Corrective Retrieval Augmented Generation

Shi-Qi Yan[1*], **Jia-Chen Gu**[2*], **Yun Zhu**[3], **Zhen-Hua Ling**[1]
[1]National Engineering Research Center of Speech and Language Information Processing, University of Science and Technology of China, Hefei, China
[2]Department of Computer Science, University of California, Los Angeles
[3]Google Research
yansiki@mail.ustc.edu.cn, gujc@ucla.edu, yunzhu@google.com, zhling@ustc.edu.cn

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 the **C**orrective **R**etrieval **A**ugmented **G**eneration (CRAG) to improve the robustness of generation. Specifically, a lightweight retrieval evaluator is designed to assess the overall quality of retrieved documents for a query, returning a confidence degree based on which different knowledge retrieval actions can be triggered. Since retrieval from static and limited corpora can only return sub-optimal documents, large-scale web searches are utilized as an extension for augmenting the retrieval results. Besides, a decompose-then-recompose algorithm is designed for retrieved documents to selectively focus on key information and filter out irrelevant information in them. CRAG is plug-and-play and can be seamlessly coupled with various RAG-based approaches. Experiments on four datasets covering short- and long-form generation tasks show that CRAG can significantly improve the performance of RAG-based approaches.[1]

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; OpenAI, 2023; Touvron et al., 2023a). 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).

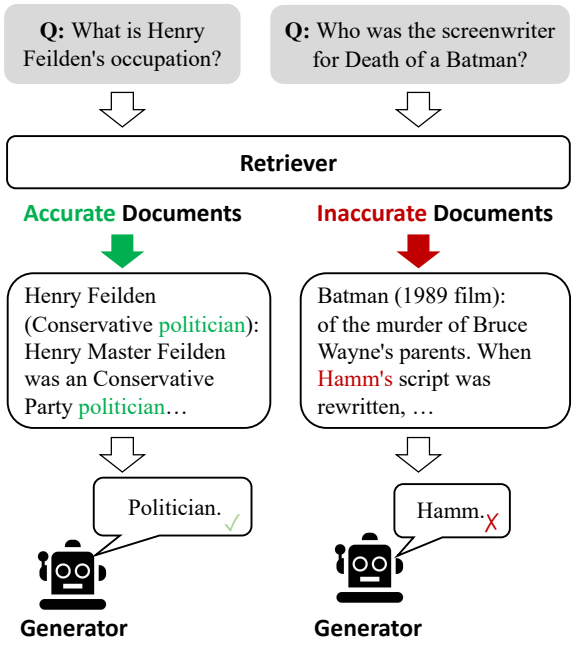

Figure 1: The examples show that a low-quality retriever is prone to introducing a substantial amount of irrelevant information, impeding the generators from acquiring accurate knowledge and potentially misleading them.

Prior research has introduced retrieval techniques to incorporate relevant knowledge 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

[*] Equal contribution.
[1]The code is available at github.com/HuskyInSalt/CRAG

about the model's behavior and performance in scenarios where retrieval may fail or return inaccurate results (Shi et al., 2023). 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 results of retriever and improve the utilization of documents for augmenting generation. A lightweight retrieval evaluator is designed to assess the overall quality of retrieved documents for a query. This serves as a crucial component in RAG, contributing to informative generation by reviewing and evaluating the relevance and reliability of the retrieved documents. A confidence degree is quantified based on which different knowledge retrieval actions of {`Correct`, `Incorrect`, `Ambiguous`} can be triggered. For the latter two actions, 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. Furthermore, to eliminate redundant contexts contained in retrieved documents that are unhelpful for RAG, a decompose-then-recompose algorithm is meticulously crafted throughout the retrieval and utilization process. This algorithm ensures the refinement of retrieved information, optimizing the extraction of key insights and minimizing the inclusion of non-essential elements, thereby enhancing the utilization of retrieved data.

CRAG is plug-and-play and experimentally implemented into RAG (Lewis et al., 2020) and Self-RAG (Asai et al., 2023) for demonstrating its adaptability to RAG-based approaches. Results on four datasets of PopQA (Mallen et al., 2023), Biography (Min et al., 2023), Pub Health (Zhang et al., 2023a), and Arc-Challenge (Bhakthavatsalam et al., 2021) show that CRAG can significantly improve the performance of standard RAG and Self-RAG, demonstrating its generalizability across short- and long-form generation tasks.

In summary, our contributions in this paper are three-fold: 1) This paper studies the scenarios where the retriever returns inaccurate results and, to the best of our knowledge, makes the first attempt to design corrective strategies 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 short- and long-form 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 (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. It usually provides an extra knowledge source from a specific corpus, i.e., Wikipedia, which greatly improves the performance of LMs in a variety of tasks, especially

in the knowledge-intensive ones. The proposed methods generally leverage information retrieval to supply documents containing relevant knowledge for generative LLMs. Earlier studies adopt either sparse or dense retrievers at the front end of a pre-trained language model that specializes in response generation. Despite this, the methods above 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 system can even exacerbate the factual error that LMs make.

Advanced RAG Many advanced approaches have been developed from the original RAG in recent years. Considering that retrieval is sometimes unnecessary for some queries, conversely, responses without retrieval are even more accurate in many situations. Asai et al. (2023) was proposed to selectively retrieve knowledge and introduce a critic model to decide whether to retrieve. Yoran et al. (2023) designed a NLI model to identify the irrelevant context and improve robustness. Luo et al. (2023) is tuned on instructions to insert retrieved documents before instructions. While Schick et al. (2023) pre-train a model for calling APIs such as Wikipedia. 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 decide when and what to retrieve in long-form generation.

Compared with recent studies (Schick et al., 2023; Luo et al., 2023; Asai et al., 2023) 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

Following previous work (Lewis et al., 2020; Asai et al., 2023), given 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 the outputs $\mathcal{Y}$. To retrieve the relevant documents and answer the input text, the entire framework is usually divided into two modules of a retriever $\mathcal{R}$ and a generator $\mathcal{G}$. The retriever $\mathcal{R}$ aims to retrieve the top-$\mathcal{K}$ documents $\mathcal{D} = \{d_{r_1}, ..., d_{r_k}\}$ that are relevant to the input text $\mathcal{X}$ from the corpus $\mathcal{C}$. Based on the input text $\mathcal{X}$ and the retrieved results $\mathcal{D}$, the generator $\mathcal{G}$ is responsible for generating the output text $\mathcal{Y}$. This framework can be formulated as:

$$P(\mathcal{Y}|\mathcal{X}) = P(\mathcal{D}|\mathcal{X})P(\mathcal{Y}|\mathcal{D}, \mathcal{X}). \quad (1)$$

It can be seen from Equ. (1) that the retriever and the generator are seamlessly coupled. This system exhibits a low tolerance for risk, as any unsuccessful retrieval can result in an unsatisfactory response, regardless of the impressive abilities of the generator. This is exactly the focus of this paper to improve the robustness of the generator.

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 the generator. Given an input query and the retrieved documents from any retriever, a lightweight retrieval evaluator is constructed to estimate the relevance score of retrieved documents to the input query (Section 4.2). The calculated relevance score is quantified into a total of three confidence degrees and then triggered the corresponding actions: {`Correct`, `Incorrect`, `Ambiguous`} (Section 4.3). If the action `Correct` is triggered, the retrieved documents will be refined into more precise knowledge strips. This refinement operation involves knowledge decomposition, filter, and recomposition (Section 4.4). If the action `Incorrect` is triggered, the retrieved documents will be discarded. Instead, web searches are resorted to and regarded as complementary knowledge sources for corrections (Section 4.5). Eventually, when the system cannot confidently make a correct or incorrect judgment, a soft action `Ambiguous` which combines both of them is triggered. After optimizing the retrieval results, an arbitrary generative model can be adopted subsequently to generate the final result.

4.2 Retrieval Evaluator

Before utilizing the retrieved documents, it is natural to identify whether the retrieved documents are accurate or not. It is essential since irrelevant or misleading messages can be removed in this way. The accuracy of the retrieval evaluator

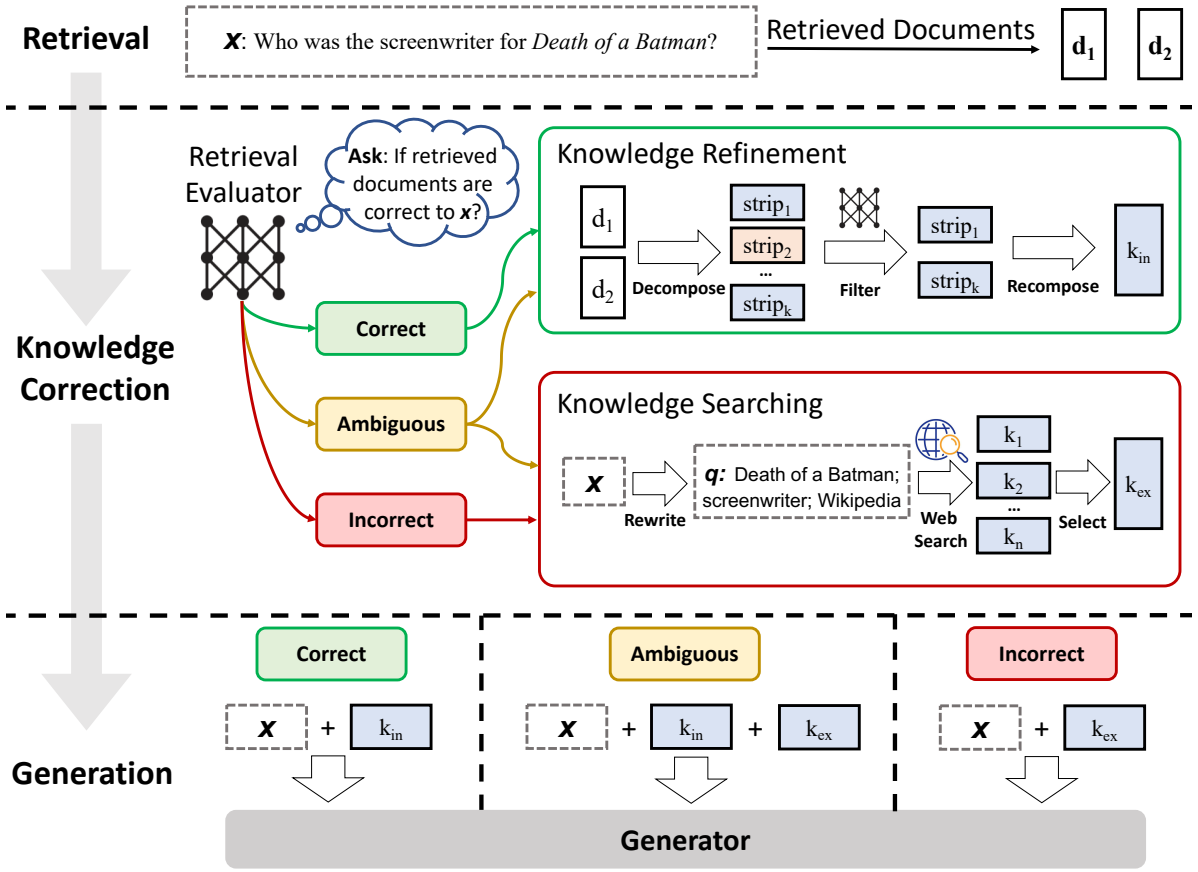

Figure 2: An overview of CRAG at inference. A retrieval evaluator is constructed to evaluate the relevance of the retrieved documents to the input, and estimate a confidence degree based on which different knowledge retrieval actions of {`Correct`, `Incorrect`, `Ambiguous`} can be triggered.

undeniably plays a pivotal role in shaping the overall system performance, as it influences the outcomes of subsequent processes. Our objective is to correct retrieved documents if they are not relevant. Specifically, T5-large (Raffel et al., 2020) is adopted and fine-tuned. For every question, there are generally 10 documents retrieved. The question is concatenated with each single document as the input, and the evaluator predicts the relevance score for each question-document pair individually. During fine-tuning, the label of positive samples is 1, while that of negative ones is -1. At inference, the evaluator scores the relevance from -1 to 1 for each document. We also tried to prompt ChatGPT to identify the retrieval relevance for comparison, but it underperforms as elaborated in Section 5.5. Compared with the critic model of Self-RAG (Asai et al., 2023), the evaluator designed in CRAG shows two great advantages. First, the evaluator is T5-large-based (0.77B), which is quite lightweight compared with the critic model that instruction tuned LLaMA-2 (7B). Second, the evaluator is free from any extra human or LLM annotation, while the critic model requires GPT-4 annotated data for instruction tuning.

4.3 Action Trigger

To correct the irrelevant document and refine the target document as needed, actions should be executed discriminately. Based on the aforementioned confidence score for each retrieved document, three types of actions are designed and triggered accordingly where the upper and lower thresholds are set. If the confidence score is higher than the upper threshold, the retrieved document is identified as `Correct`, while identified as `Incorrect` if it is below the lower threshold. Otherwise, `Ambiguous` is executed. This process is conducted for each retrieved document individually.

Correct Here, a retrieval is assumed `Correct` when the confidence score of *at least one retrieved document* is higher than the upper threshold. If so, it means there are relevant documents in the retrieved results. Even if a relevant document can be found, there is inevitably some noisy knowledge strips in this document. To extract the most

Algorithm 1: CRAG Inference

Require : E (Evaluator), W (Query Rewriter), G (Generator)
Input : x (Input question), $D = \{d_1, d_2, ..., d_k\}$ (Retrieved documents)
Output : y (Generated response)

$score_i$ = E evaluates the accuracy of each pair (x, d_i), $d_i \in D$
Confidence = Calculate and give a final judgment based on $\{score_1, score_2, ...score_k\}$
`// Confidence has 3 optional values: [CORRECT], [INCORRECT] or [AMBIGUOUS]`
if ***Confidence*** == *[CORRECT]* **then**
    Internal_knowledge = Knowledge_Refine(x, D)
    k = Internal_knowledge
else if ***Confidence*** == *[INCORRECT]* **then**
    External_knowledge = Web_Search(W Rewrites x for searching)
    k = External_knowledge
else if ***Confidence*** == *[AMBIGUOUS]* **then**
    Internal_knowledge = Knowledge_Refine(x, D)
    External_knowledge = Web_Search(W Rewrites s for searching)
    k = Internal_knowledge + External_knowledge
end
G predicts y given x, k

critical knowledge strips within this document, a knowledge refinement method is further designed which will be elaborated in Section 4.4.

Incorrect Besides, a retrieval is assumed `Incorrect` when the confidence scores of *all retrieved documents* are below the lower threshold. This indicates that all retrieved documents are considered irrelevant, which are unhelpful for generation. Therefore, we need to seek new sources of knowledge for correction. Here, web search is introduced to search from the Internet as elaborated in Section 4.5. This corrective action helps overcome the embarrassing challenge where no reliable knowledge can be referred to.

Ambiguous Except for the above two situations, the remaining will be assigned to an intermediate action of `Ambiguous`. Since the retrieval evaluator is not confident in its judgment, both types of processed knowledge in `Correct` and `Incorrect` are combined to complement each other. Implementing such a moderating and soft strategy can significantly contribute to strengthening the robustness and resilience of the system, fostering a more adaptable framework for optimal performance.

4.4 Knowledge Refinement

Given a retrieved relevant document, a decompose-then-recompose knowledge refinement method is designed to further extract the most critical knowledge strips in it. First, each retrieved document is segmented into fine-grained knowledge strips through heuristic rules, more details are available in Appendix B. Then, the retrieval evaluator fine-tuned in Section 4.2 is employed to calculate the relevance score of each knowledge strip. Based on these scores, irrelevant knowledge strips are filtered out, while relevant ones are recomposed via concatenation in order, namely internal knowledge.

4.5 Web Search

It is extremely important to seek complementary external knowledge if the retrieved results are all assumed irrelevant. 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. Specifically, the input questions are rewritten into queries composed of keywords by ChatGPT to mimic the daily usage of search engine. The prompt for the rewritten task is shown in Appendix A. In CRAG, the commercial web search API[2] is adopted to generate a series of URL links for every query. Moreover, we utilize the URL links to navigate web pages, transcribe their content, and employ the same knowledge refinement method as Section 4.4 to derive the relevant web knowledge, namely external knowledge.

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

5 Experiments

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

5.1 Tasks, Datasets and Metrics

CRAG was evaluated on four datasets, including:

PopQA (Mallen et al., 2023) is a *short*-form generation task that contains 1,399 questions. Generally, only one entity of factual knowledge is expected to be answered for each single question. Accuracy was adopted as the evaluation metric.

Biography (Min et al., 2023) is a *long*-form generation task that is tasked to generate a detailed biography about a certain entity. Following previous work, FactScore (Min et al., 2023) was adopted to evaluate the generated biographies.

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.

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 some public LLMs, LLaMA2-7B,13B (Touvron et al., 2023b), instruction-tuned models, Alpaca-7B,13B (Dubois et al., 2023), and CoVE$_{65B}$ (Dhuliawala et al., 2023) which introduces iterative engineering to improve the factuality of LLM generations.

Standard 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 same retriever as in our system. Here we adopted several public instruction-tuned LLMs, including LLaMA2-7B, 13B (Touvron et al., 2023b), Alpaca-7B,13B (Dubois et al., 2023), as well as LLaMA2-7B instruction-tuned in Self-RAG (Asai et al., 2023).

Advanced RAG. (1) SAIL (Luo et al., 2023) that instruction-tuned an LM on the Alpaca instruction-tuning data with top retrieved documents inserted before instructions. (2) Self-RAG (Asai et al., 2023) that tuned the LLaMA2 on the instruction-tuning data comtaining several sets of reflection tokens which were labeled by GPT-4 (OpenAI, 2023). (3) Following Asai et al. (2023), we also cited the results of retrieval-augmented baselines trained with private data: Ret-ChatGPT and Ret-LLaMA-chat, which deploy the same augmentation technique above, as well as perplexity.ai, an InstructGPT-based production search system.

5.3 Results

Table 1 presents the results on four datasets. The model coupling the proposed method with standard RAG is named CRAG and that coupling with Self-RAG is named Self-CRAG. Readers can refer to Appendix B for more implementation details of our proposed method. From these results, we can conclude the following findings:

First, the proposed method can significantly improve the performance of both RAG and Self-RAG. Specifically, CRAG outperformed RAG by margins of 2.1% accuracy on PopQA and 2.8% FactScore on Biography when based on *LLaMA2-hf-7b*, as well as by margins of 19.0% accuracy on PopQA, 14.9% FactScore on Biography, 36.6% accuracy on PubHealth, and 8.1% accuracy on Arc-Challenge when based on *SelfRAG-LLaMA2-7b*. Compared with the current state-of-the-art Self-RAG, Self-CRAG outperformed it by margins of 20.0% accuracy on PopQA, 36.9% FactScore on Biography, 4.0% accuracy on Arc-Challenge when based on *LLaMA2-hf-7b*, as well as by margins of 6.9% accuracy on PopQA, 5.0% FactScore on Biography, 2.4% accuracy on PubHealth, when based on *SelfRAG-LLaMA2-7b*. These results demonstrated the adaptability of CRAG which is plug-and-play and can be implemented into RAG-based approaches.

Second, the proposed method demonstrated great generalizability across a variety of generation tasks. In particular, these benchmarks reported in Table 1 respectively represent different practical scenarios including short-form entity generation (PopQA), long-form generation (Biography), and closed-set tasks (PubHealth, Arc-Challenge). The results verified the consistent effectiveness of

Method	PopQA (Accuracy)	Bio (FactScore)	Pub (Accuracy)	ARC (Accuracy)
LMs trained with propriety data				
LLaMA2-c$_{13B}$	20.0	55.9	49.4	38.4
Ret-LLaMA2-c$_{13B}$	51.8	79.9	52.1	37.9
ChatGPT	29.3	71.8	70.1	**75.3**
Ret-ChatGPT	50.8	-	54.7	**75.3**
Perplexity.ai	-	71.2	-	-
Baselines without retrieval				
LLaMA2$_{7B}$	14.7	44.5	34.2	21.8
Alpaca$_{7B}$	23.6	45.8	49.8	45.0
LLaMA2$_{13B}$	14.7	53.4	29.4	29.4
Alpaca$_{13B}$	24.4	50.2	55.5	54.9
CoVE$_{65B}$	-	71.2	-	-
Baselines with retrieval				
LLaMA2$_{7B}$	38.2	78.0	30.0	48.0
Alpaca$_{7B}$	46.7	76.6	40.2	48.0
SAIL	-	-	69.2	48.4
LLaMA2$_{13B}$	45.7	77.5	30.2	26.0
Alpaca$_{13B}$	46.1	77.7	51.1	57.6
LLaMA2-hf-7b				
RAG	37.7	44.9	**9.1**	23.8
CRAG	39.8	47.7	**9.1**	25.8
Self-RAG*	29.0	32.2	0.7	23.9
Self-CRAG	**49.0**	**69.1**	0.6	**27.9**
SelfRAG-LLaMA2-7b				
RAG	40.3	59.2	39.0	46.7
CRAG	59.3	74.1	**75.6**	54.8
Self-RAG	54.9	81.2	72.4	67.3
Self-CRAG	**61.8**	**86.2**	74.8	67.2

Table 1: Overall evaluation results on the test sets of four datasets. Results are separated based on the generation LLMs. **Bold** number indicates the best performance among all methods and LLMs. **Gray-colored** bold score indicates the best performance using a specific LLM. * indicates results that were reproduced by us, otherwise results except ours are cited from their original papers. - indicates scores that are not reported in the original paper or have not been evaluated.

CRAG. Its versatility across a spectrum of tasks underscores its robust capabilities and generalizability across diverse scenarios.

Third, the proposed method exhibited greater flexibility in replacing the underlying LLM generator. It can be seen that CRAG still showed competitive performance when the underlying LLMs was changed from *SelfRAG-LLaMA2-7b* to *LLaMA2-hf-7b*, while the performance of Self-RAG dropped significantly, even underperforming the standard RAG. The reason for these results is that Self-RAG needs to be instruction-tuned using human or LLM annotated data to learn to output special critic tokens as needed, while this ability is not learned in common LLMs. CRAG does not have any requirements for this ability. As you can imagine, when more advanced LLMs are available in the future, they can be coupled with CRAG easily, while additional instruction tuning is still necessary for Self-RAG.

It is worth mentioning that the performance on PubHealth based on *LLaMA2-hf-7b* was much worse than others. We studied these cases and found that *LLaMA2-hf-7b* is relatively weak in instruction comprehension. Most of the cases fail to generate `True` or `False` in such a binary-question task, resulting in a low accuracy during the evaluation. This situation somewhat happens in Arc-Challenge as well, when the model is tasked to generate the index of a candidate.

	LLaMA2-hf-7b	SelfRAG-LLaMA2-7b
CRAG	47.3	59.3
w/o. `Correct`	44.5	58.1
w/o. `Incorrect`	46.8	58.6
w/o. `Ambiguous`	45.7	58.5
Self-CRAG	49.0	61.8
w/o. `Correct`	43.6	59.6
w/o. `Incorrect`	47.7	60.8
w/o. `Ambiguous`	48.1	61.5

Table 2: Ablation study for removing each single action on the PopQA dataset in terms of accuracy. When the action `Correct` or `Incorrect` was removed, the proportion that originally triggered `Correct` or `Incorrect` would trigger `Ambiguous`. When `Ambiguous` was removed, all input queries clearly triggered `Correct` or `Incorrect`.

	LLaMA2-hf-7b	SelfRAG-LLaMA2-7b
CRAG	47.3	59.3
w/o. refinement	38.9	47.0
w/o. rewriting	44.8	56.6
w/o. selection	44.0	53.8
Self-CRAG	49.0	61.8
w/o. refinement	35.9	52.2
w/o. rewriting	37.2	58.4
w/o. selection	57.9	57.9

Table 3: Ablation study for removing each knowledge utilization operation on the PopQA dataset in terms of accuracy. Removing document refinement denoted that the original retrieved documents were directly fed to the generator. Removing search query rewriting denoted that questions were not rewritten into queries consisting of keywords during knowledge searching. Removing knowledge selection denoted that all searched content of web pages was all regarded as the external knowledge.

5.4 Ablation Study

The impact of each triggered action. To further verify the effectiveness of triggered actions designed in the retrieval evaluator, ablation tests for removing each single action in the proposed method were conducted as shown in Table 2. Evaluations on the PopQA dataset were conducted to demonstrate the performance change in terms of accuracy. Specifically, when the action `Correct` or `Incorrect` was removed, it was merged with `Ambiguous` so that the proportion that originally triggered `Correct` or `Incorrect` would trigger `Ambiguous`. On the other hand, when the action `Ambiguous` was removed, there was only one threshold against which all input queries clearly triggered `Correct` or `Incorrect`. From these results, it can be seen that there was a performance drop no matter which action was removed, illustrating that each action contributed to improving the robustness of generation.

	Accuracy
Our Retrieval Evaluator (T5-based)	84.3
ChatGPT	58.0
ChatGPT-CoT	62.4
ChatGPT-few-shot	64.7

Table 4: Evaluation of our retrieval evaluator and ChatGPT for the retrieval results on the PopQA dataset.

The impact of each knowledge utilization operation. Table 3 illustrated how the performance changed if a key knowledge utilization operation was ablated. Evaluations on the PopQA dataset in terms of accuracy were conducted by individually removing the knowledge utilization operations of document refinement, search query rewriting, and external knowledge selection. Removing document refinement denoted that the original retrieved documents were directly fed to the following generator, as in most existing works. Additionally, removing search query rewriting denoted that questions were not rewritten into queries consisting of keywords during knowledge searching. Eventually, removing knowledge selection denoted that all searched content of web pages was all regarded as the external knowledge without selection. These results help derive the findings that the performance of the final system degraded no matter which knowledge utilization operation was removed, revealing that each knowledge utilization operation contributed to improving the utilization of knowledge.

5.5 Accuracy of the Retrieval Evaluator

The quality of the retrieval evaluator significantly determined the performance of the entire system. Given the document retrieval results, we assessed whether the retrieval evaluator can accurately determine the overall quality of these results. The assessment accuracy of our retrieval evaluator and the commercial LLM ChatGPT on the document retrieval results was shown in Table 4. The prompts of *ChatGPT*, *ChatGPT-CoT*, and *ChatGPT-few-shot* used in our experiments can be referred to in Appendix A. Results reveal that the lightweight T5-based retrieval evaluator significantly outperformed the competitive ChatGPT in all settings.

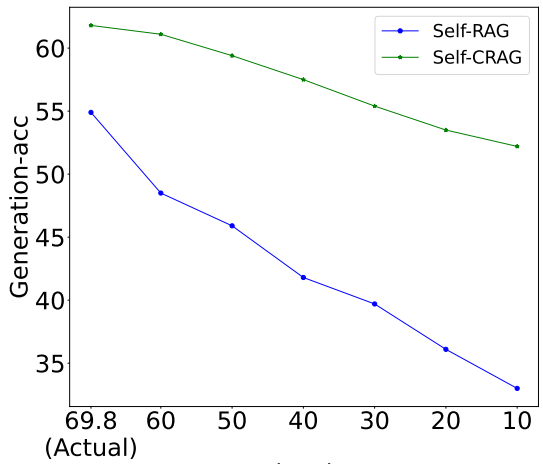

Figure 3: The generation performance of Self-RAG and Self-CRAG given different retrieval performance on the PopQA dataset.

5.6 Robustness to Retrieval Performance

To further verify the robustness of the proposed method to retrieval performance, we studied how the generation performance changed given different retrieval performance. A part of accurate retrieval results were deliberately removed at random to imitate a low-quality retriever and evaluate how the performance changed. Figure 3 demonstrated the performance change of Self-RAG and Self-CRAG on the PopQA dataset. It can be seen that the generation performance of Self-RAG and Self-CRAG dropped as the retrieval performance dropped, indicating that the generator relied heavily on the quality of the retriever. Furthermore, as the retrieval performance dropped, the generation performance of Self-CRAG dropped more slightly than that of Self-RAG. These results imply the superiority of Self-CRAG over Self-RAG on enhancing the robustness to retrieval performance.

6 Conclusion

This paper studies the problem where RAG-based approaches are challenged if retrieval goes wrong, thereby exposing inaccurate knowledge to generative LMs. To mitigate this problem, we propose the plug-and-play Corrective Retrieval Augmented Generation (CRAG) to improve the robustness of generation. Essentially, a retrieval evaluator is to estimate and trigger three actions discriminately. With the further help of web search and optimized knowledge utilization operations, CRAG has significantly improved the ability of automatic self-correction and efficient utilization of retrieved documents. Experiments coupling CRAG with standard RAG and Self-RAG extensively demonstrate its adaptability to RAG-based approaches, and those on four datasets demonstrate its generalizability across short- and long-form generation tasks.

References

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, et al. 2023. PaLM 2 technical report. *CoRR*, abs/2305.10403.

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

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. *CoRR*, abs/2302.04023.

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.

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, et al. 2023. PaLM: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. *CoRR*, abs/2309.11495.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy

Liang, and Tatsunori B. Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *CoRR*, abs/2305.14387.

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.

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.

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 Lewis, Ethan Perez, and Aleksandra others Piktus. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

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

Hongyin Luo, Yung-Sung Chuang, Yuan Gong, Tianhua Zhang, Yoon Kim, Xixin Wu, Danny Fox, Helen Meng, and James R. Glass. 2023. SAIL: search-augmented instruction learning. *CoRR*, abs/2305.15225.

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. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

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? *CoRR*, abs/2302.06476.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Md Rashad Al Hasan Rony, Ricardo Usbeck, and Jens Lehmann. 2022. Dialokg: Knowledge-structure aware task-oriented dialogue generation. *arXiv preprint arXiv:2204.09149*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *CoRR*, abs/2302.04761.

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.

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. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *CoRR*, abs/2310.01558.

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.

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.

A Task Prompts

The prompts for generating knowledge keywords as web search queries were illustrated in Table 5.

Extract at most three keywords separated by comma from the following dialogues and questions as queries for the web search, including topic background within dialogues and main intent within questions. question: What is Henry Feilden's occupation? query: Henry Feilden, occupation question: In what city was Billy Carlson born? query: city, Billy Carlson, born question: What is the religion of John Gwynn? query: religion of John Gwynn question: What sport does Kiribati men's national basketball team play? query: sport, Kiribati men's national basketball team play question: [question] query:

Table 5: The few-shot prompt to GPT-3.5 Turbo for generating knowledge keywords as web search queries.

The prompts to instruct ChatGPT as the evaluator were illustrated in Table 6, Table 7, and Table 8 respectively.

Given a question, does the following document have exact information to answer the question? Answer yes or no only. Question: [question] Document: [document]

Table 6: The direct prompt to GPT-3.5 Turbo as the evaluator.

Given a question, does the following document have exact information to answer the question? Question: [question] Document: [document] Think Step by step, and answer with yes or no only.

Table 7: The prompt to GPT-3.5 Turbo with Chain-of-Thought as the evaluator.

B Implementation Details

Evaluator: We fine-tuned the task-specific retrieval evaluator model based on the lightweight T5-large (Raffel et al., 2020) pre-trained model. Its parameter size is much smaller than the most current LLMs such as the LLaMA series (Touvron et al., 2023a,b) which are adopted by Self-RAG, PaLM (Chowdhery et al., 2023; Anil et al., 2023)

Given a question, does the following document have exact information to answer the question? Answer yes or no only. Question: In what city was Abraham Raimbach born? Document: Bancroft was born on November 25, 1839 in New Ipswich, New Hampshire to James Bancroft and Sarah Kimball. At an early age he was cared for by Mr. and Mrs. Patch of Ashby, Massachusetts, the neighboring town. While not legally adopted, they named him Cecil Franklin Patch Bancroft, adding Franklin Patch after the son Mr. and Mrs. Patch had who recently died. He attended public schools in Ashby as well as the Appleton Academy in New Ipswich. He entered Dartmouth College in 1856 at the age of sixteen and graduated in 1860 near the top of his class. Bancroft continued his education as he began his career in teaching. He took classes at the Union Theological Seminary in New York City during the 1864-65 academic year. While there he was a member of the United States Christian Commission, traveling to support soldiers during the Civil War. He then transferred to the Andover Theological Seminary where he would graduate in 1867. Answer: No. Question: In what country is Wilcza Jama, Sokółka County? Document: Wilcza Jama is a village in the administrative district of Gmina Sokółka, within Sokółka County, Podlaskie Voivodeship, in north-eastern Poland, close to the border with Belarus. Answer: Yes. Question: What sport does 2004 Legg Mason Tennis Classic play? Document: The 2004 Legg Mason Tenis Classic was the 36th edition of this tennis tournament and was played on outdoor hard courts. The tournament was part of the International Series of the 2004 ATP Tour. It was held at the William H.G. FitzGerald Tennis Center in Washington, D.C. from August 16 through August 22, 2004. Answer: Yes. Question: Who is the author of Skin? Document: The Skin We're In: A Year of Black Resistance and Power is a book by Desmond Cole published by Doubleday Canada in 2020. The Skin We're In describes the struggle against racism in Canada during the year 2017, chronicling Cole's role as an anti-racist activist and the impact of systemic racism in Canadian society. Among the events it discusses are the aftermath of the assault of Dafonte Miller in late 2016 and Canada 150. The work argues that Canada is not immune to the anti-Black racism that characterizes American society. Due to an error by the publisher, the initial printing of the book's cover did not include word B̈lackïn the subtitle. The mistake was later corrected. The book won the Toronto Book Award for 2020. In 2021, the book was nominated for the Shaughnessy Cohen Prize for Political Writing. Answer: No. Question: [question] Document: [document] Answer:

Table 8: The few-shot prompt to GPT-3.5 Turbo as the evaluator.

series, GPT series (Brown et al., 2020; Ouyang et al., 2022; OpenAI, 2023), and so on. The two confidence thresholds for triggering one of the three actions were calculated on the experience of the training set, to maximize the overall accuracy with a false-positive punishment. The maximization function can be concluded as :

$$S = \max(\frac{TP + TN}{n_{total}} - \alpha\frac{FP}{n_{neg}}) \quad (2)$$

here n_{total} means all samples and n_{neg} means all samples that are judged as incorrect, α is set to 1 in this experiments. As the trained evaluator tends to predict polarizing scores, the first threshold is often set around 0.5, such as 0.59 on PopQA and 0.5 on PubQA, and the second threshold is set around -0.9.

Internal Knowledge: To obtain fine-grained retrieval results, we segmented the retrieved results into internal strips. If a retrieved result is as short as one or two sentences, it is regarded as an individual strip, otherwise, retrieval documents are required to be split into smaller units which generally consist of a few sentences according to the total length. The scale is assumed to include an independent piece of information, and the filtering is based on the segments. We directly adopted the evaluator again for knowledge strips filtering, and the top-k is set to 5, filter threshold as -0.5.

External Knowledge: 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.

Generator: As CRAG is a plug-and-play method, all generation models that can be utilized in RAG fit our approach as well. To be consistent with baselines for comparison, we adopted LLaMA2 (Touvron et al., 2023b) for the generation. We first introduced the *LLaMA2-hf-7b* from huggingface to generate responses. Since Self-RAG (Asai et al., 2023) fine-tuned LLaMA2 and reached a new state-of-the-art performance on several tasks, we further utilized the launched model, *SelfRAG-LLaMA2-7b*, as a new generator to be consistent with their work and study the specific improvement of our method.

Self-CRAG: To demonstrate that our plug-and-play approach can be utilized in other concurrent studies, we specifically designed to insert our CRAG into the Self-RAG (Asai et al., 2023) framework and named it Self-CRAG. Self-RAG is an advanced RAG approach that introduces a critic model to decide whether to retrieve and which retrieved document to be referred for generation. It meets our demand for deciding which action to be triggered, thus we replaced the retrieved items in Self-RAG with our processed internal knowledge for `Correct`, external knowledge for `Incorrect`, and combined knowledge for `Ambiguous`.