# Revolutionizing Multi-Hop QA: A Finite State Prompt Paradigm for Enhanced Language Model Reasoning

**Anonymous ACL submission**

## Abstract

Multi-hop Question Answering (MHQA) has long captivated the attention of researchers due to its ability to test a model's proficiency in handling complex questions. While Large Language Models (LLMs) have demonstrated impressive language comprehension abilities, their application to MHQA tasks is hindered by several challenges, including hallucination, error propagation and attention dispersion. We propose a novel prompt method Finite state automaton (FSM) to enhance the reasoning capabilities of LLM for complex task in addition to improved effectiveness and trustworthiness. Unlike traditional end-to-end chain-of-thought (COT) methods, FSM addresses MHQA by iteratively decompose question into multi-turn subquestions, enabling error checks and backtracking when necessary, ensuring the standardization of the process. Extensive experiments on benchmarks show that our proposed FSM not only has a neat format and stable performance, but also nearly doubles the zero-shot performance in hardest dataset Musique. Further analysis reveals the remarkable capability of FSM to elicit direct and concise intermediate reasoning steps. We believe that our framework can fully demonstrate its advantages when applied to challenging tasks.

## 1 Introduction

Multi-hop Question Answering (MHQA) has long captivated the attention of researchers due to its ability to test a model's proficiency in handling complex questions. Although notable advancements have been made in single-hop question answering, MHQA remains a complex task due to the necessity of integrating knowledge from disparate sources and executing multiple inferential steps.

The field of MHQA currently adopts two dominant strategies. The initial strategy fine-tunes large models using domain-specific data, a notably complex process (Cao et al., 2023) that demands extensive data and intricately designed instruction formats. The alternative strategy, In-Context Learning (ICL) (Wang et al., 2023), instructs the model to address problems using human-devised techniques, typically by demonstrating problem decomposition through multiple examples.

By manually analyzing the errors of popular methods, we have summarized four types of errors, which we will further discuss in the section 5. Specific examples by common methods can be seen in Figure 3. On the one hand, we observed that LLMs are particularly error-prone in the experiments; inaccuracies in initial reasoning steps can propagate, culminating in erroneous conclusions. on the other hand, Apart from being inefficient and difficult to extend, the few-shot methods like REACT (Yao et al., 2022) and SP-COT (Wang et al., 2023)equipped at least 4-shot demostrations can dilute the LLM's focus on the context segment. This dispersion of attention hampers the model's ability to locate answers correctly and causes hallucinations.

We suppose that the current methods like COT, an end-to-end inference output in one time method, is too hard to LLMs. Therefore, we dissect the end-to-end internal reasoning into simpler steps, each involving one round of reasoning. In an automaton-like format, the model addresses one task at a time and decides on the next step based on its current result and state. In this way, model will strictly follow the prescribed process. Decomposing subproblems in iteration instead of at once. Besides, we extract the titles of supporting facts, allowing the model to search answer in context without distraction. Check step promptly and backtrack if necessary. This novel prompt-based paradigm simplifies complex tasks by decomposing them into smaller, manageable multi-turn sequences, enhancing performance on intricate tasks such as MHQA and NL2SQL.

Our experiments conducted on MHQA benchmark (Yang et al., 2018; Trivedi et al., 2022; Ho

et al., 2020) show that our results surpass the baseline. Moreover, as LLMs are generative and their output is hard to control, all other baselines, except for our framework, exhibit issues with disorganized outputs and formats that do not conform to the correct answer, necessitating further processing.

Our approach advances the theoretical understanding and practical application of multi-hop reasoning by developing:

1) A novel prompt-based paradigm that simplifies complex MHQA tasks into multi-turn sequences, ensuring each step is accurately completed before proceeding—this method effectively mitigates the prevalent issue of error propagation in common Chain of Thought (COT) reasoning.

2) A dynamic reasoning architecture that iteratively solves MHQA tasks, with a robust checking mechanism at each step and the ability to trace back through the reasoning process, ensuring the integrity and correctness of the answer.

3)A refinement step that utilizes previously referred information to further address and correct hallucinations—false answers generated with erroneous reasoning steps—thereby enhancing the reliability of the model's output.

4)Extensive experimental validation of our FSM framework across different models, establishing its efficacy and setting a new state-of-the-art for MHQA performance.

## 2 Related Work

1

**Multi-hop Question Answering** Existing approaches to solving the multi-hop QA task can be mainly categorized into question decomposition (Perez et al., 2020a; Fu et al., 2021; Perez et al., 2020b), graph-based method (Tu et al., 2019; Thayaparan et al., 2019; Fang et al., 2020), iterative method (Qi et al., 2019) and LLMs (Wang et al., 2023) prompts. These models grapple with computational complexity and extensibility, and they lack an interpretable reasoning chain, which deviates from human cognitive processes.

**Language model for reasoning.** Perhaps the most well-known work of using LLMs for reasoning is Chain-of-Thought (CoT)(Wei et al., 2022), which reveals the ability of LLMs to formulate their own "thinking procedure" for problem-solving. Several follow-up works have since been performed, including least-to-most prompting (Zhou et al., 2022) for solving complicated tasks, zero-shot CoT (Kojima et al.), graph-of-thoughtGoT(Besta et al., 2023) and reasoning with self-consistency (Wang et al., 2022). ReAct (Yao et al., 2022) interleaves the generation of reasoning traces with task-specific actions, promoting greater synergy.

**Task decomposition.** (Perez et al., 2020a) decompose a multi-hop question into a number of independent single-hop subquestions, which are answered by an off-the-shelf question answering (QA) model. Then those answers are aggregated to form the final answer. Both question decomposition and answer aggregation require training models. After the emergence in LLMs, training methods (Cao et al., 2023) are rarely used for its expensive cost. Most research is on the few-shot icl approach. (Zhou et al., 2022) chains the processes of problem decomposition and sub problem solving. The original problem and its sub problems are inherently interrelated, and forcibly breaking them down into unrelated problems would increase the difficulty in vain.

## 3 Methodology

### 3.1 Strategies

we deduce from the results of baseline methods that the model does not yet have the ability to complete complex instructions at once. LLMS will forget the previous inference during the process of reasoning. Therefore, we design the following strategies to alleviate these issues: **a)Iterative Decomposition**: FSM eschews the traditional few-shot reasoning approach in favour of a multi-turn process. Each iteration focuses on addressing a single sub-question, ensuring that LLMs can clearly understand instructions and execute them strictly. **b)Error Checking and Backtracking**: After each reasoning step, FSM performs a check to ensure the accuracy of the response. If an irregular or wrong output is detected, the model will be given a chance to self revise the answer or backtrack. **c)Final Review Step**: To avoid distractions from long contexts, we only use subquestions, corresponding supporting factual evidence and answers without distractors to check the consistency of answers and reasoning furtherly.

### 3.2 Framework

We elaborate our proposed FSM in two stages 1: In the first stage, we instruct LLMs to answer one hop subquestion in one turn. In the stage 2, LLMs

**

Figure 1: The abstract framework of our proposed FSM. First, the complex question flows into the Decomposer, where the question is broken down into the first single-hop question. Then, moving on to the Searcher, given all paragraphs, it is required to answer the sub-questions. Third, the model checks the output and processes it into the correct format. The tracker updates various states and information subsequently. Loop this process until the final answer is found. Finally, we prompt LLms to review and summarize the key information found previously and give a more precise answer.

**

Figure 2: The flow chart of our FSM and a case in detail.

is required to revise answer with the key information sub questions, answers, and relevant paragraph found in the previous stage.

We display the FSM framework in Figure 2 Specifically, we first instruct the model to decompose the initial question. Next, we compare the parent and sub-questions for semantic equivalence; if they differ, the model backtracks to re-decompose. In the third step, the model searches the context for relevant paragraphs, extracting titles and answers. The fourth step involves updating the complex question with the answered sub-question and determining the type of relationship—composition or comparison. Further checks are made to determine if the response is a simple or compound sentence, with compound sentences being promptly decomposed. The iteration continues until the sub-question cannot be further decomposed. By enforcing strict adherence to each step, our method can more precisely assess the genuine capabilities of a model, as opposed to other methods that often skip essential intermediate stages, leading to seemingly correct results despite flawed reasoning processes.

# 4 Experiments

## 4.1 Benchmark and Evaluation

We evaluate our model on three high-quality multi-hop question answering datasets: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020) and Musiqus (Trivedi et al., 2022). Drawing lessons (Min et al., 2019) from the possibility of single hop problems in HotpotQA, Musiqus is even more difficult due to its more complex question and the fact that it has more candidate paragraphs. We tackle the task with instructions from two different settings: (1) directly generate the answer from the context and question; (2) generate the whole reasoning chain with answer, supporting facts and evidence. to study the faithfulness of reasoning chain. Following traditions, We adopt the exact match (EM) and F1 scores as evaluation metrics. And for setting with supporting fact, we

Our framework has strict formatting during the inference process, so scores gained by the official evaluation scripts, without additional processed separately.

## 4.2 Baseline

(a)**Standard prompting**(Standard) is the most basic form of prompting, involving only task descriptions and output requirements, with no explicit instructions for reasoning. This method serves as a control to assess the inherent capabilities of LLMs without additional guidance.

(b)**COT** (Wei et al., 2022) has gained popularity in LLMs inference due to its simplicity and effectiveness. It encourages LLMs to generate intermediate step-by-step rationales, which facilitate the reasoning process necessary to arrive at an answer.

(c)**SP-COT** (Wang et al., 2023) introduces an automated pipeline for generating high-quality Open-Domain Multi-step Reasoning (ODMR) datasets. It includes an adaptive sampler for ICL selection and self-prompted inference via ICL. This method constructs reasoning chains in six categories, originally designed in the construction of the MUsique (Trivedi et al., 2022) dataset. We employ the zero-shot version of SP-COT, using language and symbols to describe the six types of reasoning decompositions, instead of demonstration presentations owing to limited context length of LLMs.

## 4.3 Setting

**Language models** Because the average total length of the given paragraphs in our MHQA task is very long, we choose the model that performs well in long text question answer task when selecting the model. Besides, our FSM is structured to operate over multiple rounds, necessitating inference on chat models. The models we ultimately use include

| | Hotpotqa | | | | | | | 2wiki | | | | | | | Musique | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ans | | sup | | joint | | | ans | | sup | | joint | | | ans | | sup | | joint | | |
| | EM | F1 | EM | F1 | EM | F1 | Format | EM | F1 | EM | F1 | EM | F1 | Format | EM | F1 | EM | F1 | EM | F1 | Format |
| normal | 31.6 | 42.8 | 2.6 | 26.4 | 1.3 | 13.4 | 90.7 | 6.7 | 8 | 1.6 | 5.5 | 1 | 2.6 | 89.8 | 18.2 | 30.85 | - | - | - | - | 84 |
| COT | 3.1 | 9.7 | 0.1 | 0.7 | 0.1 | 0.4 | 4.4 | 0.6 | 1.9 | 0 | 0.1 | 0 | 0 | 4.2 | 1 | 6.55 | - | - | - | - | 7 |
| FSM1 | 22.5 | 33.28 | 0.7 | 9.88 | 0.4 | 3.6 | 100 | 27.6 | 37.91 | 4.7 | 25.84 | 1.9 | 9.11 | 100 | 26.2 | 41.22 | - | - | - | - | 100 |
| FSM2 | 33.1 | 45.97 | 1.8 | 28.8 | 1.0 | 15.74 | 100 | 36.07 | 49.31 | 7.7 | 38.4 | 5.1 | 19.4 | 100 | 21.9 | 37.66 | - | - | - | - | 100 |

Table 1: **Results on the MHQA benchmark by the gpt-3.5-turbo-1106.**

| | Hotpotqa | | | | | | | 2wiki | | | | | | | Musique | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ans | | sup | | joint | | | ans | | sup | | joint | | | ans | | sup | | joint | | |
| | EM | F1 | EM | F1 | EM | F1 | Format | EM | F1 | EM | F1 | EM | F1 | Format | EM | F1 | EM | F1 | EM | F1 | Format |
| Normal | 34.0 | 45.87 | 0.7 | 15.02 | 3.0 | 80.03 | 94.3 | 37.3 | 46.56 | 1.0 | 14.08 | 9.0 | 71.65 | 95.8 | 16.7 | 27.78 | - | - | - | - | 94 |
| COT | 12.3 | 25.99 | 0.4 | 4.47 | 2.0 | 17.83 | 16.2 | 8.2 | 19.3 | 0.2 | 1.27 | 1.0 | 4.55 | 7 | 4.5 | 13.56 | - | - | - | - | 14.7 |
| FSM1 | 23.4 | 31.97 | 2.4 | 29.27 | 2.0 | 9.82 | 100 | 30.11 | 40.01 | 14.2 | 47.03 | 2.0 | 8.53 | 100 | 26.0 | 38.44 | - | - | - | - | 100 |
| FSM-summary2 | 28.4 | 36.72 | 2.2 | 21.39 | 4.0 | 26.71 | 100 | 30.61 | 37.23 | 6.9 | 29.64 | 7.0 | 19.84 | 100 | 18.6 | 27.36 | - | - | - | - | 100 |

Table 2: **Results on the MHQA benchmark by the model named Qwen-72B. The mark - means those models have no ability to extract the evidence result.**

Openai GPT3.5-turbo-32k, GPT4-16K, Qwen72B-chat (Bai et al., 2023). And the maximum length of the question with given context can reach 8k, so for input length less than 8k, max-position-embeddings in config will be forced set to 8192. Following (Wang et al., 2023), to save computational cost, we conduct this experiment on subsets of the dataset by randomly selecting 1000 samples from the development sets.

### 4.4 Results

The results of setting 1(only answer) can be seen in **??**. And the result of Setting 2(answer with supporting fact) is displayed in **??** Our approach exhibits more impressive results in Setting 2 compared to Setting 1, and its performance improves as the difficulty of the dataset increases, which aligns with our expectations. This is because the instructions in Setting 2 are more complex, making it harder for models to follow them perfectly, while our method reduces the difficulty for Large Language Models (LLMs). Besides, HotpotQA and 2Wiki dataset has subpar data quality (Min et al., 2019), where some questions did not require multiple hops or reference to the text to find the answer. Traditional Chain of Thought (COT) methods score low on supporting facts yet achieve much higher scores on answers. This indicates that although the models may identify incorrect relevant paragraphs, they still provide the correct answers. Some instances may occur due to a misunderstanding of the instructions, but it is undeniable that there are serious issues with the authenticity and logical consistency of the models' reasoning chains. We also cannot exclude the possibility of dataset leakage during evaluation. While the performance in Setting 1 on less challenging datasets like HotpotQA and 2Wiki is relatively average, our method is more precise, and instances of hallucination are fewer. In summary, we believe that our method holds a competitive advantage.

## 5 Discussion

Figure. 3 provides examples of the wrong reasoning generated by existing methods when answering questions. We conclude four types of errors. **a)Reasoning Lost Issue**: Failed to compare in 'comparision' type questions; instead, provided an answer just with the last sub-question. **b)Formatting Error**: The output is not formatted correctly. **c)Sub-question Decomposition Error**: Incorrectly decomposed the sub-questions. **d)Sub-question Response Error**: Identified the wrong paragraph but provided a correct answer. **e)Hallucination Response**: Provided an answer without locating the relevant paragraph.

## 6 Conclusion

Our framework's meticulous step-by-step approach ensures that every phase is executed with precision. This contrasts with conventional methods where models output results in one go, and only the final formatting errors are addressed during evaluation

## 7 Limitations

This multi-turn dialogue process, inherent to our framework, mandates repeated handling of improperly formatted outputs, due to the output before

will be the next input, which can be challenging for models with smaller parameter sizes and weaker follow-instruction capabilities. Therefore, models with limited capacity to follow instructions might not benefit from our method as any error in the intermediate steps could lead to an abrupt termination of the process.

## References


Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, et al. 2023. Graph of thoughts: Solving elaborate problems with large language models. *arXiv preprint arXiv:2308.09687*.

Hejing Cao, Zhenwei An, Jiazhan Feng, Kun Xu, Liwei Chen, and Dongyan Zhao. 2023. A step closer to comprehensive answers: Constrained multi-stage question decomposition with large language models. *arXiv preprint arXiv:2311.07491*.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2020. Hierarchical graph network for multi-hop question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8823–8838, Online. Association for Computational Linguistics.

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

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Takeshi Kojima, Shane Shixiang, Gu Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. Compositional questions do not necessitate multi-hop reasoning. *arXiv: Computation and Language,arXiv: Computation and Language*.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020a. Unsupervised question decomposition for question answering. *arXiv preprint arXiv:2002.09758*.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020b. Unsupervised question decomposition for question answering. *arXiv: Computation and Language,arXiv: Computation and Language*.

Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and ChristopherD. Manning. 2019. Answering complex open-domain questions through iterative query generation. *Cornell University - arXiv,Cornell University - arXiv*.

Mokanarangan Thayaparan, Marco Valentino, Viktor Schlegel, and André Freitas. 2019. Identifying supporting facts for multi-hop question answering with document graph networks. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 42–51, Hong Kong. Association for Computational Linguistics.

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

Ming Tu, Guangtao Wang, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou. 2019. Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2704–2713, Florence, Italy. Association for Computational Linguistics.

Jinyuan Wang, Junlong Li, and Hai Zhao. 2023. Self-prompted chain-of-thought on large language models for open-domain multi-hop reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2717–2731, Singapore. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

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*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. 2022. Least-to-most prompting enables complex reasoning in large language models.

| | Hotpotqa | | 2wiki | | Musique | |
|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 |
| **Normal** | 31.89 | 43.67 | 36 | 46.58 | 19.22 | 33.26 |
| **COT** | 32.11 | 45.48 | 38.11 | 53.01 | 20.56 | 35.63 |
| **SP-COT** | 24.78 | 37.38 | 23.22 | 36.02 | 14.44 | 28.44 |
| **FSM1** | 24.5 | 39.33 | 27.1 | 40.58 | 23.1 | 40.34 |
| **FSM2** | 33.33 | 45.70 | 39.22 | 50.13 | 26.7 | 40.5 |

Table 3: Comparison of different approaches on three MHQA benchmarks in setting 1. Methods based on ChatGPT is performed by gpt-3.5-turbo-1106 version.

| | Hotpotqa | | 2wiki | | Musique | |
|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 |
| **Normal** | 31 | 41.56 | 31.9 | 39.08 | 12.9 | 19.9 |
| **COT** | 30.6 | 42.74 | 39.9 | 49.84 | 14.1 | 23.98 |
| **SP-COT** | 14.6 | 28.61 | 18.5 | 31.83 | 6 | 14.72 |
| **FSM1** | 28 | 37.43 | 39.1 | 47.9 | 33.2 | 48.45 |
| **FSM2** | 32.2 | 41.3 | 40.2 | 50.3 | 27.7 | 40.42 |

Table 4: **Results on Qwen72 in setting 1.**

Figure 3: Results on baseline and our FSM methods.