Following Occam's Razor: Dynamic Combination of Structured Knowledge for Multi-Hop Question Answering using LLMs

Wei Chen[1], Zhi Zheng[1*], Lili Zhao[1], Huijun Hou[2], Tong Xu[1]

[1]University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence, [2]NIO
{chenweicw, liliz}@mail.ustc.edu.cn,
{zhengzhi97, tongxu}@ustc.edu.cn, huijun.hou@nio.com

Abstract

Multi-hop question answering is a challenging task that requires capturing information from different positions in multiple documents. Recently, several methods propose to enhance Large Language Models (LLMs) by incorporating structured knowledge, aiming to grasp key information for solving this task. Despite certain achievements, they still face the following challenges: 1) The neglect of text-based reasoning capabilities. 2) Information redundancy between text and triples. 3) Information loss during structured knowledge extraction. To solve the above challenges, in this paper, we propose **D**ynamic **C**ombination of **S**tructured **K**nowledge (**DCSK**), a novel framework for integrating text-based and triple-based paradigms. Following Occam's Razor, DCSK dynamically determine the necessity of structured knowledge by the designed multi-faceted evaluation, which systematically assess the correctness, clarity, and informativeness of text-based prediction. For questions that require structured knowledge, we develop an iterative fact refiner that screens for question-relevant triples, verifies their factual adequacy, and thereby effectively excludes irrelevant and redundant information. Furthermore, based on the verification, we construct an adaptive knowledge reasoner that dynamically adjusts the need for text supplementation, thus mitigating the information deficiency in selected triples. Extensive experiments on three MHQA datasets demonstrate the efficiency and effectiveness of DCSK [1].

1 Introduction

Multi-Hop Question Answering (MHQA) is a challenging task in the field of natural language processing, which needs to capture relevant information from multiple locations of the documents for reasoning (Yang et al., 2024b). For example, to answer "*Arthur Marryatt is a member of a sports association that has whom as president?*", it is essential to first identify the specific sports association to which "Arthur Marryatt" belongs, and subsequently ascertain the identity of its president. This process necessitates a certain level of combinatorial reasoning capability (Shi et al., 2024).

As large language models (LLMs) have demonstrated excellent text understanding and reasoning capabilities (Xu et al., 2025; Liu et al., 2024a; Zhao et al., 2025b; Yuan et al., 2025), they have proven to be indispensable in the MHQA task (Khalifa et al., 2023; Zhong et al., 2023). Recently, several studies propose to extract structured knowledge from documents by constructing triples (Panda et al., 2024; Li and Du, 2023; Liu et al., 2024b). This triple-based paradigm aims to assist LLMs in comprehending the relationships between various entities and facilitating more reliable and interpretable reasoning.

Although the certain progress made, it still faces the following challenges: (1) **The neglect of text-based reasoning capabilities.** Previous work assume that all questions must be addressed by integrating multi-step processed triples, which undoubtedly necessitates substantial computing resources. However, even when only the original unstructured text is available, LLMs possess sufficient basic reasoning capabilities to accurately answer certain questions, especially with the assistance of some plugin methods (such as Chain of Thought (CoT) (Wei et al., 2022)). As shown in Figure 1 (a), the majority of questions can be accurately addressed by both paradigms. In accordance with Occam's Razor, this subset of data does not necessitate additional complex processing. Furthermore, it is evident that a considerable number of problems remain solvable by only one paradigm. This indicates that if the two paradigms are appropriately integrated, it is possible to not only conserve unnecessary computing resources but also enhance performance to some extent.

(2) **Information redundancy between text and**

*Corresponding author
[1]https://github.com/fanshu6hao/DCSK

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 17942–17956
November 4-9, 2025

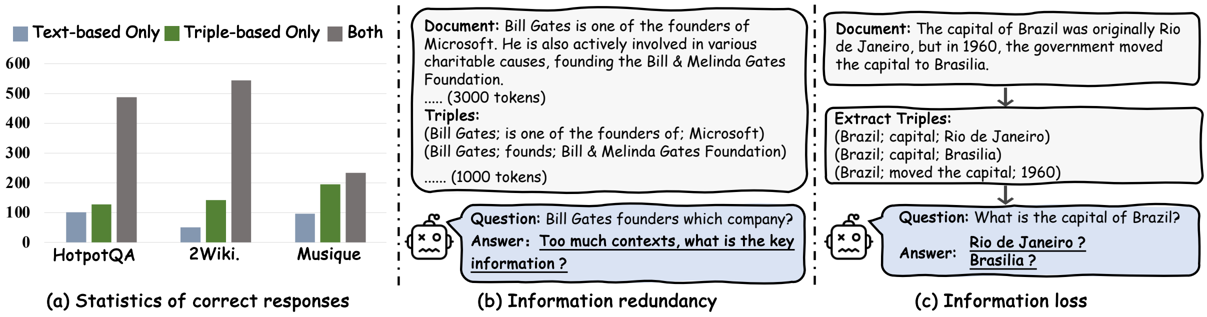

Figure 1: (a) The distribution of the number of correct responses for the two paradigms across different datasets. (b) The information redundancy between text and triples, which hinders LLMs to identify critical problem-solving information. (c) The information loss occurring in extracting triples, which prevents LLMs from fully comprehending implicit background information.

triples. Problem-solving processes typically require only a limited amount of critical information. Nevertheless, when triples are used as supplementary input, as depicted in Figure 1 (b), they introduce a significant volume of redundant information that overlaps with the text. This not only increases the length of the context but also potentially distracts the focus of LLMs. (3) **Information loss during structured knowledge extraction.** In the process of extracting triples, certain implicit background information is inevitably lost. When triples are employed as the sole input, LLMs struggle to acquire the comprehensive background knowledge necessary to address the question. As shown in Figure 1 (c), it causes LLMs to concentrate on the explicit facts regarding the capital within the triples, which may impede them to accurately infer the current location of the capital.

To address the above challenges, we propose **D**ynamic **C**ombination of **S**tructured **K**nowledge (**DCSK**), a novel framework for integrating text-based and triple-based paradigms that both increases efficiency and the reasoning capabilities of the LLMs. Specifically, we initially employ the foundational reasoning capabilities of LLMs for text-based inference. Subsequently, we leverage a multi-faceted evaluation to systematically assess the correctness, clarity, and informativeness of the predictions. Through this mechanism, we are able to identify the questions that genuinely necessitate the structured knowledge. Within the triple-based paradigm, given that answering questions requires only core and critical steps, we design an iterative fact refiner to eliminate irrelevant and redundant information. This process initially selects triples relevant to the question and subsequently verifies the factual adequacy of the selected triples to ascertain whether they provide sufficient information. If verification fails, the process repeats until it passes or the predefined maximum number of attempts is reached. Finally, we construct an adaptive knowledge reasoner, that dynamically determines whether to incorporate the original text as supplementary input based on the verification results, thereby mitigating the issue of information insufficiency in selected triples. Our main contributions are as follows:

- We propose DCSK, a novel framework that dynamically integrates text-based and triple-based paradigms, which not only enhances efficiency but also significantly improves the reasoning capabilities of LLMs.
- We design an iterative fact refiner to eliminate irrelevant and redundant information of triples. Additionally, we construct an adaptive knowledge reasoner to adaptively adjusts the need for text supplementation, thus alleviating the information deficiency in selected triples.
- We conduct extensive experiments on three MHQA datasets (HotpotQA, 2WikiMultihopQA, and Musique), which demonstrate the efficiency and effectiveness of DCSK.

2 Related Work

Multi-Hop Question Answering (MHQA). With the rapid advancement of deep learning technologies, existing methods have demonstrated exceptional performance in addressing direct questions (Yuan et al., 2024; Zhao et al., 2024, 2025a). Consequently, there is a growing interest in more challenging tasks such as Multi-Hop Question Answering (Lan et al., 2021; Zhong et al., 2023).

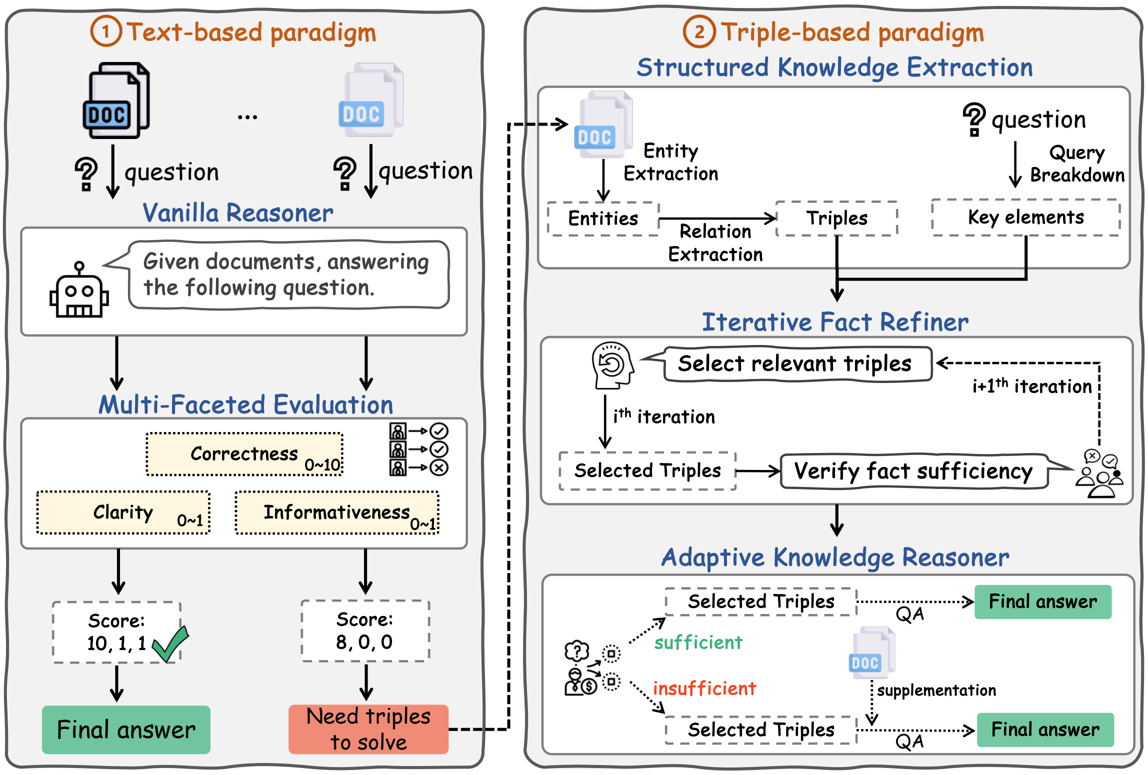

Figure 2: The overall framework of DCSK, which dynamically integrating text-based and triple-based paradigms.

The traditional approaches encompass various paradigms, such as: 1) Utilizing graph neural networks for encoding and reasoning to address questions (Qiu et al., 2019; Fang et al., 2020). 2) Decomposing the question into sub-questions and addressing them sequentially (Fu et al., 2021; Perez et al., 2020). 3) Iteratively processing different content within hierarchical documents to tackle complex questions (Sun et al., 2021). 4) Employing a graph-based iterative retrieval method for inferring pathways of reasoning (Asai et al., 2020).

Recent studies have sought to leverage the powerful reasoning capabilities of LLMs to address complex questions (Wang et al., 2024). For instance, Press et al. (2023) investigated the capability of LLMs in performing combinatorial reasoning tasks, further enhancing the chain of thought through self-ask. Zhang et al. (2024) proposed an end-to-end beam retriever to expand the search space and reduce the risk of omitting relevant passages, thereby enhancing the responses. Shi et al. (2024) repeated the processes of prompting LLMs generate sub-questions and answers, as well as modifying incorrect predictions based on retrieved documents, ultimately leading to the final answer.

However, the original unstructured text contains a large amount of noisy content and the key information is scattered, which requires further optimization in complex scenarios.

Leveraging Structured Knowledge for MHQA. Considering the challenges in unstructured documents, some studies utilized structured information extraction from original documents as a solution (Li and Du, 2023; Panda et al., 2024).

By utilizing structured triples, the model can more effectively capture the relationships among multiple entities, thereby facilitating more reliable reasoning (Liu et al., 2024b). However, the process of constructing these triples requires substantial computational resources and inevitably introduces ambiguities and information loss. Furthermore, when both documents and triples are used as input, such as StructQA (Li and Du, 2023) and ERA-CoT (Liu et al., 2024b), it can lead to information redundancy and cause LLMs to shift their focus.

Unlike existing approaches, our DCSK framework can dynamically integrate the text-based with the triple-based paradigm to optimize resource utilization and enhance performance, and address the aforementioned problems by iteratively selecting and verifying triples related to the question.

3 Methodology

As shown in Figure 2, we propose DCSK, a novel framework for integrating text-based and triple-based paradigms. Under the text-based paradigm, DCSK utilizes the basic reasoning ability of LLMs and employs multi-faceted evaluation to dynamically determine the necessity of structured knowledge. For triple-based paradigm, DCSK excludes irrelevant and redundant information by the designed iterative fact refiner. After, through the adaptive knowledge reasoner, DCSK dynamically adjusts the need for text supplementation to mitigate the information deficiency in triples.

3.1 Task Definition

In this paper, we tackle the MHQA task in a zero-shot setting without training, which makes the problem more challenging and practical (Zhao et al., 2023). Given a multi-hop question q and a series of candidate documents D_q, this task demands capturing the inter-document association information to provide an answer a. The candidate documents $D_q = \{D_q^1, ..., D_q^n\}$, where each document D_q^i encompasses a title t_q^i and a context c_q^i.

3.2 Proposed Method (DCSK)

Vanilla Reasoner. Given that CoT approach can stimulate the incremental thinking ability of LLMs, it not only leads to performance improvements but also enables the visualization of the reasoning logic, thereby facilitating a more accurate evaluation of the correctness of their responses. Additionally, the resource consumption by CoT is only marginally different from that of standard prompt. Therefore, we adopt the CoT as our default setup. Given a multi-hop question q_i, support document set D_{q_i}, we prompt the LLM M_θ to generate an answer a_{q_i} and reasoning process r_{q_i} via:

$$a_{q_i}, r_{q_i} = M_\theta(I_A, q_i, D_{q_i}), \quad (1)$$

where I_A denotes the task instruction for reasoning, the details can be found in Appendix A.2.1.

Multi-Faceted Evaluation. To identify uncertain responses within the text-based paradigm, i.e., questions that genuinely require structured knowledge for accurate answers, we further instruct LLMs to systematically assess their predictions in terms of correctness, clarity, and informativeness, building upon the prior step. For correctness, a score ranging from 0 to 10 is assigned. Since the answers are not specific options, we set scoring indicators reflecting varying degrees of correctness for LLMs as a reference. When LLMs determine that an answer is completely correct with no errors or omissions, they will award a score of 10; if they consider that the prediction is almost entirely correct but may differ slightly in phrasing, they will accordingly reduce the score by 1 to 2 points. Similar adjustments apply in other cases as well.

Since correctness is the primary criterion, if the probability of a response being correct is exceedingly low, it is unnecessary to assess its clarity or informativeness. Consequently, subsequent evaluations are only performed on answers that surpass the predefined correctness threshold $\mathcal{T}$. In terms of clarity, the desired outcome is a concise and unambiguous response devoid of superfluous elaboration. For example, when addressing the question "*Who is the father-in-law of Queen Hyojeong?*", model might responds with "*Crown Prince Hyomyeong (posthumously named Ikjong)*". However, for clarity purposes, the simplified response "*Crown Prince Hyomyeong*" suffices. Additionally, we expect the predictions to provide specific and concrete information rather than evading the question or offering vague and generalized descriptions. For example, responses such as "*Unknown*" or "*It is not mentioned*" are considered uninformative. The detailed prompts are in Appendix A.2.2.

Through the above systematic assessment, we can select the questions that truly require structured knowledge to be re-answered. For each question and its correctness score s_a, clarity score s_c, informativeness score s_i, we define the process as:

$$Q_{sk} = \begin{cases} \bigcup\{q_i\}, & \text{if } s_a < \mathcal{T} \\ \bigcup_{s_c=0, s_i=0}\{q_i\}, & else \end{cases}, \quad (2)$$

where Q_{all} is the entire question set and $q_i \in Q_{all}$.

Structured Knowledge Extractor. In this step, we perform information extraction on the selected questions Q_{sk} and corresponding documents D_{sk}, in order to derive structured knowledge.

Firstly, we conduct entity extraction. Given a document D, we prompt the LLMs to extract entities set E. However, due to limitations in the zero-shot information extraction capabilities of LLMs (Xu et al., 2024a; Chen et al., 2024), we follow StructQA by incorporating examples into task instruction to enhance the extraction performance.

Next, we need to extract the relationships to construct triples. Traditional relation extraction typically employs brief phrases of one or two words to describe these relationships. For instance, from the sentence *"Vilnius County is the largest of the 10 counties of Lithuania"*, it might identify *"(Vilnius County, part of, Lithuania)"*. However, what we require is a more contextually informative one like *"(Vilnius County; largest of 10 counties of; Lithuania)"*. Consequently, we instruct LLMs to describe the relationship in a meaningful and contextually appropriate manner while avoiding generic predicates. This instruction aims to maximize informational content. Similar to entity extraction, we also include examples within our task instructions. For a document D, the triples set could formulated as:

$$T = \bigcup_{e_j \in E, e_k \in E} \{(e_j; r_{e_j e_k}; e_k)\}, \quad (3)$$

where r denotes the relation between entities.

Finally, to effectively harness the implicit knowledge within the question and guide the subsequent iterative processes, we direct LLMs to extract the key elements, encompassing the significant objects as well as the reasoning steps necessary for resolving multi-hop questions. The full instructions of this step can be found in Appendix A.2.3.

Iterative Fact Refinement. Triples are a further refinement of the text, extracting the main information from it. When both triples and documents are used simultaneously, it not only significantly increases the input length but also repeats a large amount of redundant information. Consequently, the focus of LLMs may shift, making it difficult for them to accurately identify key reasoning information in the vast amount of content. Conversely, when directly using triples, due to the inevitable loss of some implicit background information, the model cannot fully understand the context, which in turn leads to incorrect predictions.

Therefore, we design an iterative process to identify triples relevant to the question and verify whether they contain sufficient information to answer the question. Specifically, for each question, guided by the key elements extracted from previous steps, we instruct LLMs to utilize their semantic analysis capabilities to select sub-triples that are directly or implicitly related to the corresponding question. To leverage the lessons learned from prior attempts and prevent redundant operations, we incorporate a memory area into the instruction. This allows LLMs to optimize their current decisions based on prior selections and verification results throughout the iterative process. Formally, for question q_i and its key elements $\mathcal{K}_{q_i}$, the selection process on iteration j can be formulated as:

$$T_{q_i}^{sub} = \mathcal{M}_\theta(I_s, q_i, \mathcal{K}_{q_i}, T_{q_i}, M_{q_i}^{j-1}), \quad (4)$$

where I_s represents the base instruction for this process, while $M_{q_i}^{j-1}$ denotes the memory from the previous iteration (which is empty in first iteration).

Then, we instruct the LLMs to evaluate the selected triples $T_{q_i}^{sub}$ and determine whether they contain sufficient information to address the question q_i. Similarly, the instruction for the verification process also integrated key elements of the question along with memory from previous iterations, ensuring logical consistency in verification while referencing earlier outcomes to enrich current analysis. The verification process of iteration j is as:

$$V_{q_i} = \mathcal{M}_\theta(I_v, q_i, \mathcal{K}_{q_i}, T_{q_i}^{sub}, M_{q_i}^{j-1}). \quad (5)$$

In the scenario of information loss, it becomes evident that no matter how many iterations are performed, it is impossible to fully identify a sufficient number of triples to support a correct reasoning path. Therefore, we set a maximum number of iteration limit γ. When V_{q_i} is *True* or the number of iteration $j > \gamma$, the iterative process will conclude. The detailed instructions are in Appendix A.2.4.

Adaptive Knowledge Reasoner. In this step, based on the previous verification results, we adaptively determine whether to incorporate original text as supplementation, thereby mitigating the information deficiency in triples. When the verification result is *True*, we deem that the selected triples possesses sufficient information to answer. Conversely, when the result is *False* or when the maximum number of iterations is exceed, we interpret this as an indication that there is missing information within the current triples, rendering it insufficient for independently answering the question; thus, assistance from original text is required.

Formally, for each question $q_i \in Q_{sk}$, along with its corresponding verification result V_{q_i}, we can further partition the question set:

$$\begin{aligned} Q_t &= \bigcup_{q_i \in Q_{sk}, V_{q_i} = True} \{q_i\}, \\ Q_{td} &= Q_{sk} - Q_t, \end{aligned} \quad (6)$$

where Q_t and Q_{td} denote respectively the set of questions that can be answered merely by selected triples and the set of questions that demand additional documents. Then, for each question $q_i \in Q_t$ and $q_j \in Q_{td}$, we obtain the corresponding answers a_{q_i} and a_{q_j} in a manner similar to Equation (1), and thereby form the answer sets A_t and A_{td}. The final complete answer set is as follows:

$$A_{all} = A_{cot} \cup A_t \cup A_{td}. \quad (7)$$

Methods	HotpotQA				2WikiMultihopQA				Musique			
	EM	F1	P	R	EM	F1	P	R	EM	F1	P	R
						GPT-4o-mini						
Base	0.538	0.692	0.690	0.805	0.484	0.631	0.597	0.787	0.285	0.456	0.425	0.705
CoT	0.623	0.778	0.795	0.807	0.682	0.801	0.779	0.860	0.463	0.618	0.619	0.668
CoT-SC@5	0.639	0.783	0.798	0.805	0.687	0.813	0.790	0.874	0.481	0.641	0.645	0.689
RE2	0.634	0.781	0.801	0.804	0.684	0.800	0.778	0.859	0.480	0.628	0.631	0.677
StructQA	0.646	0.780	0.797	0.799	0.694	0.821	0.807	0.862	0.511	0.648	0.662	0.674
ERA-CoT	0.666	0.801	0.827	0.807	0.692	0.815	0.799	0.866	0.482	0.636	0.642	0.673
DCSK $_{CoT}$	0.682	0.819	0.841	**0.830**	0.741	0.828	0.813	0.871	0.515	0.670	0.679	0.702
DCSK $_{SC}$	**0.689**	**0.822**	**0.844**	0.828	0.744	**0.832**	**0.818**	**0.874**	0.527	**0.683**	**0.690**	**0.708**
DCSK $_{RE2}$	0.686	0.817	0.843	0.822	**0.747**	0.829	0.813	0.873	**0.529**	0.678	0.689	0.705
						Qwen2.5-7B						
Base	0.568	0.704	0.737	0.723	0.549	0.630	0.632	0.657	0.340	0.465	0.495	0.508
CoT	0.588	0.724	0.757	0.736	0.594	0.667	0.663	0.703	0.330	0.444	0.467	0.471
CoT-SC@5	0.599	0.734	0.769	0.738	0.622	0.701	0.695	0.737	0.345	0.466	0.494	0.481
RE2	0.585	0.723	0.758	0.732	0.596	0.668	0.660	0.707	0.318	0.439	0.465	0.469
StructQA	0.604	0.730	0.756	0.739	0.646	0.717	0.708	0.748	0.350	0.476	0.495	0.491
ERA-CoT	0.577	0.710	0.721	0.735	0.606	0.700	0.690	0.738	0.334	0.467	0.502	0.478
DCSK $_{CoT}$	0.642	0.775	0.804	0.780	0.696	0.784	0.776	0.823	**0.457**	**0.585**	**0.600**	**0.614**
DCSK $_{SC}$	**0.649**	**0.779**	**0.809**	**0.785**	**0.705**	**0.794**	**0.780**	**0.833**	0.452	0.576	0.591	0.603
DCSK $_{RE2}$	0.635	0.770	0.798	0.777	0.697	0.784	0.770	0.825	0.445	0.572	0.585	0.602

Table 1: Multi-Hop Question Answering performance comparison of different methods. The best results are in **bold**.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on three common-used multi-hop question answering datasets: HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), and Musique (Trivedi et al., 2022). Each question contains at least ten candidate documents, which include both golden documents and noise documents. Unlike StructQA (Li and Du, 2023), which exclusively utilizes golden documents, we focus on simulating complex scenarios that reflect real-world conditions by incorporating noise documents as input (Trivedi et al., 2022). However, to reduce resource consumption, we limit the number of candidate documents for each question to five and adopt Qwen2.5 to process structured knowledge for all experiments. Moreover, we have doubled the dataset based on StructQA, randomly sampling 1,000 questions from each dataset to serve as the test set.

Evaluation Metrics. Follow previous work (Li and Du, 2023), we adopt the Exact-Match (EM), F1-score, Precision and Recall as the evaluation metrics. And we use the same post-processing method to calculate metrics to ensure fairness.

Baselines. To comprehensively evaluate our approach, we compare DCSK to the following baselines: Base (standard prompt), Chain-of-Thought (CoT) (Wei et al., 2022), CoT Self-Consistency @5 (CoT-SC@5) (Wang et al., 2023), RE2 (Xu et al., 2024b), StructQA (Li and Du, 2023), and ERA-CoT (Liu et al., 2024b). The detailed descriptions of these baselines are in Appendix A.1.1.

Implementation Details. We choose two LLMs as the backbone, namely GPT-4o mini [2] (Hurst et al., 2024), and Qwen2.5-7B-Instruct [3] (Yang et al., 2024a). For LLMs, we set the temperature to 1.0, and the max-token for generation is 1024. For other hyper-parameters of our method, we set threshold $\mathcal{T}$ to 8, maximum iteration γ to 3. To ensure fairness, our method and the triple-based baselines are compared using the same triples throughout the experiments. In addition, to show the compatibility of our framework, we select three methods to enhance the text-based paradigm, denoted DCSK $_{CoT}$, DCSK $_{SC}$ and DCSK $_{RE2}$, respectively.

4.2 Main Results

Table 1 shows the comprehensive performance comparison of our proposed DCSK and baselines.

[2] gpt-4o-mini-2024-07-18

[3] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

Specifically, we can observe that: (1) The triple-based baselines generally exhibit better performance compared to other methods in most cases, indicating that the incorporation of structured knowledge can enhance multi-hop reasoning. (2) DCSK significantly outperforms all baselines across three datasets and two LLMs, thereby demonstrating the effectiveness of our approach. Furthermore, DCSK consistently exhibits performance enhancements when combined with the three text paradigms respectively, thereby not only confirming its superior transferability (i.e., its ability to be seamlessly incorporated into other text-based methods). (3) The performance of ERA-CoT in different LLMs exhibit considerable inconsistency. When employing Qwen2.5, its performance was not notably different from the CoT. But when utilizing the more powerful GPT-4o-mini, the performance increase was conspicuous. We contend that this is because ERA-CoT, apart from exploiting documents and triples, also deduces a number of implicit relations for each document as additional inputs. When the quantity of documents grows, this renders the input more noisy and redundant, causing less robust models to be subjected to more severe interference.

Overall, DCSK demonstrates consistent improvements and robust resilience in more challenging noisy scenarios. Moreover, its synergistic performance when integrated with various text-based methods highlights its exceptional transferability.

4.3 Ablation Study

Methods	HotpotQA		2Wiki.		Musique	
	EM	F1	EM	F1	EM	F1
DCSK $_{CoT}$	0.642	0.775	0.696	0.784	0.457	0.585
w/o DC	0.626	0.751	0.689	0.779	0.435	0.562
w/o Verification	0.609	0.728	0.676	0.766	0.393	0.527
w/o Selection	0.595	0.723	0.639	0.721	0.368	0.472

Table 2: The performance comparison of ablation study. 2Wiki. denotes the 2WikiMultihopQA dataset.

To assess the effectiveness of each module, we conduct ablation experiments under Qwen2.5 and introduce the following variations of DCSK $_{CoT}$: 1) w/o Dynamic Combination (DC) means removing the multi-faceted evaluation, and answering all questions by the triple-based paradigm. 2) w/o Verification refers to the elimination of the verification in iterative fact refiner, indicating the question is answered directly after the selection. 3) w/o Selection indicates the removal of the selection stage.

As shown in Table 2, each module we designed plays a crucial role. When the DC is removed, all questions must rely on structured knowledge for answering, which not only consumes substantial resources (refer to Section 4.4.1 for specific statistics) but also diminishes overall performance. This is because DC not only identifies questions solvable by both paradigms but also incorporates issues that are exclusively resolvable by the text-based paradigm and remain intractable for the triple-based paradigm. Additionally, the removal of the selection and verification processes leads to a significant decline in performance, indicating that these two modules are effective in mitigating interference from noise and redundant information within structured knowledge.

4.4 Comparative Analysis

In this section, we conduct a further analysis with Qwen2.5 from the following perspectives:

- The comprehensive efficiency of DCSK.
- The detailed analysis of decision for dynamic combination.
- The impact of correctness score and threshold in multi-faceted evaluation.
- The influence of the maximum iteration limit.

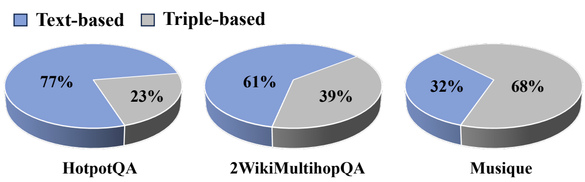

Figure 3: Detailed statistics on the number of questions addressed by different paradigms of our DCSK.

Methods	HotpotQA	2Wiki.	Musique	Avg.	
	#Token ↓	#Token ↓	#Token ↓	EM ↑	F1 ↑
StructQA	10030	8909	**9653**	0.530	0.641
ERA-CoT	25062	22738	24498	0.506	0.626
DCSK $_{CoT}$ w/o DC	13721	12644	14769	0.583	0.697
DCSK $_{CoT}$	**5599**	**6476**	11268	**0.598**	**0.715**

Table 3: The efficiency comparison based on the triple-based methods. #Token represents the average number of tokens used for each question. "Avg." is the abbreviation for "average".

4.4.1 Efficiency Analysis

In order to comprehensively demonstrate the efficiency of our dynamic combination, we present the specific statistics of answering questions using the two paradigms in Figure 3. It is evident that our method introduce structured knowledge for only 23% and 39% of the questions on the HotpotQA and 2WikiMultihopQA datasets, respectively. This significantly reduces computational resources. In contrast, for the more challenging Musique, a secondary response was necessary for 68% of the questions. This adaptive adjustment to varying levels of difficulty not only conserves resources but also ensures performance reliability.

Furthermore, we present a comparison of DCSK and its variant with other triple-based methods in terms of token consumption in Table 3. From this, it can be observed that our DCSK maintains a leading overall performance while significantly reducing resource consumption on the HotpotQA and 2WikiMultihopQA datasets. In contrast, the ERA-CoT method incurs a notable increase in resource consumption due to the necessity of inferring several implicit relationships for each document and scoring them sequentially. This process not only leads to significant increases in resource usage but also results in considerable redundancy in input data, thereby adversely affecting performance. Moreover, when DCSK eliminates dynamic combination, there is a substantial increase in resource consumption accompanied by a slight decline in performance. This finding underscores the rationale behind our approach: **not all questions necessitate intervention from structured knowledge**.

Dataset	Same	Better	Worse
HotpotQA	51%	35%	11%
2WikimultihopQA	54%	40%	6%
Musique	48%	39%	13%

Table 4: The decision analysis under structured knowledge versus text-based reasoning in subset of questions where the triple-based paradigm is selected after dynamic combination.

4.4.2 Detailed Analysis of Decision

To illustrate the effectiveness of our dynamic combination, we further analyzed the subset of questions routed to the triple-based paradigm with Qwen, comparing performance under structured knowledge versus text-based reasoning. The outcomes were categorized as follows: We categorized the outcomes as follows: 1) ***Same***: Both paradigms yield the same performance. 2) ***Better***: Structured knowledge produces better result. 3) ***Worse***: Text-based reasoning produces better results, indicating an error in routing by the dynamic combination.

As shown in Table 4, the proportion of erroneous decisions (*Worse*) remains relatively low across all datasets, while the performance gains from correct decisions (*Better*) substantially outweigh the negative impact of occasional misjudgments. These results confirm the effectiveness and robustness of decisions by our dynamic combination strategy.

4.4.3 Impact of Correctness

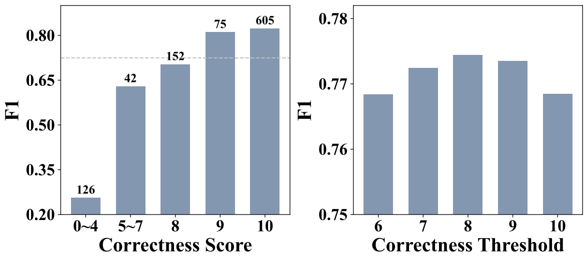

Figure 4: Left sub-figure: The distribution of F1 value corresponding to the correctness scores in multi-faceted evaluation, where the numbers atop the bars represent the count of questions for each score interval. The dashed line denotes performance across the entire dataset for comparative reference. Right sub-figure: Influence of varying thresholds.

Since correctness is the primary criterion, we further investigate the impact of correctness score and threshold in multi-faceted evaluation on the HotpotQA, as illustrated in Figure 4. We can observe that predictions with higher scores tend to exhibit better F1 value, and the top two scoring segments account for 68% of the total. This observation aligns perfectly with the premise of our approach, which posits that LLMs can address many questions effectively with unstructured text. For such questions, further processing is unnecessary.

Furthermore, by integrating the two sub-figures, we can observe a certain degree of overthinking phenomenon. In the left sub-figure, the F1 with a score of 8 did not reach the average F1 of the total. Logically, when this part of the questions are combined with more effective triples-based predictions, it should have led to a better performance. However, from the comparison of thresholds 8 and 9 in the right sub-figure, we can see that the actual situation is contrary to the ideal one. We posit that this discrepancy arises because, in this specific subset, the overall superior structured knowledge exhibits inferior performance compared to plain text. This also indicates why our dynamic combination strategy continues to enhance performance.

Metrics	γ@1	γ@2	γ@3	γ@4	γ@5
EM ↑	0.436	0.448	**0.457**	0.454	0.449
F1 ↑	0.572	0.582	0.585	0.587	**0.588**
#Token ↓	**1759**	2376	2793	3225	3697

Table 5: The influence of maximum number γ of iteration. γ@n denotes the specified value. #Token is the average token consumption for each question in the iterative process.

4.4.4 Influence of Maximum Iteration Limit

Table 5 illustrates the variation of metrics with respect to different maximum number of iterations, denoted as γ, on the Musique. We can observe that although the resource consumption in the first iteration is minimal, its performance is notably suboptimal. In contrast, from the second iteration onward, there is a discernible upward trend in model performance, indicating its capacity to optimize current decisions based on prior processes. However, despite some metrics continuing to improve with an increased number of iterations, it is important to consider both the marginal gains and rising costs. Therefore, we have set a maximum iteration limit of 3 to strike a balance between resource consumption and performance.

5 Conclusion

In this paper, we proposed Dynamic Combination of Structured Knowledge (DCSK), a novel framework for integrating text-based and triple-based paradigms to address the multi-hop question answering. DCSK dynamically determined whether the question requires the intervention of structured knowledge through the multi-faceted evaluation, which assess the correctness, clarity, and informativeness of text-based prediction. For questions that require intervention, we designed an iterative fact refiner to select and verify the triples related to the question, thereby excluding redundant information. Additionally, based on the verification, we constructed an adaptive knowledge reasoner to adjust the need for text supplementation to address the information deficiency in triples. Extensive experiments demonstrated the effectiveness of DCSK.

Limitations

In certain scenarios, our work has some limitations. First, our approach is a further improvement on the previous triple-based paradigm work, which requires leveraging the inherent capabilities of LLMs to extract structured knowledge. However, when the model parameters are insufficiently large (for instance, 220M or 1B), may encounter limitations due to the bottleneck in information extraction capability. In future work, we plan to investigate a dedicated extraction model to enhance adaptability across models with varying parameter sizes. Second, in multi-faceted evaluation, considering that simultaneously outputting answers and scores would affect the model's performance (Zhao et al., 2023), our method requires additional model calls, leading to unavoidable expenses. Moving forward, we aim to explore aspects of internal interpretability within LLMs to mitigate such costs.

Acknowledgments

This work was supported in part by the grants from National Natural Science Foundation of China (No.62222213, U22B2059), in part by the Postdoctoral Fellowship Program and China Postdoctoral Science Foundation under Grant Number BX20250387. And this work was also supported by USTC-NIO Smart Electric Vehicle Joint Lab.

References

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *8th International Conference on Learning Representations*.

Wei Chen, Lili Zhao, Zhi Zheng, Tong Xu, Yang Wang, and Enhong Chen. 2024. Double-checker: large language model as a checker for few-shot named entity recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3172–3181.

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.

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*, pages 169–180.

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.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023. Few-shot reranking for multi-hop qa via language model prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15882–15897.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. A survey on complex knowledge base question answering: Methods, challenges and solutions. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 4483–4491.

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.

Qi Liu, Yongyi He, Tong Xu, Defu Lian, Che Liu, Zhi Zheng, and Enhong Chen. 2024a. Unimel: A unified framework for multimodal entity linking with large language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 1909–1919.

Yanming Liu, Xinyue Peng, Tianyu Du, Jianwei Yin, Weihao Liu, and Xuhong Zhang. 2024b. Era-cot: Improving chain-of-thought through entity relationship analysis. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8780–8794.

Pranoy Panda, Ankush Agarwal, Chaitanya Devaguptapu, Manohar Kaul, and Prathosh A P. 2024. HOLMES: hyper-relational knowledge graphs for multi-hop question answering using llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13263–13282.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8864–8880.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 6140–6150.

Zhengliang Shi, Shuo Zhang, Weiwei Sun, Shen Gao, Pengjie Ren, Zhumin Chen, and Zhaochun Ren. 2024. Generate-then-ground in retrieval-augmented generation for multi-hop question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7339–7353.

Haitian Sun, William W Cohen, and Ruslan Salakhutdinov. 2021. Iterative hierarchical attention for answering complex questions over long documents. *arXiv preprint arXiv:2106.00200*.

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.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Ziqi Wang, Chen Zhu, Zhi Zheng, Xinhang Li, Tong Xu, Yongyi He, Qi Liu, Ying Yu, and Enhong Chen. 2024. Granular entity mapper: Advancing fine-grained multimodal named entity recognition and grounding. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3211–3226.

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.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024a. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6):186357.

Derong Xu, Xinhang Li, Ziheng Zhang, Zhenxi Lin, Zhihong Zhu, Zhi Zheng, Xian Wu, Xiangyu Zhao, Tong Xu, and Enhong Chen. 2025. Harnessing large language models for knowledge graph question answering via adaptive multi-aspect retrieval-augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25570–25578.

Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, Jian-Guang Lou, and Shuai Ma. 2024b. Re-reading improves reasoning in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15549–15575.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024a. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024b. Do large language models latently perform multi-hop reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10210–10229.

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.

Yu Yuan, Lili Zhao, Wei Chen, Guangting Zheng, Kai Zhang, Mengdi Zhang, and Qi Liu. 2025. Simulating human-like learning dynamics with llm-empowered agents. *arXiv preprint arXiv:2508.05622*.

Yu Yuan, Lili Zhao, Kai Zhang, Guangting Zheng, and Qi Liu. 2024. Do llms overcome shortcut learning? an evaluation of shortcut challenges in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12188–12200.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Liu Yong, and Shen Huang. 2024. End-to-end beam retrieval for multi-hop question answering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1718–1731.

Lili Zhao, Qi Liu, Wei Chen, Liyi Chen, Ruijun Sun, Min Hou, Yang Wang, and Shijin Wang. 2025a. Mimu: Mitigating multiple shortcut learning behavior of transformers. *arXiv preprint arXiv:2504.10551*.

Lili Zhao, Qi Liu, Linan Yue, Wei Chen, Liyi Chen, Ruijun Sun, and Chao Song. 2024. COMI: correct and mitigate shortcut learning behavior in deep neural networks. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024*, pages 218–228.

Lili Zhao, Yang Wang, Qi Liu, Mengyun Wang, Wei Chen, Zhichao Sheng, and Shijin Wang. 2025b. Evaluating large language models through role-guide and self-reflection: A comparative study. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15686–15702.

A Appendix

A.1 Supplementary Experimental Setup

A.1.1 Descriptions of Baselines

We compare our DCSK with following two kinds of baselines.
(i) Text-based methods:

- **Base** denotes the standard prompting method.
- **CoT** (Wei et al., 2022) prompts the LLMs to engage in step-by-step reasoning, thereby deriving answers through a systematic process.
- **CoT-SC@5** (Wang et al., 2023) samples the reasoning paths of five responses based on CoT and selects the most consistent one as the final answer.
- **RE2** (Xu et al., 2024b) is a plug-and-play method that enhances the reasoning capabilities of LLMs by repeating questions to obtain bidirectional understanding.

(ii) Triple-based methods:

- **StructQA** (Li and Du, 2023) constructs semantic graphs by extracting structured knowledge to facilitate the semantic understanding of documents, thereby achieving a more faithful reasoning process and improved model performance in MHQA.
- **ERA-CoT** (Liu et al., 2024b), on the basis of extracting structured triples, enables the model to analyze and infer the implicit relations among entities, thereby enhancing the reasoning performance in zero-shot scenarios.

A.2 Prompts

Here we provide the specific prompts DCSK used.

A.2.1 Reasoner

```
Based on the provided information and your own
knowledge, answering the question by thinking
step by step.

Input: {}

Question: {}
```

A.2.2 Multi-Faceted Evaluation

Correctness

```
Evaluate your prediction for the given question
based on the Document in previous step. You are
an expert evaluator for question-answering
systems, and your task is to assess the
correctness of the prediction. Your evaluation
should consider the following scoring indicators:

Scoring Indicators:
1. 10 (Fully Correct):
   - The prediction is fully correct, matching
   the document's information exactly or through
    valid reasoning.
   - It captures all essential details with no
   errors or omissions.

2. from 8 to 9 (Mostly Correct):
   - The prediction is almost entirely correct
   but may differ slightly in phrasing or omit
   minor details.
   - Example: Synonyms, alternate phrasing, or
   minor extra words.

3. from 5 to 7 (Partially Correct):
   - The prediction contains correct information
    but misses key details or introduces notable
    errors.
   - Example: Captures part of the answer but
   not all essential elements.

4. from 1 to 4 (Mostly Incorrect):
   - The prediction is loosely related to the
   document but is largely incorrect or contains
    major errors.

5. 0 (No Answer or Completely Incorrect):
   - The prediction is either completely
   unrelated to the document or fails to provide
    any meaningful answer.
   - This includes cases where the model
   predicts that the answer does not exist in
   the document, which is incorrect because the
   answer is guaranteed to be present.

---

Instructions:
1. Carefully compare the prediction with the
document, considering both explicit and implicit
 information.
2. Pay special attention to cases where the
model provides no answer or incorrectly claims
the answer is not in the document.
3. Use the scoring indicators to assign a score
from 0 to 10.
```

Clarity

```
You are an expert in evaluating prediction
clarity for question-answering systems. Your
task is to assess whether the prediction is
clear, concise, and free from unnecessary
explanation or background information. A clear
prediction should be direct and without extra
commentary.

---

Scoring Criteria:
- 1 (Clear): The prediction is concise and
direct, with no unnecessary information.
```

```
- 0 (Not Clear): The prediction includes
redundant background, is overly wordy, or could
be expressed more directly.

---

Exampls:

Question: What city is home to the university
attended by the founder of SpaceX?
Prediction: Elon Musk, the founder of SpaceX,
attended the University of Pennsylvania, which
is located in Philadelphia, a major city in the
state of Pennsylvania.
Score: 0

Question: What city is home to the university
attended by the founder of SpaceX?
Prediction: Philadelphia
Score: 1

Question: What is the nationality of the foreign
 born victim of Singapore's caning punishment
before Oliver Fricker experienced the same?
Prediction: The foreign-born victim who was
caned in Singapore before Oliver Fricker was
American, as in the case of Michael Fay in 1994.
Score: 0

Question: What is the nationality of the foreign
 born victim of Singapore's caning punishment
before Oliver Fricker experienced the same?
Prediction: American
Score: 1

---

Question: {}
Prediction: {}
```

Informativeness

```
You are an expert in evaluating prediction
informativeness for question-answering systems.
Your task is to assess whether the prediction
provides concrete information. Prediction that
avoid answering (explicitly or implicitly) or
include vague, generic statements do not count
as informative.

---

Scoring Criteria:
- '1' (Informative):
  The prediction includes a substantive,
  specific answer.
- '0' (Uninformative):
  The prediction does not provide useful content.
   It avoids answering, provides only vague or
  general remarks, or explicitly indicates the
  information is unavailable using phrases such
  as:
  - None
  - null
  - Not specified in the given documents
  - Not mentioned
  - Not provided
  - Unknown
  - Not explicitly stated

---

Examples:
Question: What is the name of the actor who
played Marty Castillo in Miami Vice?
Prediction: Not specified in the given documents
Score: 0

Question: What is the name of the actor who
played Marty Castillo in Miami Vice?
Prediction: Edward James Olmos
Score: 1

---

Question: {}
Prediction: {}
```

A.2.3 Structured Knowledge Extraction

Entity Extraction

```
Given the sentence, please refer to the possible
 entity types and extract all the named entities
 from the sentence.
Possible types: [person name, organization,
location, medical code, time expression (e.g.,
dates, times), quantities (e.g., numbers,
amounts), monetary value, percentage, other
relevant entity (if applicable)]

Use the following examples to understand the
task:

Examples:
title: Ferragosto in bikini
sentence: Ferragosto in bikini is a 1960 Italian
 comedy film directed by Marino Girolami. The
film is named after a hit song of musical group
Quartetto Cetra, who also makes a brief cameo
appearance.
entities: {Ferragosto in bikini; 1960; Italian;
Marino Girolami; Quartetto Cetra}

title: Heo Keon
sentence: Heo Keon( born 3 January 1988) is a
South Korean footballer who plays as midfielder
for Bucheon FC 1995 in K League Challenge.
entities: {Heo Keon; 3 January 1988; South
Korean; Bucheon FC 1995; K League Challenge}

---

Now you need to output the entities based on the
 following sentence in same format as examples.
```

Relation Extraction

```
Given a sentence and a set of entities, extract
the relationships between the entities and
represent them as triples (subject; predicate;
object).
Instructions for the task:
1. Only consider the entity provided in the
Entities set.
2. Identify all relevant relationships between
the entities.
3. Describe each relationship using a
contextually relevant predicate that accurately
and comprehensively reflects the content of the
```

```
sentence. Use meaningful verbs or descriptive
phrases to define the connection (e.g., "plays
as a midfielder for, "is known for, "was born on
").
4. Format your output as a list of triples,
where each triple is formatted as: '(subject;
predicate; object)'. Multiple triples should be
separated by a new line.

---

Use the following examples to understand the
task:
Examples:
Title: barry mcguire
Sentence: Barry McGuire (born October 15, 1935)
is an American singer-songwriter. He is known
for the hit song 'Eve of Destruction', and later
 as a pioneering singer and songwriter of
contemporary Christian music.
Entities set: {Barry McGuire; October 15, 1935;
American; Eve of Destruction; contemporary
Christian music}
Extracted triples:
(Barry McGuire; was born on; October 15, 1935)
(Barry McGuire; is a singer-songwriter of;
American)
(Barry McGuire; is known for; Eve of Destruction
)
(Barry McGuire; is a pioneering singer and
songwriter of; contemporary Christian music)

Title: Ferragosto in bikini
Sentence: Ferragosto in bikini is a 1960 Italian
 comedy film directed by Marino Girolami. The
film is named after a hit song of musical group
Quartetto Cetra, who also makes a brief cameo
appearance.
Entities set: {Ferragosto in bikini; 1960;
Italian; Marino Girolami; Quartetto Cetra}
Extracted triples:
(Ferragosto in bikini; is a comedy film of;
1960)
(Ferragosto in bikini; is a film of; Italian)
(Ferragosto in bikini; was directed by; Marino
Girolami)
(Ferragosto in bikini; is named after a hit song
 of; Quartetto Cetra)
(Quartetto Cetra; makes a cameo appearance in;
Ferragosto in bikini)

---

Now, apply the above method to extract the
relationships for the given sentence and
entities.
```

Query Breakdown

```
You are an expert in question decomposition and
semantic analysis. Your task is to analyze a
given question and extract key contextual
elements, focusing on both the objects mentioned
 in the question and the detailed reasoning
steps needed to answer the question.

Examples:
Question 1:
Who was born first, Mary Lou Marzian or Charles
Gonthier, Prince Of Schwarzburg-Sondershausen?
Output:
{
  "Objects": [
    "Mary Lou Marzian",
    "Charles Gonthier, Prince Of Schwarzburg-
    Sondershausen",
    "birth dates of both individuals"
  ],
  "Contextual Links": [
    "Step 1: Identify Mary Lou Marzian's birth
    date.",
    "Step 2: Identify Charles Gonthier's birth
    date.",
    "Step 3: Compare the two birth dates to
    determine who was born earlier."
  ]
}

Question 2:
Where was the place of death of Alessandro
Ruspoli, 2Nd Prince Of Cerveteri's father?
Output:
{
  "Objects": [
    "Alessandro Ruspoli, 2Nd Prince Of Cerveteri
    ",
    "father of Alessandro Ruspoli",
    "place of death of Alessandro Ruspoli's
    father"
  ],
  "Contextual Links": [
    "Step 1: Identify who is the father of
    Alessandro Ruspoli, 2Nd Prince Of Cerveteri
    .",
    "Step 2: Locate the place where Alessandro
    Ruspoli's father died."
  ]
}

---

Now, apply this method to the following question:


Question: {}
```

A.2.4 Iterative Fact Refiner

Selection

```
You are an expert in identifying relevant
information from structured data. Your task is
to analyze a set of fact tuples and select only
those that are directly or indirectly relevant
to answering the given question. You will also
be provided with a query breakdown to guide your
 reasoning process.

If available, you can also access a memory of
previously selected and verified tuples to
support your decision-making.

Task Instructions:
1. Leverage Query Breakdown:
   - Follow the breakdown steps to determine
   which tuples are necessary for each reasoning
    step.
   - Select tuples based on both explicit
   relevance and simple implicit associations:
```

```
    - Explicit relevance: Tuples that directly
    match the query breakdown steps or provide
    clear information needed to answer the
    question.
    - Simple implicit associations: Tuples that
    do not directly match the breakdown steps
    but are contextually related, such as cause-
    effect, attribute extension, or related
    entities.

2. Leverage Memory (if available):
  - Memory contains tuples and reasoning from
  previous iterations. Use it to:
    - Identify tuples that may have been missed
    in the previous selection process and add
    them to the current selection if they are
    relevant.
    - Refine your current selection based on
    additional context or previously verified
    tuples.
    - Ensure that new selections are not
    redundant unless they provide critical
    information that was missing before.

3. Focus on Relevance:
  - Carefully analyze the fact tuples provided.
  - Select only the tuples that are directly or
   indirectly relevant to the question and its
  breakdown steps.
  - Ignore irrelevant or noisy tuples.

4. Output Format:
  - Provide your output with the following
  fields:
    - ‘reasoning‘
    - ‘selected_tuples‘
```

Verification

```
You are an expert in reasoning and verifying the
 sufficiency of selected information for
answering complex questions. Your task is to
analyze a set of selected tuples and determine
if they provide enough information to answer the
 given question, either explicitly or through
reasoning.

You will also be provided with a query breakdown
 to guide your verification process and, if
available, a memory of previously verified
results to support your decision-making.

Task Instructions:
1. Verify Sufficiency of Selected Tuples:
  - Carefully evaluate the selected tuples and
  determine whether they contain enough
  information to answer the given question.
  - Consider if the answer is explicitly stated
   or can be inferred through reasoning based
  on the selected tuples.

2. Leverage Query Breakdown:
  - Use the breakdown to assess the reasoning
  steps required to answer the question.
  - Identify whether the selected tuples align
  with these steps and provide the necessary
  information for each step.
  - Attempt to infer implicit relationships
  between tuples. These may include:
    - Causal links (e.g., Tuple A leads to Tuple
     B).
    - Attribute extensions (e.g., Tuple A
    mentions an object, Tuple B describes its
    property).
    - Temporal or spatial relationships.

3. Leverage Memory (if available):
  - Memory contains tuples and reasoning from
  previous iterations. Use it to:
    - Cross-check the current verification
    against prior reasoning and results.
    - Reassess tuples marked as insufficient in
    prior iterations to see if they now provide
    enough information when combined with the
    current selection.
    - Ensure consistency in verification logic
    and avoid contradicting previous results
    without valid justification.
    - Integrate previous implicit reasoning
    steps to enrich the current analysis.

4. Output Format:
  - Provide your output in JSON format with the
   following fields:
    - ‘reasoning‘
    - ‘verification‘: A Boolean value.
```