Can LLMs Reason with Rules? Logic Scaffolding for Stress-Testing and Improving LLMs

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large language models (LLMs) have achieved impressive human-like performance across various reasoning tasks. However, their mastery of underlying inferential rules still falls short of human capabilities. To investigate this, we propose a logic scaffolding inferential rule generation framework, to construct an inferential rule base, ULogic, comprising both primitive and compositional rules across five domains. Our analysis of GPT-series models over a rule subset reveals significant gaps in LLMs' logic understanding compared to human performance, especially in compositional and structural complex rules with certain bias patterns. We further distill these rules into a smaller-scale inference engine for flexible rule generation and enhancing downstream reasoning. Through a multi-judger evaluation, our inference engine proves effective in generating accurate, complex and abstract conclusions and premises, and improve various commonsense reasoning tasks. Overall, our work sheds light on LLMs' limitations in grasping inferential rule and suggests ways to enhance their logical reasoning abilities.

1 Introduction

"*Did Leonardo da Vinci ever use a laptop for drawing pictures?*" Large language models can swiftly and confidently respond "*No*" [10, 35], demonstrating impressive reasoning ability that rivals human [18, 19]. However, when posed with more obscure questions, such as Q2 in Figure 1, LLMs are prone to exhibit uncertainty and errors. This inconsistency raises concerns about whether LLMs grasp the underlying logic of matters as proficiently as humans [38] (see "*underlying logic*" in Figure 1) and highlights challenging reasoning situations (like Q2) where current LLMs might struggle. Humans naturally abstract underlying logic as inferential rules from extensive real-world observations [3], beneficial for addressing diverse reasoning situations. An inferential rule is typically defined as a premise with a set of facts (e.g., "*Person X died before ... earlier than B*") leading to a conclusion (e.g., '‘*Person X cannot access Object Y*") [5]. Grasping this rule enables the deduction that a person cannot access an object invented posthumously. This work utilizes symbolic logic as a *scaffold* to generate challenging

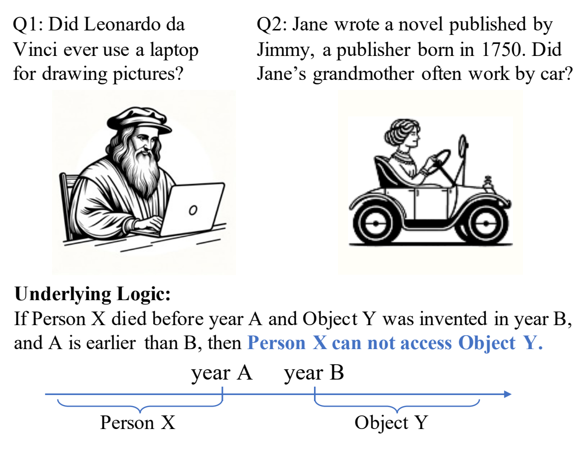


Figure 1: The underlying logic to answer Q1 and Q2.

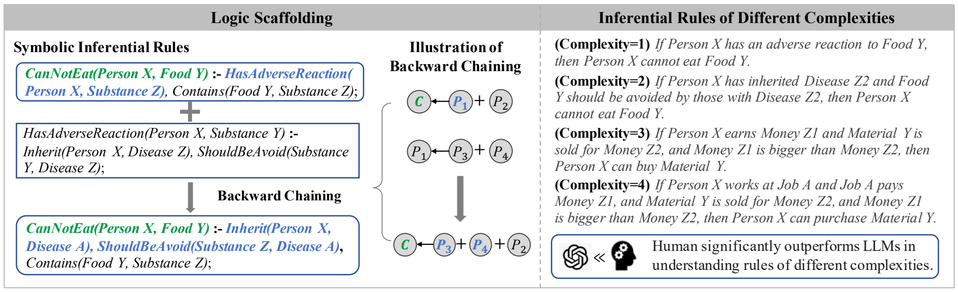


Figure 2: Logic scaffolding uncovers challenging reasoning space for LLMs.

reasoning situations for GPT-series LLMs, as shown in Figure 2. A discernible gap exists between LLMs and humans in understanding inferential rules, especially rules with complex premises.

However, collecting inferential rules at scale is challenging. Previous work relies on manual curation [24, 29] or inductive logic programming [22], which are either labor-intensive or limited in diversity. Besides, manually crafted rules often appear simple and overly specified, struggling to move beyond basic intuition or generalize across diverse situations. For example, the rule *If X runs out of steam, then X becomes tired* from [24] has only one premise fact and narrowly specifies exhaustion.

To this end, we introduce **L**ogic scaff**O**lding **I**nferential **R**ule g**E**neration (**LOIRE**), a framework to generate inferential rules of different complexities. LOIRE operates in two stages: primitive rule generation and rule composition. Initially, we define "primitive rules" to describe abstract objects like `Person` and `Food`, and ensure they cannot be decomposed into simpler rules, facilitating broad generalization and easy generation. We then incorporate GPT-4's generative capability and human expertise to generate primitive rules with high confidence. This process, consistently guided by symbolic logic, involves GPT-4 drafting potential conclusions in various domains, and forming premises with one or more facts. We ensure rules' logical soundness through the model's self-critique and human manual verification. In the second stage, we apply backward chaining [8, 1] upon primitive logical rules to automatically construct compositional rules of varied lengths and structures at scale.

Using this framework, we construct **ULogic**, an inferential rule base with around $8,000$ primitive and $6,000$ compositional rules across five domains: object affordance, accessibility, interaction, location, and human need. We hope ULogic will serve as a valuable resource, facilitating the assessment of LLMs' proficiency in underlying logic and enhancing flexible rule generation and downstream reasoning. We use ULogic to create an entailment probing task with a comprehensive and robust evaluation strategy, comparing LLMs' grasp of inferential rules to human performance. Our analysis of GPT-series LLMs (GPT-4, GPT-3.5-Turbo and GPT-3.5-Turbo-Instruct) indicates they have a basic understanding of inferential rules but fall short of human proficiency, especially in rules with complex premises. Specifically, all models struggle more as the compositional complexity increases. While GPT-4 performs consistently on verbalized and symbolic rules, the other models sharply degrade on symbolic rules. Additionally, all models exhibit disparities on various rule structures with Disjunctive-Transitive rules posing the greatest challenges. Moreover, these LLMs display notable polarity biases with GPT-4 showing a necessary bias, underscoring areas for improvement.

We further distill crafted inferential rules into a smaller-scale inference engine for flexible rule generation and downstream reasoning. We design three tasks: conclusion generation, premise completion and premise generation, to construct an instruction-tuning dataset for inferential rule distillation. Experimental results through a multi-judger evaluation mechanism incorporating automatic metrics, LLM evaluators and human preferences show that our inference engine possesses the ability for these three tasks. It outperforms GPT-3.5-Turbo across all dimensions of three tasks and even surpasses GPT-4 in generating more complex and abstract rules. Moreover, it can generate logical rules that enhance downstream commonsense reasoning.

2 Logic Scaffolding for Inferential Rule Generation

2.1 Preliminary of Inferential Rules

To better control the generative capability of LLMs for rule generation, we focus on *if-then* inferential rules with variables, that can be easily expressed as symbolic logic [16]. An inferential rule describes

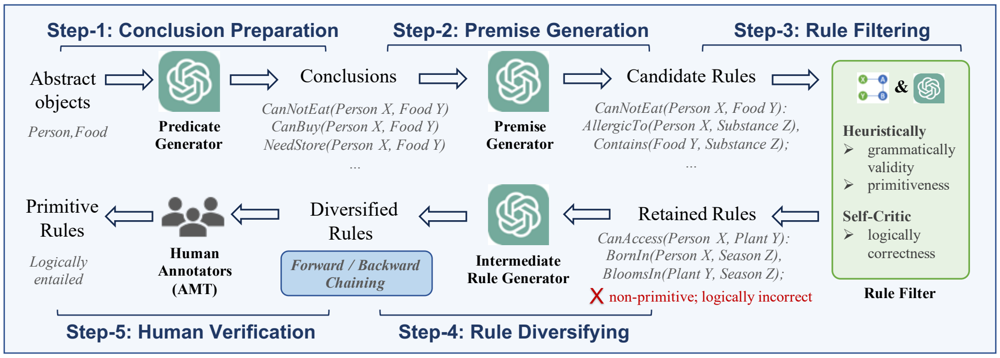


Figure 3: The pipeline of primitive rule generation.

a logical implication from a premise (a set of facts) to a conclusion (a specific fact), where each fact is a predicate expression with two variables, and each variable has a designated variable type. For each rule, we employ logic scaffolding which first generates its symbolic expression to consistently guide its verbalized form. We utilize Prolog [2] to formulate symbolic rules as `Conclusion:-Premise`, where `:-` indicates the logical implication. For example,

$$\begin{aligned}&\mathit{CanNotEat}(\mathit{Person}\ X, \mathit{Food}\ Y)\text{:-}\\&\qquad \mathit{AllergicTo}(\mathit{Person}\ X, \mathit{Substance}\ Z), \mathit{Contains}(\mathit{Food}\ Y, \mathit{Substance}\ Z).\end{aligned} \tag{1}$$

The left-hand side is the conclusion and the right hand lists premise facts connected by commas. "CanNotEat", "AllergicTo" and "Contains" are predicate verbs while `Person, Food, Substance` are variable types of variables (X, Y, Z). This symbolic rule can be verbalized as: *If Person X is allergic to Substance Z and Food Y contains Substance Z, then Person X cannot eat Food Y*.

Primitive Rule We aim to generate primitive rules for further compositions and potential generalization. We formally define primitive rules as follows: (1) they concern abstract objects, like `Person` and `Food`, rather than specific instances, and their common properties; (2) they cannot be decomposed into simpler rules. Inspired by superordinate objects such as `instrument, fruit, tool` from [23], we assemble a collection of abstract objects. We first identify the most common tail nodes of "IsA" relations from ConceptNet [30]. For those nodes that are still fine-grained, we further seek their general hypernyms by searching ConceptNet and WordNet [15]. We totally gather a list of 32 most common abstract objects for primitive rule generation, with 18 common properties generated by prompting GPT-4, as detailed in Appendix A.1.

2.2 Primitive Rule Generation Pipeline

The pipeline of primitive rule generation is illustrated in Figure 3, consisting of five steps. First, we randomly select two abstract objects, and generate potential predicates between them to form conclusions. GPT-4 is prompted to generate corresponding feasible premises with both single and multiple facts, thereby constructing candidate primitive rules. We then apply heuristic methods to filter invalid and non-primitive rules, and utilize GPT-4 to select the rules it deems logically correct. We further diversify rule predicates via backward/forward chaining [34, 27] with generated single-fact rules, and filter excessively repetitive rules. Finally, the diversified rules undergo manual verification to ensure the final set of high-confidence primitive rules.

Step-1: Conclusion Preparation From the set of abstract objects, we select any two, e.g., `Person` and `Food`, and prompt GPT-4 to generate potential predicates connecting them as conclusions, e.g., *CanEat(Person X, Food Y)*. We attempt every possible pairing of two, where the selected objects can be identical. For each pair of objects, {object_1} and {object_2}, we aim to generate conclusions across five domains: {object affordance, accessibility, interaction, location and person's need}, thereby covering diverse scenarios. Explanations and example rules of these domains, and the prompt are listed in Appendix A.2. Besides, we negate the generated predicates to yield both positive and negative conclusions, e.g., *CanNotEat(Person X, Food Y)*, across object affordance, accessibility, and interaction domains, building a complete rule set.

Step-2: Premise Generation Guided by a symbolic conclusion, we prompt GPT-4 to generate its premises in both symbolic and verbalized forms for better controllability. This process involves the logit bias setting, motivating premises to describe relationships between abstract objects and their properties. Specifically, premises are generated under the constraint of logit bias, increasing the likelihood of these objects and properties appearing in the output. For each conclusion, we create both single-fact and multi-fact premises to yield candidate rules of varying lengths. We tailor instructions and demonstrations for each domain to prompt GPT-4 for premise generation exploring different possibilities, as detailed in Appendix A.4.

Step-3: Rule Filtering After over-generating candidate primitive rules, we first design heuristic methods to filter grammatically invalid or non-primitive rules based on their symbolic forms. For grammatically validity, we check if the variables in the premises form a connected graph from node "X" to node "Y", as in Appendix A.5. For primitiveness, we exclude rules with non-primitive variable types or those comprising more than 3 premise facts. Besides, we eliminate trivial rules containing negative words in both the premise and conclusion, e.g., *CanNotEat(Person X, Food Y):- CanNotAccess(Person X, Food Z)*. Since directly generating logically correct rules is challenging, we further adopt a self-critic strategy [11] where GPT-4 critiques the accuracy of its self-generated rules in a verbalized format, and provides explanations. When prompting GPT-4, we include two demonstrations featuring both correct and incorrect rules to mitigate label bias. These demonstrations vary across different domains. An example prompt for object affordance is in Appendix A.6.

Step-4: Rule Diversifying To increase the variety of rule expressions, we diversify predicates while maintaining its logical accuracy. Based on symbolic rules, we respectively apply forward and backward chaining algorithms to their conclusion and premise with generated single-fact rules, as shown in Figure 4. In forward chaining, we take the conclusion as a new premise to generate an intermediate single-fact rule, subsequently substituting the original conclusion with this newly derived conclusion. In backward chaining, a premise is taken as a conclusion to create an intermediate single-fact rule, and replace the original premise with the new-generated one. Intermediate single-fact rules are also generated through Step-2 and 3. Each original rule undergoes one forward and one backward chaining to derive two diversified rules.

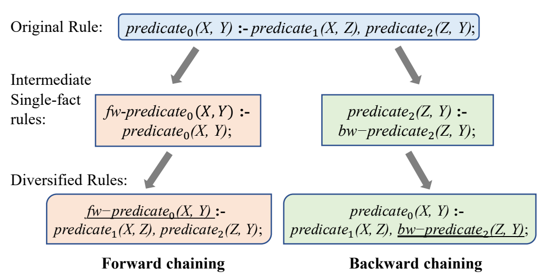


Figure 4: The forward and backward chaining process for diversifying rules.

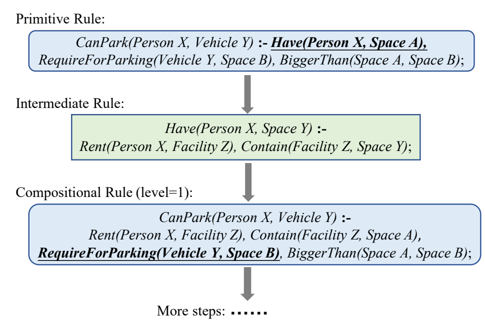


Figure 5: Illustration of one backward chaining step.

Step-5: Human Verification To obtain more reliable rules, we utilize Amazon Mechanical Turk (AMT) to recruit three annotators for manual verification of each rule. They are asked to assess the clarity and comprehensibility of its premise and conclusion, and the logical entailment from the premise to the conclusion. Only the rules unanimously validated by all three annotators are preserved. The AMT template for human verification and rule acceptance rates are listed in Appendix A.7.

2.3 Rule Composition

We create more compositional rules by applying backward chaining upon primitive rules with different chaining steps. In each step, we select a premise fact from the current rule as a conclusion, deriving a new primitive rule that describes its multi-fact premise. This selected fact is then replaced with the newly generate premise. This process is iteratively conducted 1 to 3 times, creating rules with varying compositional levels (1 to 3). An example of one backward chaining step is shown in Figure 5. The intermediate primitive rules used in backward chaining are generated via the pipeline described in Sec. 2.2, thus also contributing to our primitive rule set. As the composition of logically correct sub-rules is also logically correct, there is no need to verify these compositional rules separately.

2.4 Rule Statistics

Using LOIRE framework, we construct an inferential rule base ULogic comprising $14,647$ rules, with $7,967$ primitive and $6,680$ compositional ones. These rules span five key domains: object affordance, accessibility, interaction, location and person's need. They vary in compositional depth from 0 to 3, with rule lengths ranging from 1 to 6. Detailed statistics are in Appendix A.8.

3 Assessing LLMs' Proficiency in Capturing Inferential Rules

We utilize ULogic for a systematic evaluation of LLMs' proficiency in underlying logic compared to human competence. Specifically, we select a high-quality probing subset of 1,104 diverse, author-verified rules from our rule base (varying in lengths, polarities and structures), and create a binary entailment classification task for assessing LLMs' ability to capture inferential entailment.

3.1 Analysis Setup

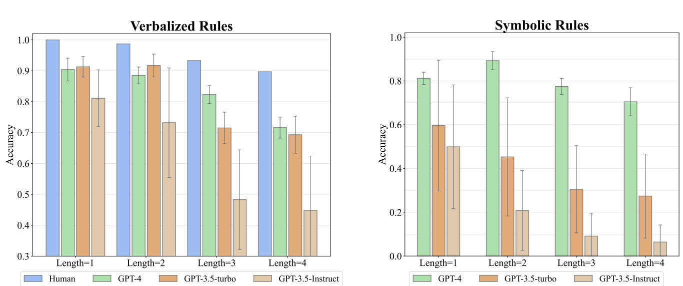
Considering LLMs' sensitivity to various input formulations and shortcut biases, we design a comprehensive and robust assessment mechanism to ensure reliable analysis. For each inferential rule, we convert it into five distinct probing questions to mitigate template bias, as summarized in Appendix B.1. We report the average accuracy and variance (the error line of each bar) across five templates. Besides, we adopt a two-shot chain of thought (CoT) prompting strategy [39] requiring the model to generate a rationale after presenting its answer, using "and also explain why." We include one correct rule and one incorrect rule in the two demonstrations to minimize label bias.

Following the Law of Non-Contradiction [21], the propositions "If X, then Y" and "If X, then not Y" are mutually exclusive that cannot both be true at the same time. To enhance the reliability of our probing, we flip each rule by negating its conclusion, and simultaneously probe both the original rule and its flipped version. A rule is accurately classified only if the original rule is affirmed (True/Right/Yes) and its flipped counterpart is negated (False/Wrong/No), as shown below. A specific example is in Appendix B.2. This dual-sided probing is applied to both human and LLMs.

If Premise, then Conclusion_original.	True/Right/Yes
If Premise, then Conclusion_flipped.	False/Wrong/No

3.2 Empirical Analysis

We conduct analysis on GPT-series LLMs, including GPT-4, GPT-3.5-Turbo and GPT-3.5-Turbo-Instruct, aiming to investigate LLMs' proficiency of inferential rules against human performance by exploring the following questions. The human performance is obtained by asking AMT annotators whether the input rule is logical correct with high probability. Each performance presented in following bar charts is calculated based on 150 instances randomly sampled from our probing subset.



(a) (b)

Figure 6: Probing results across varied lengths.

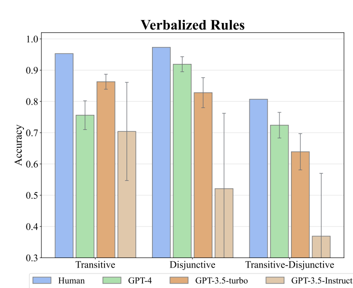


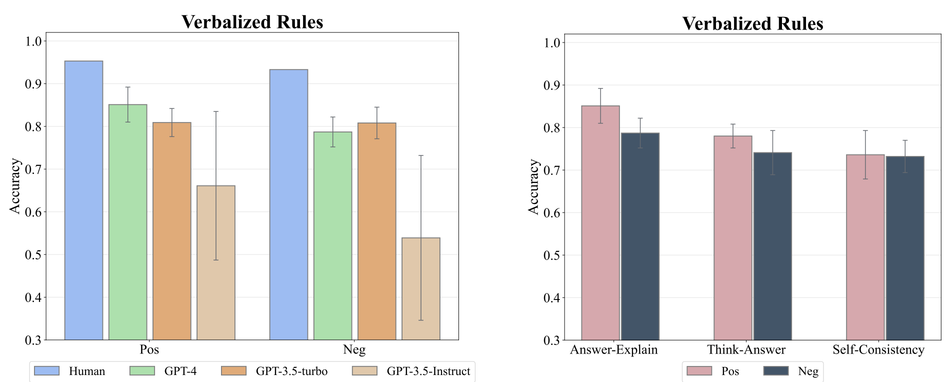
Figure 7: Result of varied structures.

(1) How does model performance vary with increasing compositional complexity? We conduct rule probing in terms of different compostional lengths, as illustrated in Figure 6a. "Length=1,2,3,4" respectively denote rules with 1∼4 facts in their premises. The analysis of different compostional depths is also provided in Appendix B.3. They both reveal that as compositional complexity increases, the performance of both human and all models drops. The primary reason is that compositional complex rules typically necessitate the aggregation of multi-step reasoning, which escalates higher-order relationships understanding and exponential error accumulation with each additional step [7]. Besides, there is a persistent performance gap between all models and human, particularly pronounced with compositional complex rules, suggesting significant potential for enhancement in this area.

(2) Are LLMs proficient in capturing both symbolic and verbalized rules? We further analyze LLM performance on symbolic rules (see Figure 6b) compared to on verbalized rules in Figure 13. GPT-4 achieves consistent performance on verbalized and symbolic rules, whereas GPT-3.5-Turbo and GPT-3.5-Instruct sharply degrade on symbolic rules. This suggests that the GPT-3.5 series may have limitations in generalizing across varied types of linguistic structures beyond natural language, whereas GPT-4 likely have undergone specific optimizations for symbolic interpretations.

(3) Are there performance disparities among models concerning different rule structures? Our generated multi-fact rules (Length > 1) have three intrinsic structures: Transitive, Disjunctive and Disjunctive-Transitive. Specific illustrations and examples of each structure are detailed in Appendix B.4. Figure 7 shows that Disjunctive-Transitive rules pose greater challenges compared to Transitive and Disjunctive ones, especially for GPT-3.5-Turbo and GPT-3.5-Instruct. We hypothesize that this discrepancy stems from increased compositional complexity and LLMs' insufficient learning of logical structures in natural language.

(4) Do LLMs exhibit a polarity bias over inferential rules? Our inferential rules contain both positive and negative conclusions. As shown in Figure 8a, GPT-4 and GPT-3.5-Instruct exhibit a pronounced positive bias, performing better on rules with positive conclusions. This bias may originate from the imbalanced distribution of LLMs' training data [9], with a higher proportion of positive statements. We further explore different CoT strategies with GPT-4: (1) first answer then explain (*Answer-Explain*), (2) first think then answer (*Think-Answer*), (3) self-consistently think then answer (*Self-Consistency*) [37]. Various CoT prompts are listed in Appendix B.5. Figure 8b shows that although advanced CoT strategies can mitigate the positive bias, they adversely impact the performance on rules with both positive and negative conclusions.



(a) Answer-Explain strategy.

(b) Various CoT strategies.

Figure 8: Rule Polarity Comparison.

(5) Why does GPT-4 significantly underperform GPT-3.5-Turbo on transitive rules? While GPT-4 generally outperforms or matches other models, this superiority disappears on transitive rules, as evidenced in Figure 7. We investigate this question in Appendix B.6, which reveals that GPT-4 exhibits a "necessary bias" that tend to consider all necessary conditions reaching a conclusion, avoiding definite judgement. This conservative style may come from LLMs' preference alignment during Reinforcement Learning with Human Feedback [19].

Overall, GPT-4 performs best in grasping inferential rules. But compared to human performance, there still remains substantial room for improvement across all models, especially in highly compositional, symbolic and structural complex rules. Besides, all models tend to exhibit a polarity bias towards rules with positive conclusions with GPT-4 also showing a necessary bias. These findings suggest potential areas for future enhancements.

4 Rule Distillation as Inference Engine

4.1 Instruction Dataset & Model Tuning

For flexible rule generation and benefiting downstream reasoning, we distill our crafted rules into a smaller-scale inference engine as illustrated in Appendix C.1. We tailor three tasks: conclusion generation, premise completion and premise generation, to construct an instruction-tuning dataset for inferential rule distillation. The detailed definitions of these tasks are also described in Appendix C.1.

We gather all primitive rules and partial compositional rules to formulate the instruction-tuning dataset, as compositional rules are constructed from primitive ones. We take 10,703 rules for training and 943 for testing. Altogether, we create 39,887 instances for instruction tuning, including 10,703, 18,500 and 10,684 for conclusion generation, premise completion and premise generation. We have 3,500 testing instances, divided as 943, 1,614 and 943 for these three tasks. We use Mistral-7b [13] as the backbone model and fine-tune it with our constructed instruction dataset as our inference engine. The training details and demo page can be found in Appendix C.2.

4.2 Rule Generation Evaluation

We compare our inference engine against GPT-4 and GPT-3.5-Turbo across three tasks to assess rule generation. For a fair comparison, we prompt GPT-4 and GPT-3.5-Turbo to simultaneously generate symbolic and verbalized responses, using similar prompts as in Step-2 of Sec. 2.2. Detailed prompts are in Appendix C.3. We introduce a multi-judger evaluation mechanism, incorporating automatic metrics, LLM evaluator and human preference to evaluate logical accuracy in conclusion generation and premise completion. For premise generation task with a specified number of facts, we generate three potential premises for each conclusion, and evaluate them on accuracy, diversity, complexity and abstractness (see Appendix C.4 for detailed metric definitions).

Automatic Evaluation For automatic accuracy evaluation of three tasks, we calculate BLEU score [20] against reference responses. For complexity of premise generation, we assess the average fact number of three generated premises. For diversity, we compute average Self-BLEU [28, 32] between three generated premises. Specifically, Self-BLEU measures the BLEU score of a generated premise against another, and a high average Self-BLEU indicates low diversity. Abstractness is not easy to evaluate automatically, so we leave it to LLM evaluation. The results are shown in Table 1.

Table 1: Automatic evaluation results.

Task	Conclusion Generation	Premise Completion	Premise Generation		
Metrics	BLEU	BLEU	BLEU	Self-BLEU	Fact Num.
Engine	**0.739**	**0.527**	**0.411**	0.687	**3.42**
GPT-4	0.414	0.179	0.149	**0.805**	2.58
GPT-3.5	0.338	0.248	0.084	0.739	1.72

LLM Evaluation We adopt GPT-4 as an evaluator to rate the generated responses on a scale from 1 to 3. The criteria of each rating along with examples are provided to the evaluator. Please see Appendix C.5 for detailed prompts. For each task, we select 100 instances for LLM evaluation, ensuring a balance across all domains and all types. The rating results are presented in Table 2.

Table 2: LLM evaluation results.

Task	Conclusion Generation	Premise Completion	Premise Generation			
Metrics	Accuracy	Accuracy	Accuracy	Diversity	Complexity	Abstractness
Engine	2.44	**2.78**	2.34	1.89	**1.62**	**2.43**
GPT-4	**2.53**	2.72	**2.77**	**2.64**	1.40	2.32
GPT-3.5	2.38	1.57	1.91	1.72	1.06	2.30

Human Evaluation To better assess premise generation in line with human value, we further recruit two annotators for each instance to compare their accuracy. We implement a pairwise comparison setting, asking annotators to determine which group of generated premise is more accurate in terms of logical consistency with the given conclusion, commonsense alignment and correctness of fact numbers. The results are shown in Fiure 9. From all evaluation, we can see that our inference engine enables the smaller-scale LLM with the capability for conclusion generation, premise completion and premise generation. It performs better than GPT-3.5-Turbo across all metrics in three tasks, and even outperforms GPT-4 to generate more complex and abstract rules.

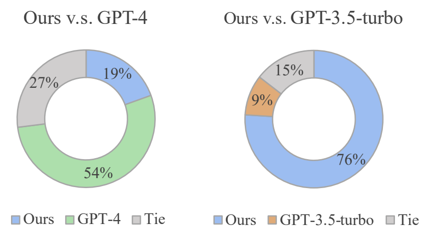


Figure 9: Human comparison results.

Figure 10: Downstream reasoning performance.

Dataset	Mistral	Mistral+rules (Mistral-7b)	LLama	LLama+rules (LLama2-7b)
StrategyQA	54.50	**56.75**	58.00	**60.48**
SOCIAL IQA	64.00	**68.50**	53.50	**60.50**
LINK head	53.68	**68.38**	58.09	**70.59**
LINK longtail	53.33	**67.50**	55.83	**65.00**
PIQA	65.00	65.00	58.5	**62.0**
CSQA2.0	59.00	**62.50**	64.00	60.00

4.3 Downstream Reasoning Evaluation

We further analyze the effectiveness of our inference engine in generating logical rules or explanations to enhance downstream reasoning tasks. We evaluate on following commonsense reasoning datasets: StrategyQA [10], SOCIAL IQA [25], LINK [14], PIQA [4] and CSQA2.0 [31]. We use a zero-shot CoT strategy to prompt two baseline models, Mistral-7B-Instruct-v0.1 and Llama-2-7b-chat [33], to answer questions with following explanations. We then utilize our inference engine to generate logical rules or explanations relevant to answer questions, and supplement these generated rationals to baseline models as input to enhance their performance. We compare the prediction accuracy of our inference engine augmented models against baselines. The comparative results are shown in Tabel 10. Our inference engine can generate logical rules or explanations that benefit multiple downstream commonsense reasoning tasks on top of different backbone models. For the lack of clear advantage on PIQA and performance decline on CSQA2.0, we speculate that PIQA may be contaminated during Mistral's training process, and CSQA2.0's focus is mainly on longtail commonsense knowledge rather than requiring logical rules inference, like "Is cotton candy sometimes made out of cotton?"

5 Related Work

Logical Rule Generation Logical inferential rules are crucial for everyday reasoning [10, 31], and collecting these inferential rules is challenging. Prior work mainly adopts inductive logic programming (ILP) [41, 22, 26] for rule generation. However, they can only generate rules from existing knowledge graphs and the generated rules has potential inaccuracies. Alternatively, [29] manually create a set of inferential rules for inductive reasoning, but their scope is limited to kinship. [24] construct a commonsense inferential rule base through crowdsourcing, but these rules tend to be overly simple and specific, struggling to move beyond basic intuition and generalize to varied situations. Abstract and complex rules are essential in tackling diverse complex questions, paving the way for complex reasoning and decision-making. Although LLMs have opened new avenues for generating inferential rules [42], they still struggle to automatically craft abstract and complex rules.

Integration of Logical Rules and LLMs The integration of inferential rules with LLMs has gained significant attention. This approach combines the logical interpretability of symbolic reasoning and adaptive power of neural computing, improving LLMs' logical reasoning ability. [36, 17] transform textual statements into logical expressions and conduct symbolic reasoning following logical rules. [40] train neural models using a set of inferential rules for dynamic application. This direction broadens LLMs' ability with flexible rule generation and application for complex reasoning.

6 Conclusion

This paper examines GPT-series LLMs' proficiency in capturing logical inferential rules and probes their challenging reasoning space. We introduce a logic scaffolding inferential rule generation (LOIRE) framework to create an inferential rule base ULogic, including nearly 8,000 primitive and 6,000 compositional rules across five domains. Our evaluations show that even advanced models like GPT-4 struggle with compositional and structural complex rules and exhibit certain biases. Furthermore, we distill ULogic into a smaller inference engine that performs well in generating inferential rules and benefit downstream reasoning tasks. Our work points out where LLMs need to improve in logical reasoning and offers a pathway to enhance their reasoning capabilities.

References

[1] A. Al-Ajlan. The comparison between forward and backward chaining. *International Journal of Machine Learning and Computing*, 5(2):106, 2015.

[2] K. R. Apt et al. *From logic programming to Prolog*, volume 362. Prentice Hall London, 1997.

[3] J. Barwise. Everyday reasoning and logical inference. *Behavioral and Brain Sciences*, 16(2):337–338, 1993.

[4] Y. Bisk, R. Zellers, J. Gao, Y. Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020.

[5] P. Boghossian. What is inference? *Philosophical studies*, 169:1–18, 2014.

[6] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[7] N. Dziri, X. Lu, M. Sclar, X. L. Li, L. Jian, B. Y. Lin, P. West, C. Bhagavatula, R. L. Bras, J. D. Hwang, et al. Faith and fate: Limits of transformers on compositionality. *arXiv preprint arXiv:2305.18654*, 2023.

[8] H. Gallaire and J. Minker. *Logic and data bases*. Springer Science & Business Media, 2012.

[9] A. Garg, D. Srivastava, Z. Xu, and L. Huang. Identifying and measuring token-level sentiment bias in pre-trained language models with prompts. *arXiv preprint arXiv:2204.07289*, 2022.

[10] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, and J. Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021.

[11] Z. Gou, Z. Shao, Y. Gong, Y. Shen, Y. Yang, N. Duan, and W. Chen. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*, 2023.

[12] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[13] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[14] H. Li, Y. Ning, Z. Liao, S. Wang, X. L. Li, X. Lu, F. Brahman, W. Zhao, Y. Choi, and X. Ren. In search of the long-tail: Systematic generation of long-tail knowledge via logical rule guided search. *arXiv preprint arXiv:2311.07237*, 2023.

[15] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[16] V. Novák and S. Lehmke. Logical structure of fuzzy if-then rules. *FUZZY sets and systems*, 157(15):2003–2029, 2006.

[17] T. X. Olausson, A. Gu, B. Lipkin, C. E. Zhang, A. Solar-Lezama, J. B. Tenenbaum, and R. Levy. Linc: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. *arXiv preprint arXiv:2310.15164*, 2023.

[18] OpenAI. Gpt-4 technical report, 2023.

[19] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[20] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[21] G. Priest, J. C. Beall, and B. Armour-Garb. *The law of non-contradiction: New philosophical essays*. Clarendon Press, 2006.

[22] M. Qu, J. Chen, L.-P. Xhonneux, Y. Bengio, and J. Tang. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. *arXiv preprint arXiv:2010.04029*, 2020.

[23] E. Rosch and C. B. Mervis. Family resemblances: Studies in the internal structure of categories. *Cognitive psychology*, 7(4):573–605, 1975.

[24] M. Sap, R. Le Bras, E. Allaway, C. Bhagavatula, N. Lourie, H. Rashkin, B. Roof, N. A. Smith, and Y. Choi. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3027–3035, 2019.

[25] M. Sap, H. Rashkin, D. Chen, R. LeBras, and Y. Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

[26] P. Sen, B. W. de Carvalho, R. Riegel, and A. Gray. Neuro-symbolic inductive logic programming with logical neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8212–8219, 2022.

[27] H. Shindo, D. S. Dhami, and K. Kersting. Neuro-symbolic forward reasoning. *arXiv preprint arXiv:2110.09383*, 2021.

[28] R. Shu, H. Nakayama, and K. Cho. Generating diverse translations with sentence codes. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1823–1827, 2019.

[29] K. Sinha, S. Sodhani, J. Dong, J. Pineau, and W. L. Hamilton. Clutrr: A diagnostic benchmark for inductive reasoning from text. *arXiv preprint arXiv:1908.06177*, 2019.

[30] R. Speer, J. Chin, and C. Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[31] A. Talmor, O. Yoran, R. L. Bras, C. Bhagavatula, Y. Goldberg, Y. Choi, and J. Berant. Commonsenseqa 2.0: Exposing the limits of ai through gamification. *arXiv preprint arXiv:2201.05320*, 2022.

[32] G. Tevet and J. Berant. Evaluating the evaluation of diversity in natural language generation. *arXiv preprint arXiv:2004.02990*, 2020.

[33] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[34] J. Urbani, F. Van Harmelen, S. Schlobach, and H. Bal. Querypie: Backward reasoning for owl horst over very large knowledge bases. In *The Semantic Web–ISWC 2011: 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I 10*, pages 730–745. Springer, 2011.

[35] L. Wang, W. Xu, Y. Lan, Z. Hu, Y. Lan, R. K.-W. Lee, and E.-P. Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*, 2023.

[36] S. Wang, W. Zhong, D. Tang, Z. Wei, Z. Fan, D. Jiang, M. Zhou, and N. Duan. Logic-driven context extension and data augmentation for logical reasoning of text. *arXiv preprint arXiv:2105.03659*, 2021.

[37] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[38] P. C. Wason. Reasoning about a rule. *Quarterly journal of experimental psychology*, 20(3):273–281, 1968.

[39] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[40] N. Weir and B. Van Durme. Dynamic generation of interpretable inference rules in a neuro-symbolic expert system. *arXiv preprint arXiv:2209.07662*, 2022.

[41] Y. Yang and L. Song. Learn to explain efficiently via neural logic inductive learning. *arXiv preprint arXiv:1910.02481*, 2019.

[42] Z. Zhu, Y. Xue, X. Chen, D. Zhou, J. Tang, D. Schuurmans, and H. Dai. Large language models can learn rules. *arXiv preprint arXiv:2310.07064*, 2023.

Limitations

Limitation on inferential rule coverage. Commonsense inferential rules may exist in diverse formats and span various domains. Our work mainly focuses on rules formatted as *if-then* statements, covering five domains: object affordance, accessibility, interaction, location and person's need. In future work, we will expand our scope to include inferential rules of other formats and explore additional domains for broader coverage.

Limitation on probing open-source models. Our work does not probe and analyze open-source models. While GPT-4 and GPT-3.5-turbo are considered as the most advanced models, open-source counterparts may exhibit different behaviors or patterns in understanding inferential rules with varying complexities. These aspects will be the subject of future exploration.

Risk of environmental impact A significant risk associated with our framework and analysis is the potential increase in environmental burdens due to the extensive use of OpenAI's APIs for LLMs. This impact can be mitigated by replacing GPT-4 with future smaller-scale open-source models that are more efficient with less environmental impact.

Potential error in rule generation. Generating inferential rules with specific requirements poses a significant challenge. As the majority of our framework's pipeline are powered by GPT-4, it may inevitably generate inferential rules with logical inaccuracies even incorporating human verification. This might result in less accurate probing of LLMs.

Ethical Consideration

All rules we collected through LLMs are released publicly for usage and its probing subset for proficiency analysis have been subjected to a thorough review by the authors. The code of our generation pipeline and probing experiments will also be publicly released. This setting guarantees transparency and reproducibility in our experiments, allowing other researchers to evaluate and expand upon our work. Our logic scaffolding framework is strictly limited to be used for rule generation that follow the ethical guidelines of the community. The authors emphatically denounce the use of our framework for generating inaccurate or harmful rules.

A Primitive Rule Generation Pipeline

A.1 Abstract Objects and Common Properties

Table 3 list 32 most common abstract objects and 18 common properties for primitive rule generation.

Table 3: List of pre-defined abstract objects and common properties.

Type	Words
Abstract Objects	"Person", "Animal", "Plant", "Food", "Alcohol", "Disease", "Drug", "Natural Phenomenon", "Condition", "Material", "Substance", "Furniture", "Publication", "Organization", "Authorization", "Facility", "Natural Place", "Event", "Show", "Artwork", "Job", "Game", "Vehicle", "Tool", "Technology", "Electronic Device", "Platform", "Financial Product", "Skill", "Legislation", "Region", "Time Period"
Common Properties	"Age", "Price", "Money", "Height", "Length", "Weight", "Strength", "Size", "Density", "Volume", "Temperature", "Hardness", "Speed", "BoilingPoint", "MeltingPoint", "Frequency", "Decibel", "Space"

A.2 Rule Domains

Table 4 illustrates the detailed explanations, example predicates and rules across five domains.

Table 4: The explanations, example predicates and rules of five different domains.

Domain	Explanation	Predicates	Examples
Object Affordance	Whether a person can take an action over an object based on its property and requirement	CanDrive(Person X, Vehicle Y); CanCreate(Person X, Artwork Y); CanAttend(Person X, Event Y);	CanDrive(Person X, Vehicle Y):- Have(Person X, Age Z1), RequireMinimumAge(Vehicle Y, Age Z2), BiggerThan(Age Z1, Age Z2);
Object Accessibility	Whether an object can access the other object based on its physical condition, spatial and temporal restriction	CanAccess(Person X, Show Y); CanAccess(Animal X, Tool Y); CanAccess(Animal X, Animal Y);	CanAccess(Person X, Show Y):- LocatedIn(Person X, Region Z), BroadcastIn(Show Y, Region Z); CanNotAccess(Person X, Tool Y):- AllergicTo(Person X, Material Z), MadeOf(Tool Y, Material Z);
Object Interaction	How an object can interact with the other object based on their physical, spatial or temporal properties	CanSubmergeIn(Substance X, Substance Y); CanAdaptedFrom(Show X, Artwork Y); CanFitIn(Tool X, Tool Y);	CanSubmergeIn(Substance X, Substance Y):- DensityOf(Substance X, Density Z1), DensityOf(Substance Y, Density Z2), BiggerThan(Density Z1, Density Z2);
Object Location	The location description of an object	OriginatedFrom(Food X, Region Y); BannedIn(Drug X, Region Y); BornIn(Person X, Region Y);	OriginatedFrom(Food X, Region Y):- ProcessedIn(Food X, Facility Z), LocatedIn(Facility Z, Region Y);
Person's Need	Person need to take an action over objects under a specific circumstance	NeedToConsume(Person X, Drug Y); NeedToWater(Person X, Plant Y);	NeedToConsume(Person X, Drug Y):- Has(Person X, Disease Z), CanTreat(Drug Y, Disease Z);

A.3 Prompt for Conclusion Preparation

An example of the prompt for conclusion preparation about affordance is below.

Prompt for Conclusion Preparation

According to commonsense knowledge in reality, please list 5 predicates between the given two objects to describe the {object affordance}.
Examples:
Object: Show, Artwork
Predicate: CanBeAdaptedFrom(Show X, Artwork Y)

Object: {object_1}, {object_2}
Predicate:

A.4 Prompts for Premise Generation

For premise generation in each domain, we design an instruction followed by two demonstrations to iteratively prompt GPT-4, and the underlined sentence is the rule description which varies according to the specific domain, as shown in Table 5.

A.5 Grammatical Validity for Rule Filtering

As Figure 11, we check whether the variables in premises form a connected graph from node "X" to node "Y" to filter grammatically invalid rules.

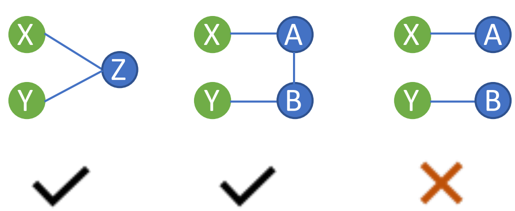


Figure 11: Grammatically valid and invalid rule graphs.

A.6 Prompts for Rule Filtering

Table 6 is an example prompt for rule filtering in object affordance domain.

Instruction for Premise Generation (Object Affordance)

According to commonsense knowledge in realistic scenarios, please generate 2 logical rules in both Prolog and natural langauge to describe the premises of the given conclusion. The rules in Prolog should have the same meaning with the rules in natural language.
Each rule should contain multiple premises and each premise should contain two variables in (X, Y, Z, Z1, Z2).
The rules should describe object affordance based on its property (such as height, age, price) and requirement (such as required skill, source, tool).
The premises should not contain negative words such as ’not’, ’no’, ’never’ and ’un-’

Conclusion: {conclusion}
Rules:

Demonstrations for Premise Generation (Object Affordance)

Conclusion: CanCook(Person X, Food Y)
Rules:
1. CanCook(Person X, Food Y):- CanUse(Person X, Tool Z), UsedForCook(Tool Z, Food Y);
If Person X can use Tool Z which is used for cooking Food Y, then Person X can cook Food Y.
2. CanCook(Person X, Food Y):- Master(Person X, Skill Z), RequiredForCooking(Skill Z, Food Y);
If Person X has mastered Skill Z which is required for cooking Food Y, then Person X can cook Food Y.

Conclusion: CanDrive(Person X, Vehicle Y)
Rules:
1. CanDrive(Person X, Vehicle Y):- Have(Person X, Age Z1), RequireMinimumAge(Vehicle Y, Age Z2), BiggerThan(Age Z1, Age Z2);
If Person X has Age Z1 and the minimum age requirement for driving Vehicle Y is Age Z2, Age Z1 is bigger than Age Z2, then Person X can drive Vehicle Y.
2. CanDrive(Person X, Vehicle Y):- Obtain(Person X, Authorization Z), RequiredForDriving(Authorization Z, Vehicle Y);
If Person X have obtained a specific Authorization Z and Authorization Z is required for driving Vehicle Y, then Person X can drive Vehicle Y.

Table 5: Prompts for rule generation in different domains.

Domain	Rule Description
Object Affordance	The rules should describe object affordance based on its property (such as height, age, price) and requirement (such as required skill, source, tool).
Object Accessibility	The rules should describe object accessibility based on its physical condition, spatial and temporal restriction.
Object Interaction	The rules should describe object interaction based on its physical, spatial or temporal properties (such as speed, hardness, density, height, time period).
Object Location	The rules should describe the location information of an object.
Person’s Need	The rules should describe person’s need to take an action over the object.

A.7 Human Verification Templates and Rates

Before human verification, we first craft a qualification task to select AMT annotators from all English-speaking countries (US, UK, New Zealand, Australia, Canada). The prospective workers are presented with three representative test cases and need to predict whether the premise and conclusion are clearly readable, and if the premise logically entails the conclusion. Only those workers correctly passing all the test cases are recruited. The detailed template for human verification is shown as Figure 12. This template is also used for getting human performance in rule probing analysis, wherein a separate cohort of workers is qualified for manual rule probing. Besides, the overall rates of rule acceptance in different domains during human verification are listed Table 7.

Prompt for Rule Filtering

True or False? Please predict whether the input rule is accurate or not according to commonsense knowledge in realistic scenarios, and also explain why.
Examples:
Input: If Person X has an Age Z1 and Vehicle Y requires an Age above Z2 for driving, with ...
Output: True. Because Person X has achieved the ...
Input: If Person X was born in Season Z and Plant Y blooms in the same Season Z, then Person X can access Plant Y.
Output: False. Because a person's birth season and a plant's blooming season has no logical connection.

Input: {candidate rule}
Output:

Table 6: A prompt for rule filtering in object affordance.

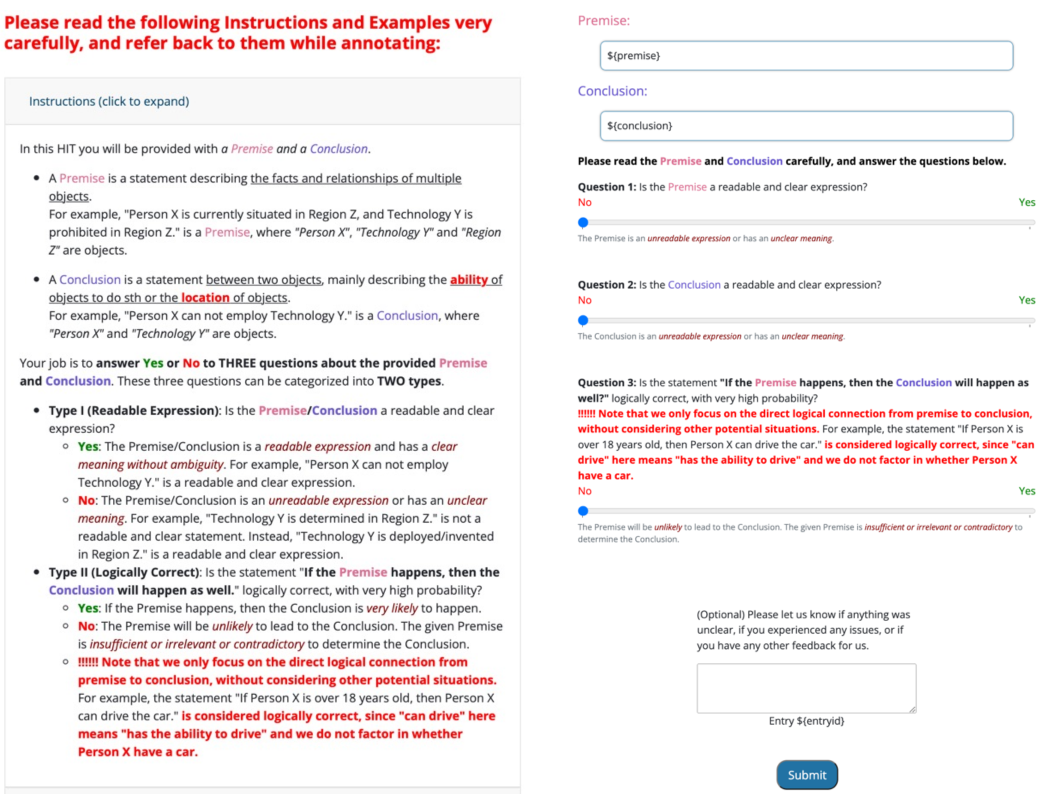


Figure 12: AMT template for human verification of primitive rules.

Table 7: The rule yield rates (%) of human verification.

	Affordance	Accessibility	Interaction	Location	Person's Need
Yield Rate	48.09	37.28	52.81	53.74	49.45

A.8 Statistics of ULogic

We construct an inferential rule base ULogic comprising $14,647$ rules, with $7,967$ primitive and $6,680$ compositional ones. These rules span five key domains: object affordance, accessibility, interaction, location and person's need. They vary in compositional depth from 0 to 3, with rule lengths ranging from 1 to 6. Detailed statistics are in Table 8.

Table 8: Statistics of constructed rule base.

Domain	Affordance	Accessibility	Interaction	Location	Need	Total
Primitive rules						7,967
Single-fact	328	513	440	194	87	1,562
Multi-fact	387	638	2,527	166	128	3,846
Intermediate	417	590	1,286	165	101	2,559
Compositional rules						6,680
Compositionality=1	322	675	936	111	91	2,135
Compositionality=2	199	773	744	100	136	1,952
Compositionality=3	229	1052	896	217	199	2,593

B Rule Probing

B.1 Rule Probing Templates

Table 9 lists five different templates for unbiased rule probing.

Table 9: Five templates for rule probing.

	Template	**Label**
1	True or False? Please predict whether the input rule is very likely to be true.	True/False
2	Right or Wrong? Please predict whether the input rule is valid and correct.	Right/Wrong
3	Yes or No? Please predict whether the premise entails the conclusion.	Yes/No
4	Premise:..., Conclusion:... Does premise entail conclusion? Please answer Yes or No.	Yes/No
5	Given the observations ..., can we draw the conclusion ...? Please answer Yes or No.	Yes/No

B.2 Dual-side Rule Probing Setting

Table 10 illustrate a concrete example of dual-side rule probing.

Table 10: A specific example of dual-side rule probing.

If Premise, then Conclusion_original.	True/Right/Yes
If Premise, then Conclusion_flipped.	False/Wrong/No
Example	
If Person X is allergic to Substance Z and Food Y contains Substance Z, then Person X ***cannot eat*** *Food Y.*	True/Right/Yes
If Person X is allergic to Substance Z and Food Y contains Substance Z, then Person X ***can eat*** *Food Y.*	False/Wrong/No

B.3 Rule Depths Probing

The analysis of GPT-series LLMs and human on different compostional depths is presented as Figure 13. “Depth=0” represents primitive rules and “Depth=1,2,3” denote compositional rules involving 1 to 3 backward chaining steps.

B.4 Illustrations of Rule Structures

Figure 14 displays several examples showcasing both symbolic and verbalized rules across different structure types.

B.5 Different CoT Prompts

Table 11 lists different prompts of three CoT strategies for rule probing.

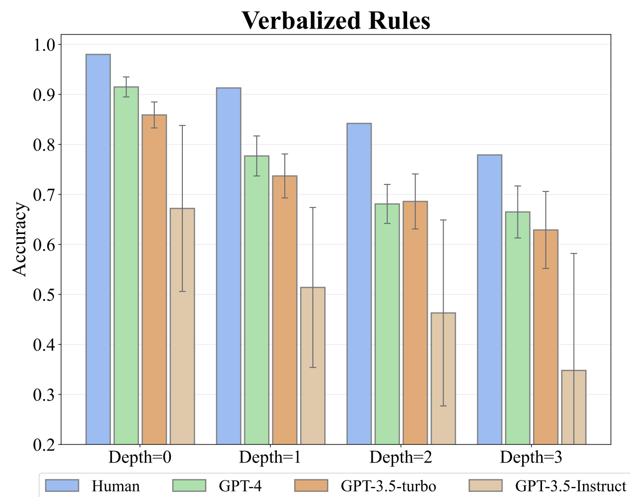


Figure 13: Probing results of varied depths.

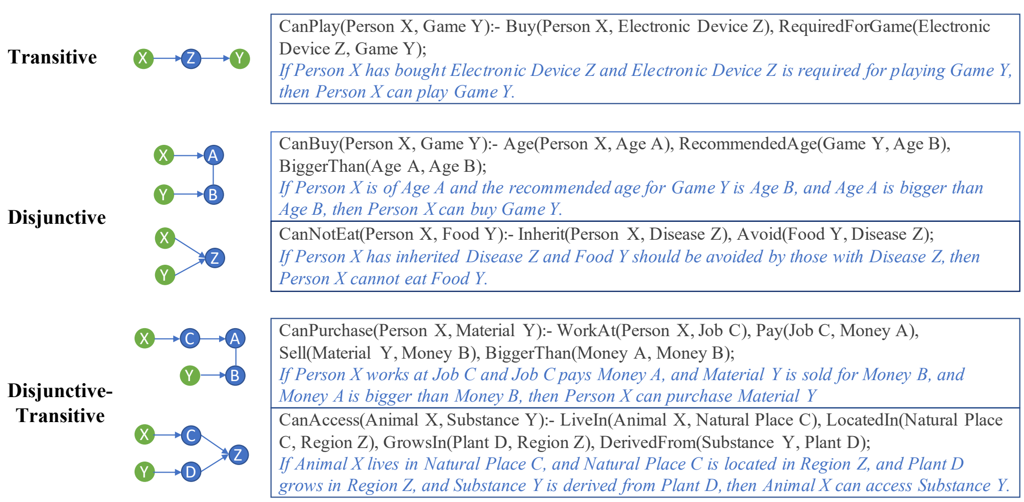


Figure 14: Example rules of different structures.

Table 11: Three CoT prompts for rule probing.

CoT strategy	Prompt
Answer-Explain	True or False? Please predict whether the input rule is very likely to be true, and also explain why.
Answer-Explain	True or False? Please predict whether the input rule is very likely to be true. Please first briefly explain your thought process in one sentence, and then give your answer.
Self-Consistency	True or False? Please predict whether the input rule is very likely to be true. Please first generate three different sentences to respectively explain your three thought processes briefly, and then based on the corresponding thought to give your answer. Finally, output the final answer according to majority voting.

B.6 Necessary Bias

As mentioned in Section 3.2, We investigate why GPT-4 significantly underperforms GPT-3.5-Turbo on transitive rules. Transitive rules typically describe a straightforward logical chain from variable X to Y, where GPT-3.5-Turbo and GPT-3.5-Instruct manage with greater ease. In contrast, we find that

GPT-4 exhibits a “necessary bias” that tend to consider all necessary conditions reaching a conclusion, which avoids drawing a definite judgement. This conservative response is more pronounced in transitive rules, where GPT-4 more frequently responds with hesitations like "it does not necessarily mean" in its explanations. We present a probing example of a transitive rule by GPT-4 in Table 12, including its generated prediction, explanation and the corresponding label. We hypothesize that this conservative style may come from LLMs’ preference alignment during Reinforcement Learning with Human Feedback [19].

Table 12: An example probing result of a transitive rules by GPT-4.

Input	True or False? Please predict whether the input rule is very likely to be true, and also explain why. Rule: If Person X has a job A that requires Skill Z, and Skill Z is required to cook Food Y, then Person X can cook Food Y.
Prediction	False
Explanation	Just because Person X has a job that requires Skill Z, it does not necessarily mean they can cook Food Y. They may have the skill, but cooking also requires other knowledge of recipes, techniques.
Label	True

Prompt for Conclusion Generation

Given the premise, please generate its conclusion between X and Y in both Prolog and natural language.
The conclusion in Prolog should have the same meaning with the conclusion in natural language.
Each conclusion should contain only two variables X and Y without mentioning other variables, like A, B, C, Z.

Examples:
Premise: If Person X is allergic to Material Z and Furniture Y is made from Material Z.
Conclusion:
[Prolog]: CanNotHold(Person X, Furniture Y);
[Natural Language]: Person X cannot hold Furniture Y.

Premise: If Substance X has a Density Z1, the density of Substance Y is Density Z2, and Density Z1 is bigger than Density Z2.
Conclusion:
[Prolog]: CanSubmerge(Substance X, Substance Y);
[Natural Language]: Substance X can submerge in Substance Y.

Premise: {premise}
Conclusion:

Table 13: Prompt ChatGPT and GPT-4 for conclusion generation.

C Inference Engine

C.1 Illustration of Instruction Tuning

Figure 15 illustrate the pipeline of instruction tuning for rule distillation as an inference engine. Our inference engine is trained for three tasks: conclusion generation, premise completion and premise generation. The conclusion generation focuses on creating a conclusion from a provided premise. For premise completion, given a conclusion and its partial premise, the inference engine must complete the remaining premise part to support the conclusion. In premise generation, the engine is tasked with creating premises of varying complexity based on a given conclusion, specifically generating premises with one, two or even more facts. We also provide an inference engine demo for flexible rule generation as shown in Figure 16.

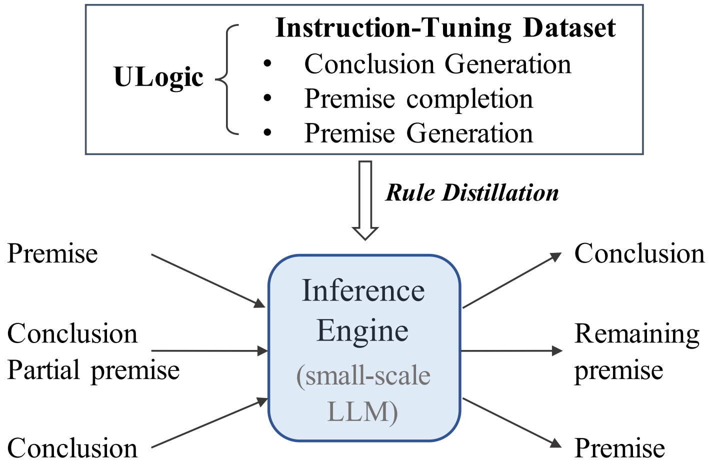


Figure 15: Rule distillation for inference engine.

C.2 Implementation Details

We fine-tune Mistral-7b with our constructed instruction dataset with Quantization LoRA (QLoRA) method [12, 6] as our inference engine. We set the learning rate to 7×10^{-5}, batch size to 8, gradient accumulation step to 16, and train the model 2 epochs. We apply QLoRA to all the linear layers of the model, including q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj, and lm_head. The α and r of the QLoRA method are both set to 16.

C.3 Prompting ChatGPT and GPT-4 for Three Tasks

As Step-2 of Sec. 2.2, we utilize two-shot prompts to instruct ChatGPT and GPT-4 in simultaneously generating symbolic and verbalized responses for three tasks, as shown in Table 13, 14, 15.

C.4 Evaluation Metrics

We detailed describe the metrics for evaluating our inference engine against ChatGPT and GPT-4 for the premise generation task.

- Accuracy: The premise is logically correct to infer the conclusion and follow the instruction regarding the specific number of facts.
- Diversity: The degree of variation among the three generated rules.
- Complexity: Assessed only for premise generation with more than 2 facts, measuring the fact number and the semantic difficulty.
- Abstractness: The variable types in premises are abstract to generalize to diverse instances. For example, the variable types " Region" and "Event" are abstrct while "New York" and "The FIFA World Cup" are specific entities with low abstractness.

C.5 LLM Evaluation Prompts

We prompt GPT-4 as the evaluator for rating the accuracy of conclusion generation and premise completion tasks, and the accuracy, diversity, complexity and abstractness of the premise generation task. We adopt one-shot prompts which are shown as Table 16 and Table 17 (with demonstrations omitted).

C.6 Human Evaluation Templates

For the human evaluation of premise generation accuracy, we qualify a new cohort of AMT annotators to pairwise compare two sets of generated premises in terms of logical consistency with the provided

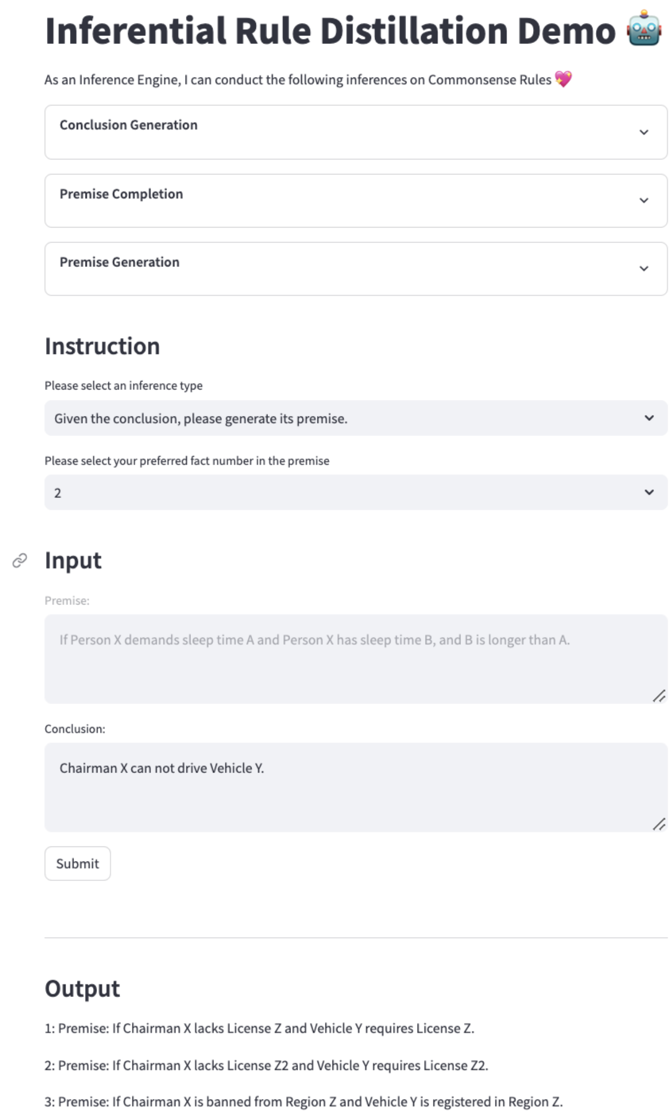


Figure 16: Inference Engine Demo.

conclusion, alignment with common sense and the inclusion of an accurate number of facts. The detailed template for human evaluation is shown as Figure 17.

C.7 Downstream Reasoning Datasets

StrategyQA and SOCIAL IQA consist of crowd-sourced questions involving reasoning of implicit logic. LINK comprises GPT-4 generated statements instantiated from abstract rules, including two subsets: head distribution statements and long-tail knowledge statements. PIQA examines operational commonsense for achieving physical goals and CSQA2.0 features adversarial commonsense examples designed to mislead AI systems.

Prompt for Premise Completion

Given the conclusion and a part of its premise, please complete the remaining portion of the premise in both Prolog and natural language.
The remaining premise in Prolog should have the same meaning with the remaining premise in natural language.
Each fact in the remaining premise should contain two variables, like X, Y, Z, Z1, Z2, A, B.

Examples:
Conclusion: Person X cannot use Furniture Y.
Partial Premise: If Person X is allergic to Material Z,
Remaining Premise:
[Prolog]: MadeFrom(Furniture Y, Material Z);
[Natural Language]: Furniture Y is made from Material Z.

Conclusion: Substance X can submerge in Substance Y.
Partial Premise: If Substance X has a Density Z1, the density of Substance Y is Density Z2,
Remaining Premise:
[Prolog]: BiggerThan(Density Z1, Density Z2);
[Natural Language]: Density Z1 is bigger than Density Z2.

Conclusion: {conclusion}
Partial Premise: {partial premise}
Remaining Premise:

Table 14: Prompt ChatGPT and GPT-4 for premise completion.

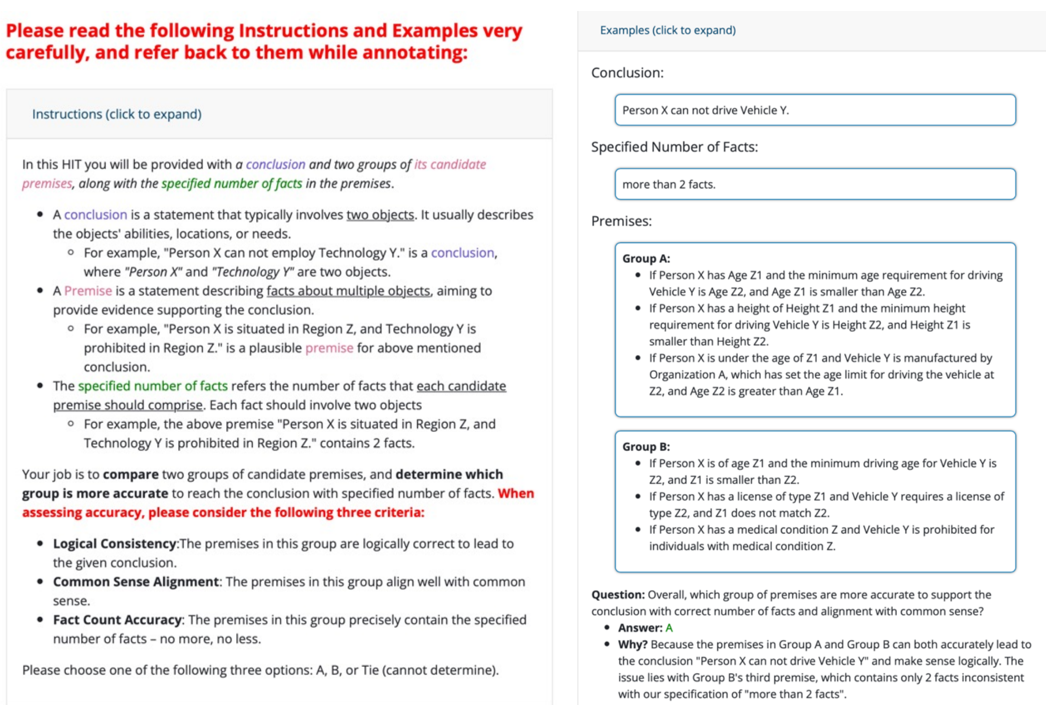


Figure 17: AMT template for human evaluation for premise generation accuracy.

Prompt for Premise Generation

Given the conclusion, please generate three different premises in both Prolog and natural language, ensuring that each Prolog premise conveys the same meaning as its natural language counterpart. Each premise should contain a specified number of facts, with each fact comprising only two variables, such as X, Y, Z, Z1, Z2, A, B.

Examples:
Fact number: 1 fact
Conclusion: Person X has Skill Y.
Three Premises:
1. [Prolog] Learned(Person X, Skill Y); [Natural Language] If Person X learned Skill Y.
2. [Prolog] Inherit(Person X, Skill Y); [Natural Language] If Person X inherits Skill Y.
3. [Prolog] Acquire(Person X, Skill Y); [Natural Language] If Person X acquires Skill Y.

Fact number: more than 2 facts
Conclusion: Person X cannot attend Event Y.
Three Premises:
1. [Prolog] Have(Person X, Age Z1), RequireMinimumAge(Event Y, Age Z2), BiggerThan(Age Z2, Age Z1); [Natural Language] If Person X has Age Z1 and the minimum age requirement for attending Event Y is Age Z2, Age Z2 is bigger than Age Z1.
2. [Prolog] Have(Person X, Height Z1), RequireAbove(Event Y, Height Z2), SmallerThan(Height Z1, Height Z2); [Natural Language] If Person X has a Height Z1, and Event Y requires a Height above Z2, and Height Z1 is smaller than Height Z2.
3. [Prolog] HaveCriminalRecord(Person X, Event Z), ProhibitedBy(Event Z, Legislation A), EnforcedIn(Legislation A, Region B), HeldIn(Event Y, Region B); [Natural Language] If Person X has a criminal record for Event Z and Event Z is prohibited by Legislation A, which is enforced in Region B, and Event Y is held in Region B.

Fact number: {fact num}
Conclusion: {conclusion}
Three Premises:

Table 15: Prompt ChatGPT and GPT-4 for premise generation.

Prompt for Rating the Accuracy of Conclusion Generation

You are a helpful scoring assistant.
Please read the provided premise carefully, and rate the accuracy of the candidate conclusion on a scale of 1 to 3:
- 1 (not accurate): The conclusion is clearly unsupported, irrelevant or contradictory to the provided premise.
- 2 (somewhat accurate): The conclusion, despite being supported by the premise, fails to state the definitive link between X and Y, or contradicts common sense, or lacks clarity.
- 3 (highly accurate): The conclusion correctly states the definitive link between X and Y, and is well-supported by the premise aligning with both established facts and common sense.

Please first output your rating based on your general knowledge and logical reasoning, and then provide a brief explaination with no more than 100 words.

[Provided Premise]: {premise}
[Candidate Conclusion]: {conclusion}
[Output]:

Prompt for Rating the Accuracy of Premise Completion

You are a helpful scoring assistant.
Please read the provided conclusion and its partial premise carefully, and rate the accuracy of its remaining premise in completing the provided premise to reach the conclusion, using a scale from 1 to 3:
- 1 (not accurate): The remaining premise fails to complete the provided premise for deducing the conclusion. It may be irrelevant or inconsistent with the provided premise or conclusion, or both.
- 2 (somewhat accurate): The remaining premise can somewhat supplement the provided premise but is not entirely sufficient for a conclusion inference. It may require additional information for comprehensive completion, or contradicts common sense, or lacks clarity.
- 3 (highly accurate): The remaining premise, combined with the provided partial premise, can correctly lead to the given conclusion, and also aligns well with common sense.

Please first output your rating based on your general knowledge and logical reasoning, and then provide a brief explaination with no more than 100 words.

[Conclusion]: {conclusion}
[Partial Premise]: {partial premise}
[Remaining Premise]: {rest premise}
[Output]:

Prompt for Rating the Accuracy of Premise Generation

You are a helpful scoring assistant.
Please carefully read the provided conclusion along with the specified number of facts, and rate the accuracy of candidate premise in both reaching the conclusion and containing the correct number of facts, using a scale from 1 to 3:
- 1 (not accurate): The premise is logically incorrect, irrelevant or contradictory for deducing the conclusion, or it contains an incorrect number of facts.
- 2 (somewhat accurate): The premise can partially infer the conclusion but is not entirely sufficient. It may require additional information, or contradicts common sense, or lacks clarity.
- 3 (highly accurate): The premise can correctly lead to the given conclusion and aligns well with common sense, and precisely contains the specified number of facts.

Please first output your rating based on your general knowledge and logical reasoning, and then provide a brief explaination with no more than 100 words.

[Fact Number]: {fact num}
[Conclusion]: {conclusion}
[Premise]: {premise}
[Output]:

Table 16: Prompts for rating the accuracy of three tasks.

Prompt for Rating the Diversity of Premise Generation

You are a helpful scoring assistant.
Please read the provided conclusion and multiple generated premises carefully, and rate the diversity of these premises using a scale from 1 to 3:
- 1 (low diversity): The premises show minimal variation, where all three premises largely repeat same perspectives with slight lexical changes.
- 2 (moderate diversity): The premises exhibit some degree of variation, with two out of the three premises sharing similar perspectives, expressions and fact numbers while the third presents different content.
- 3 (high diversity): The premises display a high level of diversity, where each premise presents distinct perspective from the others, or contains different fact numbers.

Please first output your rating, and then provide a brief explaination with no more than 50 words.

[Conclusion]: {conclusion}
[Premise]: {premise$_1$}, {premise$_2$}, {premise$_3$}
[Output]:

Prompt for Rating the Complexity of Premise Generation

You are a helpful scoring assistant.
Please carefully read the provided conclusion, and rate the complexity of candidate premise considering both the number of facts it comprises and its semantic difficulty, using a scale from 1 to 3:
- 1 (low complexity): The premise is straightforward, incorporating no more than 3 facts with clear and easy-to-understand semantics and a simple logical structure.
- 2 (moderate complexity): The premise exhibits moderate complexity, which involves 4 facts and somewhat intricate semantics and a logical structure that require some thought to understand.
- 3 (high complexity): The premise is highly complex with more than 4 facts, which also includes complex semantics and an abstract logical structure, demanding a high level of understanding.

Please first output your rating based on your general knowledge and logical reasoning, and then provide a brief explaination with no more than 50 words.

[Conclusion]: {conclusion}
[Premise]: {premise}
[Output]:

Prompt for Rating the Abstractness of Premise Generation

You are a helpful scoring assistant.
Please carefully read the provided conclusion, and rate the abstractness of objects in the candidate premise considering how broadly they can generalize to various specific instances, using a scale from 1 to 3:
- 1 (low abstractness): The objects in the premise are concrete and specific, making direct and clear reference to particular instances or examples, which focus on specific people, places, or tangible entities, such as Swimmer, New York, or SUV.
- 2 (moderate abstractness): The objects in the premise are somewhat abstract, representing a balance between specific instances and general concepts. They may pertain to fine-grained categories of people, places, or things, such as Professionals, City, or Car.
- 3 (high abstractness): The objects in the premise are highly abstract, focusing on coarse-grained people, places or things that are far removed from concrete instances, such as Person, Region, or Event, or general properties like Age and Height.

Please first output your rating based on your general knowledge and logical reasoning, and then provide a brief explaination with no more than 50 words.

[Conclusion]: {conclusion}
[Premise]: {premise}
[Output]:

Table 17: Prompts for rating the diversity, complexity and abstractness of premise generation.