AnchorCoT: Anchors Pave the Way for Multi-hop Reasoning

Tianshi Ming[1], Xian Wu[2]*, Yingying Zhang[2], Zichuan Fu[3], Dawei Cheng[1]

[1]Tongji University, [2]Tencent Jarvis Lab, [3]City University of Hong Kong

2151569@tongji.edu.cn, kevinxwu@tencent.com, dcheng@tongji.edu.cn,

Abstract

Large Language Models (LLMs) have made substantial strides in a broad array of natural language tasks. Recently, LLMs have demonstrated potential reasoning capabilities through prompt design, such as the Chain of Thought (CoT). Despite their superiority in question answering, LLMs still face challenges in answering questions that require multi-hop reasoning, often generating unreliable reasoning chains during answer generation. To improve LLMs' performance in multi-hop reasoning, we introduce a novel reasoning approach, AnchorCoT, designed to assist LLMs in answering questions involving complex logical reasoning steps. AnchorCoT first predicts key entities which work as important "anchors" to guide the reasoning process and then employs a novel ranking algorithm to ensure the logical sequence of the predicted answers. We implement AnchorCoT on Qwen2.5-7B/14B and GPT-4o and evaluate our method on widely used multi-hop reasoning datasets, including HotpotQA, 2WikiMultiHopQA, and MuSiQue-Ans. The experimental results show that AnchorCoT outperforms existing methods in multi-hop question reasoning and provides more accurate reasoning results in multi-hop question answering tasks.

1 Introduction

Large language models (LLMs) have demonstrated impressive in-context learning abilities across a range of natural language processing (NLP) tasks (Grattafiori et al., 2024; OpenAI, 2024; Hoffmann et al., 2022; Chowdhery et al., 2022), such as information retrieval (Schlatt et al., 2024; Guo et al., 2024), relation extraction (Zaratiana et al., 2024; Efeoglu and Paschke, 2024) and question answering (Sohn et al., 2024; Zhao et al., 2024). Recently, the reasoning ability of LLMs (Giadikiaroglou et al., 2024) has garnered growing attention due to its critical role in solving problems with complex logic.

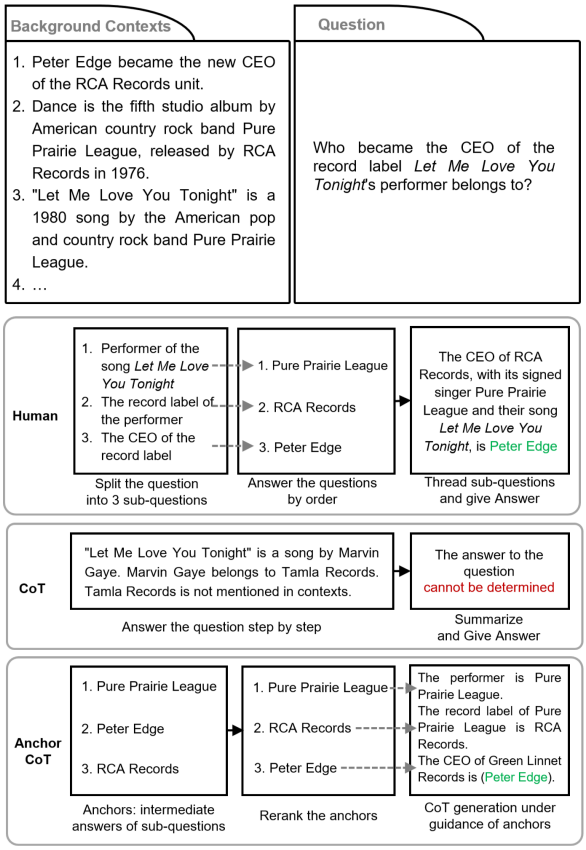

Figure 1: Human, CoT and AnchorCoT approaches in solving multi-hop question (using 3-hop question as example). Human first split multi-hop questions into sub-questions and get answers of sub-questions by order, then summarize the answer. Chain of Thought(CoT) solve multi-hop question step by step, but get wrong answer without guidance. Our approach first predict intermediate answers as anchors, then rerank the anchors in logical order. Finally, AnchorCoT uses anchors as guidance in CoT generation. The example use Qwen2.5-7B-Instruct model.

Despite the significant success of standard large language models (LLMs) in tackling question-answering tasks as demonstrated in various stud-

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 15522–15536

ies (Brown et al., 2020; Liu et al., 2023; Bao et al., 2023; Creswell et al., 2022), their performance tends to falter on complex reasoning tasks that necessitate multiple logical reasoning steps (Wei et al., 2023). LLMs have a tendency to generate hallucinations during the answer generation process in multi-hop reasoning tasks, often yielding responses that seem plausible but lack factual substantiation (Huang et al., 2025).

Current LLMs in multi-hop reasoning task generally follows Chain-of-Thought strategy. Common strategies include instruction prompting (Wei et al., 2023; Wang et al., 2023a; Xu et al., 2024), reasoning path searching (Yao et al., 2023; Besta et al., 2024; Chu et al., 2024), and majority voting (Wang et al., 2023b; Chen et al., 2023). However, the generated reasoning chains often still exhibit hallucinations due to insufficient or weak supervision during the generation process. Instruction prompting offers indirect and uncontrollable guidance, which makes the guidelines unreliable and difficult to assess their correctness. Both reasoning path searching and majority voting approaches generate reasoning steps sequentially, where an error in one step can propagate and amplify in subsequent steps, leading to a cascading effect that ultimately derails the entire reasoning chain.

Figure 1 displays an QA example which requires 3-hop reasoning. For human, they usually split the question into sub-questions: *who is the performer*, *which record label does the performer belongs to* and finally *who is the CEO of the record label*. By answering each sub-question in order, the final answer can be concluded; For LLM enabled with Chain-of-Thought strategy, it does output the reasoning process. However, it gives wrong answer for *who is the performer of the song "Let Me Love you Tonight"*. As a result, the subsequent reasoning process is incorrect;

Inspired by human reasoning process, we propose a novel chain-of-thought reasoning approach AnchorCoT, which provides direct step-wise guidance for multi-hop reasoning generation. Our approach first predicts the key entities which are critical to guide the reasoning process. We define these entities as "anchors". As depicted in Figure 1, we mine three related key entities: "Pure Prairie League", "Peter Edge" and "PCA Records" from the background context and the question. Then we employ a ranking algorithm to improve the logical order of these anchors. Finally, the anchors will serve as step-wise direct guidance for Chain-of-Thought reasoning generation. We test our approach on commonly used multi-hop reasoning datasets, including HotpotQA (Yang et al., 2018), 2WikiMultihopQA(Ho et al., 2020) and MuSiQue-Ans(Trivedi et al., 2022) datasets. Experiments are conducted on Qwen2.5-7B/14B (Qwen Team, 2024) and GPT-4o (OpenAI, 2024) model. Results demonstrate that our approach show superiority over not only existing instruction prompting methods, but also reasoning path searching and domain-specific approaches.

Our contributions can be summarized as follows:

- We introduce AnchorCoT, a novel framework designed to predict and rank intermediate key entities as reasoning anchors in multi-hop reasoning tasks with complex logic steps, which significantly strengthens reasoning capability of LLMs.
- We propose a novel ranking algorithm of anchor candidates based on multi-hop embedding and anchor embeddings.
- We conduct experiments on multi-hop reasoning datasets, including HotpotQA, 2WikiMultiHopQA and MuSiQue-Ans. Our approach excels not only on GPT-4o, but also on open-source large model Qwen2.5-7B/14B. Results demonstrate that our approach surpasses existing CoT approaches on both Exact Match and F1 score metrics.

2 Related Work

2.1 Multi-hop reasoning

Multi-hop reasoning is a challenging task of question answering tasks in natural language processing. Unlike single-hop questions, multi-hop questions require LLMs to generate intermediate conclusions step by step before arriving at the final answer. The sequential reasoning process is referred to as the reasoning chain (Mavi et al., 2024). A significant challenge for LLMs in addressing multi-hop questions is the occurrence of hallucinations, which can result in factually incorrect or unsupported reasoning. Our approach addresses the issue by enforcing LLMs to identify intermediate logical answers prior to reasoning chain generation. These predicted anchors establish a structured reasoning path and provide contextually aligned guidance throughout the process.

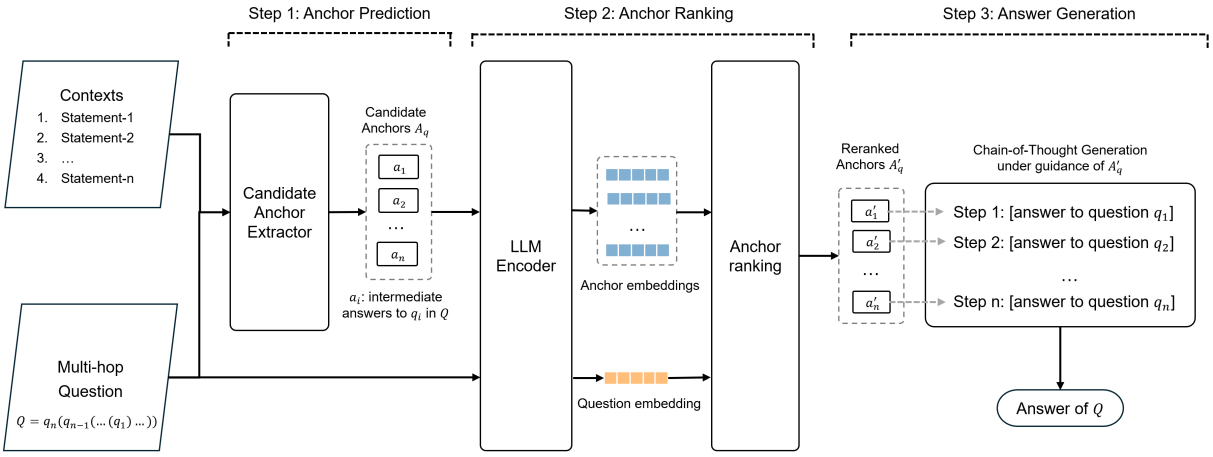

Figure 2: Overall Architecture of AnchorCoT. We first predict possible intermediate answers as candidate anchors. Then question and candidate anchors are projected into continuous space and we employ anchor ranking algorithm to get optimal permutation of anchors with better logic order. Finally, we employ Chain-of-Thought generation strategy under the guidance of anchors. Final results are extracted from generated contents.

2.2 Chain of Thought

To enhance reasoning capabilities in complex and logical tasks, Wei et al. (2023) introduced the Chain-of-Thought (CoT) framework, which extends in-context learning through a step-by-step reasoning process. CoT demonstrated strong performance in zero-shot logical reasoning. Subsequent works further improve CoT's performance by refining reasoning strategies and prompt designs. In order to mitigate hallucinations in CoT, Self-consistency (Wang et al., 2023b) selects the most consistent answer from multiple CoT paths. Plan-and-Solve Prompting (PS) (Wang et al., 2023a) proposes a planning mechanism to decompose multi-hop questions into stepwise instructions, which are then executed in accordance with the plan. PS+ (Wang et al., 2023a) enhances this approach by requiring large language models (LLMs) to extract relevant variables or descriptive details from the context and formulate a comprehensive plan prior to reasoning. RE2 (Xu et al., 2024) introduces a re-reading strategy, urging LLMs to revisit the question in a second pass to clarify its details.

Despite these advancements, CoT and its derivatives continue to face challenges when solving multi-hop questions due to the complexity of the strategies or difficulties in generating detailed reasoning plans. CoT requires direct and precise guidance to effectively support the reasoning process. In this work, we propose providing potential intermediate answers to LLMs, which directly assist in logical reasoning generation.

3 Methodology

As shown in Figure 2, we introduce AnchorCoT, a novel reasoning framework with anchors as stepwise reasoning guidance. The approach involves three steps: Anchor Prediction, Anchor Ranking and Answer Generation. The model will first predict the intermediate answers as anchors, which will serve as the guidance in reasoning generation. Second, The anchor ranking algorithm will rerank the model in logic order. Finally, the LLMs generates reasoning steps under the guidance of anchors generated in the first step using Chain-of-Thought strategy. Please refer to Appendix E for detailed case study.

3.1 Problem Formulation

Given an input sequence with instruction τ, multi-hop question Q and contexts C, the multi-hop question answering task requires the model to find the answer a to question Q via multi-steps of reasoning. Under Chain-of-Thought strategy, the model gives the answer a by

$$a = \text{LLM}(\tau, C, Q).$$

where a is extracted from LLM response. Our approach optimize reasoning process of LLMs based on the following three steps.

3.2 Step 1: Anchor Prediction

As LLMs latently perform multi-hop reasoning(Yang et al., 2024; Biran et al., 2024), they have the potential to predict correct intermediate

answers. In this step, we use LLMs to predict possible intermediate answers as anchors. In our step 1 prompt, we provide the LLM with detailed instruction τ as generation guidelines and relative contexts C as background information. The inference is in one-shot setting with 1 simple generation example. The generation is formulated as

$$A_q = \{a_0, a_1, \cdots, a_n\} = \text{LLM}(\tau, C, Q).$$

where $a_i, i \in 1, \ldots n$ are possible intermediate answers. As the final answer is counted in intermediate answers, the number of anchors n should not be less than 2. However, we do not force the model to generate the same number of sub-questions in original question Q as LLMs will analyze its hops and decide where to add anchors in reasoning during generation. Please refer to detailed prompt in Appendix A.2.

We also constrain the type of anchors within the range of entities according to Honnibal and Montani (2017). Since we expect the anchors are informative and instructive, the anchors should not be punctuation words, stop words or other symbols that do not contribute to the solution of the question. The anchors (intermediate answers) are usually certain kinds of entities that represents a meaning by itself. Therefore, we limit the choice of anchor types within entities and provide a list of entity descriptions in the prompt. Since the entity constrains resembles to that in Named Entities Recognition(NER), we take advantages of built-in NER types in spaCy package. Full list of legal anchor types in our approach are presented in Appendix B.

3.3 Step 2: Anchor Ranking

In multi-hop reasoning, the order of reasoning steps is crucial to deduction of logical reasoning results. Therefore, the predicted anchors should be reranked before they serve as guidance. Before answer generation, we add anchor ranking step to improve the logic order of predicted anchors. As anchor candidate set A_q is provided in previous step, we expect to find an optimal permutation $A_q^{'}$ that illustrates the logic order of reasoning.

Our ranking algorithm is based on iterative feature of solving nested multi-hop question step by step. For example, in the question *Who is the CEO of the record label "Let Me Love You Tonight"'s performer belongs to?*, we first answer **current question** with the performer of "Let Me Love You Tonight" and fill **current answer** to original question in order to get **next question**: *Who is the CEO of the record label Pure Prairie League belongs to?*. This process will continue to iterate until the original question is solved. Note that the questions during the process is different from decomposed sub-questions.

There are three items involved in each reasoning step: current question q_{current}, answer to atom sub-question a_{current} and next question q_{next}. Suppose the answer to q_{next} is a_{next}, our ranking algorithm follows the iterative process and consists of three steps: (1) find next question q_{next} using a_{current} and q_{current}, (2) find a_{next} using q_{next}, (3) update a_{current} and q_{current} with a_{next} and q_{next}. In our approach, the current question q_{current} is initialized with original question Q and a_{current} is initialized by following step (2) and (3). The order of reranked anchors is the order of a_{next} suggested by the algorithm along with iterative process.

In our implementation, we illustrate the iterative process aforementioned using text embeddings. Inspired by Wieting et al. (2016), we first split input text into words, excluding punctuations and stop words. Then each word will be encoded by LLMs and we use the hidden state of last layer as embeddings. Final embedding $\boldsymbol{e}$ of text is the averaging of all words embeddings. The embedding of text t is given by

$$\boldsymbol{e}(t) = \frac{1}{n}\sum_{i=1}^{n} \text{LLM}(w_i), w_i \in t.$$

where $w_i, i \in 0, 1, \ldots n$ are words in inputs (anchor or question). Under the assumption of TransE (Bordes et al., 2013), we use the following estimation to get the embedding of next question q_{next}:

$$\boldsymbol{e}(q_{\text{current}}) + \boldsymbol{e}(a_{\text{current}}) \approx \boldsymbol{e}(q_{\text{next}}).$$

where $\boldsymbol{e}(\cdot)$ denotes their embeddings in continuous space. According to Yang et al. (2024), LLMs latently reason step by step when answering multi-hop questions. And LLMs first realize the answer of atom sub-question given multi-hop question Q with nested sub-questions. Therefore, the LLM is more likely to answer of the atom sub-question rather than answers that required further steps of reasoning. Based on the finding, we match the next answer a_{next} by getting the nearest candidate anchor to next question q_{next} as next answer a_{next}.

The a_{next} is given by

$$a_{\text{next}} = \arg\min_{a_k} \|\boldsymbol{e}(q_{\text{next}}) - \boldsymbol{e}(a_k)\|_2 \approx \arg\min_{a_k} \|\boldsymbol{e}(a_{\text{current}}) + \boldsymbol{e}(q_{\text{current}}) - \boldsymbol{e}(a_k)\|_2.$$

where $a_k \in A_q, k \in \{1, 2, \ldots, n\}$. And the algorithm will iterate unless there is no anchor candidates left in candidate set Aq. We use $A_q^{'}$ to denote ranked anchors, which is the collection of a_{next} in order. Our ranking algorithm is illustrated in Appendix C.

3.4 Step 3: Answer Generation

In the last stage, we add anchors into CoT prompt and ask LLMs to generate with the guidance of anchors. The final answer is given by

$$a_{\text{AnchorCoT}} = \text{LLM}(\tau^{'}, C, Q, A_q^{'}).$$

where LLMs are given instruction $\tau^{'}$, multi-hop question Q and contexts C. Notice that we employ Chain-of-Thought strategy to facilitate answer generation. The LLM needs to connect the anchors with logic implied in the question and generate reasoning chain in response to anchors provided. For answer extraction, we ask LLMs to wrap the answer $a_{\text{AnchorCoT}}$ in bracket for extraction. Please refer to detailed prompt of answer generation in Appendix A.2.

4 Experiments

4.1 Datasets and Hyperparameters

We test our approach on three logical reasoning datasets, including HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020) and MuSiQue-Answer (Trivedi et al., 2022). Specifically, we choose HotpotQA dev(full wiki) set, 2WikiMultiHopQA dev set and Answer subset in MuSiQue, in which the multi-hop questions are answerable by reasoning on given context. Our approach will be compared with four categories of approaches, including instruction prompting, majority voting, reasoning path searching and domain-specific approaches.

We test our approach on both open-source model and GPT-4o. For open-source models, we use Qwen2.5-Instruct 7B and 14B models. Unless otherwise stated, the numbers reported in the table is average value over 3 round of generation. The generation temperature k is set to 0.3 and maximum generation tokens is set to 128. For inference accuracy, we use half precision (float16) for time efficiency. Experiment results of Instruction Prompting, Majority Voting and Reasoning Path Searching Approaches are the averaging of 3 runs for data stability. We use Exact Match and F1 score metrics in main experiments. Additionally, we use hit@k to evaluate ranking accuracy of anchors in ablation study. Please refer to Appendix D for more implementation details.

4.2 Baselines

To demonstrate the superiority of our approach compared to existing prompting methods, we choose approaches in three categories as baselines:

Instruction Prompting and Majority Voting Approaches For Chain-of-Thought (Wei et al., 2023) approach, we add *Solve the question step by step* in the beginning of instruction. Self-consistency (Wang et al., 2023b) approach use majority voting strategy to select answer from candidates. This approach generates k reasoning chains through Chain-of-Thought reasoning and pick up the answer from answer candidates with highest frequency. Considering time consumption, we follow the practice instruction in Wang et al. (2023b) and set $k = 3$ instead of best $k = 40$. Plan-and-Solve strategy (Wang et al., 2023a) first ask the model to generate a detailed and executable plan before reasoning. The plan indicates important reasoning steps and might involve if-else clause to guide the LLM to enter different reasoning branches. Re-reading strategy (Xu et al., 2024) requires the model to process questions twice before reasoning chain generation. The approach enhances the understanding of the question and capture missed information in the first reading pass. We implement RE2 by adding statement *Read the question again* in instruction prompt. All the baselines adopts 1-shot setting.

Reasoning Path Searching Approaches Tree of Thoughts (ToT)(Yao et al., 2023) is a framework that models decision-making as a tree structure, where each node represents the outcome of an intermediate reasoning step. On each step, the ToT evaluates child nodes situations and selects and expands the top promising child node situations until ToT reach the final answer. Graph of Thoughts (GoT)(Besta et al., 2024) extends ToT by adding refining and backtracking. GoT aggregates intermediate thoughts into a graph or loop over a thought to refine it before reaching the final answer. These approaches is designed to search reasoning path on

Model	Methods	MuSiQue-Ans		HotpotQA		2WikiMHQA	
		EM	F1	EM	F1	EM	F1
Qwen2.5-7B	vanilla	9.50	14.71	28.77	34.11	12.92	18.05
	CoT (Wei et al., 2023)	18.49	24.37	50.65	57.59	34.56	40.73
	CoT-SC@3 (Wang et al., 2023b)	21.31	28.05	**56.46**	64.05	36.30	41.07
	PS (Wang et al., 2023a)	19.20	24.93	53.20	62.74	35.89	42.62
	RE2 (Xu et al., 2024)	19.11	25.05	55.69	63.80	35.20	43.27
	AnchorCoT w/o ranking	21.43	28.92	56.09	64.45	36.23	43.69
	AnchorCoT w/ ranking	**21.84**	**29.44**	56.22	**65.39**	**36.65**	**44.58**
Qwen2.5-14B	vanilla	12.54	20.27	30.12	37.47	20.19	30.27
	CoT	22.67	29.33	52.67	60.78	39.89	48.47
	CoT-SC@3	26.38	36.24	53.27	61.23	41.75	50.82
	PS	24.80	33.30	55.50	65.02	48.19	56.78
	RE2	28.38	36.46	55.89	65.34	53.25	61.55
	AnchorCoT w/o ranking	**28.51**	36.23	56.54	65.15	53.47	61.78
	AnchorCoT w/ ranking	28.22	**36.58**	**57.08**	**66.34**	**54.27**	**63.50**
GPT-4o	vanilla	25.41	31.24	50.75	60.20	40.28	48.45
	CoT	30.61	40.80	51.23	62.10	45.92	54.63
	CoT-SC@3	30.82	41.28	53.77	64.37	46.59	56.74
	PS	31.47	41.28	52.27	63.27	46.74	56.78
	RE2	30.33	40.93	52.35	**65.74**	46.16	55.80
	AnchorCoT w/o ranking	**41.83**	**53.31**	54.36	64.77	54.02	63.47
	AnchorCoT w/ ranking	41.38	53.00	**54.54**	65.19	**54.42**	**63.86**

Table 1: Results of Instruction Prompting approaches and AnchorCoT approach.

Methods	MuSiQue-Ans		HotpotQA		2WikiMHQA	
	EM	F1	EM	F1	EM	F1
ToT (Yao et al., 2023)	21.47	28.54	**56.50**	65.00	36.15	41.02
GoT (Besta et al., 2024)	20.84	27.02	54.52	64.06	**36.67**	42.70
AnchorCoT w/o ranking	21.43	28.92	56.09	64.45	36.23	43.69
AnchorCoT w/ ranking	**21.84**	**29.44**	56.22	**65.39**	36.65	**44.58**

Table 2: Results of Reaoning Path Searching approaches and AnchorCoT approach using Qwen2.5-7B-Instruct model.

data structure, such as tree and graph, in order to generate a reliable reasoning path.

Domain-specific Approaches (Huang et al., 2024) propose Prompting Explicit and Implicit(PEI) knowledge framework, which integrates both explicit, implicit knowledge and question types for multi-hop QA. (Zhao et al., 2023) introduces Verify-and-Edit framework for CoT prompting, which seeks to increase prediction factuality by post-editing reasoning chains. (Li and Du, 2023) extracts semantic graphs through one or multi-step prompting, then generate reasoning chain based on relations in semantic graph. (Ding et al., 2019) builds a cognitive graph in an iterative process by coordinating an implicit extraction module and an explicit reasoning module, giving both reasoning process and results.

4.3 Main results

Comparison with Instruction Prompting and Majority Voting Approaches The experiment results of HotpotQA, 2WikiMultiHopQA and MuSiQue-Ans datasets on Qwen2.5-7B, GPT-4o are presented in Table 1. For MuSiQue-Answer dataset, AnchorCoT with ranking algorithm shows an average improvement of approximately 4% on Qwen2.5-7B and an 12% increase on GPT-4o. Our approach, even without ranking algorithm, shows superior performance over existing Chain-of-Thought approaches by a large margin. We notice that for smaller LLMs (7B), Self-consistency CoT showcased competitive performance over other baselines on MuSiQue-Answer dataset, yet still surpassed by our approach by 0.5% on Exact Match and 1.4% on F1 score. For HotpotQA dataset. our method still showcases competitive performance

compared to existing CoT approaches. On AnchorCoT with anchor ranking algorithm, our approach surpasses CoT by approximately 5% on Exact Match score and 8% on F1 score. On 2WikiMultiHopQA dataset, our methods lead on both 2 metrics and have a gain margin of 8% on GPT-4o EM metric and F1 metric compared to CoT-SC@3, PS and RE2. The previous instruction prompting approaches suffer from indirect assistance in reasoning chain generation, while our approach provides direct guidance to step-wise generation, giving logical reasoning process and more accurate results.

Comparison with Reasoning Path Searching Approaches In addition to comparison experiments with instruction prompting strategies, we compared our approach with reasoning path finding approaches including Tree-of-Thought (Yao et al., 2023) and Graph-of-Thought (Besta et al., 2024). Results are presented in Table 2. For Tree-of-Thought approach, we adopt BFS-ToT algorithm as searching strategy and set breadth limit $b = 1$, sampling 3 votes at each reasoning step. Our approach demonstrates its superiority over existing planning methods ToT and GoT on the F1 metric. On EM metric, the results of our approach surpass that of ToT and GoT on MuSiQue Dataset and hold runner-up in HotpotQA and 2WikiMultiHopQA EM metric. Considering the computation consumption in ToT and GoT approaches, our method presents both better accuracy and efficiency.

Comparison with Domain-specific Approaches We also conduct comparison experiments on domain-specific approaches. Results are shown in Table 3. We use the results reported in work Huang et al. (2024); Ding et al. (2019); Li and Du (2023); Zhao et al. (2023). Our approach has a clear increase over domain-specific methods. To be more specific, our approach has over 5% increase in Exact match score on both datasets compared to approaches using semantic graphs or relation extractions. Considering the complexity of relations in previous approaches, LLMs might be confused given irrelevant relation pairs, thus giving incorrect answers. In contrast, our approach provides accurate guidance for LLMs in prompt, giving clear guidance in reasoning chain generation.

4.4 Ablation Studies

The Potential of Anchors on Multi-hop Reasoning We conduct experiments on the potential of anchors in improving the reasoning ability. We test our approach on MuSiQue-Ans and 2WikiMultiHop datasets by providing ground-truth anchors with correct ranking for LLM Reasoner. The results are presented in Table 4. There are significant improvements on both E1 and F1 metrics using ground-truth anchors with correct ranking, which indicates that by taking advantage of the freedom to add control to generated anchors and by improving the generated guidance(anchors), the reasoning performance will be better.

Methods	HotpotQA		2WikiMHQA	
	EM	F1	EM	F1
iCAP	47.02	-	42.80	-
PCL	49.27	-	46.03	-
PEI	49.91	-	47.32	-
CogQA	12.20	35.30	-	-
SG-One prompt	-	27.02	47.00	57.90
SG-Multi prompt	-	59.25	49.00	60.10
CoT-SC+VE (Wiki.)	-	-	33.10	-
CoT-SC+VE (Dataset)	-	-	37.20	-
Ours w/ ranking	**54.54**	**65.19**	**54.42**	**63.86**

Table 3: Results of Domain-specific approaches and AnchorCoT approach using GPT-4o model. Domain-specific results are presented according to results in their references. "-" denotes the metric is not available according to their references.

Anchors	HotpotQA		2WikiMHQA	
	EM	F1	EM	F1
w/o ranking	21.43	28.92	36.23	43.69
w/ ranking	21.84	29.44	36.65	44.58
Ground truth	**75.41**	**74.33**	**80.02**	**82.16**

Table 4: Results of Exact Match and F1 metric on HotpotQA and 2WikiMultiHopQA datasets using ground truth anchors with correct order as reranked anchors.

Efficacy of Implicit anchors In multi-hop questions, intermediate answers may not be explicitly stated in the provided contexts, which are called implicit anchors. Take, for instance, the question "What is the month following the 2025 US Presidential Inaugural Ceremony?" for example. Here, the model should intuitively know that the month following January is February, even if this information isn't directly supplied in the context. This is where implicit anchors become crucial, as they can effectively guide the CoT reasoning process step by step. These implicit anchors provide additional guidance beyond the original context, which is vital for the reasoning steps.

Model	Ranking Algorithm	MuSiQue-Ans				2WikiMHQA			
		hit@2	hit@3	hit@5	hit@10	hit@2	hit@3	hit@5	hit@10
Qwen2.5-7B	w/o	18.73	26.93	35.37	40.78	20.18	28.91	37.59	39.83
	w/	**19.81**	**27.63**	**35.97**	**41.18**	**20.47**	**28.94**	**37.67**	**40.23**
Qwen2.5-14B	w/o	16.02	21.75	**29.40**	**41.91**	**19.32**	27.65	33.86	44.69
	w/	**16.37**	**23.14**	27.52	41.05	19.04	**27.77**	**35.08**	**45.88**
GPT-4o	w/o	28.55	41.19	51.47	**52.30**	27.30	39.01	47.17	**51.43**
	w/	**30.34**	**42.25**	**51.53**	**52.30**	**29.06**	**39.87**	**47.87**	**51.43**

Table 5: Results of anchor hit ratio on MuSQue-Ans dev and WikiMultiHopQA dataset. Both of the datasets provide intermediate answers to sub-questions. w/o denotes without ranking algorithm.

CoT	Reranking	MuSiQue-Ans		HotpotQA		2WikiMHQA	
		EM	F1	EM	F1	EM	F1
w/o	w/o	15.02	23.03	46.23	53.68	20.80	29.01
w/o	w/	14.96	23.27	38.97	46.52	20.07	26.77
w/	w/o	18.25	25.45	55.94	64.05	20.30	28.32
w/	w/	**21.84**	**29.44**	**56.22**	**65.39**	**36.65**	**44.58**

Table 6: Ablation study on ranking algorithm and Chain-of-Thought generation strategy.

In this section, we evaluate the efficacy of implicit anchors by using only explicit anchors for reasoning guidance. Results are presented in Table 7. Results demonstrate that introducing implicit anchors can help to improve the performance.

Implicit Anchor	MuSiQue-Ans		HotpotQA		2WikiMHQA	
	EM	F1	EM	F1	EM	F1
w/o	18.47	24.80	53.50	61.47	31.34	39.71
w/	**21.84**	**29.44**	**56.22**	**65.39**	**36.65**	**44.58**

Table 7: Ablation study on the efficacy of implicit predicted anchors. The LLMs generate the final answer only with anchors that appears in the context under "w/o Implicit Anchor" setting.

Effectiveness of Anchor Ranking Algorithm As reasoning process depends heavily on the logic sequence, we add anchor ranking before they are added in the final prompt. In addition to the evaluation of anchor ranking using final accuracy as metric, we further investigate anchor prediction using hit@k metric. Hit@k denotes the ratio of successfully predicted anchors in top k predictions to the number of all ground truth anchors. We test hit ratio on Qwen2.5-7B, 14B and GPT-4o models. As shown in Table 1, the results of Anchor-CoT reasoning with ranking algorithm shows superior performances by 1-2% on EM and F1 metrics. Specifically, we investigate into top 10 predicted anchors, which is actually larger than the number of reasoning steps in these datasets.

Results are presented in Table 5. The result demonstrates that anchor ranking algorithm improves the order of anchors on different hops, thus facilitates reliable reasoning chain generation. Our experiment on 7B and 14B model shows that even a small model has the potential to predict logical anchor correctly. For GPT-4o model, our ranking algorithm improves hit@2 metric by a margin of 1.8%, while the increase shrink to 1.0% in hit@3. The hit ratio eventually maintains the same as the approach without ranking algorithm at hit@10. Specifically, we can expect better prediction of anchors in first steps of reasoning chain. We notice that hit@2 of Qwen2.5 7B model is larger than that of 14B model, while for overall anchor prediction (hit@10), larger model has a better performance.

We also compare our algorithm with other ranking algorithms. We compare our results with LLM as ranker and RankZephyr(Pradeep et al., 2023) ranker. Results are presented in Table 8. Th results demonstrate that our method outperforms the LLM Ranker across all hit ratios. In comparison with RankZephyr, our strategy demonstrates superior results in hit@2 and hit@3, although RankZephyr outperforms our algorithm in hit@5 and hit@10 marginally. Given that the majority of multi-hop questions involve fewer than 5 hops (with the

MuSiQue dataset only containing questions up to 4 hops), our ranking approach remains highly competitive.

Methods	hit@2	hit@3	hit@5	hit@10
LLM as Ranker	18.98	25.99	33.56	36.51
RankZephyr	20.01	27.54	**38.46**	**40.54**
Ours	**20.47**	**28.94**	37.67	40.23

Table 8: Hit ratios of different ranking strategy. Conducted on 2WikiiMultiHopQA dataset using Qwen2.5-7B-instruct model. For RankZephyr we use rank-zephyr-7b-v1-full model.

Importance of Chain-of-Thought Answer Generation In the last stage, we add anchors to the prompt and ask LLMs to generate chain-of-thought deduction chain with final answer wrapped in brackets. In this way, the model will take advantage of anchors as step-wise inference guidance and find final answer along with reasoning chain. In order to illustrate the importance of employing chain-of-Thought strategy in answer generation, we force the model to output the answer directly without CoT strategy in our approach. Experiment results are shown in Table 6. Of all four settings, CoT with ranking algorithm yields the best results. For reasoning with CoT without ranking procedure, the overall performance maintains close to that of the previous setting on MuSiQue and HotpotQA datasets, while having poor performance on 2WikiMultiHop dataset. The results demonstrate that Chain-of-Thought generation strategy is crucial to the performance of AnchorCoT. Given a sequence of anchors as deduction guidance, LLMs unfolds their reasoning ability better along with reasoning chain. Otherwise, LLMs will recognize reasoning with anchors as a choice task, in which models choose one of the anchors as the answer without deduction process, thus leading to poor performance, as *w/o CoT w/ ranking* and *w/o CoT w/o ranking* shown in Table 6.

5 Conclusion

In this paper, we introduced AnchorCoT, a novel Chain-of-Thought designed for multi-hop question answering. Our approach facilitates logical reasoning chain generation by providing intermediate answers as step-wise guidance. We conduct extensive experiments on MuSiQue, HotpotQA and 2WikiMultiHopQA datasets on both open-source models and GPT-4o. Our approach exhibits superior performance over other existing multi-hop reasoning methods, including instruction prompting, reasoning path searching and domain-specific methods. We hope our method will be applied in question answering to improve LLM performance in various multi-hop reasoning tasks.

6 Limitations

Our approach has limitations under certain circumstances. First, anchor prediction may not always be accurate, and the predicted anchors might not provide sufficient guidance for effective model reasoning. Second, our model is specifically designed for logical reasoning and relies on external context provided via prompts. It may not perform well in symbolic reasoning tasks or when reasoning depends on internal knowledge without sufficient contextual input. In future work, we plan to investigate methods to improve anchor prediction accuracy and explore approaches for symbolic reasoning that do not rely on explicit prompts.

References

Qiming Bao, Gaël Gendron, Alex Yuxuan Peng, Wanjun Zhong, Neşet Özkan Tan, Yang Chen, Michael Witbrock, and Jiamou Liu. 2023. Assessing and enhancing the robustness of large language models with task structure variations for logical reasoning.

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

Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. 2024. Hopping too late: Exploring the limitations of large language models on multi-hop queries. *Preprint*, arXiv:2406.12775.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin

Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation. *Preprint*, arXiv:2311.17311.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, and Barret Zoph. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

Zheng Chu, Jingchang Chen, Qianglong Chen, Haotian Wang, Kun Zhu, Xiyuan Du, Weijiang Yu, Ming Liu, and Bing Qin. 2024. Beamaggr: Beam aggregation reasoning over multi-source knowledge for multi-hop question answering. *Preprint*, arXiv:2406.19820.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. *ArXiv*, abs/2205.09712.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2694–2703, Florence, Italy. Association for Computational Linguistics.

Sefika Efeoglu and Adrian Paschke. 2024. Retrieval-augmented generation-based relation extraction. *Preprint*, arXiv:2404.13397.

Panagiotis Giadikiaroglou, Maria Lymperaiou, Giorgos Filandrianos, and Giorgos Stamou. 2024. Puzzle solving using reasoning of large language models: A survey. *Preprint*, arXiv:2402.11291.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, and Cyrus Nikolaidis. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *Preprint*, arXiv:2410.05779.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *Preprint*, arXiv:2203.15556.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Guangming Huang, Yunfei Long, Cunjin Luo, Jiaxing Shen, and Xia Sun. 2024. Prompting explicit and implicit knowledge for multi-hop question answering based on human reading process. *Preprint*, arXiv:2402.19350.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Ruosen Li and Xinya Du. 2023. Leveraging structured information for explainable multi-hop question answering and reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6779–6789, Singapore. Association for Computational Linguistics.

Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yuexin Zhang. 2023. Evaluating the logical reasoning ability of chatgpt and gpt-4. *ArXiv*, abs/2304.03439.

Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2024. Multi-hop question answering. *Preprint*, arXiv:2204.09140.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. Rankzephyr: Effective and robust zero-shot listwise reranking is a breeze! *Preprint*, arXiv:2312.02724.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Ferdinand Schlatt, Maik Fröbe, and Matthias Hagen. 2024. Lightning ir: Straightforward fine-tuning and inference of transformer-based language models for information retrieval. *Preprint*, arXiv:2411.04677.

Jiwoong Sohn, Yein Park, Chanwoong Yoon, Sihyeon Park, Hyeon Hwang, Mujeen Sung, Hyunjae Kim, and Jaewoo Kang. 2024. Rationale-guided retrieval augmented generation for medical question answering. *Preprint*, arXiv:2411.00300.

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*, 10:539–554.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *Preprint*, arXiv:2305.04091.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

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

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. *Preprint*, arXiv:1511.08198.

Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, Jian guang Lou, and Shuai Ma. 2024. Re-reading improves reasoning in large language models. *Preprint*, arXiv:2309.06275.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024. Do large language models latently perform multi-hop reasoning? *Preprint*, arXiv:2402.16837.

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

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

Urchade Zaratiana, Nadi Tomeh, Yann Dauxais, Pierre Holat, and Thierry Charnois. 2024. Enrico: Enriched representation and globally constrained inference for entity and relation extraction. *Preprint*, arXiv:2404.12493.

Qingfei Zhao, Ruobing Wang, Yukuo Cen, Daren Zha, Shicheng Tan, Yuxiao Dong, and Jie Tang. 2024. Longrag: A dual-perspective retrieval-augmented generation paradigm for long-context question answering. *Preprint*, arXiv:2410.18050.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. *Preprint*, arXiv:2305.03268.

A Detailed Prompts

A.1 Anchor Prediction Prompt Template

In anchor prediction step, LLMs are required to generate candidate anchors. The type of candidate anchors are limited within entity types introduced in Appendix B. Moreover, we also add constraints on the number of anchors between 2 and 10, because the questions in multi-hop datasets involve 2 questions steps at least. Answers are wrapped in brackets in the output. Prompts are presented in Prompt Template 1.

A.2 Anchor Generation Prompt Template

In anchor generation step, the models are required to solve the question step by step under the guidance of reranked anchors. Prompts are presented in Prompt Template 2.

B Anchor constraints

The legal anchor types in AnchorCoT are presented in Table 9.

C Anchor Ranking Algorithm

The Anchor ranking algorithm is illustrated in Algorithm 1.

Type	Description
PERSON	People, including fictional
GPE	Countries, cities, states
EVENT	Named hurricanes, battles, wars, sports events, etc
FAC	Buildings, airports, highways, bridges, etc
LOC	Non-GPE locations, mountain ranges, bodies of water
PRODUCT	Objects, vehicles, foods, etc. (Not services.)
ORG	Companies, agencies, institutions, etc
NORP	Nationalities or religious or political groups
LANGUAGE	Any named language
LAW	Named documents made into laws
WORK_OF_ART	Titles of books, songs, etc

Table 9: Constrains of anchor types from Named Entity Recognition in SpaCy package.

Algorithm 1: Anchor Ranking Algorithm

Input: Multi-hop question: q; Anchor candidates: A_q, LLM $\mathcal{M}$

Output: Ranked anchors $A_q^{'}$

Get embeddings of q and A_q: $\boldsymbol{e}(q), \boldsymbol{e}(A_q)$.
Initialize question embedding $\boldsymbol{e}_0 \leftarrow \boldsymbol{e}(q)$.
while A_q has candidate(s) **do**
  Initialize next anchor $a_i \leftarrow A_q[0]$
  Initialize score $s \leftarrow -\infty$.
  foreach $\boldsymbol{e}(a_k) \in \boldsymbol{e}(A_q)$ **do**
    if $s < -\|\boldsymbol{e}(a_i) + \boldsymbol{e}(q) - \boldsymbol{e}(a_k)\|_2$ **then**
      $s \leftarrow -\|\boldsymbol{e}(a_i) + \boldsymbol{e}(q) - \boldsymbol{e}(a_k)\|_2$
      $a_i \leftarrow a_k$
  Add a_i to $A_q^{'}$; Remove a_k from A_q.
  $\boldsymbol{e}_0 \leftarrow \boldsymbol{e}_0 + \boldsymbol{e}(a_k)$.
return $A_q^{'}$.

D Implementation Details

Number of anchors predicted We first predict potential intermediate answers as reasoning anchors. The number of anchors vary from 2 to 10 in order to avoid direct answering and complicated reasoning guidance.

LLM Output format In anchor prediction step, the predicted anchors are wrapped in brackets and are separated by commas, as shown in Appendix E. In answer generation step, the final answer is also wrapped in brackets. We extract the content wrapped in the last brackets as the final answer. We use regular expression for anchor extraction and final answer extraction.

E Case Study

We pick up one example using GPT-4o on MuSiQue-Ans dataset to illustrate our framework, as shown in Figure E and Figure E. Responses from LLM are **bold and colored in blue**. Anchors predicted by anchor prediction module are reranked by anchor ranking algorithm and added to prompt in answer generation stage.

Prompt Template 1: Anchor prediction

Instruction:
1. Given a question and context paragraphs, identify and list explicit and implicit anchors that could be useful in answering the question step-by-step.
2. The anchors can include: <*anchor types*>
3. Each identified anchor must be wrapped in brackets ().
4. Only output the final list of anchors without explanations or intermediate commentary.
5. The number of anchors should not be greater than 10 and should not be less than 2.
6. All the anchors should be useful in reasoning step. Do not list irrelevant anchors.

Example:
Context: <*statement 1*> , <*statement 2*>, <*statement 3*> ...
Question: <*example question*>
Anchors: <*anchor 1*>, <*anchor 2*>, . . .

Your Task:
Use the format and methodology in the example to solve the following question based on the provided context.
Context: <*statement 1*> , <*statement 2*>, <*statement 3*> ...
Question: <*question*>
Anchors:

Prompt Template 2: Anchor generation

Instruction:
1. Given key anchors of the question and context paragraphs, generate the answer to the question by thinking step by step.
2. The anchors are listed in terms of the order of solving the question step by step.
3. Answer must be wrapped in brackets ().

Example:
Context: <*statement 1*> , <*statement 2*>, <*statement 3*> ...
Question: <*example question*>
Anchors: <*anchor 1*>, <*anchor 2*>, . . .
Answer: <*Chain-of-Thought Answer with final answer wrapped by bracket ()*>

Your Task:
Use the format and methodology in the example to solve the following question based on the provided context.
Context: <*statement 1*> , <*statement 2*>, <*statement 3*> ...
Question: <*question*>
Anchors: <*anchor 1*>, <*anchor 2*>, . . .
Answer:

Example 1: Anchor prediction

Instruction:
1. Given a question and context paragraphs, identify and list explicit and implicit anchors that could be useful in answering the question step-by-step.
2. The anchors can include: People, including fictional, Countries, cities, states, Named hurricanes, battles, wars, sports events, etc, Buildings, airports, highways, bridges, etc, Non-GPE locations, mountain ranges, bodies of water, Objects, vehicles, foods, etc. (Not services.), Companies, agencies, institutions, etc, Nationalities or religious or political groups, Any named language, Named documents made into laws, Titles of books, songs, etc.
3. Each identified anchor must be wrapped in brackets ().
4. Only output the final list of anchors without explanations or intermediate commentary.
5. The number of anchors should not be greater than 10 and should not be less than 2.
6. All the anchors should be useful in reasoning step. Do not list inrelevant anchors.

Example:
Context: 1. In Indiana, alcohol may be sold only to those 21 years of age or older during the hours 7 a.m. to 3 a.m.
2. Sanford H. Calhoun High School is a public high school located in Merrick, New York.
3. Greenfield-Central High School is a secondary school (grades 9-12) located in the city of Greenfield, Indiana.
4. In high school, Jess Fink began reading comics and then manga. She cites Molly Keily and Art Spiegelman as influences.
Question: What time does the state where Greenfield-Central High is stop selling booze?
Anchors: (Indiana), (3 a.m.), (3 A.M.)

Your Task:
Use the format and methodology in the example to solve the following question based on the provided context.
Context:
1. The dynasty regrouped and defeated the Portuguese in 1613 and Siam in 1614. It restored a smaller, more manageable kingdom, encompassing Lower Myanmar, Upper Myanmar, Shan states, Lan Na and upper Tenasserim. The Restored Toungoo kings created a legal and political framework whose basic features would continue well into the 19th century. The crown completely replaced the hereditary chieftainships with appointed governorships in the entire Irrawaddy valley, and greatly reduced the hereditary rights of Shan chiefs. Its trade and secular administrative reforms built a prosperous economy for more than 80 years. From the 1720s onward, the kingdom was beset with repeated Meithei raids into Upper Myanmar and a nagging rebellion in Lan Na. In 1740, the Mon of Lower Myanmar founded the Restored Hanthawaddy Kingdom. Hanthawaddy forces sacked Ava in 1752, ending the 266-year-old Toungoo Dynasty.

2. Somalis (Somali: Soomaali) are an ethnic group inhabiting the Horn of Africa (Somali Peninsula). The overwhelming majority of Somalis speak the Somali language, which is part of the Cushitic branch of the Afro-Asiatic family. They are predominantly Sunni Muslim. Ethnic Somalis number around 16-20 million and are principally concentrated in Somalia (around 12.3 million), Ethiopia (4.6 million), Kenya (2.4 million), and Djibouti (464,600), with many also residing in parts of the Middle East, North America and Europe.

3. Djibouti ((listen) jih-BOO-tee; Afar: Yibuuti, French: Djibouti, Somali: Jabuuti, officially the Republic of Djibouti) is a country located in the Horn of Africa. It is bordered by Eritrea in the north, Ethiopia in the west and south, and Somalia in the southeast. The remainder of the border is formed by the Red Sea and the Gulf of Aden at the east. Djibouti occupies a total area of 23,200 km2 (8,958 sq mi). The state of Djibouti is predominantly inhabited by two ethnic groups, the Somali and the Afar people, the Somalis being the major ethnic group of the country.
...
20. More than 20% of the world's population is Muslim. Current estimates conclude that the number of Muslims in the world is around 1,5 billion. Muslims are the majority in 49 countries, they speak hundreds of languages and come from diverse ethnic backgrounds. Major languages spoken by Muslims include Arabic, Urdu, Bengali, Punjabi, Malay, Javanese, Sundanese, Swahili, Hausa, Fula, Berber, Tuareg, Somali, Albanian, Bosnian, Russian, Turkish, Azeri, Kazakh, Uzbek, Tatar, Persian, Kurdish, Pashto, Balochi, Sindhi and Kashmiri, among many others.

Question: How were the people that the Somali Muslim Ajuran Empire made coins to proclaim independence from, expelled from the country where Mohinga is eaten?

Anchors: **(Portuguese), (Somali Muslim Ajuran Empire), (Mohinga), (Myanmar), (Christian Reconquista)**

Example 2: Answer generation

Instruction:
1. Given key anchors of the question and context paragraphs, generate the answer to the question by thinking step by step.
2. The anchors are listed in terms of the order of solving the question step by step.
3. Answer must be wrapped in brackets ().

Example:
Context: 1. In Indiana, alcohol may be sold only to those 21 years of age or older during the hours 7 a.m. to 3 a.m.
2. Sanford H. Calhoun High School is a public high school located in Merrick, New York.
3. Greenfield-Central High School is a secondary school (grades 9-12) located in the city of Greenfield, Indiana.
4. In high school, Jess Fink began reading comics and then manga. She cites Molly Keily and Art Spiegelman as influences.
Question: What time does the state where Greenfield-Central High is stop selling booze?
Anchors: (Indiana), (3 a.m.), (3 A.M.)
Answer: Greenfield-Central High School is located in Indiana. In Indiana, alcohol is stopped selling after 3 a.m. Therefore, the answer is (3 a.m.).

Your Task:
Use the format and methodology in the example to solve the following question based on the provided context.
Context:
1. The dynasty regrouped and defeated the Portuguese in 1613 and Siam in 1614. It restored a smaller, more manageable kingdom, encompassing Lower Myanmar, Upper Myanmar, Shan states, Lan Na and upper Tenasserim. The Restored Toungoo kings created a legal and political framework whose basic features would continue well into the 19th century. The crown completely replaced the hereditary chieftainships with appointed governorships in the entire Irrawaddy valley, and greatly reduced the hereditary rights of Shan chiefs. Its trade and secular administrative reforms built a prosperous economy for more than 80 years. From the 1720s onward, the kingdom was beset with repeated Meithei raids into Upper Myanmar and a nagging rebellion in Lan Na. In 1740, the Mon of Lower Myanmar founded the Restored Hanthawaddy Kingdom. Hanthawaddy forces sacked Ava in 1752, ending the 266-year-old Toungoo Dynasty.

2. Somalis (Somali: Soomaali) are an ethnic group inhabiting the Horn of Africa (Somali Peninsula). The overwhelming majority of Somalis speak the Somali language, which is part of the Cushitic branch of the Afro-Asiatic family. They are predominantly Sunni Muslim. Ethnic Somalis number around 16-20 million and are principally concentrated in Somalia (around 12.3 million), Ethiopia (4.6 million), Kenya (2.4 million), and Djibouti (464,600), with many also residing in parts of the Middle East, North America and Europe.

3. Djibouti ((listen) jih-BOO-tee; Afar: Yibuuti, French: Djibouti, Somali: Jabuuti, officially the Republic of Djibouti) is a country located in the Horn of Africa. It is bordered by Eritrea in the north, Ethiopia in the west and south, and Somalia in the southeast. The remainder of the border is formed by the Red Sea and the Gulf of Aden at the east. Djibouti occupies a total area of 23,200 km2 (8,958 sq mi). The state of Djibouti is predominantly inhabited by two ethnic groups, the Somali and the Afar people, the Somalis being the major ethnic group of the country.
...
20. More than 20% of the world's population is Muslim. Current estimates conclude that the number of Muslims in the world is around 1,5 billion. Muslims are the majority in 49 countries, they speak hundreds of languages and come from diverse ethnic backgrounds. Major languages spoken by Muslims include Arabic, Urdu, Bengali, Punjabi, Malay, Javanese, Sundanese, Swahili, Hausa, Fula, Berber, Tuareg, Somali, Albanian, Bosnian, Russian, Turkish, Azeri, Kazakh, Uzbek, Tatar, Persian, Kurdish, Pashto, Balochi, Sindhi and Kashmiri, among many others.

Question: How were the people that the Somali Muslim Ajuran Empire made coins to proclaim independence from, expelled from the country where Mohinga is eaten?
Anchors: (christian reconquista),(mohinga),(portuguese),(somali muslim ajuran empire),(myanmar)

Answer: **The context indicates that the Portuguese were expelled during the Christian Reconquista from the region that eventually became independent. Mohinga is associated with Myanmar, and the Somali Muslim Ajuran Empire is noted for its defiance of the Portuguese. Therefore, the answer is: The Portuguese, who were expelled during the Christian Reconquista, are related to the history of the region. Mohinga is a dish from Myanmar, and the Somali Muslim Ajuran Empire made coins to proclaim independence from the Portuguese. Thus, the Portuguese were not expelled from Myanmar but from the region during that historical context. Therefore, the answer is (Myanmar).**