DetectBench: Can LLMs Piece Together Implicit Evidence for Long-Context Multi-Hop Reasoning?

Anonymous ACL submission

Abstract

Detecting evidence within the context is a key step in the process of reasoning task. Evaluating and enhancing the capabilities of LLMs in evidence detection will strengthen context-based reasoning performance. This paper proposes a benchmark called DetectBench for verifying the ability to detect and piece together implicit evidence within a long context. DetectBench contains 3,928 multiple-choice questions, with an average of 190.6 tokens per question. Each question contains an average of 4.7 pieces of implicit evidence, and solving the problem typically requires 8.9 logical jumps to find the correct answer. To enhance the performance of LLMs in evidence detection, this paper proposes Detective Reasoning Prompt and Finetune. Experiments demonstrate that the existing LLMs' abilities to detect evidence in long contexts are far inferior to humans. However, the Detective Reasoning Prompt effectively enhances the capability of powerful LLMs in evidence detection, while the Finetuning method shows significant effects in enhancing the performance of weaker LLMs. Moreover, when the abilities of LLMs in evidence detection are improved, their final reasoning performance is also enhanced accordingly.

1 Introduction

The ability to perform reasoning over natural language is an important aspect of intelligence (Chen and Xiao, 2022). Tasks designed to assess inferential capabilities commonly consist of a context and a question, expecting the Large Language Models (LLMs) to respond correctly (Chu et al., 2023; Davis, 2023). Human annotators often conceal the evidence necessary for answering the question within the context. This raises a question: ***whether LLMs possess the capability to detect these pieces of evidence and understand how to formulate reasoning based upon them?***

Identifying evidence often poses a more significant challenge than reasoning, as it necessitates

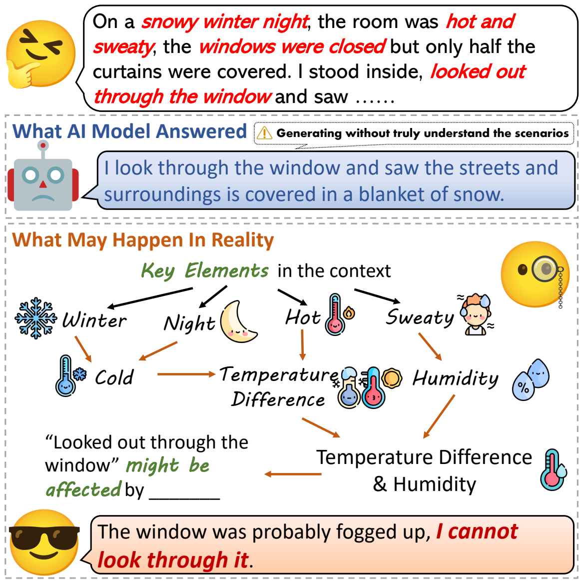

Figure 1: LLMs are hard to aware of the implicit evidence in the context so they may respond arbitrarily.

a deeper understanding of the question and context. There are many existing tasks evaluate the model's joint abilities in evidence detection and evidence-based reasoning in long contexts, such as reading comprehension (Yu et al., 2020; Kazi and Khoja, 2021; Lu et al., 2022b), retrieval reasoning (Yang et al., 2018; Chen et al., 2023), and fact verification (Thorne et al., 2018a,b; Aly et al., 2021). The existing benchmarks of these tasks often present evidence that is too explicit and direct, which is easy to find through rule-based retrieval methods. However, in real scenarios, evidence is usually implicit within the context, and accurately solving a problem often requires the integration of multiple pieces of evidence through joint reasoning. For example, as shown in Fig. 1, only when we realize that changes in temperature and humidity will make glass foggy can we figure out that details about temperature and humidity are crucial to seeing through the glass.

To evaluate whether models can detect and piece pieces of evidence together to answer questions, a benchmark consisting of multiple pieces of implicit evidence within a long context is needed. So, in this paper, we propose a multiple-choice question answering benchmark called **Detect**ive **Bench**mark (**DetectBench**). DetectBench comes from the idea that "when facing a criminal case, detectives often need to identify the most crucial evidence from a vast array of seemingly unrelated information to solve the case". This benchmark comprises 3,928 questions, each paired with a paragraph averaging 190 tokens and averaging 4.7 annotated implicit evidence to answer a question. The characteristics of DetectBench include: 1. Evidence related to question-answering cannot be detected through the character or string within questions and options. 2. It necessitates the combination of multiple pieces of evidence to derive more critical results for question answering. 3. The context contains a significant amount of misleading and irrelevant information. 4. Each question has a detailed annotation from evidence to reasoning to answer.

In experiments conducted on human participants and LLMs, we assessed their evidence detection and question-answering abilities on DetectBench. Our findings reveal that humans significantly surpassed the most advanced LLMs in both tasks. By analyzing the correlation between accuracy in evidence detection and question answering, we discovered a high degree of positive correlation between them, confirming the effectiveness of the annotations within DetectBench and underscoring the critical role of evidence identification in the reasoning process.

To enhance the model's capabilities in evidence detection and evidence-based reasoning, we proposed Detective Reasoning to improve these two capacities simultaneously. Like how experienced detectives collectively conduct evidence detection and reasoning, Detective Reasoning enhances LLMs by directing them to thoroughly consider all possible evidence, engage in reasoning, and summarize the entire reasoning process to refine the evidence. Finally, reasoning from the evidence is used to ascertain the answer to the question. Constructing prompts with Detective Reasoning further enhances the evidence detection and reasoning capabilities of state-of-the-art (SoTA) LLMs. Similarly, developing a Fine-Tuning (FT) dataset inspired by the principles of Detective Reasoning also advances the abilities of open-source LLMs in this regard.

In summary, the primary contributions of this study are as follows: (1) The introduction of DetectBench, establishing a new benchmark for evaluating models' evidence detection and reasoning capabilities within a long context. (2) We propose Detective Reasoning, which can be employed to enhance LLMs' evidence detection and reasoning skills concurrently. We propose prompt and fine-tuning methods to implement Detective Reasoning. The prompt method augments the capabilities of already powerful LLMs, while the fine-tuning, consisting mainly of a self-supervised data collection strategy, improves the capabilities of open-source LLMs. (3) Numerous experiments based on DetectBench have led to the discovery of a positive correlation between a model's reasoning abilities and its capacity for evidence detection. We have also identified deficiencies in evidence detection among large models. After reinforcement through the Detective Reasoning approach, LLMs can compensate for weaknesses in both domains. However, even with this reinforcement, they still need to catch up to the average human level.

2 Related Works

2.1 Information Retrieval

Evidence detection is one of the two main characteristics of DetectBench, which is a sub-domain of Information Retrieval. Information Retrieval aims to address pertinent tasks by extracting crucial data from many references, where the most significant challenge lies in identifying implicit key information (Zhu et al., 2023; Yang et al., 2022). Traditional benchmarks in Information Retrieval have historically segmented the task of Information Extraction to evaluate models independently (Martinez-Rodriguez et al., 2020; Cheng et al., 2021; Lu et al., 2022a). Recent endeavors, however, have led to the development of benchmarks designed for the holistic assessment of task resolution capabilities. Among these, HotPotQA (Yang et al., 2018) necessitates the discovery of question-relevant information across paragraphs to aid in response formulation, FEVER (Thorne et al., 2018a,b; Aly et al., 2021) necessitates the identification of evidentiary support to validate or negate a claim, and RECLOR (Yu et al., 2020), UQuAD (Kazi and Khoja, 2021), BIOMRC (Lu et al., 2022b) emphasizes the extraction of text segments pivotal for answering

Type	Example	#	%
How	*"How was the murder weapon handled such that it was not discovered at the scene?"*	1,647	41.9
What	*"What's the house number where Smith lives?"*	731	18.6
Which	*"Which building doesn't have any graduatestudents living in this dormitory building?"*	498	12.7
Who	*"Who is the murderer of the painter?"*	459	11.7
Why	*"Why did Harry suspect Filch?"*	378	9.6
When	*"When is Teacher's birthday?"*	167	4.3
Where	*"Where exactly does woman come from?"*	121	3.1
Other	*"Please determine the respective professions of Faulkner, Santiago, and Hemingway."*	378	9.6
All		**3928**	**100**

Table 1: All eight types of questions in DetectBench and their frequency. Note that each question in DetectBench may contain different types of questions.

Human Performance	
Average **Accuracy** in choosing right option	74.1
Average **Accuracy** in underlining right evidence	63.8

Table 2: Human performance in answering questions.

DetectBench Statistic			
#Sample	Avg #Token	Avg #Evidence	Avg #Jumps
396+1928+1604=3928	190.6	4.74	8.90

Table 3: Statistic information of DetectBench.

queries. Nonetheless, the linkage between key information and queries within these benchmarks is overtly conspicuous, allowing for the location of pertinent data through string-matching techniques and facilitating correct answer derivation via one or two inferential leaps.

However, the unique feature of the DetectBench is its reliance on evidence that is widely dispersed and implicit to answer questions.

2.2 Commonsense Reasoning

The exploration of Commonsense Reasoning encompasses a variety of research efforts, traditionally classified into single-hop reasoning, multi-hop reasoning, and reasoning that is uncommon yet plausible. Datasets facilitating single-hop reasoning, such as HellaSwag (Zellers et al., 2019) and WinoGrande (Sakaguchi et al., 2021), present challenges in reasoning through narrative continuation, where the difficulty often resides in the formulation of options and potentially in the design of adversarial options aimed at undermining specific models. Multi-hop reasoning benchmarks like StrategyQA (Geva et al., 2021) annotate the reasoning path, concentrating on the capacity of models to execute multi-hop reasoning in response to questions. Reasoning that is uncommon yet feasible, as demonstrated in datasets like α-NLG (Bhagavatula et al., 2019), d-NLI (Rudinger et al., 2020), and UnCommonsense Reasoning (Zhao et al., 2023; Arnaout et al., 2022), typically originates from pre-existing datasets by selecting the least likely option as the correct response and elucidating the rationale behind this selection.

The DetectBench is categorized as uncommon but plausible multi-step reasoning, which features finding where to start such reasoning tasks. The process of reasoning usually starts with small details that might seem unimportant. However, when looked at more closely, these details help show a clear path that leads to a clear answer.

3 Detective Benchmark

3.1 Construction

The questions in DetectBench are sourced from open-access Detective Puzzle problems, which undergo a series of selection, rewriting, and annotation to construct into the benchmark. DetectBench aims to evaluate the model's abilities in evidence detection and multi-step commonsense reasoning. Therefore, the benchmark should provide the following elements: (1). Question should not contain any ethical problem. (2). Question descriptions should contain lengthy, complex, seemingly unrelated, and even misleading information. (3). The solution to the question should involve multi-step reasoning based on the evidence that can be directly found in the question context. (4). The model's response to the question needs to be capable of being assessed objectively.

Question Selection: To ensure the benchmark focuses on "evidence detection" and "multi-step commonsense reasoning", we thoroughly verify all questions. Given that detective puzzles often contain questions with multiple potential answers and varying reasoning processes, we opt for questions whose answers and reasoning processes are the most rational and unique. Simultaneously, we excluded questions that overly rely on symbolic logic or specialized knowledge because such questions cannot be solved simply by retrieving related information or evidence but also domain knowl-

Context

On a snowy winter night, a tragic event unfolded at 68 King's West Road. A single woman was found murdered at the doorstep of her room around 8pm. The scene was set in a quaint, cozy room, warmed by a gas stove that glowed red-hot, offering a stark contrast to the cold white blanket enveloping the outside world. The soft illumination from the electric light added a serene glow to the room, which, despite its inviting warmth, bore the grim reality of the night's events. The window, tightly sealed against the winter's bite, was veiled by curtains that were drawn halfway, suggesting a hasty or distracted moment.

As the investigation unfolded, the police tape crisscrossed the snow-laden streets, casting eerie shadows under the moonlit night. The neighborhood, usually quiet and reclusive, buzzed with hushed conversations and speculative whispers. Amidst this somber atmosphere, a young man from the vicinity stepped forward, claiming to have witnessed the crime. He recounted seeing the event unfold from his room, situated 20 meters across, at around 11pm. His description was precise—a blond man with black-rimmed glasses and a beard, an image that seemed etched in his memory. Seizing this lead, the authorities apprehended the blonde boyfriend of the deceased, a decision that sent ripples through the community.

In the courtroom, the air was thick with anticipation. The defense lawyer, with a keen eye and a sharper wit, probed the young witness. "You saw the murderer through the window, didn't you?" he asked, his voice steady but laden with implication. The young man, unwavering, affirmed his earlier statement, convinced that the half-drawn curtains and the clear glass had granted him an unobstructed view of the grim spectacle.

Question

Do you think this young man is guilty or not?

Options

A) The young man was telling the truth, and the blond boyfriend was the murderer.
B) The young man lied about the time of witnessing the murder to mislead the investigation.
C) The young man could not have seen the murderer's detailed features due to the room's conditions.
D) The victim had another visitor that night who was the real murderer.

Answer

C) The young man could not have seen the murderer's detailed features due to the room's conditions.

Clue Graph

Evidence:

- "On a blustery snowy winter night, the quaint neighborhood of King's West Road was shrouded in a serene white blanket" → Serene snowy setting
- "an unsettling event unfolded at 68 King's West Road, where a single woman met her untimely demise right at her doorstep, the grim incident estimated to have occurred around the haunting hour of 8pm" → Murder at 68 King's West Road around 8pm.
- "The gas stove in the room blazed with a fierce red, filling the space with a sweltering heat" and "the window, its curtains drawn halfway" → Room's warmth with blazing gas stove, partially open window.
- "I had witnessed the murder last night at around 11pm, and although my room was 20 meters from the scene, I found the murderer to be a blond man with black-rimmed glasses and a beard" → Young man's testimony of murder at 11pm, description of murderer.

Multi-Hop Reasoning:

1. Serene snowy setting + Murder at 68 King's West Road around 8pm → Peaceful night disrupted by murder.
2. Room's warmth with blazing gas stove, partially open window + Young man's testimony of murder at 11pm, description of murderer → Questionable visibility for detailed observation.
3. Lawyer's challenge to the young man's ability to observe detailed features through the fogged window + Young man's specific description → Suggests young man's inside presence and possible guilt.

Figure 2: The example of the question in DetectBench.

edge and special training techniques. Specifically, we excluded five types of questions: 1. Questions that are not ethical or have sensitive content. 2. Questions requiring visual or auditory information to answer. 3. Questions that are anti-logical, have unreasonable answers, or are overly diverse. 4. Questions requiring extensive symbolic logic or domain knowledge. 5. Questions with too obvious evidence.

Question Rewriting: The original puzzle may mix the problem description with the question, sometimes even directly provide the answer, or lack relevant information for reasoning. Therefore, we first rewrite the puzzle into "Context" and "Question" to distinguish between the background description and the query of the question. Then, the original free-text puzzles are converted into a multiple-choice format. The converted format includes "Options" and "Answer" fields to represent the choices and the correct answer. We also constructed a "Evidence Graph" to represent the reasoning process explicitly. We annotated evidence within the context as "Evidence". Based on the evidence, we delineated the "Multi-Hop Reasoning", which encompasses the reasoning process from each piece of evidence as well as joint reasoning based on multiple pieces of evidence.

Manual Verification: All questions processed by the GPT-4-turbo-1106-preview model undergo manual verification. Five annotators are recruited to work with the authors on verification. This includes eliminating questions with unreasonable answers or options that require significant modification. Additionally, detailed adjustments are performed to the options and answers to make them more reasonable. The Appendix B provides detailed requirements and examples for annotation.

3.2 Statistic

The statistic information is shown in Tab. 3. The split of train, dev, and test sets aligns with the current trend of using only a small amount of data for finetuning or in-context learning and a large amount of data for evaluation (Zhou et al., 2023). Each question in DetectBench is organized in JSON format, comprising five main elements: "Context", "Question", "Options", "Answer" and "Evidence Graph" as shown in Fig. 2. Tab. 1 reveals a distinct preference for process-oriented questions for "How" to form the largest category. Compar-

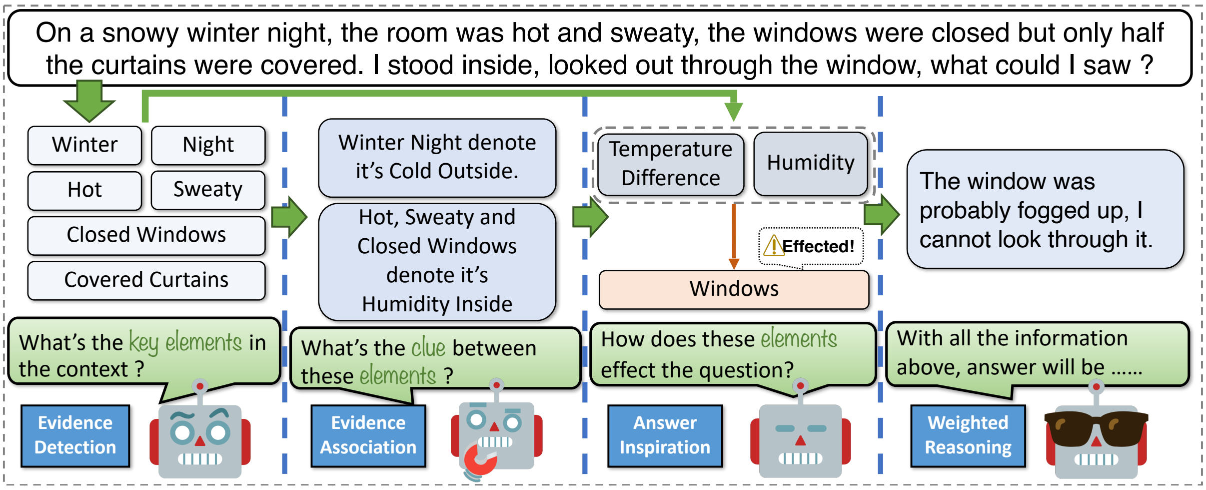

Figure 3: The figure represents the conceptual framework of "Detective Reasoning". The "Detective Reasoning Prompt" method involves providing instructions to an LLM, requiring it to output its thought process directly following the question specifications described in the figure. The Detective Reasoning Finetune involves self-generating data for finetuning the model based on the thought sequence delineated in the figure.

atively, descriptive and person-focused questions, such as "What", "Which", and "Who", are also notably present.

3.3 Human Performance

To propose a human baseline, we invited 50 participants to answer questions from the DetectBench dev set. The examination took three hours, and participants were allowed to leave early if they completed the task. The participants were comprised of undergraduate and graduate students from universities across China, each remunerated at rates exceeding the local minimum hourly wage and bonuses for each correctly answered question.

To facilitate human participation, we translated the benchmark into Chinese and used an online question-and-answer platform to collect answers and measure time spent. Expressions in Chinese or English will not have any additional impact because DetectBench mainly involves commonsense reasoning and contains no language-specific content. Each participant answered 15 questions from a subset of 250 questions from the DetectBench dev set, which ensured that each question was answered by three different participants. Participants are asked to choose the option they think is correct and underline the sentence that is useful to answer the question. The result of the human baseline is shown in Tab. 2.

4 Detective Reasoning

4.1 Detective Reasoning Prompt

The Detective Reasoning Prompt is intended to help the model identify crucial information and extract precise answers through progressively deeper logical reasoning, as demonstrated in Fig. 3. Specially, Detective Reasoning Prompt consists of four stages: **(1) Evidence Detection**, which aims to prompt the model to uncover all evidence, whether useful or not, within the given context. **(2) Evidence Association** requires the model to comprehend the inherent connections between pieces of evidence in the context and generate new related thoughts based on detected evidence. **(3) Answer Inspiration** involves identifying the evidence necessary for solving the given question and initiating reasoning around these pieces of evidence to trigger possible answers. **(4) Weighted Reasoning** reinforces the model's reliance on its generated reasoning process in determining the final answer compared to the overall context. For detailed prompts for each stage, please refer to Appendix C.2.

4.2 Detective Reasoning Finetune

Building upon the aforementioned Detective Reasoning Prompt, we propose a finetuning strategy to further improve the model's evidence detection abilities. For benchmarks that have reasoning processes explicitly annotated, such as our DetectBench, one can concatenate the reasoning outputs for each stage in the Detective Reasoning Prompt as the finetuning data. For benchmarks that have

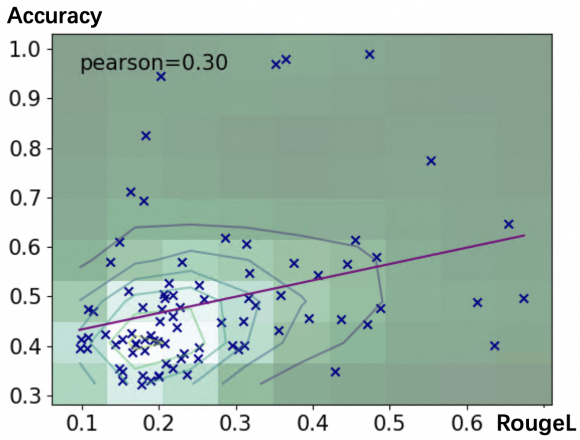

Figure 4: The Pearson Correlation between the evidence detection (RougeL) and reasoning performance (Accuracy) across all models and prompt methods.

only standard answers, the Detective Reasoning Finetune strategy uses the other powerful LLMs to complete the reasoning process based on the questions and answers and then organize this reasoning content into the format as shown in Tab. 18 in Appendix as finetuning data.

5 Experiments

5.1 Overall Setup

LLM Baselines: To test the best performance of the LLMs and ensure replicability, we have used a number of eminent models from both the API-based and the open-source domains. These include GPT4-turbo (**GPT4**) (OpenAI, 2023b), GPT3.5-turbo (**GPT35**) (OpenAI, 2023a), Llama2-7b-Base (**llama2-base**), Llama2-7b-Chat (**llama2-chat**) (Touvron et al., 2023), GLM4 (**GLM4**) (Zheng et al., 2023), ChatGLM3-6b-Base (**chatglm3-base**), and ChatGLM3-6B-Chat (**chatglm3-chat**) (Xu et al., 2023). The experimentation was conducted using the official APIs for GPT4-turbo, GPT-3.5-turbo, and GLM-4 between January 10 and January 29, 2024.

Detective Reasoning: We use four open-source LLMs to explore how Detective Reasoning enhances LLM performance. Our focus is on evaluating the effectiveness of the Detective Reasoning Prompt (**DR Prompt**), fine-tuning using DetectBench data (**DR FT w/ Detective**), and self-generated fine-tuning data based on DetectBench context, question, and answer (**DR FT w/ Generated**). A subset of 398 samples from the training dataset was used for fine-tuning over three epochs with the AdamW optimizer, as detailed in Appendix A. Appendix C.2 provides detailed descriptions of the prompts used in each method.

Prompt Baselines: A range of prompt engineering methods were analyzed for comparative insights: `Naive`, which simply inputs "Context", "Question", and "Options" into LLMs for answers. `Self-CoT` (Kojima et al., 2022), applying a step-by-step reasoning prompt. `Auto-CoT` (Zhang et al., 2022), which automates Chain of Thought (CoT) demonstrations, evaluated in a three-shot setting due to its non-zero-shot design. `Self-Consistency` (Wang et al., 2022), summarizing multiple outputs from the same model to derive a final answer. `Complexity-CoT` (Fu et al., 2022), selecting the longest reasoning steps among all outputs. `Plan-and-Solve CoT (PS-CoT)` (Wang et al., 2023), focusing on problem deconstruction before solution. `Detective Reasoning Prompt`, introduced in this study. `Naive /w Evidence` and `Naive /w Answer`, enhancing inputs with "Evidence" and the "Answer" respectively.

Some methods are not included in the experiments: Methods that involve a self-checking process, such as Tree of Thought (Yao et al., 2023) and Graph of Thought (Besta et al., 2023), were excluded because common sense reasoning is challenging to self-check during intermediate processes. Methods such as Reflexion (Shinn et al., 2023), which increase the probability of a correct answer by injecting model error, were ruled out due to the prior information that would be incurred in choosing options in an option-based QA setting.

Demonstration: Demonstration is about giving some examples in the context to improve LLM's understanding of output format and knowledge acquisition. Naive Prompt appends answers after training data examples, while Auto-CoT guides the LLM in generating reasoning processes aligned with the "Context", "Question", and "Answer".

Metrics: We evaluate the reasoning ability of LLMs based on the **Accuracy (Acc.)** in answering the multiple-choice question on DetectBench and Reclor. HotpotQA proposes to use F1 and Exact Match scores to evaluate models on extracting answers directly from the given context. However, considering that the current mainstream conversational LLMs struggle to generate content identical to the original text directly, we propose to use **RougeL-F.** for evaluation one DetectBench and HotpotQA.

	GPT4		GPT35		GLM4		ChatGLM3-chat		ChatGLM3-base		Llama2-chat		Llama2-base	
	RougeL-F.	Acc.	RougeL-F.	Acc.	RougeL-F.	Acc.	RougeL-F.	Acc.	RougeL-F.	Acc.	RougeL-F.	Acc.	RougeL-F.	Acc.
Naive Questioning														
Naive	44.4	56.5	15.3	33.0	31.1	40.2	15.3	41.3	9.71	39.6	10.8	47.5	10.7	39.6
Naive (3-shot)	40.6	54.4	15.3	34.9	30.3	39.4	10.8	41.8	13.1	42.3	11.5	47.1	9.9	**41.4**
Process Enhanced Method														
Self-CoT	31.4	60.7	17.73	32.3	31.0	45.1	17.0	40.4	21.8	35.4	20.6	50.6	16.6	38.7
Auto-CoT (3-shot)	37.5	56.7	19.91	33.9	**35.5**	43.2	18.1	41.3	22.9	37.5	20.4	47.5	19.9	40.9
Output Ensemble Method														
Self-Consistency	31.7	54.8	18.9	33.0	25.9	**49.4**	14.4	40.3	25.1	37.6	19.3	41.1	25.2	39.7
Complexity-CoT	28.6	61.9	20.0	34.1	28.1	44.8	17.0	40.6	**23.7**	34.3	21.8	50.4	29.5	40.1
Multi-step Chain-of-Thought														
PS-CoT	21.3	52.8	17.9	34.1	21.8	46.1	16.4	**42.5**	18.1	39.1	16.0	51.1	**23.2**	38.5
DR Prompt (ours)	**45.5**	**61.5**	**20.9**	**36.4**	20.1	45.1	**18.9**	42.2	22.3	**43.8**	**25.2**	**52.4**	20.7	40.5
Question with Extra Key Information														
Naive w/ Evidence	65.4	64.8	42.9	34.9	48.3	58.1	22.7	47.9	47.1	44.5	48.7	47.6	61.3	48.9
Naive w/ Evidence (3-shot)	63.6	40.1	39.5	45.6	43.7	45.5	35.8	50.2	31.6	49.7	32.5	48.3	67.4	49.6
Naive w/ Answer	47.3	99.0	20.3	94.5	36.5	98.0	23.0	57.0	18.0	69.4	17.9	47.9	13.7	56.9
Naive w/ Answer (3-shot)	55.3	77.6	18.3	82.5	35.1	97.0	20.8	49.6	16.3	71.3	14.9	35.5	14.9	61.1

Table 4: The performance of baseline models under renowned prompt methods is presented. Results in bold indicate the best results achieved without additional information.

	RougeL-F.		Acc.	
	DetectBench	HotPotQA	DetectBench	ReClor
Llama2-base				
Naive	10.8	30.6	47.5	36.7
DR Prompt	20.7	32.1	40.5	37.5
DR FT w/ Detective	**38.6**	**37.2**	**56.7**	**39.6**
DR FT w/ Generated	32.4	32.8	44.6	33.5
Llama2-Chat				
Naive	10.8	36.3	47.5	38.8
DR Prompt	25.2	39.7	52.4	42.6
DR FT w/ Detective	**40.9**	**41.7**	**58.3**	**45.5**
DR FT w/ Generated	34.6	38.6	50.5	37.1
ChatGLM3-Base				
Naive	9.7	26.8	39.6	30.1
DR Prompt	22.3	25.4	43.8	31.9
DR FT w/ Detective	**37.6**	**34.2**	**50.8**	**36.7**
DR FT w/ Generated	35.4	30.9	43.6	32.9
ChatGLM3-Chat				
Naive	15.3	31.8	41.3	33.0
DR Prompt	18.9	37.6	42.2	38.9
DR FT w/ Detective	**27.1**	**42.3**	**56.3**	**41.7**
DR FT w/ Generated	24.6	38.5	43.5	39.1

Table 5: A detailed comparison of baseline models' performances utilizing Detective Reasoning Prompt and Fine-tuning methodologies is provided. Outcomes in bold signify the most superior results within the same model under these experimental conditions.

5.2 Performance with Different Prompt

Tab. 4 displays the performance of all baseline models across different prompt methods. Based on the results in the table, we have drawn the following conclusions:

Current LLMs struggle with Evidence Detection: We notice a general insufficiency in Evidence Detection, with GPT4-Turbo's average RougeL-F score only being 44.4. Open-source models like ChatGLM3 and Llama2 have even lower scores, at 9.71 and 10.7, respectively.

There is a correlation between Evidence Detection and model reasoning performance: When Evidence is directly fed into LLMs, there is a significant performance improvement. Directly informing GPT4 of the Evidence beneficial to a question enhanced its Evidence Detection by 21%, with a 9.3% increase in reasoning outcomes. Moreover, giving the Answer directly to the LLM enables it to find Evidence consistent with human annotations more accurately. Further, we analyzed the correlation between evidence detection and the final reasoning outcomes in Fig. 4, finding a notable positive correlation.

Additionally, we discovered that telling GPT4 the answer directly could achieve an answer accuracy rate of up to 99%, whereas informing GPT4 directly about what the Evidence is only boosts its evidence accuracy to 65.4%, with other LLMs performing even worse. This may be due to the difficulty LLMs face in producing relevant long texts directly upon request.

Demonstration effects are unstable: As models become increasingly adept at interpreting complex instructions, the historical utility of demonstrations in enhancing model answer parsing has diminished. Across different prompting methods and model types, a 3-shot demonstration led to unstable performance (Gu et al., 2023).

Detective Reasoning Prompt is superior to other method: The Detective Reasoning Prompt significantly enhanced LLMs' evidence detection and reasoning capabilities. Compared to other prompting engineering strategies, this method improved accuracy and demonstrated a broader efficacy, thereby reinforcing its value in enhancing model understanding and reasoning abilities.

5.3 Optimizing Evidence Detection through Detective Reasoning Finetune

Tab. 5 shows the detailed effects of Detective Reasoning Finetune on various models and different data sets, and the analysis is developed based on the following points:

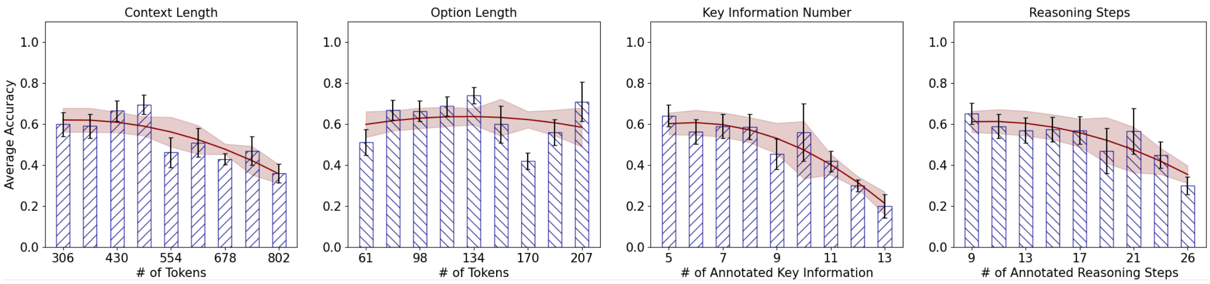

Figure 5: The performance of GPT4-Turbo is correlated with the context length, option length, the number of evidence, and the number of reasoning steps involved.

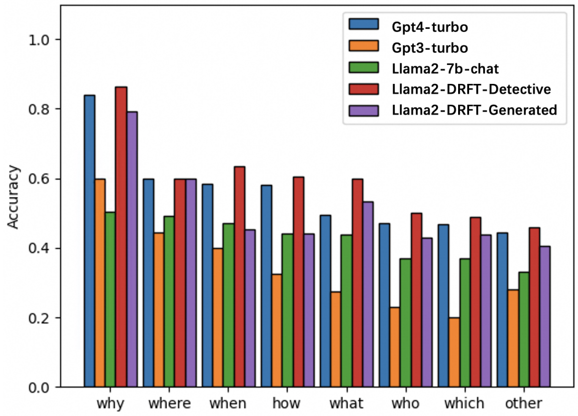

Figure 6: The performance of various models varies across different Question Types.

Joint Improvements in Evidence Detection and Reasoning Performance: Across all models, the DR FT scheme with Detective-style fine-tuning outperforms other approaches in RougeL-F scores on the DetectBench and HotPotQA tasks. For example, the Llama2-base model's score increased to 38.6 on DetectBench and 37.2 on HotPotQA. Additionally, for instance, in the Llama2-Chat model, after the improvement in evidence detection, there was a corresponding rise in reasoning accuracy, with accuracy rates reaching 58.3% This indicates that the model becomes more precise in its reasoning logic after obtaining more accurate Evidence.

Finetune with DetectBench has better performance than self-generated: Using DetectBench data for Detective Reasoning Finetuning boosts evidence detection and reasoning skills in LLMs. The observed improvements include a 15.2% increase in evidence detection accuracy and a 10.5% uplift in overall performance. These results underscore the DetectBench's effectiveness in refining models' information processing and reasoning faculties.

5.4 In-depth Performance Analysis

Factors Effect Reasoning Performance: The analysis of GPT4-Turbo's performance (see Fig. 5) highlights the impact of different context lengths and option lengths on model accuracy. The accuracy markedly decreases from about 65% to 35% as the context length increases from 400 to 800 words. An examination of our annotations based on model performance revealed a strong correlation between the amount of Evidence, depth of reasoning, and performance metrics. Specifically, as the number of evidence instances and the depth of reasoning increase, the model's accuracy significantly decreases, confirming the relationship between problem complexity and model effectiveness.

Varied Performance to Different Question Types: As shown in Fig. 6, the performance differences across various question types indicate that the existing LLMs excel in answering "why" and "where" questions, with the fine-tuned Llama-2 model achieving an impressive accuracy rate of 90%. In contrast, the accuracy rates for "who", "which", and other types of questions hover around 50%. This discrepancy suggests that while the model effectively handles questions requiring an understanding of processes and environments, it struggles with questions that require complex entity recognition and relationship discernment, pointing toward directions for future model improvements.

6 Conclusion

This paper introduces the DetectBench to assess LLMs' abilities in evidence and multi-step commonsense reasoning within a long context. We also propose a novel type of prompt and fine-tuning method named Detective Reasoning to augment LLM's performance in evidence detection and thereby augment performance in commonsense reasoning. The experiment results show that the abilities of evidence detection and reasoning performance are correlated. Detective Reasoning effectively enhances the capability of LLMs in evidence detection, thereby improving the LLMs' commonsense reasoning results in long text contexts.

7 Limitations

DetectBench is designed to facilitate LLMs' abilities in Evidence Detection and Multi-hop Commonsense Reasoning within long contexts. However, compared to the information in real-world scenarios, the complexity and breadth of data in DetectBench are noticeably insufficient. Implementing Detective Reasoning has been proven to effectively enhance the Evidence Detection capability of LLMs, thereby improving reasoning performance. However, this strategy is primarily suitable for tasks that require extracting and reasoning about relevant Evidence from long contexts. If applied in short-text scenarios, where it is necessary to combine implicit knowledge gained from common sense or experiential understanding, its effectiveness would be significantly reduced.

8 Ethical Concerns

Considering that Detective Puzzles may contain many sensitive topics, including but not limited to murder, theft, deception, etc. Existing LLMs might refuse to answer sensitive questions for safety reasons, putting those LLMs that prioritize higher safety standards at a disadvantage when assessed using Detective Puzzles. Additionally, fine-tuning LLMs on such data could inadvertently amplify security vulnerabilities.

To mitigate ethical dilemmas associated with detective reasoning benchmarks, we have invested significant effort and resources to achieve a dual objective: ensuring that models committed to safety do not refuse to answer sensitive questions; and ensuring that the use of DetectBench does not compromise the safety of the models.

References

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. FEVEROUS: Fact extraction and VERification over unstructured and structured information.

Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z Pan. 2022. Uncommonsense: Informative negative knowledge about everyday concepts. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management, pages 37–46.

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

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. 2019. Abductive commonsense reasoning. arXiv preprint arXiv:1908.05739.

Jiangjie Chen and Yanghua Xiao. 2022. Harnessing knowledge and reasoning for human-like natural language generation: A brief review. arXiv preprint arXiv:2212.03747.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2023. Benchmarking large language models in retrieval-augmented generation. arXiv preprint arXiv:2309.01431.

Qiao Cheng, Juntao Liu, Xiaoye Qu, Jin Zhao, Jiaqing Liang, Zhefeng Wang, Baoxing Huai, Nicholas Jing Yuan, and Yanghua Xiao. 2021. Hacred: A large-scale relation extraction dataset toward hard cases in practical applications. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2819–2831.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. 2023. A survey of chain of thought reasoning: Advances, frontiers and future. arXiv preprint arXiv:2309.15402.

Ernest Davis. 2023. Benchmarks for automated commonsense reasoning: A survey. ACM Computing Surveys, 56(4):1–41.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. arXiv preprint arXiv:2210.00720.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. Transactions of the Association for Computational Linguistics, 9:346–361.

Zhouhong Gu, Xiaoxuan Zhu, Haoning Ye, Lin Zhang, Jianchen Wang, Sihang Jiang, Zhuozhi Xiong, Zihan Li, Qianyu He, Rui Xu, et al. 2023. Xiezhi: An ever-updating benchmark for holistic domain knowledge evaluation. arXiv preprint arXiv:2306.05783.

Samreen Kazi and Shakeel Khoja. 2021. Uquad1. 0: Development of an urdu question answering training data for machine reading comprehension. arXiv preprint arXiv:2111.01543.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. Advances in neural information processing systems, 35:22199–22213.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022a. Unified structure generation for universal information extraction. arXiv preprint arXiv:2203.12277.

Yuxuan Lu, Jingya Yan, Zhixuan Qi, Zhongzheng Ge, and Yongping Du. 2022b. Contextual embedding and model weighting by fusing domain knowledge on biomedical question answering. In Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, pages 1–4.

Jose L Martinez-Rodriguez, Aidan Hogan, and Ivan Lopez-Arevalo. 2020. Information extraction meets the semantic web: a survey. Semantic Web, 11(2):255–335.

OpenAI. 2023a. Chatgpt: Optimizing language models for dialogue. https://openai.com/blog/chatgpt.

OpenAI. 2023b. Gpt-4 technical report.

Rachel Rudinger, Vered Shwartz, Jena D Hwang, Chandra Bhagavatula, Maxwell Forbes, Ronan Le Bras, Noah A Smith, and Yejin Choi. 2020. Thinking like a skeptic: Defeasible inference in natural language. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4661–4675.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. Communications of the ACM, 64(9):99–106.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In Thirty-seventh Conference on Neural Information Processing Systems.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. Fever: a large-scale dataset for fact extraction and verification. arXiv preprint arXiv:1803.05355.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The FEVER2.0 shared task. In Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

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

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

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions.

Yang Yang, Zhilei Wu, Yuexiang Yang, Shuangshuang Lian, Fengjie Guo, and Zhiwei Wang. 2022. A survey of information extraction based on deep learning. Applied Sciences, 12(19):9691.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. arXiv preprint arXiv:1809.09600.

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. arXiv preprint arXiv:2305.10601.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. Reclor: A reading comprehension dataset requiring logical reasoning. arXiv preprint arXiv:2002.04326.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? arXiv preprint arXiv:1905.07830.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. arXiv preprint arXiv:2210.03493.

Wenting Zhao, Justin T Chiu, Jena D Hwang, Faeze Brahman, Jack Hessel, Sanjiban Choudhury, Yejin Choi, Xiang Lorraine Li, and Alane Suhr. 2023. Uncommonsense reasoning: Abductive reasoning about uncommon situations. arXiv preprint arXiv:2311.08469.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. arXiv preprint arXiv:2306.05685.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, L. Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. ArXiv, abs/2305.11206.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. arXiv preprint arXiv:2308.07107.

A Training Details

For the models llama2-7b-base, llama2-7b-chat, ChatGPT3-6b-base, and ChatGPT3-6b-chat, we executed two distinct training methodologies:

1. Directly utilizing the training data from the Detective Reasoning Benchmark to compose the Detective Reasoning Finetune data.

2. Employing the "Context", "Question", and "Answer" in Detective Reasoning Benchmark to automatically generate Detective Reasoning Finetune data.

The specific training parameters are detailed in Tab. 6.

B Detail about Manual Annotation

B.1 Details about Annotators

The annotators for this research are the authors of this paper themselves, who are experts in the field of Computer Science and Cognitive Psychology. The entire annotation process was under the stringent supervision and scrutiny of the first author of this paper.

B.2 Annotation Tasks and Goals

The purpose of the manual annotation tasks was twofold. The first goal was to obtain comprehensive annotated datasets that encapsulate the essential features of the target text, which could be further leveraged for tasks such as training, testing, and model evaluation. The second goal was to provide a detailed, rigorous, and systematic assessment of the annotated data quality to assess its fit and reliability for the subsequent analysis. All the detailed annotation tasks and targets are listed in Tab. 7.

B.3 Case of Annotation

In our efforts to delineate the complex annotation process and ensure the replicable rigor of experiments, this section provides an in-depth display of the manual annotation cases. The aim is to elucidate the categorical distinctions and precise definitions adopted in the annotations, thereby facilitating fellow researchers in ascertaining the veracity of the annotated data. Representative cases from the annotation process have been cataloged in Tab. 8 for comprehensive reference and understanding.

C Experiments Details

C.1 Parameters in Inference

Our experiments involved two types of hyperparameters. The first type pertains to the seeds of random numbers used in various Python libraries, while the second type refers to the hyperparameters used when invoking the AutoCausalLM class from the transformers library for generation. We configured our settings as demonstrated in Table 9.

C.2 Prompt Details

This section primarily showcases the prompts employed by all Prompt Engineers throughout the experiment.

Table 10 displays the `Naive` prompts, Table 11 presents the `Naive w/ Key Info` prompts, Table 12 outlines the `Naive w/ Answer` prompts, Table 13 features the `Self-CoT` prompts, Table 15 exhibits the `Self-Consistency` prompts, Table 16 reveals the `Complexity-CoT` prompts, Table 17 shows the `PS-CoT` prompts, Table 18 displays the `Detective Reasoning Prompt` prompts, and

Training Detail				
# of Samples	# of Tokens	# of epochs	warm_up steps	learning rate
396	162,868	3	200	1e-5

Table 6: All the parameter setting in the training process.

Task	**Requirements**
Question Verification	1.1 Delete if answering the question requires non-text information, like audio or image. 1.2 Delete if there is a substantial amount of mathematical content or involve of too much domain knowledge. 1.3 Delete if there is no ample presence of daily scenarios. 1.4 Delete if the answer is not correct. 1.5 Delete if there is any discrimination or bias concerning gender, race, nation, or religion.
Question Rewrite	2.1 Standardize the Expression. 2.2 Rewrite a decent answer to the question. 2.3 Separate “Question”and “Context”. 2.4 Write decent and confusing “Options” of the question.
Clue Graph Construction	3.1 Regenerate or rewrite if the “Key Information of Context” cannot exact match to the text in “Context”. 3.2 Regenerate or rewrite if the connection or reasoning is redundant. 3.3 Delete the question or rewrite it there lack of important reasoning processes or connections in Clue Graph.

Table 7: All tasks that require manual annotation, along with the specific requirements for each task.

Task	Requirements	Cases
Question Verification	Delete if answering the question requires non-text information, like audio or image.	Context: “Listen to the following music clip...” Question: “What instrument is playing?” Hint: “Consider the type of information required to answer the question.” Answer: “Piano”
	Delete if there is a substantial amount of mathematical content.	Context: “Consider the mathematical proof of Fermat’s Last Theorem...” Question: “Can you explain the proof?” Hint: “Focus on the subject matter of the proof.” Answer: “It’s a complex proof involving modular forms...”
	Delete if there is no ample presence of daily scenarios.	Context: “In a quantum physics experiment...” Question: “What is the result?” Hint: “Consider the context of the experiment.” Answer: “A specific quantum state”
	Delete if the answer is not correct.	Context: “The cat is on the roof” Question: “Where is the cat?” Hint: “Check the location mentioned in the context.” Answer: “In the garden”
	Delete if there is any discrimination or bias concerning gender, race, nation, or religion.	Context: “All people from X are lazy...” Question: “What are people from X like?” Hint: “Considering the description of X.” Answer: “Lazy”
Question Rewrite	Standardize the Expression.	Original: “〈 /span 〉 A family decides to move into the city and looks for a house. \n \n There are three ...” Rewritten: “A family decides to move into the city and looks for a house. There are three ... ”
	Rewrite a decent answer to the question.	Original Answer: “This is a famous question, in my thought, the answer is ......” Rewritten Answer: “The answer is ......”
	Separate “Question” and “Context”.	Original: Context and Question: “In 1862, during the American Civil War, the Battle of Antietam took place near Sharpsburg, Maryland... What was the significance of the Battle of Antietam?” Separated: Context: “In 1862, during the American Civil War, the Battle of Antietam took place near Sharpsburg, Maryland...” Question: “What was the significance of the Battle of Antietam?”
	Write decent and confusing “Options” of the question.	Context: As the investigation unfolded, the police tape crisscrossed the snow-laden streets, casting eerie shadows under the moonlit night. The neighborhood, usually quiet and reclusive... Question: Do you think this young man is guilty or not? Answer: The young man could not have seen the murderer’s detailed features due to the room’s conditions Options: A) The young man was telling the truth, and the blond boyfriend was the murderer. B) The young man lied about the time of witnessing the murder to mislead the investigation. C) The young man could not have seen the murderer’s detailed features due to the room’s conditions. D) The victim had another visitor that night who was the real murderer
Clue Graph Construction	Regenerate or rewrite if the “Key Information of Context” cannot exact match to the text in “Context”.	Original Context: “On a snowy winter night ...” Key Information: “On a blustery snowy winter night” Rewritten Key Information: “On a snowy winter night ...”
	Regenerate or rewrite if the connection or reasoning is redundant	Original Reasoning Process: “Serene snowy setting + Murder at 68 King’s West Road around 8pm → Peaceful night disrupted by murder Rewritten: Reasoning Process: ~~“Serene snowy setting + Murder at 68 King’s West Road around 8pm → Peaceful night disrupted by murder~~
	Delete the question or rewrite it there lack of important reasoning processes or connections in Clue Graph.	-

Table 8: The examples in our annotation process

Random Seed				
torch.manual_seed	torch.cuda.manual_seed_all	numpy.random.seed	random.seed	torch.backends.cudnn.deterministirc
42	42	42	42	True

AutoCausalLM				
temperature	top_p	top_k	num_beams	max_new_token
0.95	0.95	5	2	2000

Table 9: All the parameter setting in model inference in our experiments.

```
# -*- coding: utf-8 -*-
Variables:
!<INPUT 0>! – Context
!<INPUT 1>! – Question
!<INPUT 2>! – Options
<commentblockmarker>###</commentblockmarker>
Below I will give you a detective reasoning question, please summarize the key clues in this question based on the Context, the options and choose the answer you think is correct. Note: When generating the answer, please only output the serial number of the option.
### Context:
!<INPUT 0>!
### Question:
!<INPUT 1>!
### Options:
!<INPUT 2>!
Your output will contain the following: ### Evidence: Please output what you consider to be the Evidence in the Context. Please note that the Evidence needs to be directly from the Context, i.e. it is a string originally in the Context that can be matched directly to the original text by string matching. ### Answer: please output only the serial numbers.
Please follow the format below for your output:
### Evidence: xxxxx
### Answer: 1/2/3/4
```

Table 10: Prompt of `Naive` method

```
# -*- coding: utf-8 -*-
Variables:
!<INPUT 0>! – Context
!<INPUT 1>! – Question
!<INPUT 2>! – Evidence !<INPUT 3>! – Options
<commentblockmarker>###</commentblockmarker>
Below I will give you a detective reasoning question, please summarize the key clues in the question based on the Context, the options, and the answer, and choose the answer you think is correct. Note: When generating the answer, please output only the serial number of the option.

### Context:
!<INPUT 0>!

### Question:
!<INPUT 1>!

### Evidence:
!<INPUT 2>!

### Option:
!<INPUT 3>!
Your output will contain the following:
### Evidence: Please output what you consider to be the Evidence in the Context. Please note that the Evidence needs to be directly from the Context, i.e. it is a string originally in the Context that can be matched directly to the original text by string matching.
### Answer: please output only the serial numbers.

Please follow the format below for your output:

### Evidence:
xxxxx

### Answer:
1/2/3/4
```

Table 11: Prompt of `Naive w/ Evidence` method

-*- coding: utf-8 -*-
Variables:
!<INPUT 0>! – Context
!<INPUT 1>! – Question
!<INPUT 2>! – Options
!<INPUT 3>! – Answer

<commentblockmarker>###</commentblockmarker>

Below I will give you a detective reasoning question, please summarize the key clues in the question based on the Context, the options, and the answer, and choose the answer you think is correct.
Note: When generating the answer, please output only the serial number of the option.

Context:
!<INPUT 0>!

Question:
!<INPUT 1>!

Options:
!<INPUT 2>!

Answer: !<INPUT 3>!

Your output will contain the following:
Evidence: Please output what you consider to be the Evidence in the Context. Please note that the Evidence needs to be directly from the Context, i.e. it is a string originally in the Context that can be matched directly to the original text by string matching.
Answer: please output only the serial numbers.

Please follow the format below for your output:

Evidence: xxxxx
Answer:
1/2/3/4

Table 12: Prompt of `Naive w/ Answer` method

```
# -*- coding: utf-8 -*-

Variables:
!<INPUT 0>! – Context
!<INPUT 1>! – Question
!<INPUT 2>! – Options

<commentblockmarker>###</commentblockmarker>

Below I will give you a detective reasoning question, please generate your thought process step by step based on the Context and the options and choose the answer you think is correct.
Note: When generating the answer, please output only the serial number of the option.

### Context:
!<INPUT 0>!

### Question:
!<INPUT 1>!

### Options:
!<INPUT 2>!

Your output will contain the following:
### Thought: please output your thinking process step by step.
### Evidence: Please output what you think is the Evidence in the Context. Please note that the Evidence needs to be directly from the Context, i.e. it is a string originally in the Context that can be matched directly to the original text by string matching.
### Answer: please output only the serial numbers.

Please have your output follow the format below:

### Thought:
xxxxxx

### Evidence:
xxxxx

### Answers:
1/2/3/4
```

Table 13: Prompt of `Self-CoT` method

```
# -*- coding: utf-8 -*-

Variables:
!<INPUT 0>! – Demonstration
!<INPUT 1>! – Context
!<INPUT 2>! – Question
!<INPUT 3>! – Options

<commentblockmarker>###</commentblockmarker>

### Demonstration
!<INPUT 0>!

### Context:
!<INPUT 1>!

### Question:
!<INPUT 2>!

### Options:
!<INPUT 3>!

Your output will contain the following:
### Thought: please output your thinking process step by step.
### Evidence: Please output what you think is the Evidence in the topic. Please note that the Evidence needs to be directly from the question, i.e. it is the original string in the question, which can be matched directly to the original text by string matching.
### Answer: When generating answers, please output only the serial numbers of the options.

Please follow the format below for your output:

### Thought:
xxxxx

### Evidence:
xxxxx

### Answer:
1/2/3/4
```

Table 14: Prompt of `Auto-CoT` method

```
# -*- coding: utf-8 -*-

Variables:
!<INPUT 0>! – Context
!<INPUT 1>! – Question
!<INPUT 2>! – Options

<commentblockmarker>###</commentblockmarker>

Below I will give you a detective reasoning question, please generate your thought process step by step
based on the Context and the options and choose the answer you think is correct.
Note: When generating the answer, please output only the serial number of the option.

### Context:
!<INPUT 0>!

### Question:
!<INPUT 1>!

### Options:
!<INPUT 2>!

Your output will contain the following:
### Thought: please generate 5 completely different perspectives of your reflections based on the questions
and options.
### Summary: Please output a summary of all your thinking.
### Evidence: Please output what you think is the Evidence in the Context. Please note that the Evidence
needs to be directly from the Context, i.e. it is the original string in the Context, which can be matched
directly to the original text by string matching.
### Answer: please output only the serial numbers.

Please have your output follow the format below:

### Thought:
1. xxxxxx
2. xxxxxx
3. xxxxxx
4. xxxxxx
5. xxxxxx

### Summarize:
xxxxxx

### Evidence:
xxxxx

### Answers:
1/2/3/4
```

Table 15: Prompt of `Self Consistency` method

-*- coding: utf-8 -*-

Variables:
!<INPUT 0>! – Context
!<INPUT 1>! – Question
!<INPUT 2>! – Options
!<INPUT 3>! – Longest Chain of Thought

<commentblockmarker>###</commentblockmarker>

Below I will give you a detective reasoning question, please generate your thought process step by step based on the question and the options and choose the answer you think is correct.
Note: When generating the answer, please output only the serial number of the option.

Context:
!<INPUT 0>!

Question:
!<INPUT 1>!

Options:
!<INPUT 2>!

Chain of thought:
!<INPUT 3>!

Your output will contain the following: ### Evidence: Please output what you consider to be the Evidence in the topic. Please note that the Evidence needs to be directly from the topic, i.e. it is a string originally in the topic that can be matched directly to the original text by string matching.
Answer: please output only the serial numbers.

Please follow the format below for your output:

Evidence:
xxxxx

Answer:
1/2/3/4

Table 16: Prompt of `Complexity CoT` method

-*- coding: utf-8 -*-
Variables:
!<INPUT 0>! – Context
!<INPUT 1>! – Question
!<INPUT 2>! – Options

<commentblockmarker>###</commentblockmarker>

Below I will give you a detective reasoning question, please generate your thought process step by step based on the Context and the options and choose the answer you think is correct.
Note: When generating the answer, please output only the serial number of the option.

Context:
!<INPUT 0>!

Question:
!<INPUT 1>!

Options:
!<INPUT 2>!

Your output will contain the following:
Thought: Please start with a general plan of how you intend to deal with the problem, and then think step-by-step about how to solve it based on your plan.
Evidence: please output what you think is the Evidence in the Context. Please note that the Evidence needs to be directly from the Context, i.e. it is the original string in the Context, which can be matched directly to the original text by string matching.
Answer: please output only the serial numbers.

Please have your output follow the format below:

Thought:
xxxxxx

Evidence:
xxxxx

Answer:
1/2/3/4

Table 17: Prompt of `Plan and Solve CoT` method

```
# -*- coding: utf-8 -*-

Variables:
!<INPUT 0>! – Context
!<INPUT 1>! – Question
!<INPUT 2>! – Options

<commentblockmarker>###</commentblockmarker>

Below I will give you a detective reasoning question, please generate your thought process step by step based on the Context and the options and choose the answer you think is correct.
Note: When generating the answer, please output only the serial number of the option.

### Context:
! <INPUT 0>!

### Question:
! <INPUT 1>!

### Options:
! <INPUT 2>!

Your output will contain the following:
### Clues: Feel free to summarize all possible clues in the Context
### Connection: Feel free to correlate the clues you summarized above and introduce new clues that may exist.
### Thought: Feel free to reason and think deeply about the clues you have summarized in the two steps above.
### Summarize: Summarize all the thinking from the perspective of solving the problem in the Context.
### Evidence: Please output what you think is the Evidence in the Context. Please note that the Evidence needs to be the direct content of the Context, i.e. it is the original string in the Context, which can be matched directly to the original text by string matching.
### Answer: Please output only the serial number.

Please have your output follow the format below:

### Clues:
xxxxxx

### Connection:
xxxxxx

### Thought:
xxxxxx

### Summarize:
xxxxxx

### Evidence:
xxxxx

### Answer:
1/2/3/4
```

Table 18: Prompt of `Detective Reasoning` method