

Topology-of-Question-Decomposition: Enhancing Large Language Models with Information Retrieval for Knowledge-Intensive Tasks

Weijie Li¹, Jin Wang^{1,*}, Liang-Chih Yu^{2,*} and Xuejie Zhang¹

¹School of Information Science and Engineering, Yunnan University, Yunnan, P.R. China

²Department of Information Management, Yuan Ze University, Taiwan

liweijie1@stu.ynu.edu.cn, {wangjin, xjzhang}@ynu.edu.cn

lcyu@saturn.yzu.edu.tw

Abstract

Large language models (LLMs) are increasingly deployed for general problem-solving across various domains yet remain constrained to chaining immediate reasoning steps and depending solely on parametric knowledge. Integrating an information retrieval system directly into the reasoning process of LLMs can improve answer accuracy but might disrupt the natural reasoning sequence. Consequently, LLMs may underperform in complex, knowledge-intensive tasks requiring multiple reasoning steps, extensive real-world knowledge, or critical initial decisions. To overcome these challenges, we introduce a novel framework, **Topology-of-Question-Decomposition (ToQD)**, which activates retrieval only when necessary. Globally, ToQD guides LLMs in constructing a topology graph from the input question, each node representing a sub-question. Locally, ToQD employs *self-verify* inference to determine whether a sub-question should retrieve relevant documents, necessitate further decomposition, or directly provide an answer. Experiments demonstrate that ToQD achieves superior performance and robustness in complex, knowledge-intensive tasks, significantly enhancing system response efficiency. The code repository is accessible at <https://github.com/DCVDB/ToQD>

1 Introduction

Despite the increased model size enabling large language models (LLMs) to excel in general knowledge domains (Hendrycks et al., 2021; Kwiatkowski et al., 2019), LLMs (OpenAI, 2020) continue to struggle with factual errors in the complex knowledge-intensive tasks (Petroni et al., 2021; Zelikman et al., 2022) that require multi-reasoning. To address the inherent knowledge constraints of LLMs (Zhang et al., 2023; Mallen et al., 2023), retrieval-augmented generation (RAG)

(Lewis et al., 2021; Gao et al., 2024) augment the input of LLMs with the relevant documents, thus reducing factual hallucination (Yu et al., 2023; Yoran et al., 2024) and bolstering performance in intricate tasks (Yang et al., 2018; Joshi et al., 2017). Additionally, advanced prompting strategies such as chain-of-thought (CoT) (Wei et al., 2023) enhance the reasoning capabilities of models by encouraging them to generate additional reasoning steps (Mavi et al., 2024). In light of these enhancements, prompt strategies incorporated within the RAG framework can be classified into two primary categories: *local reasoning* and *global planning*.

Local reasoning methods, primarily based on CoT approaches such as (Auto-CoT (Zhang et al., 2022), Self-Consistency (Wang et al., 2023c), Recite-and-answer (Sun et al., 2023) and Tree-of-Thought (Yao et al., 2023a)), facilitate sequential reasoning steps to enhance response accuracy incrementally. However, these methods often generate reasoning steps without logical connection rather than a cohesive global planning strategy, frequently resulting in a disorganized reasoning process with significant redundancy (Figure 1). Moreover, integrating the RAG framework, as demonstrated by Self-Ask (Press et al., 2023b) and DSP (Khatab et al., 2023), disrupts sequential reasoning, limiting LLMs to localized sub-questions and curtailing their broader reasoning abilities (Jain et al., 2024). Consequently, in primarily *local reasoning* methods, interaction with IR typically involves retrieving all relevant documents upfront rather than as needed, contradicting the view that ‘LLMs are knowledge warehouses.’ (Yin et al., 2023).

Compared to *local reasoning*, *global planning* methods like Plan-and-Solve (Wang et al., 2023a), Least-to-Most (Zhou et al., 2023), and SearChain (Xu et al., 2024) significantly enhance response accuracy by pre-planning a chain of reasoning to clarify logical relationships (Figure 1). However, the inherent rigidity of these methods limits their

*Corresponding author

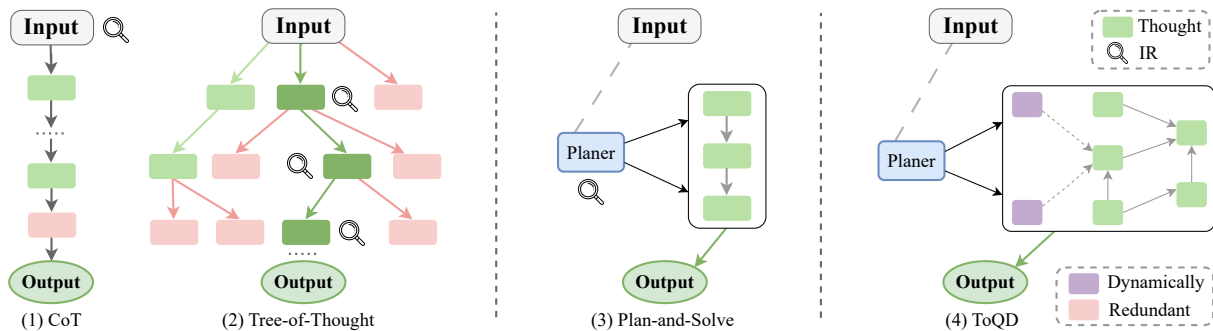


Figure 1: Schematic illustrating various approaches to problem-solving with LLMs. Each **green box** represents an intermediate *thought* aimed at resolving a problem, while the **blue box** symbolizes the global planner. The **red box** denotes a redundant *thought*, and the **purple box** indicates a dynamically adjusted *thought*. **Local reasoning**: (1) CoT (2) Tree-of-Thought. **Global planning**: (3) Plan-and-Solve. Our method: (4) ToQD.

capacity to dynamically respond to complex reasoning steps, thus impeding deeper analytical processing. Simultaneously, it should be noted that complex knowledge-intensive tasks do not invariably require sequential reasoning; parallel reasoning can accelerate the process, a facet frequently neglected by chain reasoning methodologies.

In addressing the limitations of *local reasoning* and *global planning* methods, this study proposes a novel approach termed **Topology-of-Question-Decomposition (ToQD)**, which enables retrieval only when necessary (Figure 1 and 2). **Globally**, ToQD instructs LLMs to construct a topology graph \mathcal{G}_{top} based on the input question, where each node encapsulates a specific subquestion. Within \mathcal{G}_{top} should the nodes align either parallelly or sequentially, LLMs are programmed to adjust their responses to expedite the resolution process. **Locally**, at each subquestion node within \mathcal{G}_{top} , LLMs apply *self-verify* inference to ascertain whether to retrieve relevant documents, necessitate further decomposition, or directly provide an answer. Following interactions with IR and subsequent reasoning, ToQD executes summarization at each node to construct the final answer for the input question.

Experiments demonstrate that ToQD surpasses state-of-the-art baselines in complex knowledge-intensive tasks while enhancing overall efficiency. Moreover, the ToQD prompt templates deliberately omit numerical examples to reduce dependency on the in-context learning capabilities of LLMs, thereby enhancing its applicability to models ranging from 2B to over 100B parameters.

2 Related Work

Chain-of-Thought Prompting. Chain-of-thought (CoT) prompting (Wei et al., 2023; Suzgun et al.,

2022), a gradient-free method, systematically facilitates the generation of intermediate reasoning steps by LLMs before delivering the final answer, with multiple task-specific variants (SelfAsk (Press et al., 2023b), Ask-me-anything (Arora et al., 2022), and ReSP (Jiang et al., 2024)). The fundamental principle underlying CoT prompting is its systematic decomposition of complex problems into a series of intermediate reasoning steps (Servantez et al., 2024; Kojima et al., 2023). However, CoT predominantly rely on the model’s in-context learning capabilities (Shi et al., 2024; Chung et al., 2022), limiting their applicability to smaller and medium-sized models (Raffel et al., 2023; Jiang et al., 2023). Moreover, methods (Yao et al., 2023a; Besta et al., 2024) employ LLMs to decompose complex questions and sequentially answer each sub-question iteratively.

Retrieval-Augmented LLMs. Recent studies show that retrieval-augmented generation (RAG) pipeline can enhance the reasoning ability of LLMs (Trivedi et al., 2023; He et al., 2022; Shao et al., 2023), make the responses more credible and traceable (Xu et al., 2024), reduce the factual hallucinations. However, the RAG still struggles with sourcing and assimilating factual evidence from multiple documents for complex multi-hop queries, often leading to factual errors that mislead LLMs (Tang and Yang, 2024; Mallen et al., 2023).

Self-Knowledge in LLMs. The concept of ‘*self-knowledge*’ in LLMs, initially introduced by (Kadavath et al., 2022), is further defined as the LLMs’ ability to understand limitations concerning the unknowns and has been evaluated by (Yin et al., 2023). Simultaneously, recent scholarly focus on the quality of training data (Gunasekar et al., 2023; Touvron et al., 2023) indicates that inaccuracies in

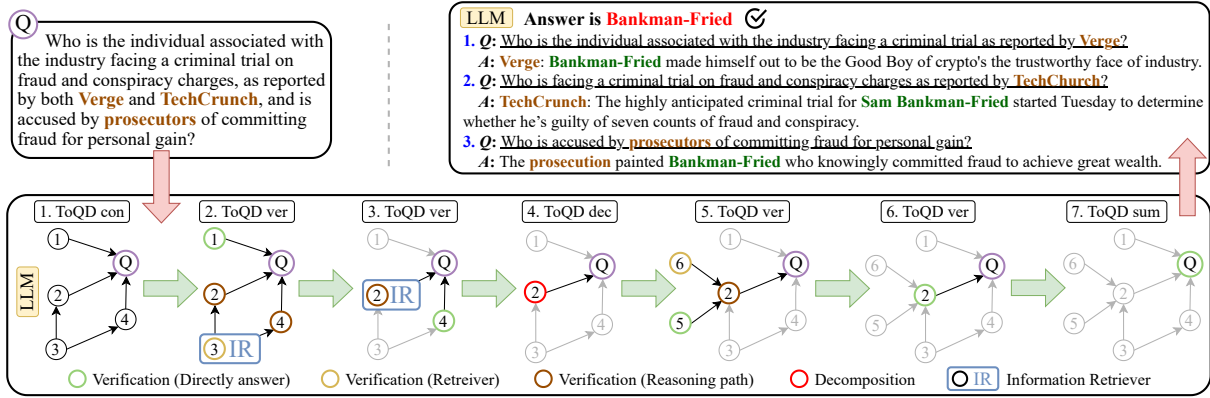


Figure 2: The Topology-of-Question-Decomposition (ToQD) process involves several key steps. **Globally**, ToQD guides the LLM to construct a topology graph \mathcal{G}_{top} from the question Q , where each node symbolizes a subquestion q . **Locally**, at each node $q \in \mathcal{G}_{top}$, the LLMs employ *self-verify* inference to decide if q should retrieve documents, requires further decomposition, or can be answered outright. In the final stage, each node within \mathcal{G}_{top} contributes to answering the question Q .

Algorithm 1 ToQD Inference

Require: Generator LM \mathcal{M} , *Self-Verify* inference

- 1: **Input:** Input the original question Q , **Output:** the answer \mathcal{A} of the input question Q
- 2: \mathcal{M} decomposes Q into the sub-questions set Q' and evaluate **IsRel** for $q \in Q'$ ▷ **Critique**
- 3: \mathcal{M} **construct** \mathcal{G}_{top} from Q and Q' ▷ **Construct**
- 4: Initialize: *indegree*, *queue*, *qa*, *path* as per \mathcal{G}_{top}
- 5: **while** not *queue.empty()* **do**
- 6: $q \leftarrow \text{queue.popleft}()$
- 7: Use the *Self-verify* to predict **Decomposition** and answer a based on the *path* ▷ **Self-verify**
- 8: **if** **Decomposition** == **YES** **then**
- 9: Though ▷ **Critique** and ▷ **Construct** to add a \mathcal{G}_{child} for \mathcal{G}_{top} ; Update status
- 10: **else if** **Decomposition** == **NO** **then**
- 11: $qa.append((q, a))$; Update status
- 12: **end if**
- 13: **end while**
- 14: \mathcal{M} predicts \mathcal{A} for Q based on qa

LLMs responses to complex knowledge-intensive questions are more likely due to the model’s fabrication of information rather than the propagation of erroneous learned content. Consequently, this paper posits that model knowledge should be the same as factual evidence.

3 Topology-of-Question-Decomposition

This section delineates the design of the Topology-of-Question-Decomposition (ToQD). The overall pipeline of ToQD is shown in Figure 2, which includes decomposing the input question Q into topology graph \mathcal{G}_{top} and using *self-verify* inference to answer each sub-question.

3.1 Overview

Algorithm 1 delineates the comprehensive procedural framework of ToQD. Given the input question Q , we use the prompts like "*Decompose original question reasoning steps into 2 to 6 simply and logically connected sub-questions for helping students reason towards the answers.*" to instruct LLMs for decomposing Q into sub-questions $Q' = \{q_1, q_2, \dots, q_n\}$. Subsequently, employing the prompt like "*Filter out any off-topic or irrelevant sub-question,*" ToQD guides the LLMs to **critique** (▷ **Critique**) the relevance of each $q \in Q'$ in addressing Q , thereby eliminating redundant reasoning steps (**IsRel** indicates relevance). Throughout the processes of decomposition and critique, each $q \in Q'_{rel}$ is sufficiently simplified for LLMs to determine their ability to respond, ideally restricting the focus to a single entity. These streamlined sub-questions enhance the precision of IR from knowledge sources by focusing exclusively on a single entity, thereby facilitating more efficient retrieval, predominantly through semantic search methodologies (Reimers and Gurevych, 2019; Chen et al., 2024). Following this, LLMs analyze the relationships between Q and Q'_{rel} as well as among Q'_{rel} themselves to **construct** a topology graph \mathcal{G}_{top} (▷ **Construct**), wherein each node within \mathcal{G}_{top} represents a sub-question $q \in \mathcal{G}_{top}$. In the resolution process of \mathcal{G}_{top} , LLMs employ *self-verify* (▷ **Self-Verify**) inference to determine whether q should retrieve documents, necessitates further decomposition (**Decomposition** is 'YES'), or can be directly answered. Within \mathcal{G}_{top} , if nodes align either in parallel ('**Composition**') or sequen-

Algorithm 2 Self-Verify Inference

Require: Generator LM \mathcal{M} , Retriever \mathcal{R} , Large-scale document collections $\{d_1, \dots, d_N\}$

- 1: **Input:** input question q_t , preceding sub-questions $(q_{<t}, a_{<t})$ **Output:** new (q_t, a_t) or **Decomposition**
- 2: \mathcal{M} rewrite q_t for q_{new} if given $(q_{<t}, a_{<t})$ for q_{new} and predicts the **Retrieve** based on the q_t or $q_{new} \triangleright$ **Rewrite**
- 3: **if Retrieve == YES then**
- 4: **Retrieve** relevant documents \mathcal{D} using \mathcal{R} given q_{new}
- 5: \mathcal{M} predicts **Decomposition** given q_{new} , each $d \in \mathcal{D}$
- 6: **if Decomposition == YES then**
- 7: Return **Decomposition** for **ToQD** inference
- 8: **else if Decomposition == NO then**
- 9: \mathcal{M} predicts a_t given q_{new} , each $d \in \mathcal{D}$
- 10: **end if**
- 11: **else if Retrieve == NO then**
- 12: \mathcal{M} predicts a_t given q_{new}
- 13: **end if**

tially (**Bridge**), the LLMs adjust their responses to expedite the resolution process. Upon resolving the sub-questions through topological sorting, ToQD summarizes each node of $q \in \mathcal{G}_{top}$ to predict the final answer \mathcal{A} for \mathcal{Q} . Throughout the ToQD pipeline, ToQD employs selective retrieval via LLMs only when necessary to enhance the traceability of generated content and accelerate the reasoning process through expedited parallel reasoning. Appendix B illustrates ToQD’s management of three prevalent question types in multi-hop questions: "**Composition**," "**Bridge**," and "**Bridge and Composition**."

3.2 Self-Verify

Within the structure of $q \in \mathcal{G}_{top}$, each sub-question node q is subject to distinct processing pathways. If q qualifies as an **internal** node, ToQD systematically **rewrites** q into a new formulation question q_{new} adhering to the established reasoning path within \mathcal{G}_{top} (\triangleright **Rewrite**). Additionally, ToQD evaluates the necessity to procure pertinent documents \mathcal{D} from an extensive knowledge source. In contrast, for **leaf** nodes within \mathcal{G}_{top} , ToQD utilizes the prompt like "*Can you directly answer the question '{question}'*" to gauge whether LLMs can directly answer subquestion q , thereby autonomously determining the necessity for immediate document retrieval. We hypothesize that LLMs are not capable (or capable) of solving the question when they respond with "**NO!**" (**Retrieve** is "**YES**"). Simultaneously, the ToQD commences interaction with IR to retrieve the relevant documents \mathcal{D} using the retriever \mathcal{R} . If LLMs respond with "**YES**", the LLMs will directly answer the subquestion q . Direct prompting operates intuitively and is inde-

pendent of the model’s intrinsic in-context learning capabilities, thus enabling effective performance even on models with reduced parameter sizes and better assisting the model in assessing and evaluating its ability to answer the subquestion q . After obtaining the relevant documents \mathcal{D} , ToQD employs the prompt like "*Can you directly answer the question '{question}' based on the document '{document}'?*" as a template to ascertain whether LLMs can answer the subquestion q based on the relevant documents \mathcal{D} . Similarly, ToQD employs the LLMs criterion to determine whether to answer the question directly or decompose it further within the ToQD inference process. When further decomposition is required, the processes of decomposition, critique, and construct are employed to add a child graph to the existing \mathcal{G}_{top} , dynamically adjusting the reasoning steps in response to evolving analytical needs. The global perspective embedded within ToQD prompts LLMs to intensify their exploration of potential answers when encountering intermediate challenges. By minimizing interaction with IR and maximizing the use of LLMs’ self-knowledge, the **self-verify** strategy accelerates reasoning speed and reduces the potential factual errors from IR.

3.3 Awareness of Unknowns in Large Language Models

LLMs, often referred to as '*knowledge warehouses*' can generate well-calibrated predictions for token probabilities under on-distribution (Guo et al., 2017). LLMs such as GPT (Radford et al., 2019) predominantly utilize the Transformer architecture (Vaswani et al., 2023) for generating textual sequences. The probability of predicting the subsequent token w_{t+1} in the Transformer architecture, based on the preceding sequence (w_1, w_2, \dots, w_t) , is mathematically expressed as:

$$P(w_{t+1}|w_1, w_2, \dots, w_t) = \text{softmax}(\mathbf{h}_t \mathbf{W} + \mathbf{b})$$

where, \mathbf{h}_t indicates the hidden state at time t , \mathbf{W} denotes a weight matrix, and \mathbf{b} is a bias vector. When LLMs encounter an unfamiliar entity, the likelihood of correctly predicting related subsequent tokens is low, indicating limited model familiarity. Thus, to determine whether black-box LLMs are capable of solving it is simple to ask them directly for a '**YES/NO**,' as detailed in Algorithm 2. Experiments (Section 4.4) also show that LLMs predominantly recognize their knowledge limitations, regardless of the prompting style used—whether

	Multi-Hop QA			Long-Form QA
	HotPotQA	MuSiQue	2WikiMultiHopQA	ELI5
<i>Without Information Retriever</i>				
Direct Prompting	31.95	5.91	25.82	21.90
Auto-CoT	33.53	10.55	29.15	21.55
CoT	35.04	9.46	30.41	21.79
CoT-SC	36.85	10.02	32.68	22.05
Recite-and-answer	36.49	10.97	32.53	22.10
Self-Ask w/o IR	33.95	11.10	35.65	21.73
Least-to-Most	34.05	11.45	32.88	21.95
Plan-and-Solve	36.33	12.95	35.68	22.23
SearChain w/o IR	38.36	13.61	40.49	22.54
ToQD w/o IR	39.47	15.91	43.85	23.17
<i>Interaction with Information Retriever</i>				
Direct Retrieval	34.09	10.22	30.01	23.40
ToolFormer	36.75	12.98	35.49	23.05
Self-Ask	40.05	14.28	39.58	23.25
Plan-and-Solve w/ IR	41.65	15.07	42.05	24.56
React → CoT-SC	43.15	15.49	40.36	24.05
Verify-and-Edit	44.03	15.57	40.83	23.80
Tree-of-Thought w/ IR	50.65	15.61	42.49	24.20
DSP	51.97	15.83	43.52	23.46
SearChain	56.91	17.07	46.27	25.57
ToQD w/ IR	49.16	19.39	51.06	26.23

Table 1: Performance of ToQD and baselines on complex knowledge-intensive tasks. **Bold text** denotes the best result in different settings. Metric for Long-Form QA: ROUGE-L. Metric for others: cover-EM.

"*Directly*," "*CoT*," or "*RAG*". Appendix B shows more visual cases illustrating how LLMs recognize their knowledge limitations.

4 Experiments

For detailed descriptions of the experimental implementation, readers are directed to Appendix A, while further visual illustrations of the ToQD process can be found in Appendix B. Additionally, the comprehensive prompt utilized within the ToQD pipeline can be found in Appendix C.

4.1 Experiments Setup

Baselines. Our baseline models are categorized into two groups. The first group focuses on enhancing the reasoning capabilities of LLMs on complex tasks, including CoT (Wei et al., 2023), CoT-SC (Wang et al., 2023c), Auto-CoT (Zhang et al., 2022), Recite-and-answer (Sun et al., 2023), and Least-to-Most (Zhou et al., 2023). The second group not only introduces IR to LLMs but also aims to enhance their reasoning abilities, featuring Direct prompt, Plan-and-Solve (Wang et al., 2023b), SelfAsk (Press et al., 2023a), ToolFormer (Schick et al., 2023), React (Yao et al., 2023b), DSP (Khattab et al., 2023), Verify-and-Edit with CoT-SC (Zhao et al., 2023), and Tree-of-Thought (Yao et al., 2023a).

Datasets and Evaluation Metrics. To enhance

the assessment of the ToQD, we engage with two complex knowledge-intensive tasks: multi-hop question-answering (HotPotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022b), and 2WikiMultiHopQA (Ho et al., 2020)) and long-form question-answering (ELI5 (Fan et al., 2019)). For the evaluation metrics, **ROUGEL** (Lin, 2004) is utilized for ELI5, given its long and free-form ground truth. For other tasks, the metric applied is **cover-EM** (Rosset et al., 2021), which assesses whether the ground truth answer is encapsulated within the generated response.

Implementation Details. By default, the LLM employed in this study is *gpt-3.5-turbo*, sourced from the OpenAI API¹. We utilized the top five documents from Contriever-MS MARCO (Izacard et al., 2022) for the Wikipedia 2017² as the knowledge corpus. A single RTX 4090 GPU powers the corresponding IR model. Additionally, within the experimental framework of this study, the interaction duration for the single-instance IR system was recorded at **1.72** seconds. All LLM APIs were tested under settings where both the *temperature* and *top_p* were set to 0.1 to minimize randomness in the responses, thereby facilitating the LLMs' recognition of their knowledge limitations during the *self-verify* inference process.

¹<https://openai.com/index/openai-api/>

²<https://www.wikipedia.org/>

4.2 Main Results

The main results are shown in Table 1.

Effect of Topology-of-Question Decomposition.

We compare ToQD against recent state-of-the-art baselines in settings without IR. Comparative analyses demonstrate that ToQD without IR merely surpasses all CoT-based baselines (CoT, Auto-CoT, CoT-SC, and Recite-and-Answer), highlighting the efficacy of topology-graph reasoning through structured sub-questions over mere presentation of intermediate results. Furthermore, ToQD without IR outshines both Self-Ask without IR and Least-to-Most, demonstrating the superior efficacy of employing a global topology-graph reasoning approach over sequentially generating and addressing sub-questions step by step. Finally, ToQD without IR almost consistently outperforms SearChain without IR and Plan-and-Solve across all datasets. This suggests that explicitly defining logical relationships in a topology graph among sub-questions provides a distinct advantage over merely employing a chain-of-reasoning approach.

Effect of Minimizing Interactions with IR. In settings involving interaction with IR, ToQD still marginally outperforms all baselines. By leveraging *self-verify* inference strategies to minimize redundant interactions with information retrieval systems, ToQD enhances the utilization of intrinsic knowledge and mitigates the influence of potential factual errors from IR compared to the Verify-and-Edit approach. Additionally, ToQD increases the precision of IR retrieval by decomposing multi-reasoning questions into sub-questions that are as simple as possible and involve only a single simple entity. By initially decomposing complex questions into a topology graph (**Globally**) and employing *self-verify* inference at each node (**Locally**), ToQD not only ensures the coherence of LLM reasoning but also exhibits superior coherence compared to methods used in Self-Ask, DSP, and React. Simultaneously, by preemptively applying critical filtering to sub-questions through critique (\triangleright **Critique**), ToQD avoids redundant reasoning, thereby enhancing the effectiveness of the inference process compared to Tree-of-Thought, DSP, and SearChain. Lastly, ToQD allows for the further decomposition of overly complex sub-questions, enabling the LLM to modify the direction of reasoning compared to Plan-and-Solve dynamically.

Method	#n ↓	#m ↓	#r ↓	t(s) ↓	Perf. (Avg) ↑
ToQD	370	110	1.30	6.29	36.46
- w/o Critique	487	145	1.47	10.05	34.95
- w/o Construct	329	87	1.16	9.97	34.41
- w/o Self-Verify	512	153	2.27	11.36	35.78
- w/o Rewrite	337	93	1.53	5.72	35.39

(a) Ablation analysis of removing each trigger action.

Method	#n ↓	#m ↓	#r ↓	t(s) ↓	Perf. (Avg) ↑
Self-Ask	401	63	2.19	6.63	29.29
Plan-and-Solve	450	71	1	6.05	30.83
React \rightarrow CoT-SC	938	110	2.35	8.25	30.76
Verify-and-Edit	565	307	2.40	13.90	31.06
Tree-of-Thought	622	341	2.29	13.28	33.24
DSP	1759	155	2.15	10.47	33.70
SearChain	390	189	2.21	8.52	36.46
ToQD	370	110	1.30	6.29	36.46

(b) Efficiency analysis of methods

Table 2: Efficiency and Ablation analysis: n (input words), m (output words), r (interaction rounds), t (time per interaction), Perf(Avg) (average accuracy). The **bold text** indicates the best performance.

4.3 Analysis

Detailed setups of the retrieval effectiveness and robustness experiments can be found in Appendix A, with the results of the robustness tests displayed in Appendix A.1. For further visual illustrations of ToQD, readers are directed to Appendix B.

Effects of Removing Each Trigger Actions. To rigorously assess the efficacy of each triggered action in ToQD, systematic ablation studies were conducted by removing each trigger action from the ToQD. Table 2.a presents metrics for individual queries, including the number of words in the LLM’s input (n) and output (m), interaction rounds with IR (r), and overall running time (t). Additionally, the average performance score ($Perf(Avg)$), representing aggregate results across all datasets, is also detailed. Exceptionally, when actions such as critique (\triangleright **Critique**), construct (\triangleright **Construct**), and rewrite (\triangleright **Rewrite**) were removed, the efficiency of ToQD marginally increases. However, the significant decrease in average performance scores indicates that excessive ineffective reasoning, overly complex sub-questions, and a chain-like approach to answering sub-questions can reduce the accuracy of responses. On the other hand, when the action *self-verify* (\triangleright **Self-Verify**) inference was removed, the efficiency of ToQD significantly decreased, and the frequency of interactions with IR substantially increased. Concurrently, a slight decrease in accuracy suggests reducing the impact of factual errors

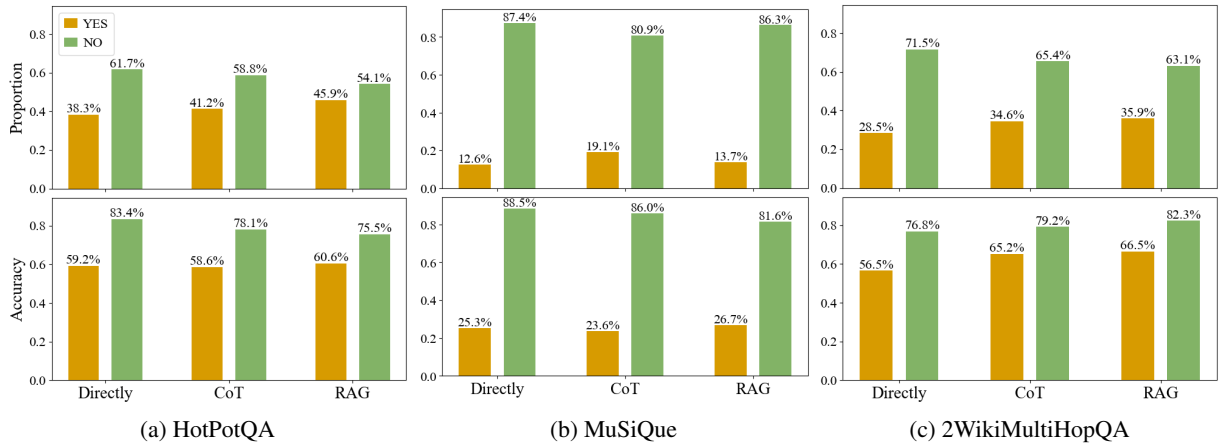


Figure 3: Accuracy and proportion of 'YES!'/'NO!' responses using different templates (Directly Prompt, CoT, and RAG) across four evaluation datasets. The upper section of the image displays the accuracy for 'YES!'/'NO!' responses, while the lower section shows the percentage for 'YES!'/'NO!' responses.

from IR. The results show a performance drop no matter which action was removed, illustrating that each action contributed to improving the efficiency and accuracy of generation.

Effects of Self-Verify Inference. To rigorously assess the efficacy of utilizing *self-verify* inference to minimize interactions with IR systems in ToQD, systematic efficiency experiments and their results are detailed in Table 2.b. This table demonstrates that our method significantly enhances task performance by minimizing interactions with IR, whereas most baselines still require multiple rounds of interaction between IR and LLMs. Additionally, the reduced number of characters in input and output text and accelerated reasoning speeds demonstrate the improvements in reasoning efficiency resulting from minimizing interactions with IR. Concurrently, our efficiency experiments confirm two critical insights: (1) In multi-reasoning tasks, not all reasoning steps must be sequential; implementing parallel reasoning can accelerate the process. (2) *LLMs are knowledge warehouses.*—by leveraging the intrinsic knowledge of the model rather than heavily relying on the RAG pipeline for query responses, we can streamline reasoning processes, reduce the impact of factual errors from IR systems, and consequently enhance the accuracy of LLMs' generations.

Effects of Different Templates on Eliciting Self-Knowledge in LLMs. To systematically elicit self-knowledge from large language models (LLMs), we designed and implemented three distinct prompting templates: **Direct**, **CoT**, and **RAG**, consistent with those used in the ToQD frame-

work. We evaluated these templates using complex knowledge-intensive questions from four multi-hop datasets—HotPotQA, MuSiQue, and 2WikiMultiHopQA—that LLMs had initially deemed challenging. The outcomes of these evaluations are depicted in Figure 3. The experimental procedures consisted of two distinct phases: initially, LLMs responded directly to questions using the specified templates. In a subsequent phase, we modified the prompts for each template to probe the LLMs' self-assessment capabilities. This modification aimed to assess the LLMs' self-awareness regarding their problem-solving abilities. We then quantitatively measured the accuracy of their self-assessment responses to determine their reliability in affirming or denying their capacity to solve the presented questions. This bifurcated methodology facilitated a thorough examination of both the direct response capabilities and the introspective accuracy of the LLMs across diverse prompt templates (**Direct**, **CoT**, and **RAG**).

Initially, across all prompt templates, LLMs exhibited either a positive response—directly providing the predicted answers—or a negative response, which indicated the necessity for external information or further decomposition in addressing specific questions. Secondly, regardless of the prompt template used, the *gpt-3.5-turbo* consistently demonstrated awareness of its limitations, a phenomenon colloquially known as “*known unknowns*.” Concurrently, the proportion of "NO!" responses from *gpt-3.5-turbo* aligned with its error rate when directly responding to queries from the dataset, revealing its ability to predominantly identify what LLMs do

Method	Hits@10	MAP@10	MRR@10
Native	0.586	0.160	0.353
HyDE	0.611	0.164	0.362
SubQuestion	0.334	0.040	0.085
MultiQuery	0.426	0.092	0.217
ToQD	0.614	0.168	0.329
w/o critique	0.573	0.142	0.357
w/o rewrite	0.597	0.151	0.334

Table 3: Retrieval performance of different rewrite query method in the MultiHop-RAG dataset.

not know.

Effects of Decomposing Complex Questions. To more accurately assess the impact of decomposing complex questions into simpler, single-entity questions on IR retrieval accuracy, we conducted our evaluation using the MultiHopRAG dataset (Tang and Yang, 2024). Our approach was compared against four query-rewrite baselines—native, HyDE (Gao et al., 2023), MultiQuery, and SubQuestion³—employing retrieval evaluation metrics such as Mean Average Precision at K (MAP@K), Mean Reciprocal Rank at K (MRR@K), and Hit Rate at K (Hit@K) to assess retrieval quality. Table 3 indicates that our method not only facilitates reasoning but also enhances the precision of IR retrieval, surpassing the performance of directly retrieving factual evidence and various baselines. Furthermore, the reduction in retrieval effectiveness resulting from the removal of the trigger actions of critique (\triangleright Critique) and rewrite (\triangleright Rewrite) underscores the significant role these actions play in aiding retrieval. The experimental results from the MultiHop-RAG dataset demonstrate that employing a simple, single-entity question format can substantially improve retrieval effectiveness.

4.4 Robustness Test

We analyzed models with varying parametric capacities to rigorously assess the robustness of our proposed method’s reasoning capabilities. Figure 4 depicts the comparative performance of CoT with IR (\bullet) and ToQD with IR (\triangle) on models ranging from 2B to over 100B parameters within the 2WikiMultiHop subdataset (5k questions). The models evaluated encompass Gemma-2b (Team et al., 2024), Chatglm-2-6B (GLM et al., 2024), Llama-2 series (Touvron et al., 2023), Llama-3 series (Dubey et al., 2024), Mistral series (Jiang et al., 2023), and

³<https://www.llamaindex.ai/>

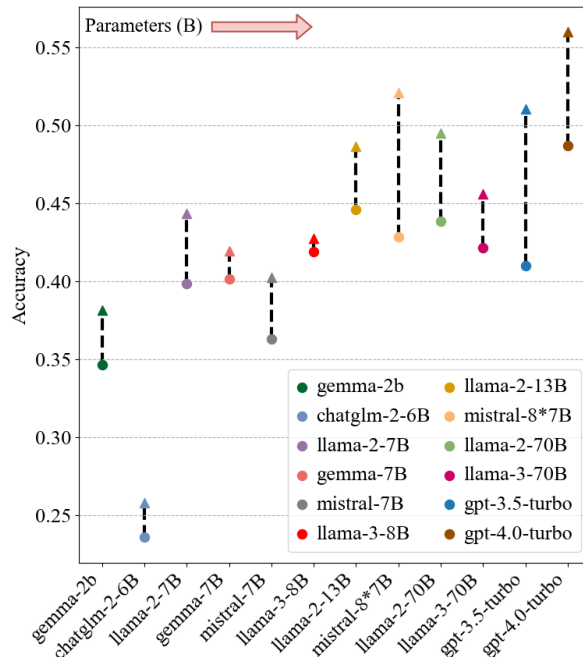


Figure 4: Evaluation of robustness across varying model sizes on the 2WikiMultiHop dataset. Key: \bullet indicates CoT with IR; \triangle denotes ToQD with IR. Color variations represent different models, with model parameters increasing from left to right.

the GPT series⁴. The analysis reveals that ToQD with IR consistently outperformed CoT across all tested parametric scales. This improvement is ascribed to implementing simpler prompt templates without numerical examples, significantly reducing the reliance on the models’ in-context learning capabilities. Consequently, these findings corroborate the enhanced robustness of reasoning performance afforded by ToQD compared to CoT.

5 Conclusion

This paper examines the limitations of integrating IR into LLMs from perspectives of reasoning and knowledge while exploring how to more effectively utilize the inherent knowledge of LLMs for addressing complex, knowledge-intensive tasks. We introduce ToQD, a novel framework tailored to facilitate efficient interaction between IR systems and LLMs. ToQD methodically constructs a reasoning process by organizing sub-questions into a topology graph and employs *self-verify* inference to ascertain whether a sub-question requires further decomposition, the retrieval of relevant documents or can directly provide an answer. Experimental results demonstrate that ToQD surpasses state-of-

⁴<https://openai.com/>

the-art baselines in handling complex tasks and significantly reduces interactions with IR, enhancing reasoning efficiency. Additionally, ToQD’s simplified template extends its applicability across models with parameters ranging from 2B to over 100B, showcasing robustness.

Limitations

In this paper, we introduced ToQD, a novel framework intended to improve interactions between IR systems and LLMs. A key aspect of our approach involved utilizing self-verify inference to identify the constraints of LLMs. Despite its utility, this method has not achieved sufficient accuracy, highlighting a need for further research to enhance our understanding of the inherent limitations of black-box LLMs. This is crucial for assisting large models in accurately addressing knowledge gaps. Additionally, the tasks of self-verification and topology graph construction were managed exclusively by LLMs. Future work should focus on optimizing these processes by investigating how smaller models might be employed to decrease dependency on LLMs.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61966038 and 62266051, and by the National Science and Technology Council (NSTC) of Taiwan under Grant No. 113-2221-E-155-046-MY3. The authors would like to thank the anonymous reviewers for their constructive comments.

References

Simran Arora, Avanika Narayan, Mayee F. Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Ré. 2022. [Ask me anything: A simple strategy for prompting language models](#). *Preprint*, arXiv:2210.02441.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. [Graph of thoughts: Solving elaborate problems with large language models](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690.

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. [Scaling instruction-finetuned language models](#). *Preprint*, arXiv:2210.11416.

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*.

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

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

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. [Retrieval-augmented generation for large language models: A survey](#). *Preprint*, arXiv:2312.10997.

Team GLM, :, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Jingyu Sun, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. [Chatglm: A family of large language models from glm-130b to glm-4 all tools](#). *Preprint*, arXiv:2406.12793.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. [Textbooks are all you need](#). *Preprint*, arXiv:2306.11644.

- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. [On calibration of modern neural networks](#). *Preprint*, arXiv:1706.04599.
- Hangfeng He, Hongming Zhang, and Dan Roth. 2022. [Rethinking with retrieval: Faithful large language model inference](#). *Preprint*, arXiv:2301.00303.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring massive multitask language understanding](#). *Preprint*, arXiv:2009.03300.
- 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.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. [Unsupervised dense information retrieval with contrastive learning](#). *Preprint*, arXiv:2112.09118.
- Palak Jain, Livio Baldini Soares, and Tom Kwiatkowski. 2024. [From rag to riches: Retrieval interlaced with sequence generation](#). *Preprint*, arXiv:2407.00361.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Zhouyu Jiang, Mengshu Sun, Lei Liang, and Zhiqiang Zhang. 2024. [Retrieve, summarize, plan: Advancing multi-hop question answering with an iterative approach](#). *Preprint*, arXiv:2407.13101.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. [TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. [Language models \(mostly\) know what they know](#). *Preprint*, arXiv:2207.05221.
- Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2023. [Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp](#). *Preprint*, arXiv:2212.14024.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. [Large language models are zero-shot reasoners](#). *Preprint*, arXiv:2205.11916.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K ttler, Mike Lewis, Wen tau Yih, Tim Rockt schel, Sebastian Riedel, and Douwe Kiela. 2021. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). *Preprint*, arXiv:2005.11401.
- Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. [Towards general text embeddings with multi-stage contrastive learning](#). *Preprint*, arXiv:2308.03281.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. [When not to trust language models: Investigating effectiveness of parametric and non-parametric memories](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.
- Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2024. [Multi-hop question answering](#). *Preprint*, arXiv:2204.09140.
- OpenAI. 2020. Language models are few-shot learners. <https://cdn.openai.com/papers/gpt-4.pdf>. Accessed: insert-access-date-here.
- Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rockt schel, and Sebastian Riedel. 2021. [KILT: a benchmark for knowledge intensive language tasks](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023a. [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.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023b. [Measuring and narrowing the compositionality gap in language models](#). *Preprint*, arXiv:2210.03350.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Preprint*, arXiv:1910.10683.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). *Preprint*, arXiv:1908.10084.
- Corby Rosset, Chenyan Xiong, Minh Phan, Xia Song, Paul Bennett, and Saurabh Tiwary. 2021. [Knowledge-aware language model pretraining](#). *Preprint*, arXiv:2007.00655.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. [Toolformer: Language models can teach themselves to use tools](#). *Preprint*, arXiv:2302.04761.
- Sergio Servantez, Joe Barrow, Kristian Hammond, and Rajiv Jain. 2024. [Chain of logic: Rule-based reasoning with large language models](#). *Preprint*, arXiv:2402.10400.
- 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.
- Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Gergely Szilvassy, Rich James, Xi Victoria Lin, Noah A. Smith, Luke Zettlemoyer, Scott Yih, and Mike Lewis. 2024. [In-context pretraining: Language modeling beyond document boundaries](#). *Preprint*, arXiv:2310.10638.
- Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2023. [Recitation-augmented language models](#). *Preprint*, arXiv:2210.01296.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. [Challenging big-bench tasks and whether chain-of-thought can solve them](#). *Preprint*, arXiv:2210.09261.
- Yixuan Tang and Yi Yang. 2024. [Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries](#). *Preprint*, arXiv:2401.15391.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shrutu Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022a. [Musique: Multi-hop questions via single-hop question composition](#). *Preprint*, arXiv:2108.00573.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022b. [MuSiQue: Multi-hop questions via single-hop question composition](#). *Transactions of the Association for Computational Linguistics*, 10:539–554.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. [Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. [Attention is all you need](#). *Preprint*, arXiv:1706.03762.
- 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.
- Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023b. [Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models](#). *Preprint*, arXiv:2305.04091.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023c. [Self-consistency improves chain of thought reasoning in language models](#). *Preprint*, arXiv:2203.11171.

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

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](#). *Preprint*, arXiv:2304.14732.

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, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. [Tree of thoughts: Deliberate problem solving with large language models](#). *Preprint*, arXiv:2305.10601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. [React: Synergizing reasoning and acting in language models](#). *Preprint*, arXiv:2210.03629.

Zhangyue Yin, Qiusi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. [Do large language models know what they don’t know?](#) In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8653–8665, Toronto, Canada. Association for Computational Linguistics.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. [Making retrieval-augmented language models robust to irrelevant context](#). *Preprint*, arXiv:2310.01558.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. [Generate rather than retrieve: Large language models are strong context generators](#). *Preprint*, arXiv:2209.10063.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. [Star: Bootstrapping reasoning with reasoning](#). *Preprint*, arXiv:2203.14465.

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

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. [Automatic chain of thought prompting in large language models](#). *Preprint*, arXiv:2210.03493.

	Multi-Hop QA			LFQA
	HotPot	MQ	WQA	ELI5
Without Information Retriever				
Direct Prompting	0	0	0	0
Auto-CoT	4	4	4	2
CoT	4	4	4	2
CoT-SC	4	4	4	2
Recite-and-answer	4	4	4	2
Self-Ask w/o IR	4	4	4	2
Least-to-Most	4	4	4	2
Plan-and-Solve	4	4	4	2
SearChain w/o IR	2	2	2	2
ToQD w/o IR	0	0	0	0
Interaction with Information Retriever				
Direct Retrieval	0	0	0	0
ToolFormer	4	4	4	2
Self-Ask	4	4	4	2
Plan-and-Solve w/ IR	4	4	4	2
React $\rightarrow\rightarrow$ CoT – SC	6	4	4	2
Verify-and-Edit	2	2	2	2
Tree-of-Thought w/ IR	4	4	4	2
DSP	16	8	8	2
SearChain	2	2	2	2
ToQD w/ IR	0	0	0	0

Table 4: Number of examples in a prompt template used for in-content learning on different datasets.

Ruo Chen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. [Verify-and-edit: A knowledge-enhanced chain-of-thought framework](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5823–5840, Toronto, Canada. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. [Least-to-most prompting enables complex reasoning in large language models](#). *Preprint*, arXiv:2205.10625.

A Experiment Details

This section presents a comprehensive examination of the experimental configurations employed in our study, encompassing the main, retrieval, and robustness experiments.

A.1 Number of Examples in Prompt Template

In Table 4, we delineate the count of examples utilized in the prompt templates for in-context learning across several datasets (HotPotQA (HotPot), MuSiQue (MQ), 2WikiMultiHopQA (WQA) and ELI5) corresponding to various baselines. Unlike

baseline methods, which utilize at least two examples, the ToQD prompt templates employ 0 examples yet achieve superior performance, surpassing competitive baselines. This approach significantly enhances the applicability of the templates across a range of models, with parameters extending from 2B to over 100B, by minimizing reliance on in-context learning.

A.2 Supplementary Details for Main Experiment Settings

The primary LLM utilized in this study segment is *gpt-3.5-turbo*, sourced from the OpenAI API. The top five documents from the Contriever-MS MARCO (Izacard et al., 2022) indexed from Wikipedia 2017 were employed for the knowledge corpus. A single RTX 4090 GPU powers the corresponding IR model. All LLM APIs were calibrated with temperature and top_p parameters set to 0.1 to curtail randomness in the generated responses. This configuration was pivotal in enabling LLMs to ascertain their knowledge limitations during the *self-verify* inference process. Moreover, the limit of inference processes, r_{max} , is restricted to 6, a measure to mitigate excessive cognitive extrapolation and constrain the topology graph’s expansion within the ToQD framework. Following methodologies such as DSP, Self-ASK, and SearChain, the model was evaluated on complete development datasets of MuSiQue and HotPotQA and selected subsets of 2WikiMultiHopQA (5k questions) and ELI5 (1.2k questions).

A.3 Supplementary Details for Retrieval Experiment Settings.

MultiHop-RAG (Trivedi et al., 2022a) introduces an innovative dataset designed to support queries that require the retrieval and synthesis of multiple evidence pieces, thus more accurately reflecting real-world scenario complexities. The dataset comprises 2,556 multi-query instances, such as "Which company among Google, Apple, and Nvidia reported the largest profit margins in their third-quarter reports for 2023?", which necessitates compiling evidence from multiple documents to derive an answer. It categorizes these multi-hop queries into four types—*Inference*, *Comparison*, *Temporal*, and *Null*—each mirroring the complexities often encountered in real-world situations. Concurrently, the MultiHop-RAG dataset defines the corpus content, enhancing the precision of search experiments and tests. When retrieving the top_k

chunks, denoted as $|\mathcal{R}_q = K|$, several retrieval evaluation metrics are employed, including Mean Average Precision at K (MAP@K), Mean Reciprocal Rank at K (MRR@K), and Hit Rate at K (Hit@K). MAP@K assesses the average precision of top-K retrievals across all queries, MRR@K calculates the average reciprocal ranks of the first relevant chunk for each query within the top-K set, and Hit@K gauges the proportion of relevant evidence within the top-K retrieved chunks. For this experimental setup, ChromaDB⁵ was utilized as the vector database, with gte-base(Li et al., 2023) employed for embedding text chunks and facilitating retrieval, the size of each chunk being 256.

Detailed analyses of the four baseline methods⁶ employed in our retrieval experiment are presented below:

(1) *Native*: This method straightforwardly retrieves relevant documents from the knowledge source based on the query.

(2) *HyDE*: The Hypothetical Document Embeddings (HyDE) method improves retrieval by generating and embedding a hypothetical document to represent a query, using this representation to find and retrieve similar real documents effectively.

(3) *SubQuestion*: This approach employs a sub-question query engine designed to address complex queries using multiple data sources. Initially, the method decomposes a complex query into sub-questions tailored to each relevant data source. Subsequently, it aggregates the intermediate responses from these sources and synthesizes a comprehensive final response.

(4) *MultiQuery*: This approach features a multi-step query engine designed to decompose complex queries into a series of sequential sub-questions, enabling detailed and focused information retrieval from various data sources.

A.4 Details of the Robustness Test

To rigorously evaluate the robustness of our proposed method’s reasoning capabilities with IR, we analyzed models with parametric capacities ranging from 2 billion to over 100 billion parameters, explicitly focusing on the 2WikiMultiHop sub-dataset, which comprises 5,000 questions. The remaining baseline settings are consistent with those of the main experiment. The ablation of 6 presents the detailed improvements of ToQD with IR over

⁵<https://www.trychroma.com/>

⁶The API source is <https://www.llamaindex.ai/>

Model	2WikiMultiHopQA		Imp (%) [↑]
	CoT w/ IR	ToQD w/ IR	
gemma-2b	34.63	38.17	3.54
chatglm-2-6B	23.57	25.79	2.19
llama-2-7B	39.85	44.33	4.48
gemma-7B	40.14	41.95	1.81
mistral-7B	36.31	40.27	3.96
llama-3-8B	41.89	42.74	0.85
llama-2-13B	44.62	48.67	4.05
mistral-8*7B	42.88	52.10	9.22
llama-2-70B	43.83	49.50	5.67
llama-3-70B	42.17	45.63	3.46
gpt-3.5-turbo	41.01	51.06	10.05
gpt-4.0-turbo	48.73	56.14	7.41

Table 5: Robustness tests for models of various sizes in 2WikiMultiHopQA: CoT w/ IR vs. ToQD w/ IR. **Red text** indicates ToQD’s accuracy percentage points lead over CoT.

CoT with IR, with the enhancement ranging from 0.8% to 10.05%.

B Case study

In this section, we evaluate the efficacy of ToQD, Direct, and CoT under various experimental conditions, including scenarios without IR, with IR, and in solely retrieval-focused contexts, within the framework of a case study on generated content. We also employ a case study of the ToQD overall pipeline to analyze further why ToQD demonstrates superior reasoning abilities compared to baselines and to *self-verify* its inferences, thereby illustrating what LLMs predominantly do not know.

Case study for ToQD vs. Baselines in Reasoning.

The case study associated with Table 6 delineates the reasoning processes employed by ToQD versus baseline models in environments without IR in three multi-hop reasoning question type: "*Bridge*," "*Composition*" and "*Bridge and Composition*". Compared to baselines, ToQD enhances logical clarity in reasoning by decomposing the input questions into sub-questions and analyzing their relationships to construct a topology graph. Furthermore, in scenarios without IR, ToQD employs *self-verify* inferences to determine whether a direct answer is feasible or if further decomposition is required, thereby dynamically adjusting the reasoning process. This approach compels LLMs to exert additional effort when confronted with intermediate difficulties, ultimately enabling them to address complex questions effectively.

Case study for ToQD vs. Baselines with IR. The

Q: Who lived longer, Giuseppe Cesari or Nicos Poulantzas? (*Bridge*)

Direct: Nicos Poulantzas lived longer than Giuseppe Cesari. ✗

CoT: Nicos Poulantzas lived longer than Giuseppe Cesari. Poulantzas was born on 1936, and passed away on 1979, 43 years old. Giuseppe Cesari, also known as Cavaliere d’Arpino, was born on 1568, and died on 1640, 72 years old. ✗

ToQD: **1.** What are the birth and death dates of Giuseppe Cesari? Giuseppe Cesari was born on 1568, and died on 1640. **2.** What are the birth and death dates of Nicos Poulantzas? Nicos Poulantzas was born on 1936, and he passed away on 1979. **So, Giuseppe Cesari** lived longer than Nicos Poulantzas. Cesari lived for 72 years, while Poulantzas lived for 43 years. ✓

Q: Who is Marie Zephyrine Of France’s paternal grandmother? (*Composition*)

Direct: Marie Z ephyrine of France’s paternal grandmother is Maria Josepha of Saxony. ✗

CoT: Marie Z ephyrine Of France’s paternal grandmother is Empress Maria Theresa of Austria. ✗

ToQD: **1.** Who is Marie Z ephyrine Of France’s father? King Louis XVI of France **2.** whose father is King Louis XVI of France? Marie Leszczyeska **So,** Marie Z ephyrine of France’s paternal grandmother is **Marie Leszczyeska.** ✓

Q: Which film has the director who was born later, A Quiet Place In The Country or When Were You Born? (*Bridge, Composition*)

Direct: The director of "When Were You Born" was born later. ✗

CoT: To determine which film has the director who was born later, we need to find out the birthdates of the directors of both films. Let’s start by looking up the birthdates of the directors of "A Quiet Place In The Country" and "When Were You Born?". ✗

ToQD: **1.** Who is the director of "A Quiet Place In The Country" The director of Elio Petri **2.** What is the birth date of Elio Petri? January 29, 1929. **3.** Who is the director of "When Were You Born"? Vondie Curtis-Hall **4.** What is the birth date of Vondie Curtis-Hall? September 30, 1956. **So,** the final answer is The director of "**A Quiet Place In The Country**" has the director who was born later. ✓

Table 6: Case study for ToQD vs Baselines in Reasoning without IR in three different question types: (*Bridge*), (*Composition*) and (*Bridge, Composition*). ✓ means right, while ✗ is error. **Red text** is the right answer.

case study associated with Table 7 delineates the reasoning processes employed by ToQD versus baseline models with IR environments for complex, knowledge-intensive tasks. ToQD enhances the precision of IR by decomposing multi-reasoning questions into sub-questions that are as simple as possible and involve only a single entity, thereby improving document retrieval compared to direct approaches. Furthermore, ToQD minimizes interactions with IR by using *self-verify* inferences to ascertain whether to retrieve relevant documents, require further decomposition, or directly provide an answer, as further substantiated in Table 7.

Case study for ToQD vs Baselines in Retrieving. The case study corresponding to Table 8 delineates the process of retrieving relevant documents for complex, knowledge-intensive questions, which are classified into two types: "*Bridge*" and "*Composition*". As evidenced in Table 8, the strategic decomposition of these questions into simpler, single-entity sub-questions significantly improves retrieval accuracy.

Case study for ToQD in Overall Pipeline. The case study associated with Table 9 provides a detailed examination of the overall ToQD pipeline, applied to complex, knowledge-intensive questions categorized into two distinct types: "*Bridge*" and "*Composition*". This categorization facilitates a nuanced understanding of the various mechanisms employed by the ToQD pipeline to manage and resolve multifaceted inquiry challenges.

Case study for ToQD in Self-Verify Inference. The case study detailed in Table 10 lucidates the mechanism by which LLMs utilize structured prompts, specifically "*Can you directly answer the question 'question'?*", to autonomously ascertain their capacity for direct question answering, thereby enabling an assessment of their intrinsic response capabilities.

C Prompts in Experiment

We present the comprehensive prompt template utilized in the ToQD across Tables 11 through 19.

Q: When did Burma change its name to the country of origin of Zaw Win Thet, that he represented at the 2012 Olympics?

A: 1989

Relevant Documents for CoT and ToQD:

1. single sculls rower Shwe Zin Latt competed at their second consecutive Olympics. Pistol shooter Maung Kyu, who competed at his first Olympics, was the oldest member of the team, at age 41. Meanwhile, track runner Zaw Win Thet, the youngest of the team, at age 21, was appointed by the committee to be Myanmar's flag bearer at the opening ceremony. Myanmar also marked its Olympic return in judo after a twenty-year absence. Myanmar, however, has yet to win its first ever Olympic medal. Myanmar sent archers to the Olympics for the fourth time, seeking the nation's first Olympic medal in

2. training center is the Yangon City Development Committee Sports Centre. Bodybuilding in Burma Bodybuilding in Burma has a long history, dating back to the 1936 Summer Olympics. Burmese bodybuilders were competing internationally for the country in 2014. Myanmar Bodybuilding and Physique Sports Federation is the national governing body. The country's interest in weightlifting predates the 1936 Summer Olympics in Berlin where Burmese weightlifter named Zaw Weik competed at the Games, where for geopolitical reasons he was put on the Indian national team. He finished fifteenth. In 2013, Aung Swe Naing was the country's most successful bodybuilder. He had won the

Direct:

Burma officially changed its name to Myanmar in 1936 . Therefore, Zaw Win Thet represented the country of Myanmar at the 2012 Olympics. ✗

CoT:

To determine when Burma changed its name to the country of origin of Zaw Win Thet, we need to consider the information provided in the texts. From the first text, we know that Zaw Win Thet represented Myanmar at the 2012 Olympics. From the second text, we see that Zaw Weik, a Burmese weightlifter, competed at the 1936 Summer Olympics in Berlin. This indicates that at that time, the country was known as Burma. Therefore, based on the information provided, Burma changed its name to Myanmar before Zaw Win Thet represented the country at the 2012 Olympics. ✗

ToQD:

q_1 : What was the original name of Burma before it changed to its current name?

a_1 : The original name of Burma before it changed to its current name was Myanmar.

 q_2 : When did Myanmar change it name?

Relevant Documents for q_2 :

A weightlifter from what is now Myanmar, U Zaw Weik, competed for India in the 1936 Olympics. The National Olympic Committee of Burma was formed in 1947 and recognized by the International Olympic Committee (IOC) on 20 June 1947 at the IOC session in Stockholm. After gaining independence in 1948, it made its Olympic debut at the London Games that same year. It has competed at all Olympic Games since then, with the exception of the 1976 Olympics. The country changed its name to Myanmar in May 1989, and has used that name at the Olympics since the 1992 Barcelona Olympics.

a_2 : May 1989

So, 1. What was the original name of Burma before it changed to its current name? The original name of Burma before it changed to its current name was Myanmar 2. When did Myanmar change it name? May 1989. The final answer is **May 1989**. ✓

Table 7: Case study for ToQD vs Baselines with IR. ✓ means right, while ✗ is error. Red text is right answer.

Q: Where was the composer of song The Trail Of The Lonesome Pine (Song) born? *(Bridge)*

Direct:

1. The Trail of the Lonesome Pine (song) "The Trail of the Lonesome Pine" is a popular song published in 1913, with lyrics by Ballard MacDonald and **music by Harry Carroll**. ...
 3. included "The Trail of the Lonesome Pine". Released as a single, the song reached No. 2 in the UK Singles Chart, thanks largely to being championed by disc jockey John Peel on his Radio 1 evening show. The song was also recorded by Vivian Stanshall and (as "Blue ...
 4. Starring Sylvia Sidney, Henry Fonda, and Fred MacMurray, the film was nominated for an Academy Award for Best Original Song for Louis Alter and Sidney D. Mitchell's "A Melody for the Sky." ...
-

ToQD:

q₁: Who is the composer of song The Trail Of The Lonesome Pine (Song)?

1. The Trail of the Lonesome Pine (song) "The Trail of the Lonesome Pine" is a popular song published in 1913, with lyrics by Ballard MacDonald and **music by Harry Carroll**. Inspired by John Fox, Jr. ..
2. novel's author: it was designated the "official outdoor drama" by the Commonwealth of Virginia in 1994, and is now the sixth longest running outdoor drama in the United States. The song was the favorite song of Gertrude Stein. ...

q₂: Where was Harry Carroll born?

1. **Harry Carroll (November 28, 1892, in Atlantic City, New Jersey – December 26, 1962, in Mount Carmel, Pennsylvania)** was an American songwriter, pianist, and composer. ...
 2. Angeles and became involved in early movies. ...
-

Q: Are Edmonton/Villeneuve Airport and Pembroke Airport located in the same country? *(Composition)*

Direct:

1. **Edmonton/Villeneuve Airport Edmonton/Villeneuve Airport or Villeneuve Airport is located west of Villeneuve in Sturgeon County, Alberta, Canada.** The closure of the Edmonton City Centre (Blatchford Field) Airport has stimulated expansion of the Villeneuve Airport as a partial replacement. ...
 2. after year. Static displays ...
 3. the facilities in 2015. Edmonton has scheduled daily non-stop service to all major Canadian cities and several major US hubs including Denver, Las Vegas, Minneapolis, Phoenix and Seattle. ...
 4. The show follows the airport's day-to-day operations and how the staff are able to keep the airport running smoothly, even during the extreme winter weather. ...
-

ToQD:

q₁: Where was Edmonton/Villeneuve Airport located?

1. **Edmonton/Villeneuve Airport or Villeneuve Airport is located west of Villeneuve in Sturgeon County, Alberta, Canada.** The closure of the Edmonton City Centre (Blatchford Field) ...
2. after year. Static displays and food services are expected to grow. Edmonton Economic Development Corporation award the airshow the 2017 Outstanding Event award at the Edmonton Events Award Gala for the 10,000+ category, ...

q₂: Where was Pembroke Airport located?

1. **Pembroke Airport (IATA: YTA, ICAO: CYTA) is located 5.2 nautical miles (9.6 km; 6.0 mi) northwest of the city of Pembroke, Ontario, Canada.** Pem-Air operated ...
 2. Haverfordwest Airport Haverfordwest Airport, also known as Withybush Airport, is a minor airport located north of Haverfordwest, Pembrokeshire. ...
-

Table 8: Case study for ToQD for vs Baselines in retrieving in two typical question type *(Composition)* and *(Bridge)*. **Red text** is the factual evidence.

Q : Who is the mother of the director of film Polish-Russian War?	<i>(Bridge)</i>
Q' : 1. What is the name of the director of the film Polish-Russian War?; 2. Who is the mother of the director of the film Polish-Russian War?; 3. What is the relationship between the director of the film and his mother?; 4. How can we determine the identity of the director’s mother based on available information?	
Q_{rel} : 1. What is the name of the director of the film Polish-Russian War? 2. Who is the mother of the director of the film Polish-Russian War?	
G_{top} : $1 \rightarrow 2 \rightarrow Q$	
Self-verify for $q_1 \in Q_{rel}$: Can you answer the following question: ‘What is the name of the director of the film Polish-Russian War?’? If you couldn’t, just simple respond with ‘NO!’; otherwise, please provide a simple answer to the question. \mathcal{A} : Xawery Żuławski.	
Rewrite q_2 for q_{new} based on the (q_1, a_1) : q_1 : Who is the mother of the director of the film Polish-Russian War? $\rightarrow q_{new}$: Who is the mother of Xawery Żuławski?	
Self-verify for $q_{new} \in Q_{rel}$: Can you answer the following question: ‘Who is the mother of Xawery Żuławski?’? If you couldn’t, just simple respond with ‘NO!’; otherwise, please provide a simple answer to the question. \mathcal{A} : NO!	
Retrieve D : Xawery Żuławski (born 22 December 1971 in Warsaw) is a Polish film director. In 1995 he graduated National Film School in Łódź. He is the son of actress Małgorzata Braunek and director Andrzej Żuławski. ...	
Self-verify for q_{new} : Given the context information: Xawery Żuławski ... Can you answer the following question based on the context information: ‘Who is the mother of Xawery Żuławski?’ If you could, please provide the answer of the question. If not, simply respond with ‘NO!’ . \mathcal{A} : Małgorzata Braunek	
Summary : 1. What is the name of the director of the film Polish-Russian War? Xawery Żuławski. 2. Who is the mother of Xawery Żuławski? Małgorzata Braunek. So, the answer is: Małgorzata Braunek ✓	
Q : Which music group has more members, Bleeker or Bracket?	<i>(Composition)</i>
Q' : 1. How many members are in the music group Bleeker? 2. How many members are in the music group Bracket? 3. How can we compare the number of members in Bleeker and Bracket to determine which group has more members?	
Q_{rel} : 1. How many members are in the music group Bleeker? 2. How many members are in the music group Bracket?	
G_{top} : $1 \rightarrow Q, 2 \rightarrow Q$	
Self-verify for $q_1 \in Q_{rel}$: Can you answer the following question: ‘How many members are in the music group Bleeker?’? If you couldn’t, just simple respond with ‘NO!’; otherwise, please provide a simple answer to the question. \mathcal{A} : Four.	
Self-verify for $q_1 \in Q_{rel}$: Can you answer the following question: ‘How many members are in the music group Bracket?’? If you couldn’t, just simple respond with ‘NO!’; otherwise, please provide a simple answer to the question. \mathcal{A} : Three.	
Summary : 1. How many members are in the music group Bleeker? Four. 2. How many members are in the music group Bracket? Three. So, the final answer is Bleeker ✓	

Table 9: Case study for ToQD for overall pipeline in two typical question type in multi-hop question: "*Bridge*" and "*composition*". ✓ means right, while ✗ is error. Red text is the LLM’s response.

Relevant Context

[1]: Şemsettin Baş (born **January 4, 1973**, in Istanbul, Turkey) is a retired Turkish professional basketball player and lastly was head coach of CSO Voluntari. At 6 ft 5.75 in (1.97 m), he played as a shooting guard and small forward. He is of Bosniak origin and cousin of former NBA and Euroleague player Mirsad Türkcan.

[2]: Loïc Gwenc’hlan Le Scouëzec (**11 November 1929**, Plouescat, Finistère – 6 February 2008[1]) was a Breton medical doctor, writer, and Grand Druid of Brittany.

Directly Prompt

q_1 : When was the Şemsettin Baş born?

a_1 : January 4, 1973 ✓

q_1 : When was the Loïc Gwenc’hlan Le Scouëzec born?

a_1 : November 29, 1932 ✗

Self-verify

q_1 : Can you answer the following question: ‘When was the Şemsettin Baş born?’? If you couldn’t, just simple respond with ‘NO!’; otherwise, please provide a simple answer to the question.

a_1 : Şemsettin Baş born in January 4, 1973 ✓

q_2 : Can you answer the following question: ‘When was the Loïc Gwenc’hlan Le Scouëzec born?’? If you couldn’t, just simple respond with ‘NO!’; otherwise, please provide a simple answer to the question. ✓

a_2 : NO!

Table 10: Awareness of Unknowns in Large Language Models Case. ✓ means right, while ✗ is error. Red text is the factual evidence.

Given the original question:

{original_question}

Your task is to decompose original question reasoning steps into 2 to 4 logically connected sub-questions for helping students reason towards the answers of the original question. Each sub-question should capture diverse critical context of the original question clearly, concisely and entirely. These sub-questions should be logically connected and are designed to guide students towards constructing a comprehensive answer through structured reasoning. Just output the sub-questions like this: ‘1. ...’

Table 11: Decompose the question Q into sub-questions Q' .

Given the main question and its sub-questions:

Main question: ‘{original_question}’

Sub questions: {sub_questions}

Your task is to construct a topological graph to analyze the reasoning process among the sub-questions and between the sub-questions and the main question. Represent the main question as ‘Q’ and just label each sub-question by its respective numbers. Use the symbol ‘->’ to denote the reasoning paths, where the left side question (e.g., a sub-question) provides useful information that helps answer the right side question (e.g., the main question or another sub-question). Ensure that all sub-questions are interconnected in a way that they collectively and coherently contribute to answer the main question. Clearly depict individual arrows from each sub-question pointing towards the main question or to another sub-question that it helps to answer. This detailed topology graph should be designed to guide students towards constructing a comprehensive answer through structured reasoning.

Table 12: Construct the topology graph \mathcal{G}_{top} .

Given the main question and its sub-questions:

Main question: ‘{original_question}’

Sub questions: {sub_questions}

Your task is:

1. Filter out the sub-question cannot help students reason towards the answers of the original question.
 2. Filter out the sub-question cannot capture critical context of the original question clearly and concisely.
 3. Filter out the sub-question semantics repetition.
-

Table 13: Critique the irrelevant sub-questions.

Given the question-answer dialog information:

{qa_dialog}

Please provide the answer of the question: '{original_question}'.

Table 14: Summary the question-answer dialog to answer the input question.

Given the answers from the sub-questions:

{qa_dialog}

Your task is to rewrite the main question: '{question}' to just incorporate the answers from the sub-questions directly into the main question. Avoid repetition of the information already provided in the sub-questions. The new question should be simply and concisely help students reasoning the original question: '{original_question}'.

Table 15: Incorporate relevant information into the question according to the reasoning sub-questions.

Can you answer the following question: '{question}'?

If you couldn't, just simple respond with 'NO!'; otherwise, please provide a simple answer to the question.

Table 16: Self-verify to decide when to decompose the question in the leaf node for \mathcal{G}_{top} .

Given the context information:

{context}

Can you answer the following question based on the context information: '{question}'

If you could, please provide the answer of the '{question}'. If not, simply respond with 'NO!'.

Table 17: Self-verify to decide when to decompose the question in the internal node for \mathcal{G}_{top} .

Given the original question and context:

Original question:{original_question}

Context:{context}

Your task is to identify aspects of the original question that remain unresolved even with the provided context. Decompose these unresolved aspects into original question into two simple sub-questions that are entirely separated from the context. Decompose these unresolved aspects for original question into a few sub-questions that are entirely separated from the context. Each sub-question should challenge students to infer missing details or apply deeper understanding beyond the explicitly provided information. These sub-questions should be focused on the unresolved aspects of original question and designed to guide students towards constructing a comprehensive answer through structured reasoning.

Table 18: Decompose the leaf node question q into the sub-question to add a child graph.

Given the original question and relevant information:

Original question:{original_question}

Relevant information:{context} {qa_dialog}

Your task is to identify aspects of the original question that remain unresolved even with the relevant information. Decompose these unresolved aspects into original question into two sub-questions that are entirely separated from the relevant information. Each sub-question should challenge students to infer missing details or apply deeper understanding beyond the explicitly provided information. These sub-questions should be focused on the unresolved aspects of original question and designed to guide students towards constructing a comprehensive answer through structured reasoning.

Table 19: Decompose the internal node question q into the sub-question to add a child graph \mathcal{Q}_{child} .