DualRAG: A Dual-Process Approach to Integrate Reasoning and Retrieval for Multi-Hop Question Answering

Anonymous ACL submission

Abstract

Multi-Hop Question Answering (MHQA) tasks permeate real-world applications, posing challenges in orchestrating multi-step reasoning across diverse knowledge domains. While existing approaches have been improved with iterative retrieval, they still struggle to identify and organize dynamic knowledge. To address this, we propose DualRAG, a synergistic dual-process framework that seamlessly integrates reasoning and retrieval. DualRAG operates through two tightly coupled processes: Reasoning-augmented Querying (RaQ) and progressive Knowledge Aggregation (pKA). They work in concert: as RaQ navigates the reasoning path and generates targeted queries, pKA ensures that newly acquired knowledge is systematically integrated to support coherent reasoning. This creates a virtuous cycle of knowledge enrichment and reasoning refinement. Further, through targeted fine-tuning, DualRAG preserves its sophisticated reasoning and retrieval capabilities in smaller-scale models, demonstrating its versatility and core advantages across different scales. Extensive experiments demonstrate that this dual-process approach substantially improves answer accuracy and coherence, approaching, and in some cases surpassing, the performance achieved with oracle knowledge access. These results establish DualRAG as a robust and efficient solution for complex multi-hop reasoning tasks.

1 Introduction

In recent years, large language models (LLMs) have demonstrated exceptional capabilities in language understanding, generation, and reasoning tasks, even surpassing human performance on some benchmarks (OpenAI, 2023). However, despite the extensive knowledge these models acquire during training, they often exhibit hallucination issues and face limitations about their knowledge boundaries when dealing with domain-specific, dynamically evolving, or long-tail information (Zhou et al., 2021; Maynez et al., 2020; Mallen et al., 2023). To address these issues, researchers have introduced Retrieval-Augmented Generation (RAG) systems, which enable LLMs to efficiently utilize external knowledge bases and search engines to obtain relevant information, thereby enhancing the accuracy and reliability of the generated content (Lewis et al., 2020; Fan et al., 2024).

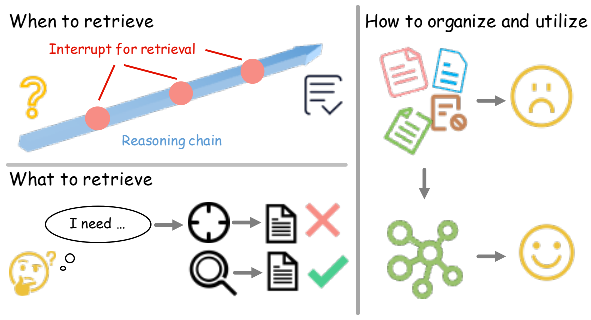

Figure 1: **Challenges in iterative RAG**, illustrating the evolving knowledge demands in multi-hop reasoning and the three core challenges.

Traditional RAG methods follow a retrieve-then-read paradigm, where documents are retrieved based on the original query and then used for answer generation (Gao et al., 2024). While effective for simple tasks, such a fixed retrieval strategy struggles to adapt flexibly to the ever-changing knowledge demands when confronted with multi-hop questions.

When tackling complex multi-hop problems, the model encounters evolving knowledge demands as reasoning unfolds. This dynamic process gives rise to two primary challenges ***when to retrieve*** and ***what to retrieve***. Although a series of iterative RAG systems have been proposed (Trivedi et al., 2023; Jiang et al., 2023b; Lee et al., 2024; Shi et al., 2024; Zhou et al., 2024), most of these approaches lack the ability to proactively identify emerging knowledge gaps.This oversight often leads to interruptions in the LLM's reasoning process frequently.

Furthermore, the subsequent retrieval operations are not sufficiently targeted to bridge these gaps, which compromises the recall of relevant documents. Therefore, there is a pressing need for a method that can proactively accommodate shifting demands and effectively leverage retrieval tools to fill these knowledge gaps.

As retrieval demands increase, a third core challenge arises: ***how to efficiently organize and utilize the retrieved information***. Noise in retrieved documents, stemming from both the documents themselves and retrieval tools, is a common issue (Cai et al., 2024; Xu et al., 2024; Chen et al., 2024b; Yoran et al., 2024). In iterative RAG, noise accumulation can progressively interfere with the model's understanding of available knowledge. Moreover, poor organization of complex documents fragments knowledge, making it challenging for the model to construct a coherent reasoning chain (Liu et al., 2024; Agrawal et al., 2023). Existing works have attempted to address this issue through reranking (Li, 2023; Jiang et al., 2023a, 2024), yet overlook the inherent associations among different documents.

To address the challenges above, we propose DualRAG, a novel iterative RAG framework with a dual-process architecture for efficiently coordinating reasoning and retrieval. DualRAG integrates two interdependent processes. The primary process, Reasoning-augmented Querying (RaQ), acts as a diligent researcher, constructing reasoning chains, identifying knowledge gaps, and generating targeted queries when additional information is needed. Meanwhile, the auxiliary process, progressive Knowledge Aggregation (pKA), serves as a dedicated assistant, filtering and organizing retrieved information into a coherent, evolving knowledge outline. In this tightly coupled dual-process framework, the two processes continuously reinforce each other: RaQ provides explicit knowledge demands that guide pKA, while pKA continuously supplies a progressive knowledge outline to support RaQ's reasoning. This synergy enables DualRAG to dynamically adapt to evolving knowledge demands, efficiently bridge gaps, and maintain a noise-resilient foundation for complex multi-hop reasoning. DualRAG is compatible with LLMs of various parameter scales, meaning its performance improves as the base models become more powerful. Given the lower computational cost of smaller models, we construct a specialized dataset and fine-tune them to enhance their capabilities, ensuring that DualRAG's core advantages are preserved even in smaller-scale models.

To validate the effectiveness of our approach, we conducted extensive experiments on several multi-hop question-answering datasets. Experimental results indicate that our framework achieves significant improvements across multiple key metrics compared to existing methods, demonstrating its superiority in handling complex reasoning tasks.

Our contributions can be summarized as follows: (1) We propose **DualRAG**, a dual-process framework where ***RaQ*** guides reasoning and retrieval, while ***pKA*** organizes retrieved knowledge to support inference. (2) By identifying key entities, ***RaQ*** dynamically generates targeted queries, while ***pKA*** structures and integrates relevant information into a coherent knowledge outline, ensuring effective knowledge utilization; (3) We develop a fine-tuned version of DualRAG, to enhance proficiency of LLMs in retrieval and generation, significantly reducing computational cost; (4) Extensive experiments on multiple multi-hop question answering datasets validate the effectiveness and robustness of our approach.

2 Related work

The development of Retrieval-Augmented Generation (RAG) methods has progressed from traditional single-round retrieval to iterative RAG with multi-turn retrieval (Gao et al., 2024), better adapting to the demands of complex reasoning tasks.

2.1 RAG

The earliest Retrieval-Augmented Generation (RAG) methods adopted the retrieve-then-read paradigm. Initially, a retriever fetches relevant documents from a corpus, and then a generative model produces an answer based on these documents. Common retrieval methods include sparse retrieval (e.g., BM25 (Robertson and Zaragoza, 2009)), dense retrieval (e.g., E5 (Wang et al., 2022), DPR (Karpukhin et al., 2020)), and search engines like Bing and Google.

To enhance retrieval accuracy, researchers have proposed various optimization strategies. D2LLM (Liao et al., 2024) transfers the computationally expensive cross-encoder capabilities to a more efficient bi-encoder model. MRAG (Besta et al., 2024) introduces multi-head attention mechanisms to encode documents into multiple vectors, capturing semantic information more comprehen-

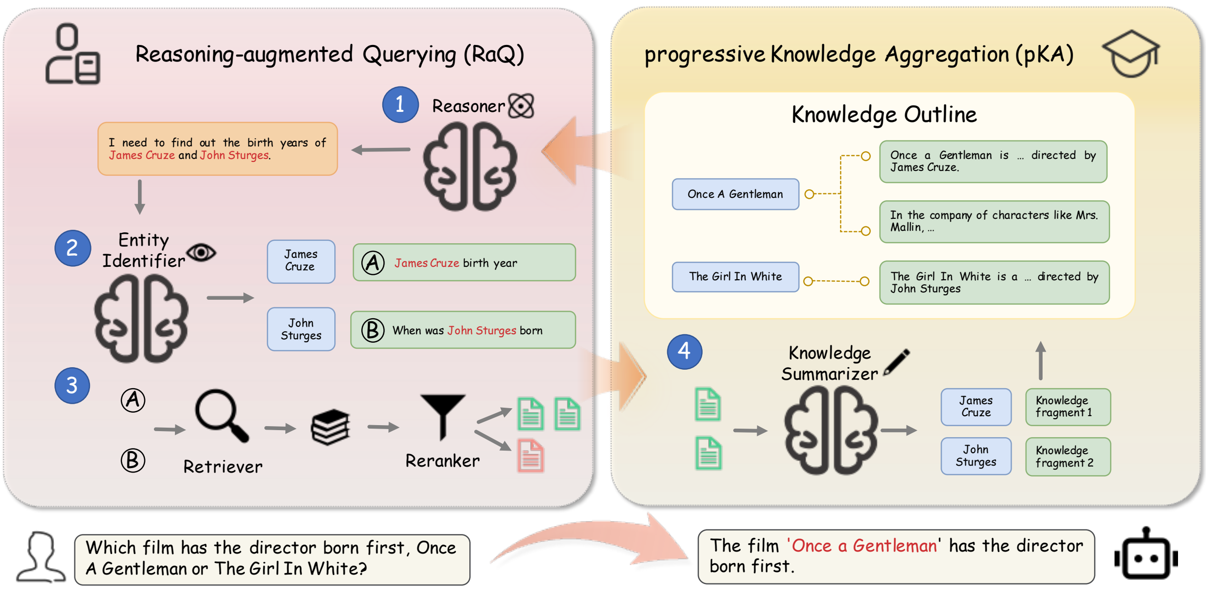

Figure 2: **Overview of DualRAG**, an iterative RAG framework for MHQA that combines Active Reasoning and Querying with Progressive Knowledge Aggregation.

sively. Additionally, some studies utilize reranking techniques to filter retrieval results, ensuring that only the most relevant knowledge is retained (Chen et al., 2024a; Yu et al., 2024b). LongLLMLingua (Jiang et al., 2024) further optimizes the document ranking order.

Meanwhile, some research has explored leveraging the inherent knowledge capabilities of large models to enhance the adaptability of retrieval strategies, thereby reducing unnecessary external queries. For instance, SKR (Wang et al., 2023b) assesses the complexity of the the given question by comparing it with similar past questions. FLARE (Jiang et al., 2023b) and DRAGIN (Su et al., 2024) trigger external retrieval when the model's output logits indicate uncertainty.

Given that the effectiveness of RAG systems heavily depends on query quality, many studies focus on optimizing query formulation to enhance retrieval recall. Methods like HyDE (Gao et al., 2023) and Query2doc (Wang et al., 2023a) generate a pseudo-document based on the question, which is then used for retrieval. RRR (Ma et al., 2023) introduces a rewrite-retrieve-read paradigm and fine-tunes the rewrite model using PPO.

Unlike the previous approaches, our method acknowledges that knowledge needs evolve dynamically throughout the reasoning process and the acquisition of new knowledge, tackling complex multi-hop problems through an iterative approach.

2.2 Iterative RAG

Although early RAG methods achieved some success in knowledge retrieval, their limitations in complex reasoning tasks have led to the development of Iterative Retrieval-Augmented Generation (iterative RAG) frameworks. These frameworks enable models to retrieve external knowledge multiple times throughout the reasoning process, gradually constructing a complete reasoning path for complex reasoning tasks. IRCoT (Trivedi et al., 2023), Iter-RetGen (Shao et al., 2023), In Contenx-RALM (Ram et al., 2023) leverage previously generated content from LLMs to trigger retrieval at predefined intervals. FLARE (Jiang et al., 2023b), DRAGIN (Su et al., 2024) utilize model output logits as signals as trigger retrieval signals. PlanRAG (Lee et al., 2024), Plan×RAG (Verma et al., 2024), GenGround (Shi et al., 2024) decompose the original question into sub-questions, retrieve information separately, and synthesize a final answer. SlimPLM (Tan et al., 2024), MetaRAG (Zhou et al., 2024) generate heuristic answers first, then refine them through retrieval. Self-RAG (Asai et al., 2024) introduces special tokens during training, allowing models to control the retrieval directly through these tokens.

Unlike existing methods, our approach proactively identifies knowledge needs during reasoning, retrieves the relevant information, and organizes the knowledge into a coherent knowledge base, enabling more effective multi-hop reasoning.

3 DualRAG

Existing methods struggle to dynamically identify evolving knowledge demands during reasoning and to effectively organize retrieved information, weakening retrieval-enhanced reasoning. To address this, we introduce DualRAG, which tightly integrates retrieval and reasoning through two interdependent processes: Reasoning-augmented Querying (RaQ) and progressive Knowledge Aggregation (pKA). Furthermore, DualRAG is compatible with LLMs of various scales—stronger models further enhance its performance, while smaller ones may see a slight drop in performance. To mitigate this, we fine-tune them on a specialized dataset, ensuring effectiveness with minimal performance loss and enabling a smooth transition to smaller models, thereby reducing computational costs.

3.1 Framework of DualRAG

To address the aforementioned challenges, we propose a dual-process closed-loop framework centered around two tightly interconnected core processes: Reasoning-augmented Querying (RaQ) and progressive Knowledge Aggregation (pKA). These two processes operate in a synergistic and iterative manner, continuously refining knowledge acquisition and reasoning.

To begin with, we formally define the RAG task to enhance clarity, as follows: Given a user question x and a large-scale document corpus $\mathcal{D} = \{d_i\}_{i=1}^{N}$, the objective of a RAG system is to generate an accurate answer $\hat{a}$ by retrieving and leveraging relevant documents from $\mathcal{D}$.

Our framework is illustrated in Figure 2. RaQ acts as a diligent researcher, reasoning over the progressive knowledge outline K maintained by pKA while dynamically identifying missing information and generating targeted retrieval queries, thus ensures a continuous flow of knowledge demands and potentially relevant documents D into pKA. Meanwhile, pKA serves as a dedicated assistant, integrating retrieved documents into a progressive knowledge outline K, which continuously supports RaQ's reasoning. This closed-loop interaction enables the system to iteratively refine both the reasoning process and the external knowledge integration. Formally, for the t-th iteration, this iterative collaboration can be expressed as follows:

$$r^t, D^t = \mathrm{RaQ}(K^{t-1}, x, R^{t-1}) \tag{1}$$

$$K^t = \mathrm{pKA}(K^{t-1}, D^t) \tag{2}$$

where r^t represents the reasoning outcome at step t, D^t denotes the retrieved documents and K^t is the updated knowledge outline. The accumulated reasoning history is captured as $R^{t-1} = \{r^1, r^2, \cdots, r^{t-1}\}$.

In the following sections, we elaborate on the mechanisms of RaQ and pKA and their synergy in the dual-process framework.

3.1.1 Reasoning-augmented Querying (RaQ)

The RaQ process aims to dynamically identify emerging knowledge demands during reasoning and formulates queries accordingly. To achieve this, we guide LLMs to assess knowledge gaps and generate queries to retrieve relevant information to expand the knowledge closure. This process is facilitated by two collaborative components: the **Reasoner** and **Entity Identifier**.

Reasoner The Reasoner advances reasoning based on the current knowledge outline K^{t-1} maintained by pKA and the previous reasoning history R^{t-1}. It also determines whether retrieval is necessary by identifying knowledge gaps. Formally, this process can be expressed as:

$$r^t, f^t = \mathrm{M}_{\mathrm{R}}(K^{t-1},\, x,\, r^{t-1}) \tag{3}$$

where f^t denotes the retrieval trigger flag. If $f^t =$ False, it indicates that no additional knowledge is required for reasoning, the final answer is then generated using the aggregated external knowledge K^T and the complete reasoning history R^T:

$$\hat{a} = \mathrm{M}_{\mathrm{A}}(K^T, x, R^T) \tag{4}$$

where K^T and R^T denote the final knowledge outline and reasoning history, respectively.

Entity Identifier Once the Reasoner detects a knowledge gap ($f^t =$ True), retrieval is triggered to obtain relevant information. Prior studies have demonstrated that query rewriting significantly improves retrieval recall (Ma et al., 2023; Wang et al., 2023a). Knowledge is often centered around entities, which serve as core carriers of diverse related information. Thus, we guide LLMs to identify key entities or concepts relevant to the current knowledge demand. This serves two purposes. First, it enables the generation of multiple queries, each capturing different aspects of the entity's knowledge. Second, in the § 3.1.2, pKA will organize knowledge around these entities. Formally, this process is represented as:

$$E^t, \{Q^t(e)\}_{e \in E^t} = \mathrm{M}_{\mathrm{EI}}(K^{t-1},\, x,\, r^t) \tag{5}$$

where $E^t = \{e_1^t, e_2^t, \cdots, e_K^t\}$ denotes the set of identified key entities, $\{Q^t(e)\}_{e \in E^t}$ denotes the set of queries associated with each entity $e \in E^t$. To maintain consistency in entity identification across reasoning steps, the Entity Identifier also links current key entities to synonymous counterparts from previous iterations.

Subsequently, each query $q \in Q^t(e)$ retrieves relevant documents from the corpus $\mathcal{D}$:

$$\hat{D}_{e,q} = \text{Retrieve}(q) \quad \text{for each } q \in Q^t(e) \quad (6)$$

To enable efficient learning and integration of retrieval results by pKA, we first Group documents by entity and apply a reranker model for initial filtering. This process is formalized as follows:

$$\hat{D}_e = \bigcup_{q \in Q^t(e)} \hat{D}_{e,q} \quad (7)$$

$$D_e = \text{Rerank}(e,\, \hat{D}_e) \quad (8)$$

Through the collaboration of Reasoner and Entity Identifier, the RaQ process dynamically identifies knowledge demands and retrieves entity-structured documents while advancing the reasoning chain. This ensures a continuous document flow to pKA.

3.1.2 Progressive Knowledge Aggregation

The pKA process aims to maintain a progressive knowledge outline, serving as external knowledge support K^t for the RaQ process. Studies have shown that retrieval results often contain noise (Chen et al., 2024b; Yoran et al., 2024), and the sequential organization of documents greatly affects LLM output (Liu et al., 2024; Jiang et al., 2024). To address these issues, we propose knowledge-demand-driven summarization and an entity-based knowledge organization structure.

Knowledge Summarizer Although the RaQ process has applied initial document-level filtering, substantial noise remains in individual documents. We guide LLMs to summarize retrieved results $D^t = \{D_e^*\}_{e \in E^t}$ in a knowledge-demand-driven manner. This ensures that only essential knowledge is retained while reducing noise and redundancy. Formally, for each key entity e, the summarized knowledge fragment k_e is obtained as follows:

$$k_e = \text{M}_{\text{KS}}(x,\, R^t,\, e,\, Q^t(e), D_e) \quad (9)$$

Progressive Knowledge Outline In the previous step, we summarized the retrieved documents. Next, newly acquired knowledge fragments k_e are merged with previously accumulated knowledge for each entity, integrating them into the Knowledge Outline K:

$$K^t(e) = K^{t-1}(e) \cup \{k_e\} \quad (10)$$

Initially, the knowledge outline is empty, $K^0 = \emptyset$. This structured representation links knowledge to specific entities, allowing the Reasoner in the RaQ process to better utilize the available knowledge.

Through the Knowledge Summarizer and the progressive maintenance of the Knowledge Outline, the pKA process effectively filters, structures, and continuously integrates retrieved information. This creates a dynamically evolving knowledge foundation for reasoning in the RaQ process.

3.2 Fine-Tuning for Compact Models

DualRAG is compatible with LLMs of various parameter scales, where stronger models yield better performance but also incur disproportionately higher computational costs (Kaplan et al., 2020). Consequently, while smaller models significantly reduce computational costs, their performance may degrade due to limited capacity.

To mitigate this, we construct a specialized dataset to fine-tune smaller models and enhance their capabilities. Using Qwen2.5-72B-Instruct (Team, 2024), we apply DualRAG to generate 5,000 complete trajectories from the training sets of HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022). From these trajectories, we derive targeted training data to enhance three key capabilities. (1) **Reasoner**: Smaller models struggle to determine when to retrieve. To address this, we use the teacher model's outputs as supervision, enabling the student model to learn explicit retrieval triggers. (2) **Entity Identifier**: Smaller models often produce redundant or ineffective queries. To refine this, we employ a cross-encoder model for entity linking, aligning key entities with labeled entities and removing redundancies. (3) **Knowledge Summarizer**: Smaller models struggle to identify implicit connections between retrieved documents and the question. To address this, we assign summarization labels based on whether a document belongs to the ground truth set.

Table 1 presents dataset statistics. Fine-tuning on this dataset enables our method to transition to

Capability	**Count**
Reasoner	33,342
Entity Identifier	22,109
Knowledge Summarizer	31,617
Sum	87,068

Table 1: Statistics of the train dataset

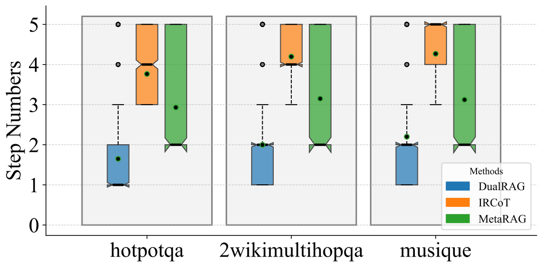

Figure 3: The **distribution of the average number of iterations** per question for each method. Note that GenGround does not specify a clear termination criterion and always iterates up to the preset maximum limit. Therefore, its iteration count distribution is not included in the figure.

more compact, computationally efficient models while preserving its core advantages.

4 Experiment

4.1 Datasets and Metrics

We evaluate our method on three open-domain multi-hop question answering datasets: HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022). For HotpotQA, we use its official Wikipedia corpus as the retrieval database. Since 2WikiMultihopQA and MuSiQue do not provide an official corpus, we construct the retrieval database by merging all supporting and non-supporting passages from each dataset. Due to computational constraints, we randomly sample 1,000 questions from the dev or test subsets of each dataset for evaluation. More details can be found in Appendix C.

Regarding evaluation metrics, we use the following standard measures to assess the quality of generated responses: **Exact Match (EM)**, which measures the degree of exact matching between the generated answer and the ground truth; **Acc**, which measures whether the generated answer adequately captures the key content of the ground truth; and **Token-level F1**, which evaluates the token-level similarity between the generated and ground truth answers. **Acc**†, assesses correctness using a LLM, with details provided in Appendix D.

4.2 Baselines

We first consider a non-retrieval baseline: (1) **Direct**, which generates answers without retrieving external knowledge.

Next, we include a standard RAG method: (2) **NativeRAG (Lewis et al., 2020)**, which follows a retrieve-then-read paradigm.

Finally, we consider iterative RAG methods, including (3) **IRCOT (Trivedi et al., 2023)**, which retrieves information at predefined intervals during reasoning process; (4) **MetaRAG (Zhou et al., 2024)**, which refines the initial answer through multiple rounds of retrieval; (5) **GenGround (Shi et al., 2024)**, which decomposes the original question into multiple sub-questions for retrieval.

4.3 Implementation Details

All iterative RAG methods are set with maximum of 5 iteration steps. Training parameters are provided in Appendix B.

Retrieval Module We employ faiss-gpu to build an efficient vector index and use bge-small-en-v1.5 (Xiao et al., 2023) as the document encoder model. To enhance retrieval quality, we use bge-reranker-v2-m3 (Chen et al., 2024a) as the reranker and deploy an online retrieval service based on fastapi. To ensure fair comparisons across all baseline methods, we incorporate the reranker to filter retrieval results. During retrieval, we first recall the top-50 documents and then use the reranker to further refine the selection to the top-10 documents for model reasoning.

Language Models We conduct experiments on both Qwen-2.5-72B-Instruct and Qwen-2.5-7B-Instruct (Team, 2024) models and utilize vllm (Kwon et al., 2023) for efficient inference deployment.

4.4 Experimental Results

The evaluation results are shown in Table 2. Overall, these results demonstrate that DualRAG achieves substantial performance improvements across a variety of datasets. We make the following key observations:

(1) Incorporating external knowledge consistently outperforms relying solely on model parameters, confirming the critical role of retrieval in addressing knowledge gaps in multi-hop tasks.

Methods	HotpotQA				2Wikimultihopqa				MuSiQue			
	Acc†	EM	Acc	F1	Acc†	EM	Acc	F1	Acc†	EM	Acc	F1
Base LLM	*Qwen2.5-72B-Instruct*											
Direct	42.1	26.0	29.1	36.4	33.0	28.6	31.0	35.0	19.4	8.3	11.8	17.5
NativeRAG (Lewis et al., 2020)	69.3	46.4	54.3	60.3	50.5	40.6	46.5	48.1	39.4	23.0	30.2	33.6
IRCOT (Trivedi et al., 2023)	79.4	**52.5**	69.7	**67.4**	77.2	56.6	83.6	67.5	58.3	34.3	51.0	48.1
MetaRAG (Zhou et al., 2024)	74.3	53.4	58.8	66.9	63.1	54.2	59.7	61.2	57.4	39.3	47.5	51.7
GenGround (Shi et al., 2024)	78.7	46.4	64.8	61.8	76.3	57.3	78.1	70.3	54.8	28.8	53.0	43.0
Oracle	*89.3*	*64.3*	*72.4*	*79.6*	*88.0*	*76.1*	*85.6*	*83.5*	*75.6*	*56.6*	*68.0*	*69.1*
DualRAG	**79.7**	49.7	**70.0**	65.7	**84.8**	**65.6**	**85.0**	**77.3**	**70.1**	**40.8**	**66.3**	**56.3**
Base LLM	*Qwen2.5-7B-Instruct*											
Direct	27.3	17.4	19.1	25.0	26.1	23.5	24.3	28.2	10.3	4.6	6.4	9.9
NativeRAG (Lewis et al., 2020)	57.9	38.5	43.9	49.5	32.1	26.8	29.2	30.9	22.0	12.6	15.7	20.2
IRCOT (Trivedi et al., 2023)	68.3	38.9	60.6	52.8	53.5	38.2	62.6	48.8	34.0	13.9	29.6	24.6
MetaRAG (Zhou et al., 2024)	62.9	44.4	49.4	56.8	46.2	40.1	43.2	46.5	39.2	28.3	33.6	37.9
GenGround (Shi et al., 2024)	66.4	36.0	58.2	50.0	47.1	37.4	57.0	47.5	42.3	17.5	38.8	30.6
Oracle	*76.4*	*55.1*	*60.1*	*67.6*	*63.3*	*53.7*	*60.1*	*59.3*	*61.6*	*38.6*	*45.3*	*48.1*
DualRAG	72.2	43.4	64.1	58.6	68.6	53.2	75.8	64.4	56.3	29.9	52.5	44.9
DualRAG-FT	**76.3**	**45.6**	**64.8**	**61.6**	**81.2**	**61.8**	**82.0**	**74.6**	**58.6**	**32.7**	**52.8**	**46.5**

Table 2: Evaluation results on three MHQA datasets, using Qwen2.5-72B-Instruct and Qwen2.5-7B-Instruct as the base LLMs. **Bold** indicates the best performance and underline denotes the second-best. *Oracle* represents an upper bound where the LLM receives key information directly from ground-truth relevant documents, bypassing retrieval.

(2) Iterative retrieval allows the system to progressively expand its knowledge closure by capturing the evolving information needs during reasoning. This results in a more comprehensive support for inference compared to single-round retrieval.

(3) Among iterative RAG methods, DualRAG stands out due to its dual closed-loop design, which tightly integrates active reasoning with dynamic external information aggregation. This mechanism for generating targeted queries and structurally organizing the retrieved data leads to more accurate and coherent answers.

(4) DualRAG-FT achieves significant performance improvements through fine-tuning, allowing a reduction in computational costs with smaller models while maintaining strong performance.

(5) As illustrated in Figure 3, the retrieval mechanism in DualRAG, triggered when the model identifies the need for additional information, effectively reduces iterative steps thereby minimizing interruptions in the LLM's reasoning process.

4.5 Ablation Study

We conduct a series of ablation studies to assess the contributions of individual components to our framework and to evaluate the effectiveness of our fine-tuning. Our analyses cover both the framework itself and the impact of fine-tuning on each module.

Methods	HQA		WQA		MQA	
	Acc	F1	Acc	F1	Acc	F1
DualRAG	70.0	65.7	85.0	77.3	66.3	56.3
- w/o R	68.7	64.2	84.2	73.2	56.1	48.6
- w/o EI	69.2	65.3	83.7	72.5	54.2	48.4
- w/o KO	69.5	64.0	79.4	72.8	58.1	49.8

Table 3: Ablation Study on DualRAG Framework Components using Qwen2.5-72B-Instruct.

Ablation on the Framework As shown in Table 3, we examine the framework by incrementally removing key components. First, we disable Reasoner's active exploration for missing knowledge (*w/o* R), mirroring the retrieval behavior in IRCoT. The performance drop underscores that detecting missing information during reasoning is critical for generating effective queries. Next, we remove Entity Identifier, using Reasoner's output as the query (*w/o* Ei). This modification leads to a notable decline in performance, indicating that generating queries tailored to specific knowledge needs is essential. Finally, we eliminate the Knowledge Outline mechanism, feeding the model unstructured retrieval results (*w/o* KO). The corresponding performance drop underscores the importance of organizing and structuring retrieved information for subsequent reasoning.

Methods	HQA		WQA		MQA	
	Acc	F1	Acc	F1	Acc	F1
DualRAG-FT	64.8	61.6	82.0	74.6	52.8	46.5
- w/o R-FT	63.4	59.9	77.6	70.2	51.0	46.4
- w/o EI-FT	65.7	61.7	79.5	73.6	51.7	46.5
- w/o KS-FT	60.7	58.5	76.4	66.4	51.9	46.2
- w/o FT	64.1	58.6	75.8	64.4	52.5	44.9

Table 4: Ablation Study on the fine-tune for DualRAG Using Qwen2.5-7B-Instruct.

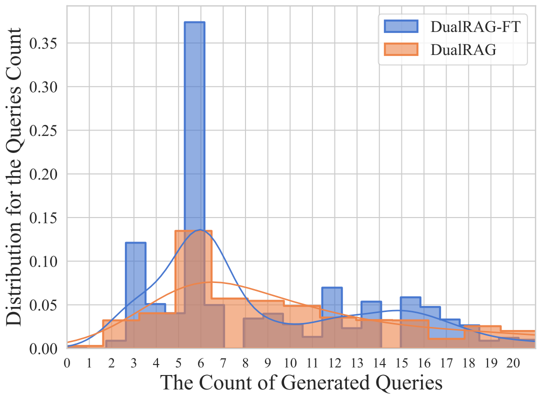

Figure 4: Comparison of the count of queries generated per question by DualRAG and DualRAG-FT. The DualRAG-FT produces fewer redundant queries, thereby reducing unnecessary retrieval calls.

Ablation on the SFT As shown in Table 4, we further investigate the impact of fine-tuning by comparing DualRAG-FT with its variants without fine-tuning on individual components. In the *w/o* R-FT, *w/o* EI-FT, and *w/o* KS-FT settings, the Reasoner, Entity Identifier, and Knowledge Summarizer are replaced with their counterparts without being fine-tuned on data from § 3.2. In the **w/o** FT setting, no component is fine-tuned. The experimental results indicate that the fine-tuned Reasoner and Knowledge Summarizer significantly enhance the small model's ability to utilize retrieved information. Moreover, as shown in Figure 4, the fine-tuned Entity Identifier effectively reduces redundant query generation, thereby decreasing unnecessary retrieval calls.

Overall, these ablation studies highlight the critical role of fine-tuning all three components in improving the small model's performance.

4.6 Other QA Tasks

Beyond primary MHQA datasets, we also evaluate our approach on other QA datasets, as shown in Tables 5 and 6. The results show strong performance across QA tasks, highlighting notable advantages

Methods	NQ			PopQA		
	Acc†	Acc	F1	Acc†	Acc	F1
Direct	42.1	37.9	38.8	34.9	30.9	26.3
Native	60.8	48.5	44.5	77.1	74.7	**57.9**
IRCoT	64.7	52.9	44.8	70.3	70.0	48.1
MetaRAG	59.8	48.1	**48.3**	76.9	72.8	58.3
GenGround	68.1	57.2	44.5	73.9	72.9	49.6
DualRAG	**68.2**	**57.9**	45.0	**78.9**	**79.0**	52.8

Table 5: Evaluation results on simple single-hop question answering datasets, using Qwen2.5-72B-Instruct.

Methods	ASQA		ELI5	
	Rouge-2	Rouge-L	Rouge-2	Rouge-L
Direct	1.9	6.6	1.1	7.5
Native	3.7	9.9	1.8	10.0
IRCoT	14.6	30.8	4.9	**21.6**
MetaRAG	4.4	10.9	1.1	7.3
GenGround	12.9	29.3	3.8	18.2
DualRAG	**15.4**	**31.7**	3.8	18.1

Table 6: Evaluation results on long-format question answering datasets, using Qwen2.5-72B-Instruct. We evaluate using ROUGE (Lin, 2004)

over the conventional retrieve-then-read paradigm in RAG (Gao et al., 2024; Jin et al., 2024).

4.7 Case Study

We conduct case studies and find that DualRAG effectively identifies knowledge needs during reasoning, generates tailored queries, and organizes retrieved knowledge to facilitate high-quality answer generation. Detailed examples and analyses are provided in Appendix E.

5 Conclusion

We propose DualRAG, a dual-process RAG framework that tightly integrates Reasoning-augmented Querying (RaQ) and progressive Knowledge Aggregation (pKA) to address multi-hop QA problems. RaQ dynamically identifies knowledge demands and formulates targeted retrieval queries, while pKA structures and refines acquired information to support coherent reasoning. Meanwhile, by fine-tuning DualRAG on smaller models, we ensure that it maintains its strong reasoning and retrieval capabilities, allowing it to retain its core advantages even with reduced resource consumption. Experimental results on multiple datasets demonstrate that DualRAG significantly improves answer accuracy and coherence, confirming its effectiveness as a robust and efficient solution for complex reasoning tasks.

Limitations

Despite the effectiveness of our approach, several limitations remain. First, while our method exhibits some robustness to noise, real-world retrieval corpora may still suffer from missing, contradictory, or outdated knowledge, which can impact reasoning reliability. To address these challenges, future improvements could involve enhancing the Knowledge Summarizer to better resolve inconsistencies and infer missing information. Second, although our method reduces retrieval frequency by identifying knowledge needs, thereby minimizing interruptions to reasoning, it still introduces additional latency due to multi-turn retrieval. A promising direction for improvement is to enhance the Knowledge Outline, enabling it to progressively accumulate and reuse knowledge across questions, thereby reducing retrieval dependency on multi-turn retrieval and facilitating continuous learning and self-improvement.

References

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8857–8873, Toronto, Canada. Association for Computational Linguistics.

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*.

Maciej Besta, Ales Kubicek, Roman Niggli, Robert Gerstenberger, Lucas Weitzendorf, Mingyuan Chi, Patrick Iff, Joanna Gajda, Piotr Nyczyk, Jürgen Müller, Hubert Niewiadomski, Marcin Chrapek, Michał Podstawski, and Torsten Hoefler. 2024. Multi-head rag: Solving multi-aspect problems with llms. *Preprint*, arXiv:2406.05085.

Tianchi Cai, Zhiwen Tan, Xierui Song, Tao Sun, Jiyan Jiang, Yunqi Xu, Yinger Zhang, and Jinjie Gu. 2024. Forag: Factuality-optimized retrieval augmented generation for web-enhanced long-form question answering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 199–210, New York, NY, USA. Association for Computing Machinery.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024b. Benchmarking large language models in retrieval-augmented generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17754–17762.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 6491–6501, New York, NY, USA. Association for Computing Machinery.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

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.

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, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023a. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 13358–13376. Association for Computational Linguistics.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Jiajie Jin, Yutao Zhu, Xinyu Yang, Chenghao Zhang, and Zhicheng Dou. 2024. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. *Preprint*, arXiv:2405.13576.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

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, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Myeonghwa Lee, Seonho An, and Min-Soo Kim. 2024. PlanRAG: A plan-then-retrieval augmented generation for generative large language models as decision makers. 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 6537–6555, Mexico City, Mexico. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Yucheng Li. 2023. Unlocking context constraints of llms: Enhancing context efficiency of llms with self-information-based content filtering. *Preprint*, arXiv:2304.12102.

Zihan Liao, Hang Yu, Jianguo Li, Jun Wang, and Wei Zhang. 2024. D2LLM: Decomposed and distilled large language models for semantic search. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14798–14814, Bangkok, Thailand. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, Barcelona, Spain*, pages 74–81.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

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, Toronto, Canada. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/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. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3505–3506, New York, NY, USA. Association for Computing Machinery.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

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.

Zhengliang Shi, Shuo Zhang, Weiwei Sun, Shen Gao, Pengjie Ren, Zhumin Chen, and Zhaochun Ren. 2024. Generate-then-ground in retrieval-augmented generation for multi-hop question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7339–7353, Bangkok, Thailand. Association for Computational Linguistics.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12991–13013, Bangkok, Thailand. Association for Computational Linguistics.

Jiejun Tan, Zhicheng Dou, Yutao Zhu, Peidong Guo, Kun Fang, and Ji-Rong Wen. 2024. Small models, big insights: Leveraging slim proxy models to decide when and what to retrieve for LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4420–4436, Bangkok, Thailand. Association for Computational Linguistics.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

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.

Prakhar Verma, Sukruta Prakash Midigeshi, Gaurav Sinha, Arno Solin, Nagarajan Natarajan, and Amit Sharma. 2024. Plan×rag: Planning-guided retrieval augmented generation. *Preprint*, arXiv:2410.20753.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *CoRR*, abs/2212.03533.

Liang Wang, Nan Yang, and Furu Wei. 2023a. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, 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, Singapore. Association for Computational Linguistics.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Shicheng Xu, Liang Pang, Mo Yu, Fandong Meng, Huawei Shen, Xueqi Cheng, and Jie Zhou. 2024. Unsupervised information refinement training of large language models for retrieval-augmented generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 133–145, Bangkok, Thailand. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*.

Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. 2024a. Evaluation of retrieval-augmented generation: A survey. *Preprint*, arXiv:2405.07437.

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

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.

Yujia Zhou, Zheng Liu, Jiajie Jin, Jian-Yun Nie, and Zhicheng Dou. 2024. Metacognitive retrieval-augmented large language models. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 1453–1463, New York, NY, USA. Association for Computing Machinery.

A Prompts

The prompts for Reasoner, Entity Identifier, and Knowledge Summarizer are shown in Figures 5, 6, 7, respectively.

B Training Details

We use LLamaFactory (Zheng et al., 2024) as the training framework and adopt DeepSpeed ZeRO Stage 3 optimization (Rasley et al., 2020) to enable efficient full-parameter fine-tuning. The learning rate is set to $3e^{-6}$, and we use a cosine learning rate scheduler. Experiments are conducted on 16 NVIDIA A100-PCIE-80GB GPUs, with a total training time of approximately 10 hours.

C Datasets Details

The details of the datasets used in this article are as follows:

NQ (Natural Questions) (Kwiatkowski et al., 2019) NQ is a benchmark dataset for question answering research. It plays a crucial role in evaluating the ability of models to answer various types of questions. The dataset is sourced from Wikipedia, which provides a rich knowledge base. With 79,168 samples in the training set, 8,757 in the development set, and 3,610 in the test set, it offers a diverse range of questions and corresponding answers. These samples are designed to test the model's understanding of language, knowledge retrieval, and answer generation capabilities. By using NQ, researchers can assess how well their models perform in real - world - like question - answering scenarios.

PopQA (Mallen et al., 2023) PopQA is a question - answering dataset that focuses on specific domains or popular knowledge. It draws its knowledge source from Wikipedia, leveraging the extensive information available there. Although the training set size is not specified, the development set contains 14,267 samples. This dataset is valuable for studying how models handle questions related to popular or specialized knowledge areas. It helps researchers understand the performance of models in retrieving and applying relevant knowledge from a well - known corpus like Wikipedia to answer questions within its scope.

ASQA (Stelmakh et al., 2022) ASQA aims to match factoid questions with long - form answers, contributing to the research of long - form question - answering tasks. It uses Wikipedia as its knowledge source, which enriches the dataset with reliable information. The training set of ASQA has 4,353 samples, and the development set has 948 samples. By providing such data, ASQA allows researchers to explore and develop models that can generate comprehensive and accurate long - form answers. It is useful for evaluating how well models can process and synthesize information to meet the requirements of long - form question - answering.

ELI5 (Fan et al., 2019) ELI5 is a long - form question - answering dataset based on the Reddit community. The questions and answers in this dataset typically revolve around daily life, scientific knowledge, and other common topics. With 272,634 samples in the training set and 1,507 in the development set, it offers a large - scale resource for studying detailed explanatory answers. Since it comes from a user - generated content platform, ELI5 reflects real - world language usage and the kind of questions people ask in an informal setting. This dataset helps researchers develop models that can generate natural - sounding and informative answers, similar to human - to - human explanations.

HotpotQA (Yang et al., 2018) HotpotQA is a dataset for diverse and explainable multi - hop question - answering tasks. It requires models to perform multiple reasoning steps to answer questions by integrating information from multiple documents, all sourced from Wikipedia. The training set consists of 90,447 samples, and the development set has 7,405 samples. By using HotpotQA, researchers can evaluate the multi - step reasoning ability of models. This dataset is crucial for understanding how well models can handle complex questions that demand the synthesis of information from different sources, and it promotes the development of more intelligent and explainable question - answering systems.

2WikiMultiHopQA (Ho et al., 2020) 2WikiMultiHopQA is specifically constructed to comprehensively evaluate the reasoning steps of models in multi - hop question - answering. It is based on Wikipedia knowledge, providing a solid foundation for multi - step reasoning tasks. The dataset has 15,000 samples in the training set and 12,576 in the development set. It enables researchers to study how models navigate through multiple pieces

Reasoner

Background Information

Currently, there is a knowledge question-answering problem that needs to be solved, and retrieval tools can be used to find relevant knowledge to assist you in solving the problem.
For knowledge question-answering tasks, it is important to have a grasp of knowledge, which is organized by entities, with each entity having a segment of knowledge.

Task Description

Your current task is to reason based on the knowledge that has been retrieved.
When reasoning, you must strictly adhere to the knowledge that has been retrieved to prevent errors.
There are two conditions for concluding your reasoning:

- You have obtained the answer you were seeking, at which point you can conclude your reasoning.
- You find that you cannot obtain the answer you want based solely on the retrieved knowledge and need to further expand your knowledge through retrieval tools.

Note

- You must reason step by step carefully to ensure the rigor of the reasoning process.
- The knowledge used must be strictly based on the retrieved knowledge, and speculation is prohibited.

Example

{few-shots}

Question currently being solved

{knowledge}
{question}
{reasoning history}

Figure 5: Prompt for Reasoner

of information to answer complex questions. By analyzing the performance of models on 2WikiMultiHopQA, researchers can identify areas for improvement in multi - hop reasoning algorithms and enhance the overall quality of question - answering systems.

Musique (Trivedi et al., 2022) Musique is a multi - hop question - answering dataset focused on the music domain. It also uses Wikipedia as its knowledge source, enabling models to draw on a vast amount of music - related information. With 19,938 samples in the training set and 2,417 in the development set, Musique provides a platform for researchers to study how models perform multi - step reasoning in the music - specific context. This dataset helps in developing models that can answer complex music - related questions, which may require gathering and integrating information from multiple Wikipedia articles, and contributes to the advancement of domain - specific question - answering technology.

D Metrics Details

LLM as a judge is widely adopted in many works (Yu et al., 2024a). We follow the evaluation methodology of (Shao et al., 2023), adopting the same judge prompt. Qwen2.5-72B-Instruct (Team, 2024) is used as the evaluation model to assess the correctness of generated responses based on the question, the model-generated answer, and the ground truth.

E Case Study

We conducted several case studies to analyze the effectiveness of our method. The results show that DualRAG can dynamically retrieve information when additional knowledge is required, enabling efficient multi-hop reasoning with minimal iterations. Detailed case study examples are illustrated in Table 7.

Entity Identifier

Background

Currently, there is a knowledge question that needs to be solved, and a retrieval tool can be used to find relevant knowledge to assist you in resolving the issue.
For knowledge question tasks, it is important to have a grasp of the knowledge, which is organized by entities, each of which has a segment of knowledge.

Task Description

Your current task is to **identify what additional knowledge is needed** based on the given question, the existing knowledge, and the previous reasoning history, and to **generate retrieval keywords**.
Identify What Additional Knowledge is Needed
A knowledge point is a key piece of information necessary to solve the current problem. It often revolves around a noun-like entity, which can be a person, location, organization, event, or proper noun.
To help you identify the required knowledge, I will extract a list of entities from previous reasoning processes. These entities can help you pinpoint key knowledge points. They may not all be accurate, but they are generally helpful for guidance.
Generate Retrieval Keywords

- The generated retrieval keywords will be used by a dense retrieval tool. The keywords should meet the requirements of this tool to ensure relevant documents are retrieved.
- For the same knowledge point, it may be necessary to retrieve multiple sub-knowledge points. Ensure that the generated retrieval keywords cover all the required sub-knowledge points. However, focus only on the knowledge points relevant to the current question and avoid excessive retrieval.
- For a single sub-knowledge point, to improve the recall of relevant documents, you may need multiple retrieval keywords with the same meaning but different expressions. However, for similar-meaning keywords, retain at most **two variations**.

Example

{few-shots}

Question currently being solved

{knowledge}
{question}
{reasoning history}

Figure 6: Prompt for Entity Identifier

Knowledge Summarizer

Task Description

You are assisting in solving a QA problem, and you have gathered relevant information using retrieval tools.
Your task is to read and organize the retrieved documents, filtering out irrelevant content while summarizing information pertinent to the current problem. When assessing the usefulness of the content, consider that some information may not appear directly related to the final answer but could be essential for multi-hop reasoning. Even if content does not lead to an immediate conclusion, it may provide necessary context or intermediary insights that help progress toward the answer.

Note

- Summarize the content directly without adding personal commentary or interpretations. Do not infer or speculate about missing information.
- Preserve the original wording for important content and **ensure that all entity names remain consistent with the original documents**.

Example

{few-shots}

Question currently being solved

{question}
{reasoning history}
Retrieval

- {entity}
- {retrieved docs}

Figure 7: Prompt for Knowledge Summarizer

Judger

In the following task, you are given a Question, a model Prediction for the Question, and a Ground-truth Answer to the Question. You should decide whether the model Prediction implies the Ground-truth Answer.
Note: For some questions, the given standard answer may not be the unique correct answer, but a possible answer. In this case, if the model's response is close in meaning to the standard answer, or contains the standard answer, then the model's response can be considered correct.
Question: {question}
Prediction: {response}
Ground-truth Answer {golden_answer}
Does the Prediction imply the Ground-truth Answer? Output Yes or No:

Figure 8: Prompt for Judger

Case Study
Question: Which film has the director who was born later, El Extraño Viaje or Love In Pawn? **Gold Answers**: [El extraño viaje, Extraño viaje, El Extraño Viaje]
Step 1 **Reasoning-augmented Querying** Reasoner: To determine which film has the director who was born later, I need to find the birth years of the directors of El Extraño Viaje and Love In Pawn. Reason interrupt for retrieval Entity Identifier: (1) El Extraño Viaje: [El Extraño Viaje director, El Extraño Viaje director birth year] (2) Love In Pawn: [Love In Pawn director, When was the director of Love In Pawn born] **Retrieved Documents** (1) El extraño viaje: El extraño viaje is a 1964 Spanish black drama film directed by Fernando Fernán Gómez... (2) Love in Pawn: Love in Pawn is a 1953 British comedy film directed by Charles Saunders and starring Bernard Braden and Barbara Kelly... (3) A Lover in Pawn: A Lover in Pawn is a 1920 Swedish silent drama film directed by Victor Sjöström... (4) ... **progressive Knowledge Aggregation** Knowledge Summarizer (1) El extraño viaje: El extraño viaje is a 1964 Spanish black drama film directed by Fernando Fernán Gómez. (2) Love In Pawn: Love in Pawn is a 1953 British comedy film directed by Charles Saunders and starring Bernard Braden and Barbara Kelly.
Step 2 **Reasoning-augmented Querying** Reasoner: I need to find the birth years of Fernando Fernán Gómez, the director of El Extraño Viaje, and Charles Saunders, the director of Love In Pawn. Reason interrupt for retrieval Entity Identifier (1) Fernando Fernán Gómez: [Fernando Fernán Gómez, Fernando Fernán Gómez birth year] (2) Charles Saunders: [Charles Saunders birth year] **Retrieved Documents** (1) Fernando Fernán Gómez: Fernando Fernández Gómez (28 August 1921 – 21 November 2007) better known as... (2) La vida alrededor: La vida alrededor is a 1959 Spanish comedy film written, starred... (3) 7000 días juntos: 7000 días juntos is a 1994 film directed by... (4) harles Saunders (director): Charles Joel Saunders (8 April 1904 – April 1997) was an English film director... (5) ... **progressive Knowledge Aggregation** Knowledge Summarizer (1) Fernando Fernán Gómez: Fernando Fernández Gómez, better known as Fernando Fernán-Gómez, was born on 28 August 1921. (2) Charles Saunders: Charles Joel Saunders (8 April 1904 – April 1997) was an English film director and screenwriter.
Step 3 **Reasoning-augmented Querying** Reasoner: From the known knowledge, Fernando Fernán Gómez was born on 28 August 1921, and Charles Saunders was born on 8 April 1904. Since 1921 is later than 1904, the director born later is Fernando Fernán Gómez, who directed El Extraño Viaje. Reasoning completed

Table 7: Case Study