RA-ISF: Learning to Answer and Understand from Retrieval Augmentation via Iterative Self-Feedback

Anonymous ACL submission

Abstract

Large language models (LLMs) demonstrate exceptional performance in numerous tasks but still heavily rely on knowledge stored in their parameters. Moreover, updating this knowledge incurs high training costs. Retrieval-augmented generation (RAG) methods address this issue by integrating external knowledge. The model can answer questions it couldn't previously by retrieving knowledge relevant to the query. This approach improves performance in certain scenarios for specific tasks. However, if irrelevant texts are retrieved, it may impair model performance. In this paper, we propose Retrieval Augmented Iterative Self-Feedback (RA-ISF), a framework that iteratively decomposes tasks and processes them in three submodules to enhance the model's problem-solving capabilities. Experiments show that our method outperforms existing benchmarks, performing well on models like GPT3.5, Llama2, significantly enhancing factual reasoning capabilities and reducing hallucinations.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023) have demonstrated their excellent performance in knowledge reasoning and outstanding capabilities across various task domain (Bang et al., 2023; Ouyang et al., 2022). However, the parameterized knowledge stored within LLMs may be incomplete and hard to incorporate up-to-date knowledge (Dhingra et al., 2022; Huang et al., 2020). To address this issue, retrieval-augmented generation (RAG) approaches can leverage external knowledge and documents, extract non-parameterized knowledge, and incorporate it into the model's prompts, thereby embedding new knowledge into the language model (Ram et al., 2023; Guu et al., 2020). This approach demonstrates outstanding performance in answering a variety of open-domain questions.

However, current RAG frameworks have two major challenges. First, retrieving irrelevant knowledge texts will impair the LLMs' ability to solve tasks (Shi et al., 2023a; Mallen et al., 2023). Second, the incorporation of LLM's existing knowledge and the retrieved knowledge may face difficulty (Izacard et al., 2022b). Some methods have conducted research based on these issues, including considering the model's problem-solving abilities (Wang et al., 2023a) and whether the retrieved passages are relevant to the question (Chen et al., 2023; Asai et al., 2024; Yu et al., 2023). However, current solutions still have drawbacks in answering knowledge-intensive questions and different levels of sub-questions. Therefore, how to fuse knowledge and utilize knowledge for question answering is very important in this process.

To overcome the above limitations, we introduce **R**etrieval **A**ugmented **I**terative **S**elf-**F**eedback (**RA-ISF**), a framework addresses problems by iteratively processing questions. Specifically, unlike directly appending retrieved knowledge into prompts, our approach employs three sub-modules for iterative processing. These three sub-modules are the *Self-Knowledge Module*, the *Passage Relevance Module*, and the *Question Decomposition Module*. We have also collected a series of data through LLMs to evaluate whether a specific module possesses the corresponding capabilities. By training a small language model or simply relying on in-context learning, these modules can demonstrate capabilities in self-knowledge, relevance judgment, and question decomposition.

As shown in Figure 1, RA-ISF first uses a self-knowledge module to determine whether the current question could be answered on its own knowledge. Then, when employing a retrieval strategy, the passage relevance module will assess the relevance of each retrieved paragraph to the problem. Relevant paragraphs will be integrated into the prompt and used for prediction. When all para-

graphs are irrelevant to the question, the question decomposition module will break down the questions into sub-questions and repeat the aforementioned steps for these sub-questions. Ultimately, the model will synthesize the answers to the sub-questions to respond to the original question.

Compared to previous RAG methods, our iterative self-feedback approach more effectively unleashes the potential of the model and better incorporates external knowledge with the model's inherent knowledge. Simultaneously, RA-ISF can address questions by decomposing them when the model lacks an initial answer or retrieves irrelevant texts, combining these solutions to answer the origin question, which is an effective problem-solving strategy. Experiments on various LLMs (e.g., GPT3.5 (OpenAI, 2023) and Llama-2 (Touvron et al., 2023)) demonstrate that RA-ISF exhibits superior performance in handling complex questions compared to existing methods.

Our Contributions. Our main contributions are summarized as follows.

- We introduce RA-ISF, an innovative retrieval-augmented framework designed to tackle diverse challenges. This approach evaluates the model's ability to solve the corresponding problem and its relevance to the retrieved content through an iterative method. This comprehensive evaluation is crucial for solving complex problems.
- To the best of our knowledge, this is the first time an iterative question decomposition approach has been used in a retrieval-augmented framework, which mitigates the impact of irrelevant text interference.
- Our proposed framework significantly enhances knowledge retrieval performance across different tasks, demonstrating the potential and robustness of our framework.

2 Related Work

2.1 Retrieval Augmented Language Model

The retrieval-augmented language model (LM) is enhanced by a non-parametric memory to facilitate external knowledge access and provide provenance (Guu et al., 2020; Lewis et al., 2020; Shi et al., 2023b). However, the improved task performance of retrieval augmentation largely depends on the relevance of the retrieved passage (Shi et al., 2023a). Recently, some studies have begun to explore when to use retrieval for diverse instruction. For instance, Asai et al. (2024) integrates special feedback tokens into the language model to the need for retrieval and confirm the output's relevance, support, or completeness. Chen et al. (2023) investigates the impact of texts with different attributes and relevance on text generation performance. Some works (Mallen et al., 2023) explore the incorporation of the LLM's inherent knowledge with in-context documents. Wang et al. (2023b) improves the performance in answering self-knowledge questions by guiding the model to acquire self-knowledge capabilities. Meanwhile, other studies have concentrated on iterative retrieval augmentation (Trivedi et al., 2023; Shao et al., 2023) and accelerating retrieval speed (Xu et al., 2023).

In comparison, our method combines the model's retrieval and understanding capabilities and reduces its susceptibility to irrelevant texts. This is achieved through the task decomposition paradigm. By iteratively processing these three sub-modules with self-feedback, we develop a versatile and robust retrieval-augmented framework.

2.2 Task Decomposition

Task decomposition is an effective method for solving knowledge-intensive and other complex tasks. It involves breaking down multi-turn questions into single-turn questions, answering each sub-task separately, and then synthesizing these answers to resolve the original task. Perez et al. (2020) trains a question decomposition and task aggregation model to split and collectively solve the original problem. Yang et al. (2022) decomposes questions into a series of slot-filling tasks, transforming natural language questions into SQL queries, and implements natural language prompts corresponding to SQL clauses through a rule-based system. Least-to-most (Zhou et al., 2023) leverages the in-context learning capabilities of large language models, solving problems by providing examples of question decomposition.

RA-ISF utilizes task decomposition to mitigate the impact of irrelevant prompt texts on the model (Shi et al., 2023a), by iteratively answering sub-questions and integrating text relevance with self-knowledge answering capabilities into the framework. This enhances the performance in solving the entire problem.

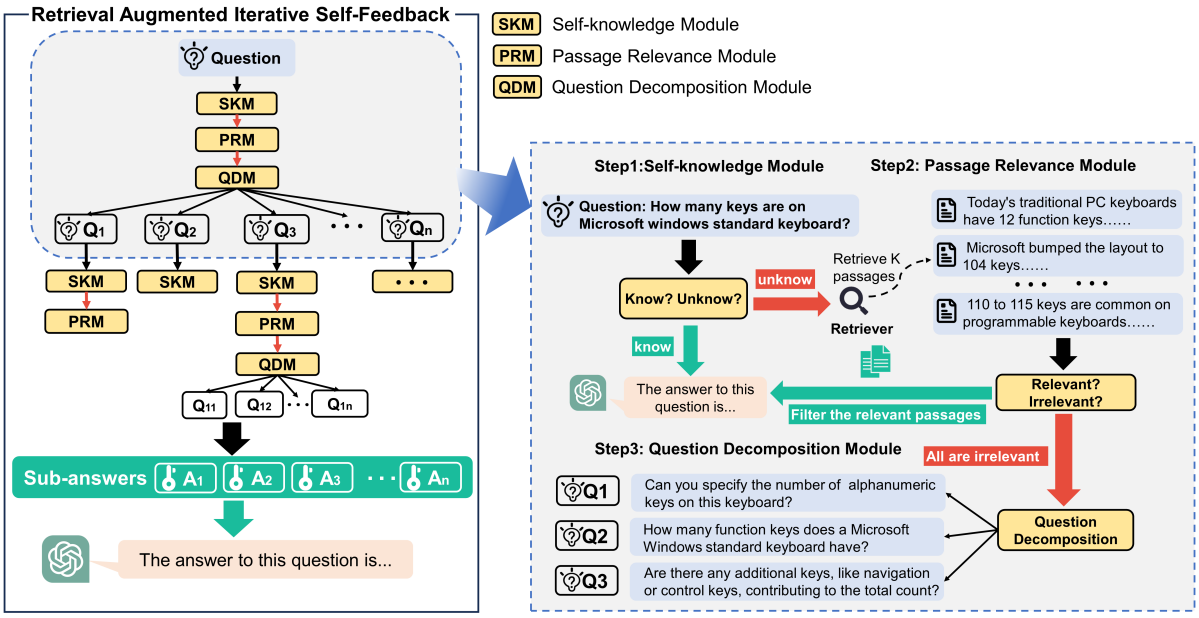


Figure 1: Overview of RA-ISF. It consists of three sub-modules: self-knowledge, passage relevance, and question decomposition.

3 Methodology

Existing retrieval augmented methods still have some shortcomings. For instance, the model may struggle to solve problems based solely on its own knowledge, and during retrieval, it might be influenced by irrelevant texts, leading to the generation of incorrect answers. Therefore, we introduce an upgraded retrieval-augmented generation framework – Retrieval Augmented Iterative Self-Feedback (RA-ISF), which improves the quality and accuracy of LLM responses through internal knowledge comprehension, external knowledge retrieval, and problem decomposition.

3.1 Overview

As shown in Figure 1, RA-ISF involves three pre-trained models: $\mathcal{M}_{know}$, $\mathcal{M}_{rel}$ and $\mathcal{M}_{decom}$, each responsible for internal knowledge assessment, external knowledge retrieval, and question decomposition functions, respectively.

In general, we input a question q_{new} and obtain its answer $\mathcal{A}$ through the RA-ISF framework. The overall process is as follows: Firstly, input q_{new} into $\mathcal{M}_{know}$ to determine if it can be solved using internal knowledge. If solvable, directly output the answer. If not, use the retriever $\mathcal{R}$ to search for relevant information for the question q_{new}. Combine the retrieved text with the question and input them into the model $\mathcal{M}_{rel}$ to assess their relevance. If relevant, generate an answer based on these related passages. If none of the retrieved text is relevant, input q_{new} into the question decomposition model $\mathcal{M}_{decom}$ to break it down into multiple sub-questions q_1, ..., q_n. Next, input these sub-questions back into the model $\mathcal{M}_{know}$ (and $\mathcal{M}_{rel}$, $\mathcal{M}_{decom}$ if needed) to obtain corresponding sub-answers. Finally, integrate these sub-answers to generate the ultimate answer.

3.2 RA-ISF Training

In this section, we will delve into the training process of the models within RA-ISF, encompassing both dataset collection and model learning. Due to the similarity in the training procedures for the three models, we will use the training of the $\mathcal{M}_{know}$ model as an illustrative example.

Data Collection. First, we need to construct a dataset generated by LLMs. Specifically, based on various training objectives, we collect corresponding questions $\mathcal{Q} = \{Q_1, Q_2, ..., Q_n\}$ and input them one by one into the LLM model $\mathcal{M}$. By providing the model with specific instructions to perform the respective tasks, and utilizing few-shot prompts and in-context learning, we enable model $\mathcal{M}$ to generate answers $\mathcal{A} = \{A_1, A_2, ..., A_n\}$ corresponding to each question.

We have collected various types of supervised training data, and through the previously described process, combined them into the training data for the model. Ultimately, this resulted in a trained dataset $\mathcal{D}^* = \{\mathcal{Q}, \mathcal{A}\}$. For specific details on the data collection process for each sub-model $\mathcal{M}_{know}$, $\mathcal{M}_{rel}$, $\mathcal{M}_{decom}$, please refer to Appendix A.

Model Learning. After collecting the training data $\mathcal{D}^*$, we initialize $\mathcal{M}_{sub}$ using a pre-trained language model and train it on $\mathcal{D}^*$ using a standard conditional language modeling objective to maximize the effectiveness of classification. Here, we use cross-entropy loss to represent this, denoted as:

$$\min_{\mathcal{M}_{sub}} -\mathbb{E}_{(\mathcal{Q},\mathcal{A})\sim\mathcal{D}^*} \log P_{\mathcal{M}_{sub}}(\mathcal{A} \mid \mathcal{Q}). \quad (1)$$

The initial model can be any pre-trained language model. Here, we initialize $\mathcal{M}_{sub}$ using the Llama 2-7B model (Touvron et al., 2023).

3.3 RA-ISF Inference

Algorithm 1: Problem Iterative Solving

Input: q_{new}, $\mathcal{M}_{know}$, $\mathcal{M}_{rel}$, $\mathcal{M}_{decom}$, $\mathcal{M}$, $\mathcal{R}$, $\mathcal{C}$
Output: $\mathcal{A}$

Function `Problm-solving`(q_t,$iter$):
  if $iter > D_{th}$ **then**
    $\mathcal{A}$=Unknow
    return $\mathcal{A}$
  if $\mathcal{M}_{know}(q_t)$=*know* **then**
    $\mathcal{A} \leftarrow \mathcal{M}(q_t)$
    return $\mathcal{A}$
  $\mathcal{P} = \{p_1, p_2, \ldots, p_k\} \leftarrow \mathcal{R}(q_t, \mathcal{C})$
  $\mathcal{P}_{rel} = \varnothing$
  for $i = 1$ **to** k **do**
    if $\mathcal{M}_{rel}(p_i)$=*relevant* **then**
      $\mathcal{P}_{rel} = \mathcal{P}_{rel} \cup p_i$
  if $size(\mathcal{P}_{rel}) > 0$ **then**
    $\mathcal{A} \leftarrow \mathcal{M}(q_t, \mathcal{P})$
    return $\mathcal{A}$
  $Q_{sub} = \{q_1, ..., q_n\} \leftarrow \mathcal{M}_{decom}(q_t)$
  for $i = 1$ **to** n **do**
    a_i=`Problm-solving`($q_i, iter + 1$)
    $\mathcal{A}_{sub} = a_i \cup \mathcal{A}_{sub}$
  $\mathcal{A} \leftarrow \mathcal{M}(q_t, Q_{sub}, \mathcal{A}_{sub})$
  return $\mathcal{A}$
$\mathcal{A} \leftarrow$ `Problm-solving`($q_{new}, 0$)

In this section, we provide a detailed explanation of how the RA-ISF framework infers and predicts answers for the question q_{new}. Algorithm 1 presents the details of RA-ISF at inference. Note that we use three pre-trained models $\mathcal{M}_{know}$, $\mathcal{M}_{rel}$, $\mathcal{M}_{decom}$, the LLM for answering questions $\mathcal{M}$, the retriever $\mathcal{R}$, and the corpus $\mathcal{C}$. Additionally, we have the question q_{new} to be addressed.

Self-Knowledge Inference. The RA-ISF framework utilizes the $\mathcal{M}_{know}$ model to infer whether the question q_{new} can be addressed using the model's own knowledge. If so, the question is input into $\mathcal{M}$ to directly predict the answer $\mathcal{A}$. The formal expression is as follows:

$$\mathcal{A} = \arg\max_{a_i} P(a_i|q_{new}). \quad (2)$$

If $\mathcal{M}$ cannot use its own knowledge to solve the question q_{new}, we move to the next step.

Passages Relevance Inference. When the model cannot solve the question q_{new} using its internal knowledge, we use the retriever $\mathcal{R}$ to search for the most suitable k passages $\mathcal{P} = \{p_1, p_2, \ldots, p_k\}$ in the corpus $\mathcal{C}$. Since the retriever may find passages unrelated to the question, potentially leading to erroneous answers, we need to filter the retrieved passages. Here, we use "relevance" as the criteria, evaluated by the model $\mathcal{M}_{rel}$.

Suppose $n(n = 0, 1, \ldots, k)$ relevant passages $\mathcal{P}_{rel}$ are finally filtered. If $n > 0$, these n passages are used as prompts, combined with q_{new}, and input into the model $\mathcal{M}$ to obtain the final answer $\mathcal{A}$. The formal expression is as follows:

$$\mathcal{A} = \arg\max_{a_i} P(a_i|q_{new}, \mathcal{P}_{rel}). \quad (3)$$

If $n = 0$, which means all the retrieved passages are irrelevant to the question, we proceed to the next step.

Problem Decomposition. If q_{new} cannot be solved using its own and external knowledge, we will decompose complex questions into a series of simpler sub-problems for resolution. In this process, we employ the $\mathcal{M}_{decom}$ model to decompose q_{new} into multiple sub-problems $Q_{sub} = \{q_1, ..., q_n\}$. Subsequently, we take each sub-problem reintroduce it to the $\mathcal{M}_{know}$ model (determining the use of $\mathcal{M}_{rel}$ and $\mathcal{M}_{decom}$ based on the specific condition), and obtain corresponding sub-answers $\mathcal{A}_{sub}$. If a sub-problem q_k has been iteratively decomposed D_{th} times, we consider that the model cannot find the

answer to this problem, and the answer for a_k is set as "unknown".

Once we have the answers $\mathcal{A}_{sub} = \{a_1, ..., a_n\}$ for all sub-problems, we use all the sub-problems Q_{sub} and their answers $\mathcal{A}_{sub}$ as prompts for q_{new}. Then input them all into the model $\mathcal{M}$ to predict the answer $\mathcal{A}$ for this question. The formal expression is as follows:

$$\mathcal{A} = \arg\max_{a_i} P(a_i | q_{new}, \mathcal{A}_{sub}, Q_{sub}). \quad (4)$$

4 Experimental Setup

4.1 Datasets

To comprehensively evaluate performance in datasets with different characteristics, we use the following five representative datasets for evaluation: Natural Question (NQ) (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), StrategyQA (Geva et al., 2021), HotpotQA (Yang et al., 2018), and 2WikiMQA (Ho et al., 2020).

4.2 Models

The models in our framework fall into two categories: an LLM for prediction and three models that serve as intermediate steps to assess the problem's characteristics. For the LLM, we experiment with open-sourced Llama2 (Touvron et al., 2023) of various sizes, as well as the GPT-3.5 (text-davinci-003) (OpenAI, 2023) through the OpenAI API. As for the three sub-models, we employ Llama2-7b as their pre-trained model.

4.3 Retriever and Corpus

For fair evaluation, we use the same retriever for different approaches to search the same corpus. Specifically, we employ Contriever-MS-MARCO (Izacard et al., 2022a) as the retriever and use the corpus from Wikipedia as of Dec. 20, 2018 (Karpukhin et al., 2020). These articles are segmented into non-overlapping fragments of 100 words. To avoid contamination, we remove input prompts x from the corpus that is contained in the dataset. To prevent the dilution of useful information, we follow Ram et al. (2023) and set the retrieval length to l = 64.

4.4 Baselines

To conduct a holistic evaluation and comparison, we use the same datasets, with the same retriever and corpus to compare our method with the following baselines:

Directly Prompting and Vanilla LM (Brown et al., 2020) involves presenting questions directly to the LLM, prompting it to generate corresponding answers without any explanations.

Least-to-most (Zhou et al., 2023) guides the LLM to break down the question and assist in solving the original problem by answering sub-questions.

IRCoT (Trivedi et al., 2023) enhances each step of the chain-of-thought generation process by incorporating knowledge retrieval steps during the generation process.

RAG (Guu et al., 2020; Lewis et al., 2020) assists in answering questions by retrieving information from external documents. We append the retrieved passage to the question in the experiment.

SKR (Wang et al., 2023a) trains a small model to determine whether the LLM can answer a question using its own knowledge, and decides whether to perform retrieval for the given question.

REPLUG (Shi et al., 2023b) adapts the framework to the corresponding downstream tasks by fine-tuning the Retriever. This method enhances retrieval effectiveness by improving the relevance of the retrieved text.

Iter-RetGen (Shao et al., 2023) conducts retrieval based on multiple iterations, relying on the content retrieved in each round to aid in finding more text information relevant to the question.

Self-RAG (Asai et al., 2024) provides a framework by training a LLM to learn specific reflection tokens, thereby controlling the decision of whether to retrieve during reasoning and examining the relevance of the retrieved content. We compares our method with the open-source Self-RAG$_{13b}$.

4.5 Implementation Details

We randomly sampled 1000 input prompts from each dataset and generated labels or answers (Relevance, Self-Knowledge) for these prompts using GPT-4. The labels or answers are then used to fine-tune these three pre-trained models. For these three models, we adopt a learning rate of 5e-4 during training. Greedy decoding is consistently used in the inference process across all experiments to maintain deterministic generation outcomes. This distillation process allows us to augment the pre-trained model with feature analysis capabilities. The default iteration threshold is set to 3. To evaluate the effectiveness of the method, we use **Exact**

Method	Avg.	NQ	TriviaQA	HotpotQA	StrategyQA	2WikiMHQA
GPT3.5 Without Retrieval						
Direct	41.8	29.2	67.3	22.1	65.2	23.6
Least-to-most	46.3	32.5	68.8	30.2	68.5	31.3
GPT3.5 With Retrieval						
IRCoT	46.5	32.9	66.8	33.7	67.9	31.1
RAG	44.2	31.7	64.2	32.2	64.7	28.4
SKR_{knn}	47.6	33.8	67.5	34.2	70.1	32.5
Iter-$RetGen_3$	-	-	-	45.2*	72.3*	34.8*
RA-ISF(ours)	**55.0**	**40.2**	**76.1**	**46.5**	**75.9**	**36.1**
$Llama\text{-}2_{13b}$ Without Retrieval						
Vanilla LM	27.1	17.4	38.5	14.0	52.2	13.3
Least-to-most	32.9	22.8	45.2	15.8	60.5	20.1
$Llama\text{-}2_{13b}$ With Retrieval						
IRCoT	34.0	23.4	48.3	17.1	59.1	21.9
RAG	33.9	21.6	47.0	17.6	60.8	22.4
SKR_{knn}	36.0	20.8	55.4	18.9	61.6	23.2
REPLUG	38.6	23.8	58.6	21.8	62.9	25.7
Self-RAG_{13B}	44.1	28.4	69.3	25.4	**67.2**	30.2
RA-ISF(ours)	**46.0**	**31.3**	**71.4**	**28.9**	66.7	**31.7**

Table 1: Main experimental results. **Bold** number indicates the best performance among all methods in this model. * indicates the results from the original paper.

Match as our standard metrics.

5 Experiment Results

5.1 Main results

The main results are shown in Table 1. From the results, we have the following observations.

Our proposed RA-ISF outperformed other methods on all five datasets on GPT3.5. On average, the performance improvement of RA-ISF is **+8.7** compared to the baseline without retrieval. Compared to the baseline with retrieval, RA-ISF surpasses all existing methods, achieving an average performance improvement of **+7.4** compared to the optimal method. In addition, compared to Iter-RetGen, which also uses iterative retrieval, RA-ISF shows an improvement of **+2.0** on HotpotQA, StrategyQA, and 2WikiMHQA.

RA-ISF is also effective on smaller-scale LLMs. We experimented with our approach on $Llama2_{13B}$, and the results showed that our method achieved SOTA on four out of five datasets, with an average improvement of +1.9 compared to the best-performing Self-RAG_{13B}. The performance of $Llama2_{13B}$ on multiple datasets reaches or even surpasses GPT-3.5 + RAG, highlighting the assistance of our method in problem-solving.

RA-ISF helps alleviate the hallucination problem associated with RAG. For instance, in TriviaQA and StrategyQA datasets, Direct RAG leads to a decrease in performance, possibly due to the negative impact of irrelevant retrieval content (Shi et al., 2023a). In our framework, three sub-modules help the model to reduce hallucinations and enhance knowledge representation. Compared to GPT-3.5 + RAG, our GPT-3.5 + RA-ISF achieves a **+11.2** performance improvement on StrategyQA. Similar performance improvements are observed on TriviaQA as well.

5.2 Ablation Studies

To assess the impact of different components of RA-ISF, we set up three variants:

- *No Self-Knowledge Module*: This variant processes questions directly through the Passage

Method	NQ (EM)	TriviaQA (EM)	HotpotQA (EM)
Direct	29.2	67.3	22.1
RAG	31.7	64.2	32.2
Least-to-Most	32.5	68.8	30.2
RA-ISF	**40.2**	**76.1**	**46.5**
No SKM	37.9	72.3	40.1
No PRM	35.8	70.3	34.7
No QDM	34.6	71.5	34.9

Table 2: Ablation of different components on GPT3.5. *No SKM*, *No PRM*, and *No QDM* stand for removing the submodel of Self-Knowledge, Passage-Relevant, and Question Decomposition.

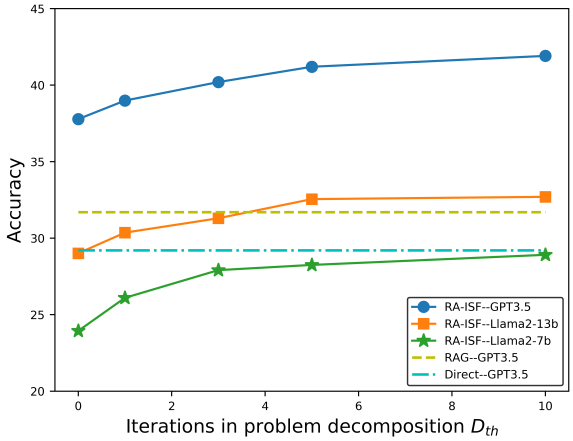


Figure 2: Question accuracy on the NQ dataset with the growth of the iteration in question decomposition D_{th}.

Relevant Module without self-knowledge judgment.

- *No Passage-Relevant Module*: After self-knowledge judgment, if the Self-Knowledge Module indicates the answer can not be addressed using the model's own knowledge, it directly decomposes the question without involving the Passage-Relevant module.
- *No Question Decomposition Module*: After assessing passage relevance through the Passage-Relevant module, if no relevant paragraphs are found, the answer is marked as "unknown," and the Question Decomposition Module does not iterate. This means the RA-ISF iteration count is set to 0.

We conducted tests on NQ, TriviaQA, and HotpotQA datasets, comparing the results with RAG, RA-ISF, and LTM methods. All experiments use GPT3.5 as the base model.

All three submodules contribute to better problem-solving performance. Table 2 presents the ablation experiment results, indicating that removing any component of RA-ISF leads to a performance decline. This suggests the importance of each component in the framework. Compared to RAG, the *No Self-Knowledge Module* variant achieves better performance by decomposing unrelated text, resulting in improved results. In contrast to the Least-to-Most prompting method, *No Self-Knowledge Module* variant achieves higher accuracy by prompting the language model with retrieved paragraphs (+6.3 on Average). When comparing Least-to-Most with variant *No Passage-Relevant Modules*, the latter first assesses self-knowledge and then iteratively decomposes information. This variant outperforms the traditional Least-to-Most paradigm. Therefore, the iterative combination of these three components not only enhances the effectiveness of RAG but also addresses certain issues (e.g., hallucinations) after retrieval and mitigates negative impacts caused by irrelevant retrieved paragraphs.

5.3 Iterations in Problem Decomposition

RA-ISF sets a threshold D_{th} to limit the iteration times of problem decomposition. Here, we experiment with different values of D_{th} on the NQ dataset of GPT-3.5 and Llama2$_{7B,13B}$. Additionally, we compare RAG and Direct Prompting with RA-ISF on GPT-3.5. The accuracy of problem-solving varies with changes in D_{th} as shown in Figure 2.

More iterations contribute to improved performance. The results indicate that as the value of D_{th} increases, the model's accuracy in answering questions improves. With the increase of D_{th}, the performance gap between RA-ISF + GPT3.5 and RAG + GPT3.5 gradually rises. More iterations also help improve the performance of small-scale LLMs in problem-solving. With the increase of D_{th}, the performance of RA-ISF + Llama2$_{13B}$ surpasses the performance of RAG and Direct Prompting on GPT3.5, and the performance on Llama2$_{7B}$ gradually approaches the accuracy of Direct on GPT3.5. This indicates that the iterative decomposition of problems contributes to enhancing the model's problem-solving ability.

Problem decomposition helps LLM to understand. The goal of problem decomposition is to ad-

dress situations where the model has on-parametric knowledge but struggles to answer due to inadequate understanding of the question. When D_{th} is relatively small, decomposing the problem helps the model extend its problem-solving approach through reasoning and derive answers. When iteration becomes larger, it indicates that after multiple rounds of knowledge retrieval and problem decomposition, no relevant passage or on-parametric knowledge has been found. This implies that the inability of the model to solve the problem is actually due to a lack of knowledge rather than insufficient understanding. At this point, further problem decomposition is less likely to be beneficial and may even introduce misleading factors, such as decomposing unrelated sub-problems to the original question, potentially reducing the accuracy of the answers.

5.4 Small Sub-model Alternatives

In this paper, we choose the Llama2$_{7B}$ model as the pretrain model when training three sub-models. Since Llama2 is a 7B LM, we also want to explore the effectiveness of using a smaller model as an intermediate component. We select the T5$_{780M}$ model for training and compare it with Llama2$_{7B}$, while the base model is GPT3.5. The accuracy comparison is shown in Table 3.

	NQ	TriviaQA	HotpotQA	StrategyQA	2WikiMHQA
Llama2$_{7B}$	40.2	76.1	46.5	75.9	36.1
T5$_{780M}$	39.6	74.8	45.8	74.7	35.3

Table 3: Evaluation for different sizes of sub-model language models in various datasets.

Training RA-ISF with a small model also yields excellent performance. When the RA-ISF method is trained on the small T5$_{780M}$ model, the accuracy of answering questions using this model is only slightly lower by one to two percentage points compared to Llama2$_{7B}$. This indicates that when training the three sub-models of RA-ISF, if there are constraints or cost limitations, using a small model like T5$_{780M}$ as the pre-trained model can still demonstrate excellent performance.

5.5 Human and Model Assessments

We conduct both manual and automated assessments to evaluate the reliability of RA-ISF. Specifically, we randomly select 40 questions from each dataset and invite 50 human annotators to assess the precision of the generated responses compared to GPT-4. For $\mathcal{M}_{know}$, if the model's judgment on whether the question can be answered using its own knowledge is consistent with GPT-4, it is considered precise. For $\mathcal{M}_{rel}$, given a question q_{new} and relevant paragraphs $\mathcal{P}_{rel}$, if the model's judgment aligns with whether the paragraphs are indeed related to the question, the judgment of$\mathcal{M}_{rel}$ is considered correct. For $\mathcal{M}_{decom}$, if both the LLM and annotators believe that each sub-question remains semantically consistent with the original question, the decomposition is considered effective.

	$\mathcal{M}_{know}$	$\mathcal{M}_{rel}$	$\mathcal{M}_{decom}$
Human	-	93.5	89.5
GPT4.0	97.0	95.0	87.0

Table 4: Human and GPT4 evaluation on the three models in RA-ISF.

The sub-modules results demonstrate high reliability. The results are shown in Table 4, indicating that both human annotators and the large model consistently agree on the effectiveness of these three models, with accuracy rates exceeding 85%. Specifically, $\mathcal{M}_{know}$ achieves an impressive accuracy of 97%, suggesting a high cognitive ability of the trained model in recognizing its own knowledge. Meanwhile, the accuracy of $\mathcal{M}_{decom}$ is slightly lower, as the task of question decomposition falls within the realm of generative tasks, where there may be multiple feasible decomposition solutions. Overall, the three sub-modules exhibit high reliability in their respective tasks.

6 Conclusions

In this paper, we introduce RA-ISF, a framework designed to enhance retrieval augmentation effects and improve performance in open-domain question answering. This approach effectively mitigates the hallucination issues that are commonly seen in traditional retrieval augmentation and question-answering tasks. Experimental results demonstrate RA-ISF's superior performance across various benchmarks, and ablation studies validate the effectiveness of sub-modules. Future research directions include further alleviating hallucination issues and improving the efficiency of the framework.

Limitation

RA-ISF innovatively introduces a three-stage iterative problem-solving strategy. However, it's important to recognize its limitations and drawbacks. Firstly, iterative problem-solving can lead to an excessive branching of issues. In particular cases, this approach might become inefficient if it continuously explores a problem and its sub-problems without finding solutions or relevant passages. Secondly, different formulations of a problem may affect the effectiveness of the problem decomposition module, leading to small differences between the number of iterations and the outcome.

Moreover, our method mainly relies on open-domain question-answering datasets. It has not been tested in specific fields such as mathematics reasoning, symbolic reasoning, or specialized areas like medicine and law. Future research could explore how it performs with these datasets. We also plan to investigate ways to use retrieval augmentation techniques more effectively and to simplify their complexity.

Ethics Statement

Our approach employs the corpus of Wikipedia and utilizes open-source datasets for training and evaluating the model. All data are openly accessible. We leverage APIs for GPT-3.5 and open-source code and weights for Llama. Due to the hallucination issue of large language models, some of the generated content may contain factual errors and reasoning errors. RA-ISF offers a potential solution based on retrieval augmentation to mitigate the hallucination problem. Our work strictly adheres to the license and policies of released LLMs and publicly available datasets.

References

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

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Hung-Ting Chen, Fangyuan Xu, Shane Arora, and Eunsol Choi. 2023. Understanding retrieval augmentation for long-form question answering.

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.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

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

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Trans. Inf. Syst.*, 38(3).

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022a. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

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

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

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)*, pages 6769–6781.

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.

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

R OpenAI. 2023. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2:3.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. *arXiv preprint arXiv:2002.09758*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

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. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274, Singapore. Association for Computational Linguistics.

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

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023b. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

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. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023a. Self-knowledge guided retrieval augmentation for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10303–10315, Singapore. Association for Computational Linguistics.

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023b. Self-knowledge guided retrieval augmentation for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10303–10315.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*.

Jingfeng Yang, Haoming Jiang, Qingyu Yin, Danqing Zhang, Bing Yin, and Diyi Yang. 2022. Seqzero: Few-shot compositional semantic parsing with sequential prompts and zero-shot models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 49–60.

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

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The*

Eleventh International Conference on Learning Representations.

A Details of Data Collection

A.1 Data Collection of $\mathcal{M}_{know}$

First, we use a QA dataset $\mathcal{D}$ for training, which includes questions q_i and their corresponding correct answers a_i, represented as $\{q_i, a_i\}_{i=1}^{|\mathcal{D}|}$. Initially, we extract the questions q_i to query the LLM $\mathcal{M}$. Through few-shot prompts and in-context learning, we enable model $\mathcal{M}$ to generate answers for each question. In this scenario, the answers generated by the model rely entirely on internal knowledge.

We compare the model-generated answer a_g with the correct answers a_i, and then categorize the questions q_i into two groups. If a_g is the same as a_i, then these questions fall into $\mathcal{Q}_{know}$, the category of problems that the model can solve on its own. Otherwise, these questions belong to $\mathcal{Q}_{unknow}$, the category of problems that the model cannot solve on its own. The specific expression is as follows:

$$q_i \in \begin{cases} \mathcal{Q}_{know} & \text{if } a_i = a_g \\ \mathcal{Q}_{unknow} & \text{if } a_i \neq a_g \end{cases} \tag{5}$$

We collect various types of supervised training data and combine them to form the model's training data, ultimately resulting in the trained dataset $\mathcal{D}^* = \{\mathcal{Q}_{know}, \mathcal{Q}_{unknow}\}$.The $\mathcal{Q}_{know}$ class comprises questions that the model $\mathcal{M}$ inherently knows, while the $\mathcal{Q}_{unknow}$ class includes questions that the model is not aware of and requires external knowledge to obtain answers.

A.2 Data Collection of $\mathcal{M}_{rel}$

For a given $\mathcal{Q}$, we input it into the retriever $\mathcal{R}$, retrieving k relevant paragraphs for each question $\mathcal{P}$ $=\{P_1, P_2, ..., P_k\}$. Subsequently, for each paragraph $P_i (i = 1, 2, ..., k)$, we traverse them one by one, querying the LLM model $\mathcal{M}$ about the relevance of the retrieved paragraph P_i to question $\mathcal{Q}$, and recording the model $\mathcal{M}$'s answer $\mathcal{A} = \{A_1, A_2, ..., A_k\}$ where $\mathcal{A} = relevant/irrelevant$ for each paragraph.

We collect various types of supervised training data and combine them to form the model's training data, ultimately resulting in the trained dataset $\mathcal{D}^* = \{\mathcal{Q} + \mathcal{P}, \mathcal{A}\}$.

A.3 Data Collection of $\mathcal{M}_{decom}$

For a given $\mathcal{Q}$, we input it into the large model $\mathcal{M}$, instructing it to decompose each question. For a given question $\mathcal{Q}$, the model breaks it down into k sub-questions, where the value of k depends on the specific question. Finally, we document the sub-questions decomposed by the model for the question, denoted as $\mathcal{Q}_{sub} = \{q_1, q_2, ..., q_k\}$.

We collect various types of supervised training data and combine them to form the model's training data, ultimately resulting in the trained dataset $\mathcal{D}^* = \{\mathcal{Q}, \mathcal{Q}_{sub}\}$.

B Details of Datasets

Natural Question (NQ) (Kwiatkowski et al., 2019) is a question-answering dataset containing 307,373 training examples, 7,830 development examples, and 7,842 test examples. Each example is comprised of a google.com query and a corresponding Wikipedia page.

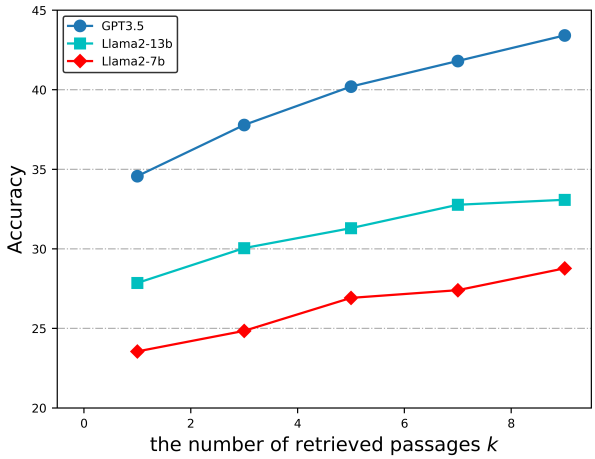
TriviaQA (Joshi et al., 2017) is a realistic text-based question-answering dataset that includes 950K question-answer pairs from 662K documents collected from Wikipedia and the web. For TriviaQA, given questions often have multiple valid answers, some of which are unsuitable for training targets, such as emoticons or spelling variations. Following Lewis et al. (2020), for TriviaQA, if a candidate answer does not appear in the top 1000 documents retrieved by the query, we filter it out.

StrategyQA (Geva et al., 2021) is a question-answering benchmark where the required reasoning steps are implicit in the question, and should be inferred using a strategy. It includes 2,780 examples, each consisting of a strategy question, its decomposition, and evidence paragraphs. Questions in StrategyQA are short, topic-diverse, and cover a wide range of strategies.

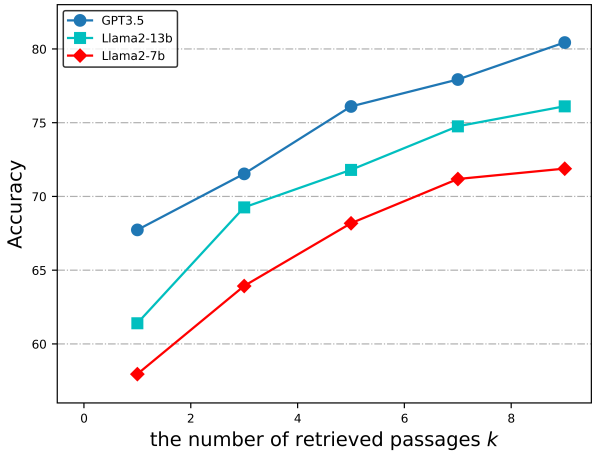
HotpotQA (Yang et al., 2018) is a multi-hop datasets from Wikipedia. The questions are diverse and not constrained to any pre-existing knowledge bases or knowledge schemas. HotpotQA is a question-answering dataset collected on the English Wikipedia, containing about 113K crowd-sourced questions that are constructed to require the introduction paragraphs of two Wikipedia articles to answer. Each question in the dataset comes with two gold paragraphs, as well as a list of sentences in these paragraphs that crowd workers identify as supporting facts necessary to answer the question.

2WikiMQA (Ho et al., 2020) utilizes both struc-

tured and unstructured data. In this dataset, evidence information is introduced, which includes reasoning paths for multi-hop questions. The evidence information serves two purposes: (i) providing a comprehensive explanation for predictions and (ii) evaluating the reasoning skills of a model. We carefully designed a pipeline and a set of templates during the generation of question-answer pairs to ensure the quality of multi-hop steps and questions.



(a) NQ



(b) TriviaQA

Figure 3: Explore the trend of question accuracy on the NQ and TriviaQA dataset with the growth of the iteration in question decomposition k, and conduct experiments on different models.

C Additional experiment:Analysis on the Number of Retrieved Passages

When the model is unable to solve a problem based solely on its own knowledge, we need to use a retriever to search for k passages. In this regard, we need to investigate the values of k. Here, we experimented with NQ and TriviaQA datasets on models including GPT-3.5, Llama2$_{7B,13B}$, with values of k set to 1, 3, 5, 7, 9. The accuracy of the questions varies with the changes in k, as shown in Figure 3.

Increasing the number of retrieved passages helps improve the accuracy of problem-solving. In general, as k increases, the accuracy of the model in answering questions continues to improve. This is because increasing the number of retrieved paragraphs helps the model find more auxiliary knowledge, enhancing the likelihood of identifying relevant articles to the question and thereby improving the accuracy of question answering.

Further observation reveals that there is a noticeable improvement in accuracy as k increases from 1 to 5; however, the improvement becomes less apparent when k increases from 5 to 9. This is because, with the increase in the number of retrieved paragraphs, the model seems to have access to more paragraphs to assist in answering questions. However, in reality, the previously retrieved articles might have been sufficient for the model to identify the correct answers. Continuing to increase the number of retrievals could result in finding irrelevant articles, which would eventually be filtered out by $\mathcal{M}_{rel}$. Therefore, the contribution to the accuracy of the question is limited.

D Examples

StrategyQA

Question: Did Snoop Dogg refuse to make music with rival gang members?
Answer: ["false"]

Iteration 1

$\mathcal{M}_{know}$
Instruction: Can you use your own knowledge base to solve this problem? Answer yes if you know, no if you need additional knowledge base to solve it.
The problem is: Did Snoop Dogg refuse to make music with rival gang members?
Output: No
$\mathcal{M}_{rel}$
(Use retriever to find 5 passages...)
Instruction: I will provide you with five additional pieces of knowledge based on the search for this question. Please assess whether these five paragraphs are relevant to the question and sufficient to answer it. If they are, please tell me what the relevant paragraphs are; if not, please answer "No."
[1] Calvin Cordozar Broadus Jr. ; born October 20, 1971), known professionally as Snoop Dogg (previously Snoop Doggy Dogg and briefly Snoop Lion)...
[2] Broadus' debut studio album, Doggystyle (1993), produced by Dr. Dre, was released by Death Row Records and debuted at number one on the Billboard 200...
[3] In 1993, Broadus was charged with first-degree murder for the shooting of a member of a rival gang who was actually killed by Snoop's bodyguard.Broadus was...
[4] While recording Doggystyle in August 1993, Broadus was arrested and charged with first-degree murder in connection with the shooting death of Philip Woldermariam...
[5] In 2002, he released the album Paid tha Cost to Be da Bo, on Priority/Capitol/EMI, selling over 1,310,000 copies. The album featured the hit singles...
Output: No.
$\mathcal{M}_{decom}$
Instruction: Please break down this question into several sub-questions and list them.
Output:
1. What is the name of the gang that Snoop Dogg is part of?
2. Which gang is the rival of the gang Snoop Dogg is associated with?
3. In Snoop Dogg's song "California Vacation," which rapper did he collaborate with?
4. Did Snoop Dogg refuse to make music with members of rival gangs?

Iteration 2

Use sub-models to answer these sub-questions...
Sub-answer:
1. Answer: American rapper Snoop Dogg is a member of the Crips gang.
2. Answer: The Crips are enemies of their rival gang, The Bloods.
3. Answer: Snoop Dogg collaborates with Xzibit and The Game on the song "California Vacation."
4. Answer: No, Snoop Dogg did not refuse to make music with rival gang members
Instruction: Base on the sub-question answer, give the answer to the origin question.
Final answer: No, Snoop Dogg did not refuse to make music with rival gang members, as evidenced by his collaboration with The Game on the song "California Vacation."

Table 5: An example from the StrategyQA dataset.

NQ
Question: When was the immigration reform and control act passed? **Answer**: ["November 6, 1986"]
Iteration 1
$\mathcal{M}_{know}$ **Instruction**: Can you use your own knowledge base to solve this problem? Answer yes if you know, no if you need additional knowledge base to solve it. The problem is:When was the immigration reform and control act passed? **Output**: Yes. **Instruction**: Give the answer to the question. **Final answer**: The Immigration Reform and Control Act (IRCA) was passed on November 6, 1986.

Table 6: An example from the NQ dataset.

TriviaQA
Question: From which country did Angola achieve independence in 1975? **Answer**: ["Portogało","Republic of Portugal"]
Iteration 1
$\mathcal{M}_{know}$ **Instruction**: Can you use your own knowledge base to solve this problem? Answer yes if you know, no if you need additional knowledge base to solve it. The problem is: From which country did Angola achieve independence in 1975? **Output**: Yes. **Instruction**: Give the answer to the question. **Final answer**: Angola achieved independence from Portugal in 1975.

Table 7: An example from the TriviaQA dataset.

HotpotQA
Question: Which private research university is located in Chestnut Hill, Massachusetts Boston College or Stanford University? **Answer**: ["Boston College"]
Iteration 1
$\mathcal{M}_{know}$ **Instruction**: Can you use your own knowledge base to solve this problem? Answer yes if you know, no if you need additional knowledge base to solve it. The problem is: Which private research university is located in Chestnut Hill, Massachusetts Boston College or Stanford University? **Output**: Yes. **Instruction**: Give the answer to the question. **Final answer**: Boston College is the private research university located in Chestnut Hill, Massachusetts.

Table 8: An example from the HotpotQA.

2WikiMultihopQA
Question: Who is the mother of the director of film Polish-Russian War (Film)? **Answer**: [“Magorzata Braunek”]
Iteration 1
$\mathcal{M}_{know}$ **Instruction**: Can you use your own knowledge base to solve this problem? Answer yes if you know, no if you need additional knowledge base to solve it. The problem is: Who is the mother of the director of film Polish-Russian War (Film)? **Output**: No
$\mathcal{M}_{rel}$ (Use retriever to find 5 passages...) **Instruction**: I will provide you with five additional pieces of knowledge based on the search for this question. Please assess whether these five paragraphs are relevant to the question and sufficient to answer it. If they are, please tell me what the relevant paragraphs are; if not, please answer “No.” [1] Polish-Russian War (Wojna polsko-ruska) is a 2009 Polish film directed by Xawery Żuławski based on the novel Polish-Russian War under the white-red flag by Dorota Masłowska... [2] Xawery Żuławski (born 22 December 1971 in Warsaw) is a Polish film director. In 1995 he graduated National Film School in Łódź. He is the son of actress Małgorzata Braunek and director Andrzej Żuławski... [3] After an argument in a bar owned by “Left” (Michał Czernecki) "Strong" meets a “Gothgirl” Angelica (Maria Strzelecka) at night, an aspiring poet dressed in black, also a virgin and pessimist, for whom “suicide is a piece of cake”... [4] “Strong” follows Magda. He turns up at the town festival, where she takes part in a miss competition. He cannot reach her, but instead he meets... [5] Production The film was shot between May 6 and 18 June 2008 in locations of Warsaw, Wejherowo, Sopot and Gdynia outskirts. The film premiered on... **Output**:Relevant paragraphs: [2] Xawery Żuławski is the director of the film "Polish-Russian War (Wojna polsko-ruska)" and is the son of actress Małgorzata Braunek and director Andrzej Żuławski. **Instruction**: Using the knowledge from the relevant paragraphs, give the answer to the question. **Final answer**: The mother of the director of the film “Polish-Russian War (Wojna polsko-ruska)” is actress Małgorzata Braunek.

Table 9: An example from the 2WikiMultihopQA dataset.