HIRAG: Hierarchical-Thought Instruction-Tuning Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

Retrieval-augmented generation (RAG) has become a fundamental paradigm for addressing the challenges faced by large language models in handling real-time information and domain-specific problems. Traditional RAG systems primarily rely on the in-context learning (ICL) capabilities of the large language model itself. Still, in-depth research on the specific capabilities needed by the RAG generation model is lacking, leading to challenges with inconsistent document quality and retrieval system imperfections. Even the limited studies that fine-tune RAG generative models often *lack a granular focus on RAG task* or *a deeper utilization of chain-of-thought processes*. To address this, we propose that RAG models should possess three progressively hierarchical abilities (1) Filtering: the ability to select relevant information; (2) Combination: the ability to combine semantic information across paragraphs; and (3) RAG-specific reasoning: the ability to further process external knowledge using internal knowledge. Thus, we introduce our new RAG instruction fine-tuning method, Hierarchical-Thought Instruction-Tuning Retrieval-Augmented Generation (HIRAG) incorporates a "think before answering" strategy. This method enhances the model's open-book examination capability by utilizing multi-level progressive chain-of-thought. Experiments show that the HIRAG training strategy significantly improves the model's performance on datasets such as RGB, PopQA, MuSiQue, HotpotQA, and PubmedQA.

1 Introduction

Retrieval Augmentation Generation (hereafter referred to as RAG) helps large language models (LLMs)(OpenAI et al., 2024) reduce hallucinations(Zhang et al., 2023) and access real-time data by incorporating an information retrieval component. While LLMs often use in-context learning(Gao et al., 2024) for generation, practical issues such as low-quality or poorly ranked retrieved documents can hinder RAG's effectiveness. These challenges emphasize the need for instruction-tuning tailored to RAG tasks. Fine-tuning generative models specifically for RAG improves their ability to integrate retrieved information(Zhang et al., 2024)(Yu et al., 2024), resulting in more accurate and contextually relevant responses compared to general-purpose models.

RAFT(Zhang et al., 2024) enhances model performance in domain-specific RAG tasks by introducing distractor documents during training. EvidenceRAG(Schimanski et al., 2024) improves large language models in evidence-based question answering by incorporating an indexing task, enhancing their ability to accurately cite and reflect source information. RankR AG(Yu et al., 2024) employs a two-stage training process to simultaneously optimize the context ranking and answer generation capabilities of large language models (LLMs) in RAG tasks.

Despite significant research efforts on RAG-specific generative models, several issues remain.

- **Lack of Granular RAG Task Focus**: Researchers have primarily concentrated on fine-tuning RAG models without enhancing their capabilities through more granular RAG tasks, limiting the potential to strengthen RAG abilities effectively.

- **Lack of Deep Utilization of CoT**: Although there have been proposals to integrate chain-of-thought (CoT) reasoning into the training process to enhance model accuracy(Wei et al., 2023), these methods are not specifically designed for RAG scenarios. Even in the rare cases where RAG models do incorporate CoT(Zhang et al., 2024), the integration is neither sufficiently deep nor granular, leaving the model to utilize documents at its discretion.

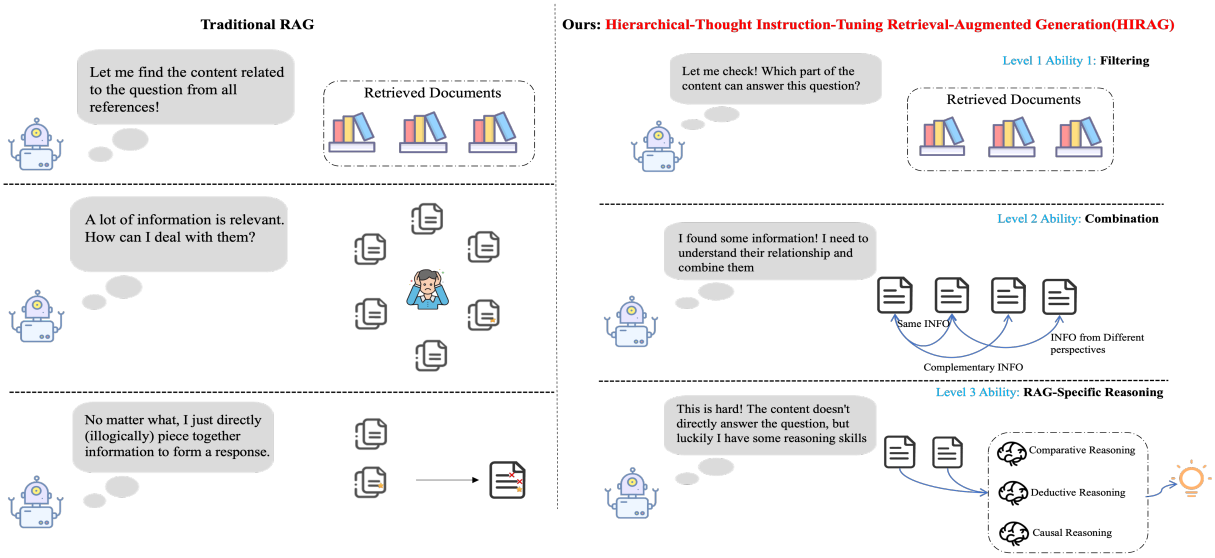

Figure 1: Traditional RAG methods have primarily focused on retrieving relevant information, with less emphasis on its effective utilization. We propose a method that enhances model performance in complex RAG scenarios by developing three progressive capabilities.

As a result, CoT has not yet fully realized its potential.

Thus, We introduce a new RAG Instruction Tuning method: Hierarchical-Thought Instruction-Tuning Retrieval-Augmented Generation (**HIRAG**) adapting to complex RAG scenarios and propose that when fine-tuning RAG generation models, we focus on three progressively hierarchical abilities shown in Figure 1: **Filtering:** The ability that LLM filters out noise and selects the direct information. **Combination:** The ability of LLMs to merge, integrate, and summarize multiple pieces of useful information. **RAG-Specific Reasoning:** The capability refers to the ability to answer a question by making implicit or explicit inferences based on the information in the documents when the relevant information is not directly provided.

To better achieve these three capabilities, a "think before answering" approach based on progressively hierarchical thought has been introduced.

The contributions of this work are summarized as follows:

- We propose three progressive hierarchical capabilities that a RAG model requires: filtering, combination, and RAG-specific reasoning to enhance the granularity and specificity of RAG tasks when dealing with complex scenarios.
- We introduce **HIRAG**, a fine-tuning strategy that utilizes different thoughts for various tasks. This approach constructs a progressive chain of thought, enabling the model to learn from easier to more complex tasks, thereby significantly enhancing its performance in RAG scenarios.
- Extensive experiments were conducted on six datasets, including the RAG-specific benchmark, single-hop open-domain data, multi-hop open-domain data, and domain-specific data. Our model significantly outperforms the current state-of-the-art models. We also conducted experiments on Chinese datasets, confirming the robustness of our approach. Furthermore, ablation studies demonstrate that the training tasks for the three capabilities contribute to the performance of HIRAG, and we explored the optimal data ratio.

2 Related Work

Retrieval-Augmented Generation (RAG). Retrieval-Augmented Generation (RAG) (Guu et al., 2020) has become a fundamental paradigm for reducing hallucinations and improving performance domain-specific problems(Asai et al., 2023a)(Lewis et al., 2021). The main problem RAG faces is that low quality of article(Liu et al., 2023) and the model is vulnerable to noise interference in the context(Shi et al., 2023). Correspondingly, the current mainstream solution relies on the upgrading of retrieval modules(Shi

et al., 2024) and the training of fixed-document generation models to improve its effect(Wang et al., 2024)(Gao et al., 2024)

Upgrading of retrieval modules. From the perspective of retrieval methods, some studies have enhanced the quality of context by employing multi-stage retrieval reasoning (Asai et al., 2023b)(Gan et al., 2024), while others have designed adaptive retrieval modules that allow models to adjust retrieval behavior according to different tasks (Jeong et al., 2024). In terms of question understanding, some studies have improved search queries by rewriting, decomposing, and disambiguating(Chan et al., 2024). After retrieving articles, incorporating a ranking module can significantly enhance the final generation outcome (Glass et al., 2022)(Ram et al., 2023). RankRAG effectively integrates the ranking module with the generation module (Yu et al., 2024). These approaches have effectively improved the quality of retrieved articles in RAG systems. However, there is no such thing as a perfect context, and the generative model needs to be capable of handling contexts in various situations.

Training Methods for Generative Models. ChatQA(Liu et al., 2024)(Xu et al., 2024) enhances the model's zero-shot dialogue capabilities through synthetic data and a two-stage instruction fine-tuning approach. In terms of identifying noisy documents, RAFT(Zhang et al., 2024) improves the model's ability to recognize and disregard irrelevant information by introducing distractor documents and employing the Chain-of-Thought (COT) method. In contrast, InstructRAG(Wei et al., 2024) achieves this by explicitly learning the denoising process. EvidenceRAG(Schimanski et al., 2024) introduces an indexing task to enhance the reliability and traceability of large language models (LLMs) in evidence-based question answering. However, the context is complex and variable, merely filtering out noise and finding relevant documents is insufficient. Our work, starting from complex context scenarios, proposes three progressive model capabilities and effectively enhances these capabilities using the "think before answering" strategy.

3 HIRAG

In this section, we introduce our RAG-focused instruction tuning methods: HIRAG (Hierarchical-Thought Instruction-Tuning Retrieval-Augmented Generation), which incorporates a "think before answering" strategy to enhance progressively RAG abilities: filtering, combination, and RAG-pecific reasoning.

3.1 Progressively Hierarchical RAG Abilities

To address the complex and diverse scenarios in RAG, we propose three progressive abilities required for generative models and enhance each of these capabilities using the COT method. Below, we provide a detailed description of these three capabilities.

3.1.1 Filtering Abilities

Filtering is the ability of LLMs to filter out noise and select the direct information that is helpful for answering questions from multiple documents or chunks of a single document. In the filtering capability, we focus on identifying relevant information in response to a query. During training, we use different types of noise and irrelevant information to improve the model's filtering skills. Noise information includes data related to the main topic, such as terms like "hospital" and "doctor" (which are thematically linked), or "output value in 2024" and "output value in January 2024" (which are about a similar subject). On the other hand, irrelevant information refers to data that is completely unrelated to the question's main point.

3.1.2 Combination Abilities

In addition to its filtering capabilities, the model has developed the ability to identify individual information points. Taking this a step further, combination is the capability of synthesizing and amalgamating all pertinent information across multiple documents to generate direct answers. This process involves a comprehensive gathering and integration of data to provide thorough responses. From the perspective of entities and attributes, this can be categorized into two primary types: one in which a single entity possesses multiple qualifying attribute values, and another where multiple entities each have their own attribute values. For example, "What are the hobbies of Tom" and "What does Tom like and what does Anny hate"

In this context, the model's ability to synthesize information represents a significant advance in information retrieval and processing. It not only underscores the model's proficiency in pinpointing discrete information but also highlights its potential in constructing a cohesive narrative or answer from disparate sources.

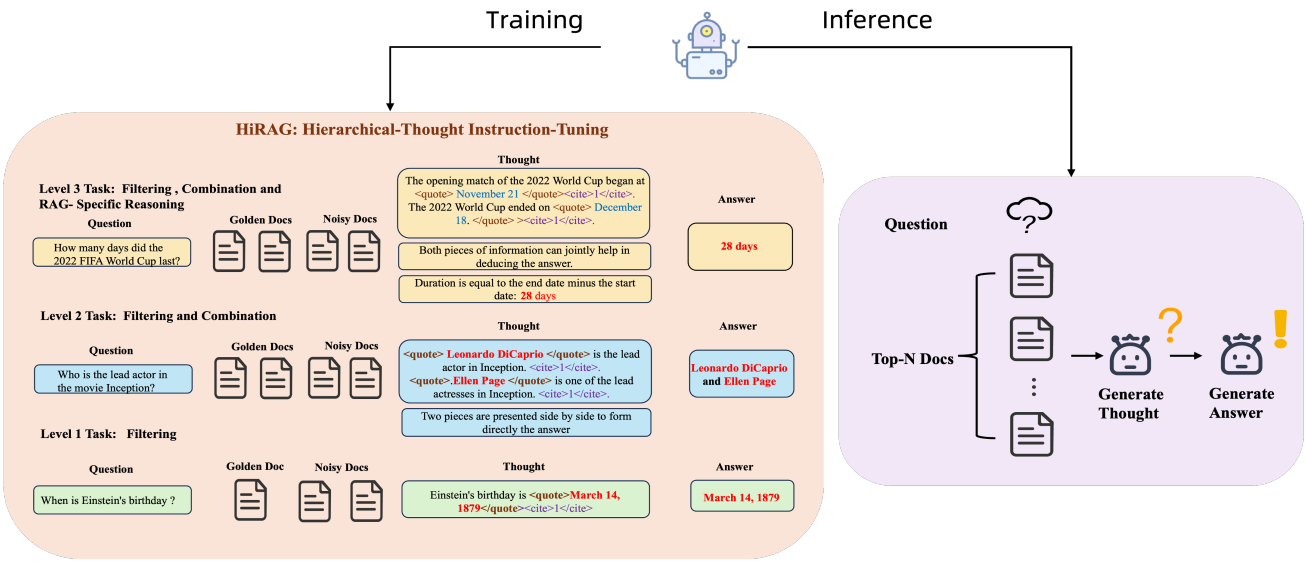

Figure 2: Overview of the HIRAG strategy: We design three progressively challenging tasks while establishing an incremental chain of thought to enhance the model's capabilities in RAG scenarios. As shown in the "Thought" section of the illustration, the thought processes differ across tasks of varying difficulty levels, ranging from basic information filtering to document combination and, finally, to reason.

3.1.3 RAG-Specific Reasoning Abilities

Once the model has developed both filtering and combination capabilities, it can identify all information relevant to a given question. If it still cannot provide an answer at this stage, it becomes necessary to engage in reasoning processes in conjunction with the documents to arrive at a solution. This step-by-step reasoning chain can significantly enhance the quality of responses generated by large language models when faced with complex reasoning tasks (Wei et al., 2023). Within the RAG framework, the integration of filtering and combination techniques empowers language models to identify and compile all pertinent information within the flow of ideas. Consequently, the model is expected to leverage this aggregated information for logical reasoning to produce accurate answers. Therefore, enhancing RAG-related reasoning within the reasoning chain is crucial.

RAG-specific reasoning that primarily involves the utilization and processing of document content, which can be categorized into explicit and implicit document reasoning. Explicit document reasoning involves multi-hop reasoning, combining information from multiple or single documents to reach a conclusion. Implicit document reasoning, on the other hand, integrates information mentioned in documents with the model's internal knowledge to infer the final result. For instance, if a document states that mammals possess a characteristic A, and the question is whether monkeys have characteristic A, implicit reasoning is required: namely, recognizing that monkeys are mammals.

From the perspective of reasoning categories, the following types can be identified:

i. Comparative Reasoning. The question involves comparing several items, and the documents do not directly provide an answer but offer various attributes or definitions of the items. Specific example as Appendix Figure 7.

ii. Deductive Reasoning. The question inquires about the attributes of A1, and the documents state that A1 belongs to A (major premise) and provide the attributes of A (minor premise). Through this deductive reasoning, the attributes of A1 can be inferred. Specific example as Appendix Figure 9.

iii. Causal Reasoning. This involves identifying the implicit or explicit causal relationships within the documents to find the cause or effect. Specific example as Appendix Figure 8.

3.2 Training Data Construction

HIRAG proposes a novel and effective supervised fine-tuning approach for enhancing generation ability in RAG 3.1. The main approach utilizes a progressive chain-of-thought (CoT) method and follows the previous work(Zhang et al., 2024) by using special tokens <|REASON|> and <|ANSWER|> to control the generation of thought and answer. As illustrated in Figure 2, the process of training and

inference is depicted. The specific data construction strategy and pipeline are outlined as follows:

3.2.1 Strategies

The effectiveness of a generative model in RAG significantly depends on the quality and composition of its training data with different documents. To enhance the model's performance, especially in environments where distinguishing relevant information from distractor documents is crucial, carefully curated training data is essential. Here, we outline three strategies for constructing such datasets:

i. Progressive RAG Tasks. As detailed in Section 3.1, key RAG capabilities include filtering, combination, and document-related reasoning. To enhance these abilities, we designed training tasks in progressive stages: filtering, filtering with combination, and filtering with combination and RAG-Specific reasoning. This method helps the RAG model excel in selecting relevant information, integrating it, and reasoning about it within document contexts.

ii. Distractor Documents. In practical RAG scenarios, not every retrieved document is useful. Introducing noisy documents in training is crucial for helping the model learn to distinguish relevant from irrelevant information, thereby improving its ability to handle noise and generate accurate responses.

iii. Chain-of-Thought for RAG. CoT reasoning enhances the model's ability to handle complex tasks by introducing intermediate reasoning steps, improving accuracy and interpretability(Wei et al., 2023). Training with Chain-of-thought also works within RAG instruction tuning process(Zhao et al., 2024), requiring thought processes specific to RAG, such as identifying relevant information from documents. Direct quotations (<quote>) and source citations (<cite>) further enhance RAG model performance, aiding in the accurate retrieval and use of information(Bezerra and Weigang, 2025)(Schimanski et al., 2024). If a document's content to be quoted is extensive, the model should summarize key points and provide a citation (<cite>) for reference.

3.2.2 Pipeline

Based on these strategies, we construct a pipeline for training data generation.The specific algorithms used for data construction are provided in the Appendix A.1.

i. Source Data Acquisition For data acquisition, we utilized a range of datasets (training set) containing RAG documents as our data source, without incorporating their QA components, including HotpotQA and PubMedQA. Besides these, we also acquired documents sourced from Wikipedia, and those generated using GPT-4-turbo or Qwen-MAX based on certain entity triples. The purpose of this approach is to gather similar documents, which can then be used to select both golden documents and distractor documents.

ii. Query Generation When documents are fixed, variations in the query can determine which RAG task—filtering, combination, or reasoning—is being focused on. For instance, if a document contains a person's biography, asking about their activities in a specific year is a filtering task. However, asking about their activities at a certain age involves reasoning, as it requires calculating the year based on the age since the document may not provide this information directly.

To effectively address different RAG tasks, we use various templates (as detailed in the Appendix B) to create queries with GPT-4-turbo or Qwen-MAX tailored for different RAG tasks.

iii. Thought&Answer Generation Once the documents and query are obtained, the next step is to create a thought process and answer based on the query and the key document. This involves using the thought process to identify the key document by applying certain rules (citing documents). It is essential to guide the model through a logical sequence, using different parts of the document step by step to reach the answer. Although the templates for generating thoughts and answers are generally similar across tasks, a few specific guidelines should be followed: (1) **Filtering:** Identify a specific piece of information within the document. (2) **Combination:** Gather all pieces of information within the document that meet the specified criteria. (3) **RAG-specific Reasoning:** Construct a reasoning pathway based on the previous steps to aid in forming a comprehensive thought.

iv. Validation After generating samples that include a query, document, thought process, and answer, it is crucial to perform a post-verification process on each sample. This serves two primary purposes: (1) **Task Definition Compliance:** Ensure that each sample adheres to the specific task definitions. This step helps identify and remove any samples that do not meet the required criteria, thereby preventing them from affecting future experimental analyses. (2) **Answer Accuracy:** As-

sess the correctness of the provided answer. This step is crucial for confirming that the answers are not only accurate but also consistently reproducible, thus ensuring the reliability of the sample data.

v. Incorporating Noisy Documents. To improve the model's ability to handle long texts and resist noise, it's important to include extra interference documents as noise. These challenge the model to differentiate between relevant and irrelevant information. Additionally, shuffling the documents (20%-30% of all samples) increases the model's robustness by forcing it to rely on its ability to identify patterns and key information, rather than on predetermined sequences.

4 Experiments

4.1 Experimental Setup

Datasets. We primarily evaluate the model's ability in question-answering scenarios given references, considering three types of tasks in our experiments: (1) **RAG-specific Benchmark**, which mainly uses the Noise Robustness(RGB-noise) and Information Integration(RGB-int) datasets from RGB(Chen et al., 2023). (2) **Open-Domain QA**, which mainly includes PopQA(Mallen et al., 2023), HotpotQA(Yang et al., 2018), and MuSiQue(Trivedi et al., 2022). PopQA is a single-hop QA task, while HotpotQA and MuSiQue are multi-hop tasks. (3) **Domain-specific QA**, mainly using PubMedQA(Jin et al., 2019), which is a question-answering dataset in the medical domain. We employ accuracy as the primary evaluation metric and additionally use Exact Match (EM) for the PubmedQA dataset. Throughout these experiments, we conduct zero-shot evaluations.

Baselines. We consider the following baselines: (1) Large-scale Models with RAG, including proprietary models, such as GPT-4(OpenAI et al., 2024) and GPT-4o-mini, through the official OpenAI APIS. Concurrently, we employ large-scale Llama models, such as Llama3-70B-Instruct. (2)Baseline Models with RAG, where we evaluate robust publicly available instruction-tuned LLMs such as Llama2-7B-Chat, Llama2-13B-Chat(Touvron et al., 2023), Llama3-8B-Instruct(Grattafiori et al., 2024). (3) RAG-specific baselines, including Self-RAG, RQ-RAG, ChatQA-1.5, and ChatQA-2.0. For these methods, we use publicly released model weights and prompts provided by their respective works. Additionally, for RankRAG and RAFT, we select parts of their evaluation results that align with our assessment for comparison. Note that since RGB is evaluated in a fixed document scenario, we do not assess methods that optimize the retrieval process.

Implementation Details. During the training stage, we employ Llama2-7B and Llama3-8B as the backbone models. For the inference stage, we utilize vLLM(Kwon et al., 2023) for accelerated inference and consistently use Contriever-MS(Izacard et al., 2022) as the retriever. More details can be found in the Appendix A.2.

4.2 Main Results

HIRAG outperforms the base models. We observed the performance of HIRAG across different tasks, and it consistently surpassed the similarly-sized Llama models. Notably, on specific datasets such as PopQA and PubMedQA, HIRAG is capable of achieving results that are comparable to, or even exceed, those of more powerful models, including the open-source Llama-70B-Instruct and the closed-source GPT-4 and GPT-4o-mini.

HIRAG is better than existing RAG-Specific models. As shown in Table 1, the HIRAG model exhibits superior overall performance compared to existing RAG methods. Specifically, with an 8B scale model, our model achieved substantial improvements of 2.5, 2.4, and 7.7 percentage points over the current state-of-the-art models on the PopQA, HotpotQA, and Musique datasets, respectively. In domain-specific tasks, when using Llama2 as the baseline model, our model exhibited significantly superior performance compared to existing models.

4.3 Experiment Results on Chinese Benchmarks

To enhance the robustness of the experimental results, we conducted experiments on a Chinese dataset. Table 2 presents the performance of HIRAG on the Chinese Benchmarks. We note that on the Chinese RGB evaluation dataset, HIRAG significantly outperforms the base model of the same size. Furthermore, compared to larger models, HIRAG surpasses Qwen2.5-32B and approaches the performance level of Qwen2.5-72B.

Table 1: Zero-shot performance of HIRAG and baselines on 6 datasets. Results unavailable in public reports are marked as "–". **Bold** numbers and underline numbers indicate the best and second-best experimental results among small-scale models, and gray-colored bold text denotes the best large-scale model when it outperforms all small-scale models.

Dataset Metric	RGB-noise Acc	RGB-int Acc	PopQA Acc	HotpotQA Acc	MuSiQue Acc	PubMedQA EM
Large-scale Models with RAG						
GPT-4	98.0	79.0	63.3	51.5	32.3	58.3
GPT-4o-mini	**99.0**	**86.0**	64.2	**54.2**	**37.7**	56.8
Llama3-70B-Instruct	98.7	83.0	**67.2**	53.2	36.9	65.4
Baseline Models with RAG						
Llama2-7B-Chat	67.3	42.0	51.4	38.2	21.1	38.6
Llama2-13B-Chat	73.6	60.0	61.2	39.9	23.3	36.4
Llama3-8B-Instruct	87.7	56.0	62.0	41.9	18.9	63.6
RAG-Specific Models with RAG						
RQ-RAG (Llama2-7B)	-	-	56.4	43.5	17.3	56.2
Self-RAG (Llama2-7B)	-	-	55.3	35.7	10.7	49.4
RAFT(Llama2-7B)	-	-	-	-	-	73.3
ChatQA-1.5 (Llama3-8B)	90.3	61.0	54.5	46.8	20.1	55.1
ChatQA-2.0 (Llama3-8B)	91.6	59.0	58.5	41.9	16.2	49.2
RankRAG(Llama3-8B)	-	-	64.1	-	-	-
HIRAG(Llama2-7B)	83.7	50.0	64.9	47.2	21.8	73.7
HIRAG(Llama3-8B)	**94.6**	**66.0**	**66.6**	**49.2**	**27.8**	**74.6**

Table 2: Results of HIRAG and Qwen-2.5 of different sizes on the RGB-int and RGB-noise Chinese datasets.

Dataset-zh Metric	RGB-noise Acc	RGB-int Acc
Qwen2.5-7B	86.3	71.0
Qwen2.5-14B	95.0	73.0
Qwen2.5-32B	89.7	77.0
Qwen2.5-70B	**96.0**	**84.0**
HIRAG(Qwen2.5-7B)	95.3	78.0

4.4 Ablation Study on HIRAG

To evaluate the impact of three progressively complex datasets on model performance, we conducted experiments with varying data ratios on both Chinese and English datasets. To ensure fairness, the only variable among the models was the data ratio, with the total amount of data kept constant. The results for the English experiments are presented in Table 3, and the results for the Chinese experiments are shown in Table 4. From the experimental results, it is evident that the introduction of combination and RAG-specific reasoning datasets has led to an enhancement in the model's overall capabilities. This improvement is particularly pronounced in the Chinese RGB-int dataset. Additionally, we observed that increasing the proportion of composite and RAG-specific reasoning data significantly improves performance on the RGB-int dataset, while maintaining comparable performance on the RGB-noise dataset. Ultimately, we selected a model trained with a 1:2:2 ratio of Filtering, Combination, and RAG-specific reasoning data, which demonstrated the best overall performance.

Table 3: The results of the ablation experiments using Llama-8B are presented. Here, i:j:k denotes the ratio of Filtering, Combination, and RAG-specific Reasoning datasets, respectively.

Dataset-en Metric	RGB-noise Acc	RGB-int Acc
$\text{HIRAG}_{1:0:0}$	94.3	48.1
$\text{HIRAG}_{1:1:0}$	94.6	58.5
$\text{HIRAG}_{1:1:1}$	**96.6**	58.5
$\text{HIRAG}_{2:1:1}$	96.3	58.5
$\text{HIRAG}_{1:2:1}$	94.3	61.3
$\text{HIRAG}_{1:1:2}$	95.6	62.2
$\text{HIRAG}_{1:2:2}$	94.6	**66.0**

4.5 Case Study

Table3 shows the specific case analysis on the MuSiQue dataset. When multiple documents have relevant information at the same time, it shows HIRAG's excellent ability in cross-document information integration and document reasoning.

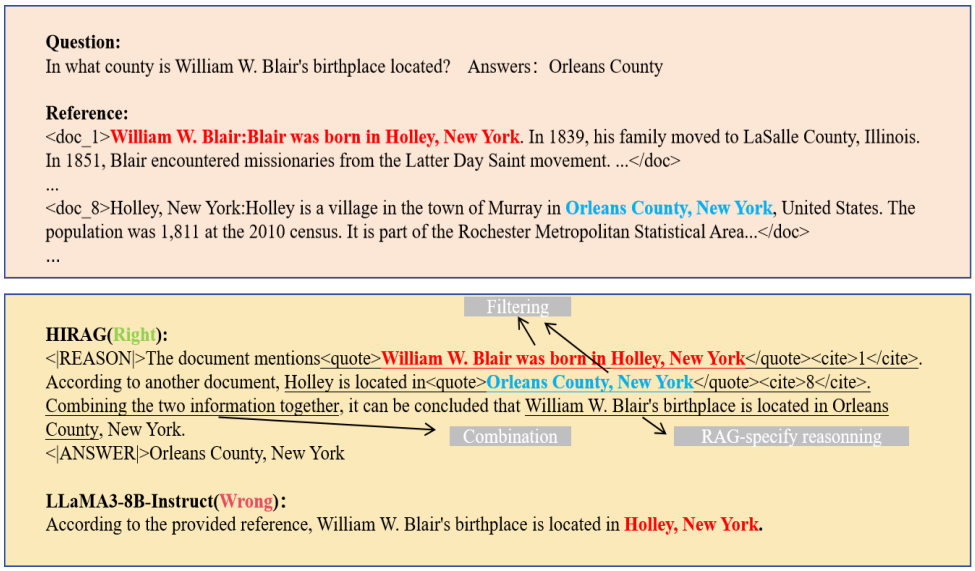

Figure 3: A case study on Musique. illustrating the effectiveness of HIRAG-8B over Llama-8B-Instruct.

Table 4: The results of the ablation experiments using Qwen2.5-7B.

Dataset-zh Metric	RGB-noise Acc	RGB-int Acc
$\text{HIRAG}_{1:0:0}$	94.0	53.0
$\text{HIRAG}_{1:1:0}$	94.7	63.0
$\text{HIRAG}_{1:1:1}$	94.7	74.0
$\text{HIRAG}_{2:1:1}$	93.7	77.0
$\text{HIRAG}_{1:2:1}$	92.7	77.0
$\text{HIRAG}_{1:1:2}$	93.3	75.0
$\text{HIRAG}_{1:2:2}$	**95.3**	**78.0**

5 Conclusion

In this work, we present HIRAG, a novel instruction tuning method specifically designed for RAG (Retrieval-Augmented Generation) models. This method provides a more granular enhancement of RAG's three core capabilities: filtering, combination, and RAG-specific reasoning. This is accomplished by employing a hierarchical "chain of thought" (CoT) approach to improve the model's performance in open-book examinations. This approach demonstrates that HIRAG exhibits strong performance across a variety of document-based question-answering benchmarks, achieving outcomes that are not only competitive with but in some instances, exceed those of much larger models. In the future, we will focus more on the reasoning aspect of the chain of thought. Using stack-based thought processes or reinforcement learning, we aim to enhance the diversity and coherence of reasoning pathways to achieve better performance in RAG scenarios.

Limitations

Heavy Dependence on Documents: Our method performs exceptionally well when the answers are present within the documents. However, its performance declines when the documents only provide supplementary information without containing the exact answers. Further experimentation and adjustments are required to optimize the model's ability to generate direct answers in such scenarios.

Domain Knowledge Enhancement: Training RAG models solely on general knowledge is insufficient for performance in specialized domains. Future work could consider a two-stage training approach. In the first stage, we conduct general domain RAG fine-tuning. Building on the general reasoning abilities established in the first phase, this phase involves setting different paths of thought and stacked thoughts based on various intents within the vertical domains.

Reliance on Synthetic Data: Our approach leverages more powerful models, such as GPT-4, for data synthesis. Despite the inclusion of a data validation step in our generation pipeline, the synthetic data may still contain errors that can impact downstream tasks. Moreover, the frequent invocation of GPT-4 at multiple stages incurs substantial computational costs. Therefore, developing cost-effective methods to generate highly accurate synthetic data remains a significant challenge.

References

Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. 2023a. Retrieval-based language models and applications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts)*, pages 41–46, Toronto, Canada. Association for Computational Linguistics.

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

Yuri Facanha Bezerra and Li Weigang. 2025. Llmquoter: Enhancing rag capabilities through efficient quote extraction from large contexts. *Preprint*, arXiv:2501.05554.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation. *Preprint*, arXiv:2404.00610.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2023. Benchmarking large language models in retrieval-augmented generation. *Preprint*, arXiv:2309.01431.

Chunjing Gan, Dan Yang, Binbin Hu, Hanxiao Zhang, Siyuan Li, Ziqi Liu, Yue Shen, Lin Ju, Zhiqiang Zhang, Jinjie Gu, Lei Liang, and Jun Zhou. 2024. Similarity is not all you need: Endowing retrieval augmented generation with multi layered thoughts. *Preprint*, arXiv:2405.19893.

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. *Preprint*, arXiv:2312.10997.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Rajaram Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2g: Retrieve, rerank, generate. *Preprint*, arXiv:2207.06300.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, and Aiesha Letman et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *Preprint*, arXiv:2002.08909.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Preprint*, arXiv:2112.09118.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *Preprint*, arXiv:2403.14403.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *Preprint*, arXiv:1909.06146.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *Preprint*, arXiv:2309.06180.

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. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Nelson F. Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. *Preprint*, arXiv:2304.09848.

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. *Preprint*, arXiv:2401.10225.

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. *Preprint*, arXiv:2212.10511.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, and Diogo Almeida et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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

Tobias Schimanski, Jingwei Ni, Mathias Kraus, Elliott Ash, and Markus Leippold. 2024. Towards faithful and robust llm specialists for evidence-based question-answering. *Preprint*, arXiv:2402.08277.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. *Preprint*, arXiv:2302.00093.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384, Mexico City, Mexico. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Preprint*, arXiv:2108.00573.

Boxin Wang, Wei Ping, Lawrence McAfee, Peng Xu, Bo Li, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Instructretro: Instruction tuning post retrieval-augmented pretraining. *Preprint*, arXiv:2310.07713.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2024. Instructrag: Instructing retrieval-augmented generation via self-synthesized rationales. *Preprint*, arXiv:2406.13629.

Peng Xu, Wei Ping, Xianchao Wu, Chejian Xu, Zihan Liu, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Chatqa 2: Bridging the gap to proprietary llms in long context and rag capabilities. *Preprint*, arXiv:2407.14482.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *Preprint*, arXiv:1809.09600.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *Preprint*, arXiv:2407.02485.

Tianjun Zhang, Shishir G. Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024. Raft: Adapting language model to domain specific rag. *Preprint*, arXiv:2403.10131.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *Preprint*, arXiv:2309.01219.

Yuetong Zhao, Hongyu Cao, Xianyu Zhao, and Zhijian Ou. 2024. An empirical study of retrieval augmented generation with chain-of-thought. *Preprint*, arXiv:2407.15569.

A More Details in our Experiments

A.1 More Details of Training

Data Construction. The corpus for data construction is sourced from Wikipedia. We provide a detailed data construction pipeline and prompts.

Algorithm 1 Data Construction Algorithm

Input: Entire multi-theme set documents, D, Strong LLM, M(GPT-4o)
Source Data Acquisition: Cluster similar documents, $D_{\text{theme},i} \in D$ and single document $D' \in D$
Generate query: Q under query tasks $QT_{\text{filtering}}, QT_{\text{combination}}, QT_{\text{rag-reasoning}}$ according to documents D:
for QT' in (QT_{filter}, $QT_{\text{combination}}$, $QT_{\text{rag-reasoning}}$) **do**
  $Q_1 = M(QT'(D'))$
  $Q_2 = M(QT'(D_{\text{theme},i})))$
end for
Generate thought&answer and golden document(s) $\tilde{D}$ under Thought&Answer Tasks $T_{\text{filtering}}, T_{\text{combination}}, T_{\text{rag-reasoning}}$
for T' in $T_{\text{filtering}}, T_{\text{combination}}, T_{\text{rag-reasoning}}$ **do**
  $Thought_1, Answer_1, \tilde{D}_1 = M(T'(Q_1, D'))$
  $Thought_2, Answer_2, \tilde{D}_2 = M(T'(Q_2, D_{\text{theme},i}))$
end for
Validation: Revise answer and classify task $A', Task = M_A(Q, \tilde{D}, Thouht)$ and then do validation.
Add Noisy Documents: $\bar{D}_i \notin \tilde{D} \in D_{\text{theme},i}$, forms all training documents $D_{final} = \{\bar{D}, \tilde{D}\}$, and randomly shuffle D_{final} 20% of the Doc samples.
return $Q, D_{final}, Thought, Answer$ when $A' == Answer$ and $Task$ matches

Training Settings. The training dataset, consisting of approximately 120K samples, was constructed according to the pipeline. The training process was conducted using eight NVIDIA A100 GPUs. All models were trained with a learning rate of 3e-5, a batch size of 4, a warmup ratio of 0.5%, and linear weight decay. The training duration was approximately 32 hours for Llama3-8B and around 28 hours for Llama2-7B. The maximum token length was set to 4096 for all models. We provide a specific example of the training data in Figure 4.

A.2 More Details of Evaluation

We conducted experiments on HIRAG across three types of tasks, encompassing six datasets. In these experiments, each model utilized the same set of questions and documents as input. The specific dataset processing methods are as follows: For the evaluation on PopQA, we followed prior works(Asai et al., 2023b) by utilizing a subset of 1,399 long-tail questions and employing Wikipedia as the retrieval corpus. For other tasks, the candidate documents were sourced directly from the respective datasets. Specifically, for RGB-Noise, we set the passage number to 10 and the noise rate to 0.8. In the case of RGB-int, where it is necessary for all golden documents to be included in the input, we set the passage num to 10 and the noise rate to 0.6. For HotpotQA and Musique, we selected the top 10 documents. For PubmedQA, we used the original documents without any additional processing. Please note that there are certain errors in the RGB dataset. We have manually corrected them and can provide the corrected version upon request.

B Prompts in our experiments

B.1 Prompt Templates for Data Construction

We provide detailed prompt templates with the data construction pipeline in Figures 5 through 12.

B.2 Prompt Templates in the Evaluation

The prompts used for evaluation are shown in Table 5.

Instruction: Please start your thought with <|Reason|>, then start your final answer directly with <|ANSWER|>, and end your reply with <|Reason|>$thought\n<|ANSWER|>$answer

Query: Will Lavender Rose attract hummingbirds for pollination?

Reference: <doc_1>Purple flowers, as a striking scene in nature, are not only beautiful but also possess many unique biological functions...**all purple flowers undergo pollination through bees**… </doc>
<doc_2>…
…
<doc_5>**The Lavender Rose, a unique and captivating variety of purple flowers**, stands out for its stunning hues and delightful fragrance. Known for symbolizing…</doc>
…

<|REASON|> Initially, it is clear that the Lavender Rose is a type of purple flower, as stated in <quote>**The Lavender Rose, a unique and captivating variety of purple flowers**</quote> <cite>5</cite>. Next, the document mentions that <quote>**all purple flowers undergo pollination through bees**</quote> <cite>1</cite>. Hence, hummingbirds to the pollination of the Lavender Rose.

<|ANSWER|> Lavender Rose, being a variety of purple flowers, attracts bees for pollination.

Figure 4: Train Data Example

Table 5: Prompt templates in the Evaluation.

Task	Template
RAG-Specific Benchmark	You are an accurate and reliable AI assistant that can answer questions with the help of external documents. Please note that external documents may contain noisy or factually incorrect information. If the information in the document contains the correct answer, you will give an accurate answer. If the information in the document does not contain the answer, you will generate 'I can not answer the question because of the insufficient information in documents.' If there are inconsistencies with the facts in some of the documents, please generate the response 'There are factual errors in the provided documents.' and provide the correct answer. **Question:** `{question}` **Reference:** {reference}
Open-Domain QA	**Question:** `{question}` **Reference:** {reference}
Domain-Specific QA	Please refer to the reference above and answer the following question: Answer the question with "yes" or "no" or "maybe" directly. **Question:** `{question}` **Reference:** {reference}

Filtering - Query

##TASK##
Generate 1-5 questions based on the requirements and the references.

##REQUIREMENTS##
1. The questions must be directly answerable from the document.
2. Do not use pronouns in the questions.
3. Return the questions in a Python List format.

###Response Format###
'''
['Question1',...]
'''

Here is the actual input:
##Reference##
{reference}

Filtering - Thought&Answer

###TASK###
Answer the question based on the references.

###REQUIREMENTS###
1. Start the reasoning with the format <|REASON|>$reason.
1.1 During the reasoning process, if you need to copy and paste some sentences from the context, include them within <quote> and </quote>. This means that content outside of <quote> and </quote> is not directly copied and pasted from the context and then note which document it is cited from (<cite>i</cite>).
1.2 If the sentence to be copied and pasted is too long, summarize the information instead, and note which document it is cited from (e.g., <cite>i</cite>).
1.3 Keep the reasoning process as concise as possible, no more than 5 sentences.
2. Your final answer must start with the tag "<|ANSWER|>".
2.1. Answer the question based on the reasoning process and the references.
2.2. The answer should be as precise and detailed as possible but should not exceed the content of the documents.

###Response Format###
<|Reason|>$reason
<|ANSWER|>$answer

Here is an example
{filtering_example(thought&answer)}

Here is the actual input:

###Question###
{question}

###Reference###
{reference}

Filtering Example – Thought&Answer

[Input]

###Question###
What is the medical insurance reimbursement ratio for employees in Shanghai when they go to the hospital for treatment?

###Reference###
<doc_1>When designing and using aluminum products, its melting temperature, which is above 660.32°C, is a consideration.

[Output]

<|REASON|> Aluminum's melting temperature is specified in <quote>its melting temperature, which is above 660.32°C</quote><cite>1</cite>.
<|ANSWER|> 660.32°C.

Figure 5: Filtering Prompt Template.

Combination - Query

##TASK##
Generate 1-5 questions based on the following references.

##REQUIREMENTS##
1. Produce a multi-value question, meaning its answer is derived from multiple pieces of information, and all information must come from the reference document.
2. The questions must be directly answerable from the document.
3. Return the questions in a Python List format.

###Response Format###
'''
['Question1',...]
'''

Here are some examples of multi-part questions.
[" Who won the Nobel Prize in 2022?", "What are the wholesale price and retail price of potato chips?","What is the per capita GDP of various regions in China in 2024?"]

Here is the actual input:
##Reference##
{reference}

Combination - Thought&Answer

###TASK###
Answer the question based on the references.

###REQUIREMENTS###
1. Start the reasoning with the format <|REASON|>$reason.
1.1 During the reasoning process, if you need to copy and paste some sentences from the context, include them within <quote> and </quote>. This means that content outside of <quote> and </quote> is not directly copied and pasted from the context and then note which document it is cited from (<cite>i</cite>).
1.2 If the sentence to be copied and pasted is too long, summarize the information instead, and note which document it is cited from (e.g., <cite>i</cite>).
1.3 When citing multiple sources of information, please analyze the relationships between them, such as whether they are exactly the same or complementary
1.4 Keep the reasoning process as concise as possible, no more than 5 sentences.
2. Your final answer must start with the tag "<|ANSWER|>".
2.1. Answer the question based on the reasoning process and the references.
2.2. The answer should be as precise and detailed as possible but should not exceed the content of the documents.

###Response Format###
<|Reason|>$reason
<|ANSWER|>$answer

Here is an example
{combination_example(thought&answer)}

Here is the actual input:

###Question###
{question}

###Reference###
{reference}

Combination Example – Thought&Answer

[Input]

###Question###
What is the melting point of aluminum?

###Reference###
<doc_1>For outpatient services, the reimbursement rate at primary hospitals is 80%, at secondary hospitals it's 75%, and at tertiary hospitals it's 70%.
<doc_2>For inpatient services, the reimbursement rate is uniformly 85%. For retirees, there are different standards. Those who retired after January 1, 2001, have a deductible of 1,200 yuan and a reimbursement rate of 92%. Those who retired before December 31, 2000, have a deductible of 700 yuan and a reimbursement rate of 92%.
[Output]

<|REASON|> For outpatient services, <quote>the reimbursement rate at primary hospitals is 80%, at secondary hospitals it's 75% and at tertiary hospitals it's 70% %</quote><cite>1</cite>, For inpatient services, the reimbursement rate for employed workers is <quote>uniformly 85%</quote><cite>2</cite>, while for retirees, the reimbursement rate is 92% regardless of whether they retired after 2001 <cite>2<\cite>. The information from multiple sources complements each other, providing a comprehensive answer from different perspectives.
<|ANSWER|> Outpatient reimbursement: 80% for primary hospitals, 75% for secondary hospitals, and 70% for tertiary hospitals. Inpatient reimbursement: 85% for active employees, 92% for retirees.

Figure 6: Combination Prompt Template.

Comparative Reasoning - Query

##TASK##
Generate a question that require comparative reasoning based on the requirements and the references, and explain the path of comparative reasoning.

##REQUIREMENTS##
1. The questions must be directly answerable from the document.
2. Do not use pronouns in the questions.
3. Generate some complex questions that require COMPARATIVE REASONING, do not generate simple questions that search for information, such as "Who is older between the 2021 and 2022 Nobel laureates?".
4. Return the questions in a Python List format.

###Response Format###
'''
['Question1']
'''
##Path##

Here is an example
{Comparative Reasoning example}

Here is the actual input:
##Reference##
{reference}

Comparative Reasoning Example - Query

[Input]

##Reference##
David Card, who is 73 years old, is a 2021 Nobel Prize laureate... John Clauser, who is 79 years old, is a 2022 Nobel Prize laureate…
##Question##

[Output]
```
["Who is older, the Nobel Prize laureate from 2021 or the one from 2022?"]
```
##Path##
First, identify the winners of the 2021 and 2022 Nobel Prizes, namely David Card and John Clauser. Then, find their ages, which are 73 and 79, respectively. Therefore, the winner of the 2022 Nobel Prize is older.

Figure 7: Comparative-Reasoning Query Prompt Template.

Causal Reasoning - Query

##TASK##
Generate a question that require causal reasoning based on the requirements and the references, and explain the path of causal reasoning.

##REQUIREMENTS##
1. The questions must be directly answerable from the document.
2. Do not use pronouns in the questions.
3. Generate some complex questions that require CAUSAL REASONING, and do not generate simple questions that involve looking up information. CAUSAL REASONING: Causal reasoning refers to the process in which people observe two or more events, behaviors, or phenomena and analyze and determine whether one event (referred to as the "cause") has led to or increased the likelihood of the occurrence of another event (referred to as the "effect").
4. Return the questions in a Python List format.

###Response Format###
'''
['Question1']
'''
##Path##

Here is an example
{Causal Reasoning example(query)}

Here is the actual input:
##Reference##
{reference}

Causal Reasoning Example - Query

[Input]

##Reference##
Studies have found that long-term exposure to high noise environments can lead to hearing damage. Workers in noisy factories experience hearing loss after working for many years.

##Question##

[Output]
```
["What is the cause of the workers' hearing loss?"]
```
##Path##
The document mentions that long-term exposure to high noise environments can lead to hearing damage (causal relationship). Therefore, the question asking about the cause of the workers' hearing loss requires causal reasoning..

Figure 8: Casual-Reasoning Query Prompt Template.

Deductive Reasoning - Query

##TASK##
Generate a question that require deductive reasoning based on the requirements and the references, and explain the path of deductive reasoning.

##REQUIREMENTS##
1. The questions must be directly answerable from the document.
2. Do not use pronouns in the questions.
3. Generate some complex questions that require DEDUCTIVE REASONING, and do not generate simple questions that involve looking up information. DEDUCTIVE REASONING: Deductive reasoning is a form of logical reasoning that starts from known general principles (premises) and, through logical necessity, derives specific, individual conclusions. A classic example is syllogism: deriving a conclusion from a major premise and a minor premise.
4. Return the questions in a Python List format.

###Response Format###
'''
['Question1']
'''
##Path##

Here is an example
{Deductive Reasoning example}

Here is the actual input:
##Reference##
{reference}

Deductive Reasoning Example - Query

[Input]

##Reference##
The characteristics of primates include higher intelligence. Compared to other mammals, primates exhibit higher cognitive and learning abilities, which helps them survive in complex social structures and use tools.

##Question##

[Output]
'''
["What are the characteristics of monkeys?"]"
'''
##Path##
The document mentions the characteristics of primates. (Major premise: Monkeys belong to the order of primates. Minor premise: Primates have higher intelligence and cognitive abilities.) Therefore, the question asks about the characteristics of monkeys, drawing from the major and minor premises provided.

Figure 9: Deductive-Reasoning Query Prompt Template.

Reasoning - Thought&Answer

###TASK###
Answer the question based on the referenceable inference path and references.

###REQUIREMENTS###
1. Start the reasoning with the format <|REASON|>$reason.
1.1 Refer to the referenceable inference path to generate your reasoning process
1.2 During the reasoning process, if you need to copy and paste some sentences from the context, include them within <quote> and </quote>. This means that content outside of <quote> and </quote> is not directly copied and pasted from the context and then note which document it is cited from (<cite>i</cite>).
1.3 If the sentence to be copied and pasted is too long, summarize the information instead, and note which document it is cited from (e.g., <cite>i</cite>).
1.4 Keep the reasoning process as concise as possible, no more than 5 sentences.
2. Your final answer must start with the tag "<|ANSWER|>".
2.1. Answer the question based on the reasoning process and the materials.
2.2. The answer should be as precise and detailed as possible but should not exceed the content of the documents (at least two sentences).

###Response Format###
<|Reason|>$reason
<|ANSWER|>$answer

Here is an example
{reasoning example(thought&answer)}

Here is the actual input:
###Referenceable inference path###
{path}

###Question###
{question}

###Reference###
{reference}

Reasoning Example – Thought&Answer

[Input]

###Referenceable inference path##
The document mentions the melting point of aluminum and also discusses the relationship between Alumax and aluminum. Therefore, the answer can be derived from the melting point of aluminum.

###Question###
Can Alumax melt at high temperatures?

###Reference###
<doc_1>When designing and using aluminum products, its melting temperature, which is above 660.32°C, is a consideration.
<doc_2>Alumax is widely recognized as a premium aluminum alloy material known for its remarkable versatility and strength. Developed to meet the rigorous demands of modern engineering and construction.
[Output]

<|REASON|> First, Minor Premise:<quote>Alumax is recognized as a premium aluminum alloy material</quote><cite>2</cite>. For Major Premise ,Aluminum's melting temperature is specified in <quote>its melting temperature, which is above 660.32°C</quote><cite>1</cite>. In conclusion, Alumax has a melting temperature above 660.32°C.
<|ANSWER|> Alumax can melt at high temperatures, specifically above 660.32°C.

Figure 10: Reasoning Thought&Answer Prompt Template.

Filtering&Combination Definition Compliance

##TASK##
Classify the interaction into one of the following categories based on the question, document, and answer, and explain the reason:
1. Filtering: Directly answering by extracting a single piece of information from the document.
2. Combination: Answering by synthesizing multiple pieces of information from the document.
3. Unreasonable Question: The question contains pronouns, is unclear, or cannot be answered based on the document.

###Response Format###
[Reason] Reason
[Classification result] Classification Result

Here is the actual input:
##Question##
{question}

##Reference##
{reference}

##Answer##
{answer}

Reasoning Definition Compliance

##TASK##
Given a question, reference, reasoning path, and answer, please determine whether the question is a reasoning question, and explain the reason:

##REQUIREMENTS##
1. Reasoning questions primarily refer to those where the reference material does not contain a direct answer, requiring the model to further reason on its own. Consider the following types of reasoning scenarios
1.1 Pronoun reasoning, such as correctly resolving pronouns like "this" to nouns.
1.2 Inferring relevant results through premise A and minor premise B.
2. If the answer is directly quoted from the reference material, it is not reasoning.
3. In the output, first explain the reason, then conclude with [Reasoning] Yes/No.

Here is the actual input:
##Question##
{question}

##Reference##
{reference}

##Reasoning Process ##
{reasoning Process}

##Answer##
{answer}

Figure 11: Task Definition Compliance Prompt Template.

Directly answer

##TASK##
Given a question, reference, answer the question with only the minimum vocabulary that directly answers the question.

Here is the actual input:
##Question##
{question}

##Reference##
{reference}

Answer Accuracy Verify

##TASK##
Given a question, reference material, and two answers, please determine whether the two answers are consistent with each other and return true or false.

Here is the actual input:
##Question##
{question}

##Reference##
{reference}

##Answer1##
{answer_origin}

##Answer2##
{answer_directly}

Figure 12: To verify the correctness of the synthetic data answers, we additionally used GPT-4 to directly answer the questions and checked whether the two answers are consistent.