On the Consistency of Commonsense in Large Language Models

Guozheng Li, Peng Wang*, **Wenjun Ke, Zijie Xu, Jiajun Liu, Ziyu Shang**

Southeast University

Data: https://github.com/1iguozheng/CoCo

Abstract

Commonsense, humans' implicit understanding of everyday situations, is crucial for large language models (LLMs). Existing commonsense evaluations for LLMs primarily focus on downstream knowledge tasks, failing to probe whether LLMs truly understand and utilize knowledge or merely memorize it. They also rely heavily on human annotation and lack automated large-scale data generation. To address this, we propose to automatically construct a large benchmark named **CoCo** (**Co**nsistency of **Co**mmonsense) comprising 39K samples derived from commonsense knowledge graphs (CSKGs), paired with symbolic questions and ground-truth answers, which systematically assesses LLMs' knowledge memorization, comprehension, and application and examines the consistency between these tasks. To enhance our evaluation, we also propose novel metrics and prompting strategies. Experimental results on multiple LLMs reveal that CoCo presents significant challenges, and our detailed analysis provides deeper insights into the strengths and limitations of LLMs' commonsense abilities.

1 Introduction

Commonsense refers to widely shared basic knowledge, which LLMs are believed to encode significantly during pre-training (Madaan et al., 2022; Jain et al., 2023; Zhao et al., 2023b). Previous commonsense evaluations (Zhou et al., 2020; Li et al., 2022; Cheng et al., 2024) only focus on commonsense assessment in LLMs using public benchmarks. While these evaluations rank overall performance, they **lack clear definitions and divisions of evaluated abilities**. Additionally, they inevitably **face data contamination and hallucination risks**—public benchmarks may leak into pre-training (Huang et al., 2023b), and correct responses might result from memorization rather than true understanding and reasoning (Ji et al., 2023; Huang et al., 2023a; Wang et al., 2024b).

Recent knowledge evaluations (Yu et al., 2024; Wang et al., 2024a; Fei et al., 2024; Sun et al., 2024) have begun refining ability definitions and addressing data contamination. For instance, KoLA (Yu et al., 2024) introduces a cognitive ability taxonomy and diverse data sources to mitigate contamination, while CHARM (Sun et al., 2024) examines the link between memorization and reasoning. Although knowledge memorization is separated from higher-level abilities, its vague definition and granularity make LLMs' reasoning errors hard to explain. Moreover, these methods heavily rely on human annotation and lack scalable dataset generation.

To this end, we propose to automatically generate large-scale evaluation datasets based on the structured knowledge in commonsense knowledge graphs (CSKGs) (Speer et al., 2017; Sap et al., 2019a; Hwang et al., 2021). Using CSKGs as an evaluation data source offers unique advantages. They support hierarchical tasks like knowledge retrieval and multi-hop reasoning, enabling different abilities assessment while reducing data leakage bias. CSKGs also facilitate automated multi-level data generation through logical queries and help track whether LLMs follow correct reasoning paths by comparing them with golden chains, aiding error analysis. We follow KoLA (Yu et al., 2024) to design our benchmark considering three key factors: **ability modeling**, **data** and **evaluation criteria**.

For ability modeling, we evaluate commonsense knowledge of LLMs and divide our commonsense evaluation task with three subtasks, **memorization**, **comprehension**, and **application**, as shown in Figure 1. Unlike previous benchmarks (Yu et al., 2024; Wang et al., 2024a; Fei et al., 2024) that rely on existing disparate datasets, we leverage consistency and establish an intrinsic connection between memorization and other tasks, similar to CHARM (Sun et al., 2024). However, CHARM focuses solely

*Corresponding author

Findings of the Association for Computational Linguistics: ACL 2025, pages 16205–16225
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

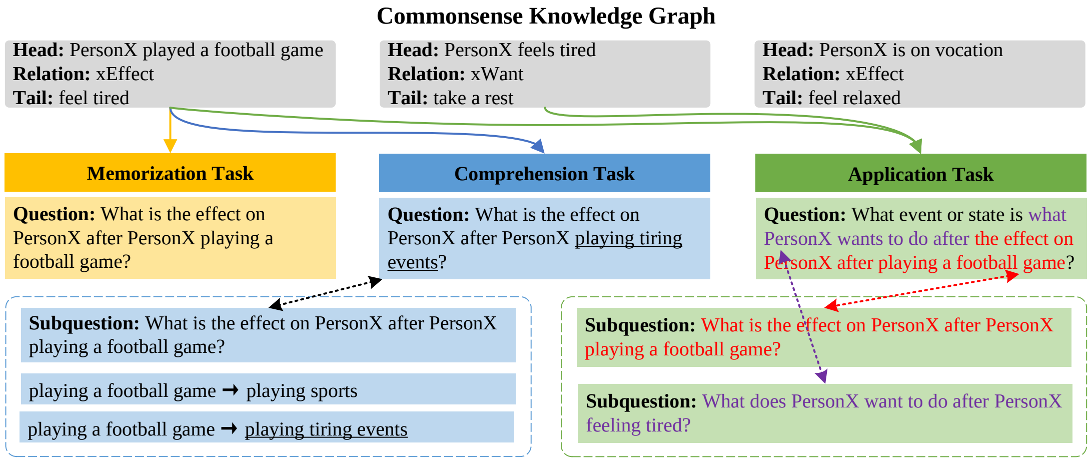

Figure 1: Examples of CoCo. CoCo consists interconnected memorization, comprehension and application tasks.

on the link between reasoning and memorization, while its manual annotation of required knowledge for reasoning questions is labor-intensive and often incomplete, as multiple solutions may exist, introducing biases in knowledge-reasoning correlation analysis. In contrast, we start by testing memorization with **atomic knowledge** from CSKGs, then evaluate comprehension and application, effectively reversing CHARM's process and providing the foundation for assessing higher-level abilities.

For data, for reducing manual annotation, we introduce the **CoCo** (**Co**nsistency of **Co**mmonsense) **dataset**. Its specificity is that commonsense questions posed in natural language are grounded in CSKGs. By sampling triples from CSKGs, our symbolic questions and answers are then verbalized to natural language (Shen et al., 2023; Fang et al., 2024). We compose more than 39K commonsense questions across three rungs, giving rise to scenarios which require different commonsense abilities. Moreover, instead of eliminating data contamination and hallucination, which is challenging or impossible, evaluating the consistency of commonsense in LLMs mitigates their effects by aligning memorized samples with internal knowledge.

For evaluation criteria, we design a **consistent evaluation system** with specialized metrics for the three tasks, guided by the principle of consistency. Traditional benchmarks report absolute metrics for each task separately, overlooking their interconnections and mutual influences (Yu et al., 2024). For example, using standard Accuracy to evaluate reasoning can be affected by data contamination (Ji et al., 2023) and knowledge gaps (Sun et al., 2024). And CHARM only shows the overall correlation between knowledge and reasoning results but lacks specific metrics to evaluate individual samples. We therefore propose new metrics to measure comprehension and application based on memorization.

We perform extensive experiments on seven LLMs and discover that CoCo is in general very challenging for LLMs. Exploiting CoCo, we also introduce a method to elicit consistent commonsense reasoning in LLMs. Specifically, we develop KnowCoT, a chain-of-thought prompting strategy (Wei et al., 2022) inspired by the knowledge storage and manipulation in LLMs (Allen-Zhu and Li, 2023), which prompts the LLM to recall relevant knowledge, and perform consistent commonsense reasoning. Our experiments indicate that KnowCoT substantially improves the consistency performance of LLMs especially GPT-4 (Achiam et al., 2023) on CoCo. We also analyze fine-grained errors to showcase the limitations of LLMs in commonsense knowledge and reasoning.

2 Preliminary

Commonsense Knowledge in CSKGs. Denote the commonsense knowledge triples in the CSKG as $\mathcal{K} = \{k = (h, r, t) \,|\, h \in \mathcal{H}, r \in \mathcal{R}, t \in \mathcal{T}\}$, where $\mathcal{H}$, $\mathcal{R}$, and $\mathcal{T}$ are the set of heads, relations, and tails in the CSKG. Each element $k \in \mathcal{K}$, e.g., (*PersonX is on vacation*, *xEffect*, *feel relaxed*), is a specific piece of knowledge, which can be expressed by various records, e.g., a text record "*PersonX is on vacation, as a result, PersonX will feel relaxed.*" We term such triple as a piece of atomic knowledge which is the foundation for abstract knowledge acquisition and multi-hop reasoning.

Knowledge Memorization. Given an LLM denoted as $\mathcal{M}$, we formulate that $\mathcal{M}$ memorize com-

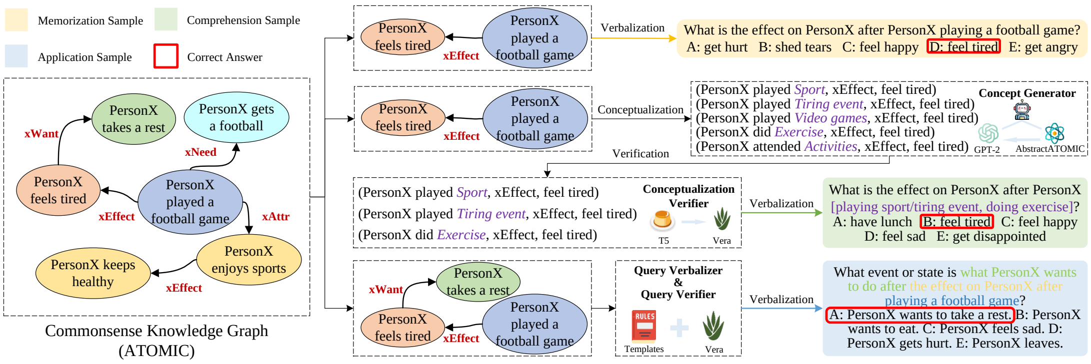

Figure 2: Overview of our dataset construction process.

monsense knowledge $k = (h, r, t)$ if $\mathcal{M}$ can correctly answer the corresponding question $q_{k\setminus t}$:

$$\mathcal{M}(q_{k\setminus t}) = t \quad (1)$$

where $t \in \mathcal{T}$, $q_{k\setminus t}$ is a record about knowledge k that lacks pivot information t. Taking Figure 1 as an example, $q_{k\setminus t}$ is "*What is the effect on PersonX after PersonX playing a football game?*". Then we drop $_{\setminus t}$ and use only q_k for simplicity. Formally, given an atomic knowledge triple k, an LLM $\mathcal{M}$ is expected to answer question q_k with $\mathcal{M}(q_k) = t$.

Knowledge Comprehension. The triples sampled from CSKGs can be used to directly evaluate knowledge memorization. However, rote memorization does not necessarily mean comprehension. In Figure 1, understanding that playing football leads to feeling tired involves recognizing it as a physically demanding activity. If LLMs truly comprehend, they should generalize the knowledge to infer concepts like "*Tiring events such as sports and exercise can make someone feel tired*". Thus the acquired abstract commonsense knowledge can be used to evaluate comprehension. Deriving such knowledge from CSKGs involves conceptualization (He et al., 2024). The objective of conceptualization is to form a conceptualized head event, denoted as h^c, from the original head h. This is achieved by linking a component $o \subseteq h$ to a concept c, forming h^c by replacing o with c. Thus abstract knowledge is formed by combining the conceptualized head event with the original relation and tail, represented by $k^c = (h^c, r, t)$. Formally, given an atomic knowledge triple k, its conceptualized triple is denoted as k^c. An LLM $\mathcal{M}$ is expected to answer question q_{k^c} with $\mathcal{M}(q_{k^c}) = t$. The prerequisite is the LLM memorizes k.

Knowledge Application. For application evaluation, LLMs are expected to answer commonsense reasoning questions, provided that they have mastered all the necessary atomic knowledge to answer this question. We therefore leverage the concept of logical queries (Hamilton et al., 2018) to acquire large-scale complex reasoning data from CSKGs which requires minimum human efforts (Fang et al., 2024). The query structures (*2i*, *2p*, *ip* and *pi*) that we study in this work are introduced in Appendix A.3. Figure 1 illustrates an example of *2p*. If LLMs memorize two atomic knowledge triples (*PersonX played a football game*, *xEffect*, *feel tired*) and (*PersonX feels tired*, *xWant*, *take a rest*), we expect LLMs to correctly answer the reasoning question constructed by the logical query "*What event or state is what PersonX wants to do after the effect on PersonX after playing a football game?*". Formally, given several atomic knowledge triples $k_1, ..., k_n$, an LLM $\mathcal{M}$ is expected to answer the question $q_{(k_1,...,k_n)}$ with $\mathcal{M}(q_{(k_1,...,k_n)}) = t$. The prerequisite is the LLM memorizes $k_1, ..., k_n$.

3 CoCo Benchmark

Task Formulation. We formulate the proposed task in the form of Multiple Choice Question Answering (MCQA). Our dataset $\mathcal{D} = \{\mathcal{Q}_i, \mathcal{A}_i\}_{i=1}^{N}$ consists of N pairs, each containing a question set $\mathcal{Q}_i$, and an answer set $\mathcal{A}_i$. Our main task is to test the accuracy of the prediction function $\mathcal{M} : \mathcal{Q} \mapsto \mathcal{A}$, i.e., an LLM which maps natural language questions to the corresponding answers:

$$\begin{aligned} &\mathcal{Q}_m = \{q_k\}, \mathcal{A}_m = \{a_k\} \\ &\mathcal{Q}_c = \{q_k, q_{k_1^c}, ..., q_{k_m^c}\}, \mathcal{A}_c = \{a_k, a_{k_1^c}, ..., a_{k_m^c}\} \\ &\mathcal{Q}_a = \{q_{(k_1,...,k_n)}, q_{k_1}, ..., q_{k_n}\}, \\ &\mathcal{A}_a = \{a_{(k_1,...,k_n)}, a_{k_1}, ..., a_{k_n}\} \end{aligned} \quad (2)$$

where $\mathcal{Q}_m$, $\mathcal{Q}_c$, $\mathcal{Q}_a$ and $\mathcal{A}_m$, $\mathcal{A}_c$, $\mathcal{A}_a$ are question sets and answer sets for memorization, comprehension and application evaluation, respectively. Note m represents the number of conceptualization operations, and n represents the number of atomic knowledge involved in a reasoning question. The correct answer corresponding to the multiple choice option $a_k \in \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}\}$ is t.

Construction Process. We use ATOMIC (Sap et al., 2019a) as the atomic knowledge source. The generation pipeline of CoCo is shown in Figure 2. Here we briefly introduce this construction process. Please refer to Appendix A for more details.

The **memorization** task involves sampling and representing the atomic knowledge triples from ATOMIC. This is achieved by selecting 2K diverse triples for each of the nine relations in ATOMIC. Diversity is ensured by embedding triples using Sentence-BERT (Reimers and Gurevych, 2019) and constructing a graph where nodes represent triples, and edges connect the most similar ones based on cosine similarity. A scoring mechanism prioritizes diversity by penalizing overly similar triples (Su et al., 2023). During verbalization, each triple is transformed into MCQA form with four distractors: two random ones from the CSKG and two adversarial ones sampled from related triples.

The **comprehension** task extends atomic knowledge by abstracting it into higher-level concepts through conceptualization. Using the same diversity mechanism as before, 20K head events are sampled. Conceptualizations for each head event are generated by a GPT-2 (Radford et al., 2019) model fine-tuned on ABSTRACTATOMIC (He et al., 2024), a corpus of abstract commonsense knowledge. Multiple candidates are filtered by Vera (Liu et al., 2023), a T5 (Raffel et al., 2020)-based plausibility scorer, which removes low-plausibility triples. Head events with at least three valid conceptualizations are retained, yielding 39K conceptualized triples for 13K head events. These triples are then verbalized into MCQA pairs.

The **application** task broadens reasoning by requiring inferences over multiple pieces of atomic knowledge. Logical queries are sampled from normalized ATOMIC (Shen et al., 2023), with tail entities adjusted for consistency. For each query type (i.e., *2i*, *2p*, *ip* and *pi*), 3K diverse instances are sampled, avoiding over representation of high-degree nodes. Distractors are carefully designed to challenge reasoning without ambiguity. Queries and answers are verbalized using templates (Fang et al., 2024), and refined with Vera (Liu et al., 2023) to remove flawed samples.

Task Type	Aspects	Question Type	# Sets	# Words / Set	# Questions / Set
Memorization	oEffect	Single Test	2,000	66.29	1.00
	oReact	Single Test	2,000		
	oWant	Single Test	2,000		
	xAttr	Single Test	2,000		
	xEffect	Single Test	2,000		
	xIntent	Single Test	2,000		
	xNeed	Single Test	2,000		
	xReact	Single Test	2,000		
	xWant	Single Test	2,000		
Comprehension	oEffect	1+3 Joint Test	727	268.68	4.00
	oReact	1+3 Joint Test	448		
	oWant	1+3 Joint Test	693		
	xAttr	1+3 Joint Test	1,898		
	xEffect	1+3 Joint Test	2,158		
	xIntent	1+3 Joint Test	1,299		
	xNeed	1+3 Joint Test	1,839		
	xReact	1+3 Joint Test	1,104		
	xWant	1+3 Joint Test	2,894		
Application	*2i*	2+1 Joint Test	2,490	258.05	3.42
	2p	2+1 Joint Test	2,125		
	ip	3+1 Joint Test	1,998		
	pi	3+1 Joint Test	1,329		

Table 1: Overview of CoCo. The sample (question set) numbers of memorization, comprehension and application tasks are 18,000, 13,060 and 7,942, respectively. Aspects include the relation types in ATOMIC and different query types. For comprehension, 1+3 Joint Test represents 1 memorization test and 3 conceptualization tests. For application, 2+1 (3+1) Joint Test represents 2 (3) memorization tests and 1 reasoning test.

Dataset Statistics. Our data generating procedure is able to algorithmically generate a vast number of questions. In practice, we pick a dataset size that is large enough to be representative, but not too large to be problematic given the expensive inference costs of LLMs. We set our dataset size to be 39K. The dataset roughly balances across the relation and query types, as shown in Table 1.

Quality Check. Our dataset is generated algorithmically, which has the following potential benefits: formal correctness, zero human annotation cost, and, most importantly, controllability (e.g., for the question distribution, as well as for making it more unlikely that the data was previously seen by LLMs). However, since our dataset is different from common NLP datasets collected from human natural language writing, we also need to perform additional data quality checks. We therefore checked for a list of natural language properties. For **grammatically**, we ran a grammatical error check using LanguageTool (Naber et al., 2003), and got on average 1.47 grammatical errors per 100 words (i.e., 98.53% correctness), showing most of the language in CoCo follows English grammar. For **human readability**, we checked how comprehensible the questions are to average persons. We selected 100 questions from CoCo, and let an undergraduate student annotator go through the questions to judge whether they could understand or not, where 94% of the questions were deemed readable. Lastly, we conducted a **sanity check** where one author of this paper tried to solve a random sample of 100 questions from the dataset, and we recorded an accuracy of 87% on this task.

4 Evaluation Methods

Evaluation Setup. We instruct LLMs to answer questions from three sets: $\mathcal{Q}_m$, $\mathcal{Q}_c$, and $\mathcal{Q}_a$, with their predicted answer sets defined as:

$$\begin{aligned} \mathcal{P}_m &= \{p_k\} \\ \mathcal{P}_c &= \{p_k, p_{k_1^c}, ..., p_{k_m^c}\} \\ \mathcal{P}_a &= \{p_{(k_1,...,k_n)}, p_{k_1}, ..., p_{k_n}\} \end{aligned} \tag{3}$$

where we set $m = 3$, $n \in \{2, 3\}$. We evaluate LLMs by comparing $\mathcal{P}_m$, $\mathcal{P}_c$, $\mathcal{P}_a$ and $\mathcal{A}_m$, $\mathcal{A}_c$, $\mathcal{A}_a$.

Memorization. Accuracy is used as the evaluation metric for the memorization task. Let $\mathcal{X} \subset \mathcal{D}$ be the memorization subset, and $\mathcal{M}$ be an LLM to be evaluated. Consider a response $\mathcal{P}_m = \mathcal{M}(\mathcal{Q}_m)$ for $(\mathcal{Q}_m, \mathcal{A}_m) \in \mathcal{X}$, the MEMSCORE of $\mathcal{M}$ is:

$$\text{MEMSCORE}(\mathcal{M}) = \frac{1}{|\mathcal{X}|} \sum_{(\mathcal{Q}_m,\mathcal{A}_m)\in\mathcal{X}} \mathbb{1}_{p_k=a_k} \tag{4}$$

where $|\mathcal{X}|$ is the number of samples in the dataset. MEMSCORE simply describes the capabilities of LLMs to memorize atomic knowledge.

Comprehension. Let $\mathcal{Y} \subset \mathcal{D}$ be the comprehension subset, and $\mathcal{M}$ be an LLM to be evaluated. Consider a response $\mathcal{P}_c = \mathcal{M}(\mathcal{Q}_c)$ for $(\mathcal{Q}_c, \mathcal{A}_c) \in \mathcal{Y}$, we define a new metric for comprehension evaluation. The key idea is that if an LLM comprehends a certain atomic knowledge, then it is likely to master the corresponding conceptualizations. The COMSCORE of $\mathcal{M}$ is:

$$\text{COMSCORE}(\mathcal{M}) = \frac{1}{\sum\limits_{(\mathcal{Q}_c,\mathcal{A}_c)\in\mathcal{Y}} \mathbb{1}_{p_k=a_k}} \sum_{(\mathcal{Q}_c,\mathcal{A}_c)\in\mathcal{Y}} \frac{\mathbb{1}_{p_k=a_k} \sum\limits_{*=1}^{m} \mathbb{1}_{p_{k_*^c}=a_{k_*^c}}}{m} \tag{5}$$

where $|\mathcal{Y}|$ is the dataset size, and m denotes conceptualization operations. The first term reflects the extent of atomic knowledge memorized by $\mathcal{M}$, while the second measures its mastery of related conceptualizations. COMSCORE evaluates LLMs' capabilities to omprehend abstract concepts.

Application. Let $\mathcal{Z} \subset \mathcal{D}$ be the application subset, and $\mathcal{M}$ be an LLM to be evaluated. Consider a response $\mathcal{P}_a = \mathcal{M}(\mathcal{Q}_a)$ for $(\mathcal{Q}_a, \mathcal{A}_a) \in \mathcal{Z}$, we consider two conditions: (1) the LLM answers the question correctly (i.e., $p_{(k_1,...,k_n)} = a_{(k_1,...,k_n)}$); (2) the LLM memorizes all the atomic knowledge (i.e., $p_{k_*} = a_{k_*}, \forall * \in [1, n]$). Generally, the overall reasoning performance of $\mathcal{M}$ is defined as follows:

$$\text{REASCORE}(\mathcal{M}) = \frac{1}{|\mathcal{Z}|} \sum_{(\mathcal{Q}_a,\mathcal{A}_a)\in\mathcal{Z}} \mathbb{1}_{p_{(k_1,\ldots,k_n)}=a_{(k_1,\ldots,k_n)}} \tag{6}$$

where $|\mathcal{Z}|$ is the dataset size. While REASCORE assesses overall performance, it cannot evaluate an LLM's ability to avoid hallucination or utilize knowledge. For the first case, a correct answer with partial atomic knowledge may result from data contamination or hallucination. Thus we define the FAISCORE to measure the faithfulness of $\mathcal{M}$:

$$\text{FAISCORE}(\mathcal{M}) = \frac{\sum_{(\mathcal{Q}_a,\mathcal{A}_a)\in\mathcal{Z}} \mathbb{1}_{p_{(k_1,\ldots,k_n)}=a_{(k_1,\ldots,k_n)}} \cdot \mathbb{1}_{p_{k_*}=a_{k_*},\forall*\in[1,n]}}{\sum_{(\mathcal{Q}_a,\mathcal{A}_a)\in\mathcal{Z}} \mathbb{1}_{p_{(k_1,\ldots,k_n)}=a_{(k_1,\ldots,k_n)}}} \tag{7}$$

where the denominator represents the number of questions correctly answered by $\mathcal{M}$, while the numerator counts those correctly answered whose required atomic knowledge are memorized. For the second case, we define another metric APPSCORE:

$$\text{APPSCORE}(\mathcal{M}) = \frac{\sum_{(\mathcal{Q}_a,\mathcal{A}_a)\in\mathcal{Z}} \mathbb{1}_{p_{k_*}=a_{k_*},\forall*\in[1,n]} \cdot \mathbb{1}_{p_{(k_1,\ldots,k_n)}=a_{(k_1,\ldots,k_n)}}}{\sum_{(\mathcal{Q}_a,\mathcal{A}_a)\in\mathcal{Z}} \mathbb{1}_{p_{k_*}=a_{k_*},\forall*\in[1,n]}} \tag{8}$$

where the denominator represents the number of samples with all atomic knowledge memorized by $\mathcal{M}$, while the numerator counts questions correctly answered by $\mathcal{M}$. APPSCORE reflects an LLM's ability to answer reasoning questions using all required atomic knowledge, with a higher score indicating stronger knowledge utilization.

5 KnowCoT Prompting

In order to guide LLMs in correctly answering the questions in CoCo and improve their consistency of commonsense knowledge, we develop KnowCoT, a multi-step chain-of-thought prompt in Figure 3.

Given a commonsense question q, we provide the LLM a list of instructions: $l = (s_1, s_2, s_3)$ consisting of the detailed descriptions of the three

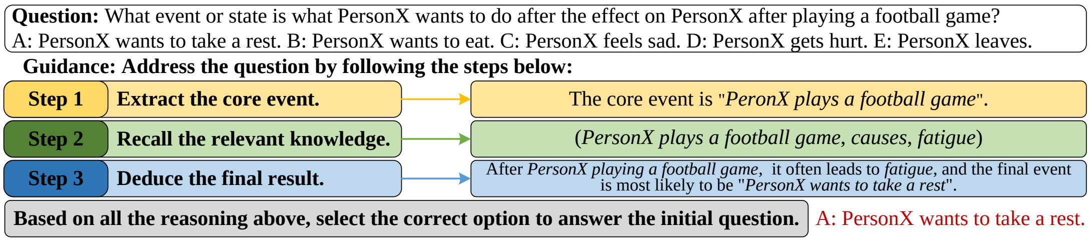

Figure 3: Illustration of KnowCoT. Compared with directly prompting the LLMs questions, we impose an *inductive bias* upon LLMs by explicitly recalling relevant knowledge, thus improving the comprehension and application.

steps. As the model $f_{\text{LLM}} : s_i \mapsto r_i$ produces responses r_1, r_2, r_3 sequentially corresponding to the three steps, we concatenate all the above before asking the final question "Based on all the reasoning above, select the correct option to answer the initial question." See the complete prompt in Appendix B.

6 Experiments

6.1 Experimental Setup

We use popular API-based and open-source LLMs as baselines, including Mistral (Jiang et al., 2023), Llama (Dubey et al., 2024), Qwen (Yang et al., 2024), GPT-3 (Brown et al., 2020) and GPT-4 (Achiam et al., 2023), with various parameter sizes. Besides the vanilla evaluation, we also evaluate LLMs using popular zero-shot CoT (Kojima et al., 2022) and our KnowCoT prompting strategies. Note that we do not conduct few-shot experiments due to the bias of sample selection on the final evaluation results. The complete list of model versions is shown in Table 2 and experimental details can be found in Appendix C.

Models	Is Open	Main Language	Size
Mistral-7B-Instruct-v0.3	✓	en	7B
Llama3-8B-Instruct	✓	en	8B
Qwen2.5-7B-Instruct	✓	zh	7B
Llama2-13B-Chat	✓	en	13B
Qwen2.5-14B-Instruct	✓	zh	14B
GPT-3.5-turbo	✗	en	> 175B
GPT-4o	✗	en	> 175B

Table 2: LLMs evaluated in our experiments.

6.2 Main Results

Table 3 presents the main results of LLMs on CoCo, where we have the following findings.

Overall, CoCo presents a significant challenge for all LLMs. GPT-4 achieves the highest performance across five dimensions. However, despite its advancements, a substantial performance gap of 17.7% still exists between the most capable LLM and human performance. Notably, the

Models	Methods	MEM.	COM.	REA.	FAI.	APP.	Average
Human	*Sampling Test*	90.53	94.43	87.80	90.25	93.46	91.29
Mistral-7B	*Vanilla*	60.88	78.67	52.35	42.94	68.18	60.60
	CoT	62.77	79.35	52.82	44.34	69.65	61.79
	KnowCoT	61.55	79.93	54.32	45.91	71.48	62.64
Llama3-8B	*Vanilla*	63.74	80.66	50.53	40.37	61.21	59.30
	CoT	64.26	80.89	50.88	42.29	62.89	60.24
	KnowCoT	63.88	81.26	52.87	43.81	64.59	61.28
Qwen2.5-7B	*Vanilla*	59.52	79.37	48.76	40.18	58.52	57.27
	CoT	58.35	80.24	48.95	41.16	58.97	57.53
	KnowCoT	60.87	80.90	51.37	43.06	60.94	59.43
Llama2-13B	*Vanilla*	66.48	82.74	56.16	44.00	68.26	63.53
	CoT	65.25	82.86	57.00	45.32	70.75	64.24
	KnowCoT	66.42	83.52	58.73	46.97	71.43	65.41
Qwen2.5-14B	*Vanilla*	67.83	81.22	56.99	44.10	69.36	63.90
	CoT	68.00	81.88	58.20	45.53	70.52	64.83
	KnowCoT	68.95	82.37	59.67	48.87	73.64	66.70
GPT-3.5-turbo	*Vanilla*	75.25	85.96	62.87	46.97	72.38	68.69
	CoT	77.62	85.20	65.78	49.52	74.96	70.62
	KnowCoT	75.78	86.25	**67.19**	**51.63**	**76.07**	71.38
GPT-4o	*Vanilla*	**79.81**	**87.37**	65.16	49.68	75.12	**71.43**
	CoT	**81.54**	**88.63**	**66.64**	**52.57**	**76.17**	**73.11**
	KnowCoT	**81.66**	**89.18**	**68.84**	**55.49**	**80.44**	**75.12**

Table 3: Main Results. Global top-3 results are **bold**.

gap widens to 38.5% on FAISCORE, indicating that LLMs struggle significantly with maintaining internal consistency. This suggests that while LLMs excel in reasoning tasks, they still face fundamental limitations in aligning their responses with coherent and logically consistent knowledge structures.

As model scale decreases, its knowledge reservoir shrinks, leading to gradual performance degradation. In memorization, GPT-4 falls behind human performance by only 9.8%, highlighting its extensive internal knowledge retention. This suggests that larger-scale models can store and retrieve commonsense knowledge more effectively. In contrast, Mistral-7B and Qwen2.5-7B exhibit the weakest performance in knowledge memorization, reflecting the limitations of smaller models in capturing and recalling vast amounts of knowledge.

LLMs that achieve good performance in memorization and comprehension may exhibit performance degradation in application. For instance, LLaMA outperforms Mistral by an absolute aver-

Models	MemScore									ComScore								
	oEffect	oReact	oWant	xAttr	xEffect	xIntent	xNeed	xReact	xWant	oEffect	oReact	oWant	xAttr	xEffect	xIntent	xNeed	xReact	xWant
Mistral-7B	52.1↓	65.7↑	51.9↓	57.4↓	55.9↓	78.6↑	66.8↑	64.5↑	55.2↓	72.7↓	81.0↑	70.5↓	84.4↑	75.9↓	87.2↑	76.8↓	83.9↑	74.2↓
Llama3-8B	55.9↓	60.8↓	57.1↓	63.4↓	58.8↓	81.8↑	70.0↑	66.3↑	59.7↓	72.2↓	79.5↓	76.1↓	85.3↑	77.5↓	87.9↑	80.1↓	85.2↑	77.4↓
Qwen2.5-7B	51.2↓	56.8↓	54.1↓	60.1↑	55.8↓	78.1↑	66.8↑	62.6↑	50.2↓	72.3↓	78.8↓	77.3↓	87.2↑	77.8↓	88.4↑	77.3↓	81.3↑	73.4↓
Llama2-13B	59.5↓	64.7↓	60.1↓	66.0↓	61.6↓	85.2↑	72.8↑	70.1↑	58.3↓	76.7↓	84.6↑	76.2↓	88.5↑	79.6↓	89.7↑	80.7↓	86.3↑	78.5↓
Qwen2.5-14B	60.8↓	66.7↓	60.3↓	68.4↑	61.9↓	87.3↑	74.4↑	70.2↑	60.5↓	75.2↓	82.6↑	76.3↓	86.7↑	78.0↓	89.6↑	76.3↓	86.5↑	79.9↓
GPT-3.5-turbo	65.4↓	79.5↑	66.2↓	72.4↓	69.8↓	89.2↑	82.0↑	79.7↑	73.2↓	79.2↓	89.3↑	80.6↓	90.6↑	82.2↓	94.7↑	85.4↓	89.1↑	80.7↓
GPT-4o	71.1↓	82.8↑	71.8↓	76.3↓	75.7↓	94.0↑	84.3↑	82.9↑	79.9↑	81.6↓	88.2↑	83.1↓	93.3↑	84.2↓	96.0↑	81.7↓	92.4↑	85.4↓

Table 4: Results of each relation. ↓ and ↑ represent the performance is lower or higher than its average performance.

Models	Methods	*2i*				*2p*				*ip*				*pi*			
		Rea.	Fai.	App.	Avg.	Rea.	Fai.	App.	Avg.	Rea.	Fai.	App.	Avg.	Rea.	Fai.	App.	Avg.
Mistral-7B	*Vanilla*	67.8↑	47.0↑	82.0↑	65.6↑	40.0↓	56.2↑	52.7↓	49.6↓	42.1↓	35.7↓	61.3↓	46.3↓	59.5↑	32.9↓	76.7↑	56.4↑
	CoT	66.8↑	48.9↑	83.6↑	66.4↑	41.6↓	56.3↑	53.0↓	50.3↓	42.1↓	37.1↓	63.8↓	47.7↓	60.7↑	35.1↓	78.2↑	58.0↑
	KnowCoT	69.7↑	49.5↑	84.5↑	67.9↑	42.4↓	59.9↑	54.8↓	52.4↓	43.4↓	38.1↓	65.9↓	49.1↓	61.8↑	36.1↓	80.7↑	59.5↑
Llama3-8B	*Vanilla*	62.0↑	47.5↑	73.8↑	61.1↑	39.6↓	51.7↑	47.9↓	46.4↓	46.5↓	33.6↓	56.9↓	45.7↓	54.1↑	28.7↓	66.2↑	49.7↓
	CoT	61.1↑	48.9↑	75.9↑	62.0↑	38.6↓	54.3↑	48.7↓	47.2↓	47.8↓	36.3↓	58.2↓	47.4↓	56.0↑	29.7↓	68.8↑	51.5↓
	KnowCoT	64.0↑	50.8↑	77.1↑	64.0↑	42.2↓	55.3↑	51.1↓	49.5↓	48.7↓	36.8↓	60.0↓	48.5↓	56.5↑	32.3↓	70.2↑	53.0↓
Qwen2.5-7B	*Vanilla*	62.0↑	48.0↑	69.4↑	59.8↑	36.7↓	53.6↑	43.0↓	44.4↓	44.2↓	31.9↓	55.5↓	43.9↓	52.3↑	27.2↓	66.2↑	48.6↓
	CoT	62.0↑	47.2↑	70.2↑	59.8↑	36.5↓	55.2↑	42.6↓	44.8↓	45.6↓	32.8↓	57.0↓	45.2↓	51.7↑	29.4↓	66.1↑	49.1↓
	KnowCoT	64.5↑	51.4↑	71.5↑	62.4↑	39.4↓	56.8↑	45.4↓	47.2↓	47.2↓	34.5↓	58.0↓	46.6↓	54.4↑	29.6↓	69.0↑	51.0↓
Llama2-13B	*Vanilla*	70.6↑	47.4↑	80.4↑	66.1↑	40.6↓	56.4↑	50.6↓	49.2↓	48.5↓	37.8↓	63.5↓	50.0↓	64.8↑	34.4↓	78.5↑	59.3↑
	CoT	71.6↑	48.2↑	82.6↑	67.5↑	40.8↓	57.1↑	53.6↓	50.5↓	49.9↓	39.7↓	65.9↓	51.8↓	65.7↑	36.2↓	80.9↑	61.0↑
	KnowCoT	73.9↑	49.6↑	83.3↑	68.9↑	43.4↓	59.7↑	54.4↓	52.5↓	50.7↓	40.2↓	66.9↓	52.6↓	66.9↑	38.3↓	81.1↑	62.1↑
Qwen2.5-14B	*Vanilla*	71.1↑	48.9↑	81.5↑	67.2↑	42.4↓	57.3↑	51.7↓	50.5↓	48.1↓	36.4↓	64.8↓	49.8↓	66.4↑	33.6↓	79.4↑	59.8↑
	CoT	72.0↑	51.4↑	81.8↑	68.4↑	44.3↓	59.7↑	54.0↓	52.7↓	49.7↓	36.8↓	66.6↓	51.0↓	66.8↑	34.1↓	79.6↑	60.1↑
	KnowCoT	73.1↑	54.1↑	85.6↑	70.9↑	44.9↓	62.6↑	55.9↓	54.5↓	50.8↓	41.2↓	69.3↓	53.8↓	69.9↑	37.7↓	83.7↑	63.8↑
GPT-3.5-turbo	*Vanilla*	79.4↑	50.0↑	87.1↑	72.2↑	43.1↓	59.9↑	53.5↓	52.2↓	55.3↓	41.3↓	68.9↓	55.2↓	73.7↑	36.7↓	80.1↑	63.5↑
	CoT	82.9↑	52.8↑	88.8↑	74.8↑	45.8↓	62.6↑	56.9↓	55.1↓	57.4↓	44.2↓	71.0↓	57.5↓	77.0↑	38.5↓	83.1↑	66.2↑
	KnowCoT	84.6↑	55.0↑	90.5↑	76.7↑	46.2↓	65.5↑	56.7↓	56.1↓	61.2↓	45.6↓	72.4↓	59.7↓	76.8↑	40.5↓	84.7↑	67.3↑
GPT-4o	*Vanilla*	81.7↑	54.7↑	90.9↑	75.8↑	45.5↓	63.3↑	53.9↓	54.2↓	57.9↓	43.3↓	72.0↓	57.7↓	75.5↑	37.5↓	83.6↑	65.5↑
	CoT	82.6↑	58.8↑	91.6↑	77.7↑	46.3↓	65.3↑	55.3↓	55.7↓	59.7↓	46.3↓	72.9↓	59.6↓	77.9↑	39.8↓	84.9↑	67.5↑
	KnowCoT	83.5↑	60.1↑	94.6↑	79.4↑	53.0↓	69.5↑	60.0↓	60.8↓	60.5↓	49.4↓	78.8↓	62.9↓	78.3↑	43.0↓	88.3↑	69.9↑

Table 5: Results of each query. ↓ and ↑ represent the performance is lower or higher than its average performance.

age of 1.92% in MemScore and ComScore, yet it experiences a significant decline of 6.87% in AppScore. This suggests that while certain LLMs excel at storing and retrieving knowledge, they may face challenges in applying that knowledge to reasoning-intensive tasks.

KnowCoT consistently improves the LLMs' performance especially on application. Our KnowCoT achieves the highest performance of 75.12%, which is substantially better than the vanilla GPT-4 by 3.69 points on average. And there is also an absolute gain of 14.81% on ReaScore, FaiScore and AppScore. The impact of CoT across tasks can be found in Appendix D.

LLM's faithfulness to knowledge is closely tied to its comprehension, whereas its overall reasoning ability is determined by both memorization and application. This conclusion is supported by the strong correlation between MemScore × ComScore and FaiScore, showing that knowledge faithfulness depends more on comprehension than memorization. Likewise, the correlation between MemScore × AppScore and ReaScore confirms that effective reasoning requires both memorization and application, not just knowledge recall. These findings validate the proposed metrics and demonstrate their effectiveness in distinguishing and characterizing different LLM capabilities. See the detailed results in Appendix E.

6.3 Challenges in Commonsense

LLMs excel or fall short in different aspects of commonsense knowledge. We analyze LLMs' performance across various commonsense aspects, as shown in Table 4. We regard the LLMs' average performance in memorization and comprehension as the baseline. If the LLM outperforms the baseline in a specific aspect, it suggests greater proficiency in this relation type of knowledge, and vice versa. The findings indicate LLMs generally demonstrate good knowledge of *xIntent* and *xReact*. However, their proficiency of *o/xEffect* and *o/xWant* is relatively weaker. The uneven mastery of knowledge significantly affects the LLMs's reasoning performance, especially when dealing with complex questions that involve multiple types of knowledge. Moreover, showing good memoriza-

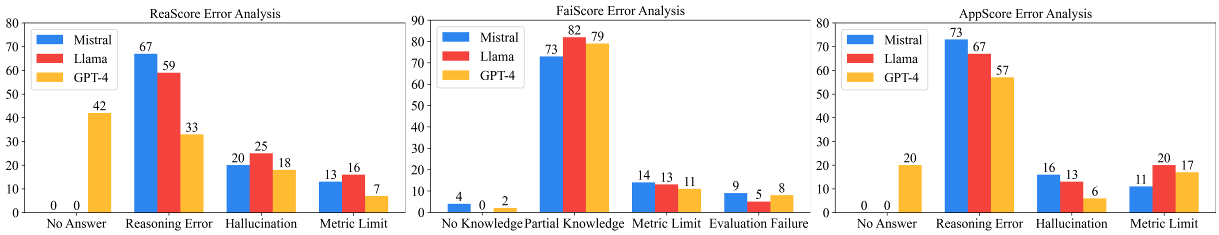

Figure 4: Error analysis for fine-grained commonsense reasoning results. We select 100 error cases from each subtask (i.e., REASCORE, FAISCORE and APPSCORE) for Mistral, Llama and GPT-4, respectively.

tion results in certain aspects does not necessarily mean good comprehension, and vice versa. For example, LLMs generally good at memorize *xNeed* knowledge but the comprehension is below average level, while *xAttr* knowledge is hard for LLMs to memorize but shows better comprehension results.

LLMs underperform in (multi-hop) commonsense reasoning. We analyze LLMs' reasoning ability across different query types, as shown in Table 5. The performance of all LLMs in commonsense reasoning is unsatisfactory. An intuitive conclusion is higher FAISCORE for query types involving less atomic knowledge (e.g., *2i* and *2p*). For different reasoning results, a noticeable decrease is observed in *2p*, *ip* and *pi* query types compared to *2i*. This is because these three tasks necessitate a two-step reasoning step. They contain multi-hop projection which involves inferring hidden reasoning contexts. In contrast, the *2i* task only requires intersection operations that can be completed with a single reasoning step. For intersection and projection results, LLMs are more struggle with the projection cases. The APPSCORE of *2p* and *ip* is much lower than others, because the corresponding query structure are overally a projection structure, while *2i* and *pi* require reasoning about complex intersections between event. In summary, LLMs struggle with commonsense reasoning, especially in multi-step reasoning scenarios.

6.4 Error Analysis

We manually analyze 100 error cases in REASCORE, FAISCORE and APPSCORE by Mistral, Llama and GPT-4, as shown in Figure 4.

REASCORE. We divide errors into: (a) *No Answer*: The model fails to provide a final answer. (b) *Reasoning Error*: The model encounters reasoning errors. (c) *Hallucination*: The model's prediction does not exist in the options. (d) *Metric Limit*: The model's prediction is correct, but the metric is limited by the evaluation criteria. We observe that GPT-4 have a higher *No Answer* rate, while Mistral and Llama are always able to provide answers. This discrepancy can be attributed to two factors: (1) the LLMs may lack the necessary commonsense knowledge to formulate an answer; (2) advanced LLMs abstain from answering questions beyond their knowledge scope, while weaker LLMs often attempt to answer, regardless of reliability.

FAISCORE. We divide errors into: (a) *No Knowledge*: The model answers the reasoning question correctly but has no atomic knowledge. (b) *Partial Knowledge*: The model answers the reasoning question correctly but has partial atomic knowledge. (c) *Metric Limit*: The model's prediction is correct, but the metric is limited by the evaluation criteria. (d) *Evaluation Failure*: The model answers the reasoning question correctly but does not use annotated atomic knowledge. The first two cases are due to the hallucination of LLMs. Moreover, there are very few cases of *No Knowledge* and in most cases LLMs have partial knowledge, which indicates that LLMs are relatively easy to obtain the final answer through partial knowledge. However, in some cases, evaluation by FAISCORE fails. We manually check these reasoning chains and find that LLMs can deduce the final answer using other atomic knowledge. Although we strictly construct reasoning questions through queries based on atomic knowledge, it is inevitable that other knowledge can also lead to the correct answer. But our error analysis also shows that this situation is rare and mastering the required knowledge is still necessary, demonstrating the rationality of FAISCORE.

APPSCORE. We divide errors into four groups as same as REASCORE. Compared to REASCORE, the *No Answer* and *Hallucination* rates are decreased, which is intuitive because the premise of all error cases is all atomic knowledge has been memorized. It can be observed that more error cases stem from reasoning errors and metric limitation. Despite the imperfect metric calculation, LLMs still have flaws in grounding atomic knowledge from reasoning questions and perform consistent commonsense reasoning.

7 Conclusion

We introduce CoCo, a large-scale benchmark for commonsense consistency, featuring automatic construction, novel evaluation metrics, and prompting methods. Extensive experiments on current LLMs assess their performance in memorization, comprehension, and application of commonsense knowledge. Our findings show a significant gap between LLMs and humans, and we provide a detailed analysis of the challenges LLMs face and potential improvement directions.

Limitations

CoCo not fully encompass dimensions such as temporal reasoning, causality, or broader contextual adaptability. The use of predefined templates for verbalizing queries and knowledge triples, while practical, might not fully represent the diversity of natural language expressions. The reliance on resources like ATOMIC provides a strong foundation but may not entirely reflect performance across more diverse or unseen commonsense domains. These observations highlight areas for further exploration, such as expanding task diversity, enhancing adaptability to real-world scenarios, and broadening the scope of knowledge.

Acknowledgement

This work was supported by National Science Foundation of China (Grant Nos.62376057), the Start-up Research Fund of Southeast University (RF1028623234) and SEU Innovation Capability Enhancement Plan for Doctoral Students (No. CXJH_SEU 25031).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Physics of language models: Part 3.2, knowledge manipulation. *arXiv preprint arXiv:2309.14402*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.

Michael Boratko, Xiang Li, Tim O'Gorman, Rajarshi Das, Dan Le, and Andrew McCallum. 2020. ProtoQA: A question answering dataset for prototypical common-sense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems*.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Qi Cheng, Michael Boratko, Pranay Kumar Yelugam, Tim O'Gorman, Nalini Singh, Andrew McCallum, and Xiang Li. 2024. Every answer matters: Evaluating commonsense with probabilistic measures. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. 2024. Navigate through enigmatic labyrinth a survey of chain of thought reasoning: Advances, frontiers and future. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Tianqing Fang, Zeming Chen, Yangqiu Song, and Antoine Bosselut. 2024. Complex reasoning over logical queries on commonsense knowledge graphs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Tianqing Fang, Weiqi Wang, Sehyun Choi, Shibo Hao, Hongming Zhang, Yangqiu Song, and Bin He. 2021a. Benchmarking commonsense knowledge base population with an effective evaluation dataset. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Tianqing Fang, Hongming Zhang, Weiqi Wang, Yangqiu Song, and Bin He. 2021b. Discos: Bridging the gap between discourse knowledge and commonsense knowledge. In *Proceedings of the 30th Web Conference*.

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Alan Huang, Songyang Zhang, Kai Chen, Zhixin Yin, Zongwen Shen, Jidong Ge, and Vincent Ng. 2024. LawBench: Benchmarking legal knowledge of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Will Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. 2018. Embedding logical queries on knowledge graphs. In *Proceedings of the 32rd Annual Conference on Neural Information Processing Systems*.

Mutian He, Tianqing Fang, Weiqi Wang, and Yangqiu Song. 2024. Acquiring and modeling abstract commonsense knowledge via conceptualization. *Artificial Intelligence*, 333:104149.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023a. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, et al. 2023b. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems*.

Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*.

Oana Ignat, Zhijing Jin, Artem Abzaliev, Laura Biester, Santiago Castro, Naihao Deng, Xinyi Gao, Aylin Gunal, Jacky He, Ashkan Kazemi, et al. 2023. A phd student's perspective on research in nlp in the era of very large language models. *arXiv preprint arXiv:2305.12544*.

Raghav Jain, Daivik Sojitra, Arkadeep Acharya, Sriparna Saha, Adam Jatowt, and Sandipan Dandapat. 2023. Do language models have a common sense regarding time? revisiting temporal commonsense reasoning in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Zhijing Jin, Abhinav Lalwani, Tejas Vaidhya, Xiaoyu Shen, Yiwen Ding, Zhiheng Lyu, Mrinmaya Sachan, Rada Mihalcea, and Bernhard Schoelkopf. 2022. Logical fallacy detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*.

Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. 2024. Gpt-4 passes the bar exam. *Philosophical Transactions of the Royal Society A*, 382(2270):20230254.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems*.

Guozheng Li, Peng Wang, and Wenjun Ke. 2023. Revisiting large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Xiang Lorraine Li, Adhiguna Kuncoro, Jordan Hoffmann, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. 2022. A systematic investigation of commonsense knowledge in large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Bill Yuchen Lin, Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Xiang Ren, and William W Cohen. 2021. Differentiable open-ended commonsense reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. Commongen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016.

How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Jiacheng Liu, Wenya Wang, Dianzhuo Wang, Noah Smith, Yejin Choi, and Hannaneh Hajishirzi. 2023. Vera: A general-purpose plausibility estimation model for commonsense statements. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. 2022. Language models of code are few-shot commonsense learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Swaroop Mishra, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Welleck, Chitta Baral, Tanmay Rajpurohit, Oyvind Tafjord, Ashish Sabharwal, Peter Clark, and Ashwin Kalyan. 2022. LILA: A unified benchmark for mathematical reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Daniel Naber et al. 2003. A rule-based style and grammar checker. *GRIN Verlag Munich, Germnay*.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Hongyu Ren, Weihua Hu, and Jure Leskovec. 2020. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. In *Proceedings of the 8th International Conference on Learning Representations*.

Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Joshua Robinson, Christopher Rytting, and David Wingate. 2023. Leveraging large language models for multiple choice question answering. In *Proceedings of the 11th International Conference on Learning Representations*.

Jürgen Rudolph, Samson Tan, and Shannon Tan. 2023. Chatgpt: Bullshit spewer or the end of traditional assessments in higher education? *Journal of applied learning and teaching*, 6(1):342–363.

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.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019a. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019b. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Xiangqing Shen, Siwei Wu, and Rui Xia. 2023. Dense-ATOMIC: Towards densely-connected ATOMIC with high knowledge coverage and massive multi-hop paths. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2023. Selective annotation makes language models better few-shot learners. In *Proceedings of the 11th International Conference on Learning Representations*.

Jiaxing Sun, Weiquan Huang, Jiang Wu, Chenya Gu, Wei Li, Songyang Zhang, Hang Yan, and Conghui He. 2024. Benchmarking Chinese commonsense reasoning of LLMs: From Chinese-specifics to reasoning-memorization correlations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Zhiyang Teng, Ruoxi Ning, Jian Liu, Qiji Zhou, Yue Zhang, et al. 2023. Glore: Evaluating logical reasoning of large language models. *arXiv preprint arXiv:2310.09107*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems*.

Bin Wang, Zhengyuan Liu, Xin Huang, Fangkai Jiao, Yang Ding, AiTi Aw, and Nancy Chen. 2024a. SeaEval for multilingual foundation models: From cross-lingual alignment to cultural reasoning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Mengru Wang, Yunzhi Yao, Ziwen Xu, Shuofei Qiao, Shumin Deng, Peng Wang, Xiang Chen, Jia-Chen Gu, Yong Jiang, Pengjun Xie, et al. 2024b. Knowledge mechanisms in large language models: A survey and perspective. *arXiv preprint arXiv:2407.15017*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. arxiv. *arXiv preprint arXiv:1910.03771*.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, et al. 2024. Kola: Carefully benchmarking world knowledge of large language models. In *Proceedings of the 12th International Conference on Learning Representations*.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. Reclor: A reading comprehension dataset requiring logical reasoning. In *Proceedings of the 8th International Conference on Learning Representations*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Zhuosheng Zhang, Yao Yao, Aston Zhang, Xiangru Tang, Xinbei Ma, Zhiwei He, Yiming Wang, Mark Gerstein, Rui Wang, Gongshen Liu, et al. 2023. Igniting language intelligence: The hitchhiker's guide from chain-of-thought reasoning to language agents. *arXiv preprint arXiv:2311.11797*.

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

Zirui Zhao, Wee Sun Lee, and David Hsu. 2023b. Large language models as commonsense knowledge for large-scale task planning. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems*.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020. Evaluating commonsense in pre-trained language models. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2024. Can large language models transform computational social science? *Computational Linguistics*, 50(1):237–291.

A Dataset Construction

A.1 Construction of Memorization Task

Atomic Knowledge Sampling. The first step of our data generating process is to sample a set of atomic knowledge triples from the CSKG. Because ATOMIC contains 877K textual descriptions of triples, triple sampling is required for the *memorization* task data construction. For 9 relations in ATOMIC, we sample 2K diverse and representative (Su et al., 2023) triples for each relation. We first compute a vector representation for each triple using Sentence-BERT (Reimers and Gurevych, 2019) by averaging the resulting vectors over the text input words. We then use the embedding vectors to create a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the vertices $\mathcal{V}$ are the embedded triples as defined above. For each vertex $v \in \mathcal{V}$, we create an edge to its k nearest vertices in terms of the cosine similarity between the embeddings. Let $\mathcal{D}$ and $\mathcal{U}$ denote the sets of already chosen and remaining samples, respectively. Initially, $\mathcal{D} = \emptyset$. Every vertex $u \in \mathcal{U}$ is scored by a modified degree:

$$\begin{aligned} \text{score}(u) &= \sum_{v \in \{v | (v,u) \in \mathcal{E}, v \in \mathcal{U}\}} s(v), \\ \text{where } s(v) &= \rho^{-|\{c \in \mathcal{D} | (v,c) \in \mathcal{E}\}|}, \quad \rho > 1 \end{aligned} \tag{9}$$

where s discounts v that is close to the already selected vertices, thereby encouraging diversity. We take $\arg\max_{u \in \mathcal{U}} \text{score}(u)$ and move it from $\mathcal{U}$ to $\mathcal{D}$ in every iteration. We set k to 150, ρ to 10, and run 2K of these iterations for each relation, where the current $\mathcal{D}$ has 18K triples.

Verbalization. After obtaining the representative triple set $\mathcal{D}$ of the entire CSKG, the commonsense relation within each triple is verbalized into human-readable text (Fang et al., 2021b), as shown in Table 6. For each question q_k verbalized by $k = (h, r, t) \in \mathcal{D}$, we sample 4 additional distractors for the answer t, where 2 of them are randomly sampled across the whole CSKG, and others are sampled from the neighbors of k but not the answers, represented as adversarial negative samples.

A.2 Construction of Comprehension Task

We construct the *comprehension* task based on the conceptualization (He et al., 2024). Instead of human annotation using Probase (Wu et al., 2012), conceptualization is achieved by instructing language models to generate knowledge based on concrete triples while carefully considering the original context throughout the process, where low-quality generations are eliminated by filtering models.

Relation	Human Readable Text
oEffect	What is the effect on PersonY after
oReact	What does PersonY feel after
oWant	What does PersonY want to do after
xAttr	What is PersonX seen as given
xEffect	What is the effect on PersonX after
xIntent	What is the intention of PersonX before
xNeed	What does PersonX need to do before
xReact	What does PersonX feel after
xWant	What does PersonX want to do after

Table 6: Textual prompt for commonsense relations. Commonsense triple (h, r, t) is translated to human language "[prompt] h", and the answer is t.

Concept Generation. Due to the knowledge abstraction process only involves head events, we compute a vector representation for each head event and then sample 20K head events in total from ATOMIC via Equation 9. For each head event, we sample a concrete triple and utilize language models to collect conceptualizations in a one-step inference manner. Specifically, we train a GPT-2 (Radford et al., 2019) based concept generator using ABSTRACTATOMIC (He et al., 2024) as abstract knowledge corpus. We generate possible concepts for the candidate in a way similar to COMET (Bosselut et al., 2019). Each sample (h_i, h_i^c) in ABSTRACTATOMIC is formed as a sequence of tokens $t_i = [h_i; \text{[EOS]}; h_i^c]$, with ; indicated the concatenation operation. The standard causal language model loss on h_i^c is used. Suppose h_i plus [EOS] correspond to first m tokens in t_i with total n tokens, the loss is:

$$L = -\sum_{t_i} \sum_{j=m+1} \log P(t_{i,j} | t_{i,<j}) \tag{10}$$

Then we utilize this fine-tuned model to sample five candidate conceptualizations for each event.

Conceptualization Verification Finally, we feed the possible event conceptualizations into a neural model as a gatekeeper to filter out those not matching the context. For all conceptualizations generated, we use an existing plausibility scorer Vera (Liu et al., 2023), a T5 (Raffel et al., 2020) based scorer, to score every triple in terms of plausibility of commonsense (between 0 and 1). We filter out triples with a plausibility score less than 0.5. For all remaining triples, we retain head events with more than 3 conceptualizations, resulting in

Relation	Mapping Rules
oEffect	Add PersonY in front of the tail
oReact	Add PersonY and "is" in front of the tail
oWant	Add PersonY in front of the tail and remove the initial "to"
xAttr	Add PersonX and "is" in front of the tail
xEffect	Add PersonX in front of the tail
xIntent	Add PersonX in front of the tail and remove the initial "to"
xNeed	Add PersonX in front of the tail and remove the initial "to"
xReact	Add PersonX and "is" in front of the tail
xWant	Add PersonX in front of the tail and remove the initial "to"

Table 7: Normalization rules for ATOMIC tail events.

13K original triples with their 39K conceptualized triples after random sampling. In other words, we provide 3 sets of corresponding abstract knowledge triples for each atomic knowledge triple.

Verbalization. We add 13K samples in *comprehension* task to $\mathcal{D}$, and now $\mathcal{D}$ has 31K samples. Similar to *memorization* task, each original triple and conceptualized triple is verbalized into a question answering pair. For questions q_k and q_{k^c}, we sample 4 additional distractors for each question to construct MCQA samples, respectively.

A.3 Construction of Application Task

We construct the *application* task based on multiple pieces of atomic knowledge, involving reasoning on unobserved edges and multiple events in CSKGs. Following previous works (Ren et al., 2020; Fang et al., 2024), we use basic projections *2p*, intersections *2i* and complex queries *ip* and *pi* as evaluation queries. In this formulation, multi-hop projection involves inferring hidden reasoning contexts, while intersection operations require reasoning about complex interactions between events.

Query Structures. The specific query structures that we study in this work are visualized in Figure 5. For *1p*, it can be simply instantiated by an atomic knowledge triple, such as $q[t] = t : \text{xEffect}(\text{PersonX is on vacation}, t)$. Following previous works (Ren et al., 2020; Fang et al., 2024), we use basic projections *2p*, intersections *2i* and complex queries *ip* and *pi*. The logical expressions for these four queries are as follows:

$$
\begin{aligned}
2i &: q[t] = t : r_1(h_1, t) \wedge r_2(h_2, t) \\
2p &: q[t] = t : r_1(h_1, V) \wedge r_2(V, t) \\
ip &: q[t] = t : r_1(h_1, V) \wedge r_2(h_2, V) \wedge r_3(V, t) \\
pi &: q[t] = t : r_1(h_1, V) \wedge r_2(h_2, t) \wedge r_3(V, t)
\end{aligned} \tag{11}
$$

where V denotes the free variable.

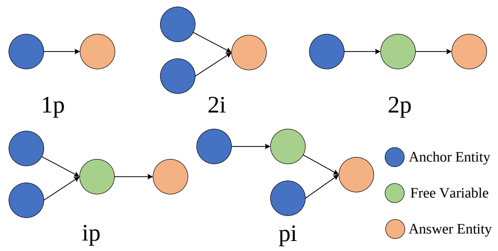

Figure 5: Visualization of query structures. The anchor entities and relations are specified to instantiate the query. "p" and "i" represent *projection* and *intersection*, and the number ahead of p and i indicates the number of anchor entities and free variables.

Relation	Prompt Template
oEffect	the effect on PersonY after
oReact	what PersonY feels after
oWant	what PersonY wants to do after
xAttr	what PersonX is seen as given
xEffect	the effect on PersonX after
xIntent	the intention of PersonX before
xNeed	what PersonX needed to do before
xReact	what PersonX feels after
xWant	what PersonX wants to do after

Table 8: Templates for verbalizing relations in queries.

Nodes Normalization. Before sampling queries, we first normalize the tail entities with simple rules following previous works (Fang et al., 2021a; Shen et al., 2023). In ATOMIC, heads are pre-defined complete sentences (e.g., "*PersonX says sorry*") while tails are usually short phrases without a subject (e.g.,"*to say sorry*"). We develop simple rules to add "*PersonX*" or "*PersonY*" in front of the tails to make them a complete sentence, as shown in Table 7. This allows the head and tail nodes to merge, enabling the query sampling from ATOMIC.

Query Sampling. Given a query structure, we use pre-order traversal to sample free variables and anchor events starting from an answer event. We sample predecessors uniformly based on (relation, event) pairs. For 4 query types, we sample 3K

Query Type	Question Template
2i	What event or state is both Prompt (r1) [V1] and also Prompt (r2) [V2]?
2p	What event or state is Prompt (r1) Prompt (r2) [V1]
ip	What event or state is Prompt (r3) both Prompt (r1) [V1], and also Prompt (r2) [V2]?
pi	What event or state is both Prompt (r1) Prompt (r3) [V3], and also Prompt (r2) [V2]?

Table 9: Templates for verbalizing four query types.

Q: `[question from the dataset]`

Guidance: Address the question by following the steps below:

Step 1) Extract the core event: Identify the core event in the question. The event should simply consists of its subject, predicate, and object.

Step 2) Recall the relevant knowledge: Recall the relevant knowledge triple of the core event implied by the question. The knowledge triple should simply consists of its head event, relation and tail event.

Step 3) Deduce the final result: Given all the information above, deduce the final result and answer step by step.

A: `[LLM previous response]`

Q: Based on all the reasoning above, select the correct option to answer the initial question.

A: `[LLM final answer]`

Figure 6: Details of our KnowCoT prompting strategy.

instances for each type. During sampling, to avoid over-sampling on nodes with high degree, we only sample from top 10 neighbors of a node scored by Equation 9. Besides, 4 additional distractors for each query are also sampled. We also conduct a post-order traversal starting from the anchor events to find all the answers of the query, ensuring that the sampled distractor is not the correct answer.

Verbalization. The sampled logical queries and answers are verbalized into human-readable text using a rule-based verbalizer (Fang et al., 2024). We use conversion rules and pre-defined templates to compose questions based on the relations in the queries. Based on the definition of each commonsense relation (Sap et al., 2019a; Hwang et al., 2021), we use the templates in Table 8 to verbalize each relation. In terms of logical queries, we use the conversion rules in Table 9 to convert the query to a question. Then we use Vera (Liu et al., 2023) to filter out low quality queries and answers. Specifically, we ask Vera for the correctness of the 5 options in MCQA. If all options are incorrect or multiple answers are correct, then this sample will be filtered out. Finally, around 8K samples are retained in *application* task, and $\mathcal{D}$ is expanded to 39K samples in total. Besides the reasoning question $q_{(k_1,\ldots,k_n)}$, all the atomic knowledge it contains is also verbalized into MCQA pairs.

B KnowCoT Prompting

B.1 Prompt Details

Compared with the standard strategy of directly prompting the LLMs a question, we impose an *inductive bias* upon LLMs by explicitly recalling atomic knowledge, thus improving the knowledge comprehension and application. In this way, we enhance the consistent commonsense reasoning abilities of LLMs via recalling relevant knowledge. The complete prompt is shown in Figure 6. In the end, we extract and obtain the answer $a \in \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}\}$ as the final prediction.

B.2 Prompt Examples

In order to verify the rationality of KnowCoT prompting, which guides the LLMs to extract core events, recall relevant important knowledge, and ultimately solve reasoning problems, we illustrate the KnowCoT prompting results of Mistral-7B-Instruct-v0.3 and GPT-4o as examples, as shown in Figure 7 and Figure 8. The reasoning question is the same as the question in Figure 3. It can be observed that although Mistral is difficult to follow instructions to generate knowledge in triple form, it describes the same meaning in natural language, Step 1) Extract the core event: The core event in the question is PersonX plays a football game.

Step 2) Recall the relevant knowledge: After playing a sport like football, the body can become tired and in need of rest to recover from the exertion.

Step 3) Deduce the final result: To recover from the physical fatigue, PersonX wants to take a rest.

Final answer: PersonX wants to take a rest.

Figure 7: An example of KnowCoT prompting response in Mistral-7B.

Step 1) Extract the core event:

The core event is **PersonX playing a football game**.

Step 2) Recall the relevant knowledge:

- **Playing a football game** is a physically demanding activity, so it often leads to fatigue. **Knowledge triple**: *(Playing a football game, causes, fatigue).*
- After physical exertion, common responses include resting or eating to recover energy. **Knowledge triple**: *(Fatigue, motivates, taking a rest or eating).*
- Emotional outcomes like sadness or injury depend on context (e.g., losing the game or an accident). However, they are not universal.

Step 3) Deduce the final result:

- **Option A: "PersonX wants to take a rest"** is highly plausible, as playing a football game often causes fatigue.
- **Option B: "PersonX wants to eat"** is also plausible, but secondary, as eating might follow after resting or only occur if hunger is specifically felt.
- **Option C: "PersonX feels sad"** is unlikely unless there is additional context indicating a disappointing result of the game.
- **Option D: "PersonX gets hurt"** is possible, but injury is situational and not implied by the question.
- **Option E: "PersonX leaves"** is neutral and does not address the core cause-and-effect relationship.

Final Answer: A: PersonX wants to take a rest.

Figure 8: An example of KnowCoT prompting response in GPT-4o.

while the powerful GPT-4o can generate triples that express the same meaning with atomic knowledge.

C Experimental Details

Following the standard practice of prompting LLMs for QA (Robinson et al., 2023), we initialize a prompt that takes "[Instruction] [Question] [Options]" as the input and ask the model to output the correct option in the QA pair as the prediction. The input prompt template is shown in Figure 9.

We access proprietary models through official API. For open-source models, we use the checkpoints from Huggingface's Transformer library (Wolf et al., 2019). We set the temperature to zero for greedy decoding in all experiments. Zero-shot CoT (Kojima et al., 2022) simply concatenates the input prompt with "Let's think step by step." After obtaining the reasoning intermediate results, we use "Therefore, among A through E, the answer is" for MCQA. And KnowCoT follows the similar idea as shown in Figure 6. For the correct or incorrect judgment of the final output answer of LLMs, we match the first option that appears in the final output answer of the LLMs as the predicted option.

D Chain-of-Thought Results

D.1 CoT Analysis

The Chain-of-Thought (CoT) prompting method demonstrates slight improvements over the vanilla baseline across all tasks. For memorization, CoT achieves a small gain (68.26 vs. 67.64), suggesting that CoT provides marginal benefits in this task but does not significantly enhance the model's ability to recall atomic knowledge. For comprehension, CoT scores 82.72, slightly higher than the vanilla score of 82.28, indicating its limited contribution to improving reasoning over abstract concepts. For application, CoT achieves 57.38 compared to the vanilla score of 55.91, showing a more noticeable improvement in tasks requiring multi-step reasoning, though the gap remains modest. These results suggest that while CoT reasoning aids in structuring the reasoning process, its influence on memorization and fundamental comprehension remains minimal. However, its impact becomes more pronounced in complex reasoning tasks, particularly those requiring higher-order thinking, such as application and problem-solving.

D.2 KnowCoT Analysis

The KnowCoT prompting method outperforms both the vanilla and CoT approaches across all tasks. For memorization, KnowCoT achieves the highest score (68.44), slightly better than CoT (68.26) and vanilla (67.64), indicating an enhanced ability to retrieve and represent atomic knowledge. For comprehension, KnowCoT reaches 83.34, showing incremental improvements over CoT (82.72) and vanilla (82.28), demonstrating its effectiveness in reasoning over abstract conceptual knowledge. For application, KnowCoT achieves a noticeable improvement (59.4) compared to CoT (57.38) and vanilla (55.91), highlighting its superior performance in handling complex reason-

Answer this commonsense reasoning question, where you are supposed to handle a multiple-choice question answering task to select the correct answer. Select one correct answer from A to E.

Question: [Question]

A: [Option A]. B: [Option B]. C: [Option C]. D: [Option D]. E: [Option E].

Answer:

Figure 9: The input prompt template for multiple-choice question answering.

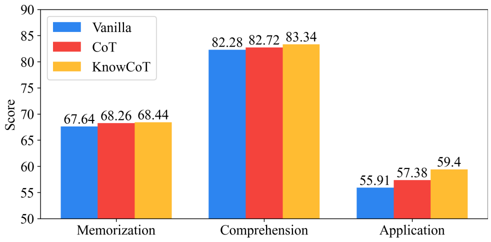

Figure 10: Performance gap with and without CoT and KnowCoT prompting. The results are averaged from all LLMs evaluated in Table 2.

ing tasks involving multiple pieces of knowledge. Overall, KnowCoT consistently outperforms CoT, especially in tasks requiring multi-step reasoning (application), suggesting its potential as a more robust prompting method for leveraging LLMs' commonsense reasoning capabilities.

E Correlation of Metrics

To provide a deeper understanding of our metrics, we evaluate the correlation between these metrics, as shown in Figure 11. First, MemScore is not strictly linearly correlated with ComScore or AppScore, indicating that memorization alone does not directly translate to other abilities. However, the product of MemScore and ComScore exhibits an almost perfect linear correlation with FaiScore, suggesting that LLMs' faithfulness to knowledge is closely linked to comprehension. Similarly, the product of MemScore and AppScore shows a near-perfect linear correlation with ReaScore, implying that knowledge memorization and application together determine overall reasoning performance. These findings validate our metrics and highlight their effectiveness in **distinguishing and characterizing different abilities**.

F Related Work

F.1 Large Language Models

In recent years, there has been rapid progress in the research of large language models (LLMs) (Zhao et al., 2023a). They exhibit outstanding performance across a multitude of tasks without the need for fine-tuning (Brown et al., 2020; Wei et al., 2022; Kojima et al., 2022). Furthermore, they have achieved astonishing results in complex reasoning tasks, such as mathematical reasoning (Cobbe et al., 2021; Mishra et al., 2022) and logical reasoning (Yu et al., 2020; Teng et al., 2023). Moreover, some studies suggest that the chain-of-thought prompting (Wei et al., 2022) can further enhance the model's capabilities in complex reasoning scenarios (Zhang et al., 2023; Chu et al., 2024). While LLMs are believe to encode various knowledge during pre-training (Madaan et al., 2022; Jain et al., 2023; Zhao et al., 2023b), existing commonsense evaluation works (Zhou et al., 2020; Li et al., 2022; Cheng et al., 2024) focus on commonsense knowledge assessment in LLMs.

F.2 Commonsense Benchmarks

Commonsense knowledge spans many categories, such as physical commonsense (e.g., a car is heavier than an apple), social commonsense (e.g., a person will feel happy after receiving gifts), and temporal commonsense (e.g., cooking an egg takes less time than baking a cake). Given this diverse nature of commonsense knowledge, various benchmarks have been proposed to test these different types of knowledge. Commonsense benchmarks broadly consist of two tasks: (a) multiple-choice evaluation (Zellers et al., 2019; Sakaguchi et al., 2021; Sap et al., 2019b; Bisk et al., 2020), where a model needs to choose the correct answer from a list of plausible answers; (b) generative evaluation (Boratko et al., 2020; Lin et al., 2020, 2021), which requires a model to generate an answer given a question and some additional context. In this study, we focus on multiple-choice benchmarks, since they provide a more reliable automatic metric (i.e., accuracy), whereas automated metrics used to evaluate language generation (e.g., BLEU (Papineni et al., 2002)) do not correlate perfectly with

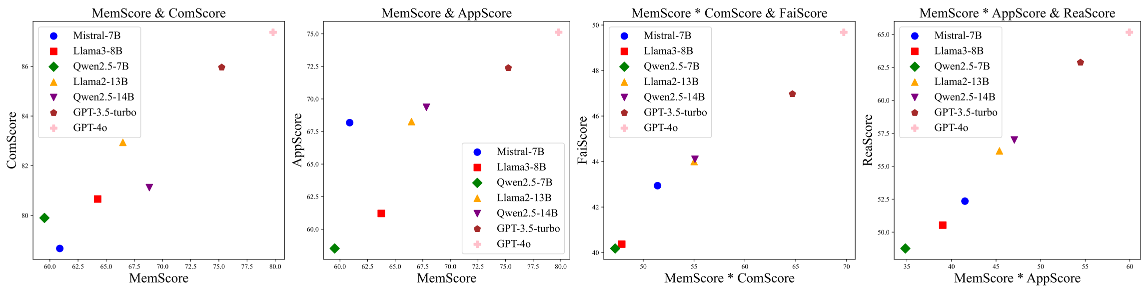

Figure 11: Correlations of our metrics.

human judgment (Liu et al., 2016; Novikova et al., 2017). However, unlike the previous works (Zhou et al., 2020; Li et al., 2022) that used existing commonsense benchmarks to directly evaluate LLMs, we reduce the impact of data contamination and hallucination by assessing consistency between commonsense knowledge and reasoning.

F.3 Evaluation for LLMs

Our work may be seen as part of the literature aimed at evaluating the performance of current LLMs (Brown et al., 2020; Jiang et al., 2023; Achiam et al., 2023; Yang et al., 2023; Riviere et al., 2024; Dubey et al., 2024; Yang et al., 2024), focusing on understanding their strengths and weaknesses. Various studies into the capabilities of LLMs (Bubeck et al., 2023; Ignat et al., 2023; Qin et al., 2023; Li et al., 2023) change people's perception of domains such as education (Rudolph et al., 2023), medicine (Singhal et al., 2023), law (Katz et al., 2024), and computational social science (Ziems et al., 2024). However, most work evaluates new models on existing datasets from previously curated large-scale benchmarks (Wang et al., 2019; Srivastava et al., 2022; Wang et al., 2022), or human exams (Jin et al., 2022; Katz et al., 2024) which is becoming increasingly unreliable due to data contamination. Our work alleviates the impact of data contamination and hallucination by evaluating LLM's knowledge memorization, comprehension, and application capabilities from a new perspective of consistency.

F.4 Correlations of Memorization and Others

There are benchmarks which assess both the knowledge memorization and reasoning capabilities of the LLMs within specific domains. For instance, KoLA (Yu et al., 2024) with its focus on world knowledge, includes tasks related to knowledge memorization, understanding, applying and creating. SeaEval (Wang et al., 2024a) emphasizing cross-language consistency and multicultural reasoning, involves tasks for cultural understanding and complex reasoning. CHARM (Sun et al., 2024) is built for comprehensive and in-depth evaluation of LLMs in Chinese commonsense reasoning and revealing the intrinsic correlation between memorization and reasoning. There are also benchmarks aimed at specialized fields, like LawBench (Fei et al., 2024), which include tasks for both memorization and application. However, these methods heavily rely on human annotation and lack scalable dataset generation. Due to the vague definition and granularity of basic knowledge memorization, it is hard to explain LLM's reasoning errors and provide in-depth insights for evaluation processes. Compared to other methods, CSKG-based LLM evaluation is more structured, fine-grained, and interpretable. It distinguishes memorization, comprehension, and application while reducing data contamination. Additionally, it enables automated and multi-level assessment, making it a powerful tool for evaluating commonsense abilities.

G Examples in CoCo

As illustrated in Table 1, the number of samples (question sets) for the memorization, comprehension, and application tasks are 18,000, 13,060, and 7,942, respectively. These tasks encompass various relation types from ATOMIC and different query structures. For the comprehension task, the 1+3 Joint Test consists of one memorization test and three conceptualization tests. Similarly, for the application task, the 2+1 (or 3+1) Joint Test includes two (or three) memorization tests along with one reasoning test. We present different task examples in CoCo with different relation types and query structures in Table 10, Table 11 and Table 12, respectively.

Type	Question	Options	Answer
oEffect	What is the effect on PersonY after PersonX throws stones at PersonY?	A: happy. B: lucky. C: head bleeds. D: to block the sun from their eyes. E: become experiment subject.	C
oReact	What does PersonY feel after PersonX plays PersonY's guitar?	A: thanks PersonX. B: to swallow the liquid in their mouth. C: the dog takes it. D: positive. E: interested.	E
oWant	What does PersonY want to do after PersonX keeps PersonY's promises?	A: to understand the subject. B: accept what PersonX decides. C: to thank PersonX. D: good about themselves. E: knows where PersonY is.	C
xAttr	What is PersonX seen as given PersonX watches youtube videos?	A: looking for entertainment. B: PersonY goes to the ocean. C: sad. D: problem. E: gets a new job.	A
xEffect	What is the effect on PersonX after PersonX sits quietly?	A: poor. B: PersonY listens carefully. C: to prevent breakage. D: surprised. E: stays quiet.	E
xIntent	What is the intention of PersonX before PersonX props up the bar?	A: to increase income. B: to become intoxicated. C: To read the book. D: gains stained carpet. E: to be challenged.	B
xNeed	What does PersonX need to do before PersonX climbs onto the bed?	A: Learns a new skill. B: students receive homework. C: becomes nervous. D: to put on their pajamas. E: PersonX's son thanks them.	D
xReact	What does PersonX feel after PersonX begs PersonY to take?	A: gets sentenced to 10 years. B: ambitious. C: PersonX gets a walking cast. D: helpful. E: studious.	D
xWant	What does PersonX want to do after PersonX kicks the ball?	A: to score a goal. B: finds something to cheer them up. C: to rest up. D: tumbles. E: to be noticed.	A

Table 10: Examples of the memorization task.

Type	Subquestion	Question	Options	Answer
oEffect	What is the effect on PersonY after PersonX asks PersonX's grandma?	What is the effect on PersonY after PersonX asks family elder?	A: cuddle PersonY. B: sympathetic. C: shows love for PersonX. D: leave their old job. E: responsive.	C
		What is the effect on PersonY after PersonX asks grandparent?	A: to submit their checklist. B: shows love for PersonX. C: gets in line. D: motivated. E: safety.	B
		What is the effect on PersonY after PersonX asks family patriarch?	A: mean. B: food burns. C: to go to the beach. D: to solve his problem. E: shows love for PersonX.	E
oReact	What does PersonY feel after PersonX argues with PersonY's boyfriend?	What does PersonY feel after PersonX argues with significant other of PersonY?	A: States they don't believe X. B: They can afford to buy more things. C: looks into getting a bullet proof vest. D: hurt. E: to wipe their face.	D
		What does PersonY feel after PersonX argues with romantic partner of PersonY?	A: to enjoy life. B: hurt. C: Hold arm together. D: professional. E: to cure his problem.	B
		What does PersonY feel after PersonX argues with PersonY's beau?	A: Feed and clothe them. B: study. C: hurt. D: a job. E: catch personY doing something wrong.	C
oWant	What does PersonY want to do after PersonX achieves PersonY's effect?	What does PersonY want to do after PersonX achieves anticipated impact?	A: PersonY is safe. B: to contact someone. C: arrested for littering. D: to paint the walls. E: to see.	E
		What does PersonY want to do after PersonX achieves desired consequence?	A: updates. B: to figure out solution. C: to see. D: happy about decision. E: content.	C
		What does PersonY want to do after PersonX achieves intended result?	A: well satisfied. B: to see. C: to go back home and sleep. D: walks away. E: to have a night cap at the bar.	B
xAttr	What is PersonX seen as given PersonX agrees to a date?	What is PersonX seen as given PersonX agrees to optimistic behavior?	A: hopeful. B: happy. C: to learn Japanese. D: to ask a question. E: help accommodate others.	A
		What is PersonX seen as given PersonX agrees to positive social interaction?	A: happy. B: hopeful. C: in need. D: gets a loan. E: satisfied.	B
		What is PersonX seen as given PersonX agrees to optimistic activity?	A: gives someone a raise. B: to play at the park with the dog. C: soft-hearted. D: to get medical help. E: hopeful.	E
xEffect	What is the effect on PersonX after PersonX adopts a dog?	What is the effect on PersonX after PersonX adopts loyal companion?	A: helpful. B: excited. C: PersonX relaxes neck. D: PersonX names it. E: to call PersonY.	D
		What is the effect on PersonX after PersonX adopts four-legged friend?	A: hopeful. B: to watch how he does. C: PersonX names it. D: strong. E: unfit.	C
		What is the effect on PersonX after PersonX adopts faithful friend?	A: gesture and use body to demonstrate skills. B: spend his winnings. C: blissful. D: Has no troubles. E: PersonX names it.	E
xIntent	What is the intention of PersonX before PersonX arranges PersonY's interview?	What is the intention of PersonX before PersonX arranges job interview?	A: try to get refund. B: to sign the petition. C: to analyze. D: to save money. E: opens the door.	C
		What is the intention of PersonX before PersonX arranges applicant evaluation?	A: Waits for a response. B: Nosey. C: skinny. D: to analyze. E: to know what they want.	D
		What is the intention of PersonX before PersonX arranges interview process?	A: to analyze. B: gets energized. C: PersonX sighs as PersonY's dog barks loudly. D: bends down the body. E: to find someone to talk to.	A
xNeed	What does PersonX need to do before PersonX accepts into college?	What does PersonX need to do before PersonX accepts into higher education institution?	A: to apply to college. B: to see it succeed. C: to get a scissors. D: to defend their position. E: to congratulate PersonY.	A
		What does PersonX need to do before PersonX accepts into post-secondary institution?	A: to apply to college. B: to help PersonY. C: very proud. D: Goal setter. E: To be patient.	A
		What does PersonX need to do before PersonX accepts into university?	A: to apply to college. B: donates to charity. C: to call his friend for playing. D: is no longer confused. E: good pay.	A
xReact	What does PersonX feel after PersonX asks PersonX's doctor?	What does PersonX feel after PersonX asks medical expert?	A: to avoid doing something. B: responsible. C: Goal setter. D: personX is snuggled by the cat. E: knowledgable.	E
		What does PersonX feel after PersonX asks knowledgeable professional?	A: gather materials. B: knowledgable. C: to make friends. D: Wondering. E: remorseful.	B
		What does PersonX feel after PersonX asks trusted advisor?	A: appreciative. B: elated that they have proved their client to be innocent. C: knowledgable. D: to take some medicine. E: calls principal.	C
xWant	What does PersonX want to do after PersonX acts strange?	What does PersonX want to do after PersonX acts bizarre demeanor?	A: to have looked at PersonY's resume. B: to get to safety. C: Voters think about PersonX. D: finished. E: anticipating.	B
		What does PersonX want to do after PersonX acts unusual behavior?	A: to be dry. B: to be looked up to. C: Regretful. D: to go to the 19th hole for a drink. E: to get to safety.	E
		What does PersonX want to do after PersonX acts odd conduct?	A: to get to safety. B: guilty. C: free-spirited. D: to reassure PersonY. E: to decide they like pizza.	A

Table 11: Examples of the comprehension task.

Type	Subquestion	Question	Options	Answer
2i	What is PersonX seen as given PersonX fills PersonY's glass? What does PersonX feel after PersonX gets beer?	What event or state is both what PersonX is seen as given PersonX fills PersonY's glass and also what PersonX feels after PersonX gets beer?	A: PersonX gets hit on. B: PersonX is tipsy. C: PersonX sees at school. D: PersonY they do the dishes. E: PersonX goes camping with PersonX's friends.	B
2p	What is the effect on PersonX after PersonX does PersonX's hair and makeup? What does PersonX need to do before PersonX looks pretty?	What event or state is what PersonX needed to do before the effect on PersonX after PersonX does PersonX's hair and makeup?	A: PersonX is tired. B: PersonX curl hair. C: PersonX smiles. D: PersonX goes from bad to worse. E: PersonY communicate with PersonX.	B
ip	What does PersonX need to do before PersonX work hard and well? What is PersonX seen as given PersonX is deserving? What is the effect on PersonX after PersonX gets promoted?	What event or state is the effect on PersonX after both what PersonX needed to do before PersonX work hard and well, and also what PersonX is seen as given PersonX is deserving?	A: PersonX finishes the movie. B: PersonX learns a new language. C: PersonX looks at persony. D: PersonX is sick. E: PersonX orders a cake.	B
pi	What is the effect on PersonX after PersonX asks PersonY out on a date? What is PersonX seen as given PersonX gets a date with PersonY? What is the effect on PersonX after PersonX loses twenty pounds?	What event or state is both what PersonX is seen as given the effect on PersonX after PersonX asks PersonY out on a date, and also the effect on PersonX after PersonX loses twenty pounds?	A: PersonX is attractive. B: PersonX brews PersonX's own beer. C: PersonX buys all the ingredients. D: PersonX speak out loud. E: PersonY is angry.	A

Table 12: Examples of the application task.