Piecing Together Clues: A Benchmark for Evaluating the Detective Skills of Large Language Models

Anonymous ACL submission

Abstract

Detectives frequently engage in information detection and reasoning simultaneously when making decisions across various cases, especially when confronted with a vast amount of information. With the rapid development of large language models (LLMs), evaluating how these models identify key information and reason to solve questions becomes increasingly relevant. We introduces the DetectBench, a reading comprehension dataset designed to assess a model's ability to jointly ability in key information detection and multi-hop reasoning when facing complex and implicit information. The DetectBench comprises 3,928 questions, each paired with a paragraph averaging 190 tokens in length. To enhance model's detective skills, we propose the Self-Question Framework. These methods encourage models to identify all possible clues within the context before reasoning. Our experiments reveal that existing models perform poorly in both information detection and multi-hop reasoning. However, the Self-Question Framework approach alleviates this issue.

1 Introduction

The essence of detective skills in handling vast amounts of information across various cases lies in the simultaneous processes of locating information and reasoning from it. Experienced detectives typically begin by identifying the information they require, isolating the crucial details, deducing insights from these details, and subsequently making informed decisions. With the development of LLMs (OpenAI, 2023a; Touvron et al., 2023), it raise a question that *Do LLMs possess akin detective capabilities for identifying key information and employing it for effective reasoning and problem-solving when facing with complicated information?*

When facing overloaded information in real-cases, obtaining key information is not always straightforward. This typically requires that models not only comprehend the question's meaning fully but also understand multiple approaches to solve it, allowing them to identify the specific information they need to search for. For example, as shown in Figure 1, only when we realize that changes in temperature and humidity can make glass foggy, can we figure out that details about temperature and humidity are key to seeing through the glass. There has many existing benchmarks evaluate the model's joint abilities in information detection and multi-hop reasoning, 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). However, in these tests, the important information often links directly to the question, which can be found through searching for specific keywords or characters.

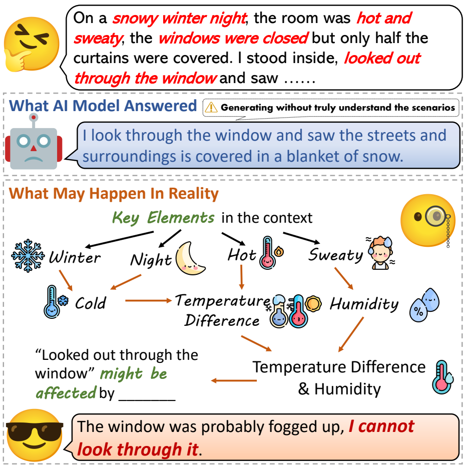

On a snowy winter night, the room was hot and sweaty, the windows were closed but only half the curtains were covered. I stood inside, looked out through the window and saw ……
What AI Model Answered
Generating without truly understand the scenarios
I look through the window and saw the streets and surroundings is covered in a blanket of snow.
What May Happen In Reality
Key Elements in the context
Winter
Night
Hot
Sweaty
Cold
Temperature Difference
Humidity
"Looked out through the window" might be affected by ________
Temperature Difference & Humidity
The window was probably fogged up, I cannot look through it.

Figure 1: When facing overloaded information LLMs may produce outputs arbitrarily due to their inability to engage in deep contemplation. In contrast, humans who are experienced, like detectives, analyze and correlate all available information, thereby identifying pivotal clues that lead to the answer of the problem.

Inspired by the combined clues mining and reasoning done by detectives when facing huge amount of implicit information, we designed the

DetectBench. This benchmark includes 3,928 questions, with each question pair with a paragraph that averages 190 tokens, to measure how well models in finding and reasoning from key information in complex texts to answer questions. DetectBench mimics the intricate stories, situations, and character interactions found in detective puzzles, offering a challenging test of reasoning. We transformed original detective puzzles into multiple-choice question&answer benchmark consists of context, question, options, answer, and an explanation of the answer. Each question comes with a thorough explanation of how to arrive at the answer, using what we call "Clue Graphs", as seen at the bottom of Figure 1. These graphs start with important information that exact matches to the text in context, then step by step, they connect new clues through reasoning and linking information, leading directly to the answer.

In experiments conducted on human participants and LLMs, we assessed their abilities to discover key information and provide the most reasonable answers. We found that humans significantly outperformed the most advanced LLMs in both tasks. Most models, even those capable of locating key information in other information retrieving benchmarks, struggled with the detective reasoning task. However, when key information is provided to the models as contextual input, their reasoning performance improves significantly. This demonstrates the effectiveness of the DetectBench and emphasizes the fact that finding needle in a hay stack is critical to problem solving.

To jointly enhance model's key information detection ablities and reasoning abilities, we proposed two baseline methods: Self-Question Prompt and Self-Question Finetune. The Self-Question Prompt aims to enhance the zero-shot reasoning capabilities of existing LLMs by guiding them to consider all possible clues comprehensively, reason, and then summarize and refine content related to the question. The Self-Question Finetune, by utilizing benchmarks annotated with reasoning processes or constructing data through reasonable guidance from answers, effectively enhances model's abilities to discover key information and perform multi-hop reasoning.

Using the Self-Question Prompt directly improved model's effectiveness in discovering key information and solving problems compared to other prompt engineers. Moreover, when using data from the DetectBench for Self-Question Finetune, LLMs not only achieved significant improvements on the DetectBench but also showed noticeable performance enhancements on other benchmarks requiring information mining and reasoning.

In summary, the main contributions of this study include:

1. The introduction of the DetectBench, providing a new standard for assessing model's key information detection and reasoning abilities.

2. The development of the Self-Question Prompt and Self-Question Finetune method, significantly enhancing model's joint performance in information detection and reasoning abilities.

3. Through extensive experiments, the limitations of existing models in discovering key information and conducting deep reasoning are verified, and it was shown that these limitations could be mitigated after using the Self-Question Prompt/Finetune methods.

2 Related Works

2.1 Information Retrieval

The domain of Information Retrieval aims to address pertinent tasks through the extraction of crucial data from a plethora of references, where the most significant challenge lies in the identification of implicit key information (Zhu et al., 2023; Yang et al., 2022). Traditional benchmarks in Information Retrieval have historically segmented the task of Information Extraction for the purpose of evaluating 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 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

Benchmark	# of Questions	Ave. Length	Explanation to Answer	Ansering Format	Metrics
HotpotQA (Yang et al., 2018)	112,779	137.9		Free Text	Rouge
HellaSwag (Zellers et al., 2019)	59,950	38.5		Choice QA	Accuracy
Reclor (Yu et al., 2020)	6,138	66.4		Choice QA	Accuracy
WinoGrande (Sakaguchi et al., 2021)	12,282	21.1	✓	Choice QA	Accuracy
StrategyQA (Geva et al., 2021)	2,780	9.6	✓	Bool QA	Accuracy
DetectBench	396 (train)+1928 (dev) +1604 (test)=3,928 (all)	190.2	✓	Choice QA & Free Text Reasoning	Accuracy & Rouge

Table 1: The comparison between the DetectBench with other and Information Retrieval Benchmarks and Common Sense Reasoning Benchmarks.

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

Table 2: All eight question type in Detective Reasoning and their frequency.

Human Performance	
Average Accuracy	74.1%
Top Accuracy	93.3%
Lowest Accuracy	53.3%

Table 3: Human performance in answering questions.

and facilitating correct answer derivation via one or two inferential leaps.

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 commonsense 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. In contrast, multi-hop reasoning benchmarks like StrategyQA (Geva et al., 2021) annotate the reasoning trajectory, concentrating on the capacity of models to execute multi-hop reasoning in response to posed 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 framework is categorized as uncommon but plausible multi-step thinking, feature on finding where to start such thinking tasks. The process of thinking usually starts with small details that might seem unimportant. But, when looked at more closely, these details help show a clear path that leads to a clear answer.

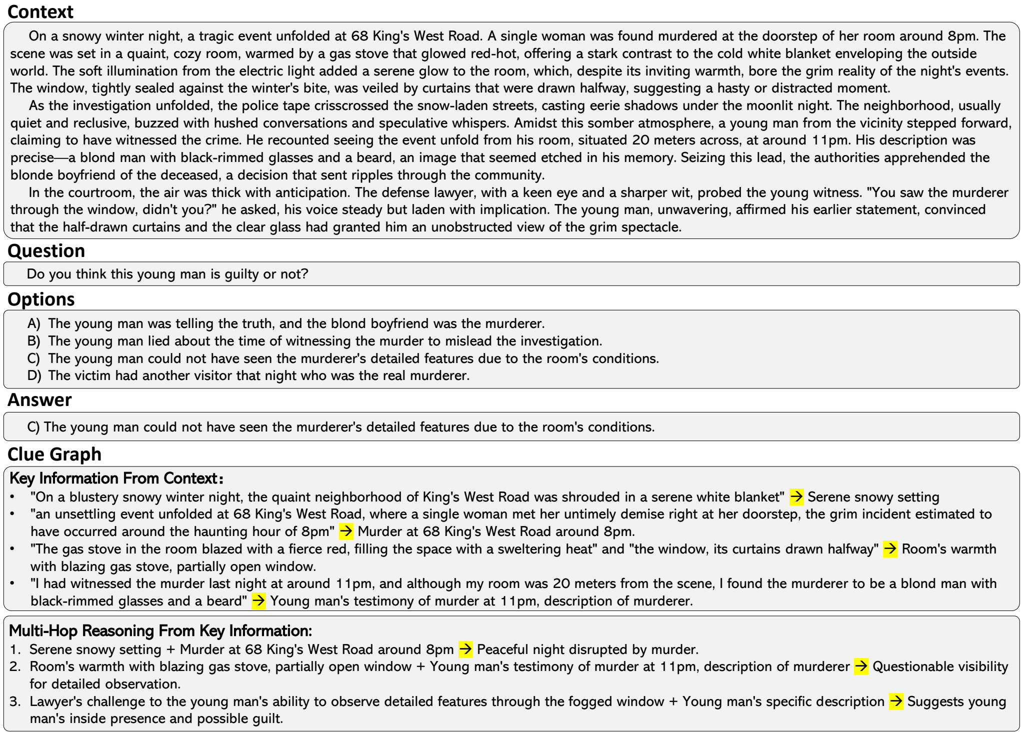
3 Benchmark Construction

3.1 Benchmark Construction

The DetectBench aims to evaluate model's joint abilities in information detection and multi-step commonsense reasoning. Therefore, benchmark should provide the following elements: (1). Question that lack ethical integrity or encompass topics of a sensitive nature. (2). Question descriptions should contain lengthy, complex, and seemingly unrelated information. (3). The solution to the question should involve multi-step reasoning based on the original information. (4). The model's response to the question should be assessed objectively and accurately.

Each question is organized in JSON format, comprising five main elements: "Context", "Question", "Options", "Answer" and "Clue Graph" as shown in Fig. 2. Data processing includes question selection, question rewriting, and manual verification stages, with the first two stages primarily assisted by the GPT-4-turbo-1106-preview model.

Question Selection: To ensure the benchmark focuses on "key information discovery" and "multi-



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

Key Information From Context:

- "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 From Key Information:

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

step commonsense reasoning", we screened questions. Given the potential for multiple answers and reasoning paths in detective reasoning questions, we endeavored to ensure each question's reasoning scheme was as clear and straightforward as possible to ensure the reasonableness and uniqueness of the answers and reasoning processes. Simultaneously, we excluded questions overly dependent on symbolic logic or specialized knowledge because such questions cannot be solved simply by retrieving related infomation. Specifically, we excluded five types of questions: 1. Questions that are not ethical or have sensitive matters. 2. Questions requiring visual or auditory information for support; 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 overly obvious key information.

Question Rewriting: The original questions might conflate the problem description with the question itself, sometimes even revealing the answer directly, and some questions do not contain much irrelevant content. Therefore, we needed to rewrite the questions, using "Context" and "Question" to distinguish between the background of the problem and the inquiry. We converted the original natural text questions into multiple-choice format, with "Options" and "Answer" fields representing the choices and the correct answer. Addationally, we construct "Clue Graph" to explicitly represent the reasoning process. We annotated important content within the original text as "Key Information from Context". Based on these key information, we delineated the "Multi-Hop Reasoning From Key Information" encompasses the reasoning process from a single piece of information as well as joint reasoning based on multiple pieces of information.

Manual Verification: All questions processed by the GPT-4-turbo-1106-preview model are subject to manual verification. We recruited five annotators to work with the authors on verification, which included initial screening (eliminating questions whose answers or options were unreasonable or required significant modification) and detail adjustment (fine-tuning options and answers to make them more reasonable and natural). Specific requirements and examples for annotation are detailed in the Appendix B.

3.2 Human Performance

To explore the nuances of human performance on the DetectBench and to compile benchmark results, we engaged 50 human participants to address questions within the dev set. The examination spanned a total of three hours, with participants afforded the option to leave upon early completion. This cohort consisted of undergraduate and graduate students

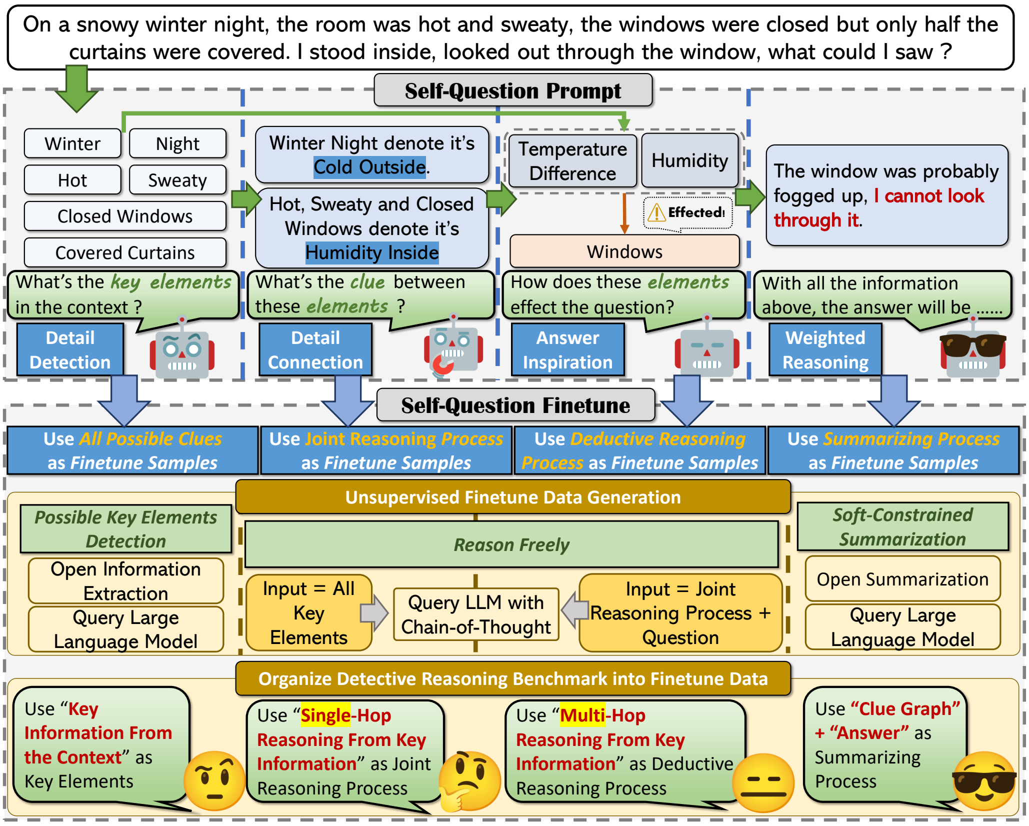

On a snowy winter night, the room was hot and sweaty, the windows were closed but only half the curtains were covered. I stood inside, looked out through the window, what could I saw ?
Self-Question Prompt
Winter
Night
Hot
Sweaty
Closed Windows
Covered Curtains
What's the key elements in the context ?
Detail Detection
Winter Night denote it's Cold Outside.
Hot, Sweaty and Closed Windows denote it's Humidity Inside
What's the clue between these elements ?
Detail Connection
Temperature Difference
Humidity
Effected
Windows
How does these elements effect the question?
Answer Inspiration
The window was probably fogged up, I cannot look through it.
With all the information above, the answer will be ......
Weighted Reasoning
Self-Question Finetune
Use All Possible Clues as Finetune Samples
Use Joint Reasoning Process as Finetune Samples
Use Deductive Reasoning Process as Finetune Samples
Use Summarizing Process as Finetune Samples
Unsupervised Finetune Data Generation
Possible Key Elements Detection
Open Information Extraction
Query Large Language Model
Reason Freely
Input = All Key Elements
Query LLM with Chain-of-Thought
Input = Joint Reasoning Process + Question
Soft-Constrained Summarization
Open Summarization
Query Large Language Model
Organize Detective Reasoning Benchmark into Finetune Data
Use "Key Information From the Context" as Key Elements
Use "Single-Hop Reasoning From Key Information" as Joint Reasoning Process
Use "Multi-Hop Reasoning From Key Information" as Deductive Reasoning Process
Use "Clue Graph" + "Answer" as Summarizing Process

Figure 3: Within the Self-Question Prompt paradigm, the process is bifurcated into distinct phases: Detail Detection and Detail Connection, followed by Answer Inspiration and Weighted Reasoning. The Self-Question Finetune strategy is predominantly aimed at collecting data for fine-tuning. The first three phases permits free generation via open-source models, culminating in the aggregation of these outputs into a cohesive answer during the final stage.

from universities across China, each remunerated at rates exceeding the local minimum hourly wage. Additionally, participants were awarded bonuses for each correctly answered question. For the purpose of facilitating human participation, the benchmark was translated into Chinese, and responses were provided in the same language.

We utilized an established online question-and-answer platform. Each participant was tasked with responding to 15 questions, employing a subset of 250 questions from the dev set of the DetectBench. This approach ensured that each question received responses from three distinct participants. The performance of humans on the DetectBench is documented in Tab. 2.

4 Self-Question Method

4.1 Construction of the Self-Questioning Model

The construction of the self-questioning model encompasses four primary stages: Detail Detection, Detail Association, Answer Elicitation, and Weighted Reasoning. This process is designed to enable the model to identify key information and thereby extract precise answers through progressively deeper logical reasoning as shown the above of Fig. 3. Detailed prompts for each stage is provided in the Appendix C.2.

Detail Detection aims to stimulate the model to unearth details and facts within the given content, especially those not explicitly stated in the original text. **Detail Association** ask the model to understand the intrinsic connections between pieces of information in the text and to generate new related information based on identified details. **Answer Elicitation** is to identify key information crucial for solving the question and to initiate reasoning around this information to trigger possible answers. **Weighted Reasoning** to reinforce the model's reliance on its generated reasoning outcomes, value these outcomes more in the determination of the final answer compared to the overall context.

4.2 Self-Question Finetune

Building upon the aforementioned self-questioning model, we propose a finetuning strategy for jointly enhancing the ability of information detection and commonsense reasoning as shown the below of

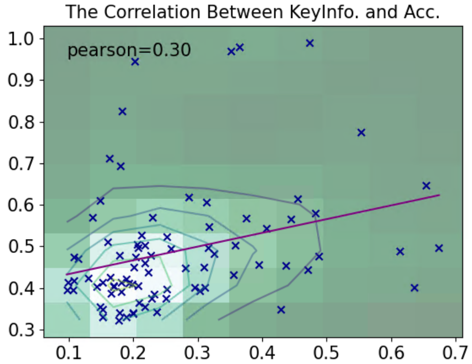


Figure 4: The Pearson Correlation between the Key Information metric and the Accuracy metric across all models and prompt methods.

Fig. 3. For benchmarks explicitly annotated with reasoning processes (such as the DetectBench), one can directly concat reasoning outputs for each stage as the finetune data. For benchmarks with only standard answers, the model can automatically complete the reasoning process based on the questions and answers and organize these reasoning contents as finetuning data. The advantage of this method lies in using the freely output of a LLM as finetune data in the first three stage, which significantly reducing the complexity of constructing datasets that include inferential processes.

5 Experiments

5.1 Overall Setup

Models: In our quest to leverage the most sophisticated models for enhanced replicability and robustness in results, we utilized a suite of eminent models from both the API-based and 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.

Metrics: The DetectBench, comprised of multiple-choice questions, employs Accuracy as the metric for evaluating the likelihood of model correctness in answer selection. Additionally, the benchmark assesses models' ability to identify crucial information from the context, a task akin to machine reading comprehension, using Accuracy for evaluation. However, given the challenge in direct content segment generation from “Context”, RougeL was utilized for evaluations where applicable.

5.2 Performance with Different Prompt Engineering

5.2.1 Experimental Setup

Baselines: A range of prompt engineering methods were analyzed for comparative insights. These include:

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)**, focusing on problem deconstruction before solution. **Self-Question Prompt** (Wang et al., 2023), introduced in this study. **Naive /w Key Info** and **Naive /w Answer**, enhancing inputs with “Key Information” and the “Answer” respectively.

Methods involving self-verification processes, such as Tree of Thought (Yao et al., 2023) and Graph of Thought (Besta et al., 2023), and those increasing correct answer probability through model error injections, like Reflexion (Shinn et al., 2023), were excluded due to incompatibility with the benchmark's question setup or potential bias in a four-choice format.

Demonstration: Demonstration incorporates correct answers in test data format and a small number of examples to improve output format comprehension and knowledge acquisition. The Naive Prompt method appends answers after training data examples, while Auto-CoT guides the LLM to generate reasoning processes aligned with the “Context”, “Question” and “Answer”.

5.2.2 Analysis

Tab. 4 displays the performance of all baseline models across different prompt methods.

Varied Prompt Engineering Method Efficacy: Data shows that proprietary models like GPT4, GPT3.5, and GLM4 excel beyond open-source models such as ChatGLM3 and Llama2. Significant accuracy gains were observed with GPT4 and GLM4 using prompt engineering, whereas methods

	GPT4		GPT35		GLM4		ChatGLM3-chat		ChatGLM3-base		Llama2-chat		Llama2-base	
	KeyInfo.	Acc.	KeyInfo.	Acc.	KeyInfo.	Acc.	KeyInfo.	Acc.	KeyInfo.	Acc.	KeyInfo.	Acc.	KeyInfo.	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
Self-Question Prompt	**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/ Key Info	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/ Key Info (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 engineering techniques is presented. Results in bold indicate the best results achieved without additional information.

	Detective		HotPotQA	Reclor
	KeyInfo.	Acc.	RougeL-F.	Acc.
Llama2-base				
Naive	10.8	47.5	30.6	36.7
SQ Prompt	20.7	40.5	32.1	37.5
SQ Prompt w/ MR Chat	23.6	45.1	33.6	35.2
SQ FT w/ Detective	**38.6**	**56.7**	**37.2**	**39.6**
SQ FT w/ Generated	32.4	44.6	32.8	33.5
Llama2-Chat				
Naive	10.8	47.5	36.3	38.8
SQ Prompt	25.2	52.4	39.7	42.6
SQ Prompt w/ MR Chat	22.7	50.1	37.1	40.5
SQ FT w/ Detective	**40.9**	**58.3**	**41.7**	**45.5**
SQ FT w/ Generated	34.6	50.5	38.6	37.1
ChatGLM3-Base				
Naive	9.7	39.6	26.8	30.1
SQ Prompt	22.3	43.8	25.4	31.9
SQ Prompt w/ MR Chat	23.6	45.3	26.0	32.4
SQ FT w/ Detective	**37.6**	**50.8**	**34.2**	**36.7**
SQ FT w/ Generated	35.4	43.6	30.9	32.9
ChatGLM3-Chat				
Naive	15.3	41.3	31.8	33.0
SQ Prompt	18.9	42.2	37.6	38.9
SQ Prompt w/ MR Chat	14.6	41.9	35.4	38.4
SQ FT w/ Detective	**27.1**	**56.3**	**42.3**	**41.7**
SQ FT w/ Generated	24.6	43.5	38.5	39.1

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

like Self-CoT saw a minor performance reduction in GPT3.5, ChatGLM3, and Llama2. This indicates that while advanced models benefit from prompt-guided reasoning, imposing such techniques on models with less sophisticated reasoning abilities may lead to performance decrements.

Key Information Detection Shortcomings: A general shortfall in key information detection was noted, especially with GPT4-Turbo's average accuracy standing at 40%. While accurate answers don't always require pinpointing key information, a direct correlation exists between identifying such information and answer accuracy. Directly presenting key information to models notably improved RougeL scores and answer accuracy, emphasizing the importance of precise key information identification.

Reduced Demonstration Effectiveness: The historical utility of demonstrations in enhancing model response parsing has diminished as models have grown adept at interpreting complex instructions. Integration of three-shot demonstrations resulted in unstable performance across various prompt methods and model types (Gu et al., 2023).

Self-Question Prompt Superiority: The Self-Question method, unique to this study, markedly improved key information detection and reasoning across models. This approach not only enhanced accuracy but also demonstrated a broader efficacy compared to other prompt engineering strategies, reinforcing its value in augmenting model understanding and reasoning capabilities.

5.3 Optimizing Model Capabilities through Fine-Tuning

5.3.1 Experimental Setup

Baselines: This investigation employed four prominent open-source models to explore fine-tuning's role in augmenting model capabilities. Our focus was on assessing the effectiveness of the Self-Question Prompt (SQ Prompt) applied directly, the SQ Prompt within multi-round dialogues (SQ Prompt w/ MR Chat), fine-tuning using DetectBench data (SQ FT w/ Detective), and generating fine-tuning data from the DetectBench's Context, Question, and Answer (SQ FT w/ Generated). The appendices provide in-depth prompt descriptions utilized in each method.

The experiments aimed to evaluate the impact of Self-Question Fine-tuning on improving models' key information detection and reasoning abilities. To this end, a subset of 398 training dataset samples was used for fine-tuning over 3 epochs with the AdamW optimizer, detailed in the Appendix A.

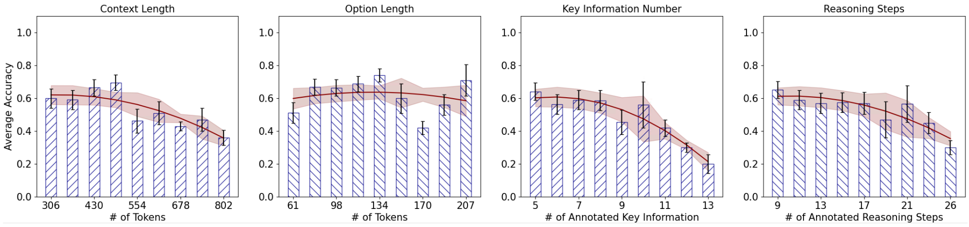


Figure 5: The performance of GPT4-Turbo is correlated with the context length, option length, the quantity of key information, and the number of reasoning steps involved.

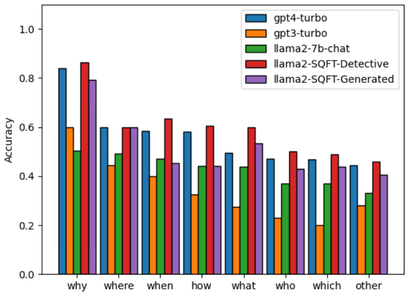


Figure 6: The performance of various models varies across different Question Types.

5.3.2 Insights and Evaluations

Effects of Transitioning to Multi-Round Dialogues: Our analysis revealed that transitioning from single-round to multi-round dialogues negatively influences chat model accuracy by 1.3%, while base models experience a 3.0% accuracy enhancement. This differential impact suggests that base models, due to their heightened sensitivity to context, benefit from multi-round dialogues as a mechanism to distill relevant information. On the contrary, chat models, which are adept at key information extraction, face performance setbacks when dialogue is fragmented into multiple segments.

Enhancements from DetectBench Data in Fine-Tuning: Utilizing DetectBench data for Self-Question Fine-tuning significantly boosts key information detection and reasoning skills in models. The observed post-fine-tuning improvements include a 15.2% increase in key information detection accuracy and a 10.5% uplift in overall model performance. These results underscore the DetectBench dataset's effectiveness in refining models' information processing and reasoning faculties.

5.4 In-depth Performance Analysis

5.4.1 Performance Influencing Factors

The analysis of GPT4-Turbo's performance, as detailed in Figure 5, highlights the effects of varying Context Length and Options Length on model accuracy. A notable decline in accuracy was observed as Context Length increased from 400 to 800 words, with accuracy dropping from approximately 65% to 35%. Additionally, the variability in Options Length indicated a struggle with reasoning complexity at both extremes of option length.

An examination of our annotations against model performance revealed a strong correlation between the volume of Key Information, reasoning depth, and performance metrics. Specifically, as the number of key information instances and reasoning depth escalated, a marked decrease in model accuracy was recorded, affirming the relationship between question complexity and model effectiveness.

5.4.2 Varied Responses to Different Question Types

The performance variation across different question types, as presented in Figure 6, shows models excelling in answering "Why" and "Where" questions, with the fine-tuned Llama-2 model achieving an impressive 90% accuracy. In contrast, the accuracy for "Who", "Which" and other question types hovered around 50%. This disparity suggests that while models effectively handle questions requiring an understanding of processes and environments, they struggle with questions that demand sophisticated entity recognition and relational discernment, pinpointing areas for future model enhancement.

6 Conclusion

In this paper, we introduce the DetectBench, which integrates information retrieval and reasoning, catering to the current demand for task-oriented complex information retrieval. This involves identifying key information from a plethora of data and conducting in-depth reasoning based on this key information to accomplish tasks. Additionally, we propose a novel type of prompt engineering and fine-tuning method termed the Self-Question Framework, designed to concurrently augment model performance in key information detection and commonsense reasoning.

7 Limitations

The DetectBench is conceptualized to facilitate the assessment of machine learning models' capabilities in simultaneously detecting information and engaging in commonsense reasoning. However, when juxtaposed with the complexity and breadth of information encountered in real-world scenarios, the data encompassed within detective reasoning puzzles appears markedly condensed.

The implementation of a Self-Question Prompt has demonstrated efficacy in enhancing the performance of models on the DetectBench. Nevertheless, this strategy is predominantly effective for tasks necessitating the extraction and inference of pivotal information from extensive datasets. Its efficacy diminishes substantially in scenarios where the information at hand is minimal and necessitates the incorporation of implicit knowledge derived from common sense or experiential understanding.

8 Ethical Concerns

Given that a benchmark concentrating on detective deduction puzzles is predisposed to encompass a multitude of sensitive subjects, including but not limited to homicides and thefts. If not meticulously moderated, there exists a risk that models might refuse responding to sensitive questions for security purposes, consequently disadvantaging models that prioritize higher security standards. Moreover, models that undergo fine-tuning using benchmark data may inadvertently amplify security vulnerabilities. Considerable effort and resources have been allocated towards mitigating the ethical dilemmas associated with the Detective Reasoning Benchmark, with the dual objectives of ensuring that models committed to security do not eschew responding to sensitive questions and that the utilization of our dataset does not compromise model security.

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.

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.

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.

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 Self-Question Finetune data.
2. Employing the “Context”, “Question”, and “Answer” in Detective Reasoning Benchmark to automatically generate Self-Question Finetune data.

The specific training parameters are detailed in Table 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 `Self-Question 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: ### Key Information: Please output what you consider to be the key information in the Context. Please note that the key information 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:
### Key Information: xxxxx
### Answer: 1/2/3/4
```

Table 10: Prompt of `Naive` method

```
# -*- coding: utf-8 -*-
Variables:
!<INPUT 0>! – Context
!<INPUT 1>! – Question
!<INPUT 2>! – Key Information !<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>!

### Key Information:
!<INPUT 2>!

### Option:
!<INPUT 3>!
Your output will contain the following:
### Key Information: Please output what you consider to be the key information in the Context. Please note that the key information 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:

### Key Information:
xxxxx

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

Table 11: Prompt of `Naive w/ Key Information` 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:
### Key Information: Please output what you consider to be the key information in the Context. Please note that the key information 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:

### Key Information: 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.
### Key Information: Please output what you think is the Key Information in the Context. Please note
that the Key Information 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

### Key Information:
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.
### Key Information: Please output what you think is the key information in the topic. Please note that the key information 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

### Key Information:
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.
### Key Information: Please output what you think is the Key Information in the Context. Please note
that the Key Information 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

### Key Information:
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: ### Key Information: Please output what you consider to be the
key information in the topic. Please note that the key information 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:

### Key Information:
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.
### Key Information: please output what you think is the key information in the Context. Please note
that the Key Information 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

### Key Information:
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.
### Key Information: Please output what you think is the key information in the Context. Please note
that the Key Information 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

### Key Information:
xxxxx

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

Table 18: Prompt of `Self-Question` method