case study (§H).

3.2 Retrieval

The retrieval step includes two parts: retrieval and retrieval post-processing. Retrieval directly retrieves the factual documents using the task input from a trusted knowledge source (Guu et al., 2020; Lewis et al., 2020; Izacard et al., 2022). The retrieval post-processing conducts two things: 1. filtering out unrelated documents or reranking and picking up top-k documents; 2. truncating or compressing the documents based on the LLM maximum context window length.

Let X be the task input and R be the retrieved outputs. The retrieval and retrieval post-processing are formulated as follows:

$$R = \texttt{Retrieve}(X) \tag{1}$$

$$R' = \texttt{Compress}(\texttt{Rerank}(R)) \tag{2}$$

R' represents the post-processed retrieved documents. Our post-processing contains two operations: `Compress`, which truncates and compresses, and `Rerank`, which filters and reranks the documents.

The retrieval post-processing ensures the retrieved documents are related, faithful, and compact for the task input. This is important since incorrect or unfaithful retrieved contents can directly cause the failure of the proposed approach, as intuitively, one cannot use incorrect contents to verify and correct a statement. In the ablation studies, we discuss how the correctness of the retrieved content can affect our approach. The details of retrieval and post-processing processes are in the appendix.

3.3 Atomic Fact Extraction

Atomic fact extraction breaks down the original task outputs from LLMs into several independent factual statements. This strategy is inspired by

Factscore (Min et al., 2023). This can lead to targeted corrections to the statements, further enhancing the correction and providing a clear interpretation of which part of the original outputs are corrected.

Let the LLM task outputs be M_{out}, and the extracted atomic facts S. Then:

$$S = \texttt{Extract}(M_{out}) = \{s_1, s_2, s_3, ..., s_n\} \quad (3)$$

where n is the number of the statements, s_i is the ith atomic fact.

3.4 Correction (C)

We add a retrieval process into the factual correction stage, which improves upon self-correction, which was introduced in prior work (Wang et al., 2024). We find this this step greatly enhances factual correction. The correction stage corrects the statements based on the verification results using retrieved document sets; then, the revision stage can use them to revise the original LLM task output. This stage ensures that all statements fed into the revision stage are faithful. We also include the task input X during correction to avoid diverging from the task input. The processed statements C:

$$\begin{aligned} C = \{&\texttt{Correct}(s_1, R', X), ..., \\ &\texttt{Correct}(s_n, R', X)\} \end{aligned} \quad (4)$$

3.5 Revision (*R*)

The revision stage uses previously corrected statements to revise the original task outputs. Unlike prior work using self-revision (Madaan et al., 2023; Ye et al., 2023b), we use corrected statements from the correction stage to guide the revision stage, which reduces hallucinations. To enable the revision to be still consistent with the task input, we include the task input X, and the revised outputs O is:

$$O = \texttt{Revise}(X, M_{out}, C) \quad (5)$$

4 Combining RAC with RAG

To further improve results, we can combine our proposed method with retrieval augmented generation (RAG), which we discuss in this section. When used with RAG, the retrieved documents for the verification, correction and revision stages are the same as the RAG retrieved documents.

4.1 RAG for LLMs

We first review retrieval augmented generation for LLMs.

Given an input X, RAG first retrieves documents related to X from a document set $D = \{d_1, ...d_m\}$ (d_i represents the ith document) to obtain a relevant document set $R = \{d_1, ..., d_n\}$. The generation probability Y is the standard next-token prediction probability conditioned on the input context X and retrieved relevant documents R.

$$P(Y|X, R) \quad (6)$$

RAG suffers from potential hallucination issues because retrieved documents may contain other unrelated information that could cause hallucination (Shi et al., 2023), and the retrieved documents may contradict what the model initially learns internally and the model sticks to their original training because of internal prior is very strong during training (Wu et al., 2024). To alleviate the above issues, we need to take extra steps to verify and correct using retrieved content during post-generation steps to further reduce the hallucination caused by the above reasons.

4.2 Verification (V)

Since many of the extracted statements after using RAG are correct, we find we do not need to correct all statements. Instead, we added a verification stage to enable the LLMs to correct only false statements, which reduces the hallucinations introduced by correcting already correct statements. Some previous self-verification works consider only LLM self-consistency (Manakul et al., 2023) or require additional models for verification (Mündler et al., 2024). Unlike these previous work, we add a retrieval process to the verification. The verification is done using LLMs without additional training. The verification stage verifies the extracted atomic facts using the retrieved documents. The verified results are then fed into the correction stage.

Let V be the verification results, Then:

$$V = \texttt{Verify}(R', S) = \{b_1, b_2, b_3, ..., b_n\} \quad (7)$$

where b_i is the verification result for the ith atomic fact. The value of b_i is **True**, or **False**, or **Not Mentioned**. **True** means a similar statement can be found in the retrieved documents and has the same meaning, which indicates the statement is consistent with the retrieved documents. **False** means a similar statement can be found in the retrieved documents but has a different meaning, which indicates the statement contradicts the retrieved documents. **Not Mentioned** means a similar statement cannot be found in the retrieved documents, which indicates the statement cannot be verified by the retrieved documents.

4.3 Correction (C)

Let S_t, S_f, and S_{nm} be the set of atomic facts labeled by the verifier as True, False, or Not Mentioned, respectively. We use the following strategy to make the correction:

$$C = S_t \cup \{\texttt{Correct}(s, R', X) | s \in S_f\} \cup S_{nm} \quad (8)$$

where True statements are always kept, False statements always be corrected, and Not Mentioned statements are kept. We also experimented with a strategy that removes all not mentioned statements, but in preliminary experiments found it to give worse results:

$$C = S_t \cup \{\texttt{Correct}(s, R', X) | s \in S_f\} \quad (9)$$

4.4 Revision (R)

The revision stage with the verification stage is the same as the revision stage without verification (see §3.5). However, to avoid more newly introduced hallucinations for the initial model generations during revision, we also tried a Keep All True (KAT) strategy: only revise model generations with one or more incorrect statements during verification and keep those without any incorrect statements unchanged. Our ablation study in the appendix (§F) analyzes the performance of this strategy.

5 Experimental Settings

Datasets and Metrics We use the three available datasets for factuality evaluation on open-ended long-form generation: TruthfulQA (Lin et al., 2022), biography generation (Min et al., 2023) and LongFact (Wei et al., 2024). We focus on long-form generation, and do not evaluate on classification tasks, because they are not suitable for our method. For TruthfulQA we use the generation task. Following the TruthfulQA evaluation, we report the accuracy of BLEURT (Sellam et al., 2020), BLEU (Papineni et al., 2002), and ROUGE (Lin, 2004). Accuracy is computed by comparing the predictions with correct and incorrect answers collected. We use these metrics because Xu et al. (2023) shows that BLEURT and ROUGE perform only slightly worse than GPT-judge, and LLMs are known to suffer from the preference leakage problem (Li et al., 2025; Ye et al., 2025).[3] Biography is a long-form generation task where the evaluation metric is Factscore. Factscore uses OpenAI GPT-3 to judge the accuracy of factuality compared to the corresponding Wikipedia biography. Since GPT-3 is no longer available, all reported numbers for Factscore use GPT-3.5-Turbo-Instruct. LongFact is an MMLU-style benchmark for long-form factuality, they generated LongFact by prompting GPT-4 to generate questions that ask about a specific concept or object within a given topic and that require a long-form response containing multiple detailed factoids. We use its corresponding SAFE (Wei et al., 2024) to evaluate.

Models and Baselines We use GPT-3.5-Turbo (OpenAI, 2024), GPT-4o (OpenAI, 2024), Llama 2-7B-Chat (Touvron et al., 2023), Llama3-8B-Instruct (Meta, 2024), and Mistral-7B-Instruct (Jiang et al., 2023a) as baseline models to evaluate our method on closed and open LLMs. Appendix B has hyperparameters.

We report numbers for all previous state-of-the-art baselines. To compare our method to the previous method RARR (Gao et al., 2023), CRITIC (Gou et al., 2024) and EVER (Kang et al., 2024), we run them using the same model (GPT-3.5-Turbo) and search engine (Google search) or retrieved documents as ours. EVER is reproduced in a post-correction manner per sentence rather than correction of each sentence during generation to speed up experiments.

6 Results

Results on Biography and TruthfulQA are shown in Table 3. Results on LongFact are shown in Table 4. We report our findings for each dataset below.

Results on TruthfulQA For the TruthfulQA dataset, our method performs similarly or better than previous methods across different LLMs and metrics with and without RAG.

We note the instruction-tuned model Llama2-7B-Chat is better than previous methods using the Llama2-7B model (models listed under "Llama2-7B With Additional Training" in Table 3), in both RAG and non-RAG settings. In RAG settings, previous methods RARR (Gao et al., 2023), CRITIC (Gou et al., 2024) and EVER (Kang et al., 2024)

[3] Additionally GPT-Judge accuracy for this task is unavailable because OpenAI deprecated the related evaluation model.

	Biography		**TruthfulQA**					
	Factscore		BLEURT	BLEU	ROUGE	BLEURT	BLEU	ROUGE
	w/o RAG	**w/ RAG**	**w/o RAG**			**w/ RAG**		
Llama2-7B (With Additional Training)								
DoLA (Chuang et al., 2024)	39.0±2.9	-	39.0±1.4	36.4±1.3	36.0±1.3	-	-	-
FACTTUNE-MC (Tian et al., 2024)	42.7±3.0	-	-	-	-	-	-	-
SELF-EVAL-SKT (Zhang et al., 2024)	46.5±3.1	-	-	-	-	-	-	-
Self-RAG((Asai et al., 2023))*	-	81.2±1.9	-	-	-	52.8±1.4	41.2±1.4	43.9±1.4
Self-CRAG((Yan et al., 2024))*	-	86.2±1.5	-	-	-	53.4±1.4	40.0±1.4	40.1±1.4
Instruct LLMs (Without Additional Training)								
GPT-3.5-Turbo	78.9±2.0	93.4±0.8	58.3±1.4	48.7±1.4	51.5±1.4	67.2±1.3	62.2±1.4	65.0±1.3
+ RARR (Gao et al., 2023)*	79.6±2.0	85.5±1.5	58.9±1.4	47.4±1.4	50.6±1.4	63.3±1.3	58.1±1.4	62.3±1.4
+ CRITIC (Gou et al., 2024)*	85.2±1.5	92.0±0.9	60.8±1.4	47.2±1.4	52.0±1.4	61.8±1.4	46.3±1.4	51.0±1.4
+ EVER (Kang et al., 2024)*	**93.3**±0.8	93.7±0.7	59.6±1.4	49.2±1.4	50.6±1.4	55.7±1.4	44.6±1.4	48.8±1.4
+ RAC (ours)	88.2±1.3	**93.7**±0.7	**61.9**±1.9	**50.7**±1.4	**52.8**±1.4	**70.4**±1.2	**67.9**±1.3	**70.4**±1.2
GPT-4o	90.6±1.0	92.7±0.8	69.9±1.2	57.5±1.4	61.3±1.4	77.8±1.0	73.1±1.1	76.3±1.0
+ RARR (Gao et al., 2023)*	91.1±1.0	92.6±0.8	70.4±1.2	57.8±1.4	61.6±1.4	77.6±1.0	72.8±1.1	76.1±1.0
+ CRITIC (Gou et al., 2024)*	90.6±1.0	92.6±0.8	**72.5**±1.2	58.3±1.4	61.4±1.4	69.9±1.2	60.1±1.4	62.1±1.4
+ EVER (Kang et al., 2024)*	94.7±0.6	**96.5**±0.4	66.3±1.3	55.0±1.4	60.1±1.4	78.2±1.0	71.8±1.2	76.0±1.1
+ RAC (ours)	**94.7**±0.6	92.9±0.8	70.1±1.2	**60.3**±1.4	**63.6**±1.3	**79.1**±1.0	**73.2**±1.1	**77.6**±1.0
Llama2-7B-Chat	48.7±3.1	90.2±1.1	60.7±1.4	49.9±1.4	55.2±1.4	63.4±1.3	54.8±1.4	53.6±1.4
+ RAC (ours)	**79.8**±2.0	**91.5**±1.0	**77.8**±1.0	**84.5**±0.8	**76.5**±1.0	**77.0**±1.0	**80.2**±0.9	**72.0**±1.2
LLama3-8B-Instruct	51.1±3.1	91.0±1.0	60.7±1.4	49.9±1.4	55.2±1.4	61.1±1.4	52.1±1.4	53.4±1.4
+ RAC (ours)	**82.6**±1.8	**92.1**±0.9	**70.8**±1.2	**62.3**±1.4	**65.0**±1.3	**65.5**±1.3	**57.5**±1.4	**61.9**±1.4
Mistral-7B-Instruct	49.8±3.1	90.3±1.1	67.7±1.3	54.6±1.4	56.8±1.4	63.2±1.3	51.2±1.4	51.3±1.4
+ RAC (ours)	**80.0**±2.0	**91.2**±1.0	**67.9**±1.3	**59.0**±1.4	**62.3**±1.4	**65.0**±1.3	**53.6**±1.4	**55.2**±1.4

Table 3: Experimental results on Biography and TruthfulQA. BLEURT, BLEU, and ROUGE are accuracy scores (see §5). We report numbers with retrieval augmented generation (RAG) and without RAG. * indicates we reproduced a previous approach using the same retrieved documents and LLM as our approach for a fair comparison. The confidence interval is estimated by student t-distribution (Student, 1908). Aggregated across all settings, RAC performs similar or better than all previous methods with p-value ≤ 0.01.

have a lower performance than GPT-3.5-Turbo, indicating that these methods introduce new hallucinations when applied in the RAG setting. In contrast, our method improves upon GPT-3.5-Turbo even in the RAG setting. Across base LLM models, our method improves upon the baseline instruction tuned model by up to approximately 35% on BLEU accuracy, 18% on BLEURT accuracy, and 21% on ROUGE accuracy without RAG and up to 15% on BLEURT accuracy, 26% on BLEU accuracy and 20% on ROUGE accuracy with RAG. Surprisingly, our approach with Llama2-7B-Chat and LLama3-8B-Instruct without RAG is better with RAG, which indicates there are cases where using RAC without RAG is better than with RAG.

Results on LongFact We compare our method on 100 examples of the LongFact dataset using the GPT-4o model. The results are in Table 4. We find that RAC outperforms previous baseline methods RARR and Critic both with RAG and without RAG.

Results on Biography For the Biography dataset, our method performs similarly or better than previous methods across different LLMs with and without RAG, with the exception of our reimplementation of EVER. However, EVER is much slower and has larger token consumption than our method (see §7).

Approach	LongFact SAFE
without RAG	
GPT-4o	71.9 ± 11.5
*+RARR**	73.5 ± 11.3
*+Critic**	30.5 ± 30.3
+RAC	**73.6** ± 11.3
with RAG	
GPT-4o	82.9 ± 6.8
*+RARR**	77.2 ± 12.7
*+Critic**	50.8 ± 24.5
+RAC	**83.0** ± 7.0

Table 4: Experimental results on LongFact. * indicates we reproduced a previous approach using the same retrieved documents and LLM as our approach for a fair comparison. EVER (Kang et al., 2024) is too expensive to run for LongFact on GPT-4o.

Similar to TruthfulQA, we note the instruction-tuned model Llama2-7B-Chat is better than previous methods using Llama2-7B model, in both RAG and non-RAG settings. The previous approaches

Results	Percentage
R0 All stages are correct	90%
R1 Incorrect retrieval content	4%
R2 Incorrect atomic facts	1%
R3 Incorrect verification	2%
R4 Uncorrected errors	2%
R5 Newly introduced errors	1%

Table 5: Distribution of results of RAC

RARR and CRITIC improve performance slightly without RAG but have a degraded performance with RAG. In contrast, our method improves performance by up to 31% without RAG across three open-sourced models and up to 1.5% with RAG compared to strong instruction-tuned baselines. Although EVER is slightly better than our method with and without RAG setting, EVER's latency is much larger (see §7). Considering the baseline RAG performance is already over 90% in this dataset, our method still shows robust improvement with and without RAG settings across the range of LLMs, especially for open-sourced models.

Error Analysis To conduct an error analysis, we extracted 100 TruthfulQA examples from GPT-4o with RAG and RAC. We conducted a human analysis of the errors. The error analysis was a blind analysis performed by one of the authors. We focused on analyzing the types of errors shown in Table 5.[4] Table 5 shows the distribution of the results in the pipelines. Most of the results are correct. Among all the error responses, most of the time, the error is due to the retrieval being inaccurate. Other errors, such as incorrect verification and uncorrected errors during the correction stage, are insignificant. Notably, the newly introduced errors during correction and revision are surprisingly rare.

7 Ablation Experiments

Ablation of Verification Table 7 in the appendix shows full ablation results with or without verification, and with or without RAG. For LLMs without RAG, in most cases, performance drops significantly after adding verification, although performance is still better than the baseline. The reason for this is that without RAG, the original generated content has more content that needs to be corrected, and the verification step removes some critical corrections. For LLMs with RAG, the situation is different and verification improves performance. The reason is that RAG's performance is already very high, so if we correct all the statements, the correction process may introduce hallucinations which lowers the performance.

	Latency	Token Consumption
Generation	1 x	1x
RARR	70 x	21x
CRITIC	10 x	21x
CRITIC*	8.1 x	21x
EVER	150 x	13x
EVER*	98 x	13x
RAC (ours)	**3.9 x**	**3x**

Table 6: Experimentally measured latency and token consumption relative to uncorrected RAG generation for different methods on the Biography dataset. * indicates using the same retrieved documents and LLM as our approach for a fair comparison. EVER* corrects each sentence after generating all sentences of the output, rather than correction of each sentence during generation used in EVER.

To conclude, for models without RAG, correcting all statements is optimal, regardless of whether statements are true or false. In the RAG setting, adding a verification stage and correcting only false statements avoids introducing hallucinations during correction and revision.

Latency Table 6 shows the experimentally measured latency and token consumption on the Biography dataset for our method and previous approaches. Our method has reduced latency of 2x to 40x and reduced token consumption of 4x to 7x compared to previous approaches.

We describe the major sources of latency for each method. RARR generates a set of questions for each sentence in the output and then performs retrieval and reranking for each question, which introduces latency. CRITIC has several correction iterations, increasing the number of LLMs API calls and retrieval calls. EVER generates and corrects the output sentence by sentence, and for each sentence, retrieves using three different types of information; although the performance is slightly better than ours on the Biography dataset without RAG, the latency is the largest of all approaches and may be unacceptable for some applications. Note that the latency is evaluated under all possible parallel processing if they have for the previous approaches. In contrast to prior methods, our method retrieves once and corrects once, which reduces latency while remaining highly effective.

[4] Appendix (§H) presents a case study.

8 Conclusion

We introduce a simple but effective post-processing approach for improving factual correctness for instruction-tuned LLMs. Our method has improved latency over prior methods, does not involve additional training, and can be applied to settings with and without RAG. Experiments demonstrate that the proposed Retrieval Augmented Correction (RAC) approach can greatly reduce the correction latency while keeping a similar or better performance compared to previous post-correction methods.

Limitations

Our verification, correction, and revision prompts for each LLM are not highly optimized but can be tuned for the application. Our approach requires high-quality retrieval data, which may not be available or may require additional steps to acquire it. Like other post-correction methods, our method increased the latency compared to the original generation but is the best compared to similar work. Due to budget and hardware constraints, we were not able to experiment with our approach on larger open-sourced LLMs.

Acknowledgements

This work used resources available through the National Research Platform (NRP) at the University of California, San Diego. NRP has been developed, and is supported in part, by funding from National Science Foundation, from awards 1730158, 1540112, 1541349, 1826967, 2112167, 2100237, and 2120019, as well as additional funding from community partners.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *Preprint*, arXiv:2310.11511.

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

Akari Asai, Zexuan Zhong, Danqi Chen, Pang Wei Koh, Luke Zettlemoyer, Hannaneh Hajishirzi, and Wen tau Yih. 2024b. Reliable, adaptable, and attributable language models with retrieval. *Preprint*, arXiv:2403.03187.

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. *Preprint*, arXiv:2005.14165.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.

Souvik Das, Lifeng Jin, Linfeng Song, Haitao Mi, Baolin Peng, and Dong Yu. 2024. Entropy guided extrapolative decoding to improve factuality in large language models. *Preprint*, arXiv:2404.09338.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada. Association for Computational Linguistics.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning LLMs on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7765–7784, Miami, Florida, USA. Association for Computational Linguistics.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2024. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *Preprint*, arXiv:2002.08909.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *Preprint*, arXiv:2311.05232.

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

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).

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. 2023a. Mistral 7b. *Preprint*, arXiv:2310.06825.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Haoqiang Kang, Juntong Ni, and Huaxiu Yao. 2024. Ever: Mitigating hallucination in large language models through real-time verification and rectification. *Preprint*, arXiv:2311.09114.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. In *Advances in Neural Information Processing Systems*, volume 35, pages 34586–34599. Curran Associates, Inc.

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.

Dawei Li, Renliang Sun, Yue Huang, Ming Zhong, Bohan Jiang, Jiawei Han, Xiangliang Zhang, Wei Wang, and Huan Liu. 2025. Preference leakage: A contamination problem in llm-as-a-judge. *Preprint*, arXiv:2502.01534.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. The dawn after the dark: An empirical study on factuality hallucination in large language models. *arXiv preprint arXiv:2401.03205*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. *Preprint*, arXiv:2109.07958.

Hongyin Luo, Tianhua Zhang, Yung-Sung Chuang, Yuan Gong, Yoon Kim, Xixin Wu, Helen Meng, and James Glass. 2023. Search augmented instruction learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3717–3729, Singapore. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

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.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models.

In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *Preprint*, arXiv:2305.14251.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2024. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *Preprint*, arXiv:2305.15852.

Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. 2022. Disentqa: Disentangling parametric and contextual knowledge with counterfactual question answering. *Preprint*, arXiv:2211.05655.

OpenAI. 2022. large-scale generative pre-training model for conversation. *OpenAI blog*.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

OpenAI. 2024. Openai models.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *Preprint*, arXiv:2309.05922.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 31210–31227. PMLR.

Student. 1908. The probable error of a mean. *Biometrika*, pages 1–25.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2024. Fine-tuning language models for factuality. In *The Twelfth 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, 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, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ante Wang, Linfeng Song, Baolin Peng, Ye Tian, Lifeng Jin, Haitao Mi, Jinsong Su, and Dong Yu. 2024. Fine-grained self-endorsement improves factuality and reasoning. *Preprint*, arXiv:2402.15631.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Zixia Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V Le. 2024. Long-form factuality in large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Kevin Wu, Eric Wu, and James Zou. 2024. How faithful are rag models? quantifying the tug-of-war between rag and llms' internal prior. *Preprint*, arXiv:2404.10198.

Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. 2023. A critical evaluation of evaluations for long-form question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3225–3245, Toronto, Canada. Association for Computational Linguistics.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *Preprint*, arXiv:2401.15884.

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2023. Alignment for honesty. *Preprint*, arXiv:2312.07000.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. 2023a. Cognitive mirage: A review of hallucinations in large language models. *Preprint*, arXiv:2309.06794.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V Chawla, and Xiangliang Zhang. 2025. Justice or prejudice? quantifying biases in LLM-as-a-judge. In *The Thirteenth International Conference on Learning Representations*.

Seonghyeon Ye, Yongrae Jo, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, and Minjoon Seo. 2023b. Selfee: Iterative self-revising llm empowered by self-feedback generation. Blog post.

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*.

Xiaoying Zhang, Baolin Peng, Ye Tian, Jingyan Zhou, Lifeng Jin, Linfeng Song, Haitao Mi, and Helen Meng. 2024. Self-alignment for factuality: Mitigating hallucinations in llms via self-evaluation. *Preprint*, arXiv:2402.09267.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *Preprint*, arXiv:2309.01219.

A Retrieval & Post-Processing

We apply Google search for the retrieval process to obtain high-quality retrieval data. We then applied different post-processing strategies for different tasks since different tasks have different features.

For the biography task, we use the keyword "{Named Entity} Wikipedia" to search Google since the biography dataset is mainly from Wikipedia. After retrieving the top 10 results, we have two stages for the postprocessing; one is the filtering, and the other is truncating to fit the input length of LLMs. We first filter out the results that do not have the searched named entity, increasing the retrieval performance to a very good level. We then picked the first result from Wikipedia among the filtered results. Then, we truncate Wikipedia's useless sections for generating biographies, such as "References," "Footnotes," "Notes and references," "Notes," etc. After that, if the length is still too long, we remove some non-textual sections for generating biographies, such as the "Filmography" list, "Production" list, "Career statistics" table, etc. After removing them, most retrieved content is enough to fit the LLMs' context.

For the TruthfulQA task, we use their questions to search Google. Note that we found that Google search was contaminated with the dataset in this case since we can find the exact match of the answers in the dataset. Considering this, we retrieved the top 30 results and removed all of those data-leaking results by links such as "huggingface", "paperswithcode", "kaggle", "openreview", "github", "arxiv", etc to avoid cheating. We then only keep results that have longer than one sentence since some results are empty with just a hooked title. To fit for LLMs context length, in this case, we directly truncate all retrieved content to a fixed length since most of the related answers are on the very first sections of a retrieved page.

B Hyperparameters

For GPT-3.5-turbo, we use nucleus sampling with top_p = 0.3, meaning only the tokens comprising the top 30% probability mass are considered during generation. For Llama 2-Chat-7B or Llama 3-Chat-7B, we use their default setting. For different approaches, the hyperparameter settings for each LLM are the same.

C Full Ablation Results

Table 7 shows full results, including ablations of usage of verification stage on different settings.

D Effect of Retrieval Correctness

To analyze the effect of the retrieval correctness, we tested the performance using the gold data from the Biography dataset as the retrieved documents instead of our retrieving methods since this can ensure that the retrieval process is 100% accurate. We use GPT-3.5-turbo as the verification, correction, and revision model since it is the most robust model. The results are even promising compared to our sub-optimal retrieving accuracy. Table 8 in the appendix shows the results of this case. Without RAG, using gold data as the retrieval data for correction only improves the performance little. RAG using gold data has improved RAG a lot, and our approach can further enhance the RAG with gold retrieved data, achieving a performance of nearly 98%. This demonstrates that high-quality retrieved data is important to the success of our approach.

	Biography		**TruthfulQA**					
	Factscore		BLEURT	BLEU	ROUGE	BLEURT	BLEU	ROUGE
	Without RAG	**With RAG**	**Without RAG**			**With RAG**		
GPT-3.5-Turbo	78.9	93.4	58.3	48.7	51.5	67.2	62.2	65.0
+Self C+R	**88.2**	93.0	**61.9**	**50.7**	52.8	**72.6**	62.3	58.8
+Self V+C+R	88.0	**93.6**	59.7	49.2	**53.5**	70.4	**67.9**	**70.4**
Llama2-7B-Chat	48.7	90.2	60.7	49.9	55.2	63.4	54.8	53.6
+Self C+R	**79.8**	81.4	**77.8**	**84.5**	**76.5**	76.0	78.0	72.0
+Self V+C+R	50.4	90.7	67.1	73.8	67.7	**77.0**	**80.2**	**72.0**
+GPT C+R	77.2	90.2	71.1	61.6	64.5	70.3	60.8	64.0
+GPT V+C+R	70.1	**91.5**	62.2	53.7	59.5	68.7	58.5	61.9
LLama3-8B-Instruct	51.1	91.0	60.7	49.9	55.2	61.1	52.1	53.4
+Self C+R	76.7	89.4	60.2	51.2	55.2	57.9	49.1	52.9
+Self V+C+R	54.8	91.0	56.5	47.6	51.9	58.0	44.3	51.4
+GPT C+R	**82.6**	90.5	**70.8**	**62.3**	**65.0**	**67.7**	55.6	58.3
+GPT V+C+R	73.3	**92.1**	63.8	55.8	58.9	65.5	**57.5**	**61.9**
Mistral-7B-Instruct	49.8	90.3	67.7	54.6	56.8	63.2	51.2	51.3
+Self C+R	**80.0**	90.5	64.4	54.2	55.0	60.7	51.8	52.1
+Self V+C+R	53.0	90.8	64.4	51.4	53.4	60.5	50.1	52.4
+GPT C+R	72.2	89.4	**67.9**	**59.0**	**62.3**	**66.3**	**54.6**	**56.8**
+GPT V+C+R	68.2	**91.2**	67.8	55.0	60.2	65.0	53.6	55.2

Table 7: Ablation results for Biography and TruthfulQA. Self means the models of V, C and R are the same as the baseline models, GPT means the models of them are GPT-3.5-turbo when the baseline is not the GPT-3.5-turbo. BLEURT, BLEU, and ROUGE are accuracy scores (see §5).

Approach	Factscore
GPT3.5-Turbo	
C + R	88.2
C + R w/ Gold Retrieved Docs	88.4
RAG w/o filter	93.1
RAG	93.4
RAG w/o filter + V+C+R	93.5
RAG + V+C+R	93.6
RAG w/ Gold Retrieved Docs	97.6
RAG + V+C+R w/ Gold Retrieved Docs	**97.8**

Table 8: Results comparison of using and without using gold data for the Biography datatset as retrieval documents.

E Different LLMs Capabilities

Based on the analysis of the above results, we can infer the performance comparison of verification and correction with revision for selected LLMs in different RAG settings for each task. Table 9 shows model ability ranking for each component inferred from the results. Generally, Llama2-7B-chat has the best performance among all settings, while LLama3-8B-Instruct has the worst performance. While Llama series performance is not stable across the dataset (either Llama2-7B-chat or LLama3-8B-Instruct has been ranked third in one or more settings and components), the performance of GPT-3.5-turbo has not been ranked third, indicating that the closed-source model is more robust than the open-sourced model. The model ability is also task-related, i.e., Mistral-7B-instruct performs decently in the Biography dataset but poorly in the TruthfulQA.

F Keep All True (KAT) Ablation

Table 10 shows results comparison of using and without using KAT.

G Prompts

Table 11 shows prompts for each operation.

H Case Study

We analyze several examples manually to see the effect of our method. We find the baseline LLM often generates hallucinated content, which is factually incorrect. After applying our correction and revision on this setting without using RAG, all errors are corrected. However, there is still missing information. Using just RAG, the LLMs generate mostly factually correct answers, but there are still some factually incorrect texts. Only applying the correction and revision in this setting may introduce new factual errors since most statements are correct. However, after we add the verification process, correction and revision only get applied to the original texts with errors, which further improves

	Biograph	TruthfulQA
Without RAG		
Correction + Revision	Llama 2 >Mistral >GPT >Llama 3	Llama 2 >GPT >Llama 3=Mistral
With RAG		
Correction + Revision	GPT >Mistral >Llama 3 >Llama 2	Llama 2 >GPT >Llama 3=Mistral
Verification performance	GPT >Llama 2 >Mistral >Llama 3	Llama 2 >GPT >Llama 3=Mistral

Table 9: Model ability ranking table for each component inferred from the results. Llama 2 represents the Llama2-7B-Chat model, Llama 3 represents the LLama3-8B-Instruct model, and GPT represents GPT-3.5-turbo, Mistral represents the Mistral-7B-intruct.

	Biograph	TruthfulQA		
	Factscore	BLEURT	BLEU	ROUGE
GPT-3.5-turbo				
RAG + Self-(V+C+R)	93.6	70.4	67.9	70.4
RAG + Self-(V+C+R) + KAT	93.7	68.4	64.3	66.6
Llama2-7B-Chat				
RAG + Self-(V+C+R)	91.5	77.0	80.2	72.0
RAG + Self-(V+C+R)+KAT	90.6	67.2	62.5	59.6
LLama3-8B-Instruct				
RAG + Self-(V+C+R)	92.1	58.0	44.3	51.4
RAG + Self-(V+C+R)+KAT	91.5	59.0	49.1	52.9

Table 10: Keep All True results

Operation	Prompt
RAG	{Task Question}\n{Task Related Instructions}\n\"{retrieved documents}\"\n\nAnswer:\n
Atomic Fact Extraction	Please breakdown the following content into independent facts without pronouns(Do not use He, She, It...)(each fact should be a full sentence, each fact per line):"origiinal model generation"\nFacts:\n
Verification	{Task Question}\npassage:"{retrieved documents}\"\nPlease verify the below statements to the above question into true or false or not mentioned based on the above passages (one answer per line with label true or false or not mentioned.)\nTrue means the similar statement can be found in the above passage and have the same meaning.\nFalse means the similar statement can be found in the above passage but have the different meaning.\nNot Mentioned means the similar statement cannot be found in the above passage.\n\nStatements:"{extracted atomic facts}"\n\nOutput Format:\nStatement 1: True\nStatement 2: False \n ... \nStatement N: Not Mentioned\n\nAnswer(start with the output directly without additional comments):\n")
Correction for all	{Task Question}\npassage:"{retrieved documents}"\nCorrect the following statement and output the corrected version based on the above passage. If the statement is correct, directly output the original statement. In your answer, start with the corrected answer or original correct statement directly without repeating the question. The answer should be a single sentence and should be concise and to the point of the question. \n\nStatement:"{extracted atomic facts}"\n\nAnswer:\n
Correction for False	{Task Question}\npassage:"{retrieved documents}"\nCorrect the following statement and output the corrected version based on the above passage. In your answer, start with the corrected answer directly without repeating the question or the original statement. \n\nStatement:\"{a false statement after RAV}\"\n\nAnswer:\n"
Revision	{Task Question and Original Generated Answer}\n\nPlease correct the above answer into a corrected one based on the following verified facts. In your answer, start with the corrected answer directly without repeating the question or the original answer.\n{Task Related Instructions}\nVerified facts:"{statements after correction}"\n\nCorrected answer:\n

Table 11: Prompts for each operation

the performance without introducing new factual errors for the original statements that have already been verified as truths. In a manual comparison to RARR (Gao et al., 2023), we find RARR consistently misses information and introduces more hallucinations, while our method retains facts of the original LLM output and rarely introduces more hallucinations. Table 12 gives an example of generation output.

I Example Output

Table 12 shows the outputs for each stage for an example from the Biography dataset.

Component	Task: Biography
Llama2-7B-Chat	Sara Paxton is an American actress, singer**, and model. She was born on November 25, 1988, in Woodridge, New Jersey,** USA. Her parents are Lucille and Steve Paxton. **She has one older brother, Sean Paxton.** Sara began her acting career at a young age, appearing in various TV commercials and films. **Her breakthrough role came in 2003 when she played the lead role in the Disney Channel original movie "Summerland."** Her performance earned her a Daytime Emmy Award nomination for Outstanding Young Actress in a Drama Series. **In 2005, Sara landed a recurring role on the hit TV series "Dirt," playing the character of Riley Richmond.** She also appeared in the films "Aquamarine" and "The Last House on the Left." **In 2009, Sara starred in the horror movie "The Last Exorcism," which was a box office success. She also released her debut album, "Sara Paxton," which included the hit single "Birthday."** Sara has continued to act and sing in various projects, **including the TV series "Houdini & Doyle" and the film "The Circle." She has also been a guest judge on the reality TV show "American Idol."** Sara is known for her versatility and range as an actress and singer. She has been praised for her performances in various genres, from comedy to drama to horror. **She is also recognized for her philanthropic work, particularly in the area of animal welfare.**
RARR (Gao et al., 2023)	Sara Paxton is an American actress and singer, born on April 25, 1988, in Woodland Hills, California. She began her acting career at a young age, appearing in various television shows and commercials. Paxton gained recognition for her role as Mari Collingwood in the horror film "The Last House on the Left" (2009) and **as Amanda in the teen comedy film "Aquamarine" (2006)**. In addition to her acting career, Paxton is also a talented singer and has released several singles and music videos. She has also lent her voice to various animated projects, **including the role of Marnie in the "Happily N'Ever After" film series**. Paxton continues to work in film and television, with recent credits including roles in the TV series "Good Girls" and the film "The Front Runner" (2018). She is known for her versatility as an actress and her ability to portray a wide range of characters across different genres.
GPT (C+R)	Sara Paxton is an American actress and singer. She was born in Woodland Hills, Los Angeles, California, USA. Her parents are Lucia and Steve Paxton. Sara is the only child of her parents and does not have an older brother named Sean Paxton. Sara began her acting career at an early age, appearing in minor roles in both films and television shows before rising to fame in Early October 2002. Her breakthrough role came in 2004 when she played the role of Sarah Borden in the television series "Summerland." She received an Emmy nomination for Outstanding Young Actress in a Drama Series for her performance in the television series "Darcy's Wild Life." Sara appeared in the films "Aquamarine" and "The Last House on the Left." She did not release a debut album titled "Sara Paxton" with the hit single "Birthday." She has not appeared in the TV series "Houdini & Doyle" or the film "The Circle." Sara is known for her versatility and range as an actress and singer. She has been praised for her performances in various genres, from comedy to drama to horror. Sara is married to Zach Cregger and resides in Los Angeles, California. Her advocacy work and involvement with animal rights organizations to promote animal welfare are not mentioned in the passage. In her free time, she enjoys singing, writing music, and spending time with her pets. Sara received an Emmy nomination for her role in the Discovery Kids television series, Darcy's Wild Life, in 2006, not in 2003. Her breakthrough role was in the television series "Summerland" in 2004, not in the Disney Channel original movie "Summerland" in 2003. Sara did not have a starring role in the horror movie "The Last Exorcism" in 2009. She did not release her debut album in 2009.
RAG	Sara Paxton is an American actress and singer **who was born on November 25, 1988**, in Woodland Hills, Los Angeles, California. She began her acting career at a young age, appearing in minor roles in both films and television shows before rising to fame in 2002. Paxton has starred in numerous films and television series, including "Aquamarine," "Return to Halloweentown," "Sydney White," "Superhero Movie," "The Last House on the Left," and "The Innkeepers." She has also provided backing vocals on her co-star Drake Bell's theme song for the movie "Superhero!" Song. Paxton was raised in the San Fernando Valley and graduated from El Camino Real High School in 2006. She did not attend college, choosing instead to pursue her acting career. Paxton has been nominated for several awards, including an Emmy nomination for her role in the Discovery Kids television series "Darcy's Wild Life." In her personal life, Paxton married Zach Cregger in October 2019.
RAG Atomic Facts	Sara Paxton is an American actress and singer. **She was born on November 25, 1988,** in Woodland Hills, Los Angeles, California. Paxton began her acting career at a young age. She appeared in minor roles in both films and television shows before rising to fame in 2002. Paxton has starred in numerous films and television series. Some of her notable works include "Aquamarine," "Return to Halloweentown," and "The Last House on the Left." She also provided backing vocals on her co-star Drake Bell's theme song for the movie "Superhero!" Paxton was raised in the San Fernando Valley. She graduated from El Camino Real High School in 2006. Paxton did not attend college, choosing instead to pursue her acting career. She has been nominated for several awards, including an Emmy nomination for her role in the Discovery Kids television series "Darcy's Wild Life." In her personal life, Paxton married Zach Cregger in October 2019.
RAG verified results	Statement 1: True \n **Statement 2: False** \n Statement 3: True\n Statement 4: True\n Statement 5: True\n Statement 6: True\n Statement 7: True\n Statement 8: True\n Statement 9: True\n Statement 10: True\n Statement 11: True\n Statement 12: True
RAG + GPT-(V+C+R)	Sara Paxton is an American actress and singer. She was born in Woodland Hills, Los Angeles, California. Paxton began her acting career at a young age, appearing in minor roles in both films and television shows before rising to fame in 2002. She has starred in numerous films and television series, including "Aquamarine," "Return to Halloweentown," and "The Last House on the Left." Paxton has also provided backing vocals on her co-star Drake Bell's theme song for the movie "Superhero!" She was raised in the San Fernando Valley and graduated from El Camino Real High School in 2006. Paxton did not attend college, choosing instead to pursue her acting career. She has been nominated for several awards, including an Emmy nomination for her role in the Discovery Kids television series "Darcy's Wild Life." In her personal life, Paxton married Zach Cregger in October 2019.

Table 12: An example generation output flow from biography using different operations. **Bold texts** represent incorrect facts. No **Bold texts** means no factual error. Outputs are formatted to fit the table without changing the original content.