UniRAG: A Unified RAG Framework for Knowledge-Intensive Queries with Decomposition, Break-Down Reasoning, and Iterative Rewriting

Gun Il Kim[1], Jong Wook Kim[2*], Beakcheol Jang[1*]

[1]Graduate School of Information, Yonsei University, Seoul, South Korea
[2]Department of Computer Science, Sangmyung University, Seoul, South Korea
***Correspondence:** B.Jang(bjang@yonsei.ac.kr), J.W.Kim(jkim@smu.ac.kr)

Abstract

Knowledge-intensive queries require accurate answers that are explicitly grounded in retrieved evidence. However, existing retrieval-augmented generation (RAG) approaches often struggle with query complexity, suffer from propagated reasoning errors, or rely on incomplete or noisy retrieval, limiting their effectiveness. To address these limitations, we introduce UniRAG, a unified RAG framework that integrates entity-grounded query decomposition, break-down reasoning, and iterative query rewriting. Specifically, UniRAG decomposes queries into semantically coherent sub-queries, explicitly verifies retrieved sub-facts through a dedicated reasoning module, and adaptively refines queries based on identified knowledge gaps, significantly improving answer completeness and reliability. Extensive benchmark evaluations on complex question-answering datasets, including multi-hop HotPotQA and 2WikiMultihopQA, biomedical MedMCQA and MedQA, and fact-verification FEVER and SciFact, demonstrate that UniRAG consistently achieves performance improvements across various state-of-the-art LLMs, such as LLaMA-3.1-8B, GPT-3.5-Turbo, and Gemini-1.5-Flash.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language processing tasks (Brown et al., 2020; Grattafiori et al., 2024; Yang et al., 2024; Team et al., 2024), yet their static training hinders access to up-to-date factual knowledge and causes hallucinations due to difficulty keeping pace with evolving world knowledge. To mitigate these limitations, Retrieval-Augmented Generation (RAG) emerged (Izacard and Grave, 2021; Shao and Huang, 2022; Izacard et al., 2023; Shi et al., 2024); however, despite providing external grounding, these methods achieved only modest gains on complex queries, such as complex multi-hop and domain-specific, requiring nuanced external knowledge, highlighting the core challenge of effectively bridging the semantic gap between such complex questions and relevant information scattered across large knowledge bases.

Recent advancements in RAG have shifted towards enabling LLMs to actively guide retrieval for improved relevance and quality. Early active retrieval methods can be categorized into query decomposition, which breaks down complex questions into sub-queries (Min et al., 2019; Perez et al., 2020; Wei et al., 2022; Press et al., 2023; Trivedi et al., 2023), and query rewriting, where the LLM refines the query (Ma et al., 2023a; Ye et al., 2023). Despite their promise, these strategies face distinct challenges, in which decomposition often requires substantial supervision and struggles with optimal sub-query granularity (Guo et al., 2022; Dua et al., 2022), while rewriting can suffer from LLM hallucinations leading to inaccurate queries and often necessitates large training datasets (Ma et al., 2023b; Ye et al., 2023).

More recent efforts in active retrieval have focused on knowledge reasoning, where an LLM processes retrieved evidence to generate an answer or determine whether additional retrieval is needed (Yao et al., 2023; Shao et al., 2023; Wang et al., 2024; Asai et al., 2024). Reasoning-based approaches, such as Chain-of-Thought (CoT) prompting and iterative retrieval, enable deeper interaction with retrieved content and typically outperform earlier methods based solely on query decomposition or query rewriting. Despite these strengths, current knowledge reasoning methods remain limited by their susceptibility to propagating early errors, sensitivity to prompt variations, and the substantial computational overhead introduced by iterative retrieval-generation cycles (Wei et al., 2022; Yao et al., 2023; Shao et al., 2023).

These limitations become particularly critical when addressing knowledge-intensive queries,

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 18795–18810
November 4-9, 2025

whose accuracy heavily depends on the reliable grounding of evidence. Typical examples include (i) multi-hop questions requiring reasoning across multiple documents, (ii) complex-factoid queries demanding simultaneous verification of several independent constraints, and (iii) domain-specific questions involving interpretation or synthesis of externally retrieved domain knowledge. Existing knowledge reasoning-based methods frequently struggle to explicitly verify every required aspect or to adequately handle incomplete or noisy retrieval. Consequently, their answers often suffer from incompleteness or insufficient evidential support, significantly reducing their reliability in practice.

Thus, in this paper, we introduce UniRAG, a novel unified RAG framework specifically designed to enhance performance on knowledge-intensive queries. UniRAG synergistically integrates three complementary active retrieval strategies–query decomposition, break-down reasoning, and iterative query rewriting–to effectively address existing limitations. The main contributions of this work are summarized as follows:

- We propose an entity-grounded query decomposition scheme within UniRAG that generates semantically coherent sub-queries, ensuring each sub-query accurately captures distinct informational aspects of the original query.
- We develop a break-down reasoning module where the LLM explicitly examines retrieved documents, verifies each individual sub-fact, and evaluates evidence sufficiency, thus systematically ensuring comprehensive support for the final answer.
- We design an iterative query rewriting mechanism that leverages knowledge gaps identified during the reasoning phase, enabling adaptive and targeted refinement of queries to address incomplete or insufficient retrieval effectively.
- We demonstrate the effectiveness of our proposed framework through extensive benchmark evaluations, showing consistent and substantial improvements over state-of-the-art retrieval-augmented LLM baselines

2 Related Works

Query Decomposition Previous research has extensively investigated the decomposition of complex questions into simpler sub-queries to enhance question answering capabilities, exploring methods ranging from supervised approaches for multi-hop reading comprehension (Min et al., 2019) to unsupervised techniques for question decomposition (Perez et al., 2020) and multi-stage frameworks designed to make multi-hop question-answering more interpretable (Fu et al., 2021). More recently, the advent of LLMs, it has spurred approaches that utilize prompting strategies, including CoT (Wei et al., 2022), Least-to-Most, and iterative successive prompting, to guide the decomposition process and facilitate complex reasoning (Guo et al., 2022; Dua et al., 2022; Trivedi et al., 2023).

However, these methods often rely on extremely large language models requiring significant computational resources (Guo et al., 2022), necessitate substantial amounts of costly or difficult-to-obtain intermediate supervision or synthetic data for complex reasoning types, introduce challenges in selecting the appropriate decomposition granularity (Dua et al., 2022), or depend on external models for factual correction that may lack interpretability or be limited in scope (Zhou et al., 2022).

Query Rewriting Query rewriting transforms complex questions into more effective inputs for retrieval systems, evolving from early rule-based and statistical methods to recent advancements leveraging LLMs to generate improved queries through prompting or fine-tuning (Ma et al., 2023b; Ye et al., 2023). More advanced techniques utilize LLMs to generate queries from multiple perspectives, such as simulating different user demographics, aiming to enhance retrieval robustness against query variations (Li et al., 2023a). Some frameworks employ iterative refinement processes where an LLM acts as a rewrite editor, often guided by reinforcement learning signals based on retrieval performance, emphasizing the generation of informative rewrites that provide rich context for the retriever (Ma et al., 2023b).

Despite its promise, LLM-based query rewriting faces significant challenges as LLMs can suffer from hallucination, leading to inaccurate or irrelevant generated queries (Ma et al., 2023b; Ye et al., 2023). Ensuring the generated rewrites are consistently correct, informative, and non-redundant is difficult (Ye et al., 2023). Furthermore, these approaches often require substantial computational resources or large datasets of high-quality human-annotated rewrites for effective training and reliable performance (Ma et al., 2023b).

Knowledge Reasoning Recent RAG research has increasingly focused on enabling LLMs to

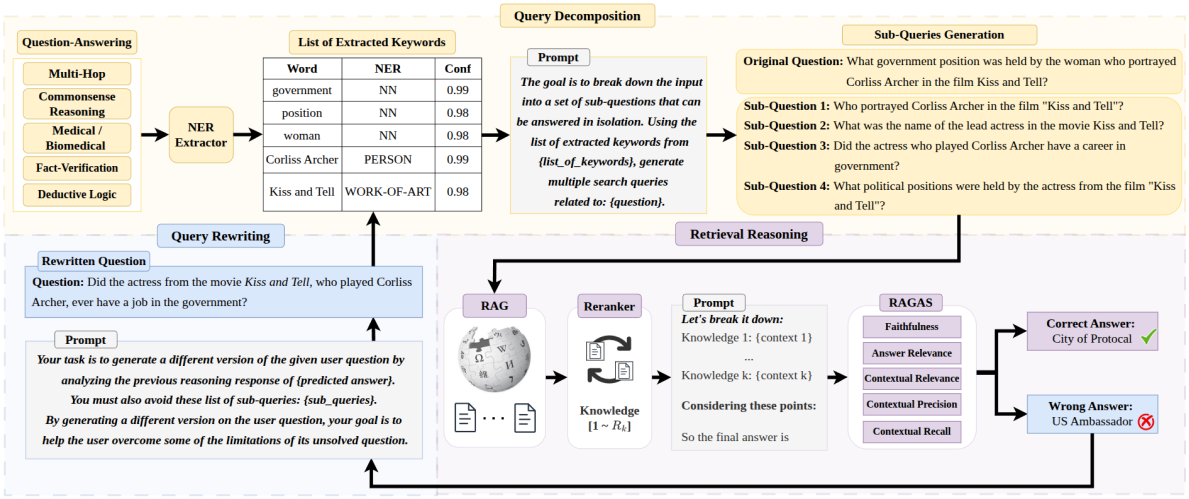

Figure 1: Architecture of the proposed UniRAG framework.

perform sophisticated external-knowledge-based reasoning, moving beyond simple Direct Prompting (Brown et al., 2020).

Early work on CoT prompting revealed its emergent reasoning ability is often limited to sufficiently large models, resulting in illogical outputs or failures at smaller scales and contributing to high computational costs, while also highlighting that CoT does not guarantee correct reasoning paths, can suffer from factual hallucinations, and is sensitive to prompting details (Wei et al., 2022). Approaches like ReAct (Yao et al., 2023), which synergize reasoning and actions by allowing interaction with external tools, introduce structural constraints that can reduce generation flexibility and may lead to reasoning errors or repetitive outputs if retrieval is non-informative or the model struggles to recover (Yao et al., 2023). Also, Self-Ask (Press et al., 2023) which decomposes questions into explicit follow-up steps, can face challenges in obtaining reliable intermediate answers or integrating external information sources effectively, sometimes being limited in scope or interpretability. Iterative retrieval-generation methods, like ITER-RETGEN (Shao et al., 2023), aim for improved relevance by guiding retrieval with previous generations, but risk using incorrect augmentation outputs and incurring iterative process overheads. Prompting-based RAG approaches generally face challenges in effectively balancing parametric versus non-parametric knowledge and suffer from unreliable automated evaluation metrics (Shao et al., 2023). BlendFilter (Wang et al., 2024) improves RAG for complex questions and noisy retrieval by blending multi-source query generation and using the LLM for filtering, although it introduces a tunable hyperparameter K, may not eliminate all noise, and the fundamental challenge of capturing relevance for complex queries persists.

The proposed UniRAG framework builds upon query decomposition, query rewriting, and knowledge reasoning, but differs from existing methods reviewed in this section in several important respects. Our query decomposition method is entity-grounded, explicitly preserving semantic alignment with the original query and ensuring each sub-query remains focused and contextually precise. Our knowledge reasoning employs a dedicated breakdown reasoning module to independently verify each sub-fact, jointly ensuring factual completeness and accuracy. Finally, our query rewriting component actively leverages insights from this reasoning process, adaptively refining queries to address identified knowledge gaps and improve retrieval quality.

3 Methods

In this section, we introduce UniRAG, a unified RAG framework designed to improve performance on knowledge-intensive queries. Our UniRAG framework consists of three main phases (Figure 1):

- *Entity-Grounded Query Decomposition:* Given an input query q, we first extract key entities using a pretrained language model (PLM). These entities are then used to prompt the LLM to decompose q into a set of focused sub-queries that target distinct aspects of the original information need.

- *Break-Down Reasoning:* For each sub-query generated in the previous phase, a retriever is used to retrieve relevant documents from the knowledge base. These retrieved documents are further reranked based on contextual and semantic relevance. Then, the top-ranked documents are passed to the LLM, which is prompted with the phrase "*Let's Break It Down*" to perform fine-grained reasoning over the retrieved content and synthesize a final response.

- *Iterative Query Rewriting:* If the LLM determines that a confident response cannot be produced in the current iteration, the system initiates an iterative loop. The LLM rewrites the original query based on its reasoning history, and the retrieval and reasoning steps are repeated using the rewritten query.

We note that UniRAG operates entirely through prompting and does not require any additional model training, enabling efficient and flexible deployment. In the following subsections, we describe each of these phases in detail.

3.1 Entity-Grounded Query Decomposition

Given the original query q, we identify key entities using the PLM-based FLERT model (Schweter and Akbik, 2020), a well-established Named Entity Recognition (NER) approach that uses a Conditional Random Field (CRF) layer to model tag dependencies and produce confidence scores. The resulting entity list $E = [e_1, e_2, \ldots, e_m]$ includes core elements such as named persons, locations, dates, works of art, or subject nouns, which guide query decomposition.

Once the list of core entities $E = [e_1, e_2, \ldots, e_m]$ is obtained, we leverage the reasoning capabilities of the LLM to generate a set of focused sub-queries $SQ = [sq_1, sq_2, \ldots, sq_p]$, where each sq_i is a simplified, single-hop question derived from the original query q, guided by one or more entities in E. By grounding the generation of sub-queries in the extracted entities, the resulting questions remain semantically relevant to the original query while isolating distinct aspects of the information need.

3.2 Break-Down Reasoning

3.2.1 Relevance Filtering

In this phase, we first retrieve the top-k most relevant documents from the knowledge base for both the original query q and each sub-query sq_i. Let D_q denote the top-k documents retrieved for q, and let D_{sq_i} represent the top-k documents retrieved for each sub-query sq_i. The final set of documents used for reasoning is constructed as the union of all retrieved sets $D_c = D_q \cup D_{sq_1} \cup D_{sq_2} \cup \cdots \cup D_{sq_p}$. This combined document set D_c captures both general and fine-grained evidence relevant to the original query.

Although query decomposition improves the precision of individual sub-query retrievals, the combined document set D_c may still contain irrelevant or misleading documents. To address this, we introduce an additional semantic reranking step to align the retrieved documents more closely with the original query q. Specifically, we employ the mGTE model (Zhang et al., 2024) to assign a semantic relevance score $s(d, q) \in [0, 1]$ to each document $d \in D_c$. Documents are filtered based on a predetermined relevance threshold θ, retaining only those with sufficiently high semantic alignment. Empirically, a threshold within the range of $0.8 \leq \theta \leq 0.9$ effectively balances filtering irrelevant content with maintaining critical information.

Formally, the filtered document set is given by $D_\theta = \{d \in D_c | s(d, q) \geq \theta\}$. The resulting refined document set D_θ is then used as input to the LLM in the retrieval reasoning phase, ensuring that the final answer generation is focused exclusively on highly relevant content.

3.2.2 Retrieval Reasoning

Given the filtered document set D_θ, the next step is to perform reasoning to generate the final answer to the query q. To facilitate this, we propose the use of a structured prompt based on the phrase "*Let's Break It Down*". This prompt instructs the LLM to decompose the knowledge-intensive query into clearly defined conceptual components, each representing a distinct sub-fact or constraint embedded within the original question. Then, the LLM independently verifies each component by reasoning over the information in D_θ, producing an intermediate output, $Response_{int}$. Once all components have been evaluated, the individual outcomes in $Response_{int}$ are aggregated to construct a complete answer. This break-down-based reasoning explicitly verifies each query constraint, thereby improving factual accuracy and reducing the likelihood of hallucinated or partially correct responses. Figure 2 illustrates an example of this reasoning approach.

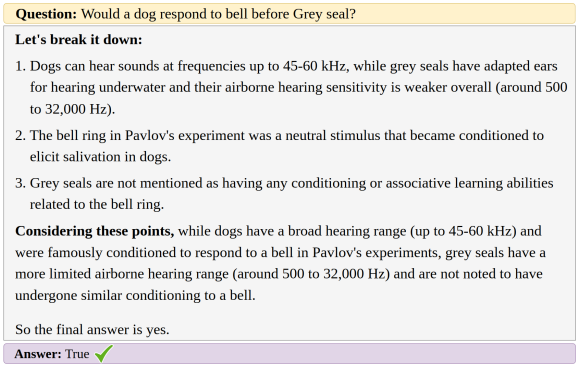

Figure 2: An intermediate response of prompting an input question (e.g. from StrategyQA) with "Let's Break It Down" to decompose and summarize the retrieved documents with reasoning in providing its final answer.

We note that the proposed prompt fundamentally differs from the standard CoT prompting paradigm (Wei et al., 2022). CoT prompting encourages the model to reason through a problem sequentially, where each inference step builds upon the previous one. While this approach is effective for procedural tasks involving arithmetic or logical deduction, it is less suited to knowledge-intensive queries, which typically involve multiple independent constraints rather than sequential dependencies. In contrast, our break-it-down prompt aligns directly with the conjunctive nature of knowledge-intensive queries by explicitly promoting parallel verification of each individual constraint. By verifying each sub-fact independently, our approach prevents error propagation and enables the model to systematically assess the accuracy and completeness of each constraint.

3.2.3 Confidence-Based Answer Decision

After completing its initial reasoning, the LLM evaluates whether it can produce a confident answer. If it determines that the filtered document set (D_θ) provides sufficient evidence, it proceeds to generate a candidate answer. Before this answer is accepted, or if the LLM indicates uncertainty due to insufficient evidence or a potentially flawed reasoning path, a metric-based validation step is applied. To support this decision, we employ RAGAs (Retrieval Augmented Generation Assessment) (ES et al., 2024), a framework for reference-free evaluation of RAG pipelines. The RAGAs framework evaluates the generated candidate answer and the retrieval process using five key criteria, such as faithfulness, answer relevancy, contextual precision, contextual recall, and contextual relevancy.

We apply RAGAs to evaluate the candidate answer, generating an overall score based on predefined metrics. If the score exceeds a predefined threshold, the answer is accepted, indicating that the retrieved documents D_θ sufficiently support a faithful and correct response. If the score falls below the threshold, the answer is rejected, and a query rewriting step is triggered to improve retrieval. This RAGAs-gated process ensures that only answers sufficiently supported by the retrieved knowledge are finalized, while insufficient responses prompt iterative refinement.

3.3 Iterative Query Rewriting

When the previous phase fails to produce a sufficiently supported answer, the query rewriting step is activated. This step leverages insights from the LLM's intermediate response $Response_{int}$. Specifically, we prompt the LLM to generate a new, refined query, denoted as q'. This new query is formulated based on the content of $Response_{int}$, incorporating the knowledge gaps identified by the LLM during the previous phase.

The prompt explicitly instructs the LLM to "*Write a different version of its original query based on the reasoning response you provided previously. Also, avoid any sub-queries that were used previously*". This approach ensures that the rewritten query is informed by the LLM's current understanding and reasoning, effectively acting as a targeted follow-up query. The previous intermediate response $Response_{int}$ thus serves as a form of short-term memory for the LLM, guiding the generation of q'. The constraint to avoid previously used sub-queries (sq_i) encourages exploration of alternative search avenues or a reformulation of the information need based on the initial retrieval outcome.

This iterative process results in a refined query, which is then sent back to the query decomposition phase to repeat the process, allowing the framework to dynamically adapt its search strategy based on the results of the preceding retrieval and reasoning steps, aiming to acquire the specific missing information required to construct a complete and accurate final answer.

4 Experimental Setup

4.1 Datasets and Retrieval Database

In our experiment, we used several benchmark datasets including multi-hop HotPotQA (Yang

Table 1: Performance of UniRAG with GPT-3.5-Turbo. IMP represents the percentage of improvements compared to baselines with respect to Exact Match on HotPotQA and 2WikiMultihopQA and Accuracy on StrategyQA.

Tasks	Multi-hop			Multi-hop			Commonsense	
Method	HotPotQA			2WikiMultihopQA			StrategyQA	
	EM	F1	IMP	EM	F1	IMP	ACC	IMP
Direct	0.400	0.4563	46.00%	0.340	0.3743	71.76%	0.606	20.46%
CoT	0.262	0.2900	122.90%	0.168	0.2056	247.62%	0.560	30.36%
ReAct	0.344	0.3937	69.77%	0.272	0.3139	114.71%	0.526	38.78%
Self-Ask	0.318	0.358	83.65%	0.252	0.2903	131.75%	0.468	55.98%
ITER-RETGEN	0.498	0.5399	17.27%	0.424	0.4643	37.74%	0.700	4.29%
BlendFilter	0.362	0.4016	61.33%	0.302	0.3279	93.38%	0.676	7.99%
UniRAG (Ours)	**0.584**	**0.6138**	-	**0.584**	**0.6090**	-	**0.730**	-

Table 2: Performance of UniRAG with LLaMA-3.1-8B as the backbone.

Tasks	Multi-hop			Multi-hop			Commonsense	
Method	HotPotQA			2WikiMultihopQA			StrategyQA	
	EM	F1	IMP	EM	F1	IMP	ACC	IMP
Direct	0.278	0.3708	175.54%	0.268	0.3292	190.30%	0.714	3.36%
CoT	0.292	0.3652	162.33%	0.262	0.3147	196.95%	0.698	5.73%
ReAct	0.308	0.3366	148.70%	0.412	0.4186	88.83%	0.562	31.32%
Self-Ask	0.318	0.4011	140.88%	0.220	0.2738	253.64%	0.702	5.13%
ITER-RETGEN	0.488	0.5175	56.97%	0.340	0.3612	128.82%	0.700	5.43%
BlendFilter	0.370	0.3967	107.03%	0.228	0.2377	241.23%	0.698	5.73%
UniRAG (Ours)	**0.766**	**0.7834**	-	**0.778**	**0.7820**	-	**0.738**	-

et al., 2018) and 2WikiMultiHopQA (Ho et al., 2020), commonsense reasoning StrategyQA (Geva et al., 2021), Biomedical MedMCQA (Pal et al., 2022) and MedQA (Jin et al., 2020), fact-verification SciFact (Wadden et al., 2020) and FEVER (Thorne et al., 2018), logical deduction LogiQA (Liu et al., 2020) and FOLIO (Han et al., 2024) and arithmetic problem-solving GSM8K and MAWPS (Kadlcík et al., 2023; Cobbe et al., 2021; Koncel-Kedziorski et al., 2016) sets. These datasets collectively provide challenging benchmarks for evaluating complex factoid and reasoning capabilities. To retrieve knowledge, we used the retriever of Generalized Text Embeddings (GTE) (Li et al., 2023b) model, and the knowledge database we used for our experiment is from the Wikipedia abstract dumps in 2017, implemented by Khattab et al. (2024) authors for multi-hop, commonsense, and fact verification, and from the PubMed abstract dumps, developed by Xiong et al. (2024) authors for biomedical datasets. We excluded an external knowledge base for logical deduction and arithmetic problem-solving datasets.

4.2 Method Comparison

By following previous state-of-the-art (SOTA) baselines from Shao et al. (2023) and Wang et al. (2024) authors, we compare and evaluate our UniRAG with the following baselines of 1) Direct Prompting (Brown et al., 2020), 2) CoT (Wei et al., 2022), 3) ReAct (Yao et al., 2023), 4) Self-Ask (Press et al., 2023), 5) ITER-RETGEN (Shao et al., 2023), and 6) BlendFilter (Wang et al., 2024) methods in retrieval settings. We also compared our UniRAG framework between the two prompting variations of the CoT (Wei et al., 2022) and our proposed "Let's Break It Down" prompt for various complex tasks to evaluate our proposed method in terms of its effectiveness in generalizability.

4.3 Model Selection

To evaluate whether our proposed method is effective in performance, we experimented with 6 SOTA LLMs including three black-box models of GPT-3.5-Turbo[1], GPT-4o[2], and Gemini-1.5-

[1]https://platform.openai.com/docs/models/gpt-3.5-turbo
[2]https://platform.openai.com/docs/models/gpt-4o

Table 3: Performance of UniRAG framework between CoT and "Let's Break It Down" with GPT-3.5-Turbo in various complex datasets.

Tasks	Biomedical		Fact-Verification		Deductive Logic		Arithmetic Problems	
Method	MedMCQA	MedQA	SciFact	FEVER	LogiQA	FOLIO	GSM8K	MAWPS
	ACC	ACC	ACC	ACC	ACC	ACC	EM	EM
UniRAG (CoT)	0.604	0.574	0.848	0.678	0.372	0.466	**0.778**	**0.974**
UniRAG (Ours)	**0.616**	**0.636**	**0.878**	**0.718**	**0.392**	**0.546**	0.774	0.954

Table 4: Performance of UniRAG framework between CoT and "Let's Break It Down" with LLaMA-3.1-8B in various complex datasets.

Tasks	Biomedical		Fact-Verification		Deductive Logic		Arithmetic Problems	
Method	MedMCQA	MedQA	SciFact	FEVER	LogiQA	FOLIO	GSM8K	MAWPS
	ACC	ACC	ACC	ACC	ACC	ACC	EM	EM
UniRAG (CoT)	0.624	0.628	0.640	0.470	0.432	**0.600**	0.844	0.848
UniRAG (Ours)	**0.668**	**0.720**	**0.656**	**0.562**	**0.472**	0.578	**0.854**	**0.888**

Flash[3] and three white-box models of LLaMA-3.1-8B (Grattafiori et al., 2024), Qwen-2.5-7B (Yang et al., 2024) and Gemma-2-9B (Team et al., 2024) models, which are all instruction-tuned models.

4.4 Evaluation Metrics

Based on the authors of Shao et al. (2023) and Wang et al. (2024), we test the first 500 questions for each dataset, in which we experiment from the development datasets for multi-hop, biomedical, and fact-verification sets, the training dataset for commonsense reasoning, and test datasets for logical deduction and arithmetic sets accordingly. Following Yao et al. (2023), Shao et al. (2023) and Wang et al. (2024) authors, we evaluate by the exact match (EM) and F1 scores for the multi-hop and arithmetic problem-solving, and the accuracy (ACC) for the commonsense reasoning, biomedical, fact-verification and logical deduction datasets.

5 Results

5.1 Performance Comparison

We experimented and compared the previous SOTA baselines, detailed in Tables 1 and Table 2 (also see Appendix section Tables 10-13), which demonstrate the consistent superiority of our proposed UniRAG method across a diverse range of LLMs and datasets. UniRAG achieved the highest scores in both EM and F1-score on HotPotQA, 2WikiMultihopQA, and accuracy on StrategyQA for five out of the six tested LLMs, including LLaMA-3.1-8B, Qwen-2.5-7B, Gemma2-9B, GPT-3.5-Turbo, and Gemini-1.5-Flash, when compared against all previous SOTA baselines. For instance, with the LLaMA-3.1-8B model, UniRAG achieved an F1 score of 0.7834 on HotPotQA and 0.7820 on 2WikiMultihopQA, and an accuracy of 0.738 on StrategyQA. While ITER-RETGEN showed slightly better performance than UniRAG on the HotPotQA dataset specifically with the GPT-4o model, UniRAG maintained its lead on 2WikiMultihopQA and StrategyQA even with this powerful black-box model, underscoring the general robustness and significant performance gains from our method.

5.2 Generalizability on Various Tasks

Also, in terms of proving our proposed method is indeed effective in generalizability on various complex tasks, given the same UniRAG framework, the results show that compared to the standard CoT prompt, our "Let's Break It Down" prompt has a higher accuracy score for biomedical (e.g. MedMCQA and MedQA) and fact-verification (e.g. SciFact and FEVER) datasets. In Tables 3 and 4, there are noticeable gaps of improvements for FEVER and MedQA, with an improvement of 5.57% and 9.75% for GPT-3.5-Turbo and 16.01% and 12.78% for LLaMA-3.1-8B models respectively. In addition, there were marginal improvements in other datasets including MedMCQA, SciFact, and LogiQA. Although there was a small difference, our empirical results show that using the CoT prompting is good on logical deductive and arithmetic problem-solving datasets (e.g. FOLIO, GSM8K and MAWPS). Based on our findings, the

[3]https://ai.google.dev/gemini-api/docs/models#gemini-1.5-flash

Table 5: Comparison of module-wise performance in benchmark datasets with GPT-3.5 model.

Method	HotPotQA		2WikiMultihopQA		StrategyQA
	EM	F1	EM	F1	ACC
Let's Break It Down	0.440	0.5063	0.410	0.4452	0.724
w/ Decomp	0.478	0.5354	0.422	0.4608	0.702
w/ Decomp+Rewrite	0.492	0.5459	0.436	0.4825	0.706
w/ Decomp+Rewrite+Rerank	**0.584**	**0.6138**	**0.584**	**0.6090**	**0.730**

Table 6: Comparison of module-wise performance in benchmark datasets with LLaMA-3.1 model.

Method	HotPotQA		2WikiMultihopQA		StrategyQA
	EM	F1	EM	F1	ACC
Let's Break It Down	0.346	0.385	0.270	0.2827	0.728
w/ Decomp	0.454	0.501	0.482	0.5145	0.680
w/ Decomp+Rewrite	0.548	0.5854	0.534	0.5618	0.734
w/ Decomp+Rewrite+Rerank	**0.766**	**0.7834**	**0.778**	**0.7820**	**0.738**

common feature of biomedical and fact-verification datasets lies on the need for factual and domain-specific expertise (e.g. specialized-medical terms and scientific evidences) to guide the LLM to think rationally before concluding with a definite answer to the original query. Unlike the standard CoT, which takes on procedural thinking, our observation shows that our "break down" reasoning is more effective when the complex question requires complementary and interrelated thinking processes.

5.3 Module-Wise Experiment

We conducted a comprehensive ablation study, with the results presented in Table 5 and 6 for the GPT-3.5-Turbo and LLaMA-3.1-8B models, respectively. Our analysis begins with a baseline using only the reasoning module ("Let's Break It Down"), which yields modest performance. The introduction of the "Decomposition" module results in a consistent and significant performance increase across all datasets for both models. Subsequently, integrating the "Rewriting" module further elevates the performance, which indicates that refining the generated reasoning steps is beneficial for accuracy. Most notably, the full framework, which combines decomposition, reasoning, rewriting, and reranking, demonstrates a substantial leap in performance that far surpasses any of the partial configurations. This synergistic effect is particularly pronounced in the LLaMA-3.1-8B model, where the EM score on HotpotQA, for instance, provided substantial increase from 0.346 with only reasoning to 0.766 with all modules enabled.

This large performance delta strongly validates our hypothesis that each component is integral to the framework's success, working in concert to effectively tackle complex question-answering tasks.

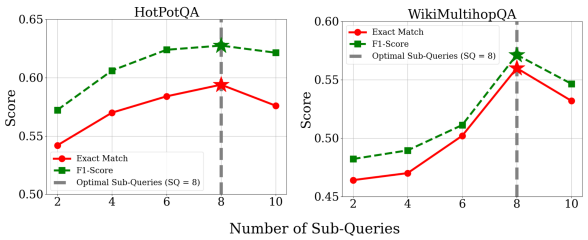

Figure 3: Experimentation on observing the optimal number of sub-queries from HotPotQA and 2WikiMultihopQA datasets with LLaMA-3.1-8B model.

Other LLMs including GPT-4o, Gemini-1.5-Flash, Qwen-2.5-7B and Gemma2-9B have shown similar substantial performance results (see Appendix Tables 14-17).

5.4 Optimal Number of Sub-Queries

To determine the optimal number of sub-queries for our query decomposition framework, we performed using the LLaMA-3.1-8B model, with the results for the HotPotQA and 2WikiMultihopQA datasets presented in Figure 3. These graphs illustrate the relationship between the number of generated sub-queries (ranging from 2 to 10) and the resulting performance in terms of EM and F1-score. For both datasets, a consistent trend was observed: performance metrics generally improved as the number of sub-queries increased from two, reaching a distinct peak before declining with further increases. Specifically, optimal performance was achieved with 8 sub-queries for both the HotPotQA (e.g. 0.595 for EM and 0.63 for F1) and 2WikiMultihopQA (e.g. 0.565 for EM and 0.57 for F1) datasets. Increasing the sub-query count to 10 led to a discernible drop in performance across both metrics and datasets, indicating that 8 sub-queries represent the most effective balance for these tasks with the LLaMA model.

Furthermore, our study reveals that different task domains such as fact verification, biomedical and arithmetic problems tasks, naturally benefit from different levels of decomposition, but a near-optimal performance can often be achieved without extensive, fine-grained tuning. For instance, in domains requiring direct fact-verification, such as SciFact and FEVER, performance is optimal with just 2 sub-queries. This is intuitive, as these tasks often involve verifying a single proposition rather than solving a multi-step problem. Increasing the number of sub-queries for these tasks shows a slight, graceful decline in performance, indicating that

Table 7: Optimal sub-query analysis on benchmark datasets using LLaMA-3.1-8B. EM represents the Exact Match and ACC represents the Accuracy.

	HotpotQA	2WikiMultihopQA	StrategyQA	GSM8K	MAWPS	MedMCQA	MedQA	LogiQA	FOLIO	SciFact	FEVER
Sub-Queries	EM	EM	ACC	EM	EM	ACC	ACC	ACC	ACC	ACC	ACC
2	0.542	0.464	0.782	0.868	**0.914**	0.672	0.676	0.546	**0.682**	**0.932**	**0.782**
4	0.570	0.470	0.780	0.850	0.906	0.664	0.696	0.566	0.672	0.922	0.778
6	0.584	0.502	0.794	**0.882**	0.896	0.668	0.672	0.548	0.674	0.912	0.776
8	**0.594**	**0.560**	**0.800**	0.866	0.906	**0.686**	**0.708**	**0.570**	0.660	0.898	0.766
10	0.576	0.532	0.788	0.858	0.904	0.668	0.676	0.556	0.610	0.910	0.776

Table 8: Iteration analysis across benchmark datasets using LLaMA-3.1-8B. EM represents the Exact Match, ACC represents the Accuracy, CC represents the Correct Count, and IMP represents the percentage of Improvements (%).

	HotpotQA			2WikiMultihopQA			StrategyQA			GSM8K			MAWPS			MedMCQA		
Max Iter.	EM	CC	IMP	EM	CC	IMP	ACC	CC	IMP	EM	CC	IMP	EM	CC	IMP	ACC	CC	IMP
1	0.766	383	2.61	0.776	388	2.06	0.840	420	4.52	0.854	429	3.50	0.888	444	2.93	0.668	334	6.59
2	0.770	385	2.08	0.774	387	2.33	0.858	429	2.33	0.866	433	2.54	0.910	455	0.44	0.680	340	4.71
3	0.764	382	2.80	0.782	391	1.28	0.872	436	0.69	0.870	435	2.07	0.906	453	0.88	0.700	350	1.71
4	**0.786**	**393**	-	0.790	395	0.25	0.856	428	2.57	0.888	443	0.23	**0.914**	**457**	-	**0.716**	**356**	-
5	0.770	385	2.08	**0.792**	**396**	-	**0.877**	**439**	-	**0.890**	**444**	-	0.904	452	1.11	0.702	351	1.42

Table 9: Iteration analysis across benchmark datasets using LLaMA-3.1-8B.

	MedQA			LogiQA			FOLIO			SciFact			FEVER		
Max Iter.	ACC	CC	IMP	ACC	CC	IMP	ACC	CC	IMP	ACC	CC	IMP	ACC	CC	IMP
1	0.732	366	2.20	0.472	239	7.53	0.560	280	9.64	0.656	329	2.74	0.670	335	8.36
2	0.782	390	3.68	0.492	246	4.47	0.600	300	2.33	0.666	333	1.50	0.698	349	4.01
3	**0.804**	**402**	-	0.500	250	2.80	**0.614**	**307**	-	**0.676**	**338**	-	0.698	349	4.01
4	0.796	398	2.82	0.508	254	1.18	0.608	304	0.99	0.642	321	5.30	0.712	356	1.97
5	0.784	392	2.73	**0.514**	**257**	-	0.608	304	0.99	0.644	322	4.97	**0.726**	**363**	-

a simple, low-overhead default works best. Conversely, for more complex reasoning tasks, such as those in the biomedical domain (MedMCQA and MedQA), performance peaks at 8 sub-queries. This aligns with our findings for multi-hop datasets and suggests that tasks requiring deeper, multi-faceted reasoning benefit from more extensive decomposition. For the GSM8K arithmetic problem-solving dataset, the peak EM is at 6 sub-queries (0.882), but using 8 sub-queries still yields a strong score (0.866). This indicates that a single, well-chosen default (e.g., 6 or 8) can serve as a robust baseline across a variety of complex domains, minimizing the need for exhaustive tuning for every new application. Table 7 provides a comprehensive analysis on observing the optimal number of sub-query parameter from LLaMA-3.1-8B model in various benchmark datasets.

5.5 Iteration Analysis

Acknowledging concerns about our pipeline's iterative nature, we argue that UniRAG achieves a near-optimal balance of accuracy and efficiency within just a few steps, making it practical for real-world deployment. To substantiate this, we conducted an ablation study, presented in Table 8 and 9, analyzing performance improvement across multiple iterations on various benchmark datasets. Our findings reveal a clear pattern of diminishing returns, where the most significant accuracy gains occur in the initial steps.

6 Conclusion

The UniRAG framework enhances LLM efficacy for knowledge-intensive questions by integrating query decomposition and rewriting with our proposed reasoning strategy. Our framework breaks down the complex information needs while improving query searchability in RAG systems. UniRAG further utilizes a reranker and a "Let's Break It Down" prompting strategy to ensure logical synthesis and reduce hallucinations. Demonstrating superior performance over the previous SOTA baselines, UniRAG shows robust generalization across diverse LLMs like LLaMA-3.1-8B, GPT-3.5-Turbo, and Gemini-1.5-Flash.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF), funded by the Korean

government under Grant No. RS-2023-00273751 and the Institute for Information & Communications Technology Planning & Evaluation (IITP), funded by the Korea government under Grant No. RS-2024-00397085.

Limitations

Our UniRAG framework uses confidence score decision-making with external RAGAs assessments that could potentially introduce biases or unreliable decisions due to threshold sensitivity. Fortunately, our empirical observations in this study indicated that the model's performance was not significantly sensitive to this threshold. Consequently, we set it to a fixed value that provided the best performance in this paper.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Dheeru Dua, Shivanshu Gupta, Sameer Singh, and Matt Gardner. 2022. Successive prompting for decomposing complex questions. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 1251–1265. Association for Computational Linguistics.

Shahul ES, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - System Demonstrations, St. Julians, Malta, March 17-22, 2024*, pages 150–158. Association for Computational Linguistics.

Ruiliu Fu, Han Wang, Xuejun Zhang, Jun Zhou, and Yonghong Yan. 2021. Decomposing complex questions makes multi-hop QA easier and more interpretable. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 169–180. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Trans. Assoc. Comput. Linguistics*, 9:346–361.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Xiao-Yu Guo, Yuan-Fang Li, and Gholamreza Haffari. 2022. Complex reading comprehension through question decomposition. *arXiv preprint arXiv:2211.03277*.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, and 16 others. 2024. FOLIO: natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 22017–22031. 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, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 874–880. Association for Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24:251:1–251:43.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*.

Marek Kadlcík, Michal Stefánik, Ondrej Sotolár, and Vlastimil Martinek. 2023. Calc-x and calcformers: Empowering arithmetical chain-of-thought through interaction with symbolic systems. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12101–12108. Association for Computational Linguistics.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. Dspy: Compiling declarative language model calls into self-improving pipelines.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A math word problem repository. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1152–1157. The Association for Computational Linguistics.

Xiaopeng Li, Lixin Su, Pengyue Jia, Xiangyu Zhao, Suqi Cheng, Junfeng Wang, and Dawei Yin. 2023a. Agent4ranking: Semantic robust ranking via personalized query rewriting using multi-agent llm. *arXiv preprint arXiv:2312.15450*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023b. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3622–3628. ijcai.org.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023a. 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.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023b. 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.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6097–6109. Association for Computational Linguistics.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on Health, Inference, and Learning, CHIL 2022, 7-8 April 2022, Virtual Event*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8864–8880. Association for Computational Linguistics.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5687–5711. Association for Computational Linguistics.

Stefan Schweter and Alan Akbik. 2020. FLERT: Document-level features for named entity recognition. *Preprint*, arXiv:2011.06993.

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, Singapore, December 6-10, 2023*, pages 9248–9274. Association for Computational Linguistics.

Zhihong Shao and Minlie Huang. 2022. Answering open-domain multi-answer questions via a recall-then-verify framework. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1825–1838. Association for Computational Linguistics.

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

Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024, pages 8371–8384. Association for Computational Linguistics.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 809–819. Association for Computational Linguistics.

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), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10014–10037. Association for Computational Linguistics.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7534–7550. Association for Computational Linguistics.

Haoyu Wang, Ruirui Li, Haoming Jiang, Jinjin Tian, Zhengyang Wang, Chen Luo, Xianfeng Tang, Monica Xiao Cheng, Tuo Zhao, and Jing Gao. 2024. Blendfilter: Advancing retrieval-augmented large language models via query generation blending and knowledge filtering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 1009–1025. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. 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 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 6233–6251. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6442–6454. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Fanghua Ye, Meng Fang, Shenghui Li, and Emine Yilmaz. 2023. Enhancing conversational search: Large language model-aided informative query rewriting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5985–6006, Singapore. Association for Computational Linguistics.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: EMNLP 2024 - Industry Track, Miami, Florida, USA, November 12-16, 2024*, pages 1393–1412. Association for Computational Linguistics.

Ben Zhou, Kyle Richardson, Xiaodong Yu, and Dan Roth. 2022. Learning to decompose: Hypothetical question decomposition based on comparable texts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 2223–2235. Association for Computational Linguistics.

Table 10: Performance of UniRAG with GPT-4o. IMP represents the percentage of improvements compared to baselines with respect to Exact Match on HotPotQA and 2WikiMultihopQA and Accuracy on StrategyQA.

Tasks	Multi-hop			Multi-hop			Commonsense	
Method	HotPotQA			2WikiMultihopQA			StrategyQA	
	EM	F1	IMP	EM	F1	IMP	ACC	IMP
Direct	0.464	0.5195	18.97%	0.522	0.5460	33.72%	0.662	24.77%
CoT	0.452	0.4989	22.12%	0.462	0.4853	51.08%	0.762	8.40%
ReAct	0.418	0.4635	32.06%	0.382	0.4175	82.72%	0.768	7.55%
Self-Ask	0.510	0.5600	8.24%	0.632	0.6555	10.44%	0.736	12.23%
ITER-RETGEN	**0.592**	**0.6484**	-6.76%	0.620	0.6445	12.58%	0.810	1.98%
BlendFilter	0.480	0.5366	15.00%	0.334	0.3542	108.98%	0.806	2.48%
UniRAG (Ours)	0.552	0.6116	-	**0.698**	**0.7196**	-	**0.826**	-

Table 11: Performance of UniRAG with Gemini-1.5-Flash as the backbone.

Tasks	Multi-hop			Multi-hop			Commonsense	
Method	HotPotQA			2WikiMultihopQA			StrategyQA	
	EM	F1	IMP	EM	F1	IMP	ACC	IMP
Direct	0.344	0.3900	63.95%	0.320	0.3586	76.25%	0.678	14.75%
CoT	0.374	0.4220	50.80%	0.370	0.4148	52.43%	0.648	20.06%
ReAct	0.312	0.3619	80.77%	0.320	0.3615	76.26%	0.688	13.08%
Self-Ask	0.428	0.4757	31.78%	0.448	0.4808	25.89%	0.706	10.20%
ITER-RETGEN	0.548	0.5912	2.92%	0.472	0.4975	19.49%	0.760	2.37%
BlendFilter	0.316	0.3938	78.48%	0.206	0.2413	173.79%	0.754	3.18%
UniRAG (Ours)	**0.564**	**0.6161**	-	**0.564**	**0.5932**	-	**0.778**	-

Table 12: Performance of UniRAG with Qwen-2.5-7B as the backbone.

Tasks	Multi-hop			Multi-hop			Commonsense	
Method	HotPotQA			2WikiMultihopQA			StrategyQA	
	EM	F1	IMP	EM	F1	IMP	ACC	IMP
Direct	0.304	0.3764	64.47%	0.312	0.3668	72.44%	0.698	5.73%
CoT	0.254	0.3271	96.85%	0.268	0.3036	100.75%	0.698	5.73%
ReAct	0.166	0.1932	201.20%	0.248	0.2723	116.94%	0.588	25.51%
Self-Ask	0.390	0.4368	28.21%	0.412	0.4454	30.58%	0.662	11.48%
ITER-RETGEN	0.384	0.4439	30.21%	0.260	0.3114	106.92%	0.728	1.37%
BlendFilter	0.338	0.3771	47.93%	0.210	0.2178	156.19%	0.678	8.85%
UniRAG (Ours)	**0.500**	**0.5465**	-	**0.538**	**0.5606**	-	**0.738**	-

Table 13: Performance of UniRAG with Gemma-2-9B as the backbone.

Tasks	Multi-hop			Multi-hop			Commonsense	
Method	HotPotQA			2WikiMultihopQA			StrategyQA	
	EM	F1	IMP	EM	F1	IMP	ACC	IMP
Direct	0.356	0.4404	92.13%	0.258	0.3087	131.78%	0.650	14.15%
CoT	0.336	0.4175	103.57%	0.220	0.2612	171.82%	0.658	12.77%
ReAct	0.156	0.1823	338.46%	0.130	0.1382	360.00%	0.574	29.27%
Self-Ask	0.392	0.4771	74.49%	0.318	0.3664	88.05%	0.680	9.12%
ITER-RETGEN	0.362	0.4544	88.95%	0.296	0.348	102.03%	0.640	15.94%
BlendFilter	0.302	0.3864	126.49%	0.242	0.2811	147.11%	0.704	5.40%
UniRAG (Ours)	**0.684**	**0.7323**	-	**0.598**	**0.6135**	-	**0.742**	-

A Entity Extraction Comparison

To improve query decomposition, often hindered by inconsistent entity extraction from LLMs, we leverage a specialized PLM to identify crucial entities within complex questions. Our ablation study compared the NER capabilities of RoBERTa (Liu et al., 2019), LUKE (Yamada et al., 2020), and FLERT (Schweter and Akbik, 2020) to select the optimal PLM for generating high-quality entity inputs essential for forming effective sub-queries. We evaluated these models based on Semantic Textual Similarity (STS) cosine similarity scores (detailed in Appendix Figure 4). While both RoBERTa and LUKE are proficient in entity extraction, our experiments highlighted a distinct advantage of the FLERT model to extensively extract and distinguish a more granular array of specific entities. Unlike RoBERTa and LUKE models, FLERT demonstrated a superior ability to identify detailed entity types beyond broader categories, such as recognizing the title of a film or book, geographical locations of places, identifying the types of various products, being accessible to adopt multilingual terms, and dates of events with greater specificity. This nuanced entity recognition, not observed to the same detailed extent in RoBERTa and LUKE during our experiments, proved crucial. Consequently, the STS benchmark results consistently favored FLERT, which achieved the highest STS score in 10 out of the 18 evaluated configurations. This robust ability to capture more specific and vital entities makes FLERT our chosen PLM, as it is expected to provide the main LLM with more precise entity sets, thereby enhancing sub-query generation and overall performance.

B Prompt Templates

In this section, we provide our representative prompting templates that we used in testing on various benchmark datasets including multi-hop HotPotQA and 2WikiMultihopQA, and commonsense reasoning StrategyQA for brevity. Figure 5 and 6 provide the visualization of examples of our prompt template. These specific prompts include instructions for query decomposition, retrieval reasoning, and query rewriting. Our structural prompting is the same throughout our experiment, varied by few-shot examples for each dataset.

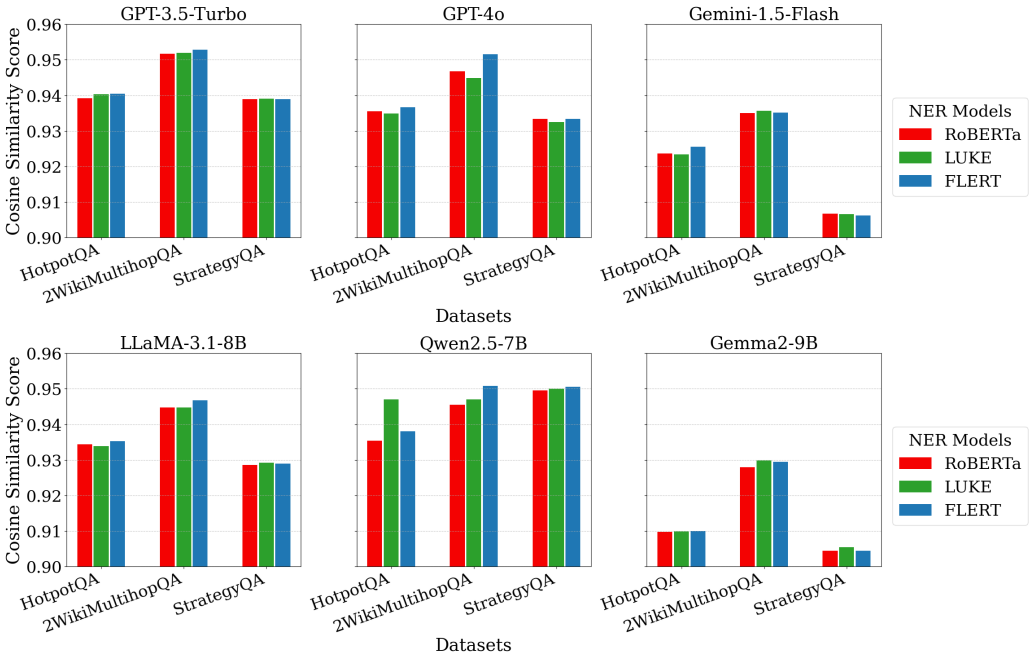

Figure 4: Semantic-Textual-Similarity(STS) performance comparisons from black-box and white-box models.

Table 14: Comparison of module-wise performance in benchmark datasets with GPT-4o model.

Method	HotPotQA		2WikiMultihopQA		StrategyQA
	EM	F1	EM	F1	ACC
Let's Break It Down	0.520	0.5789	0.580	0.6134	**0.840**
w/ Decomp	0.530	0.5896	0.592	0.6083	0.832
w/ Decomp+Rewrite	0.570	0.6339	0.622	0.6399	0.812
w/ Decomp+Rewrite+Rerank	**0.552**	**0.6116**	**0.698**	**0.7196**	0.826

Table 15: Comparison of module-wise performance in benchmark datasets with Gemini-1.5-Flash model.

Method	HotPotQA		2WikiMultihopQA		StrategyQA
	EM	F1	EM	F1	ACC
Let's Break It Down	0.304	0.3773	0.332	0.4024	0.758
w/ Decomp	0.460	0.5121	0.402	0.4303	0.762
w/ Decomp+Rewrite	0.502	0.5447	0.474	0.5045	0.754
w/ Decomp+Rewrite+Rerank	**0.564**	**0.6161**	**0.564**	**0.5932**	**0.778**

Table 16: Comparison of module-wise performance in benchmark datasets with Qwen-2.5-7B model.

Method	HotPotQA		2WikiMultihopQA		StrategyQA
	EM	F1	EM	F1	ACC
Let's Break It Down	0.262	0.3081	0.248	0.3014	0.714
w/ Decomp	0.412	0.4605	0.370	0.4000	0.716
w/ Decomp+Rewrite	0.490	0.5369	0.518	0.5492	0.712
w/ Decomp+Rewrite+Rerank	**0.500**	**0.5465**	**0.538**	**0.5606**	**0.738**

Table 17: Comparison of module-wise performance in benchmark datasets with Gemma2-9B model.

Method	HotPotQA		2WikiMultihopQA		StrategyQA
	EM	F1	EM	F1	ACC
Let's Break It Down	0.328	0.3700	0.324	0.3620	0.726
w/ Decomp	0.392	0.4330	0.398	0.4266	0.606
w/ Decomp+Rewrite	0.514	0.5594	0.510	0.5365	0.716
w/ Decomp+Rewrite+Rerank	**0.684**	**0.7327**	**0.598**	**0.6135**	**0.742**

Prompt for Query Decomposition

You are a helpful assistant that creates multiple, unique, and concise search queries related to a main question.
The goal is to break down the input question into key searchable queries based on the provided keywords, making each query focused and easy to search.
Always include the main subject in each search query.
Do NOT include assumptions or irrelevant information.
Ensure each search query covers a different aspect of the main question—no duplicates.
Using **{list_of_keywords}**, generate a numbered list of simple and precise search queries for: **{original_query}**
Output ONLY the numbered list of search queries without any explanations or extra information.

Prompt for Retrieval Reasoning on HotpotQA and 2WikiMultihopQA

The following are few examples of answering the questions. Answer the following question with the given format.

Question: What is the name of this American musician, singer, actor, comedian, and songwriter, who worked with Modern Records and born in December 5, 1932?
Let's break it down:

1. Artists who worked with Modern Records include Etta James, Joe Houston, Little Richard, Ike and Tina Turner and John Lee Hooker in the 1950s and 1960s.
2. Of these Little Richard, born in December 5, 1932, was an American musician, singer, actor, comedian, and songwriter.

Considering these points, Little Richard worked with Modern Records, meets the description of an American musician, singer, actor, comedian, and songwriter, and was born on December 5, 1932.

So the final answer is Little Richard

By following the above examples, answer the question below with the given context.

Context:
{Knowledge}
Question: {input}
Let's break it down:

_____

Considering these points,

_____

So the final answer is

Figure 5: Our proposed prompt for query decomposition and retrieval reasoning for HotPotQA and 2WikiMultihopQA datasets.

Prompt for Retrieval Reasoning on StrategyQA

The following are few examples of answering the questions. Answer the following question with the given format.

Question: Can you get Raclette in YMCA headquarters city?
Let's break it down:

1. YMCA headquarters is located in Paris, France.
2. Raclette is a dish native to parts of Switzerland, but it is also popular in France.
3. So it is likely that Raclette can be found in Paris.

Considering these points, the YMCA headquarters is located in Paris, France, and Raclette is a dish popular in France.

So the final answer is yes

By following the above examples, answer the question below with the given context.

Context:
{Knowledge}
Question: {input}
Let's break it down:

_____

Considering these points,

_____

So the final answer is

Prompt for Query Rewriting

Your task is to analyze the previous reasoning response (**{reasoning}**) and identify information or context that was missing or not explicitly provided.
Then, generate a new query of the original question that specifically targets this missing or unaddressed information.
AVOID including these sub-queries: **{sub_queries}**.
Your goal is to help the user probe deeper into aspects left unanswered or unclear in the previous context.
Output ONLY the rewritten question.
Original question: **{input}**

Figure 6: Our proposed prompt for retrieval reasoning for StrategyQA dataset and prompt for query rewriting.