SG-FSM: A Self-Guiding Zero-Shot Prompting Paradigm for Multi-Hop Question Answering Based on Finite State Machine

Xiaochen Wang [1,2,3], Junqing He[3,†], Liang Chen[1], Reza Haf[4], Zhe Yang[1],
Yiru Wang[5], Xiangdi Meng[1,2], Kunhao Pan[3], Jiaxing Zhang[3], and Zhifang Sui [1,†]

[1]*National Key Laboratory for Multimedia Information Processing, Peking University*
[2]*School of Software & Microelectronics, Peking University*
[3]*International Digital Economy Academy*
[4]*Data Science and Artificial Intelligence, Monash University,* [5]*ModelTC*

Abstract

Large Language Models with chain-of-thought prompting, such as OpenAI-o1, have shown impressive capabilities in natural language inference tasks. However, Multi-hop Question Answering (MHQA) remains challenging for many existing models due to issues like hallucination, error propagation, and limited context length. To address these challenges and enhance LLMs' performance on MHQA, we propose the Self-Guiding prompting Finite State Machine (SG-FSM), designed to strengthen multi-hop reasoning abilities. Unlike traditional chain-of-thought methods, SG-FSM tackles MHQA by iteratively breaking down complex questions into sub-questions, correcting itself to improve accuracy. It processes one sub-question at a time, dynamically deciding the next step based on the current context and results, functioning much like an automaton. Experiments across various benchmarks demonstrate the effectiveness of our approach, outperforming strong baselines on challenging datasets such as Musique. SG-FSM reduces hallucination, enabling recovery of the correct final answer despite intermediate errors. It also improves adherence to specified output formats, simplifying evaluation significantly.[1]

1 Introduction

Multi-hop Question Answering (MHQA) is a challenging QA task that asks models to answer a complex and indirect question given multiple passages. Agents need to reason twice/more on documents to get the final answer. It has intrigued researchers for its complexity and practical implications (Ho et al., 2020; Yang et al., 2018; Trivedi et al., 2022).

Researchers employ three primary strategies to address MHQA using Large Language Models (LLMs) due to their powerful and promising ability. One effective method is In-Context Learning (ICL) (Wang et al., 2023; Zhou et al., 2022; Yao et al., 2022), where models are instructed to solve problems based on detailed demonstrations. However, few-shot methods are considered ineffective and inefficient as they require a minimum of 4-shot of manual designed demonstrations. Long context in these demonstrations may exceed context boundaries and distract attention (Liu et al., 2024). Another approach involves fine-tuning LLMs with domain-specific data, which requires substantial high-quality data and computational resources. It is effective but ineffcient and non-generalizable. The third method reduces the training cost by training a new module for only part of the procedure without training LLMs. For example, (Cao et al., 2023) trained a beam-retrieval model. After retrieving results, they utilize a few-shot LLM as a reader to answer the question. However, they only improved the system's retrieval capabilities, while the LLM still exhibits hallucinations and propagates errors.

In this paper, we summarize four common inference errors in the previous approaches and demonstrate the effectiveness of our method in tackling these issues. We found that LLMs struggle particularly in intermediate reasoning stages, where errors in initial steps can propagate, leading to incorrect conclusions. We define this error as lost-in-the-middle reasoning path. Detailed analysis is presented in the § 2.

Humans usually decompose sophisticated problems to solve them, supported by cognitive discoveries (Correa et al., 2023; Cheng et al., 2015). Many decomposition methods to assist LLMs have shown their effectiveness in other tasks (Fu et al., 2021; He et al., 2024). Inspired by the insights, this paper adapts the decomposing progress to MHQA to improve the performance. Herein we decompose an MHQA task in predefined order: first, identify the initial sub-question, then search for its answer in the text, and continue solving each subsequent sub-question in sequence until the final answer is reached. The process is similar to a Finite State

[1]https://github.com/2018211801/SG-FSM.

Findings of the Association for Computational Linguistics: NAACL 2025, pages 6040–6052
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

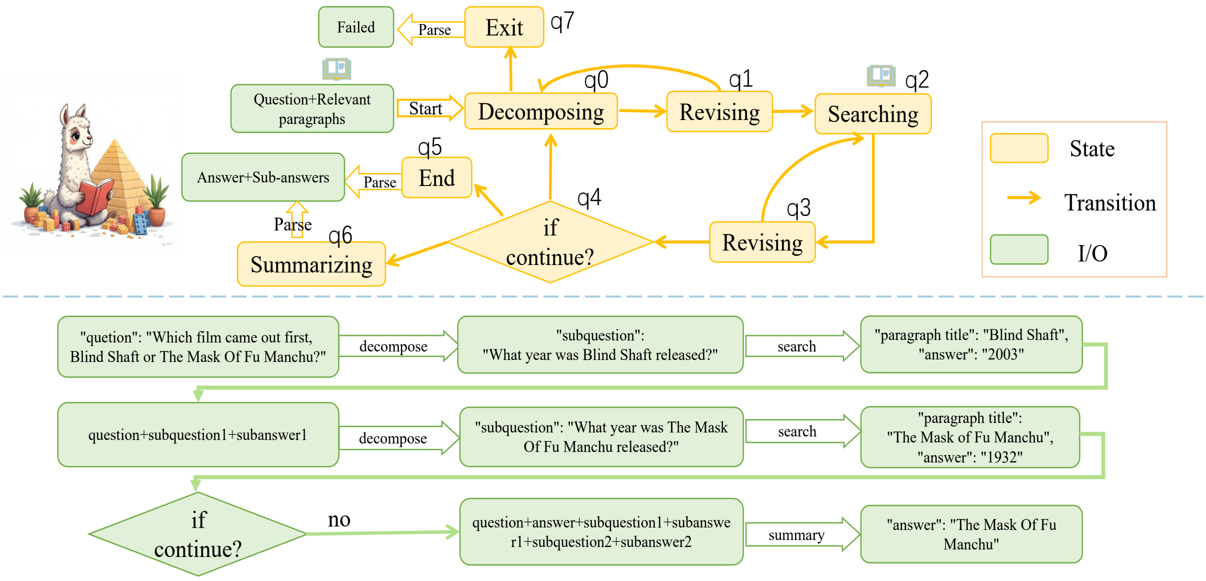

Figure 1: The flow chart of proposed SG-FSM. The upper part illustrates the flow of SG-FSM with state transitions. q_i is state defined in five-tuples. The lower part shows the input/output flows of an example.

Machine (FSM), which can constrain the intermediate reasoning process and shorten the lengthy reasoning path.

According to the analysis above, we propose a zero-shot method named **S**elf-**G**uiding **F**inite **S**tate **M**achine prompting (SG-FSM), simplifying the MHQA task into four sub-tasks: decomposing questions, searching for answers in candidate paragraphs, revising the format, and judging whether to continue. These four tasks resemble states in SG-FSM, and SG-FSM loops through these subtask states sequentially until the final answer is found. Lastly, SG-FSM summarizes key information ahead and self-corrects. Figure 1 depicts the process of the SG-FSM. We declare the advantages of SG-FSM to REAC as follows. In REACT, the 'control' over different stages is straightforward, with transitions following a chain-like structure across the states (ie the next step CoT, call the API, get the answer, go to the next step CoT). Whereas in SG-FSM, the control transitions across a fine-state-automata over the states, adaptively selecting the next state based on the results of the previous steps.

Extensive experiments on MHQA benchmarks (Yang et al., 2018; Trivedi et al., 2022; Ho et al., 2020) with GPT-3.5-turbo-1106 and Qwen-72B demonstrate that our approach outperforms baselines, nearly doubling the F1 score on Musique (Trivedi et al., 2022). Baselines often generate outputs in unexpected format errors which are hard to process while SG-FSM greatly reduces the format error and shows robustness. More importantly, SG-FSM also improves the correctness of intermediate reasoning and supporting evidence significantly while those of baselines are faulty even if they give a correct answer, as demonstrated in Figure 2.

Our contributions are as follows:

- We summarize common error cases in MHQA and analyze the reasons. Besides, we discover an unexpected phenomenon in the long free reasoning path using COT, which we called "lost-in-the-middle reasoning path".
- We introduce SG-FSM, a self-guiding zero-shot prompting paradigm to decompose complex questions iteratively, enhancing the capability of LLMs to solve complex problems through a controlled reasoning path.
- Extensive experiments on MHQA benchmarks in different settings validate SG-FSM's effectiveness, especially on challenging datasets.

2 Preliminary

Figure 2 presents error examples from the baseline methods and highlights the advantages of the proposed SG-FSM approach. Specifically, the direct method struggled with the long candidate text sequence, leading to errors in selecting the correct option and failing to provide supporting evidence. The CoT method did not produce the required JSON format and was unable to effectively decompose the problem, addressing only one

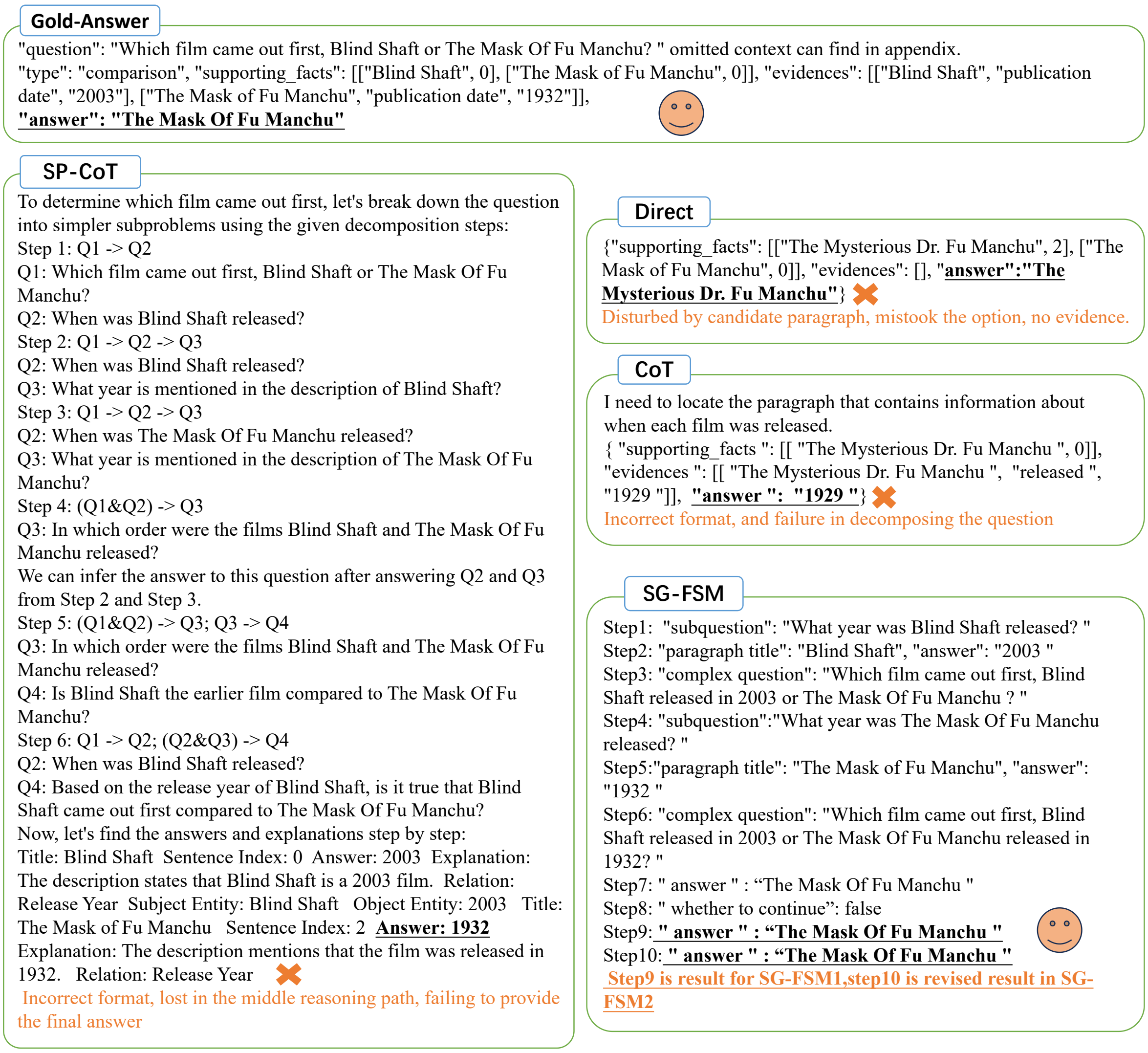

Figure 2: Outputs for different methods. Each error is marked, and SG-FSM can solve these errors.

sub-question. SP-COT repeatedly decomposed the problem but lost track of the original question, ultimately failing to deliver the final answer after comparison. In contrast, SG-FSM correctly inferred each step, arriving at the accurate final answer. After analyzing extensive cases, we concluded four primary reasons for bad cases in MHQA as follows:

a) Error Propagation: CoT is prone to introducing errors during intermediate reasoning steps, such as mistakes in decomposition and searching. These errors accumulate and propagate through the reasoning process, a phenomenon known as Rationale Drift in (Li et al., 2024).

b) Lost-in-the-middle reasoning path: Extended context in the reasoning process can lead the LLM to lose focus on the original question, making it difficult to provide the correct answer. It is called Answer Drift in (Li et al., 2024), when the model loses track of the question during inference.

c) Format Mismatch: Correct answers may not be recognized during evaluation due to format errors. It fails to follow instructions probably due to long contexts or hallucinations. Examples are presented in Section B.

d) Hallucination Response: Provided a correct answer without locating the relevant paragraph.

Issues of unfaithfulness and self-contradiction in the LLMs reasoning process are emerging areas. Lanham et al. (2023); Liu et al. (2023); Mündler et al. (2023) point out that in existing reasoning research, much work overly focuses on the predictive accuracy of models, neglecting the quality and consistency of the reasoning process itself.

3 Methodology

3.1 Task Definition

The issue of multi-hop QA is characterized by a question q and a set of pertinent (gold) supporting context documents $d_1, \ldots, d_S$ that hold the answer a. These context documents create a logical sequence essential for reaching the answer, drawn from a vast collection of documents D where the size of D greatly exceeds S. Given a multi-hop question and multiple related paragraphs, the model is required to provide the final answer, as well as the paragraph location to find the sub-answers for sub-questions.

3.2 Framework

We present our proposed Self-Guided Finite State Machine (SG-FSM) in two distinct stages as illustrated in Figure 1. Initially, we instruct LLMs to address sub-questions iteratively during the first phase, SG-FSM1 for short. Subsequently, in stage 2, LLMs are asked to summarize key information from each sub-question and instruct it to revise based on results of SG-FSM1. This is the difference between SG-FSM1 and SG-FSM2.

The SG-FSM is formally described as a five-tuple $(Q, \Sigma, \delta, q_0, F)$, where:

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$ is the set of states, where:
  - q_0: Decomposing the question
  - q_1: Revising the output of decomposing
  - q_2: Searching in the given paragraph
  - q_3: Revising the output of searching
  - q_4: Judging if question can be decomposed further
  - q_5: The end of SG-FSM1, final answer is found
  - q_6: Summarizing with key reasoning information
  - q_7: Early withdrawal
- $\Sigma = \{\text{question}\}$ is the complex question, representing the input to the system
- $\delta : Q \times \Sigma \to Q$ is the transition function, defined as follows:

$$\delta(q_0, \text{question}) = \begin{cases} q_1, \text{if correct} \\ q_7, \text{elif iterations} > 6 \\ q_0, \text{else} \end{cases}$$

$$\delta(q_1, \text{output in } q_0) = \begin{cases} q_2, \text{if correct} \\ q_0, \text{else} \end{cases}$$

$$\delta(q_2, \text{paragraph}) = q_3$$

$$\delta(q_3, \text{output in } q_2) = \begin{cases} q_4 & \text{if correct} \\ q_2 & \text{else} \end{cases}$$

$$\delta(q_4, \text{history}) = \begin{cases} q_0 & \text{if continue} \\ q_5 & \text{else} \end{cases}$$

$$\delta(q_5, \text{paragraph}) = q_6$$

  Where "if correct" means output can be parsed correctly, and "if continue" indicates whether the question can be further decomposed.
- q_0 is the initial state
- $F = \{q_5, q_6, q_7\}$ is the set of accept states, indicating the terminal state where the final answer is found or early exit.

The specific finite state automaton diagram is located in the upper part of Figure 1. It shows clear process of the FSM and how the states are transitioned. As the problem-solving steps for MHQA tasks adhere to a consistent pattern, they can be classified into four clear phases, with the model concentrating on addressing one task sequentially. These components are outlined as Decomposer, Searcher, Revisor, Terminator and Summarizer. All of them are performed by the same LLM, each utilizing a different prompt to ensure that only one task is processed in each round. Combining the four tasks may introduce more complexity than performing them individually, as a single error in reasoning could lead to the failure of the entire task. The specific prompt can be found in the Appendix A. The components will be described in detail in the following subsections with the inputs and outputs of each being presented.

An example is depicted in Figure 1. Let's describe its specific steps in detail. When parsing the output results at each step, if there are formatting errors or other issues, it is necessary to immediately use revision to correct output. If two attempts fail, SG-FSM1 will exit the loop early. Therefore, we omit revision steps for brevity.

First, the decomposer breaks down the original question and gives one sub-question *"What year was Blind Shaft released?"*. In the next turn, the searcher finds its answer in the multiple candidate paragraphs and outputs the referred paragraph title and answer. Since the first sub-question is addressed, it comes to the terminator to judge if the question can be decomposed further. Now the complex question actually becomes *"Which film came first, Blind Shaft released in 2003 or The Mask Of Fu Manchu?"*. Obviously, the SG-FSM1 should continue. Then, decomposer gives another sub-question *"What year was The Mask Of Fu Manchu released?"*. Then the searcher finds the answer "1932" in the reference paragraph. Next, the terminator answers the question *"Which film came first, Blind Shaft released in 2003 or The Mask Of Fu Manchu released in 1932?"* and quits this loop. The answer is *"The Mask Of Fu Manchu"*.

In the SG-FSM2 stage, we give LLMs key reasoning information ahead and instruct LLMs to revise these and answer the question again.

3.3 Decomposer

We need to ensure that the LLMs solve the problem step by step, so we decompose the complex problem to make it theoretically easier to answer. The input and output are as follows:

I/O for Decomposer

Input:
Please determine whether the question is simple sentence or compound sentence. If it is a simple sentence, return {"simple":true,"subquestion":null }.Otherwise decompose the question and generate the first answerable simple sentence.
Reply in the form of {"simple":false, "subquestion":xxx }. + Requirements.
Question: "Which film came first, Blind Shaft or The Mask Of Fu Manchu?"
Output: {"simple": false, "subquestion": What year was The Mask Of Fu Manchu released?}

To instruct LLM to output compliant JSON format for convenient parsing, the requirements should be strictly defined: *{examples of output format}. Do not reply any other words and provide answers in JSON format!* The output requirements in subsequent prompts are similar; for brevity, they will be omitted.

3.4 Searcher

Given sub-question ahead and candidate paragraph in origin task, searcher will find the answer (and supportting evidence in setting 2) directly.

I/O for Searcher

Input:
Given the paragraph below, please find out the paragraph that contains the answer of "{}" Please take a moment to thoroughly understand the content before proceeding to the questions, then carefully read the relevant paragraphs based on the question and provide the most likely answer.
Question: "What year was The Mask Of Fu Manchu released?
Context: paragraph...
Output: The answer is {"subanswer: 1932,

3.5 Revisor

After each step, the LLMs output content should be immediately parsed for analysis, and any errors should be corrected immediately. Only outputs with format errors enter this revisor step to self-correction. If there is still an error after two retries, we terminate the loop early and mark the answer as blank.

I/O for Revisor

Input:
Please rewrite the illegal json text below into an legal json string. Text: The answer is {"subanswers": ï932,}
Output: {"subanswer": 1932}

3.6 Terminator

Currently, most MHQA questions require 2-4 hops of reasoning. After addressing a sub-question, we need to determine whether the question has been fully decomposed. If the final answer has been discovered, exit this loop and proceed to the final SG-FSM2 summary stage.

I/O for Terminator

Input:
Can the original question be further decomposed into other different sub-question? Please output in the form of {"continue":true or false}.
original question: "Which film came first, Blind Shaft or The Mask Of Fu Manchu?
sub-question: "What year was The Mask Of Fu Manchu released?
Output: {"continue": true}

3.6.1 Summarizer

The previous modules together form SG-FSM1. Although the SG-FSM1 phase has already generated the answer, it may still contain logical errors, so we add SG-FSM2, listing all the key information and letting the LLMs check it again.

I/O for Summarizer

Input:
Original question: Which film came first, Blind Shaft or The Mask Of Fu Manchu?
Sub-question 1: What year was The Mask Of Fu Manchu released?
Paragraph: The Mask of Fu Manchu...
Evidence: (The Mask of Fu Manchu, released in, 1932)
Sub-answer: 1932
Sub-question 2: What year was Blind Shaft released?
Paragraph: Blind Shaft...
Evidence: (Blind Shaft, released in, 2003)
Sub-answer: 2003 Answer: The Mask of Fu Manchu
Please check based on the above information whether each sub-question's answer is correct, and whether the given answer is correct to the original question. Output the final correct answer in the form of {"Answer": xxx, "Reason": xxx}.."
Output: {"Answer": The Mask of Fu Manchu,"Reason": ...}

4 Experiments

4.1 Benchmark and Evaluation

We evaluate models on three high-quality MHQA datasets: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (2WikiQA) (Ho et al., 2020) and Musique (Trivedi et al., 2022). Learning from the shortcut phenomenon (Min et al., 2019) of single hop questions in HotpotQA, Musique (Trivedi et al., 2022) strictly controls the composition of the question, ensuring it undergoes multiple inferences to find the answer. Both HotpotQA and 2WikiQA have ten candidate paragraphs for each question and originally have supporting facts. In comparison, Musique has twenty candidates with longer text and no supporting facts. Question composition types and number of hops are listed in Table 1. More question types and hops are contained in Musique. Therefore, Musique is the most difficult MHQA dataset.

Datasets	Hotpot QA	2WikiQA	Musique
#Hops	1-2	2	2-4
#Types	2	4	6
#Paragraphs	10	10	20

Table 1: Statistics of datasets in experiments. Type is short for question composition types. Paragraphs represent the number of candidate paragraphs for each sample.

Following (Wang et al., 2023), we adopt the exact match (EM) and F1 scores as evaluation metrics and conduct experiments on subsets of the datasets by randomly selecting 1000 samples from the test sets. Despite having similar basic instructions and a clearly defined output format for all methods, the model's consistency in following instructions may vary across different methods. This variation can result in difficulty in answer extraction during evaluation. To address this issue, we introduce a new metric, format, measuring the accuracy of the output format.

		Musique		**HotpotQA**		**2WikiQA**	
		EM	F1	EM	F1	EM	F1
GPT	**Direct**	19.2	33.3	31.9	43.7	36.0	46.6
	CoT	20.6	35.6	32.1	45.5	38.1	**53.0**
	SP-CoT	14.4	28.4	24.8	37.4	23.2	36.0
	SG-FSM1	23.1	40.3	24.5	39.3	27.1	40.6
	SG-FSM2	**26.7**	**40.5**	**33.3**	**45.7**	**39.2**	50.1
Qwen	**Direct**	12.9	19.9	31.0	41.6	31.9	39.1
	CoT	14.1	24.0	30.6	**42.7**	39.9	49.8
	SP-CoT	6.0	14.7	14.6	28.6	18.5	31.8
	SG-FSM1	**33.2**	**48.5**	28.0	37.4	39.1	47.9
	SG-FSM2	**33.2**	**48.5**	**32.2**	41.3	**40.2**	**50.3**

Table 2: Results on the MHQA benchmark by the GPT-3.5-turbo-1106 and Qwen-72B in setting 1 do not provide supporting evidence in the reasoning.

4.2 Baselines

We conduct experiments by considering the following baselines with both open-sourced and API endpoints:

• The **Direct** strategy inference the answer directly, which is the basic form, using only task descriptions and output requirements as the prompt.

• The **CoT** (Wei et al., 2022) prompts LLMs to create intermediate step-by-step rationales, aiding in the reasoning process for obtaining answers.

		Musique			HotpotQA							2WikiQA						
		ans			ans		sup		joint			ans		sup		joint		
		EM	F1	Format	EM	F1	EM	F1	EM	F1	Format	EM	F1	EM	F1	EM	F1	Format
Qwen	**Direct**	18.2	30.9	84.0	31.6	42.8	**2.6**	26.4	**1.3**	13.4	90.7	6.7	8.0	1.6	5.5	1.0	**2.6**	89.8
	CoT	1.0	6.6	7.0	3.1	9.7	0.1	0.7	0.1	0.4	4.4	0.6	1.9	0	0.1	0.0	0.0	4.2
	SP-CoT	5.6	13.97	60.3	13.1	26.86	0.6	3.94	0.5	1.89	32.2	16.6	29.85	1.5	4.24	0.5	1.32	35.5
	SG-FSM1	**26.2**	**41.2**	**100.0**	22.5	33.3	0.7	9.9	0.4	3.6	**100.0**	27.6	37.9	4.7	25.8	1.9	9.1	**100.0**
	SG-FSM2	21.9	37.7	**100.0**	**33.1**	**46.0**	1.8	**28.8**	1.0	**15.7**	**100.0**	**36.1**	**49.3**	**7.7**	**38.4**	**5.1**	**19.4**	**100.0**
GPT	**Direct**	16.7	27.8	94.0	**34.0**	**45.9**	0.7	15.0	3.0	8.0	94.3	**37.3**	**46.6**	1.0	14.1	**9.0**	7.2	95.8
	CoT	4.5	13.6	14.7	12.3	26.0	0.4	4.5	2.0	17.8	16.2	8.2	19.3	0.2	1.3	1.0	4.6	7.0
	SP-CoT	14.33	27.92	84.9	24.0	36.48	2.56	24.52	1.1	10.11	82.8	23.11	36.03	9.56	34.87	3.0	12.50	79.2
	SG-FSM1	**26.0**	**38.4**	**100.0**	23.4	32.0	**2.4**	**29.3**	2.0	9.8	**100.0**	30.1	40.0	**14.2**	**47.0**	2.0	8.5	**100.0**
	SG-FSM2	18.6	27.4	**100.0**	28.4	36.7	2.2	21.4	**4.0**	**26.7**	**100.0**	30.6	37.2	6.9	29.6	7.0	**19.8**	**100.0**

Table 3: Results on the MHQA benchmark by the GPT-3.5-turbo-1106 and Qwen-72B with zero-shot in setting 2, which requires providing the supporting evidence in the reasoning. "Ans" means answer. "Sup" means supporting paragraph index and title. "Joint" means evidence triples including relationshipS with sub-answers.

• The **SP-CoT** (Wang et al., 2023) organizes reasoning chains into six categories, inspired by the construction of the Musique (Trivedi et al., 2022) dataset. It designs multiple demonstrations and then selects the suitable ones for in-context learning.

All the prompts for the baselines above are included in the Appendix A.

4.3 Settings

Our study explores two settings: (1) only asks for answers given the context and question without the need for supporting facts, and (2) building a complete reasoning chain that includes the answer, supporting evidence, and facts to assess the coherence of the reasoning process. Due to the lack of golden evidence for Setting 2 in Musique, our evaluation did not report its results. Existing methods mostly adopt Setting1 and do not report Setting 2. By adding Setting 2, we can observe how enhancing the inference process affects the output of the LLMs, highlighting instances where the model provides the correct answer through an incorrect intermediate process.

4.4 Models

Since MHQA requires models with the ability to process lengthy text for multiple rounds of reasoning, we selected GPT-3.5-turbo-32k [2] and Qwen72B-chat (Bai et al., 2023) for our study. Additionally, we employed Vllm (Kwon et al., 2023) to accelerate the inference process.

[2]https://platform.openai.com/docs/models/gpt-3-5-turbo

4.5 Results

The results for Setting 1 (sole answer) are presented in Table 2, while the outcomes for Setting 2 (answers paired with supporting facts) are shown in Table 3. Our method generally outperforms the baseline, showcasing the effectiveness of SG-FSM. By dissecting the question step by step, we enhance the accuracy of each step towards the sub-questions. This focused approach during searching ensures that attention is not diverted by extraneous texts, leading to more precise results.

Notably, SG-FSM excels on the Musique dataset, showing a significant improvement of 6-10 percentage points. This dataset, as discussed in Section 4.1, poses more hops and longer reasoning paths, making it more challenging than others. Our method provides clear guidance to LLMs at each step, promptly verifying the actions taken, thus easing the task complexity. Additionally, its performance surpasses that of HotpotQA and 2WikiQA, as it avoids single-hop shortcuts (Min et al., 2019), further validating the effectiveness of our method.

Besides, the direct method struggles with supporting facts but achieves substantially higher scores on answers than CoT. This phenomenon indicates that although LLMs may misinterpret intermediate reasoning steps, they still yield correct answers, hinting at underlying data leakage and hallucination. While some errors may stem from misinterpreting instructions, significant concerns regarding the authenticity and logical coherence of the models' reasoning chains.

Because of severe hallucination issues and data leakage problems in the LLMs, we call for a new benchmark that uses GPT to evaluate the logical consistency throughout the entire reasoning process. Under this benchmark, the effectiveness of our method can be better demonstrated.

4.6 Ablation Study

In our previous case analysis, we observed instances of hallucination and deceptive reasoning in LLMs when responding to questions: they provided correct answers but used erroneous reasoning. To investigate this phenomenon further, we compared the preliminary results obtained before the final summary revision to those after the revision, specifically contrasting SG-FSM1 with SG-FSM2.

Generally, the results of SG-FSM2 are more accurate compared to SG-FSM1, although its scores fluctuate. In one scenario, if answer is correct, relevant documents are incorrect, then after correction, the answer becomes unavailable. This corrects the illusion, and the score will decrease. In the other scenario, if relevant documents are correct but answer is wrong, and only then the individual corrects the answer to be correct, the score will increase. In summary, SG-FSM2 can only provide the correct answer when the appropriate relevant documents are supplied. The case where SG-FSM2 is worse than SG-FSM1, correct answer and wrong intermediate reasoning evidence, indicates that LLMs suffers from severe hallucination issues.

5 Related Work

5.1 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.

5.2 Lage Language model for reasoning.

CoT(Wei et al., 2022) reveals the ability of large language models to formulate their reasoning procedure for problem-solving. Several follow-up works have since been performed, including the least-to-most prompting technique (Zhou et al., 2022) for solving complicated tasks, zero-shot CoT (Kojima et al., 2022), graph-of-thought (GoT) (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. Recently, OpenAI-o1 [3] series models perform remarkably well, as they are trained with reinforcement learning to execute complex reasoning tasks. The key feature of OpenAI-o1 is its methodical approach, generating a long internal chain of thought before responding to user queries. Our work proved the necessity of extending and splitting the inference chain before it.

5.3 Task decomposition.

Perez et al. (2020a) decomposes a multi-hop question into a number of independent single-hop subquestions, which are answered by an off-the-shelf question-answering (QA) model. These answers are then aggregated to form the final answer. Both question decomposition and answer aggregation require training models. After the emergence of Large Language Models (LLMs), traditional training methods (Cao et al., 2023) are rarely used due to their expensive nature. Most current research focuses on the few-shot 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 unnecessarily increase the difficulty.

6 Conclusion

We investigated and classified error reasons in traditional methods where LLMs underperform the MHQA task in the paper. Besides, we discover an unexpected phenomenon in the long free reasoning path using CoT, called "lost-in-the-middle reasoning path". To address these issues, we propose SG-FSM, a self-guiding zero-shot prompting approach to break down intricate questions step by step iteratively. This method improves the ability of LLMs to tackle difficult problems by guiding them through a controlled and extended reasoning process. Extensive experiments on multiple benchmarks show the superiority of SG-FSM over strong baselines and its effectiveness. SG-FSM delivers more robust and explainable reasoning output including answers and supporting facts by guiding the reasoning process and performing timely revisions.

[3] https://openai.com/o1/

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.

The primary factor is the LLM's capability, with prompts playing a supporting role. LLMs exhibit varying abilities largely dependent on how they are trained. We believe that the performance of a model is mostly influenced by whether the testing and training data distributions are consistent. Thus, if the model uses our method to incorporate step-by-step inference of control during training, the effect will be better.

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*.

Justin Cheng, Jaime Teevan, Shamsi T Iqbal, and Michael S Bernstein. 2015. Break it down: A comparison of macro-and microtasks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 4061–4064.

Carlos G Correa, Mark K Ho, Frederick Callaway, Nathaniel D Daw, and Thomas L Griffiths. 2023. Humans decompose tasks by trading off utility and computational cost. *PLOS Computational Biology*, 19(6):e1011087.

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.

Junqing He, Kunhao Pan, Xiaoqun Dong, Zhuoyang Song, LiuYiBo LiuYiBo, Qianguosun Qianguosun, Yuxin Liang, Hao Wang, Enming Zhang, and Jiaxing Zhang. 2024. Never lost in the middle: Mastering long-context question answering with position-agnostic decompositional training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13628–13642, Bangkok, Thailand. 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, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Jiachun Li, Pengfei Cao, Chenhao Wang, Zhuoran Jin, Yubo Chen, Daojian Zeng, Kang Liu, and Jun Zhao. 2024. Focus on your question! interpreting and mitigating toxic cot problems in commonsense reasoning. *arXiv preprint arXiv:2402.18344*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Ziyi Liu, Isabelle Lee, Yongkang Du, Soumya Sanyal, and Jieyu Zhao. 2023. Score: A framework for self-contradictory reasoning evaluation. *arXiv preprint arXiv:2311.09603*.

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*.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2023. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *arXiv preprint arXiv:2305.15852*.

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, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Appendix

A Prompt

A.1 SG-FSM1

Decomposer: Please determine whether the question is simple sentence or compound sentence. If it is a simple sentence, return "simple":true,"subquestion":null.Otherwise, simple: false, decompose the question and generate the first answerable simple sentence. reply in the form of "simple":false,"subquestion":xxx. Do not reply any other words and provide answers in JSON format!

Searcher: Given the paragraph below, please find out the paragraph that contains the answer of "" Please take a moment to thoroughly understand the content before proceeding to the questions, then carefully read the relevant paragraphs based on the question and provide the most likely answer. Return the title of the paragraph and the answer no more than 5 words in the form of "question":xxx, "paragraph title":xxx, "answer":xxx. Do not reply any other words and provide answers in JSON format!

Judge-if-continue: Please compare the complex question and subquestion, answer whether they are semanically identical in the form of "identical":true or false. Do not reply any other words and provide answers in JSON format!

A.2 SG-FSM2

FSM2-post-summary-again: Documents: paragraphs:paragraphs found in FSM1 subquestion and answers:subquestion and answers given in FSM1 Question:origin question Answer the question reasoning step-by-step based on the Doucments. If it is a general question, please respond with 'Yes' or 'No'. Finally, you must return the title of the context, the sentence index (start from 0) of the paragraph and the concise answer no more than 10 words and explaination in the form of "supporting-facts": [[title, sentence id], ...], "evidences": [[subject entity, relation, object entity],...], "answer":"xxx","explain":"xxxx". Do not reply any other words.

A.3 Baseline

SP-CoT(Wang et al., 2023): This is a two-hop to four-hop reasoning question-answering task that requires decomposing the questions into simple, answerable single-hop questions. The decomposition process involves four types of questions: comparison, inference, compositional, and bridge-comparison. There are six specific decomposition steps in total, denoted by Q representing the decomposed subproblems. The steps are as follows: First, Q1 -> Q2 Second, Q1 -> Q2 -> Q3 Third, Q1 -> Q2 -> Q3 Fourth, (Q1&Q2) -> Q3 Fifth, (Q1&Q2) -> Q3; Q3 -> Q4 Sixth, Q1 -> Q2; (Q2&Q3) -> Q4 The process involves first determining the type of question and then identifying the decomposition process type. It's important to note that the decomposition of questions cannot be provided all at once; it must be done step by step. Each subproblem needs to be decomposed and answered before moving on to the next one, as there is interdependence between the subproblems .Finally, you must return the title of the context, the sentence index (start from 0) of the paragraph and the concise answer and explaination in the form of "explain":"xxxx","supporting-facts": [[title, sentence id], ...], "evidences": [[subject entity, relation, object entity],...],"answer":"no sentence and no more than 10 words ". Do not reply any other words.

CoT-setting1-w/o-evidence: Answer the question according to the context,Let's think step by step, and explain your reasoning process. You must return in the form of "explain":"xxxx","answer":answer. Do not reply any other words.

direct-setting1-w/o-evidence: Answer the question according to the context. You must return in the form of "explain":"xxxx","answer":answer. Do not reply any other words.

direct-setting2-w-evidence: Answer the question according to the context. Find the paragraph that contains the answer of question, and summarize a triple that contains [subject entity, relation, object entity]. Finally, you must return the title of the context, the sentence index (start from 0) of the paragraph and the concise answer no more than 10 words in the form of "supporting-facts": [[title, sentence id], ...], "evidences": [[subject entity, relation, object entity],...], "answer":answer. Do not reply any other words.

prompt-step: Answer the question according to the context,Let's think step by step, and explain your reasoning process. Find the paragraph that contains the answer of question, and summarize a triple that contains [subject entity, relation, object entity]. Finally, you must return the title of the context, the sentence index (start from 0) of the paragraph and the concise answer no more than 10 words in the form of "supporting-facts": [[title, sentence id], ...], "evidences": [[subject entity, relation, object entity],...], "answer":answer. Do not reply any other words.

React-setting2-w-evidence: Solve a question answering task with interleaving Thought, Action, Observation steps. Thought can reason about the current situation, and Action can be three types: (1) Search[entity], which searches the exact entity on given context and returns the first paragraph if it exists. If not, it will return some similar entities to search. (2) Lookup[keyword], which returns the next sentence containing keyword in the current passage. (3) Finish[results], which returns the answer and finishes the task. You should plan and reason in the Thought, then perform your Action, lastly, observe the result of action. Loop this process until the problem was finished. At last, you must additional output the title of the paragraphs, the sentence index (start from 0) of the paragraph and the concise answer no more than 10 words and explaination in the form of Thought: reasoning Action: Search[entity] or Lookup[keyword] or Finish[results] Observation: observe the results of action end with Finish["supporting-facts": [[title, sentence id], ...], "evidences": [[subject entity, relation, object entity],...], "answer":answer]

B Format Error

Gold-answer

{"question": "What government position was held by the woman who portrayed Corliss Archer in the film Kiss and Tell?", "answer": "Chief of Protocol", "type": "bridge", "supporting_facts": [["Kiss and Tell (1945 film)", 0], ["Shirley Temple", 0], ["Shirley Temple", 1]], "level": "hard"}

{"question": "What science fantasy young adult series, told in first person, has a set of companion books narrating the stories of enslaved worlds and alien species?", "answer": "Animorphs", "type": "bridge", "supporting_facts": [["The Hork-Bajir Chronicles", 0], ["The Hork-Bajir Chronicles", 1], ["The Hork-Bajir Chronicles", 2], ["Animorphs", 0], ["Animorphs", 1]], "level": "hard"}

{"question": "Are the Laleli Mosque and Esma Sultan Mansion located in the same neighborhood?", "answer": "no", "type": "comparison", "supporting_facts": [["Laleli Mosque", 0], ["Esma Sultan Mansion", 0]], "level": "hard"}

{"question": "The director of the romantic comedy \"Big Stone Gap\" is based in what New York city?", "answer": "Greenwich Village, New York City", "type": "bridge", "supporting_facts": [["Big Stone Gap (film)", 0], ["Adriana Trigiani", 0]], "level": "hard"}

FSM

{"answer": "Ambassador to Ghana and Czechoslovakia, Chief of Protocol of the United States.", "supportin_facts": [["Kiss and Tell (1945 film)", 0], ["Shirley Temple", "1"], ["Shirley Temple", "1"]], "evidences": [["Kiss and Tell (1945 film)", "portrayed", "Shirley Temple"], ["Shirley Temple Black", "was named", "United States ambassador to Ghana and to Czechoslovakia and also served as Chief of Protocol of the United States"], ["Shirley Temple Black", "had", "notable achievements outside of her acting career"]]}

{"answer": "The Hork-Bajir Chronicles", "supportin_facts": [["The Hork-Bajir Chronicles", "2"], ["The Hork-Bajir Chronicles", "0"]], "evidences": [["Animorphs series", "involves", "enslaved worlds and alien species"], ["The Hork-Bajir Chronicles", "involving", "enslaved worlds and alien species"]]}

{"answer": "Ortaköy neighborhood of Istanbul, Turkey", "supportin_facts": [["Esma Sultan Mansion", "0"], ["Esma Sultan Mansion", "0"]], "evidences": [["Esma Sultan Mansion", "location", "Ortaköy neighborhood of Istanbul, Turkey"]]}

{"answer": "Greenwich Village", "supportin_facts": [["Big Stone Gap (film)", "0"], ["Adriana Trigiani", "0"]], "evidences": [["Big Stone Gap", "director", "Adriana Trigiani"], ["Adriana Trigiani", "based in", "Greenwich Village, New York City"]]}

Figure 3: The outputs of SG-FSM are standard json format.

COT

{"answer": "To answer the question and provide a reasoning process, first, I will search for information related to the woman who portrayed Corliss Archer in the film Kiss and Tell. Then, I will search for the film Kiss and Tell in the given context to find the relevant paragraph. After finding the relevant paragraph, I will summarize a triple that contains [subject entity, relation, object entity] and return the title of the context, the sentence index (start from 0) of the paragraph, and the concise answer.\n\nUpon analyzing the context, I found the relevant paragraph in the section \"Kiss and Tell (1945 film)\". The paragraph contains the information related to the woman who portrayed Corliss Archer, which is Shirley Temple.\n\nTriple: [\"Shirley Temple\", \"portrayed\", \"Corliss Archer\"]\n\nTitle: Meet Corliss Archer\n\nSentence Index: 6\n\nConcise Answer: Shirley Temple portrayed Corliss Archer.\n\nTherefore, the final output is:\n```json\n{\n \"supporting_facts\": [[\"Meet Corliss Archer\", 6]],\n \"evidences\": [[\"Shirley Temple\", \"portrayed\", \"Corliss Archer\"]],\n \"answer\": \"Shirley Temple portrayed Corliss Archer.\"\n}\n```", "question": "\"What government position was held by the woman who portrayed Corliss Archer in the film Kiss and Tell?\""}

{ "answer": "To find the answer, let's first identify the science fantasy young adult series told in first person. The paragraph that contains the relevant information is from the \"Animorphs\" section. The triple containing the solution is [\"Animorphs\", \"is a science fantasy series of young adult books written by\", \"Katherine Applegate and her husband Michael Grant\"]. The title of the context, sentence index, and the concise answer are as follows:\n\nTitle: \"Animorphs\"\nSentence Index: 0\nAnswer: \"Animorphs\"", "question": "\"What science fantasy young adult series, told in first person, has a set of companion books narrating the stories of enslaved worlds and alien species?\""}

{"answer": "To find out if the Laleli Mosque and Esma Sultan Mansion are located in the same neighborhood, first, I need to locate the paragraph that contains the relevant information about the locations of both entities. Then, checking the information from the paragraph, the relation \"located in\" will be identified between \"Laleli Mosque\" and \"Laleli, Fatih\" or \"Esma Sultan Mansion\" and \"Ortaköy neighborhood\", to conclude if they are in the same neighborhood or not.\n\nThe relevant paragraph is:\n\n[\"Laleli Mosque\", [\"The Laleli Mosque (Turkish: \\\"Laleli Camii, or Tulip Mosque\\\") is an 18th-century Ottoman imperial mosque located in Laleli, Fatih, Istanbul, Turkey.\"]],\n[\"Esma Sultan Mansion\", [\"The Esma Sultan Mansion (Turkish: \\\"Esma Sultan Yal\\u0131s\\u0131\\\"), a historical yal\\u0131 (English: waterside mansion) located at Bosphorus in Ortak\\u00f6y neighborhood of Istanbul, Turkey and named after its original owner Esma Sultan, is used today as a cultural center after being redeveloped.\"]],\n\nThe triple that contains the necessary information is: \n[\"Laleli Mosque\", \"located in\", \"Laleli, Fatih\"]\n[\"Esma Sultan Mansion\", \"located at\", \"Bosphorus in Ortaköy neighborhood\"]\n\nBased on this information, the concise answer is: \"No, they are not located in the same neighborhood.\"\n\nNow, I will use this evidence to provide a summary triple and supporting facts.\nThe summary triple is:\n{\"supporting_facts\": [[\"Esma Sultan Mansion\", 6], [\"Laleli Mosque\", 5]], \"evidences\": [[\"Laleli Mosque\", \"located in\", \"Laleli, Fatih\"], [\"Esma Sultan Mansion\", \"located at\", \"Bosphorus in Ortaköy neighborhood\"]], \"answer\":\"No, they are not located in the same neighborhood.\"}", "question": "\"Are the Laleli Mosque and Esma Sultan Mansion located in the same neighborhood?\""}\

{"answer": "To find the answer, let's start by identifying the paragraph containing information about the director of the romantic comedy \"Big Stone Gap.\" Upon examining the paragraphs, we can see that the paragraph about \"Big Stone Gap (film)\" contains the relevant information.\n\nTriple: [\"Adriana Trigiani\", \"is the director of\", \"Big Stone Gap\"]\n\nThen, the concise answer is: \"Adriana Trigiani\"\n\nBased on this, the requested details are as follows:\n- Title of the context: \"The director of the romantic comedy \"Big Stone Gap\" is based in what New York city?\"\n- Sentence index: 8\n- Concise answer: \"Adriana Trigiani\"\n- Supporting facts: [[\"Big Stone Gap (film)\", 0]]\n- Evidence: [[\"Adriana Trigiani\", \"is the director of\", \"Big Stone Gap\"]]", "question": "\"The director of the romantic comedy \\\"Big Stone Gap\\\" is based in what New York city?\""}

Figure 4: There are many error format examples for CoT. It is hard to get the answer for evaluation.