Tree-of-Reasoning Question Decomposition for Complex Question Answering with Large Language Models

Kun Zhang[1,2*], **Jiali Zeng**[3], **Fandong Meng**[3], **Yuanzhuo Wang**[1,2,4†], **Shiqi Sun**[4], **Long Bai**[2], **Huawei Shen**[1], **Jie Zhou**[3]

[1]CAS Key Laboratory of AI Security, Institute of Computing Technology, Chinese Academy of Sciences
[2]School of Computer Science and Technology, University of Chinese Academy of Sciences
[3]Pattern Recognition Center, WeChat AI, Tencent Inc, China
[4]Big Data Academy, Zhongke
{zhangkun18z, wangyuanzhuo}@ict.ac.cn, lemonzeng@tencent.com

Abstract

Large language models (LLMs) have recently demonstrated remarkable performance across various Natual Language Processing tasks. In the field of multi-hop reasoning, the Chain-of-thought (CoT) prompt method has emerged as a paradigm, using curated stepwise reasoning demonstrations to enhance LLM's ability to reason and produce coherent rational pathways. To ensure the accuracy, reliability, and traceability of the generated answers, many studies have incorporated information retrieval (IR) to provide LLMs with external knowledge. However, existing CoT with IR methods decomposes questions into sub-questions based on a single compositionality type, which limits their effectiveness for questions involving multiple compositionality types. Additionally, these methods suffer from inefficient retrieval, as complex questions often contain abundant information, leading to the retrieval of irrelevant information inconsistent with the query's intent. In this work, we propose a novel question decomposition framework called TRQA for multi-hop question answering, which addresses these limitations. Our framework introduces a reasoning tree (RT) to represent the structure of complex questions. It consists of four components: the Reasoning Tree Constructor (RTC), the Question Generator (QG), the Retrieval and LLM Interaction Module (RAIL), and the Answer Aggregation Module (AAM). Specifically, the RTC predicts diverse sub-question structures to construct the reasoning tree, allowing a more comprehensive representation of complex questions. The QG generates sub-questions for leaf-node in the reasoning tree, and we explore two methods for QG: prompt-based and T5-based approaches. The IR module retrieves documents aligned with sub-questions, while the LLM formulates answers based on the retrieved information. Finally, the AAM aggregates answers along the reason tree, producing a definitive response from bottom to top. We evaluate our proposed framework on four benchmark datasets. The experimental results demonstrate that our proposed methods consistently outperform baseline methods outperform strong baselines by a substantial margin across all datasets.

Introduction

Recently, Large Language Models (LLMs) have exhibited an impressive capability for question-answering tasks, particularly in scenarios where resolving complex questions requires a multi-hop reasoning process (Touvron et al. 2023; OpenAI 2023). For example, Wei et al. (2022) proposes few-shot chain-of-thought (CoT) prompting, which enables LLMs to generate intermediate reasoning steps explicitly before predicting the final answer with a few manual step-by-step reasoning demonstration examples. However, the intricate nature of complex questions remains a significant challenge. To address this, Kojima et al. (2022) introduces zero-shot CoT, which eliminates the need for manually crafted examples in prompts by appending "Let's think step by step" to the target problem fed to LLMs. This simple prompting strategy surprisingly yields performance similar to few-shot CoT without requiring any manual clues. To mitigate the issue of generating incorrect information (i.e., hallucination) while retaining real-time knowledge, several studies have incorporated information retrieval (IR) techniques into LLM reasoning (Press et al. 2022; Khattab et al. 2022; Wang et al. 2023; Kandpal et al. 2023; Azamfirei, Kudchadkar, and Fackler 2023). This integration of IR significantly improves the quality of LLM-generated answers.

Although COT with IR methods has achieved great success in solving complex questions, they still face two challenges. **Challenge 1: Multi-compositional Types of Sub-questions.** The interrelations among the sub-questions of complex questions can be categorized into two distinct classes (Pan et al. 2020): 1) **nest-type**: the solution of the sub-question depends on the answers of the previous sub-question; 2) **branch-type**: the solution needs to summarize or compare the answers of other sub-questions. As shown in Figure, previous methods mainly consider single compositional type questions,i.e., nested-type questions through the formulation of a sequence of prompt strategies. However, they tend to disregard branch-type questions and multi-compositional type questions, resulting in diminished efficacy when tackling such questions. The hallucinations encountered during decomposition also exacerbate this disadvantage. **Challenge 2: Inefficient Retrieval.** Complex questions are often lengthy and contain abundant information,

*The work is done during internship at WeChat AI.
†Corresponding Author

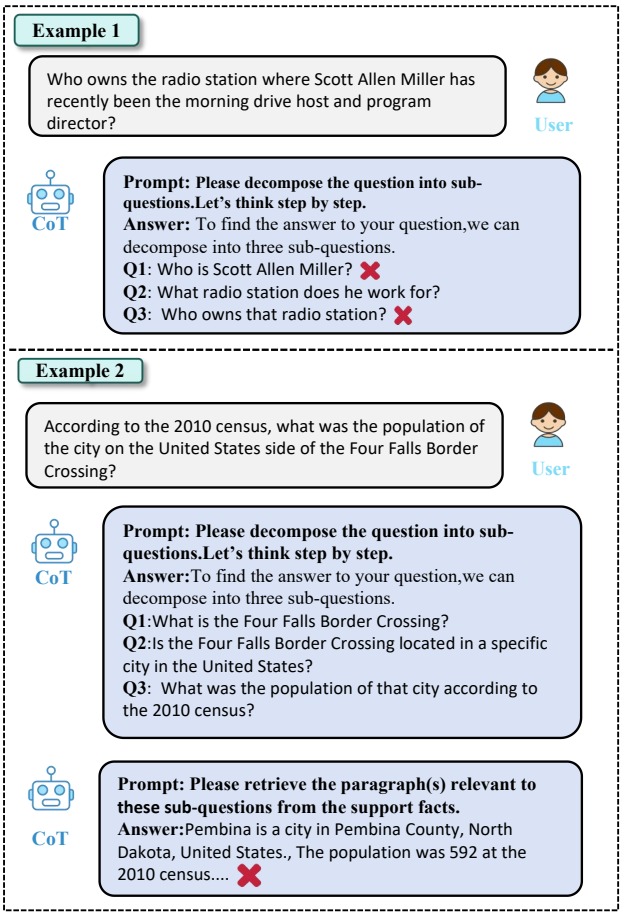

Figure 1: Example 1 and 2 are two bad instances in LLM with IR methods.

which can inadvertently result in the retrieval of irrelevant information inconsistent with the query's intent. When such inappropriate information is fed to the QA model, it introduces noise into the generated answer. Some efforts (Press et al. 2022) have been made to tackle this problem by integrating verification modules. However, it's notable that the increased frequency of application programming interface (API) calls leads to a rapid increase in time consumption.

To address the above challenges, we focus on improving the question decomposition process. We propose an innovative structure, named Reasoning Tree (RT), to better model the complex question's structure. The RT contains three types of nodes: root node, middle node, and leaf node. As shown in Figure 3, the root node is a question node that represents the whole question. The middle node represents the intermediate dependency parse subtree decomposed by the nest and branch relation, while the leaf node represents the inseparable question substructure.

To generate the reasoning tree and formulate the answer, we propose a novel framework called TRQA. This framework is a composite of four integral components, namely the Reasoning Tree Constructor (RTC), the Question Generator (QG), the Retrieval And LLM Interaction Module (RAIL), and the Answer Aggregation Module (AAM). To train RTC, we use T5 as the basis model and train it to predict the location of labels [NEST] and [BRANCH]. For each leaf node in the reasoning tree, QG is designed to translate the substructure into a complete sub-question. To realize QG, we explore two methods: prompt-based and T5-based. RAIL retrieves similar documents align with each sub-question, and LLM formulates answers by combining retrieval documents and questions. Finally, AAM aggregates answers along the reasoning tree to produce the definitive response from bottom to top.

In general, our main contributions are listed as follows:

- We propose a novel framework named TRQA, which leverages question decomposition to construct a global reasoning structure and generates reliable answers through interaction with LLM and IR.
- We design a novel approach, the structure-driven question decomposition model, which employs dependency parse trees to augment the process of reasoning structure generation.
- We verify the effectiveness of the proposed framework on four widely-used datasets and the experimental results show that our proposed methods consistently outperform baseline methods across all benchmarks by a large margin.

Preiliminary

In this section, we propose the reasoning tree(RT), a tree-based structure, to model the decomposition structure of complex questions. The detail of the definition is shown below.

Definition

Given a question Q, the RT of Q is a tree containing two types of nodes: the middle node and the leaf node. As shown in the example in the right part of Figure 2, the root question node indicates the original question, whose answer is constrained by the middle node M1. Note that M1 contains a placeholder "#1", indicating the answer of the subtree M2. Meanwhile, the middle node M2 points to two leaf nodes, L1 and L2. The structure of RT enables it to represent the combinations of multiple compositionality types. We also provide an equivalent linear representation of RT under the tree illustration by introducing two separators, i.e., [NEST], and [BRANCH]. [NEST] means that the placeholder of the current question is the answer of the subtree. [BRANCH] means that the placeholder of the current question needs to conduct some function to aggregate the answers of each subtree, including intersection, union, comparison, etc.

Methodology

As shown in Figure 2, we present the design of our framework, denoted as TRQA, which comprises four main components: reasoning tree constructor, question generator, retrieval and interaction with LLM, and answer aggregation module. RTC decomposes a question into the reasoning

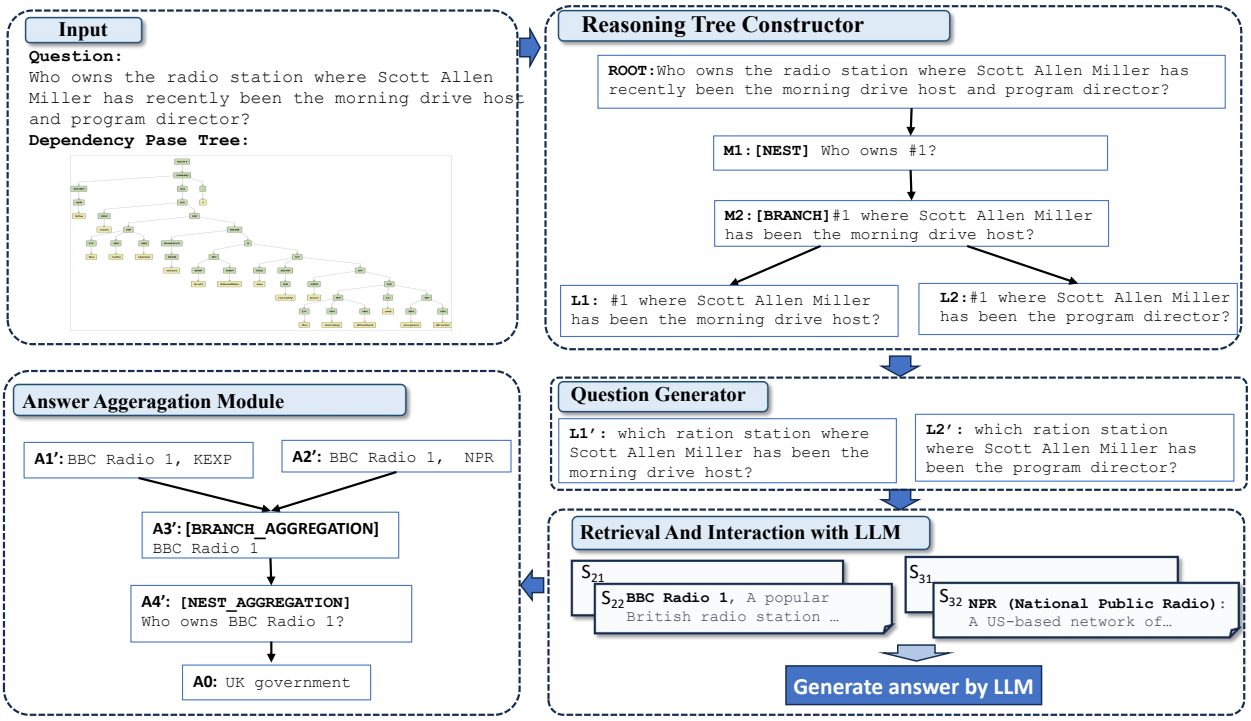

Figure 2: The overall architecture of our framework TRQA.

tree devised by a dependency parse tree. QG translates the leaf-node substructure of the reasoning tree into a question. RAIL retrieval similar documents and generate the answer. AAM aggregates answers following the reasoning Tree.

Reasoning Tree Constructor

As mentioned above, there are mainly two types of relations between sub-questions, i.e., nest and branch, which is also the basis relation in the reasoning tree. The dependency parse tree reveals the dependency relation between tokens in the relation, which can assist us to predict the relation between different sub-structures. Thus, we introduce the dependency parse tree to decompose the question and generate the reasoning tree. We collect some labeled data according to the dependency parse tree and train it with T5.

Data Collection To train a model for constructing the reasoning tree, we construct a dataset called RTrees, with 7,00 samples from existing muti-hop question-answering datasets. During the annotation process, we ask annotators to insert [NEST] and [BRANCH] into the dependency parse tree, divide them into different subtrees, and then use a depth-first search algorithm to flatten them into sequences. We take some examples in the appendix file.

Training We used T5 as our basic model and trained a model to predict the location of separators. After obtaining the position of the separator, we divide the dependency parse tree into different substructures based on its position in the tree. Finally

Question Generator After obtaining the inference tree, we rewrite the substructure into a natural language question

Algorithm 1: Processing a Complex Question

Question: q; Dependency Parse Tree: dpt;
procedure PROCESS(complex_question)
  $RT \leftarrow$ RTC.ConstructTree(q, dpt)
  leaf_substructures $\leftarrow$ RTC.GetLeafNode(RT)
  for each substructure **in** leaf_substructures **do**
    question $\leftarrow$ QG.GenerateQuestion(substructure)
    similar_documents $\leftarrow$ RAIL.Retrieve(question)
    answer $\leftarrow$ RAIL.GenerateAnswer(similar_documents)
    reasoning_tree.PlaceAnswer(substructure, answer)
  end for
  aggregated_answers $\leftarrow$ AAM(reasoning_tree)
  return aggregated_answers
end procedure

based on the substructures of the leaf nodes, including questions and corresponding analysis tree subtrees. Given the dependency sub-tree C_i, and part of question S_i, we design two methods to translate the C_i into corresponding question q_i.

Prompt-based Question Generator We have designed a prompt to rewrite the dependency analysis tree subtree and corresponding question section into a natural language question, which is shown as follows,

"This is a question Q, and there is a part of question. Its dependency parse is C_i, and the corresponding token in question is S_i, please convert it into a complete question"

In this way, we can transform the substructure into a question.

T5-based Question Generator We collect a few human-labeled datasets and train a T5-based model to generate the question. The input is the dependency parse subtree and part-of-question and the output is the complete question. The detail is shown below.

Data Collection To train a model for question generation, we construct a dataset called D2Q, with 4,00 samples. During the annotation process, we ask annotators to write the question according to the substructure. We take some examples in the appendix file.

Training We take the linearized dependency parse subtree and corresponding question sequence as input and the annotated question as output.

$$q_i = \text{T5}(C_i) \tag{1}$$

where q_i is denoted as the ith sub question.

Retrieval And Interactive with LLM

In the context of each individual question, the operational workflow encompasses the relay of generated sub-questions from the Large Language Model (LLM) to the Information Retrieval (IR) system. In this component, the IR mechanism assumes a dual function encompassing confirmation and augmentation. Concretely, for each sub-question q_i, the IR module assumes the role of verification, furnishing pertinent and complementary data. This serves not only to corroborate the sub-question but also to enrich its content. Then the retrieval information incorporated with the question is subsequently fed back to the LLM, effectively facilitating the production of accurate and reliable sub-answers.

Furthermore, this interaction can contribute to the completeness of the content generation process within LLM. Owing to the correlation between IR and LLM, the former evolves into a repository of collated documents, meticulously documenting the records acquired from every node within the reasoning tree architecture. This approach heightens both the traceability and reliability of the content generated from LLM. Through the retrieval interaction between each sub-question and the IR system, the most pertinent document for each sub-question, denoted as d_i, can be sourced as the supporting document for the corresponding sub-question q_i.

Answers Aggregation Module In AAM, we aggregate the answers of leaf nodes along the reasoning tree from bottom to top. When the answer is aggregated from the child node to the parent node, we designed two aggregation functions, i.e. NEST_AGGREGATION and BRANCH_AGGREGATION. They correspond to nest and branch tags in the reasoning tree. We will introduce these two operations in detail below.

NEST_AGGREGATION We use the answer of the child node of this node directly as the placeholder. For example in Figure 2, in the answer aggregation module, the answer of A3' is directly used to replace the value of placeholder in M1. Then we use QG to translate the current node substructure into a question. RAIL is used to generate the answer of the current node.

BRANCH_AGGREGATION We collect answers from different child nodes and replace them with the position in the current substructure. We design a prompt and directly use LLM to generate the answer. The prompt is shown below,

Prompt: *We have several answers, I want to get the final answer from "Answer_1 [operation] Answer_2 [operation]"*, [operation] represents aggregation functions, like intersection, cooperation, etc. In this way, we get the final answer for the current node.

We follow the reasoning tree from bottom to top and gradually aggregate the answers based on these two aggregation methods to obtain the final answer

Experiment

In this section, we present a comprehensive evaluation of our proposed framework, TRQA, using four extensively utilized benchmark datasets. The obtained results validate the effectiveness of our method.

Datasets and Preprocessing

We select four complex multi-hop question-answering datasets, HotpotQA(Yang et al. 2018), HyBridQA(Chen et al. 2020), Musique(Trivedi et al. 2022), and WikiMultiHopQA (WMHQA)(Ho et al. 2020). The details are shown below.

HotpotQA The HotpotQA dataset stands as a pivotal benchmark within the field of Natural Language Processing (NLP), designed to evaluate models' capabilities in reasoning and multi-hop question-answering tasks. It boasts an impressive repository of over 110,000 diverse question-answer pairs, meticulously crafted to span a wide spectrum of domains and topics. We focus on its full wiki setting.

HyBridQA is an extensive question-answering dataset, integrating structured Wikipedia tables and related free-form text corpora, designed to necessitate reasoning over both forms of information. It includes around 70,000 question-answer pairs aligned with 13,000 unique Wikipedia tables.

Musique is a challenging multi-hop question answering dataset comprising 25,000 questions with 2-4 hops, developed via a systematic bottom-up approach of selecting interconnected single-hop questions.

WikiMultiHopQA is an extensive and high-quality multi-hop question-answering dataset created utilizing Wikipedia and Wikidata, with an emphasis on providing exhaustive explanations from question to answer that enrich the understanding of predictions. The dataset introduces natural questions that demand multi-hop reasoning, crafted using logical rules embedded within the knowledge base (KB).

Baselines

The baseline models can be categorized into two classes. The first class focuses on designing prompts to improve the reasoning ability of LLM (CoT (Wei et al. 2022), Auto-CoT(Zhang et al. 2022b), Recite-and-answer(Sun et al. 2022), and Least-to-Most(Zhou et al. 2022)). And the second class introduces IR to LLM (Self-Ask(Press et al. 2022), Plan-And-Solve(Wang et al. 2023), React(Yao et al. 2022)

Approach	HotpotQA		HyBridQA		Musique		WMHQA	
	EM	**Recall**	**EM**	**Recall**	**EM**	**Recall**	**EM**	**Recall**
Direct prompting	25.4	36.2	12.8	19.2	6.0	8.2	25.8	28.4
AUTO-COT	36.8	45.8	18.2	25.6	10.6	13.2	29.2	32.2
COT	38.2	48.6	16.2	21.8	9.4	11.0	30.4	34.2
Recite-and-answer	36.6	38.8	16.6	19.8	11.0	13.6	32.6	36.2
Self-Ask w/o IR	34.4	36.2	17.4	22.2	11.2	14.4	35.8	39.6
Least-to-Most	34.2	38.6	26.4	32.6	11.6	13.4	32.8	36.6
Plan-And-Solve	37.4	41.6	27.8	30.2	13.4	16.8	34.8	37.8
Least-to-Most w/ IR	42.6	44.2	30.2	35.4	15.2	18.2	32.8	36.6
Self-Ask	40.4	49.8	24.4	30.2	14.4	15.6	39.6	42.6
Plan-And-Solve w/ IR	41.6	45.6	29.2	33.6	15.0	17.8	42.4	46.2
React	44.6	48.2	32.6	35.6	15.6	18.4	40.4	43.6
DSP	53.0	56.8	33.2	34.8	16.2	20.8	43.2	46.8
TRQA	**61.2**	**62.4**	**35.2**	**42.4**	**24.2**	**26.8**	**52.6**	**54.8**
TRQA w/o IR	55.4	60.2	28.6	37.2	18.6	20.4	44.8	46.8
TRQA w/o AAM	59.2	69.2	31.8	38.4	22.4	25.2	49.2	50.6
TRQA w/ PQG	60.3	66.8	33.2	39.6	20.2	24.6	48.2	50.2

Table 1: EM and Recall results on four benchmark datasets (%).

Approach	HotpotQA		HyBridQA		Musique	
	EM	**Recall**	**EM**	**Recall**	**EM**	**Recall**
TRQA	64.2	72.2	35.2	42.4	24.2	26.8
TRQA w/ cluenet	58.2	67.4	27.8	36.4	17.8	18.8
TRQA w/ HSP	56.8	66.2	26.2	34.8	16.4	19.2
TRQA w/ DecompRC	52.6	64.8	24.4	30.2	14.2	17.2

Table 2: EM and Recall results on HotpotQA, HyBridQA adn Musique dataset (%).

and DSP (Khattab et al. 2022)), aims at retaining real-time knowledge and alleviating hallucination in the generated answer.

Metrics

We employ two key metrics to evaluate the performance of our large model's generated responses: Cover-EM (EM) and Recall. The Cover-EM metric evaluates the percentage of correct answers that appear as substrings within the model's generated responses. This metric effectively measures the model's ability to generate comprehensive answers by determining if the correct answer is present in its output. On the other hand, Recall is a commonly used metric to evaluate the model's ability to retrieve relevant information. Specifically, it quantifies the ratio of correctly identified relevant items by the model to the total number of relevant items.

Human Annotation

We invite four graduate students, who majored in Computer Science and are familiar with natural language processing, to annotate the dataset. Before annotation, they are informed of the detailed instructions with clear examples. For the RTC task, the inter-annotator agreement score is 0.87, and for the QG task, the inter-annotator agreement score is 0.76.

Implementation Details

We harnessed the gpt-3.5-turbo, a prominent large language model accessible through OpenAI's API, for our experimentation. Additionally, we employed ColBERTv2 as our retrieval model, following (Xu et al. 2023). Our baselines, which incorporate information retrieval, were subjected to identical experimental settings as our proposed framework. Given that the majority of baseline methods were tested using the text-davinci-002 model, we reenacted their experiments on the gpt-3.5-turbo model, adhering to the configurations outlined in their respective publications, consequently yielding enhanced performance.

Main Results

The performance of and baselines on the multi-hop question-answering task are shown in Table 1. We compare it with recent competitive baselines in the setting without IR. TRQA w/o IR outperforms all baselines based on CoT methods, including CoT, Auto-CoT, CoT-SC, and Recite-and-answer, which indicates that constructing the reasoning tree, i.e., a global reasoning process(Xu et al. 2023), is better than just giving intermediate reasoning results. The results of Self-Ask, Plan-And-Solve w/IR, React, and DSP outperform CoT methods, which indicates that IR can introduce external knowledge to assist reasoning and answer complex

	HyBridQA	HotpotQA	Musique
Self-Ask	4.28	4.32	5.68
DSP	4.02	4.26	5.36
Least-to-Most	3.66	3.88	4.26
TRQA	**2.68**	**3.22**	**3.42**

Table 3: Efficiency analysis.

questions. Besides, TRQA w/o IR outperforms Self-Ask w/o IR and Plan And Solve, which indicates that it is more effective to construct the reasoning tree and generate answers along it than generate an inference chain and answer subquestions step by step. TRQA outperforms TRQA w/ PQG, which shows that our T5-based question generator is more effective in translating substructure into the complete question than LLM-based.

Ablation Study To evaluate the effectiveness of each model module, we perform ablation studies where we remove the IR module and AAM. The results are shown in Table **??**.

After removing each module, the performance of the model deteriorated. In Table **??**, the results of (TRQA w/o IR) denote the model that removes the IR module. This indicates that IR interacting with each node of the reasoning tree ensures the reliability and accuracy of LLM-generated answers. The results of (TRQA w/o AAM) denote the model that removes the AAM module. We directly use a prompt to combine answers from leaf nodes and generate the final answer by LLM. The result shows the effectiveness of AAM, which can aggregate the answers in a reasonable way.

The Effectiveness of Different QDT Method To demonstrate the effectiveness of our question decomposition method, we perform an ablation study by removing QDT and replacing it with different question decomposition methods. We choose cluenet(Huang et al. 2023), HSP(Zhang et al. 2019), and DecompRC(Min et al. 2019) as the contrastive question decomposition model. Results show that QDT is superior to other methods on EM. This suggests that our proposed QDT can further promote the generative method. Besides, we also find that none of the examples become worse after the incorporation of IR, which means IR is stable and safe as an external module for the generative decomposition method.

Efficiency Analysis We further analyze the difference in running efficiency between TRQA and baselines from the perspective of the average number of interactions between IR and LLM per question. Table **??** shows the results of the analysis on four multi-hop QA datasets. Based on Table **??** and Table **??**, TRQA not only exhibits superior task performance but also has the least number of interactions with API, consequently entailing the least time cost.

Case Study: TRQA vs New Bing in Tracing Following (Xu et al. 2023), we conduct a comparative analysis between the performance of TRQA and New Bing in the task of attributing references to generated content, as illustrated in our case study (Figure 3). Notably, TRQA shows a finer-grained ability to attribute references to each segment of knowledge engaged in the reasoning process, which corresponds to each correct node within the TRQA framework. In contrast, the referencing provided by New Bing exhibits gaps, failing to encompass the entirety of the relevant knowledge. There are also some instances where New Bing is unable to locate certain knowledge segments.

Related Work

Chain-of-Thought Prompting (Wei et al. 2022) introduces a method termed Chain-of-Thought (CoT), aiming to enhance the reasoning capabilities of large language models (LLMs). CoT employs a few-shot paradigm to enable LLMs to generate intermediate reasoning outcomes during the solution of intricate problems, thereby ameliorating their reasoning aptitude. By utilizing the guiding prompt "Let's do it step by step," CoT achieves promising zero-shot performance. A derivative of this approach, Auto-CoT, leverages language models to automatically craft few-shot learning examples for the CoT framework (Wei et al. 2022). Various studies have delved into diverse facets of CoT, encompassing concerns like self-consistency (Wang et al. 2022), utilization of moderate-sized models (Zelikman et al. 2022), and selection (Fu et al. 2022). Additionally, certain methodologies iteratively employ LLMs to break down intricate queries into manageable sub-questions, systematically addressing them. Notable instances include Least-to-Most (Drozdov et al. 2022), Dynamic Least-to-Most, Self-Ask (Press et al. 2022), and DSP (Khattab et al. 2022). This body of work serves as a catalyst for our proposed approach, namely the Tree of Reasoning. Preceding studies have predominantly concentrated on furnishing intermediate reasoning outputs, segmenting intricate queries, and stepwise resolution of sub-questions. These efforts largely focus on local sub-questions resolution, without adequate attention to the planning of a global reasoning chain. Approaches like AgentGPT and PS initially formulate sub-questions and subsequently address them, revealing their limitations when subsequent sub-question solutions hinge on insights garnered from preceding ones. Such a scenario commonly arises in knowledge-intensive tasks like multi-hop question-answering (Wang et al. 2023; Zhang et al. 2023b,a).

Question Decomposition Question decomposition plays a pivotal role in generating an ungrounded query graph, serving as a valuable aid in structure disambiguation. Such structural clarity aids in steering the QA system away from inefficient path traversal within relations (Chen et al. 2021). The realm of question decomposition encompasses three primary methodologies:

(1) Splitting-based Methods: Strategies like SplitQA (Talmor and Berant 2018) and DecompRC (Min et al. 2019) harness pointer networks to bifurcate a question into two segments. While effectively preserving the core essence of the original inquiries, these approaches exhibit a need for increased flexibility when dealing with intricate structures.

(2) Generative Methods(Zhang et al. 2019, 2022a): These methodologies introduce heightened adaptability and expansibility, amenable to a wide range of target formats.

Method \ Question	The band that released Women and Children First had how many number ones ?	Which country contains Conran in the residence state of the author of Little Gidding ?	What group of languages includes the old version of the language that the Quran was first translated in ?
TRQA	The band that released Women and Children First is Van Halen[1].Van Halen had thirteen number ones on the Billboard Mainstream Rock chart.[2] [1] Women and Children First is the third studio album by American hard rock band Van Halen, released on March 26, 1980 on Warner Bros.... [2] As of 2007, Van Halen has sold 75 million albums worldwide and have had thirteen No. 1 hits on the Billboard Mainstream Rock chart.	The author of Little Gidding is T.S. Eliot [1]. Eliot was born in St. Louis, Missouri, United States. However, he later moved to England and became a British citizen [2].Conran is located about ten miles southwest of New Madrid on U.S. the country that contains Conran in the residence state of T.S. Eliot is the United States. [1] Little Gidding is the fourth and final poem of T. S. Eliot's Four Quartets, a series of poems that discuss time, perspective, humanity, and salvation... [2] Conran is an unincorporated community in eastern New Madrid County, Missouri, United States.It is located approximately ten miles southwest of New Madrid on U.S. Route 61.	The specific language that the Quran was first translated into is Persian [1]. The group of languages that includes the old version of the language that the Quran was first translated in is the Western Iranian languages [2]. [1] The first fully attested complete translations of the Quran were done between the 10th and 12th centuries in Persian. ... [2] The Western Iranian languages are a branch of the Iranian languages, attested from the time of Old Persian (6th century BC) and Median. ....
New Bing	The band that released the album "Women and Children First" is Van Halen. According to my sources, Van Halen had one number one hit on the Billboard Hot 100 chart with their song "Jump" in 1984 [1]. Is there anything else you would like to know? [1] Eddie Van Halen rode the top of the Billboard charts alongside his namesake band Van Halen, scoring five No. 1 albums on the Billboard 200 chart, and more than 20 hits on the Billboard Hot 100 songs...	I'm sorry but I couldn't find any information about Conran. However, Little Gidding is a poem by T.S. Eliot. It is the fourth and final poem of his Four Quartets series [1]. The poem refers to a small Anglican community in Little Gidding in Huntingdonshire, England [2]. I hope this helps. [1] Little Gidding is the fourth and final poem of T. S. Eliot's Four Quartets, a series of poems that discuss time, perspective, humanity, and salvation... [2] The title refers to a small Anglican community in Little Gidding in Huntingdonshire, established by Nicholas Ferrar	The first complete translation of the Quran was into Persian, which is a member of the Iranian branch of the Indo-Iranian language family [1]. The first chapter of the Quran, Al-Fatiha, was translated from Arabic to Persian by Salman the Persian in the 7th-10th centuries [2]. Is there anything else you would like to know? [1] Persian, also known by its endonym Farsi, is a Western Iranian language belonging to the Iranian branch of the Indo-Iranian subdivision of the Indo-European languages. [2] Salman the Persian translated the first chapter of the Qur'an, Al-Fatiha, from Arabic to Persian ....

Figure 3: Case study of TRQA and New Bing in marking references to supporting documents.

HSP (Zhang et al. 2019) employs a Seq2Seq model integrated with a copy mechanism to engender sub-questions. However, these approaches remain confined to decomposing questions into two segments, thereby failing to assure the unchanged retention of sentence semantics. Consequently, a potential exists for token loss or unexpected token generation, which in turn could compromise the semantic coherence of the input question, complicating performance evaluation.

(3) Rule-based Methods: Notably, EDG (Hu et al. 2021) exemplifies this category. EDG undertakes an iterative transformation of the constituency tree into an entity-centric graph, propelled by meticulously designed rules. While proficient in handling multiple forms of compositional diversity, this approach tends to rely heavily on constituency parsing, thus potentially limiting its coverage.

Conclusion

In this study, our focus centers on the utilization of question decomposition as a potent mechanism for addressing the challenge of answering complex questions through Large Language Models (LLMs). To achieve this goal, we introduce a novel structure, named the Reasoning Tree (RT), which represents the global reasoning structure of the question reasoning process. To construct the reasoning tree, we propose a novel framework named TRQA, which leverages question decomposition to construct the reasoning tree and generates reliable answers through interaction with LLM and IR. We design a novel approach, the structure-driven question decomposition model, which employs dependency parse trees to augment the process of reasoning structure generation. We verify the effectiveness of the proposed framework on four widely-used datasets and the experimental results show that our proposed methods consistently outperform baseline methods across all benchmarks by a large margin.

Acknowledgements

Thanks to reviewers for their helpful comments on this paper. This paper is funded by the National Natural Science Foundation of China (No.62172393, U1836206, and U21B2046), Zhongyuanyingcai program-funded to central plains science and technology innovation leading talent program (No.204200510002), Major Public Welfare Project of Henan Province (No.201300311200).

References

Azamfirei, R.; Kudchadkar, S. R.; and Fackler, J. 2023. Large language models and the perils of their hallucinations. *Critical Care*, 27(1): 1–2.

Chen, W.; Zha, H.; Chen, Z.; Xiong, W.; Wang, H.; and Wang, W. 2020. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. *arXiv preprint arXiv:2004.07347*.

Chen, Y.; Li, H.; Hua, Y.; and Qi, G. 2021. Formal query building with query structure prediction for complex question answering over knowledge base. *arXiv preprint arXiv:2109.03614*.

Drozdov, A.; Schärli, N.; Akyürek, E.; Scales, N.; Song, X.; Chen, X.; Bousquet, O.; and Zhou, D. 2022. Compositional semantic parsing with large language models. *arXiv preprint arXiv:2209.15003*.

Fu, Y.; Peng, H.; Sabharwal, A.; Clark, P.; and Khot, T. 2022. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*.

Ho, X.; Nguyen, A.-K. D.; Sugawara, S.; and Aizawa, A. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Hu, X.; Shu, Y.; Huang, X.; and Qu, Y. 2021. EDG-Based Question Decomposition for Complex Question Answering over Knowledge Bases. In Hotho, A.; Blomqvist, E.; Dietze, S.; Fokoue, A.; Ding, Y.; Barnaghi, P.; Haller, A.; Dragoni, M.; and Alani, H., eds., *The Semantic Web – ISWC 2021*, 128–145. Cham: Springer International Publishing. ISBN 978-3-030-88361-4.

Huang, X.; Cheng, S.; Shu, Y.; Bao, Y.; and Qu, Y. 2023. Question Decomposition Tree for Answering Complex Questions over Knowledge Bases. *arXiv preprint arXiv:2306.07597*.

Kandpal, N.; Deng, H.; Roberts, A.; Wallace, E.; and Raffel, C. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, 15696–15707. PMLR.

Khattab, O.; Santhanam, K.; Li, X. L.; Hall, D.; Liang, P.; Potts, C.; and Zaharia, M. 2022. Demonstrate-Search-Predict: Composing retrieval and language models for knowledge-intensive NLP. *arXiv preprint arXiv:2212.14024*.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213.

Min, S.; Zhong, V.; Zettlemoyer, L.; and Hajishirzi, H. 2019. Multi-hop reading comprehension through question decomposition and rescoring. *arXiv preprint arXiv:1906.02916*.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Pan, L.; Chen, W.; Xiong, W.; Kan, M.-Y.; and Wang, W. Y. 2020. Unsupervised multi-hop question answering by question generation. *arXiv preprint arXiv:2010.12623*.

Press, O.; Zhang, M.; Min, S.; Schmidt, L.; Smith, N. A.; and Lewis, M. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Sun, Z.; Wang, X.; Tay, Y.; Yang, Y.; and Zhou, D. 2022. Recitation-augmented language models. *arXiv preprint arXiv:2210.01296*.

Talmor, A.; and Berant, J. 2018. The web as a knowledge-base for answering complex questions. *arXiv preprint arXiv:1803.06643*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Trivedi, H.; Balasubramanian, N.; Khot, T.; and Sabharwal, A. 2022. MuSiQue: Multihop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, 10: 539–554.

Wang, L.; Xu, W.; Lan, Y.; Hu, Z.; Lan, Y.; Lee, R. K.-W.; and Lim, E.-P. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837.

Xu, S.; Pang, L.; Shen, H.; Cheng, X.; and Chua, T.-s. 2023. Search-in-the-Chain: Towards the Accurate, Credible and Traceable Content Generation for Complex Knowledge-intensive Tasks. *arXiv preprint arXiv:2304.14732*.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Zelikman, E.; Wu, Y.; Mu, J.; and Goodman, N. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35: 15476–15488.

Zhang, H.; Cai, J.; Xu, J.; and Wang, J. 2019. Complex Question Decomposition for Semantic Parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4477–4486. Florence, Italy: Association for Computational Linguistics.

Zhang, K.; Chen, C.; Wang, Y.; Tian, Q.; and Bai, L. 2023a. CFGL-LCR: A Counterfactual Graph Learning Framework for Legal Case Retrieval. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3332–3341.

Zhang, K.; Lin, X.; Wang, Y.; Zhang, X.; Sun, F.; Jianhe, C.; Tan, H.; Jiang, X.; and Shen, H. 2023b. ReFSQL: A Retrieval-Augmentation Framework for Text-to-SQL Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 664–673.

Zhang, K.; Qiu, Y.; Wang, Y.; Bai, L.; Li, W.; Jiang, X.; Shen, H.; and Cheng, X. 2022a. Meta-CQG: A Meta-Learning Framework for Complex Question Generation over Knowledge Bases. In *Proceedings of the 29th International Conference on Computational Linguistics*, 6105–6114.

Zhang, Z.; Zhang, A.; Li, M.; and Smola, A. 2022b. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q.; et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.