Corrective Retrieval Augmented Generation

Anonymous ACL submission

Abstract

Large language models (LLMs) inevitably exhibit hallucinations since the accuracy of generated texts cannot be secured solely by the parametric knowledge they encapsulate. Although retrieval-augmented generation (RAG) is a practicable complement to LLMs, it relies heavily on the relevance of retrieved documents, raising concerns about how the model behaves if retrieval goes wrong. To this end, we propose **C**orrective **R**etrieval **A**ugmented **G**eneration (CRAG), an autonomous agent that dynamically corrects and refines the retrieval content. Specifically, an agent core is introduced that perceives, decides, and executes in a closed loop. It dynamically selects and orchestrates tools, including internal retrieval, web search, keyword evaluators, semantic evaluators, and LLM evaluators, to perform multi-dimensional contextual quality evaluation and knowledge supplementation, thereby constructing accurate and precise retrieval content. Furthermore, because CRAG refines and corrects retrieved context without requiring modifications to the underlying system, it naturally serves as a plug-and-play framework that can be seamlessly integrated with a wide range of RAG-based approaches. Experiments on five datasets demonstrate that CRAG outperforms previous approaches, while its autonomous decision-making capabilities improve tool-calling efficiency and retrieval context quality.

1 Introduction

Large language models (LLMs) have attracted increasing attention and exhibited impressive abilities to understand instructions and generate fluent language texts (Brown et al., 2020; Ouyang et al., 2022; Touvron et al., 2023). Nevertheless, LLMs inevitably manifest hallucinations (Ji et al., 2023) due to their struggle with factual errors (Mallen et al., 2023; Min et al., 2023) and inability to secure the accuracy of generated texts solely by the parametric knowledge they encapsulate (Zhang et al., 2023b; Muhlgay et al., 2023).

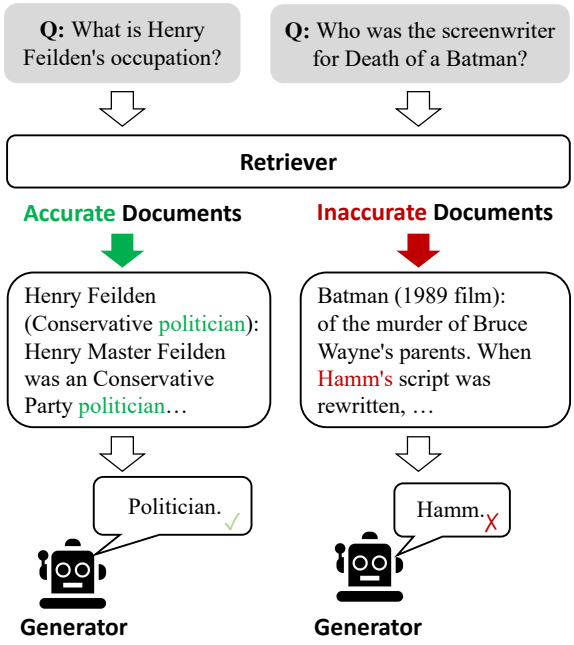

Figure 1: The examples show that a low-quality retriever is prone to introducing a substantial amount of irrelevant information, impeding the generators from acquiring accurate knowledge and potentially misleading them.

Prior research has introduced the retrieval techniques to incorporate the knowledge relevant to input and augment generation, as exemplified by retrieval-augmented generation (RAG) (Lewis et al., 2020). In this framework, the input to models is augmented by prepending relevant documents that are retrieved from an external knowledge corpus (Guu et al., 2020). While RAG serves as a practicable complement to LLMs, its effectiveness is contingent upon the relevance and accuracy of the retrieved documents (Li et al., 2022; Tan et al., 2022). The heavy reliance of generation on the retrieved knowledge raises significant concerns about the model's behavior and performance in scenarios where retrieval may fail or return inaccurate results (Shi et al., 2023). Although recent works adopted an agent mechanism to enhance the robustness and attempt to overcome the over-reliance issue, the bottleneck of the retrieval quality

still remains unsolved. As Figure 1 shows that a low-quality retriever is prone to introducing a substantial amount of irrelevant information, impeding the models from acquiring accurate knowledge and potentially misleading them, resulting in issues such as hallucinations (Zhang et al., 2023b). However, most conventional RAG approaches indiscriminately incorporate the retrieved documents, regardless of whether these documents are relevant or not (Rony et al., 2022). Furthermore, current methods mostly treat complete documents as reference knowledge both during retrieval and utilization. But a considerable portion of the text within these retrieved documents is often non-essential for generation, which should not have been equally referred to and involved in RAG.

On account of the above issues, this paper particularly studies the scenarios where the retriever returns inaccurate results. A method named **C**orrective **R**etrieval-**A**ugmented **G**eneration (CRAG) is proposed to self-correct the retrieval results and improve the utilization of documents for augmenting generation. An autonomous agent is designed to dynamically perceive retrieved contexts and iteratively decide and execute appropriate tools in a closed loop for retrieval context correction. To address the perspective of the corrective mechanism, two main categories of tools are introduced for *context evaluation* and *knowledge supplementation*, respectively. For *context evaluation*, three evaluators are adopted, including a keyword evaluator, a semantic evaluator, and an LLM evaluator, to provide a multi-dimensional evaluation at the word-, sentence-, to document-level. Since the evaluation results directly influence the subsequent correction process, this flexible evaluation mechanism greatly enhances the reliability.

Based on the evaluation score, the agent core further decides whether additional knowledge is required and how to obtain it. For *knowledge supplementation*, two tools are employed: an additional retriever and large-scale web search. First, an additional retriever recovers potentially missing relevant documents, particularly for multi-hop reasoning tasks. Sencond, large-scale web searches (Piktus et al., 2021; Komeili et al., 2022) are integrated as a strategic extension, since retrieval from static and limited corpora can only return sub-optimal documents in terms of scope and diversity. This augmentation is implemented to broaden the spectrum of retrieved information, harnessing the expansive and dynamic nature of the web to complement and enrich the initially obtained documents. After obtaining the supplied knowledge, the agent core automatically removes the incorrect and unrelated content, iteratively deciding whether the context requires further correction. This process refines the retrieved information by retaining key insights while filtering out irrelevant content.

CRAG is plug-and-play and experimentally implemented into various RAG-based approaches, demonstrating its adaptability. Results on five diverse datasets of PopQA (Mallen et al., 2023), PubHealth (Zhang et al., 2023a), Arc-Challenge (Bhakthavatsalam et al., 2021), HotpotQA (Yang et al., 2018), and Musique (Trivedi et al., 2022) show that CRAG can significantly improve the performance of standard RAG and state-of-the-art RAG approaches, demonstrating its generalizability across both single- and multi-hop generation tasks with different complexity. Further analysis show that CRAG consistently achieves substantially higher tool-use efficiency, demonstrating the effectiveness of its self-correction mechanism for RAG. To facilitate others in reproducing our results, we will publish all source code later.

In summary, our contributions in this paper are three-fold: 1) This paper studies the scenarios where the retriever returns inaccurate results, and a corrective strategy is designed for RAG to improve its robustness. 2) A plug-and-play method named CRAG is proposed to improve the ability of automatic self-correction and efficient utilization of retrieved documents. 3) Experimental results extensively demonstrate CRAG's adaptability to RAG-based approaches and its generalizability across various generation tasks.

2 Related Work

Hallucinations of LLMs Although LLMs have exhibited impressive abilities to understand instructions and generate fluent language texts (Bang et al., 2023; Qin et al., 2023; Zhong et al., 2023), one of the most severe issues that LLMs have still been struggling with is hallucinations. As many studies found (Tonmoy et al., 2024; Zhang et al., 2023b; Shuster et al., 2021), either outdated information or incorrect knowledge that is activated would seriously result in hallucinations. Large-scale unregulated training data collection, low proportion

of high-quality sampling data, imperfection of data allocation in the input space, and many other realistic factors could impact the LLMs and exacerbate the problems. Thus, it is obvious that the lack of accurate and specific knowledge can lead to misleading or even inaccurate generation, which will severely hurt the experience of users in most practical applications.

Retrieval-Augmented Generation RAG (Lewis et al., 2020; Guu et al., 2020) is regarded as a useful method to address the issues above, which enhances the input questions of generative LMs with retrieved documents. The proposed methods generally leverage information retrieval to supply documents with relevant knowledge for generative LLMs. Despite this, conventional RAG approaches usually ignore a question, *what if the retrieval goes wrong?* Since the purpose of introducing a retrieval is to secure that generative LMs can obtain relevant and accurate knowledge. If retrieved documents are irrelevant, the retrieval can even exacerbate the factual error that LMs make.

Advanced RAG Many advanced approaches have been developed from the original RAG in recent years (Zhang et al., 2024; Kim et al., 2024; Wang et al., 2024; Liu et al., 2024). Considering that retrieval is sometimes unnecessary for some queries, conversely, responses without retrieval are even more accurate in many situations. Self-RAG (Asai et al., 2024) is proposed to selectively retrieve knowledge and introduce a critic model to decide whether to retrieve. In addition, in some long-text generation tasks, external knowledge is needed more than once, and when to retrieve should be concerned. Jiang et al. (2023) actively anticipate future content and decide when and what to retrieve in long-form generation.

Agentic RAG Agentic retrieval-augmented generation (RAG) has recently emerged as a prominent research direction. By incorporating agent planning, tool utilization, memory, and reflection mechanisms, this paradigm shifts the retrieval process from passive data fetching to active reasoning. Agentic RAG offers distinct advantages over traditional frameworks. It supports multi-step reasoning and dynamic task planning, which enables the system to autonomously determine whether to perform additional retrieval, adjust query strategies, or switch data sources based on intermediate outputs. The integration of external tools and memory modules further facilitates closed-loop processing for complex tasks and maintains persistent context states. Representative implementations illustrate these mechanisms: ReAct (Yao et al., 2023) integrates retrieval with tool invocation and memory to support multi-turn interactions, state maintenance, and closed-loop planning. Adaptive-RAG (Jeong et al., 2024) employs a query complexity classifier to dynamically route retrieval strategies by adaptively selecting zero, single-step, or multi-step iterative retrieval paths according to query difficulty. A-RAG (Du et al., 2026) delegates tool selection and task planning entirely to the agent to enable highly flexible retrieval patterns. Collectively, Agentic RAG transcends the static limitations of conventional pipelines by reconstructing retrieval as an interactive and plannable agent workflow.

Compared with recent studies (Schick et al., 2023; Luo et al., 2023; Asai et al., 2024) that are the most relevant to our work, a main difference should be highlighted. These approaches target on exploiting retrieval as a useful tool to augment generation or whether retrieval is necessary, while this study particularly studies the scenarios where the retriever returns inaccurate results. To the best of our knowledge, this paper makes the first attempt to explore and design corrective strategies for RAG to improve its robustness of generation.

3 Task Formulation

Given an input $\mathcal{X}$ and an accessible corpus containing a large amount of knowledge documents $\mathcal{C} = \{d_1, ..., d_N\}$, the system is expected to generate an output $\mathcal{Y}$. The retriever aims to retrieve the top-k documents $\mathcal{D} = \{d_{r_1}, ..., d_{r_k}\}$ that are relevant to the input $\mathcal{X}$ from the corpus $\mathcal{C}$. Additionally, the agent has access to a tool registry $\mathcal{T} = \{t_1, ..., t_N\}$, where each tool t_i is described by structured metadata, including functionality description, input-output schema, and applicability conditions. The agentic RAG with n-turn execution loop can be formulated as a Markov Decision Process:

$$P(\mathcal{Y}|\mathcal{X}) = P(\mathcal{Y}|\mathcal{X}, \mathcal{D}, \mathcal{K}_n) \cdot \sum_{i=1}^{n} P(\tau|\mathcal{X}, \mathcal{D}, \mathcal{K}_{1:i-1}). \quad (1)$$

where τ represents a trajectory of tool invocations $(t_1, t_2, ..., t_n)$, K_i is the returned results of i tools.

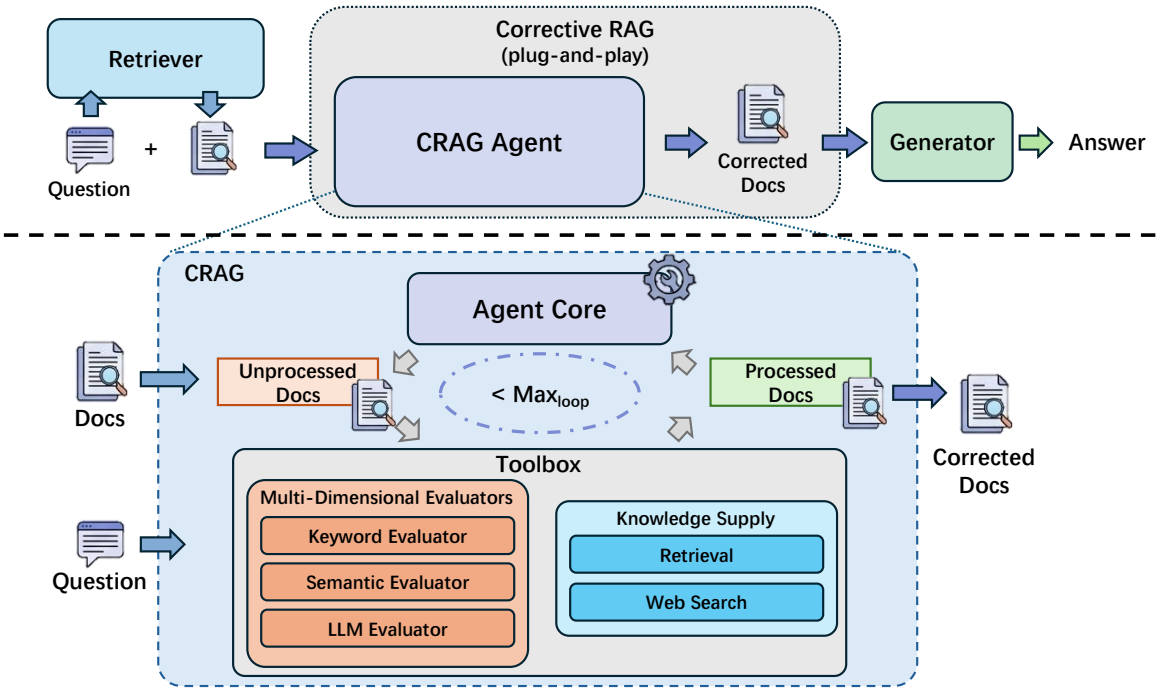

Figure 2: An overview of the proposed CRAG at inference. An autonomous agent is proposed to dynamically evaluate and search for additional knowledge for retrieval context correction. Five tools are included and invocated iteratively, obtaining a precise and accurate context for answer generation.

4 CRAG

4.1 Overview of Model Inference

Figure 2 and Algorithm 1 present an overview of CRAG at inference, which designs corrective strategies to improve the robustness of generation. The retrieved content and the input question are sent to the agent for subsequent corrective processing. The architecture of the agent comprises three components: (1) Agent core. The cognitive center of the system, powered by an agentic model with autonomous decision-making and tool invocation. (2) Multi-dimensional evaluator tools. Three evaluators are adopted for supporting effective content quality evaluation. (3) Web search tool. Large-scale web search is included as an extension to support the knowledge where retrieval correcting is required. After the multi-turn agent, the corrected content is then sent to a generator for answer generation. Since only the retrieval content is modified throughout the system, CRAG remains plug-and-play and can be seamlessly integrated into various RAG-based approaches, while adding autonomous adaptability through its agentic architecture.

4.2 Agent Core

The agent core is the cognitive center of CRAG. While the original evaluator assessed retrieval quality and triggered fixed actions, the controller serves as an autonomous decision-making module that perceives state, reasons about strategies, and generates structured tool invocation plans.

The controller receives as input the current working memory state S, which encompasses: (1) Query representation: original query and its potential rewritten variants. (2) Document history: all retrieved document sets before and after correction with their metadata and quality vectors assessed by previous evaluators. (3) Tool execution history: sequence of invoked tools with parameters, outcomes, latency, and resource consumption. (4) Current knowledge corpus: accumulated internal and external knowledge with source attribution and confidence scores.

This rich state representation enables context-aware decision-making that considers not only current retrieval quality but also historical tool effectiveness, resource constraints, and query-specific characteristics.

4.3 Content Evaluator

It is natural to wonder whether the retrieved documents are accurate or not before using them, which is significant since irrelevant or misleading messages can be identified in this way. The accuracy of the retrieval evaluator undeniably

Algorithm 1: CRAG Inference

Tools : $E_{keyword}$ (Keyword Evaluator), $E_{semantic}$ (Semantic Evaluator), E_{LLM} (LLM Evaluator), W (Web search), R (Retriever)

Require : $T = \{E_{keyword}, E_{semantic}, E_{LLM}, W, R\}$ (Tools), $Agent$, G (Generator)

Input : x (Input question), $D = \{d_1, d_2, ..., d_k\}$ (Retrieved documents), $C_0 = D$ (Corrected Context)

Output : y (Generated response)

```
// Two conditions to quit the agent loop: reach the MAX_TURN or no tool is
   called.
for i = 1 to MAX_TURN do
    T_selected = Agent(x, C_{i-1})
    if T_selected ≠ φ then
        Context = [T.execute() for T ∈ T_selected]
        C_i = Agent(x, C_{i-1}, Context)
    else
        break;
    end
end
y = G.predict(x, C_i)
```

plays a pivotal role in shaping the overall system performance, as it influences overall system performance and directly determines the subsequent decisions and actions of the agent. Prior studies frequently incorporate mechanisms for self-awareness, self-critique, or performance evaluation. Recent mainstream approaches predominantly adopt the LLM-as-a-judge paradigm or directly activate the latent evaluation capabilities of agents. We argue that relying exclusively on LLMs for assessment presents inherent limitations. LLMs typically evaluate quality by processing the complete retrieved text, which captures only approximate global accuracy and fails to satisfy fine-grained requirements. Furthermore, the black-box nature of LLM judgments produces scores that lack interpretability and controllability. To address these issues, we integrate multiple evaluation tools, including an LLM evaluator, a semantic evaluator, and a keyword evaluator. This design provides document-, sentence-, and word-level evaluations respectively. The multi-dimensional framework substantially enhances the evaluation capability of the agent, delivers more reliable guidance for downstream decision making, and mitigates the risk of hallucinations during agent reasoning.

Keyword evaluator We employ keyword evaluation to provide a fine-grained evaluation. Precise concept matching is essential for many queries, particularly those involving proper nouns, ambiguous entities, specific dates, and numerical values, where relying exclusively on semantic similarity or LLM-based reading frequently leads to misjudgments. Through exact keyword matching, the agent rapidly determines whether the retrieved text contains the required terms. By invoking the keyword evaluator, the agent obtains the keyword coverage ratio (KCR) and identifies missing keywords for each segment.

Semantic evaluator This evaluation approach is widely adopted in recent years. It employs a lightweight cross-encoder that processes the candidate text and the query to output their semantic similarity. The method delivers strong overall performance, combining high versatility, robustness, and rapid execution. It maintains reliable accuracy across most scenarios and excels at the sentence level, where it effectively compares the semantics of sentence pairs or claims. Invoking the semantic evaluator, the agent obtains the similarity score between the candidate text and the query.

LLM evaluator The LLM-as-a-judge paradigm represents a prominent research area and receives extensive study. The evaluation capabilities of these models are widely validated. In this work, the LLM evaluator is primarily invoked to process large-scale documents. By invoking the LLM evaluator, the agent obtains scores that quantify both the relevance of the retrieved text to the input query and its intrinsic quality.

4.4 Web Search

It would be more intelligent if a system itself could determine that its existing knowledge corpus could not solve the problem well and turn to additional external knowledge for help. On the contrary, even if a system knows that the existing knowledge cannot solve the problem, but still sticks to the limited knowledge corpus, it will only give a fabricated fact in the end, which is called hallucination. Therefore, it is extremely important to seek complementary external knowledge if the retrieved results are all assumed irrelevant, and we consider a system that knows what it doesn't know and what it cannot answer to be more intelligent than one that clings to limited knowledge and is incapable of seeking external knowledge. Since retrieval from static and limited corpora can only return sub-optimal documents in terms of scope and diversity, large-scale web searches (Piktus et al., 2021; Komeili et al., 2022) are integrated as a strategic extension of RAG. In CRAG, a public and accessible commercial web search API is adopted to generate a series of URL links for every query. [1] Considering that knowledge from large-scale web searches could introduce biases or unreliable information, authoritative and regulated web pages like Wikipedia are preferred, which can significantly help mitigate these issues. Moreover, we utilize the URL links to navigate web pages, transcribe their content, and provide to the agent core to derive the relevant web knowledge.

5 Experiments

We conducted experiments to extensively demonstrate CRAG's adaptability to RAG-based approaches and its generalizability across both short- and long-form generation tasks.

5.1 Tasks, Datasets and Metrics

CRAG was evaluated on four datasets, including **PopQA** (Mallen et al., 2023) (*short*-form generation), **PubHealth** (Zhang et al., 2023a) (*true-or-false* question), and **Arc-Challenge** (Bhakthavatsalam et al., 2021) (*multiple-choice* question). **HotpotQA** (Yang et al., 2018) (*multi-hop* question). **Musique** (Trivedi et al., 2022) (*multi-hop* question). Following previous work, accuracy was adopted as the evaluation metric. Readers can refer to Appendix C.1 for more details.

[1] In this study, Google Search API is utilized for searching.

5.2 Baselines

We primarily compared CRAG with both approaches with and without retrieval, where the latter can be further split into standard RAG and latest advanced RAG, including:

- **Baselines without retrieval.** We evaluated the direct performance of the utilized public LLMs, including GPT5-mini (OpenAI, 2026) and DeepSeek-V4-pro (DeepSeek-AI, 2026).

- **Naive RAG.** We evaluated the standard RAG (Lewis et al., 2020) where an LM generates output given the query prepended with the top retrieved documents using the default retriever as in our system.

- **Advanced RAG.** (1) Linear-RAG (Zhuang et al., 2025). A linear-scalable GraphRAG paradigm that eliminates costly relation extraction by constructing a relation-free Tri-Graph (entity-sentence-passage) with lightweight semantic linking, enabling efficient two-stage retrieval through local semantic bridging and global importance aggregation without any LLM token consumption during indexing. (2) HippoRAG2 (Gutiérrez et al., 2025). A RAG framework that mimics human hippocampal memory mechanisms to create associative knowledge links, achieving non-parametric continual learning from RAG. (3) A-RAG (Du et al., 2026). An adaptive retrieval-augmented generation framework that dynamically tailors retrieval strategies based on query complexity, ranging from direct generation for straightforward queries to multi-step reasoning for complex questions, balancing accuracy and efficiency.

5.3 Experimental Settings

Retrieval Settings Following prior work (Asai et al., 2024), we employ Contriever to retrieve documents from the full Wikipedia corpus for PopQA, Arc Challenge, and PubHealth. As for HotpotQA and Musique, we adhere to conventional approaches (Du et al., 2026; Gutierrez et al., 2024) as well by constructing a retrieval corpus from the subset of documents associated with all questions in the test set. These distinct retrieval settings lead to varying levels of retrieval quality. The vast scale of the Wikipedia corpus often limits the ability to locate sufficiently relevant information within

Method	PopQA (Accuracy)	ARC (Accuracy)	PubHealth (Accuracy)	HotpotQA (LLM Accuracy)	HotpotQA (Accuracy)	Musique (LLM Accuracy)	Musique (Accuracy)
				GPT5-mini			
Vanilla	42.4	95.3	33.8	56.6	56.3	33.0	25.4
Naive RAG	46.9	95.6	32.7	56.1	59.3	32.7	26.3
LinearRAG	49.8	96.0	52.0	72.0	60.5	34.8	26.3
HippoRAG2	43.5	**96.5**	54.2	**80.7**	69.7	40.6	38.4
CRAG (Ours)	**73.4**	94.1	**84.0**	77.6	**84.3**	**55.2**	**54.3**
				DeepSeek-V4-pro			
Vanilla	52.6	94.1	39.8	66.0	63.6	42.4	34.8
Naive RAG	53.3	93.9	37.6	65.9	63.4	41.4	35.7
A-RAG	44.2	72.2	51.7	69.7	74.5	**74.5**	**62.0**
CRAG (Ours)	**72.9**	**97.0**	**83.5**	**74.4**	**85.2**	57.8	61.2

Table 1: Overall evaluation results on the test sets of five datasets. Results are separated based on the generation LLMs. **Bold** numbers indicate the best performance among all methods for each LLM. CRAG is particularly effective in challenging retrieval environments (PopQA and PubHealth), showing effectiveness in content correction.

a constrained index, resulting in relatively lower-quality retrieved text for PopQA, Arc Challenge, and PubHealth. In contrast, the smaller, curated corpora for HotpotQA and Musique facilitate higher-quality retrieval, thereby yielding greater benefits from the retrieval process. Consequently, we incorporate the retrieval tool specifically for these two datasets to leverage the high potential of multi-hop reasoning.

Agent Settings Experimental results in agent-based systems typically improve with an increasing number of iteration rounds. To ensure a fair and comparable evaluation between CRAG and baseline methods, we cap the maximum number of iterations at five for all agent designs. This constraint balances the efficiency and capability of RAG methods while establishing a consistent basis for comparison across different approaches.

5.4 Results

Table 1 presents the results on five datasets. We can draw the following conclusions.

First, CRAG Outperforms Prior Methods Across Most Benchmarks. On the GPT5-mini backbone, CRAG achieves the highest accuracy on PopQA (73.4%), PubHealth (84.0%), HotpotQA (84.3%), and Musique (54.3%), substantially surpassing HippoRAG2 and LinearRAG. Specifically, on PubHealth, CRAG outperforms HippoRAG2 by 29.8%, LinearRAG by 32.0%, and on Musique, CRAG exceeds HippoRAG2 by 15.9%, LinearRAG by 28.0%. While on PopQA, CRAG outperforms HippoRAG2 by 29.9%, LinearRAG by 23.6%, and on HotpotQA, CRAG outperforms HippoRAG2 by 14.6%, LinearRAG by 23.8%. Additionally, on the DeepSeek-V4-pro backbone, CRAG consistently achieves the highest accuracy on PopQA (72.9%), Arc (97.0%), PubHealth (83.5%), and HotpotQA (85.2%). Specifically, on PopQA, CRAG outperforms A-RAG by 28.7%, 31.8% on PubHealth and 10.7% on HotpotQA.

Second, CRAG achieves greater improvement in more challenging retrieval environments. Comparing CRAG with previous RAG approaches, including Naive RAG and other baselines, we observe that our method yields more substantial gains on PopQA and PubHealth. On these two benchmarks, retrieval quality is relatively low. Following prior work, we retrieve relevant documents from the full Wikipedia corpus using a retriever. In contrast, for HotpotQA and Musique, previous methods also follow prior work and retrieve from the sub-corpus used to construct the benchmarks, which has a smaller retrieval scope and provides higher quality retrieval content. Experimental results indicate that CRAG is particularly effective under poor retrieval quality, consistent with our goal of enabling content correction in low-quality retrieval scenarios.

5.5 Ablation Study

Since CRAG tightly couples several tools, removing either web search or the retrieval evaluator would render the approach meaningless. We therefore conduct ablation studies primarily on the design of the system framework. Since prior work already explored retrieval via web search, we focus on analyzing the specific improvement brought by CRAG within an agent framework. To this end, we examine three variants of CRAG: one without

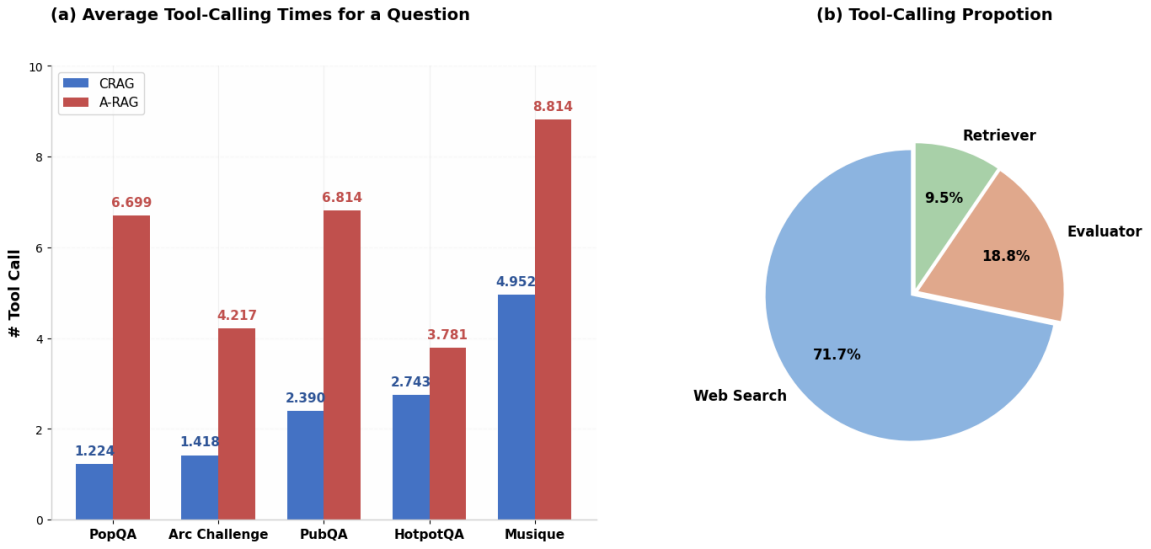

Figure 3: The statistics of the tool calling. The left figure (a) indicates Average tool-calling numbers for a question of CRAG and A-RAG across different benchmarks under the same maximum turn setting. While the right figure (b) indicates the proportion of the tools that were utilized in all benchmarks.

	PopQA	PubHealth
DeepSeek-V4-pro		
CRAG	**72.9**	**83.5**
w/o strong generator	65.5	52.8
w/o generator	52.2	53.8
w/o loop	58.7	76.9

Table 2: Ablation study of CRAG. CRAG w/o strong generator indicates the replacement of the powerful generator (DeepSeek-V4-pro) with a finetuned LLaMA-2-7B to amplify the impact of retrieval quality on the response. While w/o generator takes a further step by directly using the corrected results as the answer. Finally, w/o loop, e.g. removing loop, means discarding iterative tool calls and implementing CRAG through a workflow framework.

a strong generator (i.e., using a weaker generative model), one without any generator (answering solely with the corrected text), and one without the agent-like iterative tool use (a workflow that first evaluates then decides whether to invoke web search). These experiments isolate the impact of removing a strong generator on the question answering task and highlight the importance of CRAG's iterative information refinement.

5.6 Efficiency Analysis

In this paper, we further investigate the efficiency of CRAG and compare it with previous state-of-the-art methods. Since different tools in an agent require varying numbers of tokens and execution times, we adopt the average number of tool calls per question as a metric to measure the efficiency of agentic RAG approaches. Comparing with the state-of-the-art method A-RAG, the figure presents the efficiency comparison. Experimental results in Figure 3 show that CRAG consistently outperforms A-RAG in terms of tool usage efficiency across all datasets. Under the same maximum loop constraint, CRAG invokes tools far less frequently than A-RAG, validating both the high efficiency of our method and the effectiveness of the self-correction mechanism in RAG.

6 Conclusion

This paper studies the problem where RAG-based approaches are challenged if retrieval goes wrong, thereby exposing inaccurate and misleading knowledge to generative LMs. Corrective Retrieval Augmented Generation is proposed to improve the robustness of generation. Essentially, an autonomous agent is introduced to drive a dynamic and adaptive retrieval correction. With the further leverage of web search and optimized knowledge utilization, CRAG has significantly improved the ability of automatic self-correction and efficient utilization of retrieved documents. Since we primarily proposed to improve the RAG framework from a corrective perspective, CRAG can be seamlessly coupled with different RAG-based approaches. Experiments extensively demonstrate its adaptability to RAG-based approaches as well as generalizability across various QA tasks.

Limitations

Although our experiments demonstrate consistent improvements of CRAG across multiple benchmarks, there are limitations that persist. The marginal improvements observed on HotpotQA and MuSiQue indicate that our approach yields limited benefits in retrieval environments that already maintain high information quality. When a model can acquire all necessary information through a few standard retrieval steps, incorporating web search introduces additional variability and serves as an unstable knowledge source. Future work will investigate strategies to mitigate the overhead of the noise-robust self-correction mechanism in low-noise settings and explore automatic noise detection capability for the agent.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. pages 675–718.

Sumithra Bhakthavatsalam, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, and Peter Clark. 2021. Think you have solved direct-answer question answering? try arc-da, the direct-answer AI2 reasoning challenge. *CoRR*, abs/2102.03315.

Tom B Brown, Benjamin Mann, Nick Ryder, et al. 2020. Language models are few-shot learners. In *Advances in neural information processing systems*, pages 1877–1901.

DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Shaohan Wang, Pengyu Wang, Xiaorui Wang, and Zhendong Mao. 2026. A-RAG: scaling agentic retrieval-augmented generation via hierarchical retrieval interfaces. *CoRR*, abs/2602.03442.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From RAG to memory: Non-parametric continual learning for large language models. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 7036–7050. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12):248:1–248:38.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7969–7992. Association for Computational Linguistics.

Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. 2024. Sure: Summarizing retrievals using answer candidates for open-domain QA of llms. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8460–8478. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman

Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. 2022. A survey on retrieval-augmented text generation. *CoRR*, abs/2202.01110.

Yanming Liu, Xinyue Peng, Xuhong Zhang, Weihao Liu, Jianwei Yin, Jiannan Cao, and Tianyu Du. 2024. RA-ISF: learning to answer and understand from retrieval augmentation via iterative self-feedback. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 4730–4749. Association for Computational Linguistics.

Hongyin Luo, Tianhua Zhang, Yung-Sung Chuang, Yuan Gong, Yoon Kim, Xixin Wu, Helen Meng, and James R. Glass. 2023. Search augmented instruction learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 3717–3729. Association for Computational Linguistics.

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

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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12076–12100. Association for Computational Linguistics.

Dor Muhlgay, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, and Yoav Shoham. 2023. Generating benchmarks for factuality evaluation of language models. *CoRR*, abs/2307.06908.

OpenAI. 2026. Openai GPT-5 system card. *CoRR*, abs/2601.03267.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

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

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1339–1384. Association for Computational Linguistics.

Md. Rashad Al Hasan Rony, Ricardo Usbeck, and Jens Lehmann. 2022. Dialokg: Knowledge-structure aware task-oriented dialogue generation. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2557–2571. Association for Computational Linguistics.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools.

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.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 3784–3803. Association for Computational Linguistics.

Chao-Hong Tan, Jia-Chen Gu, Chongyang Tao, Zhen-Hua Ling, Can Xu, Huang Hu, Xiubo Geng, and Daxin Jiang. 2022. Tegtok: Augmenting text generation via task-specific and open-world knowledge. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1597–1609. Association for Computational Linguistics.

S. M. Towhidul Islam Tonmoy, S. M. Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *CoRR*, abs/2401.01313.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. 9835 musique: Multihop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554.

Zihao Wang, Anji Liu, Haowei Lin, Jiaqi Li, Xiaojian Ma, and Yitao Liang. 2024. RAT: retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation. *CoRR*, abs/2403.05313.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Tianhua Zhang, Hongyin Luo, Yung-Sung Chuang, Wei Fang, Luc Gaitskell, Thomas Hartvigsen, Xixin Wu, Danny Fox, Helen Meng, and James R. Glass. 2023a. Interpretable unified language checking. *CoRR*, abs/2304.03728.

Tianjun Zhang, Shishir G. Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024. RAFT: adapting language model to domain specific RAG. *CoRR*, abs/2403.10131.

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. 2023b. Siren's song in the AI ocean: A survey on hallucination in large language models. *CoRR*, abs/2309.01219.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Can chatgpt understand too? A comparative study on chatgpt and fine-tuned BERT. *CoRR*, abs/2302.10198.

Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. 2025. Linearrag: Linear graph retrieval augmented generation on large-scale corpora. *CoRR*, abs/2510.10114.

A Task Prompts

Prompt for Agent

Given a question and several documents segmented by "\n\n" that provide the relevant knowledge to answer the question. Your task is to use the provided tool and judge if the information in the each document can FULLY answer the question. If the existing documents are NOT sufficient to answer the question, use the provided information seeking tool function to search for more information.

Select one of the evaluator tools each time if you need to evaluate the relevance of the new document that you want to score. Select one of the information seeking tools each time if you need to find more information about the question from different sources.

If you have gathered enough information, directly synthesis all the useful content of the documents and try to answer the question. The content of documents should be concise and DO NOT say any extra content that is not related to answering the question. While the answer should be as SHORT as possible.

TASK-SPECIFIC INSTRUCT
YOU QUESTION HERE

Note:
Tool Use:
- Use more evaluator tools to evaluate the quality of the documents as reference. The corrected results MUST be evaluated.

Output Format:
After calling all the tools, the format of the ultimate output MUST follow:
> ## Corrected Results:
> [Generate the useful content of documents here]
> ## Answer:
> [Generate your answer to the question here]

The prompts for the agent and the generator across various tasks are demonstrated in this section.

PubHealth Instruction

Is the following statement correct or not? Say true if itś correct; otherwise say false.

Arc-Challenge Instruction

Given four answer candidates, A, B, C and D, choose the best answer choice.

B Ethics Statement

We declare that AI was used solely for language polishing and revision in this paper, and NO factual content, such as data or citations, was generated by AI. All assets involved in this work are open-source and strictly comply with their respective license requirements. We will release the assets created in this paper, such as experimental data and code, which will be distributed under the CC-BY 4.0 license.

C Experiments

C.1 Tasks, Datasets and Metrics

CRAG was evaluated on five datasets, which are in public domain and licensed for research purposes, including:

PopQA (Mallen et al., 2023) is a *short*-form generation task. Generally, only one entity of factual knowledge is expected to be answered for each single question. In our experiments, we exactly followed the setting in Self-RAG (Asai et al., 2024) which evaluated methods on a long-tail subset consisting of 1,399 rare entity queries whose monthly Wikipedia page views are less than 100. Accuracy was adopted as the evaluation metric.

PubHealth (Zhang et al., 2023a) is a task in health care domain consisting of true-or-false questions. Claims are represented about health with factual information, and the model is tasked to verify the authenticity and give the judgment. Accuracy was adopted as the evaluation metric.

Arc-Challenge (Bhakthavatsalam et al., 2021) is a multiple-choice question task about some daily commonsense science phenomena. Given a scientific event that occurs in daily life, the model is required to select the correct description among

3 or 4 optional choices. Accuracy was adopted as the evaluation metric as well.

HotpotQA (Yang et al., 2018) is a multi-hop question answering dataset that requires models to retrieve and synthesize information across multiple Wikipedia documents to answer complex bridge and comparison questions. It is characterized by its provision of sentence-level supporting facts, which enables rigorous evaluation of both multi-step reasoning capabilities and model explainability.

Musique (Trivedi et al., 2022) is a multi-hop QA dataset systematically constructed by composing independent single-hop questions, ensuring that every reasoning step is strictly necessary and eliminating dataset artifacts. Its defining feature is the explicit annotation of intermediate answers and diverse compositional patterns, making it a stringent benchmark for evaluating faithful, shortcut-resistant multi-hop reasoning.

C.2 Experiments compute Resources

We used NVIDIA A800 80GB GPU for experiments. We utilized LLaMA-2-7B that was fine-tuned by Self-RAG (Asai et al., 2024) in the ablation study to replace the strong generator. As for the encoder evaluator, Qwen3-0.6B-Embedding is adopted for similarity calculation.

C.3 Implementation Details

Web Search: Google Search API was adopted to search for the relevant URLs, top-k is set to 5, and pages from Wikipedia will be added preferentially. The searched web pages are generally in the form of HTML files, where content is split with special tokens like and . Thus an extra segmentation like the knowledge refinement is not required, related knowledge paragraphs can be directly selected with the evaluator similar to internal knowledge. In this way, the accuracy of the search outcomes can be ensured without compromising the quality and relevance of the information used for generation.