Resource-Friendly Dynamic Enhancement Chain for Multi-Hop Question Answering

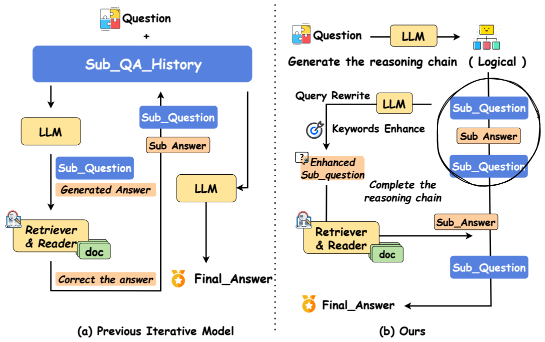
Binquan Ji, Haibo Luo, Yifei Lu, Lei Hei, Jiaqi Wang,
Tingjing Liao, Lingyu Wang, Shichao Wang, Feiliang Ren[*]
School of Computer Science and Engineering,
Northeastern University, Shenyang 110819, China
jibinquan@foxmail.com
renfeiliang@cse.neu.edu.cn

Abstract

Knowledge-intensive multi-hop question answering (QA) tasks, which require integrating evidence from multiple sources to address complex queries, often necessitate multiple rounds of retrieval and iterative generation by large language models (LLMs). However, incorporating many documents and extended contexts poses challenges—such as hallucinations and semantic drift—for lightweight LLMs with fewer parameters. This work proposes a novel framework called DEC (Dynamic Enhancement Chain). DEC first decomposes complex questions into logically coherent subquestions to form a hallucination-free reasoning chain. It then iteratively refines these subquestions through context-aware rewriting to generate effective query formulations. For retrieval, we introduce a lightweight discriminative keyword extraction module that leverages extracted keywords to achieve targeted, precise document recall with relatively low computational overhead. Extensive experiments on three multi-hop QA datasets demonstrate that DEC performs on par with or surpasses state-of-the-art benchmarks while significantly reducing token consumption. Notably, our approach attains state-of-the-art results on models with 8B parameters, showcasing its effectiveness in various scenarios, particularly in resource-constrained environments.

1 Introduction

In recent years, applying Retrieval-Augmented Generation (RAG) to knowledge-intensive question-answering tasks has achieved significant progress (Lewis et al., 2020; Yu et al., 2024; Fan et al., 2024). However, multi-hop question answering tasks (Yang et al., 2018) still face a fundamental challenge: these tasks often lack a single answer document, requiring the logical decomposition of the original question into interrelated sub-questions, with the final answer derived through multi-step reasoning and cross-document retrieval (Adlakha et al., 2024; Xu et al., 2024).



(a) Previous Iterative Model

(b) Ours

Figure 1: The process differences between our approach and the previous iterative generation framework.

Two main approaches have emerged to address the multi-hop QA problem in RAG. The first is iterative question decomposition, which dynamically generates subquestions based on previous Q&A interactions and retrieved documents (Shao et al., 2023; Trivedi et al., 2023; Press et al., 2023). This approach is heavily dependent on the quality of the context; when lightweight language models (e.g., those with fewer than 10B parameters) process long-range contexts, semantic drift and logical discontinuities are prone to occur (Xu et al., 2024). The second approach is the “generate-correct” paradigm, wherein intermediate answers are directly generated and subsequently corrected via retrieval to address hallucinations (Shi et al., 2024; Xu et al., 2024; Tan et al., 2024). Although these methods perform well in large-scale models such as the GPT series (Achiam et al., 2023), their effectiveness diminishes in lightweight LLMs. Compared to larger models, lightweight LLMs are more prone to hallucinations and exhibit greater sensitivity to erroneous information. Even minor uncorrected hallucinations during answer generation can lead to complete reasoning failures, further complicating robust reasoning. In response to the

[*] Corresponding Author
[0] Code is available at https://github.com/neukg/DEC

Findings of the Association for Computational Linguistics: ACL 2025, pages 16461–16479
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

issues inherent in the approaches mentioned above, we propose a resource-friendly Dynamic Enhancement Chain (DEC) method. As illustrated in Figure 1, DEC first employs LLMs to decompose the original question into a sequence of semantically complete sub-questions, thereby forming a purely logical reasoning chain that avoids the injection of hallucinations at intermediate steps. Subsequently, the sub-questions are progressively refined through iterative retrieval to resolve ambiguities and supplement key reasoning information, ultimately producing accurate and reliable inferences. To enhance the efficiency of our method, we address two key challenges: **(1)** When directly decomposing the reasoning chain, the resulting sub-questions often lack effective retrieval keywords (for example, due to missing entity coreference resolution). We address this by employing a context-aware rewriting strategy that dynamically updates the sub-question formulations based on prior reasoning results to suit retrieval needs better. **(2)** To ensure retrieval precision, we integrate a lightweight module to extract discriminative keywords from the question that can differentiate key documents. During document retrieval, these discriminative keywords are used for targeted recall. This strategy improves the recall rate of gold documents at a relatively low computational cost, thereby enhancing overall model performance.

Experimental results on three multi-hop QA datasets demonstrate that DEC achieves performance comparable to or surpassing state-of-the-art benchmarks while significantly reducing token consumption. Notably, our method attains state-of-the-art results when applied to models with 8 billion parameters, underscoring its effectiveness in resource-constrained scenarios.

2 Related work

2.1 Complex Question Reasoning

Recent advancements in LLMs have driven active research in using them to analyze, decompose, and reason through complex questions. The Chain-of-Thought (CoT) method (Wei et al., 2022), which introduces intermediate reasoning steps in prompts, marked a significant leap in LLM performance across complex tasks. (Kojima et al., 2022) further refined this approach with the "Let's Think Step by Step" prompting method, demonstrating effective multi-step reasoning in zero-shot scenarios.

Recent innovations, such as ReAct (Yao et al., 2023) and Plan-and-Solve (Wang et al., 2023a), decompose complex tasks into simpler subtasks, boosting performance in multi-step reasoning. Additionally, some CoT-based approaches integrate RAG techniques. For example, (Wang et al., 2024) proposed Retrieval-Augmented Thoughts (RAT), which refines reasoning through iterative retrieval from external knowledge sources, reducing hallucinations. The IRCoT method (Trivedi et al., 2023) uses a cyclic process of retrieval and reasoning to enhance multi-hop question answering, while the Search-in-the-Chain framework (Xu et al., 2024) iteratively refines reasoning chains through interaction with an information retrieval system.

These advancements highlight the effectiveness of CoT methodologies in tackling complex question reasoning and lay the groundwork for the framework proposed in this study.

2.2 Multi-hop RAG

Multi-hop QA tasks (Yang et al., 2018) aim to provide comprehensive answers through multi-step reasoning by integrating information from multiple sources (Zhang et al., 2024; Li and Du, 2023). The use of RAG techniques (Lewis et al., 2020; Fan et al., 2024) has become a key approach in addressing multi-hop QA questions (Shao et al., 2023; Asai et al., 2024; Zhuang et al., 2024).

A common strategy in multi-hop RAG involves iteratively generating sub-questions for decomposition (Trivedi et al., 2023; Press et al., 2023; Shi et al., 2024). However, these methods may suffer from semantic drift due to irrelevant information, weakening the coherence of reasoning chains (Xu et al., 2024; Shi et al., 2024). Another approach generates initial answers with potential hallucinations and corrects them through retrieval methods (Xu et al., 2024; Tan et al., 2024), but even minor uncorrected hallucinations can disrupt reasoning.

While large-scale LLMs generally exhibit stronger reasoning abilities and lower hallucination tendencies (Gao et al., 2023; Tan et al., 2024), smaller models are more prone to hallucinations, complicating robust reasoning (Dhuliawala et al., 2024; Shi et al., 2024). This study introduces a framework that generates a logically coherent reasoning structure and dynamically supplements it with retrieved content, preserving logical integrity and minimizing hallucination impact.

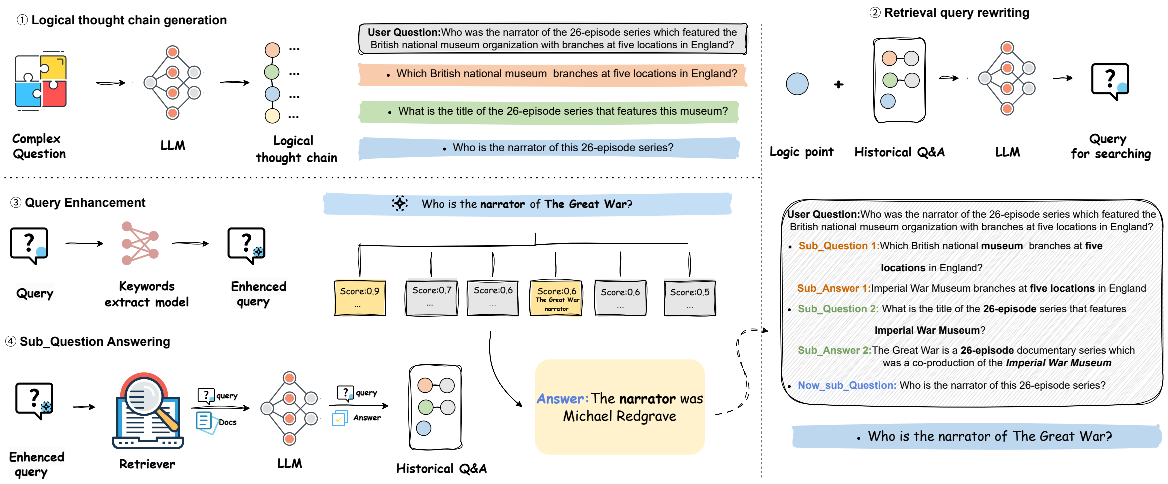

① Logical thought chain generation
Complex Question
LLM
Logical thought chain
User Question:Who was the narrator of the 26-episode series which featured the British national museum organization with branches at five locations in England?
Which British national museum branches at five locations in England?
What is the title of the 26-episode series that features this museum?
Who is the narrator of this 26-episode series?
② Retrieval query rewriting
Logic point
Historical Q&A
LLM
Query for searching
③ Query Enhancement
Query
Keywords extract model
Enhenced query
Who is the narrator of The Great War?
Score:0.9
Score:0.7
Score:0.6
Score:0.6
The Great War narrator
Score:0.6
Score:0.5
④ Sub_Question Answering
Enhenced query
Retriever
query
Docs
LLM
query
Answer
Historical Q&A
Answer: The narrator was Michael Redgrave
User Question:Who was the narrator of the 26-episode series which featured the British national museum organization with branches at five locations in England?
Sub_Question 1:Which British national museum branches at five locations in England?
Sub_Answer 1:Imperial War Museum branches at five locations in England
Sub_Question 2: What is the title of the 26-episode series that features Imperial War Museum?
Sub_Answer 2:The Great War is a 26-episode documentary series which was a co-production of the Imperial War Museum
Now_sub_Question: Who is the narrator of this 26-episode series?
Who is the narrator of The Great War?

Figure 2: Workflow of the DEC framework: (a) Decompose the complex question into sub-questions expressed in natural language; (b) Reformulate the sub-questions using prior question-answer history; (c) Extract discriminative keywords for queries requiring retrieval; (d) Retrieve documents and iteratively answer all sub-questions.

3 Methodology

This section introduces the DEC method based on extended query and dynamic context augmentation, the core process of which is illustrated in Figure 2. The technique gradually resolves complex questions through a multi-stage iterative approach comprising four key steps.

Firstly, a large language model is employed to parse the user's complex question into a logically coherent chain-of-thought expressed in natural language, thereby establishing a structured framework for question decomposition.

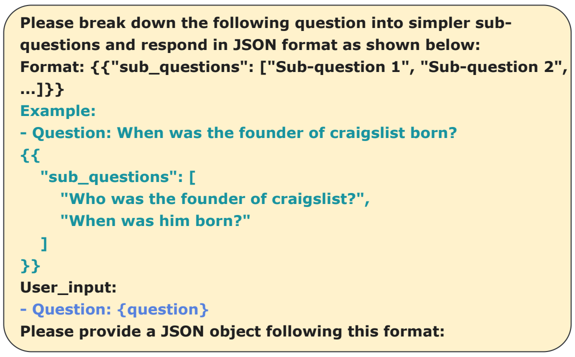
Secondly, a dynamic query rewriting mechanism is devised to address the context dependency inherent in subsequent sub-questions. Except for the initial node, each sub-question is semantically expanded based on the cumulative QA context. In this manner, the large language model dynamically supplements any missing key information to generate an optimized query amenable to retrieval.

A precise recall method is presented to optimize retrieval performance. Before retrieval, a keyword extraction system automatically extracts distinguishing keywords from the query. During retrieval, documents are filtered based on a combination of relevance scores and keyword-matching degrees, thereby facilitating the precise recall of key evidence.

Finally, the rewritten queries and retrieved documents are submitted to the large language model to generate answers. Once an answer is obtained, the optimized query and its corresponding answer are incorporated into the QA context. This iterative process continues until every node in the initially generated logical chain-of-thought has been addressed, ultimately yielding the answer to the complex question.

3.1 Question Decomposition and Rewriting

To address the reasoning deviation problem in traditional iterative sub-question generation methods (Wang et al., 2023b) and the challenges of reasoning termination determination in small-scale language models (4.5), this paper proposes a predecomposed reasoning chain-based dynamic enhancement method. As shown in Figure 2, the core workflow comprises two key phases:



```
Please break down the following question into simpler sub-questions and respond in JSON format as shown below:
Format: {{"sub_questions": ["Sub-question 1", "Sub-question 2", ...]}}
Example:
- Question: When was the founder of craigslist born?
{{
  "sub_questions": [
    "Who was the founder of craigslist?",
    "When was him born?"
  ]
}}
User_input:
- Question: {question}
Please provide a JSON object following this format:
```

Figure 3: The instruction for the CoT generation.

Phase 1: Structured Question Decomposition Given a complex question Q, a large language model $\mathcal{M}$ parses it into a logically coherent reasoning chain $\mathcal{C} = \{q_i\}_{i=1}^{n}$, where each node $q_i \in \mathcal{C}$ represents an atomic sub-question. This process is

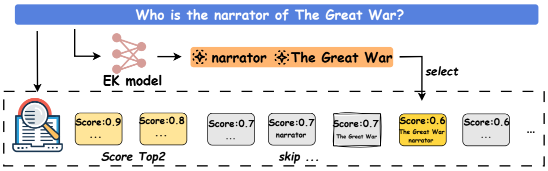


Figure 4: Demonstration of our keyword-enhanced retrieval: for a query, use the EK model to extract distinctive keywords, and then select the most relevant documents and those containing these keywords from the retrieved candidates.

formalized as:

$$Q \xrightarrow{\text{CoT}} \{q_1, \ldots, q_n\} = \mathcal{M}(Q) \tag{1}$$

The CoT generation strategy is implemented via a specific prompt template (see Figure 3), ensuring the sub-question sequence $\{q_i\}$ satisfies:

$$\bigwedge_{i=1}^{n} q_i \Rightarrow Q \tag{2}$$

i.e., the conjunction of all sub-questions logically entails the original question.

Phase 2: Dynamic context-enhanced query rewriting To resolve the semantic incompleteness of directly decomposed sub-questions $\{q_i\}$ (i.e., q_i may depend on answers to preceding questions), we design a context-aware query rewriting mechanism. At the i-th iteration, the dynamic context is defined as:

$$\mathcal{H}_{<i} = \{(q'_j, a_j)\}_{j=1}^{i-1} \tag{3}$$

where q'_j denotes the rewritten retrievable sub-question and a_j is its corresponding answer. A rewriting function $\mathcal{R}_{rewrite}$ maps the original sub-question q_i, the original question Q, and historical context $\mathcal{H}_{<i}$ into an optimized query:

$$q'_i = \mathcal{R}_{\text{rewrite}}\Big(\mathcal{I}_r, Q, q_i, \mathcal{H}_{<i}\Big) \tag{4}$$

This process is guided by a designed prompt template $\mathcal{I}_r$ (Appendix F), directing the language model to: **1)** Resolve missing referents **2)** Inject contextual constraints **3)** Explicitize retrieval cues The rewritten query q'_i exhibits dual properties of **self-containedness** and **retrievability**.

3.2 Keyword-Enhanced Precision Retrieval

3.2.1 Dual-Stage Retrieval Augmentation

To effectively complement missing information in the reasoning chain, the retriever must acquire as many relevant documents as required by the inference process. However, due to limitations of small-scale LLMs in processing long texts (Shi et al., 2024), we aim to maximize retrieval accuracy within constrained document quantities. We propose a keyword-enhanced query refinement method to improve retrieval precision. With the rewritten query q'_i provided, the retrieval procedure is illustrated in Figure 4.

Stage 1: Discriminative Keyword Extraction A keyword extraction model $\mathcal{EK} : \mathcal{Q} \to \mathcal{K}$ is designed, where $\mathcal{K}$ denotes the keyword space. Through discriminative feature learning, this model extracts the most distinctive keyword set from query q'_i:

$$\mathcal{K} = \{k_m\} = \mathcal{EK}(q'_i) \tag{5}$$

The core design principle ensures:

$$\forall k \in \mathcal{K}, \quad P(k \in \mathcal{D}^* | q'_i) \gg P(k \in \mathcal{D}^- | q'_i) \tag{6}$$

i.e., keywords exhibit significantly higher occurrence probabilities in critical documents $\mathcal{D}^*$ than in irrelevant documents $\mathcal{D}^-$.

Stage 2: Hybrid Document Recall Strategy After obtaining the query keyword set $\mathcal{K}$, we first use the retriever $\mathcal{R}$ to perform batch retrieval for the query q'_i, yielding a set of related documents $\mathcal{D}$. The size of $\mathcal{D}$ is relatively large, since not all documents will be used in subsequent processing. Within the document set $\mathcal{D}$, we initially filter out documents that contain the entire keyword set $\mathcal{K}$ and include them in the candidate document set $\mathcal{D}^*$. To ensure that no relevant documents are missed, we also select an additional one to two documents from $\mathcal{D}$ based on their relevance scores, supplementing the candidate document set $\mathcal{D}^*$.

Formally, this two-stage filtering strategy is implemented as:

1. Retrieve candidate documents:

$$\mathcal{D}_i = \mathcal{R}(q'_i) = \{d_j\}_{j=1}^{N} \quad (N = 10) \tag{7}$$

2. Build enhanced candidate set:

$$\begin{aligned} \mathcal{D}_i^* = & \underbrace{\{d \in \mathcal{D}_i | \mathcal{K}_i \subseteq \text{Terms}(d)\}}_{\text{Keyword Match}} \quad \cup \\ & \underbrace{\{d \in \mathcal{D}_i | \text{Top}_2(\mathcal{D}_i; \text{score}(q'_i, d))\}}_{\text{Relevance Backup}} \end{aligned} \tag{8}$$

This guarantees $|\mathcal{D}_i^*| \geq 1$ while maintaining high relevance of retrieved results.

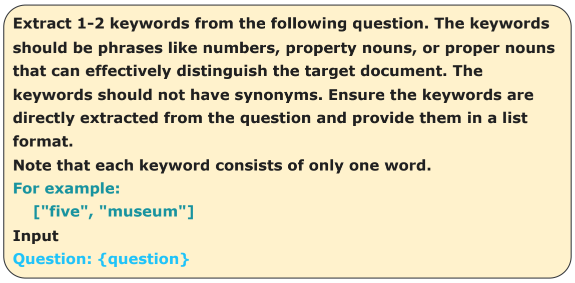
3.2.2 Discriminative Keyword Model Training

To build an efficient $\mathcal{EK}$ model, we propose a self-supervised enhanced training scheme:

Data Construction Accurate keyword extraction is critical for ensuring subsequent document recall precision. Since this task is relatively simple for LLMs, we opt to train a cost-effective Llama 3.2-3B model. Specifically: **(1)**Execute the described retrieval workflow on the HotpotQA dataset using simple prompts, relying solely on keywords for document recall. **(2)**Validate keyword effectiveness by checking whether the retrieved documents contain the dataset-provided golden documents d_g. **(3)** Define a keyword validity indicator function:

$$\mathbb{I}(\mathcal{K}_i) = \begin{cases} 1, & \text{if } \exists d_g \in \mathcal{D}_{\text{gold}} \text{ s.t. } \mathcal{K}_i \subseteq \text{Terms}(d_g) \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

(4) According to the specified formula, collect effective keywords $\mathcal{K}^+ = \{k | \mathbb{I}(\mathcal{K}) = 1\}$ and their corresponding queries q_t during iterations.



Extract 1-2 keywords from the following question. The keywords should be phrases like numbers, property nouns, or proper nouns that can effectively distinguish the target document. The keywords should not have synonyms. Ensure the keywords are directly extracted from the question and provide them in a list format.
Note that each keyword consists of only one word.
For example:
["five", "museum"]
Input
Question: {question}

Figure 5: The instruction for the Keywords extract.

Model Fine-tuning Based on the Llama 3.2-3B (Dubey et al., 2024) architecture, we design an instruction-tuning objective:

$$\mathcal{L} = -\sum_{m} \log P(k_m | q_t; \theta) \tag{10}$$

where θ denotes the model parameters. This formulation aims to maximize the likelihood of generating discriminative keywords $\{k_m\}$ given input queries q_t. Consequently, it enables the model to learn question-aware keyword extraction patterns.

4 Experiments

This section systematically evaluates the effectiveness of the proposed method in typical multi-hop reasoning scenarios. We conducted comparative experiments on three benchmark multi-hop QA datasets and performed performance comparisons with current mainstream baseline models.

4.1 Datasets

This study selects three multi-hop QA benchmarks with distinct reasoning characteristics:
(1) HotpotQA (Yang et al., 2018) requires models to perform cross-document information integration for reasoning, with question designs mandating at least two inference steps.
(2) 2WikiMultiHopQA (Ho et al., 2020) is constructed based on structured Wikipedia knowledge, particularly emphasizing explainable causal reasoning path modeling, providing comprehensive explanations for multi-hop questions.
(3) MuSiQue (Trivedi et al., 2022) serves as a high-complexity benchmark, featuring question designs guaranteed through dual constraints: (a) minimum two-hop reasoning requirement; (b) answers cannot be directly obtained through single-hop retrieval.
Due to experimental resource constraints, we adopted a random sampling strategy to extract 500 samples from the original validation sets of each dataset to form our test set.

4.2 Evaluation Metrics

Following the latest research paradigm in multi-hop QA (Xu et al., 2024; Shi et al., 2024), we employ a three-level evaluation framework. **(a) Coverage Exact Match (CoverEM)**: Validates whether generated answers contain ground-truth answers through strict string matching. **(b) Token-level F1 Score**: Calculates precision (ratio of shared tokens in predicted text) and recall (ratio of shared tokens in reference text) by counting overlapping tokens between predicted and reference texts, with F1 score computed as their harmonic mean. Token matching is based on word frequency intersection. **(c) Semantic Accuracy (Acc†)**: To overcome limitations of rule-based metrics, we introduce Llama-3.1-8B-Instruct-based semantic evaluation (Shi et al., 2024). This model assesses semantic equivalence between generated and reference answers through structured prompting (see Appendix F for details), effectively capturing semantic-level similarity in open-domain generation.

Additionally, in order to evaluate the resource consumption of different methods, we record the number of sub-questions that each method generates and retrieves when addressing complex reasoning tasks, denoted as **#SQA**. A higher **#SQA** indicates a longer reasoning chain employed by the framework, which in turn corresponds to increased retrieval and reasoning overhead.

Method	HotpotQA				2WikiMultiHopQA				MuSiQue			
	#SQA	CoverEM	F1	ACC†	#SQA	CoverEM	F1	ACC†	#SQA	CoverEM	F1	ACC†
Llama-3.1-8B-Instruct without Retrieval												
Vanilla Chat	1.00	23.00	28.81	24.50	1.00	23.50	25.71	19.00	1.00	4.00	7.08	4.50
Direct CoT	1.00	33.00	35.90	36.00	1.00	28.00	30.73	29.00	1.00	9.00	14.88	13.00
Llama-3.1-8B-Instruct with Retrieval												
Direct RAG	1.00	37.24	40.84	42.35	1.00	22.61	24.05	22.11	1.00	6.81	11.05	8.38
Self-Ask	4.98	35.20	38.33	37.20	4.95	44.60	44.03	43.60	4.99	13.23	16.93	15.23
SearChain	3.75	40.16	43.06	44.18	3.62	48.60	44.72	43.80	3.73	13.05	17.77	17.47
GenGround	5.00	36.20	39.88	39.60	5.00	38.40	35.91	33.60	5.00	10.40	15.99	11.60
DEC (Ours)	3.47	**47.19**	**50.96**	**49.60**	3.28	**49.18**	**46.54**	**45.70**	3.43	**17.21**	**20.96**	**19.26**
GPT-4o without Retrieval												
Vanilla Chat	1.00	33.50	41.14	32.00	1.00	33.50	35.98	30.00	1.00	10.50	17.28	12.50
Direct CoT	1.00	55.00	59.31	58.50	1.00	62.00	59.97	55.50	1.00	25.50	28.59	29.50
GPT-4o with Retrieval												
Direct RAG	1.00	57.00	57.39	62.00	1.00	66.50	55.79	59.00	1.00	22.00	24.70	27.00
Self-Ask	3.09	53.40	54.49	55.60	3.65	72.00	66.79	67.60	3.44	20.2	26.17	22.80
SearChain	2.24	53.80	58.12	58.00	2.90	63.40	55.66	54.60	2.62	23.45	27.77	27.66
GenGround	5.00	**60.50**	**62.37**	**63.00**	5.00	74.50	68.97	63.00	5.00	23.00	24.63	22.50
DEC (Ours)	3.32	58.52	62.11	60.32	3.41	**78.51**	**73.26**	**71.29**	3.22	**27.91**	**31.30**	**30.72**

Table 1: Evaluation results of DEC and the baseline on three QA benchmarks. #SQA denotes the average sub-questions generated per complex question for retrieval. ACC† indicates semantic similarity assessed by an LLM.

4.3 Baseline Models

The selection of baseline models is based on an incremental comparative approach, including two main categories: non-retrieval methods and retrieval-enhanced methods.

Non-retrieval Baselines:

(1) Vanilla Chat: Directly uses the original question as input to test the zero-shot question-answering capability of large language models.

(2) Direct CoT (Wei et al., 2022; Kojima et al., 2022): Employs Chain-of-Thought techniques to guide the model in explicitly generating reasoning paths, thereby improving answer generation quality through step-by-step derivation.

Retrieval-Enhanced Baselines:

(3) Direct RAG: Constructs a single-stage retrieval-enhanced framework that performs dense retrieval on the input question, selecting the top 10 most relevant documents as contextual input for the model.

(4) Self-Ask (Press et al., 2023): Utilizes an explicit question generation mechanism to transform composite questions into a sequence of sub-questions, thereby establishing a transparent and controllable multi-step reasoning architecture.

(5) SearChain (Xu et al., 2024): Builds a dynamic retrieval-generation interaction chain that addresses the challenges of real-time knowledge updates through iterative context augmentation.

(6) GenGround (Shi et al., 2024): Utilizes a two-stage framework consisting of hypothesis generation and retrieval verification. In the first stage, candidate answers are generated, followed by retrieval in the second stage to correct these candidates, effectively mitigating error propagation.

4.4 Implementation Details

We conduct experiments on both mainstream closed-source LLM and lightweight LLM to demonstrate the generalization performance of our approach. Specifically, we validate our method on the commercial GPT-4o model (Hurst et al., 2024) as well as on the open-source Llama-3.1-8B-Instruct model (Dubey et al., 2024). Regarding the construction of the knowledge base, HotpotQA and 2WikiMultiHopQA utilize the Wikipedia snapshots provided by the dataset creators, whereas MuSiQue—lacking an officially curated knowledge base—adopts the 2020 Wikipedia version from 2WikiMultiHopQA. The retrieval system employs the E5-base dense retrieval model (Wang et al., 2022). More implementation details are provided in Appendix A.

4.5 Experimental Results

As shown in Table 1, the DEC framework demonstrates superior or competitive performance across models of varying parameter scales (Llama-3.1-8B-

method	HotpotQA		2WikiMultiHopQA		MuSiQue	
	CoverEM	F1	CoverEM	F1	CoverEM	F1
DEC (Ours)	47.19	50.96	49.18	46.54	17.21	20.96
w/o EK	46.00(↓2.5%)	49.04(↓3.7%)	42.60(↓13.3%)	41.40(↓11.0%)	16.87(↓1.9%)	20.42(↓2.6%)
w/o QD	44.44(↓5.8%)	45.97(↓9.8%)	35.68(↓27.5%)	38.28(↓17.7%)	12.24(↓28.9%)	17.40(↓17.0%)
w/o QR	36.47(↓22.7%)	40.56(↓20.4%)	27.80(↓43.5%)	28.54(↓38.7%)	8.74(↓49.2%)	14.78(↓29.5%)
w/o COT	39.00(↓17.3%)	43.18(↓15.3%)	26.80(↓45.5%)	27.82(↓40.2%)	8.62(↓49.9%)	12.95(↓38.2%)
Self-Ask	35.20	38.33	44.60	44.03	13.23	16.93
Self-Ask$_{w/EK}$	43.40	47.69	47.28	45.39	15.07	17.70

Table 2: Results of the ablation study conducted on Llama-3.1-8B-Instruct. "EK", "QD", "QR", and "COT" denote keyword extraction-enhanced retrieval, structured question decomposition, dynamic query rewriting, and the combination of QD and QR modules with Chain-of-Thought reasoning, respectively.

Instruct and GPT-4o) and three multi-hop reasoning benchmark datasets. Experimental results validate the significant effectiveness and strong generalization capability of the proposed multi-hop reasoning architecture. Through in-depth analysis, we derive the following key findings:

Correlation Between Model Capacity and Hallucination Suppression The closed-source GPT-4o exhibits exceptional zero-shot reasoning capabilities in the retrieval-free Direct CoT method, significantly outperforming Llama-3.1-8B (e.g., a 34% gap in CoverEM on the 2WikiMultiHopQA dataset). Notably, GPT-4o achieves a 37.95% reduction in reasoning chain length compared to Llama-3.1-8B (HotpotQA dataset/Self-Ask method) while improving CoverEM by 18.2%. This phenomenon confirms the positive correlation between model parameter scale and reasoning accuracy: expanding model capacity enhances semantic understanding depth and logical coherence, thereby reducing error-prone reasoning path generation and suppressing hallucination.

Quantitative Comparison of Reasoning Mechanism Efficiency Compared to iterative reasoning baselines (Self-Ask, GenGround), the DEC framework reduces reasoning chain length by 27% on average for Llama-3.1-8B while maintaining overall performance superiority. This discrepancy highlights two core advantages of the single-stage reasoning chain generation mechanism: (1) mitigating semantic deviation in intermediate steps through logical chain-of-thought; (2) avoiding error accumulation effects inherent in multi-step iterative generation, particularly critical for resource-constrained lightweight models.

Synergistic Gains from Retrieval-Generation Coordination On Llama-3.1-8B, the performance advantage of DEC over hypothesis-refinement methods (SearChain, GenGround) validates the effectiveness of our structured problem decomposition and dynamic query rewriting approach. By reducing the output of untrusted information, we successfully suppressed hallucination generation in lightweight models. On the 2WikiMultiHopQA dataset, DEC achieves a 12.1% improvement in semantic accuracy (ACC†) over GenGround for Llama-3.1-8B, significantly exceeding the 8.29% gain observed with closed-source models. This finding suggests that the proposed method provides better generalization capabilities, particularly in enhancing the performance of resource-constrained models.

5 Further Analyses

5.1 Ablation Study

To validate the effectiveness of the modules in the DEC framework, we conducted experiments by removing individual modules or key methods from the framework (see Appendix B for more details). The experimental results demonstrate the effectiveness of the proposed method design from the following three perspectives:

(1) Impact of Keyword Extraction on Retrieval Quality When the keyword extraction module was removed (w/o EK), the performance metrics on three datasets showed a significant decrease (1.9%-13.3%). This indicates that extracting discriminative keywords through the EK model can effectively focus on the core retrieval needs, avoiding document noise caused by generic vocabulary.

(2) Synergistic Effect of Question Rewriting and Reasoning Chain Decomposition When the structured question decomposition module is removed (w/o QD), performance drops by 5.8% to 28.9%, indicating that the process of breaking down complex questions into structured reasoning chains provides essential contextual dependencies and logical constraints for subsequent question rewriting and retrieval.

When the question rewriting module is removed (w/o QR), performance declines by 22.7% to 49.2%, confirming the critical role of the dynamic question rewriting mechanism. By incorporating the question-answer history to supplement implicit semantics, it significantly enhances both the completeness and retrieval relevance of sub-questions generated through structured decomposition.

Furthermore, removing both the question decomposition and question rewriting modules—thereby disabling the dynamic chain-of-thought construction approach (w/o COT)—results in a performance degradation of up to 49.9%. This underscores the critical role and effectiveness of the synergistic interaction between these two components.

(3) Performance Comparison between Structured Decomposition and Iterative Methods A comparison between Self-Ask and its enhanced version Self-Ask w/EK revealed: 1. Introducing keyword extraction improved Self-Ask's performance across all three datasets (2.5%-15.6%), demonstrating the effectiveness and generalizability of our retrieval strategy. 2. However, DEC still maintains a significant advantage over Self-Ask w/EK (MuSiQue F1 +18.4%), suggesting that reasoning chain decomposition based on priors can more systematically plan the problem-solving path, avoiding path deviation and semantic accumulation errors commonly encountered in iterative methods.

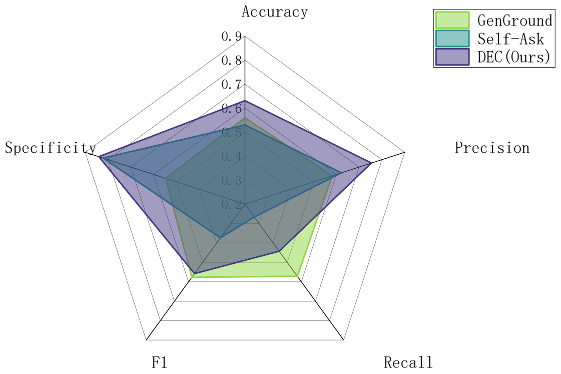


Figure 6: Performance metrics of DEC and two baselines on MuSiQue_full for unanswerable questions.

5.2 Performance Advantages

Advantages in Handling Unanswerable Queries In RAG tasks—where stringent reliability is paramount—accurately identifying unanswerable queries is crucial. Our experimental results on the MuSiQue_full dataset demonstrate that the proposed DEC method significantly outperforms existing approaches in dealing with unanswerable questions. Specifically, the DEC method achieves an accuracy of 63.13% in answerability prediction, compared to 56.00% for GenGround and 53.00% for Self-Ask, while also yielding a more balanced performance with a precision of 75.41% and an F1 score of 55.76%.

This further confirms our approach's advantages in reducing hallucinations and enhancing judgment accuracy on unanswerable questions.

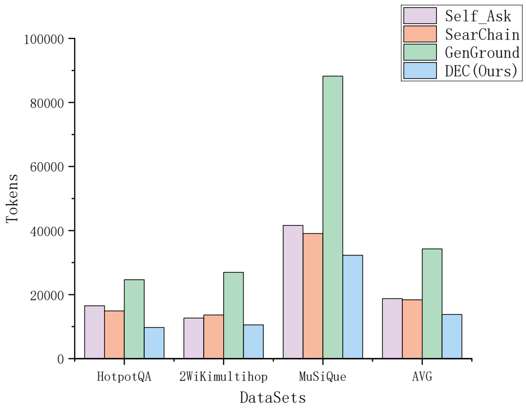


Figure 7: Statistics on the average number of tokens consumed to correctly answer a question for DEC and three baselines across three datasets.

Saving Computational Resources In parallel with our main experiments, we also evaluated the resource efficiency of different methods by recording the total token consumption for each dataset task. Specifically, we computed the total token consumption of Llama 3.1-8B-Instruct for each task and normalized it by the number of correct answers (i.e., achieving a semantic accuracy of 1). The resulting metric, **ATC**, represents the **a**verage **t**oken **c**onsumption per correct answer, allowing us to assess the trade-off between accuracy and resource expenditure. As shown in Figure 7, our method consistently consumes fewer tokens per correct answer across all tasks, demonstrating superior resource efficiency while maintaining answer quality.

6 Conclusions

In summary, our study presents a novel RAG paradigm. The DEC utilizes LLMs to directly generate logical reasoning chains, thereby minimizing the introduction of hallucinated information. It also employs iterative query rewriting to incorporate key details that may have been overlooked during the reasoning and querying processes. Furthermore, in the document retrieval phase, we implement a query strategy enhanced by discriminative keywords, effectively improving the recall rate of crucial documents. Experimental results indicate that across multiple datasets and assessments using commercial closed-source and lightweight LLMs, our method achieves—or even exceeds—the performance of existing approaches while consuming significantly fewer computational resources. Meanwhile, our approach also excels in identifying unanswerable questions. These advantages are particularly notable when applied to lightweight models with lower parameter counts.

Limitations

Although the framework proposed in this paper demonstrates encouraging results, there are still some limitations:

Our approach heavily relies on the high-quality decomposition and knowledge supplementation of complex questions, especially when dealing with multiple sub-questions. The effectiveness of this process depends on the semantic understanding of the initial question and the accurate supplementation of relevant knowledge. If the initial decomposition or knowledge supplementation is insufficient or biased, it may lead to failure in subsequent reasoning and retrieval, thus affecting the accuracy of the final answer.

Although we have achieved significant performance improvements on lightweight LLMs, the gap in reasoning and expressive capabilities due to differences in model scale remains substantial. When faced with highly complex questions, lightweight models in our approach still struggle to reach the performance of LLMs with a higher parameter count.

Ethics Statement

This study leverages large-scale language models to implement decomposition of reasoning chains and enhancement of retrieval, aiming to improve the response effectiveness for multi-hop questions. Throughout the research process, we strictly adhere to academic ethical standards to ensure the rigor and effectiveness of the work. With the exception of the GPT series models, all datasets, models, and methods used are publicly available and free to access, providing a high level of transparency and reproducibility for the experiments. We strive to use open-source data and frameworks to minimize potential biases as much as possible and promote fairness. Meanwhile, we ensure that our research does not harm any individual or group, nor does it involve any form of deception or information misuse.

Acknowledgments

This project is funded by Zhiyuan Laboratory (NO. ZYL2024021) and CCF-Zhipu AL Large Model Fund (NO.202221).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. 2024. Evaluating correctness and faithfulness of instruction-following models for question answering. *Transactions of the Association for Computational Linguistics*, 12:681–699.

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*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3563–3578, Bangkok, Thailand. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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.

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.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

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

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

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*.

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.

Ruosen Li and Xinya Du. 2023. Leveraging structured information for explainable multi-hop question answering and reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6779–6789, Singapore. Association for Computational Linguistics.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

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.

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.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. musique: Multihop 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.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2212.03533*.

Liang Wang, Nan Yang, and Furu Wei. 2023b. 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.

Zihao Wang, Anji Liu, Haowei Lin, Jiaqi Li, Xiaojian Ma, and Yitao Liang. 2024. Rat: Retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation. *arXiv preprint arXiv:2403.05313*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2024. Search-in-the-chain: Interactively enhancing large language models with search for knowledge-intensive tasks. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 1362–1373, New York, NY, USA. Association for Computing Machinery.

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.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. 2024. Chain-of-note: Enhancing robustness in retrieval-augmented language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14672–14685, Miami, Florida, USA. Association for Computational Linguistics.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Liu Yong, and Shen Huang. 2024. End-to-end beam retrieval for multi-hop question answering. 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 1718–1731, Mexico City, Mexico. Association for Computational Linguistics.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 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)*, Bangkok, Thailand. Association for Computational Linguistics.

Ziyuan Zhuang, Zhiyang Zhang, Sitao Cheng, Fangkai Yang, Jia Liu, Shujian Huang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. 2024. EfficientRAG: Efficient retriever for multi-hop question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3392–3411, Miami, Florida, USA. Association for Computational Linguistics.

A Implementation Details

To ensure experimental generalizability and reproducibility, both the generator and retriever components were implemented with default parameter configurations. Specifically, we deployed the Llama model on a single NVIDIA A6000 GPU using the vLLM (Kwon et al., 2023) framework, while the E5 retriever was constructed through the FlashRAG (Jin et al., 2024) framework.

For the number of retrieved texts, we have made effort to adhere to the original configurations of the respective methods to ensure fairness. Specifically, Self-Ask selects the top three most relevant documents per iteration; GenGround employs a batch verification strategy by selecting three documents based on relevance per batch, with a maximum of three batches (i.e., a total of nine documents); SearChain, due to methodological constraints, selects only the single most relevant document for answer modification in each iteration; the DEC method selects the two most relevant documents from the top 10 along with documents that match specific keywords; and Direc RAG uses the top 10 most relevant documents per iteration.

For the training of the EK model, we employ the Llama-3.2-3B-Instruct architecture (Dubey et al., 2024) as the base model, utilizing Low-Rank Adaptation (LoRA) for efficient parameter fine-tuning with a learning rate of 5e-5 and training for two batches. We completed the aforementioned training using the LLaMA Factory (Zheng et al., 2024) framework. The model achieves convergence after 30 minutes of training on a single NVIDIA A6000 GPU, leveraging a dataset of 1,000 multi-hop question answering samples.

B Ablation Study Details

To thoroughly investigate the roles of the key components in our proposed DEC method, we designed the following ablation experiments:

w/o EK: In this variant, the keyword extraction step is removed and document retrieval is performed using only the rewritten query (selecting the top three most relevant documents each time). This configuration is intended to verify the role of the keyword extraction module in filtering discriminative retrieval cues.

To validate the improvement of the EK model on the recall rate of golden documents, we conducted systematic comparative experiments on the 2WikiMultihopQA dataset. As shown in Table 3, after applying the EK model for keyword extraction–enhanced retrieval, the number of successfully matched documents increased significantly from 861 (70.69%) with the baseline method to 1009 (82.84%). In particular, in the full recall evaluation at the question level, the complete recall rate of golden documents improved from 46.6% (233/500) to 65.6% (328/500), representing a relative increase of 40.8%. These results indicate that the EK model effectively increases the coverage of relevant documents during retrieval, thereby significantly enhancing the likelihood of fully recalling the gold documents for complex questions.

w/o QD: In this variant, we remove the question decomposition module, while still preserving the iterative nature of query rewriting from the original DEC method. Specifically, without performing complex question reasoning chain decomposition, we conduct five iterations (a number that corresponds to the maximum iteration count set by many previous iterative methods), where in each iteration the original complex question is rewritten once based on the reasoning history, while retaining the keyword extraction module. This experiment aims to demonstrate the advantage of our "first performing structured question decomposition followed by query rewriting" method over the "generate iteratively while reasoning" approach, further showcasing the importance of the synergistic effect between query rewriting and question decomposition for the overall framework performance.

w/o QR: Here, the question rewriting step is omitted, and retrieval relies solely on the sub-questions generated in the initial reasoning chain, while all other procedures remain identical to those in DEC. This variant examines the effectiveness of the question rewriting module in enriching query information and integrating the QA history.

Metric	DEC	w/o EK
Total Docs	1218	1218
Successful Matches	1009	861
Document Matching Ratio	82.84	70.69 (↓14.7%)
Fully Recalled Questions	328	233
Fully Recalled Ratio	65.60	46.60 (↓29.0%)

Table 3: Comparison of Retrieval Performance with and without the EK Model on the 2WikiMultihopQA Dataset

w/o COT: In this setup, the reasoning chain generation step is excluded. Instead, document retrieval and answer generation are carried out directly using the original complex query in conjunction with keyword extraction. This setting validates the importance of decomposing the question into a multi-step reasoning chain for enhancing both retrieval and inference performance.

Furthermore, to demonstrate the superiority of our strategy—generating a reasoning chain prior to rewriting the query—we also compare against the typical iterative sub-question generation method, Self-Ask, as well as a variant of Self-Ask that incorporates a keyword extraction retrieval module (denoted as Self-Ask w/EK).

C Performance evaluation details

C.1 Handling Unanswerable Questions

In order to conduct a comprehensive evaluation of the performance disparities between our model and baseline approaches for unanswerable question detection, we selected the MuSiQue-Full subset from the MuSiQue benchmark dataset as the experimental platform. This subset is particularly characterized by its construction of contrastive pairs that consist of both answerable and unanswerable questions. In contrast to the MuSiQue-Ans dataset, which solely comprises answerable questions, the MuSiQue-Full subset introduces unanswerable contrastive questions, thereby establishing a more stringent evaluation setting. This setup effectively mitigates the risk of model exploitation via irrelevant reasoning paths, providing a more robust foundation for assessing the model's multi-hop reasoning capabilities and overall robustness. Due to computational resource constraints, we randomly selected 200 multi-hop questions from the validation set of MuSiQue-Full to form the test set for this experiment.

Besides the standard binary classification metrics, including Accuracy, Precision, Recall, F1-Score, and Specificity, we also introduce two supplementary metrics to assess the model's accuracy in answering questions and distinguishing between correct and incorrect responses.

(1) Conditional Accuracy (C Acc) Conditional accuracy refers to the proportion of correct answers when the model predicts a question as answerable

Eval. Metrics	Self Ask	GenGround	DEC (Ours)
Accuracy	53.00	56.00	**63.13**
Precision	62.22	58.25	**75.41**
Recall	26.67	**57.14**	44.23
F1-Score	37.33	**57.69**	55.76
Specificity	82.11	54.74	**84.04**
C Acc	21.43	35.00	**58.70**
O Acc	42.00	36.50	**53.54**

Table 4: MuSiQue-Full Evaluation Metrics Comparison

and the question is indeed answerable. It is calculated as:

$$CA = \frac{\text{Correct Answers}}{TP + \text{True Answerable Subset of FP}} \tag{11}$$

Note: If the model predicts the question as answerable but the question is actually unanswerable (i.e., FP), then the accuracy field is invalid and must be excluded.

(2) Overall Accuracy (O Acc) To comprehensively evaluate the model's end-to-end performance, we combine both the answerability prediction and the correctness of the answers. This holistic evaluation is captured by the overall accuracy metric. Overall accuracy evaluates the model's performance under the following two conditions:

- Correctly abstaining from answering unanswerable questions (TN),
- Correctly answering answerable questions with correct answers (TP and accuracy=true).

It is computed as:

$$\text{Overall Accuracy} = \frac{TN + TP_{\text{acc}}}{\text{Total Number of Samples}} \tag{12}$$

where TP_{acc} denotes the true positives with correct answers.

Among the aforementioned evaluation metrics, the accuracy metric for answerable questions is determined using semantic accuracy (ACC† 4.2). The specific results of our method compared to the baseline in this experiment are shown in Table 4.

C.2 Saving Computational Resources

In the experiment, we simultaneously recorded the token consumption for each method when solving 500 questions from different datasets using Llama 3.1-8B-Instruct, as well as the average number of tokens required to correctly answer a single question (ATC). The specific data is shown in Table 5.

method	Tok. Cons	ATC
HotpotQA		
Self_Ask	3074644	16529.38
SearChain	3281899	14916.61
GenGround	4882838	24660.80
DEC (Ours)	**2400690**	**9719.08**
2WikiMultiHopQA		
Self_Ask	2760014	12660.61
SearChain	2988310	13645.25
GenGround	4530000	26964.29
DEC (Ours)	**2349779**	**10536.37**
MuSiQue		
Self_Ask	3161428	41599.00
SearChain	3401388	39096.14
GenGround	5117786	88237.69
DEC (Ours)	**3035764**	**32299.21**

Table 5: The resource consumption of different methods.

D Further Analyses

D.1 Experiments on the 14B-Scale Model

To further validate the generalization capability of the proposed method across different model scales, we conducted additional experiments based on the Qwen2.5-14B-Instruct model, using the same experimental setup as in the main experiments. The results are presented in Table 6. As shown in the table, the proposed method outperforms all baseline methods on nearly all evaluation metrics, fully demonstrating its effectiveness and generalization capability across different parameter scales.

D.2 The Effectiveness of the DEC in Mitigating Model Hallucinations

The core idea behind the DEC method is to decompose the end-to-end reasoning process into a series of relatively simple task-specific modules, generating intermediate and final outputs under explicit knowledge constraints. This design effectively reduces the risk of model hallucinations. Based on this framework, we conducted evaluation experiments along two dimensions: (1) measuring the consistency between the outputs of the keyword-extraction module and the original text; and (2) assessing the fidelity of the core content after structured question decomposition.

Method	HotpotQA				2WikiMultiHopQA				MuSiQue			
	#SQA	CoverEM	F1	ACC†	#SQA	CoverEM	F1	ACC†	#SQA	CoverEM	F1	ACC†
Qwen2.5-14B-Instruct without Retrieval												
Vanilla Chat	1.00	23.50	29.53	31.00	1.00	22.00	25.71	26.50	1.00	4.00	7.95	7.50
Direct CoT	1.00	34.50	34.04	39.50	1.00	31.50	29.85	35.00	1.00	13.50	18.96	24.00
Qwen2.5-14B-Instruct with Retrieval												
Direct RAG	1.00	51.05	48.33	58.70	1.00	50.00	40.75	44.68	1.00	12.82	16.96	20.51
Self-Ask	3.98	50.50	52.65	57.00	4.04	59.00	53.28	59.00	4.37	20.00	22.20	22.50
SearChain	2.79	59.00	53.72	**64.00**	3.12	48.00	39.48	42.50	3.31	16.50	22.26	23.00
GenGround	5.00	54.00	57.04	61.50	4.99	63.00	60.41	63.00	5.00	17.50	22.78	23.50
DEC (ours)	2.74	**60.00**	**60.68**	63.00	2.75	**70.35**	**66.83**	**69.85**	3.04	**22.56**	**25.45**	**28.72**

Table 6: Performance of DEC and Other Baseline Approaches on Qwen2.5-14B-Instruct

D.2.1 Hallucination Assessment of the Keyword Extraction Module

In the Discriminative Keyword Extraction (EK) module, the model's sole responsibility is to extract retrieval-appropriate substrings from each subquestion. Because the task is so well-defined, any deviation can be directly attributed to hallucination. We quantified the hallucination rate of the EK module by calculating the match rate between the keywords extracted by the EK model and the corresponding substrings in the original sub-questions. The results are presented in Table 7.

Model	hotpotqa	2wikimqa	musique
Llama-3.1-8B	97.39%	98.81%	96.44%
GPT-4o	98.31%	99.06%	97.75%

Table 7: EK Model Extraction Accuracy Statistics

As the table shows, across all datasets the EK module's keyword-extraction match rate exceeds 96%, indicating that the module virtually never "invents" keywords and thus has an extremely low hallucination risk.

D.2.2 Content Fidelity after Structured Question Decomposition

The structured decomposition module must logically reorganize the original, complex question—an evaluation that poses automated challenges. Nonetheless, the core linguistic elements before and after decomposition should remain largely unchanged, with only additional logical auxiliary words (such as interrogative words) introduced. Based on this, we removed stop words from the decomposed questions to retain the core vocabulary and then computed the proportion of these core words present in the original question. Considering factors such as variations in word forms, we allowed for some degree of fuzzy matching—deeming a word as present in the original question when its similarity exceeds 0.8. The results appear in Table 8.

Model	hotpotqa	2wikimqa	musique
Llama-3.1-8B	84.25%	85.29%	89.21%
GPT-4o	92.96%	90.86%	95.33%

Table 8: Problem Decomposition Fidelity Statistics

It can be observed that when using the GPT-4o model, the average matching ratio of the core vocabulary in the decomposed questions compared to the original question exceeds 93%, and even for the Llama-3.1-8B-Instruct model (where hallucinations are relatively more apparent), the average matching ratio remains above 86%. Although hallucinations in question decomposition may have some impact, the overall performance is still within an acceptable range and sufficient for practical applications.

The two sets of experiments above confirm that DEC's modular design combined with explicit knowledge constraints significantly reduces hallucination risk throughout the entire pipeline—from keyword extraction to question reconstruction. Even with smaller parameter models, DEC maintains high match rates and high content fidelity, ensuring the overall system's stability and reliability.

D.3 Accuracy Analysis of the DEC Method Across Question Types

To evaluate the capability of the DEC method across different question types, we analyzed its performance on the 2WikiMultiHopQA dataset,

Method	Bridge Comparison			Compositional			Comparison			Inference		
	CoverEM	F1	ACC†	CoverEM	F1	ACC†	CoverEM	F1	ACC†	CoverEM	F1	ACC†
Llama-3.1-8B-Instruct												
Self-Ask	52.29	51.97	52.29	29.41	26.83	29.41	76.74	74.30	74.41	12.06	22.20	8.62
DEC	59.25	57.67	53.70	42.71	34.03	37.68	62.69	63.52	61.90	21.81	31.03	21.81
GPT-4o												
Self-Ask	81.65	81.10	81.65	57.35	46.11	49.50	88.37	86.82	86.04	68.69	68.05	63.79
DEC	90.74	90.18	88.88	61.27	48.28	47.05	96.09	96.09	92.96	77.58	79.20	75.86

Table 9: Performance of DEC across Four Question Types

which defines four categories of questions: bridge_comparison, compositional, comparison, and inference. The detailed performance metrics for each category are shown in the Table 9.

Specifically, DEC demonstrates a notable advantage in bridge_comparison and comparison questions, suggesting that for these types, the accurate identification and integration of relational evidence is more straightforward. In contrast, the model performs relatively worse on compositional and inference questions, indicating that these categories require more comprehensive retrieval of intermediate facts and deeper logical integration. Under such circumstances, the iterative completion process employed by DEC may still miss critical reasoning steps, resulting in lower CoverEM and F1 scores.

Nevertheless, DEC generally outperforms Self-Ask across various question types and model sizes, validating the effectiveness of our strategy for improving overall model performance.

E Assessing the Reliability of Smaller LLMs for Semantic Evaluation

Given that answers in QA datasets are typically short, the associated semantic evaluation tasks are relatively straightforward. Smaller LLMs, such as Llama-3.1-8B-Instruct, are capable of producing highly reliable evaluation results. To investigate whether there is a performance gap between evaluations conducted using Llama-3.1-8B-Instruct and those using GPT-series models (e.g., GPT-4o), we conducted a comparative analysis.

Specifically, we first used Llama-3.1-8B-Instruct to evaluate the outputs of a GPT-based DEC on the HotpotQA dataset, which exhibits high accuracy, and a Llama-3.1-8B-based DEC on the MusiQue dataset, which has a higher error rate. We then used GPT-4o to re-evaluate the same model outputs under identical prompting conditions as those previously used for Llama-3.1-8B-Instruct. This ensured a fair comparison between the two evaluators under consistent evaluation settings. The results are presented in Table 10.

Data	Evaluation Consistency Ratio (%)
DEC-GPT on HotpotQA	90.36%
DEC-Llama3.1-8B on MusiQue	95.57%

Table 10: Evaluation Consistency Ratio

The results demonstrate that the agreement between the two evaluation methods exceeds 90% in both cases, supporting the validity of using Llama-3.1-8B-Instruct for semantic evaluation tasks.

F Prompts

This section details the methods we employed and the prompts used during the experiments. Note that the examples given in the prompts were not part of the dataset used for testing.

Prompt for semantic accuracy

You are an experienced linguist who is responsible for evaluating the correctness of the generated responses.

You are provided with question, the generated responses and the corresponding ground truth answer.

Your task is to compare the generated responses with the ground truth responses and evaluate the correctness of the generated responses.

##Example:

Example_1:

User input:

-Question: The city where Alex Shevelev died is the capital of what region?

-Ground-truth Answer: the Lazio region

-Prediction: the answer is Lazio

Model output:

-Correctness: yes

Example_2:

User input:

-Question: Which drink is larger, the Apple-Kneel or the Flaming volcano?

-Ground-truth Answer: The flaming volcano

-Prediction: The Apple-Kneel

Model output:

-Correctness: no

Now analyze the following question.Please be sure to output in the agreed format.

User input:

-Question: question

-Ground-truth Answer: {answer}

-Prediction: {prediction}

Model output:

Table 11: Prompt for semantic accuracy

Prompt for Dynamic context-enhanced query rewriting

You are an auxiliary query assistant who modifies queries to better find answers to solve problems.

Follow these precise steps:

1. Dependency Check: For each sub-question, identify if it depends on the answer to any previous sub-question.

- State the dependency reason if it exists, otherwise, state "None".

2. Dynamic Adjustment: Modify the sub-question to include necessary information if a dependency is present.

- If no change is required, keep the original sub-question.

Input Data:

- Key_Question:The key question that ultimately needs to be answered. The modified sub-questions should be queries that can provide crucial information for answering this question.

- Previous_QA_History: "The question-and-answer history of previous sub-questions, which provides crucial information for solving the key question and for the rewriting of subsequent sub-questions.

- Modifiable_Question: The sub-questions that need to be modified.

Format your output as follows:

Inference_process: Dependency reason or 'None' if not dependent

Modified_question: Modified sub-question or original if no changes are required

##Example:

- Key_Question:When was the founder of craigslist born?

- Previous_QA_History:

sub_question_1:Who was the founder of craigslist?, sub_answer:Craigslist was founded by Craig Newmark.

- Modifiable_Question:"When was him born?"

Inference_process: The sub-question "When was him born?" depends on the answer to sub-question_1 because "him" refers to the previously identified founder, Craig Newmark. Modified_question: When was Craig Newmark born?

Now analyze the following question. Please be sure to output in the agreed format.

User input:

- Key_Question:{question}

- Previous_QA_History:{history}

- Modifiable_Question:{sub_question}

Model output:

Table 12: Prompt for Dynamic context-enhanced query rewriting

Prompt for answering sub_question

Answer the following question briefly based on relevant information:

Question: {sub_question}

Context: {rel_text}

Prompt for reasoning through the chain of thought to the answer

Synthesize an answer to the original question based on the answers to sub-questions:

"Your reasoning process should be separated into two fields from the answer. In the answer field, please provide the answer as concisely as possible. The answer should be given in the form of words or phrases as much as possible.

Input Data:

- Original_Question:The key question that ultimately needs to be answered.

- Evidence:Question-and-answer pairs of the sub-questions split from the original question, which are used to answer the final original question.

Format your output as follows:

Inference_process: Your reasoning process

Answer: Modified Provide answers as concisely as possible

##Output Example:

Inference_process: Based on the sub-questions and answers, I identified the series that matches the description as Animorphs, a science fantasy young adult series told in first person. The series has companion books that narrate the stories of enslaved worlds and alien species, which aligns with the nature of the companion books in the Square Enix series.

Answer: Animorphs

Now analyze the following question. Please be sure to output in the agreed format.

User input:

- Original_Question:{question}

- Evidence:{history}

Model output:

Table 13: Prompt for reasoning through the chain of thought to the answer

Prompt for Direct CoT

You are a question-answering system capable of constructing a reasoning chain based on your world knowledge. Given the input question, follow these steps to infer the answer:
1. Break down the question and identify key facts that will help in the reasoning process.
2. Use your world knowledge to find relevant information that can help answer the question.
3. Build a chain of inferences leading to the final answer.
4. Format the output as follows:
- First, list the reasoning steps, clearly numbered.
- Then, conclude with the final answer in the format:
’So the final answer is: <answer>’
Example:
Question: What government position was held by the woman who portrayed Corliss Archer in the film Kiss and Tell?
Inference_process:
1. Kiss and Tell is a 1945 American comedy film starring then 17-year-old Shirley Temple as Corliss Archer.
2. Shirley Temple Black was named United States ambassador to Ghana and to Czechoslovakia and also served as Chief of Protocol of the United States.
So the final answer is: Chief of Protocol
Now, given the following question, please provide the inference process and the final answer.
Question: question

Prompt for Direct RAG

You are a question-answering system capable of combining world knowledge and information from provided documents to answer a question. Given the input question and a list of relevant documents retrieved based on the question, please follow these steps:
1. Read through the provided documents and identify relevant information.
2. Filter out irrelevant or redundant information and focus on the most useful content.
3. Combine the knowledge from the documents with your own world knowledge to construct a reasoning chain.
4. Format the output as follows:
- First, list the reasoning steps, clearly numbered, describing how you combined the information from the documents with your world knowledge.
- Then, conclude with the final answer in the format:
’So the final answer is: <answer>’
Example:
Question: What government position was held by the woman who portrayed Corliss Archer in the film Kiss and Tell?
Documents:
1. "Kiss and Tell" is a 1945 American comedy film starring Shirley Temple as Corliss Archer.
2. Shirley Temple Black served as U.S. ambassador to Ghana and Czechoslovakia and was also appointed Chief of Protocol.
3. The film was a major success in the 1940s, helping Shirley Temple become one of the most famous child stars of the era.
Inference_process:
1. The film "Kiss and Tell" featured Shirley Temple as Corliss Archer.
2. Relevant documents indicate that Shirley Temple Black later became a U.S. ambassador to two countries and served as Chief of Protocol.
3. Combining this with world knowledge about Shirley Temple’s later career, the most relevant position she held was Chief of Protocol.
So the final answer is: Chief of Protocol
Now, given the following question and documents, please provide the inference process and the final answer.
Question: {question}
Documents:{documents}

Table 14: Prompt for reasoning through the chain of thought to the answer