KDD Cup Meta CRAG 2024 Technical Report: Three-step Question-Answering Framework

Sungho Park
Graduate School of Artificial Intelligence, POSTECH
Pohang, South Korea
shpark@dblab.postech.ac.kr

Jeongeum Seok
Department of Computer Science and Engineering, POSTECH
Pohang, South Korea
jeseok@dblab.postech.ac.kr

Jooyoung Lee
Department of Computer Science and Engineering, POSTECH
Pohang, South Korea
jylee@dblab.postech.ac.kr

Joohyung Yun
Department of Computer Science and Engineering, POSTECH
Pohang, South Korea
jhyun@dblab.postech.ac.kr

Wonseok Lee
Department of Convergence IT Engineering, POSTECH
Pohang, South Korea
wslee@dblab.postech.ac.kr

Abstract

Large language models (LLMs) have shown significant capabilities in the question-answering task, but they often suffer from hallucination, where generated content deviates from real-world facts. Retrieval-augmented generation (RAG) has been proposed to address this issue, which enhances LLM performance by retrieving relevant information from external knowledge sources. KDD Cup Meta 2024 is a competition for advancing the practical application of RAG in real-world scenarios. Participants are asked to develop an innovative RAG system that can accurately and efficiently answer complex questions by integrating relevant external data while minimizing hallucination. Our team dRAGonRAnGers, composed of members from POSTECH Data Systems Lab, propose a methodology that addresses two primary challenges of RAG: reducing unnecessary retrievals and preventing the propagation of incorrect information. We enhance the standard RAG framework by incorporating the inherent knowledge of LLMs to avoid unnecessary retrievals and introducing a verification step to reassess generated answers. This approach optimizes the efficiency and reliability of QA systems, improving both response accuracy and computational efficiency. Our team is the first prize winner of the comparison question category for all three tasks and also the first prize winner of the post-processing category for task 1 in KDD Cup 2024.

CCS Concepts

• **Computing methodologies** → **Information extraction**; • **Information systems** → **Question answering**.

Keywords

Retrieval-Augmented Generation, Hallucination, External Knowledge Retrieval, Answer Confidence

KDD'24, August 25-29, 2024, Barcelona, Spain

ACM Reference Format:
Sungho Park, Jeongeum Seok, Jooyoung Lee, Joohyung Yun, and Wonseok Lee. 2024. KDD Cup Meta CRAG 2024 Technical Report: Three-step Question-Answering Framework. In *Proceedings of 2024 KDD Cup Workshop for Retrieval Augmented Generation (KDD'24).* ACM, New York, NY, USA, 8 pages.

1 Introduction

Recent advancements in large language models (LLMs) have shown promise in question answering, demonstrating their ability to understand and respond to a wide range of questions [10, 12, 23]. However, these models frequently suffer from a phenomenon known as hallucination, where the generated content deviates from real-world facts, particularly when handling queries that fall outside the model's training data or require up-to-date information [5, 14, 24].

To address the limitations of LLMs, Retrieval-Augmented Generation (RAG) [8] has been proposed as a promising approach. RAG systems enhance the performance of LLMs by retrieving relevant information from external knowledge sources, thus grounding the generated content in real-world data. This approach helps to ground the generated answers in factual data, thereby mitigating the issue of hallucination [2, 3, 8]. Despite these advancements, RAG still faces several challenges. The process of retrieving and integrating external information increases the computational cost and response time, which can be inefficient when LLMs are capable of generating accurate responses independently [17]. Additionally, incorrect retrievals can exacerbate the hallucination problem, leading to the propagation of incorrect information [21].

In response to these challenges, the Meta KDD Cup 2024 [1] has been introduced to advance the capabilities of RAG systems. Our approach in this competition aims to address two key issues associated with RAG: unnecessary retrievals and the propagation of incorrect information. We propose enhancing the standard RAG framework by incorporating two additional steps. First, we leverage the inherent knowledge stored in LLMs to avoid unnecessary retrievals when the model can generate accurate responses independently. Second, we introduce a verification step to reassess generated answers, thereby reducing the likelihood of delivering incorrect information to users. This approach not only optimizes

the efficiency of QA systems but also enhances their reliability and accuracy.

Our contributions can be summarized as follows:

- We propose a three-step question-answering framework that builds upon a basic RAG structure by adding additional steps before and after it. This framework utilizes the latent knowledge within large language models to minimize unnecessary data retrieval and includes a verification phase to ensure the factual accuracy of the generated responses.
- We present an evaluation of our enhanced RAG system within the Meta KDD Cup 2024, demonstrating significant improvements in response accuracy and computational efficiency compared to traditional RAG implementations.
- We release the implementation of our prototype, providing an accessible and practical framework for the community to build upon, thereby promoting further research and development in enhancing the reliability of QA systems.

2 Problem Definition

This section defines the problem and explains the terms and notations used in our work. We denote a question as Q and answer as A.

Definition 2.1. **Web Search Results** $\mathcal{W} = [w_1, \ldots, w_n]$ is a list of web pages, where each web page w_i consists of a sequence of tokens.

Definition 2.2. **Mock Knowledge Graph** $\mathcal{K} = \{(s_1, p_1, o_1), \ldots, (s_m, p_m, o_m)\}$ is a structured data source composed of triples (s, p, o), where s represents a subject, p a predicate, and o an object. These triples encapsulate factual information and relationships.

To evaluate the performance of Retrieval-Augmented Generation (RAG) systems in the KDD Cup, three distinct tasks are designed. Each task involves generating an answer A correctly in response to a question Q, utilizing knowledge from either external sources E and a large language model (LLM). The tasks vary in the type and volume of external sources provided, testing different capabilities of end-to-end RAG systems.

Definition 2.3. **Task 1: Retrieval Summarization.** Given a question Q and a set of external sources $E = \{\mathcal{W}\}$ ($|\mathcal{W}| \leq 5$), generate an answer A using relevant information from these web pages to form a *coherent* and *accurate* response.

Definition 2.4. **Task 2: KG and Web Retrieval Augmentation.** Given a question Q and a set of external sources $E = \{\mathcal{W}, \mathcal{K}\}$ ($|\mathcal{W}| \leq 5$), generate a *coherent* and *accurate* answer A by leveraging both the web pages and the structured data comprehensively. This task extends **Task 1** by incorporating structured data $\mathcal{K}$.

Definition 2.5. **Task 3: End-to-End RAG.** Given a question Q and a set of external sources $E = \{\mathcal{W}, \mathcal{K}\}$ ($|\mathcal{W}| \leq 50$), *efficiently* generate a *coherent* and *accurate* answer A utilizing both web pages and the structured data.

Algorithm 1: ThreeStepQA

Input: Q: A question
Input: E: External sources
Output: A: An answer
/* Step1:Answering With Parameterized Knowledge */
/* A_P:Answer generated with parameterized knowledge */
/* C_P:Confidence score of A_P */
/* T_P:Threshold for C_P */
$A_P, C_P \leftarrow GenerateWithParameterized(Q)$;
if $C_P > T_P$ **then**
  return A_P;
else
  /* Step2:Answering With External Sources */
  /* A_E:Answer generated using external knowledge */
  /* C_E:Confidence score of A_E */
  /* S_Q:Retrieved results for Q */
  /* T_E:Threshold for C_E */
  $A_E, C_E, S_Q \leftarrow GenerateWithExternal(Q, E)$;
  if $C_E > T_E$ **then**
    return A_E;
  else
    /* Step3:Final Answer Selection */
    /* A_F:Answer finally selected */
    /* C_F:Confidence score of A_F */
    /* T_F:Threshold for C_F */
    $A_F, C_F \leftarrow SelectFinalAnswer(Q, S_Q, A_P, A_E)$;
    if $C_F > T_F$ **then**
      return A_F;
    else
      return "I don't know"

3 Three-Step Question-Answering Framework

Three-step question-answering framework is designed to provide accurate and reliable answers to user queries. The framework consists of the following stages: (1) Question Answering With Parameterized Knowledge, (2) Question Answering With External Sources, and (3) Final Answer Selection, as outlined in Algorithm 1. Each stage serves a distinct purpose: the first stage leverages the knowledge stored in large language models to provide initial answers without unnecessary retrieval, thus reducing the end-to-end response time. The second stage utilizes external sources to accurately address the query by retrieving additional information. Finally, the third stage reassesses the validity of the answers generated in the previous stages to enhance the reliability of the final response.

Figure 1 illustrates the question-answering process using the three-step framework. For instance, when asked, "What school won the women's gymnastics NCAA championship in 2022?" the first stage attempts to answer the question based solely on the model's parameterized knowledge. If the generated answer, such as "Florida," does not surpass a predefined confidence threshold, indicating insufficient information, the process moves to the second stage. In the second stage, the system searches for relevant information from external sources and generates a new answer, such as "Oklahoma." If this answer still does not meet the confidence threshold, the third stage is invoked. The final stage involves selecting

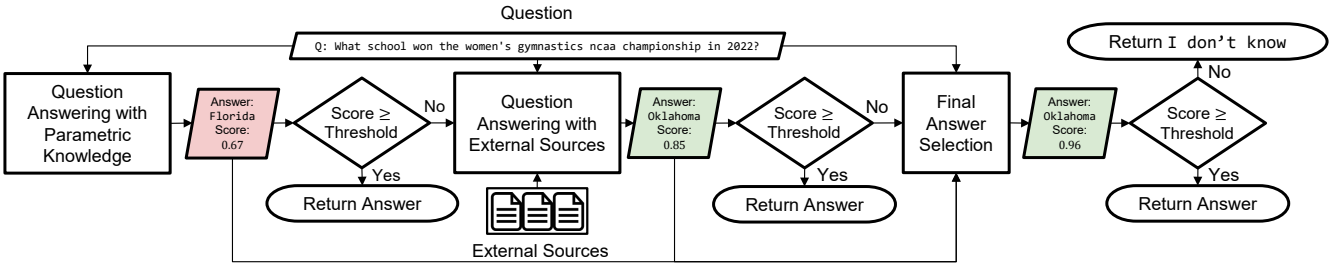

Figure 1: Flowchart for Three-step question-answering framework. Framework consists of three stages: (1) Question Answering With Parameterized Knowledge, (2) Question Answering With External Sources, and (3) Final Answer Selection.

from the previously generated answers—"Florida," "Oklahoma," or "I don't know"—to ensure the most accurate and reliable response is returned. In the following sections, we will provide a detailed explanation of each stage of the three-step question-answering framework in sequence.

3.1 Question Answering with Parameterized Knowledge

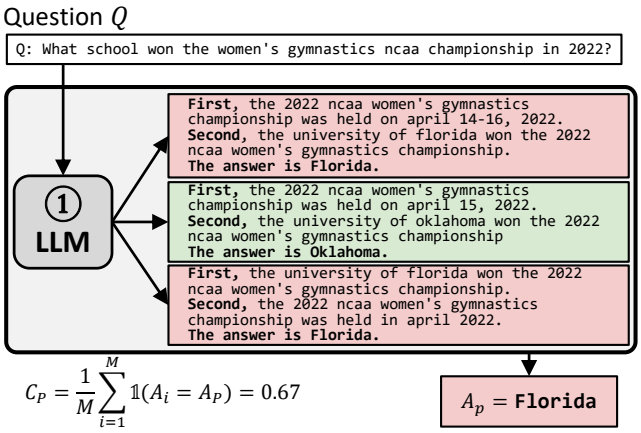

Figure 2: Question answering with parameterized knowledge.

Step 1 involves using only the parameterized knowledge of the LLM to answer questions. To enable the LLM to respond to complex questions, we utilized Chain-of-Thought (CoT) prompting to generate answers [20]. CoT is a prompting technique that produces intermediate rationales before arriving at the final answer:

$$R, A = LLM_\theta(COT(Q)) \tag{1}$$

where R represents the sequence of intermediate rationales generated before deriving the final answer A. CoT has demonstrated that guiding the model to decompose the complex problem into simpler sub-problems allows it to effectively leverage the knowledge embedded in the LLM [20].

Moreover, we fine-tuned the LLM to improve its performance on the question-answering task and enhance the quality of rationale generation. The training data comprised 90% of the validation and public test sets provided by the CRAG benchmark [21]. Training pairs consisted of (prompt with question, answer with rationales). Since the training dataset did not include rationales, we synthesized rationales by providing the model with both the question and the answer, prompting it to generate the rationales that led to the given answer. In section 4.4, we conduct an ablation study that empirically verifies the performance gains from fine-tuning the model with synthesized rationales.

To identify questions that require external knowledge, we utilize a self-consistency score [19]. The self-consistency score is calculated by sampling multiple reasoning paths and their corresponding answers, and then assessing the frequency of the most common answer. Let A_i denote the answer obtained from the i-th reasoning path, and A_P represent the most common answer. The self-consistency score C_P is defined as follows:

$$C_P = \frac{1}{M}\sum_{i=1}^{M} \mathbb{1}(A_i = A_P) \tag{2}$$

where M is the number of sampled reasoning paths. Previous studies have demonstrated that this score is highly correlated with the accuracy of the generated answer, with a lower score indicating less confidence in the response [9, 22, 25]. Therefore, if the self-consistency score C_P falls below a predefined threshold T_P, it indicates that the model's parameterized knowledge might not be sufficient to produce a confident or accurate answer. In such cases, the model proceeds to the next step, which involves utilizing external knowledge sources. In Section 4.1.4, we describe how the thresholds for each step were determined.

Figure 2 illustrates a question-answering process that leverages parameterized knowledge. When the question "What school won the women's gymnastics NCAA championship in 2022?" is posed, the large language model (LLM) generates three pairs of intermediate rationales and answers. An example of the intermediate rationales includes statements like, "First, the 2022 NCAA women's gymnastics championship was held on April 14-16, 2022. Second, the University of Florida won the 2022 NCAA women's gymnastics championship." The corresponding answer generated is, "Florida" Among these generated answers, the most frequent one, "Florida", is selected as the answer for step 1. The frequency of this answer, 0.67, is used as the confidence score.

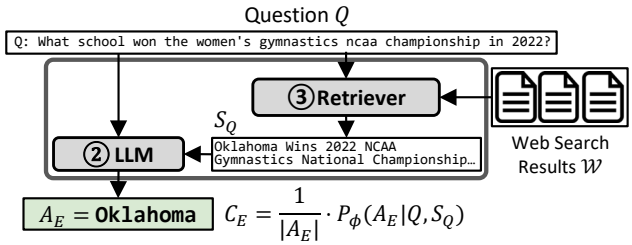

Figure 3: Question answering with external sources.

Algorithm 2: GenerateWithExternal

Input: Q: A question
Input: E: A set of external sources
Output: A_E: An answer
Output: C_E: A confidence score
Output: S_Q: Retrieved results for Q

```
/* Initialize an empty set L to store answers,
   confidence scores, and retrieved results          */
L ← ∅;
/* Iterate over each source S in E                   */
foreach S ∈ E do
    if S = 𝒦 then
        S_Q ← MockAPI(Q, S);
    else
        S_Q ← Retriever(Q, S);
    /* A_S:Answer generated with S_Q                  */
    /* C_S:Confidence score of A_S                    */
    A_S ← LLM_φ(Q, S_Q);
    P_φ ← GetProbDistr(LLM_φ);
    C_S ← (1/|A_S|) · P_φ(A_S|Q, S_Q);
    L ← L ∪ {(A_S, C_S, S_Q)};
(A_E, C_E, S_Q) ← argmax_{(A_S,C_S,S_Q)∈L} {C_S};
return A_E, C_E, S_Q
```

3.2 Question Answering With External Sources

In Step 2, the process involves searching for information related to the query from external sources and using the retrieved information to answer the given question as described in Algorithm 2. There are two main types of external sources: knowledge graphs $\mathcal{K}$ and web search results $\mathcal{W}$. We employ different methods to retrieve information based on the type of external source:

$$S_Q = \begin{cases} \text{MockAPI}(Q, S) & \text{if } S = \mathcal{K} \\ \text{Retriever}(Q, S) & \text{if } S = \mathcal{W} \end{cases} \tag{3}$$

where $S_Q \subset S$ is the set of retrieved results for the question Q from the external source $S \in E$.

For the knowledge graph $\mathcal{K}$, we follow the method proposed in CRAG [21]. This involves the LLM selecting the appropriate mock API for the question and generating the necessary API arguments to execute it, producing relevant results for the query. A total of 38 mock APIs are available, and the selection of the appropriate API(s) is determined using a decision tree. When the LLM generates the query's domain and corresponding API arguments, the decision tree is used to sequentially call the relevant mock APIs based on the provided domain and arguments. The outputs from these API calls are then utilized as the final results for the query.

For web search results $\mathcal{W}$, we divide the web pages into chunks of 512 tokens with a 32-token overlap and retrieve the top-4 most relevant chunks using an all-to-all interaction-based retriever [18]. The rationale for selecting an all-to-all interaction-based retriever is its ability to achieve significantly higher accuracy compared to bi-encoder and late-interaction-based retrievers [4, 6, 7, 16]. Additionally, since web pages cannot be pre-encoded offline, the speed advantage typically associated with bi-encoder and late-interaction methods is not realized, making the all-to-all interaction approach preferable in this context.

To generate answers, we utilize the ChatQA [13], fine-tuned version of the Llama-3 base model. This system is specifically optimized for leveraging retrieved results through a two-step instruction-tuning process. The generation of the answer is formally represented as follows:

$$A_S = LLM_\phi(Q, S_Q) \tag{4}$$

Since the ChatQA model generates the answer directly rather than employing chain-of-thought reasoning, we cannot compute a self-consistency score. Instead, we use the probability of the answer as a confidence score:

$$C_S = \frac{1}{|A_S|} \cdot P_\phi(A_S|Q, S_Q) \tag{5}$$

Here, A_S represents the sequence of tokens in the generated answer, Q is the question, and S_Q is the corresponding context. The confidence score C_S is normalized by the length of the sequence $|A_S|$ to ensure comparability between answers of different lengths. This normalization process mitigates length bias, making confidence scores more consistent and reliable across varying answer lengths.

When multiple external sources are available ($|E| > 1$), it is crucial to decide which source's information to utilize. We assume that the confidence score reflects the relevance of the provided information to the query. Therefore, we independently conduct question-answering for each source and select the answer with the higher confidence score:

$$A_E, C_E, S_Q = \text{argmax}_{(A_S,C_S,S_Q)} \{C_S \mid S \in E\} \tag{6}$$

If the score C_E does not exceed a predefined threshold T_E, the model proceeds to the next step.

Figure 3 illustrates the question-answering process with external sources in Task 1. When the question "What school won the women's gymnastics NCAA championship in 2022?" is presented, the retriever searches for relevant information from web search results. For instance, it retrieves the information "Oklahoma Wins 2022 NCAA Gymnastics National Championship...". Based on the retrieved results, the LLM provides an answer to the question. The confidence score is then calculated for the answer "Oklahoma" based on the probability value assigned.

3.3 Final Answer Selection

In the final stage, we transform the free-response question answering task into a multiple-choice format, allowing the LLM to re-evaluate potentially overlooked information by choosing the

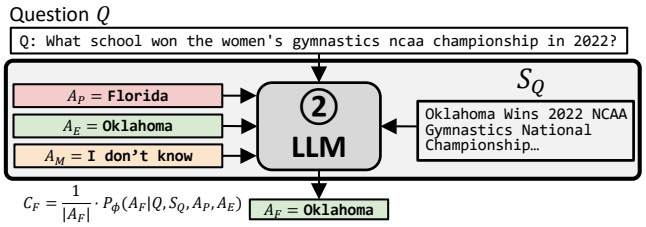

Figure 4: Final answer selection.

final answer from answer candidates. The answer candidates consist of responses generated in steps 1 and 2 and an "I don't know" option. We use the same LLM from step 2 to generate the answers, formally represented as follows:

$$A_F = LLM_\phi(MultipleChoice(Q, S_Q, A_P, A_E)) \tag{7}$$

The confidence score for the generated response was determined using the same method as in step 2:

$$C_F = \frac{1}{|A_F|} \cdot P_\phi(A_F|Q, S_Q, A_P, A_E) \tag{8}$$

If the confidence score does not exceed the predefined threshold, the model returns "I don't know" to prevent hallucinations.

Figure 4 shows the final answer selection process. The question "What school won the women's gymnastics NCAA championship in 2022?" is presented along with the answer options from the first step, "Florida," the second step, "Oklahoma," and "I don't know." Using the results retrieved in step 2, the LLM re-evaluates these options to determine which one is the correct answer or if the correct answer is unknown.

4 Experiments

In this section, we present a detailed comparison of the accuracies of answers inferred by our algorithm against those generated by competing algorithms. Second, we present the maximum scores achieved by participating teams for each type of question. Lastly, we conduct an ablation study to understand the contribution of different components of our algorithm to its overall performance.

4.1 Experimental Settings

Table 1: The number and percentages (% in parentheses) of questions for each question type in the CRAG dataset [21].

Question type	Finance	Sports	Music	Movie	Open	Total
Simple	466 (45%)	23 (3%)	112 (18%)	519 (46%)	85 (11%)	1205 (27%)
Simple w. condition	113 (11%)	250 (30%)	92 (15%)	112 (10%)	122 (15%)	689 (16%)
Set	48 (5%)	93 (11%)	72 (12%)	104 (9%)	86 (11%)	403 (9%)
Comparison	146 (14%)	85 (10%)	102 (16%)	105 (9%)	98 (12%)	536 (12%)
Aggregation	69 (7%)	137 (16%)	96 (15%)	71 (6%)	116 (15%)	489 (11%)
Multi-hop	86 (8%)	64 (8%)	55 (9%)	90 (8%)	87 (11%)	382 (9%)
Post-processing	26 (3%)	24 (3%)	26 (4%)	28 (2%)	76 (10%)	180 (4%)
False Premise	85 (8%)	157 (19%)	69 (11%)	96 (9%)	118 (15%)	525 (12%)
All	1039	833	624	1125	788	4409

4.1.1 Dataset. We utilize the Comprehensive Retrieval-Augmented Generation (CRAG) dataset to evaluate our algorithm. CRAG is a benchmark designed specifically for factual question answering using Retrieval-Augmented Generation (RAG) systems. It consists of 4,409 question-answer pairs, meticulously curated to represent a diverse array of real-world queries.

Each data entry in the CRAG dataset comprises a question and its corresponding answer, derived from either external sources or a large language model. The dataset spans five distinct domains: *Finance*, *Sports*, *Music*, *Movies*, and *Open Domain*. These domains represent the spectrum of information change rates: fast-changing (*Finance* and *Sports*), slow-changing (*Music* and *Movies*), and static (*Open Domain*). CRAG includes eight types of questions, which encompass a wide range of complexities and retrieval challenges:

- *Simple question*: Questions asking for simple facts, such as *the birth date of a person* and *the authors of a book*.
- *Simple question with some condition*: Questions asking for simple facts with some given conditions, such as *stock price on a certain date* and *a director's recent movies in a certain genre*.
- *Set question*: Questions that expect a set of entities or objects as the answer. An example is *what are the continents in the southern hemisphere?*
- *Comparison question*: Questions that may compare two entities, such as *who started performing earlier, Adele or Ed Sheeran*?
- *Aggregation question*: Questions that may need aggregation of retrieval results to answer, for example, *how many Oscar awards did Meryl Streep win?*
- *Multi-hop questions*: Questions that may require chaining multiple pieces of information to compose the answer, such as *who acted in Ang Lee's latest movie?*
- *Post-processing question*: Questions that need reasoning or processing of the retrieved information to obtain the answer, for instance, *How many days did Thurgood Marshall serve as a Supreme Court justice?*
- *False Premise question*: Questions that have a false preposition or assumption; for example, *What is the name of Taylor Swift's rap album before she transitioned to pop?* (Taylor Swift did not release any rap album.)

Specific statistics about the CRAG dataset is expressed in Table 1.

4.1.2 Metrics. A structured scoring method has been implemented to evaluate the response quality of Retrieval-Augmented Generation (RAG) systems in the KDD CUP 2024 competition. The evaluation was conducted using a large language model, which assessed each answer in the evaluation set according to the following categories:

- *Perfect*: The response accurately answers the user's query without any hallucinated content.
- *Acceptable*: The response offers a useful answer but may contain minor inaccuracies that do not diminish its overall utility.
- *Hallucination*: The response provides wrong or irrelevant information to answer the user question.
- *Missing*: The response indicates a lack of information, using phrases such as "`I don't know`", "`I'm sorry I can't find ...`" or results from system errors, including empty responses or requests for clarification.

Each category is assigned a score: 1 for *Perfect*, 0.5 for *Acceptable*, −1 for *Hallucination*, and 0 for *Missing*. This scoring system penalizes hallucinations more than missing answers, reflecting a preference for incomplete but accurate responses over incorrect

ones [21]. The overall score is calculated as a macro-average across all domains. Additionally, the weight assigned to each question depends on the popularity of its type and the entity's popularity [1]. This method ensures a comprehensive and balanced evaluation of RAG system performance, aligning with the competition's objectives and standards.

4.1.3 Hardware and Software Settings. Our and the competitors' algorithms were run on an AWS G4dn.12xlarge instance equipped with 4 NVIDIA T4 GPUs with 16GB GPU memory. Any kind of network connection was disabled during evaluation. Each example has a time-out limit of 30 seconds. Each answer is truncated to 75 bpe tokens before evaluation.

4.1.4 Our Implementation Details. The optimal threshold for each stage was determined using a grid search with a step size of 0.1, applied to 10% of the data from the validation and public test sets of the CRAG benchmark. This 10% subset was selected after excluding the data used for fine-tuning the LLM. `Meta-Llama-3-70B-Instruct` model was used to generate rationales and answers in our step 1 module (§ 3.1, ① in Figure 2). We quantized the model using AWQ [11] to fit the GPU limitation. `Llama3-ChatQA-1.5-8B` [13] model was used in both step 2 module (§ 3.2, ② in Figure 3) and step 3 module (§ 3.3, ② in Figure 4). Lastly, `ms-marco-MiniLM-L-6-v2` [15] cross encoder was used to rank each text fragment (§ 3.3, ③ in Figure 4).

4.2 Accuracy Comparison

To evaluate the accuracy of our algorithm, we employed the comprehensive automatic evaluation method introduced by KDD Cup 2024. If an answer exactly matches the ground truth, it is classified as *perfect*. In cases where the answer does not match exactly, large language models (LLMs) are utilized to assess whether the response is *acceptable*, a *hallucination*, or *missing*. This evaluation process incorporates rule-based matching and GPT-4 assessment to verify the correctness of the answers. The *overall score* is calculated by averaging the weighted scores assigned to each classification: perfect answers receive 1 point, acceptable answers also receive 1 point, missings receive 0 points, and hallucinations are penalized with -1 point. The evaluation results for each task are shown in Table 2, Table 3, Table 4. In the tables, the terms are abbreviated in the columns as follows: perfect (P), acceptable (A), hallucination (H), missing (M), overall score (OS).

Table 2: Evaluation results for task 1 showing the performance metrics (P, A, H, M, OS) for the top 8 teams.

Team	P↑	A↑	H↓	M	OS↑
AIFIRST	14.7	40.8	9.6	34.8	45.9
db3	19.8	23.7	15.6	40.8	27.9
dRAGonRAnGers †	16.8	14.4	8.1	60.7	23.1
TieMoJi	21.3	5.4	4.2	69.1	22.5
md_dh	18.6	8.4	8.4	64.6	18.6
StarTeam	16.2	20.1	18.9	44.7	17.4
ETSLab	13.5	19.5	16.8	50.2	16.2
ElectricSheep	12.3	12.6	12.0	63.1	12.9
... 376 other teams ...					

Table 3: Evaluation results for task 2 showing the performance metrics (P, A, H, M, OS) for the top 7 teams.

Team	P↑	A↑	H↓	M	OS↑
db3	39.0	12.0	18.9	30.0	32.1
APEX	33.3	11.7	13.2	41.7	31.8
TieMoJi	24.3	5.7	6.6	63.4	23.4
dRAGonRAnGers †	16.5	14.7	8.4	60.4	22.8
StarTeam	24.9	15.6	17.7	41.7	22.8
md_dh	21.9	9.9	13.5	54.7	18.3
ElectricSheep	16.2	11.4	14.1	58.3	13.5
... 377 other teams ...					

Table 4: Evaluation results for task 3 showing the performance metrics (P, A, H, M, OS) for the top 7 teams.

Team	P↑	A↑	H↓	M	OS↑
db3	32.8	18.9	22.3	26.0	29.4
APEX	27.9	16.4	19.2	36.5	25.1
StarTeam	20.4	16.1	15.5	48.0	21.1
TieMoJi	23.2	14.0	17.0	45.8	20.1
dRAGonRAnGers †	13.9	14.3	11.5	60.3	16.7
ElectricSheep	11.1	15.5	15.2	58.2	11.5
Future	11.1	3.4	6.5	78.9	8.0
... 377 other teams ...					

We identified two significant observations. First, our algorithm recorded missing scores exceeding 60% across all three tasks, while maintaining lower hallucination scores compared to other competitors. This is attributed to the comprehensive confidence-checking mechanisms in Step 1, Step 2, and Step 3, where less confident answers were converted to `"I don't know"`, effectively marking potential hallucinations as missing. By prioritizing accuracy and avoiding the risks associated with generating potentially incorrect information, our approach emphasizes reliability over speculative responses. Second, compared to Task 1, the perfect scores for our algorithm remained consistent, while the acceptable scores decreased in Tasks 2 and 3. This decline is due to our current limitations in utilizing mock APIs effectively, which we identify as a crucial area for future improvement.

4.3 Human Evaluation

Table 5: Highest-ranked accuracy scores among KDD Cup competitors for each task (columns) and question type (rows). Scores achieved by our algorithm are highlighted in blue, bold font.

Question Type	Task 1	Task 2	Task 3
Simple w. Condition	17.9	23.9	42.2
Set	21.25	36.65	31.7
Comparison	**37**	**38**	**37.25**
Aggregation	21.5	18.75	26.6
Multi-hop	16.8	23.2	25.7
Post-processing	**8.6**	11.75	8.3
False Premise	65.2	64.6	72.2

Human evaluation was conducted by manually classifying each answer into one of four categories: perfect, acceptable, hallucination, and missing. Evaluators assessed the correctness and completeness of the responses, ensuring that perfect answers were accurate and free from hallucinations, while acceptable answers, though useful, could contain minor errors. The evaluation process also considered the fluency and coherence of the responses, with the entire answer being reviewed to detect any hallucinations.

Our algorithm secured the first place across all three tasks in the comparison question category (see the blue, bold-fonted results in Table 5). While the detailed results of the human evaluation were not publicly disclosed, we hypothesize that our chain-of-thought method significantly contributed to this success. This approach involves generating two rationales based on the relevant information for each entity involved in the comparison, and then synthesizing a final answer from these rationales. This method is particularly effective for comparison questions, as it systematically evaluates and contrasts the pertinent details of the entities in question. For example in Table 6, when asked which movie cost more to create, the method first considers the production budgets of "Star Wars: A New Hope" and "Avengers: Endgame" separately, then compares these figures to provide a final answer.

Table 6: An example of a question-answer pair where our algorithm correctly answered a comparison question.

Query	which movie cost more to create, star wars: new hope or avengers: endgame?
Answer	avengers: endgame
Rationale 1	first, the production budget for star wars: a new hope (1977) was approximately *$11 million.*
Rationale 2	second, the production budget for avengers: endgame (2019) was approximately *$356 million.*
Our prediction	avengers: endgame

4.4 Ablation Study

Table 7: Ablation study results showing the performance metrics (P, A, H, M, OS, Time) for different combinations of the three stages (S1, S2, S3) in our framework.

S1 (§ 3.1)	**S2** (§ 3.2)	**S3** (§ 3.3)	**P**↑	**A**↑	**H**↓	**M**	**OS**↑	**Time**↓ (s)
✓			17.1	18.9	15.0	48.9	21.0	**6.7533**
	✓		**18.9**	**20.7**	21.7	38.7	17.9	14.5084
✓	✓		17.1	14.7	11.7	56.5	20.1	11.6402
✓	✓	✓	16.5	14.7	**8.4**	60.4	**22.8**	11.8623

To assess the impact of each stage in our three-step framework, we conducted an ablation study. The results, presented in Table 7, reveal the contributions of each stage to the overall performance of our algorithm.

We highlight three key observations from our study. First, when Step 1 is used in conjunction with Step 2, the algorithm execution time is significantly reduced. Specifically, compared to using only Step 2, the runtime decreases by 19.8%. This improvement is due to the self-consistency score used in Step 1, which allows the system to skip the more time-consuming Step 2 (approximately 14.5 seconds) if the parametric knowledge is deemed sufficient. Second, the naive RAG system, represented by using only Step 2, demonstrates notable drawbacks. Hallucinations occur frequently because the system always relies on retrieved knowledge to generate answers, regardless of its accuracy. Additionally, the execution time is prolonged (14.5 seconds) as the system consistently performs retrieval operations, even when they are unnecessary. Third, Introducing Step 3 to the framework resulted in a 28.2% reduction in hallucinations. This decrease is attributed to the confidence score mechanism, which effectively filters out hallucinated responses by evaluating the reliability of answers from various sources.

These findings underscore the importance of each step in our framework, particularly the role of self-consistency scoring in enhancing efficiency and confidence-based filtering in improving response accuracy. The ablation study confirms that our multi-step approach not only optimizes runtime but also significantly mitigates the issue of hallucinations, thereby enhancing the overall reliability and performance of our RAG system.

Table 8: Ablation study results showing the performance metrics (P, A, H, M, OS) for base model and fine-tuned model.

Fine-tuned	**P**↑	**A**↑	**H**↓	**M**	**OS**↑
	7.2	10.8	7.8	74.2	10.2
✓	**16.5**	**14.7**	**8.4**	60.4	**22.8**

We also conducted ablation studies to assess the impact of fine-tuning the model to generate a reasoning path composed of two steps before producing the final answer. Our findings reveal that fine-tuning the model in this way resulted in more than a twofold improvement in the overall score compared to the baseline without fine-tuning. As depicted in Figure 2, the reasoning path follows the structure: "First, [Rationale 1]. Second, [Rationale 2]. The answer is [Answer]." Any generated reasoning path that did not conform to this structure was considered a failure and excluded from evaluation.

In the case where the model was not fine-tuned, a significant number of reasoning paths deviated from the desired format, leading to less accurate self-consistency scores. The fine-tuning process reduced the frequency of these format errors, enabling more precise calculation of self-consistency scores. This indicates that fine-tuning plays a critical role in improving the model's ability to generate structured reasoning paths, thereby enhancing overall performance.

5 Conclusion

In this paper, we introduced a three-step question-answering framework that enhances Retrieval-Augmented Generation (RAG) systems by leveraging the inherent knowledge of large language models (LLMs), integrating external knowledge sources, and incorporating a final verification step to reduce hallucinations and improve computational efficiency. The framework's effectiveness was confirmed by achieving first place in the comparison question category across all tasks, as well as first place in the post-processing category for task 1. By releasing our prototype, we aim to provide a practical tool for the research community, fostering further advancements in question-answering systems.

References

[1] AIcrowd. 2024. Meta Comprehensive Rag Benchmark: KDD Cup 2024: Challenges. https://www.aicrowd.com/challenges/meta-comprehensive-rag-benchmark-kdd-cup-2024

[2] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking Large Language Models in Retrieval-Augmented Generation. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 16 (Mar. 2024), 17754–17762. https://doi.org/10.1609/aaai.v38i16.29728

[3] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997 [cs.CL] https://arxiv.org/abs/2312.10997

[4] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. In *International Conference on Learning Representations*.

[5] Ziwei Ji and et al. 2023. Survey of hallucination in natural language generation. *Comput. Surveys* 55, 12 (2023), 1–38.

[6] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6769–6781.

[7] Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 39–48.

[8] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 793, 16 pages.

[9] Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2024. Chain-of-Knowledge: Grounding Large Language Models via Dynamic Knowledge Adapting over Heterogeneous Sources. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=cPgh4gWZlz

[10] Valentin Liévin, Christoffer Egeberg Hother, Andreas Geert Motzfeldt, and Ole Winther. 2024. Can large language models reason about medical questions? *Patterns (New York, N.Y.)* 5, 3 (8 March 2024), 100943. https://doi.org/10.1016/j.patter.2024.100943 © 2024 The Authors..

[11] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration. In *MLSys*.

[12] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023), 1–35.

[13] Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. 2024. ChatQA: Surpassing GPT-4 on Conversational QA and RAG. arXiv:2401.10225 [cs.CL] https://arxiv.org/abs/2401.10225

[14] Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S.M Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. 2023. The Troubling Emergence of Hallucination in Large Language Models - An Extensive Definition, Quantification, and Prescriptive Remediations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 2541–2573. https://doi.org/10.18653/v1/2023.emnlp-main.155

[15] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[16] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 3715–3734.

[17] Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. DRAGIN: Dynamic Retrieval Augmented Generation based on the Real-time Information Needs of Large Language Models. *CoRR* abs/2403.10081 (2024). https://doi.org/10.48550/arXiv.2403.10081

[18] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. https://openreview.net/forum?id=wCu6T5xFjeJ

[19] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*.

[20] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[21] Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas Scheffer, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen tau Yih, and Xin Luna Dong. 2024. CRAG – Comprehensive RAG Benchmark. *arXiv preprint arXiv:2406.04744* (2024). https://arxiv.org/abs/2406.04744

[22] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations (ICLR)*.

[23] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, Online, 535–546. https://doi.org/10.18653/v1/2021.naacl-main.45

[24] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. arXiv:2309.01219 [cs.CL] https://arxiv.org/abs/2309.01219

[25] Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-Edit: A Knowledge-Enhanced Chain-of-Thought Framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 5823–5840. https://doi.org/10.18653/v1/2023.acl-long.320

Received 9 August 2024