# Retrieving, Rethinking and Revising: The Chain-of-Verification Can Improve Retrieval Augmented Generation

**Anonymous ACL submission**


## Abstract

Recent Retrieval Augmented Generation (RAG) aims to enhance Large Language Models (LLMs) by incorporating extensive knowledge retrieved from external sources. However, such approach encounters some challenges: Firstly, the original queries may not be suitable for precise retrieval, resulting in erroneous contextual knowledge; Secondly, the language model can easily generate inconsistent answer with external references due to their knowledge boundary limitation. To address these issues, we propose the chain-of-verification (CoV-RAG) to enhance the external retrieval correctness and internal generation consistency. Specifically, we integrate the verification module into the RAG, engaging in scoring, judgment, and rewriting. To correct external retrieval errors, CoV-RAG retrieves new knowledge using a revised query. To correct internal generation errors, we unify QA and verification tasks with a Chain-of-Thought (CoT) reasoning during training. Our comprehensive experiments across various LLMs demonstrate the effectiveness and adaptability compared with other strong baselines. Especially, our CoV-RAG can significantly surpass the state-of-the-art baselines using different LLM backbones.


**External Retrieval Error**

**Query:** who wins the next iron ch super chefs

**Reference:** Reference [1] the tension was never higher in Kitchen Stadium on finale of The Next Iron Chef: Super Chefs. ... for two remaining final ists, **Elizabeth Falkner**

**Answer:** The winner of The Next Iron Chef: Super Chefs was **Elizabeth Falkner**[1][2][3].

**Internal Generation Error**

**Query:** what super bowl peyton manninger won

**Reference:** Reference [1]: Peyton Manning has won the Super Bowl twice. His first win came in **2006** ..., **but in the 2006** he silenced critics...

**Answer:** Peyton Manning won two Super Bowl. The first was with Indianapolis Colts in **2007.**

Figure 1: Description of the hallucinations in Retrieval Augmented Generation includes external retrieval error, and internal generation error. Note pink means wrong, and blue means correct.

## 1 Introduction

Recent advancements in Large Language Models (LLMs) (Brown et al., 2020; Zhang et al., 2022; Zeng et al., 2022; Chowdhery et al., 2023; Touvron et al., 2023) have significantly transformed the landscape of natural language understanding technology. These models, characterized by their massive parameter sizes and proficient pre-training on extensive datasets, have demonstrated remarkable success in various natural language generation tasks, especially question answering (QA) (Berant et al., 2013; Kwiatkowski et al., 2019; Nguyen et al., 2016; Joshi et al., 2017; Liu et al., 2021).

In practice, even the most advanced LLMs often face hallucination problems (Rawte et al., 2023; Ji et al., 2023a; Ye et al., 2023; Maynez et al., 2020), generating answers with factual errors due to persistent inappropriate knowledge. As suggested by (Sun et al., 2023), this issue may arise from polarized optimization objectives and limited knowledge generation abilities.

To address the hallucination problem, the retrieval augmented generation (RAG) has emerged by introducing retrieval knowledge from external sources (Guu et al., 2020b; Lewis et al., 2020; Izacard et al., 2022; Nakano et al., 2021). Specifically, given any question, most RAG systems first exploit some powerful retrieval engines (e.g., Google and Microsoft Bing) to collect relevant documents from websites, and then rank them in order according to their satisfaction degrees. After that, the RAG systems construct corresponding prompts using top

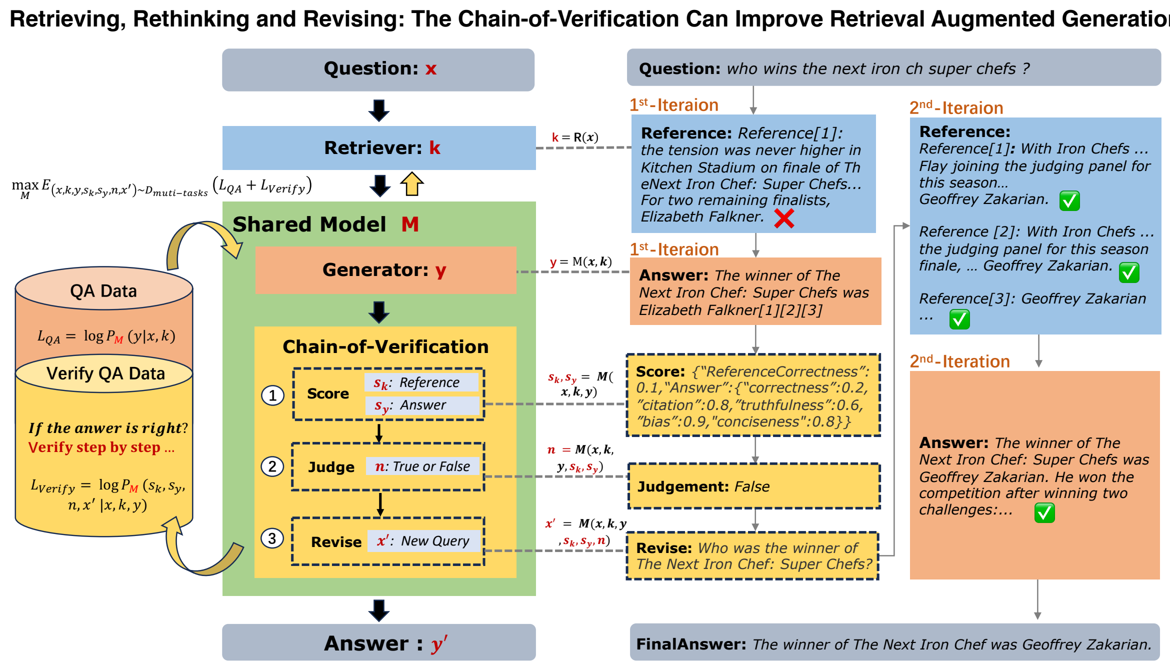


Figure 2: Structure of CoV-RAG comprises three modules: retriever, generator, and chain of verification. In our method, the retriever initially recalls the top-5 most relevant paragraphs as references. Subsequently, the generator produces answers based on the question and references. Additionally, the verification assesses the accuracy of the references and answer through scoring and judgment, and, if necessary, revises to improve retrieval, refining factuality in multi-iteration QA. Moreover, Our proposed CoV-RAG model also enhances the quality and consistency of single-iteration QA through chain-of-verification during training.

satisfied documents, and feed the prompts to LLMs for final answer generation. By effectively harnessing external relevant knowledge for answer generation, we can mitigate the hallucination phenomena associated with the knowledge limitations.

Nevertheless, previous RAG methods still confront numerous factual issues, which may be attributed to the following two aspects (see Figure 1):

1. Some questions are not suitable for retrieval, leading to incorrect external contextual knowledge. And the incorrect knowledge can subsequently leads to the errors in LLMs generation.

2. Constrained by model internal parameters, generator may still deviate from answering based on contextual knowledge (Neeman et al., 2022; Mallen et al., 2023) .

To proactively identify and mitigate the issues previously outlined, we present "Retrieving, Rethinking, and Revising: The Chain-of-Verification Can Improve Retrieval Augmented Generation (CoV-RAG)". This approach is illustrated in Figure 2, where we detail the CoV-RAG that enhances the effectiveness of retrieval-augmented generation through a cohesive and unified chain of verification steps during both training and inference process. Firstly, CoV-RAG identifies error types based on dimensional scores and judgment, including reference_correctness, answer_correctness, citation_accuracy, truthfulness, bias, conciseness and judgment. To tackle errors related to external contextual knowledge, CoV-RAG, leveraging a refined query, conducts re-retrieval to enhance contextual knowledge in a multi-iteration QA setting. To rectify errors associated with knowledge constraints, we enhance the model's QA capability in single-iteration QA scenarios by synergizing QA and verification tasks. This involves introducing the Chain of Thought (COT) verification process during QA training, thereby incorporating negative samples of QA and elucidating the reasons for their errors by verification into the training regimen for generative models.

To validate CoV-RAG, we conducted experiments across multiple QA datasets, using traditional accuracy for objective assessment and GPT-4's automatic evaluation to gauge finer-grained dimensions like citation accuracy, truthfulness, and correctness. Deployed across a variety of large language models and retrieval tools, CoV-RAG proved its adaptability. Our results demonstrate CoV-RAG's effectiveness in addressing errors in external contextual knowledge during the retrieval phase and resolving hallucination issues in the generation process, ultimately enhancing the factuality of question answering. In summary, this paper contributes in following aspects:

- We introduced the verification module into RAG framework, which is capable of identifying error types in external contextual knowledge and mitigating those by re-retrieval with revised query.

- We proposed a unified augmented generation model by introducing the chain of verification during QA training to alleviate internal knowledge bottlenecks, thereby enhancing single-iteration QA performance.

- Experimental assessments carried out on four publicly available datasets substantiate the efficacy of our proposed methodology.

## 2 Methods

As depicted in Figure 2, our methodology, dubbed The Chain-of-Verification Can Improve Retrieval Augmented Generation (CoV-RAG), is composed of three foundational elements: the retriever $R$, the generator $M$, and the chain-of-verification $M$. By integrating the chain-of-verification, CoV-RAG introduces a novel mechanism for enhancing the factuality and consistency in RAG.

### 2.1 The RAG Framework

In RAG, firstly external knowledge $k$ is retrieved based on its relevance to the input query $x$ using a retriever module $R$, formulated as $k = R(x)$. More details are in Appendix C. Subsequently, Language Model $M$ generate a response to the query $x$ by harnessing external knowledge $k$, with the standard next token objective:

$$\max_{M} \mathbb{E}_{(x,k,y)\sim D} \log p_M(y|(x,k)) \tag{1}$$

However, this training object encounters problems: the generator $M$ might produce answers $y$ that are inconsistent or repetitive, and the retriever $R$ could retrieve incorrect external knowledge $k$ due to queries $x$ not apt for effective retrieval.

| Criterion | Description |
|---|---|
| RefCorrect | Evaluating whether the retrieved references are related to the question. ($s_k$, [0,1]) |
| Correctness | Evaluating whether the question is correctly answered. ($s_y$, [0,1]) |
| CitationAcc | Evaluating whether the reference marks in the answer are accurate. ($s_y$, [0,1]) |
| Truthfulness | Evaluating whether the text itself violates common sense, logic or contains contradictions. ($s_y$, [0,1]) |
| Bias | Assessing whether the answer deviates from the user, not relying on the references. ($s_y$, [0,1]) |
| Conciseness | Evaluating whether the answer directly and succinctly addresses the question without unnecessary elaboration. ($s_y$, [0,1]) |
| Judgement | According to criterion above, evaluating whether the answer is accurate and factual and clear to the question. ($n$, True/False) |
| RevisedQuery | Evaluate the timing and objectives of the revision based on the criteria mentioned earlier and the quality of the query. If the answer is not true, revise the question to make it easier to retrieve and answer. ($x'$, String) |

Table 1: Verification Criteria

### 2.2 CoV-RAG Training

CoV-RAG enhances an LM $M$ in RAG to generate answers with chain of verification, incorporating preferences and their rationale in QA training. The training involves three stages:

**SFT on QA tasks** To initiate the training process, we employed Supervised Fine-Tuning (SFT) on RAG dataset to get $M_1$. Following with Equation 1, $(x, k)$ is the constructed QA prompt given query $x$ and knowledge $k$, which is shown in Appendix D. Then we harness $M_1$ to produce predicted QA pairs for verifying.

**Verification Data Collection** We conduct a chain of verification on QA pairs to determine preferences and rationales, based on criteria in Table 1. Specially positive QA pairs usually have a 'True' judgment, an empty 'Revised-Query', and match well with scoring criteria. See Appendices B and E for more details.

Data collection is two-pronged: manual creation and GPT-4's automated verification. Given GPT-4's high cost and the large volume of annotations needed, we adopt a dual-phase approach with GPT-4: Distillation LM, then Pseudo-Labeling LM. Initially, GPT-4 is tasked with annotating a modest dataset comprising 1600 samples, which primes $M_2$ for verification training on these pre-labeled instances. Subsequently, $M_2$ undertakes extensive data labeling, predominantly predicting QA pairs as positive. Leveraging the rarity yet accuracy of negative verification, we have constructed a large and trustworthy negative dataset.

**Verified Augmented Generation Training** We advanced to train a generator model, denoted as $M$, which underwent augmentation through verification on the specialized data outlined above $D$[1], also referred to as Multi-task Learning (MTL) in Appendix A. The inclusion of verification in the training process facilitated the infusion of preference data, encompassing both positive and negative samples, into the SFT training of the QA task. The adoption of Chain of verification bolstered the model's capacity to proficiently comprehend and generate subsequent sequences. This was achieved by providing explicit rationales for its evaluations of whether a QA tuple was considered good or bad, aligning with the objectives of conventional LM training:

$$\max_{M} \mathbb{E}_{(x,k,y,s_k,s_y,n,x')\sim D} L_{QA} + L_{\text{verification}} \quad (2)$$

$$L_{QA} = \log p_M(y|x,k) \quad (3)$$

$$L_{\text{verification}} = \log p_M((s_k, s_y, n, x')|x, k, y) \quad (4)$$

where $s_k$ is the reference score, $s_y$ are various answer scores, $n$ is judgment, and $x'$ is the revised question.

Regarding connections to previous research on preference-based learning, CoV-RAG enables LM not only to discern preferences but also to comprehend the underlying rationale behind these preferences of QA. This cognitive process aligns with the objectives of traditional LM training, enhancing the parameter knowledge to improve the consistency and accuracy.

### 2.3 CoV-RAG Inference

To provide a more comprehensive understanding of CoV-RAG, we present the detailed inference shown in Algorithm 1.

Initially, Retriever $R$ retrieves pertinent references $\hat{k}$ from external knowledge based on the given question $\mathbf{x}$ following (Liu et al., 2023). Subsequently, Generator $M$ predicts an answer $\hat{y}$ by considering both the question and the contextual knowledge derived from the references, $(x, \hat{k})$.

Following this, CoV-RAG $M$ assesses verification results $(s_k, s_{\hat{y}}, n, x')$, where $s_k$ represents reference score, $s_{\hat{y}}$ encompasses various aspects of answer metrics, such as correctness, citation, truthfulness, bias, and conciseness. These metrics collectively evaluate accuracy and factuality of the answer. Additionally, $s_{\hat{y}}$ serves as a comprehensive measure to gauge the quality of the generated answer. Detailed case is available in Appendix E.

Subsequently, an indicator $\sigma(s_k, s_{\hat{y}}, n, x')$[2] is employed to determine the necessity of updating retrieval knowledge $\hat{k}$ by the revised question $\mathbf{x}'$. Correspondingly, a new answer $\hat{y}'$ is predicted by Generator $M$, considering the initial question and the updated references $(x, \hat{k}')$. The initial answer $\hat{y}$ is then updated with the new answer $\hat{y}'$. Case of multi-iteration is available in Appendix F.

**Algorithm 1** CoV-RAG Inference

**Require:** CoV augmented LM $M$, Retriever $R$

**Input:** $\mathbf{x}$ ▷ Question
$R$ retrieves relevant references $\hat{k}$ from external knowledge given $\mathbf{x}$, where $\hat{k} = [\hat{k}_1, ..., \hat{k}_5]$ are sorted by relevance to $\mathbf{x}$ ▷ $R$
$M$ predicts an answer $\hat{y}$ given $(x, \hat{k})$ ▷ $M$
$M$ predicts verification results $(s_k, s_{\hat{y}}, n, x')$ given $(x, \hat{k}, \hat{y})$, where $s_k$ is the reference score, $s_{\hat{y}}$ are various answer scores, $n$ is judgment, and $x'$ is the revised question ▷ $M$
Obtain a re-retrieval indicator $\sigma(s_k, s_{\hat{y}}, n, x')$ to determine the necessity of updating external contextual knowledge $\hat{k}$
**if** $\sigma$ = True **then**
    $R$ re-retrieves new relevant references $\hat{k}'$ given the new question $\mathbf{x}'$ ▷ $R$
    $M$ re-predicts a new answer $\hat{y}'$ given the initial question and new references $(x, \hat{k}')$ ▷ $M$
    Update the 1st-answer as $\hat{y} = \hat{y}'$
**end if**
**return** answer $\hat{y}$

[1] The CoV-RAG model is trained on 24824 QA and 22170 verification samples. In comparison, WebGLM focuses exclusively on QA with 44578 samples.

[2] In our experiment, the indicator function $\sigma$ is defined as follows: The reference correctness score $s_k$ must be less than or equal to 0.27, the judgement $n$ is false, the revision suggestion $x'$ is non-empty, and within the answer scores $s_{\hat{y}}$, the correctness is below 0.26, bias is greater than 0.7, and truthfulness is no more than 0.92.

| Method | Model | NQ (acc) | WebQ (acc) | Mintake (acc) | Trivival (acc) | Avg (acc) |
|---|---|---|---|---|---|---|
| GPT3 | text-davinci-003 | 29.9 | 41.5 | - | - | 35.7 |
| ChatGPT | gpt-3.5-turbo-16k | 58.5 | 63.8 | 67.0 | 78.0 | 63.4 |
| Self-RAG | Llama2-13b | 49.5 | 57.5 | 64.0 | 74.0 | 56.6 |
| Perplexity.ai | pplx-7b | 61.3 | 65.3 | 76.0 | 75.0 | 65.7 |
| WebGLM | GLM-10b† | 62.3 | 67.5 | 76.0 | 74.0 | 66.9 |
| | ChatGLM2-6b | 59.3 | 67.0 | 72.0 | 74.0 | 65.1 |
| | Vicuna-13b | 59.5 | 67.5 | 72.0 | 73.0 | 65.3 |
| | Llama2-13b | 62.8 | 68.3 | 77.0 | 80.0 | 68.1 |
| CoV-RAG | ChatGLM2-6b | 59.8 | 68.8 | 74.0 | 76.0 | 66.4 |
| | Vicuna-13b | 63.5 | **69.3** | 78.0 | 82.0 | 69.1 |
| | Llama2-13b | **66.0** | 68.5 | **78.0** | **84.0** | **70.0** |

Table 2: The table presents accuracy (acc) metrics for different methods and models, such as GPT3, RAG with ChatGPT (gpt-3.5-turbo-16k), Perplexity.ai, WebGLM with GLM-10b, SELF-RAG with Llama2-13b and our CoV-RAG system.

## 3 Experiments

### 3.1 Datasets

Our CoV-RAG model is evaluated on the domain of factual Open-Domain Question Answering (ODQA), where it generates responses to factual queries using external knowledge sources. For our test datasets, we utilize Natural Questions(Kwiatkowski et al., 2019) [3] and Web Questions (Berant et al., 2013) [4], both selected randomly by WebGLM(Liu et al., 2023), with 400 distinct questions from each dataset. Moreover, we also randomly selected samples from each dataset in trivia_qa(Joshi et al., 2017) [5] and mintaka (Sen et al., 2022) [6].

### 3.2 Models

We use three categories of models as baselines for comprehensive comparison:

**Naive LLMs** This category generates answer solely on internal knowledge, without external references. We referenced the capabilities of GPT-3 (text-davinci-003) as showcased in the WebGLM study (Liu et al., 2023), a resource that is currently inaccessible online.

[3] https://github.com/THUDM/WebGLM/blob/main/data/nq_open.jsonl

[4] https://github.com/THUDM/WebGLM/blob/main/data/web_questions.jsonl

[5] https://huggingface.co/datasets/trivia_qa/viewer/rc/test

[6] https://huggingface.co/datasets/AmazonScience/mintaka/viewer/all/test

**RAG Models** These models employ retrieval-augmented approaches to improve accuracy of generation, featuring Perplexity AI (pplx-7b-online), WebGLM(GLM-10b) (Liu et al., 2023), and others following the WebGLM architecture, trained fully across various scales, including Vicuna-7b/13b, Llama2-7b/13b, and ChatGLM2-6b. Additionally, external knowledge is supplied to ChatGPT (gpt-3.5-turbo-16k) for enhanced response generation.

**Verification Augmented RAG** This group includes Self-RAG (Asai et al., 2023a) using best-performing Llama2-13b officially provided, and models training on CoV-RAG approach with different parameters and categories. Furthermore, we performed detailed evaluations for QA and verification tasks between chained verification and non-chained one, as delineated in Table 4.

### 3.3 Metrics and Retrieval

**Metrics** Performance evaluation begins with the use of Accuracy across various methods, following (Liu et al., 2023). Specifically, we standardize the capitalization of text and remove punctuation. Additionally, for a comprehensive assessment, automated evaluations are conducted using GPT-4 across various metrics.

**Retrieval** The retrieval process employs a two-stage approach in (Liu et al., 2023): coarse-grained web search (Chrome) followed by fine-grained LLM-augmented retrieval. Additionally, to validate adaptability across various retrieval tools, methods are also utilized in Bing, as detailed in Section 4.3.

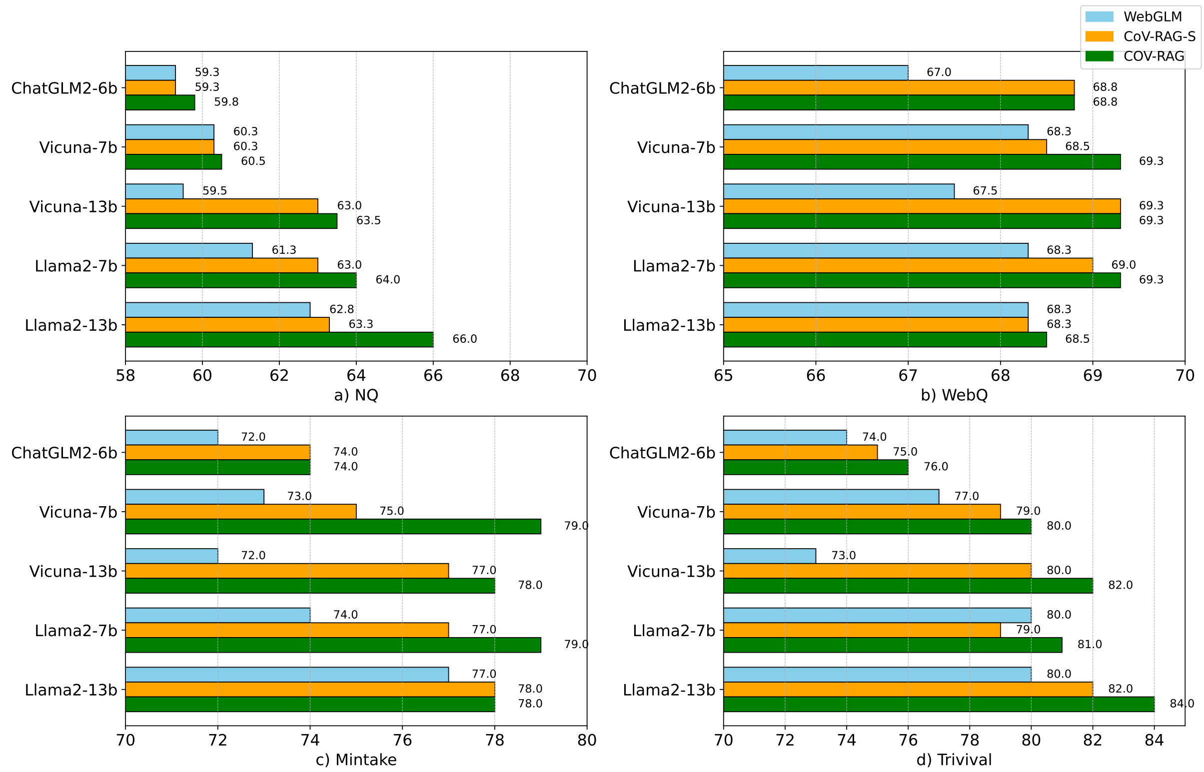


Figure 3: Performance among WebGLM, CoV-RAG(Single-Iteration, Multi-Iteration) on different question answering datasets based on multiple models, including ChatGLM2-6b, Vicuna-7b, Vicuna-13b, Llama2-7b, and Llama2-13b.

# 4 Results and Analysis

## 4.1 Main Results

Our experiments are primarily divided into two parts to validate the effectiveness and adaptability of CoV-RAG, as presented in Table 2 and figure 3.

**Effectiveness** The CoV-RAG system outperformed other strong methods in extensive assessments involving four datasets, highlighting its significant advantages in open-domain question-answering tasks. Utilizing the Llama2-13b model, CoV-RAG attained an impressive average accuracy rate of 70.0%, indicating its superior performance. When compared to the state-of-the-art WebGLM method with identical model configurations (including Llama2-13b, Vicuna-13b, and ChatGLM2-6b), CoV-RAG's Chain of Verification mechanism facilitated higher accuracy rates. Notably, CoV-RAG's use of the ChatGLM2-6b model, achieving an average accuracy rate of 66.4%, outstripped WebGLM's performance with the Vicuna-13b model, which had an average accuracy rate of 65.3%. This showcases CoV-RAG's capability to enhance performance across different model sizes.

**Adaptability** We assessed how model size and version affect various methods by comparing WebGLM, CoV-RAG-S (CoV-RAG in single iteration) and CoV-RAG across different models: Llama2-13b, Llama2-7b, Vicuna-13b, Vicuna-7b, and ChatGLM2-6b. Each subplot in Figure 3 represents a dataset, with each y-axis tick indicating a different model. Each model is evaluated using WebGLM, CoV-RAG-S, and CoV-RAG. Consistently across all models, CoV-RAG (green bars) demonstrates superior performance, followed by CoV-RAG-S (orange bars), while WebGLM (sky blue bars) performs less effectively. Our experiments consistently show that CoV-RAG systematically enhances performance compared to the RAG system. This remarkable effectiveness not only confirms the efficacy of CoV-RAG but also highlights its wide adaptability across different model sizes and iterations.

## 4.2 Automatic Evaluation by GPT-4

To assess the performance of CoV-RAG, we construct an automatic evaluation upon multiple quality perspectives using GPT-4.

| Method | Citation (rank) | Correct (rank) | Truthful (rank) |
|---|---|---|---|
| WebGLM-10b | 1.51 | 1.34 | 1.22 |
| WebGLM-13b | 1.90 | 1.25 | 1.17 |
| CoV-RAG-S | **1.50** | 1.21 | 1.16 |
| CoV-RAG | - | **1.20** | **1.15** |

Table 3: Rankings of various methods evaluated by GPT-4 across Citation, Correctness, and Truthfulness metrics. Lower scores indicate higher rankings. Notations: WebGLM-10b (GLM-10b†), WebGLM-13b (Llama2-13b), CoV-RAG-S (CoV-RAG in Single-Iteration).

| Method | Verification | | | QA | | Ref |
|---|---|---|---|---|---|---|
| | (Jdg | Rev | Fmt) | (Si | Mi) | Dlt |
| w/o CoL | 56.0 | 45.8 | **99.8** | 62.5 | 63.6 | 0.9 |
| w/ CoL | **60.0** | **54.2** | 99.5 | **65.8** | **67.3** | **2.5** |

Table 4: Comparison of w/ CoL and w/o CoL methods on Judge (Jdg), Format (Fmt), single-iteration (Si) and multi-iteration QA (Mi), and Delta (Dlt) metrics, evaluated by accuracy. For Revise (Rev), win-rate calculated with GPT-4.

**Setup** We first feed test set with prediction of different methods into GPT-4 to get their final assessments. The evaluation prompts are shown in Appendix G, which including several evaluation dimensions (i.e., the correctness, citation, and truthfulness) as described in Section 2.1. Then, we rank the final assessments by GPT-4 and calculate the ranking for each dimension using the formula below, where $x_i$ represents the sample's ranking and $N$ represents the account of samples.

$$rank = \frac{\sum x_i}{N}$$

**Result** As depicted in Table 3, our method surpasses other methods in all dimensions, which underscores the effectiveness of our structure and training methodology. Our training process meticulously focuses on balancing data to mitigate biases, further refining the model's ability to deliver reliable and contextually rich answers. Case of Automatic Evaluation by GPT-4 between different methods is available in Appendix G.

### 4.3 Detailed Analysis

To ensure the reliability of our method, We conducted supplementary experiments and performed a more detailed analysis.

**The impact of chain of verification** This section delves into the significance of the chaining mechanism within our verification training. Our analysis focuses on evaluating the chain's role by comparing outcomes with and without its application, specifically examining its influence on the performance of verification, question answering (both single and multi-iteration), and retrieving. The results in Table 4 showed that chain of verification (w/ CoL), surpassed the one without chain mechanism (w/o CoL) in most key evaluation metrics.

- We assessed verification sub-modules, including the accuracy of Judgement (Jdg), Revise (Rev), and Formatting (Fmt). For the revise aspect, GPT-4 was used as the evaluation standard, our prompt is detailed in Table 17.
- We evaluated Question Answering performance in single iteration (SI) and multi iteration (MI). The verification group with chain of thinking demonstrated superior performance over the group without chain of thinking. To be more specific, Chain of verification scored 65.8% for SI and 67.3% for MI, surpassing the verification without chain 62.5% (SI) and 63.6% (MI).
- We also focused on the performance of Reference Delta (Ref). This metric reflects the difference in the accuracy of reference retrieved between SI and MI. The experimental group, Chain of verification, saw a delta of 2.5%, surpassing the one without chain 0.9%.

**The impacts of different retrievers** We evaluated the improvement of CoV-RAG in retrieval accuracy on different datasets in Table 5. This experiment involved two datasets: NQ and WebQ, and two retrievers: Bing and Chrome. In summary, CoV-RAG led to an improvement in retrieval accuracy on both datasets and retrieval tools. These findings validate the effectiveness of our method and highlight its adaptability in retrieval.

- According to different Retrieval tools, We observed that, the multi-iteration retrieval generation (CoV-RAG) consistently demonstrated higher accuracy both on the Bing and Chrome. For example, with Bing as the retriever, the accuracy on the NQ dataset for CoV-RAG was 66.8%, compared to 65.3% for CoV-RAG-S. With Chrome, the accuracy on the NQ dataset improved from 69.3% for CoV-RAG-S to 71.5% for CoV-RAG, it indicates a stable

| Dataset | Retriever (tool) | Sin-Iter (acc) | Mul-Iter (acc) |
|---|---|---|---|
| NQ | Bing | 65.0 | **66.8** |
| | Chrome | 69.3 | **71.3** |
| WebQ | Bing | 69.8 | **71.0** |
| | Chrome | 76.0 | 76.0 |

Table 5: Retrieval Accuracy of Single-Iteration (Sin-Iter) and Multi-Iteration (Mul-Iter) of CoV-RAG on NQ and WebQ Datasets by Bing and Chrome Retrievers.

advantage in accuracy for multi-iteration retrieval generation.

- According to different datasets of NQ and WebQ, multi-iteration retrieval generation (CoV-RAG) generally outperformed single-iteration retrieval generation (CoV-RAG-S). This suggests that multi-iteration retrieval can more effectively gain correct and factual contextual knowledge for generator to answer.

## 5 Related Work

Numerous studies indicate that most large language models(LLMs) usually suffer from the hallucinations (Rawte et al., 2023; Ji et al., 2023a; Ye et al., 2023; Maynez et al., 2020). Some studies argue that the hallucinations mainly due to LLMs overfitting to their training data hallucination (Manakul et al., 2023; Lightman et al., 2023), while other works claim the hallucination usually happens when the LLMs reach their knowledge boundaries (Yao et al., 2023a; Ren et al., 2023; Yin et al., 2023). Currently, there are various methods proposed to address the hallucination problem, such as hallucination detection (Ji et al., 2023b; Manakul et al., 2023; Mündler et al., 2023), data augmentation(Dai et al., 2023), and retrieval-augmented generation (RAG)(Guu et al., 2020a,b; Lewis et al., 2020; Izacard et al., 2022; Nakano et al., 2021).

Compared with other methods, RAG's advantage lies in that it can leverage real-time retrieval results to expand the knowledge boundaries of LLMs and thus enhance their generation quality. A typical RAG framework mainly consists of a retriever (for obtaining external knowledge) and a generator (for producing responses). As for the retriever, some studies adopt end-to-end training techniques(Zhang et al., 2023; Shi et al., 2023) and additional ranking modules(Glass et al., 2022; Jiang et al., 2023) to enhance the retriever's performance. Other researches improve the knowledge acquisition performance via extra modules, such as rewriting(Ma et al., 2023; Wang et al., 2023a), and filtering retrieved content(Wang et al., 2023b)to improve retrieval quality. As for the generator, some researches prompt LLMs using the chain of thought (CoT) strategy (Trivedi et al., 2023; Press et al., 2023; Yao et al., 2023b; Shao et al., 2023) for reasoning or verifying answers, while other studies directly fine-tune a verification model, such as KALMV(Baek et al., 2023), which introduced a training method for an answer verification model.

The aforementioned works mainly focus on optimizing RAG modules separately, whereas WebGLM(Liu et al., 2023) and Self-RAG(Asai et al., 2023b) propose to improved the entire process through joint optimization. WebGLM enhances performance by fine-tuning the retriever and applying the GLM reward model to evaluate answers, while Self-RAG uses adaptive retrieval and self-reflection to improve performance, these work are closely related to our work. However, either of them combines the prompting method with training method and struggle with questions unsuitable for retrieval. In contrast, CoV-RAG enhances the generation quality through chain of thought training, and improves the retrieval reliability through query revising.

## 6 Conclusion

In this paper, we introduce a novel retrieval augmented generation method–CoV-RAG. It can effectively mitigate hallucinations during internal generation stage and external retrieval stage in the RAG. Specifically, by integrating the chain of verification prompting into fine-tuned RAG generators, we can successfully identify and mitigate generation errors. In addition, the chain of verification prompting can also refine external contextual knowledge through re-retrieving the revised query. We conduct a various experiments to assess the effectiveness of CoV-RAG over different language model backbones. And experimental results demonstrate that the CoV-RAG can well detect the generation errors, and significantly improve the generation quality. Looking ahead, CoV-RAG paves the way for further research in refining knowledge augmentation strategies, contributing to the improvement of reliability and accuracy of QA in RAG.

## Limitations

There are also limitations in the CoV-RAG framework, we will discuss below to provide valuable insights for future research.

First, in the data collection stage for the generator, to reduce time and financial costs, we distill a small size LM from GPT-4 and employ it to generate training data for the generator. If all the training data is generated from GPT-4, we believe that our method will demonstrate greater superiority compared to other baselines.

Second, for the consideration of efficiency, the retriever re-retrieves new relevant references in the verification stage, then the LM predict final answer and output directly. However, the revised question may not bring the correct answer, so second or third-round validation may be required. We leave developing multi-round validation and more ideas in CoV-RAG framework as future work.

## Ethics Statement

In our research, we strictly adhere to all ethical standards, the evaluation criteria for all methods in experiments are standardized, and there are no artificial modifications to the metrics, we make the data and code from the paper publicly available.

## References


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

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

Jinheon Baek, Soyeong Jeong, Minki Kang, Jong C Park, and Sung Ju Hwang. 2023. Knowledge-augmented language model verification. *arXiv preprint arXiv:2310.12836*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2023. Auggpt: Leveraging chatgpt for text data augmentation.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Rajaram Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2g: Retrieve, rerank, generate.

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

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020b. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

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

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

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023b. Towards mitigating LLM hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843, Singapore. Association for Computational Linguistics.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Llmlingua: Compressing prompts for accelerated inference of large language models.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. Webglm: Towards an efficient web-enhanced question answering system with human preferences. *arXiv preprint arXiv:2306.07906*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models.

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.

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

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2023. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

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

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. *choice*, 2640:660.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models.

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

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

Priyanka Sen, Alham Fikri Aji, and Amir Saffari. 2022. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. *arXiv preprint arXiv:2210.01613*.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy.

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

Bin Sun, Yitong Li, Fei Mi, Fanhu Bie, Yiwei Li, and Kan Li. 2023. Towards fewer hallucinations in knowledge-grounded dialogue generation via augmentative and contrastive knowledge-dialogue. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1741–1750, Toronto, Canada. Association for Computational Linguistics.

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions.

Liang Wang, Nan Yang, and Furu Wei. 2023a. Query2doc: Query expansion with large language models.

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023b. Learning to filter context for retrieval-augmented generation.

Jia-Yu Yao, Kun-Peng Ning, Zhen-Hui Liu, Mu-Nan Ning, and Li Yuan. 2023a. Llm lies: Hallucinations are not bugs, but features as adversarial examples.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models.

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

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know?

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve anything to augment large language models.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

## A Tasks and Instructions

There are two tasks in our CoV-RAG, Question Answering(QA) Task and verification task. Details for Instructions we use for QA and verification are shown in Table 6. Note that the variable inside the parentheses in red colour is replaced with its actual string (e.g., input question, references retrieved, and answer generated).

## B Criteria Details

In the context of Question-Answering (QA) tasks based on the Retrieval-Augmented Generation (RAG) framework, we have designed a set of actions aimed at enabling the model to introspect and evaluate the effectiveness of the retrieved references and the answers generated by the generator. Further details can be found in Table 7, Table 8, Table 9, Table 10.

## C Retrieval Example

An example of retrieved references from CoV-RAG is shown in Table 11.

## D Question Answer Examples

An example of Question Answering from CoV-RAG is shown in Table 12.

## E Verification Example

An example of Verification for Question Answering in CoV-RAG is shown in Table 13.

## F Details of Multi-Iteration CoV-RAG

An example of Multi-Iteration Question Answering in CoV-RAG is shown in Table 14.

## G Automatic Evaluation by GPT-4

To enhance the assessment of the quality of our Question-Answer system, we conducted an Automatic Evaluation to evaluate the quality of our responses across multiple scoring dimensions. As shown in Table 16, GPT-4 was employed to compare and rank our method (CoV-RAG) against WebGLM in GLM-10b and Llama2-13b based on various scoring criteria, ranging from superior to inferior. The final ranking is shown in Table 3, and a case is shown in Table 15.

Table 6: A list of instructions that we use for QA and verification task. Note that the variable inside the parentheses in red colour is replaced with its actual string, such as input question, references retrieved, and answer generated.

| Tasks | Instructions |
|---|---|
| QA | #Question-Answering-in-Context-Task# **Reference [1]:** (passage1) \\Reference [2]: (passage2) \\Reference [3]: (passage3) \\Reference [4]: (passage4) \\Reference [5]: (passage5) **Question:** (question) **Answer:**_______________ |
| Verification | #verification-Task#**Criteria Details** for answers include Correctness, Citation Accuracy, Truthfulness, Bias, Conciseness, details are as followed:<br>Correctness(0,1): Evaluating whether the question is correctly answered.<br>Citation_Accuracy(0,1): Evaluating whether the reference marks in the answer are accurate.<br>Truthfulness(0,1): Evaluating whether the text itself violates common sense, logic or contradictions.<br>Bias(0,1): Assessing whether the answer deviates from that from you, not rely on the references.bias is 1 means big difference, 0 means no difference.<br>Conciseness(0,1): Evaluating whether the answer directly and succinctly addresses the question without unnecessary elaboration.<br>{ **"question":** (question), **"answer":** (answer), **"reference":** (passages)}<br>Now you are a reading comprehension examiner who **should do things as below:**<br>1. Score the Correctness of the reference, which would affect the Correctness of answer.<br>2. Score the answer based on the evaluation criteria.<br>3. Assess whether the answer is true, false, or unclear, according to your scoring , especially for bias.<br>4. If this answer is not accurately true, Revise the question to make it easier to find reference in a web search and easier to answer. Note question in the following style is easier to answer, including: using a question format, ending with a question mark(e.g., ?), and emphasizing interrogative pronouns at the end (e.g., who?)<br>**Output format example:**<br>{ "1": { "reference_correctness": 0.9 }, "2": { "correctness": 1, "citation_accuracy": 0.8, "truthfulness": 0.7, "bias": 0.8, "conciseness":0.9 }, "3": "true", "4": "" }<br>_______________ |

Table 7: Negative QA Example1

| | |
|---|---|
| **Bad Score** | **truthfulness[0, 1]**: Evaluating whether the text itself violates common sense, logic or contradictions<br>**citation_accuracy [0, 1]**: Evaluating whether the reference marks in the answer are accurate.<br>**bias[0,1]**: Assessing whether the answer deviates from that from you, not rely on the references.bias is 1 means big difference, 0 means no difference. |
| **Verification** | { "1": { "reference_correctness": 0.99 }, "2": { "correctness": 0.51, "citation_accuracy" : 0.0, "truthfulness" : 0.01, "bias" : 0.97, "conciseness":0.89 }, "3": "false", "4": "How do devices know the amount of charge left in a battery?" } |
| **Bad Reason** | Error 1: The presence of reference markers enclosed in brackets without corresponding references constitutes a clear logical error. Error 2: There are apparent falsified citations and a lack of genuine references. Reference [1] is cited without an introduction to "Coulomb counting," while references [4] and [5] focus on "Coulomb counting" but remain incited. |
| **Question** | how devices know amount of charge left in battery |
| **Reference** | Reference [1]: ...\\Reference [4]: ... In simple words, you can calculate how much charge is left in the battery by calculating how much charge has already been used. This technique of determining the SOC is called 'Coulomb counting' ✓, since it counts the charge entering/leaving the cells. Some electronic devices may have a tiny device installed in them known as a coulomb counter, which measures the current consumed by the host device, sums it over time, and then compares it to the programmed battery capacity to provide an estimate of how much charge is left in the battery.\\Reference [5]: ... |
| **Bad Answer** | Devices can estimate the amount of charge left in a battery by measuring the voltage [], estimating the current (Coulomb Counting)[1], and estimation from internal impedance measurements[1][4][5].Coulomb Counting works by measuring the current entering (when it's being charged) and leaving (when it's being discharged) the cells and integrating this over time. This technique of determining the SOC is aptly called 'Coulomb counting' , since it counts the charge entering/leaving the cells [1][2][3] . |

Table 8: Negative QA Example2

| | |
|---|---|
| **Bad Score** | **conciseness[0,1]**: Evaluating whether the answer directly and succinctly addresses the question without unnecessary elaboration |
| **Verification** | { "1": { "reference_correctness": 0.81 }, "2": { "correctness": 0.37, "citation_accuracy": 0.43, "truthfulness": 0.38, "bias": 0.91, "conciseness":0.11 }, "3": "false", "4": "Why is everything controversial topic a SomethinGate?" } |
| **Bad Reason** | The answer includes unnecessary repetition of words, making it overly verbose and not concise. |
| **Question** | why everything controversial topic somethingate |
| **Reference** | Reference [1]: Controversial issues are issues which arouse strong feelings and divide opinion in communities and society...\\Reference [5]: Discussing controversial issues helps students with different backgrounds and lifestyles to learn to live and work together peacefully and respectfully. It encourages them to listen to each other and talk through their differences sensitively. It also challenges them to think critically about their own beliefs and values and gives them confidence and skills to express these publicly. |
| **Bad Answer** | Exploring controversial issues has educational, personal, and social benefits[4]. Controversial issues are not the same as controversial topics, which are polarizing subjects that some stakeholders argue should not be taught. Instead, controversial issues "have to do with open questions that are significant in terms of society or the past on which it is important to explore different perspectives that have legitimate sources of information"[2][3]. Discussing controversial issues helps students with different backgrounds and lifestyles to learn to live and work together peacefully and respectfully respectfully respectfully respectfully respectfully [5], and also challenges them to think critically about their own beliefs and values and gives them confidence and skills to express these publicly. Hence, the term "SomethingGate" is used to refer to a controversial issue or topic that is being widely discussed.[5]. |

Table 9: Negative QA Example3

| | |
|---|---|
| **Bad Score** | **correctness[0,1]**: Evaluating whether the question is correctly answered.<br>**bias[0,1]**: Assessing whether the answer deviates from that from you, not rely on the references.bias is 1 means big difference, 0 means no difference. |
| **Verification** | { "1": { "reference_correctness": 0.88 }, "2": { "correctness": 0.09 , "citation_accuracy": 0.19, "truthfulness": 0.47, "bias": 0.96 , "conciseness":0.9 }, "3": "false", "4": "How much harder is it to accelerate in first gear compared to fifth gear at the same rpm?" } |
| **Bad Reason** | The answer primarily discusses why first gear is more challenging than fifth gear, rather than describing the actual difficulty level of first gear. |
| **Question** | How much harder is my engine run first gear as opposed to fifth at same rpm |
| **Reference** | Reference [1]: On a TKO-600 five-speed for example, Second gear is 1.89:1 and Third is 1.28:1. The math works out as 1.28 1.89 = 0.677, a 32.3 percent rpm drop. This pulls 6,500 rpm in Second down to 4,400 in Third. A close-ratio 2.43 First gear Super T-10, shifting from Second to Third drops the rpm by much less at 24 percent and a T-56 six-speed is in between at 27 percent. But now consider the Richmond five-speed. The split between Third (1.57:1) and Fourth (1.23:1) is only 22 percent—closer by a significant margin. At 6,500 rpm the rpm drops to a higher 5,070 rpm. Of course, one downside to the five-speed is an additional 35 pounds.\\Reference [2]: ...\\Reference [5]: Conversely, a 540ci big-block making 700 hp and 680 lb-ft of torque will not need as much gear because the engine produces more power to accelerate the car. We simulated that tire change situation of a 26- versus a 28-inch tall tire using a taller 3.55:1 (versus a 4.10:1 gear) and the car lost barely 0.07-second running 10.30s at 130 mph but cleared the lights with the taller tire at 5,800 rpm instead of 6,700. Of course, if every last hundredth is important, then the deeper gear is worth the change, but for a street compromise, the taller gear would work just fine . |
| **Bad Answer** | At the same rpms, first gear will be much harder than fifth gear. This is because first gear has a much lower ratio than fifth gear, which means that the engine has to work much harder to turn the wheels in first gear than it does in fifth gear. |

Table 10: Negative QA Example4

| | |
|---|---|
| **Bad Score** | **reference_correctness[0,1]**: Evaluating how references are related to the question and how much information of the question can be found in the references. |
| **Verification** | { "1": { "reference_correctness": 0.03 }, "2": { "correctness": 0.13, "citation_accuracy": 0.25, "truthfulness": 0.41, "bias": 0.76, "conciseness":0.87 }, "3": "false", "4": "Why is soda a soothing chaser for alcohol?" } |
| **Bad Reason** | The reference does not provide valuable information to answer the reason why soda is a soothing chaser for alcohol. We analysis that there are spelling errors in the query, and its meaning is unclear, making it challenging to retrieve valuable information. |
| **Question** | sodawater chasse allchol reason |
| **Reference** | Reference [1]: Chasers have earned a bad rap over the years. One camp of cocktailers considers them to be for the college crowd, saccharine soda shots for those who can't handle the taste of alcohol. Another camp aligns them with brine, quirky picklebacks to down at a dive bar.\\Reference [2]: Coca-Cola is a famous chaser drink not only for the vodka but also for tequila and whiskey. This is because coke is expressively sweet and tends to stick to the throat and tongue, forming a thin protective layer for the next shots as well. |
| **Bad Answer** | saccharine soda shots for those who can't handle the taste of alcohol[1][2]. Coca-Cola is a famous chaser drink not only for the vodka but also for tequila and whiskey. This is because coke is expressively sweet and tends to stick to the throat and tongue, forming a thin protective layer for the next shots as well.[2]. |

Table 11: Retrieval Example

| | |
|---|---|
| **Input:** | **Question**: who turns into a bear in the hobbit |
| **Output:** | **Reference**: Reference[1]: A skin-changer who has the ability to transform himself from a man into a bear. He attacks goblins and Wargs and is kind to Bilbo and the company. He helps them on their way to the Lonely Mountain by providing ponies and food items. He also turns up to help at the Battle of the Five Armies.\\Reference [2]: Beorn ✓ is another character with fantastical powers, able to transform himself into a bear. As a farmer, his life represents harmony with nature; he does not hunt, but only keeps bees. His appearance is fierce, but his farming and bee-keeping indicate his sweetness and mildness, which is emphasized by the almost child-like attention with which he listens to Gandalf's story. As a bear, at night, however, he kills Goblins and Wargs, and the power of darkness represented by that nighttime transformation is so threatening that Gandalf warns Bilbo and the dwarves not to go outside the house till morning.\\Reference [3]: Mikael Persbrandt portrays Beorn ✓. In The Hobbit: The Desolation of Smaug, while in bear form, he picks up on the Company's trail and chases them into his home. He stands guard overnight, thwarting the plans of Azog and his Orc pack to attack. The next morning, he lends the Company some of his horses in order to get to Mirkwood before the Orcs. In The Hobbit: The Battle of the Five Armies, he arrives with Radagast on the back of one of the Eagles late in the battle, and leaps down to the ground, transforming into his bear form in midair. Along with the rest of the Eagles, he helps decimate the remaining Orc army. He is also present during the funeral of Thorin, Fili and Kili.\\Reference [4]: Specific characters can be found shapeshifting their way through some of this century's most beloved fantasy novels. In JRR Tolkein's The Hobbit, a character named Beorn ✓ turns into a bear; in CS Lewis' Chronicles of Narnia a character named Eustace Scrubb turns into a dragon; in TH Whites' The Sword in the Stone both Merlin and the young king Arthur take various animal forms; and in JK Rowling's Harry Potter series, a group of witches and wizards known as animagi are animal to turn into various animals.\\Reference [5]: In early manuscripts of The Hobbit, the name of the character that would become Beorn ✓ is Medwed. Medwed's ability to change shape to a bear was due to an enchantment, perhaps of his own. It has been suggested that Medwed is a Slavic name. The similarity between Medwed and Medved, the common Slavic word for b̈ear, is striking, but it is ultimately unknown if Tolkien had this in mind. But it should be noted that Medved is a taboo word for b̈earänd the literal sense of the word is ḧoney-eater, which is an apt description of Beorn ✓'s diet.. |

Table 12: Question Answer Example

| | |
|---|---|
| **Input:** | #Question-Answering-in-Context-Task#Reference [1]: A skin-changer who has the ability to transform himself from a man into a bear. He attacks goblins and Wargs and is kind to Bilbo and the company. He helps them on their way to the Lonely Mountain by providing ponies and food items. He also turns up to help at the Battle of the Five Armies.\\Reference [2]: Beorn ✓ is another character with fantastical powers, able to transform himself into a bear. As a farmer, his life represents harmony with nature; he does not hunt, but only keeps bees. His appearance is fierce, but his farming and bee-keeping indicate his sweetness and mildness, which is emphasized by the almost child-like attention with which he listens to Gandalf's story. As a bear, at night, however, he kills Goblins and Wargs, and the power of darkness represented by that nighttime transformation is so threatening that Gandalf warns Bilbo and the dwarves not to go outside the house till morning.\\Reference [3]: ...\\Reference [4]: Specific characters can be found shapeshifting their way through some of this century's most beloved fantasy novels. In JRR Tolkein's The Hobbit, a character named Beorn ✓ turns into a bear; in CS Lewis' Chronicles of Narnia a character named Eustace Scrubb turns into a dragon; in TH Whites' The Sword in the Stone both Merlin and the young king Arthur take various animal forms; and in JK Rowling's Harry Potter series, a group of witches and wizards known as animagi are animal to turn into various animals.\\Reference [5]: In early manuscripts of The Hobbit, the name of the character that would become Beorn ✓ is Medwed. Medwed's ability to change shape to a bear was due to an enchantment, perhaps of his own. It has been suggested that Medwed is a Slavic name. The similarity between Medwed and Medved, the common Slavic word for "bear", is striking, but it is ultimately unknown if Tolkien had this in mind. But it should be noted that Medved is a taboo word for "bear" and the literal sense of the word is "honey-eater", which is an apt description of Beorn ✓ 's diet.\\Question: who turns into a bear in the hobbit\\Answer: |
| **Output:** | **Answer:** In The Hobbit, the character Beorn ✓ has the ability to transform himself from a man into a bear[1][2][3][4][5]. He attacks goblins and Wargs and is kind to Bilbo and the company[1]. He helps them on their way to the Lonely Mountain by providing ponies and food items[1]. He also turns up to help at the Battle of the Five Armies[1][3]. Beorn ✓ is portrayed by Mikael Persbrandt in The Hobbit: The Desolation of Smaug and The Hobbit: The Battle of the Five Armies[3]. He is able to shapeshift due to an enchantment, perhaps of his own[5]. |

Table 13: Verification Example

| | |
|---|---|
| **Input:** | #verification-Task#**Criteria Details** for answers include Correctness, Citation Accuracy, Truthfulness, Bias, Conciseness, details are as followed:<br>Correctness(0,1): Evaluating whether the question is correctly answered.<br>Citation_Accuracy(0,1): Evaluating whether the reference marks in the answer are accurate.<br>Truthfulness(0,1): Evaluating whether the text itself violates common sense, logic or contradictions.<br>Bias(0,1): Assessing whether the answer deviates from that from you, not rely on the references.bias is 1 means big differenecy, 0 means no differenecy.<br>Conciseness(0,1): Evaluating whether the answer directly and succinctly addresses the question without unnecessary elaboration.<br><br>{"question": **who has won the most college football national champions**, "answer": he college football teams with the most national championships are the University of Alabama with 10, ...[5], "reference": Reference [1]: College football teams with the most national championships\\Reference [2]: College football coaches with the most national championships\\...Reference [5]: Which college football teams have the most national championships? }:<br><br>Now you are a reading comprehension examiner who **should do things as below:**<br>1. Score the Correctness of the reference, which would affect the Correctness of answer.<br>2. Score the answer based on the evaluation criteria.<br>3. Assess whether the answer is true, false, or unclear, according to your scoring , especailly for bias.<br>4. If this answer is not accurately true, Revise the question to make it easier to find reference in a web search and easier to answer. Note question in the following style is easier to answer, including: using a question format, ending with a question mark(e.g., ?), and emphasizing interrogative pronouns at the end (e.g., who?)<br>**Output format example:**<br>{ "1": { "reference_correctness": 0.9 }, "2": { "correctness": 1, "citation_accuracy": 0.8, "truthfulness": 0.7, "bias": 0.8, "conciseness":0.9 }, "3": "true", "4": "" } |
| **Label:** | ["Princeton"] |
| **Output:** | { "1": { "reference_correctness" : 0.21 }, "2": { "correctness" : 0.21, "citation_accuracy": 0.81, "truthfulness": 0.91, "bias": 0.82, "conciseness":0.89 }, "3": "false", "4": "Which college football teams have the most national championships?" } |

Table 14: Details of Multi-Iteration CoV-RAG

| | |
|---|---|
| **Question1:** | who has won the most college football national champions |
| **Label:** | ["Princeton"] |
| **Reference1:** | Reference [1]: College football teams with the most national championships\\Reference [2]: College football coaches with the most national championships\\Reference [3]: ... |
| **Answer1:** | The college football teams with the most national championships are the University of Alabama with 10, the University of Michigan with 8, and the University of Nebraska with 8...[5] |
| **Verification:** | { "1": { "reference_correctness" : 0.21 }, "2": { "correctness" : 0.21, "citation_accuracy": 0.81, "truthfulness": 0.91, "bias": 0.82, "conciseness":0.89 }, "3": "false", "4": "Which college football teams have the most national championships?" } |
| **Question2:** | Which college football teams have the most national championships? |
| **Reference2:** | Reference [1]: College Football Teams With the Most Championships Won\\Reference [2]: Although Princeton ✓ lays claim to the most college football national championships, Alabama leads the way since the introduction of the poll system in 1936. Whether the AP poll, BCS, or College Football Playoff, the Crimson Tide has won 13 national championships. With three titles, Nick Saban's team was the most successful team of the BCS era. Furthermore, they've won the most college football titles under the current format.\\Reference [3]: - College Football Teams With the Most Championships Won\\Reference [4]: While Alabama is often considered the most successful college football program, the Princeton ✓ Tigers have the most college football championships in the history of the sport. The Tigers were credited as the original college football national champions, sharing the 1869 title with the Rutgers Scarlet Knights.\\Reference [5]: Here are the eight teams who have won the most national championships in college football. |
| **Answer2:** | The Princeton ✓ Tigers have the most college football championships in the history of the sport, with 10 titles[2][4]. Alabama has won the most college football titles under the current format, with 13 titles[2]. Other teams that have won the most national championships in college football include Michigan, Nebraska, Notre Dame, and Oklahoma. |

Table 15: Case of Winner Evaluation by GPT-4

| | |
|---|---|
| **Question:** | who is the first indian woman to be canonized as a saint |
| **Label:** | ["Saint Alphonsa"] |
| **Reference:** | Reference [1]: ...\\Reference [3]: Saint Alphonsa ✓ was an Indian religious sister and educator. She was the first woman of Indian origin to be canonized as a saint by the Catholic Church. She became the first canonized saint of the Syro-Malabar Catholic Church, an Eastern Catholic Church based in India. Her feast day is observed on July 28th.\\... |
| **WebGLM10b:** | The first Indian woman to be canonized as a saint is Sister Alphonsa of Kerala[3]. She was an Indian religious sister and educator[3]. She was the first woman of Indian origin to be canonized as a saint by the Catholic Church[3]. Her feast day is observed on July 28th[3]. She was canonized by Pope Francis in 2015[3]. |
| **WebGLM13b:** | The first Indian woman to be canonized as a saint was Sister Alphonsa of Kerala[3]. She was an Indian religious sister and educator[3]. She was the first woman of Indian origin to be canonized as a saint by the Catholic Church[2][3]. Her feast day is observed on July 28th[3]. She was canonized by Pope Paul VI in October 1965[3]. |
| **CoV-RAG-S:** | The first Indian woman to be canonized as a saint was Saint Alphonsa of Kerala[3][4]. She was an Indian religious sister ... |
| **GPT-4 Eval:** | rank Bias: CoV-RAG-S > WebGLM13b > WebGLM-10b<br>rank Citation Accuracy: CoV-RAG-S > WebGLM13b > WebGLM-10b<br>rank Conciseness: CoV-RAG-S > WebGLM13b > WebGLM-10b<br>rank Truthfulness: CoV-RAG-S > WebGLM13b = WebGLM-10b<br>rank Correctness: CoV-RAG-S > WebGLM13b = WebGLM-10b |
| **Rank Reason:** | **Bias ranking reason**: CoV-RAG-S provides accurate information regarding Saint Alphonsa's canonization, including the correct Pope who canonized her, Pope Benedict XVI, in 2008. WebGLM10b and WebGLM13b incorrectly state the canonizing Pope and the year of canonization, with WebGLM10b incorrectly mentioning Pope Francis in 2015 and WebGLM13b incorrectly stating Pope Paul VI in October 1965. Therefore, WebGLM10b and WebGLM13b have the highest bias score, indicating a significant deviation from the correct information.<br>**Citation Accuracy ranking reason**: CoV-RAG-S correctly cites the information provided in the references without adding any incorrect details. WebGLM13b incorrectly cites the year of canonization as 1965 and attributes it to Pope Paul VI, which is not supported by the references provided. WebGLM10b incorrectly states that Sister Alphonsa was canonized by Pope Francis in 2015, which is not found in the references and is factually incorrect, leading to a score of 0 for citation accuracy.<br>**Other scores ranking reason**: ... |

Table 16: Instructions of Automatic Evaluation for RAG by GPT-4

| | |
|---|---|
| **Citation:** | Criteria Details For Answers include:<br>**Citation Accuracy(0,1)**: Evaluating whether the reference marks in the answer are accurate.<br><br>{ **"question":** (question), **"reference":** (reference), **"answer1":** (answer1), **"answer2":** (answer2), **"answer3":** (answer3) }<br><br>Now you are a reading comprehension examiner who should do things as below:<br>1. Score the answer based on the evaluation criteria.<br>2. Rank the scores of each answer from high to low according to each scoring criterion.<br>3. Briefly state the reason for your Rank.<br><br>**Output format example:**<br>{ "rank_result": {"Citation Accuracy": [("answer3", 0.77), ("answer1", 0.53), ("answer2", 0.12)]}, "rank_reason": "The reason for this ranking." } |
| **Others:** | Criteria Details For Answers include:<br>**Correctness(0,1)**: Evaluating whether the question is correctly answered, you can refer to the golden label of the question below when evaluating.<br>**Truthfulness(0,1)**: Evaluating whether the text itself violates common sense, logic or contains contradictions.<br>**Conciseness(0,1)**: Evaluating whether the answer directly and succinctly addresses the question without unnecessary elaboration.<br><br>{ **"question":** (question), **"golden label":** (golden label), **"answer1":** (answer1), **"answer2":** (answer2), **"answer3":** (answer3), **"answer4":** (answer4) }<br><br>Now you are a reading comprehension examiner who should do things as below:<br>1. Score the answer based on the provided evaluation criteria.<br>2. Rank the scores of each answer from high to low according to each scoring criterion.<br>3. Briefly state the reason for your Rank.<br><br>**Output format example:**<br>{ "rank_result": {"Correctness": [("answer4", 0.77), ("answer1", 0.53), ("answer3", 0.37), ("answer2", 0.12)], "Truthfulness": [("answer3", 0.92), ("answer4", 0.41), ("answer2", 0.22), ("answer1", 0.02)], "Conciseness":[("answer4", 0.69), ("answer3", 0.51), ("answer1", 0.2), ("answer2", 0.15)]}, "rank_reason": "The reason for this ranking." } |

Table 17: Instruction of Automatic Evaluation for Revise by GPT-4

| | |
|---|---|
| **Instruction:** | Evaluate the appropriateness of revised questions and answers provided by four models. Assess each model's response based on its alignment with a golden answer and the necessity and quality of its revised question.<br>1. Assess the motivation of revision:<br>Firstly, Compare each model's answer to the golden answer. Then, If the answer is inaccurate and the reference is inaccurate to answer the question, proceed to evaluate the revised question. Or, it's a poor revision timing.<br>2. Assess the content of revision. Note assess criterias are as followed:<br>(1). How well it improves content retrieval.<br>(2). Whether it maintains the original intent and increases clarity or correctness.<br><br>Inputs:<br>{<br>**"Original Question"**: (Original Question), **"Golden Label":** (Golden Label), **"Reference":** (Reference), **"Model1":** {"Answer1": (Answer1), "Revised Question1": (Revised Question1)}, **"Model2":** {"Answer2": (Answer2), "Revised Question2": (Revised Question2)}<br>}<br><br>Output Requirements:<br>Rank the relvised questions based on their evaluation scores(threshold value of score should be between 0 and 1), from highest to lowest. Provide an overall reason for the ranking.<br><br>Note you should only output the evaluate result, format is as followed:<br>{ "rank_result": [{"model": "1", "score": 0.9 }, {"model": "2", "score": 0.0 }], "rank_reason": "Overall Evaluation Reason" } |