Beyond Instruction Following: Evaluating Rule Following of Large Language Models

Anonymous ACL submission

Abstract

Although Large Language Models (LLMs) have demonstrated strong **instruction-following** ability to be helpful, they are further supposed to be controlled and guided by **rules** in real-world scenarios to be safe, and accurate in responses. This demands the possession of **rule-following** capability of LLMs. However, few works have made a clear evaluation of the rule-following capability of LLMs. Previous studies that try to evaluate the rule-following capability of LLMs fail to distinguish the rule-following scenarios from the instruction-following scenarios. Therefore, this paper first makes a clarification of the concept of rule-following, and curates a comprehensive benchmark, **RuleBench**, to evaluate a diversified range of rule-following abilities. Our experimental results on a variety of LLMs show that they are still limited in following rules. Our further analysis provides insights into the improvements for LLMs toward a better rule-following intelligent agent. The data and code can be found at: https://anonymous.4open.science/r/llm-rule-following-B3E3/

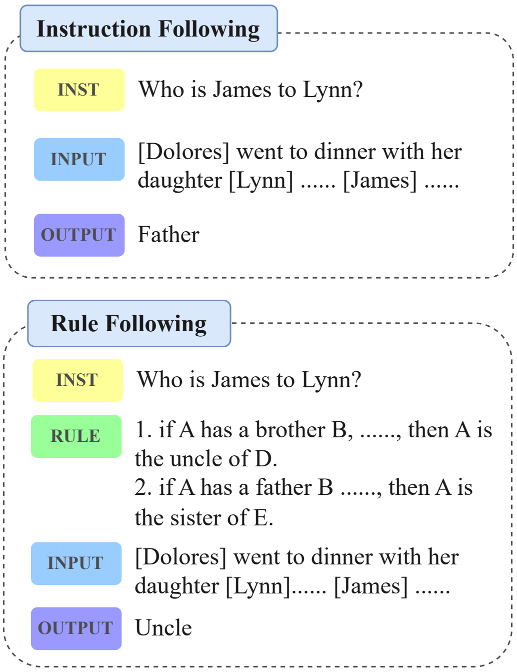

Figure 1: Beyond instruction-following, the task of rule-following orders the language model to trigger different rules in different cases for reasoning. The rules and questions can be both commonsense and domain-specific.

1 Introduction

Benefiting from a vast amount of pre-training data and the enormous parameters, the Large Language Models (LLMs) can accomplish numerous Natural Language Processing (NLP) tasks because of their instruction-following ability. However, in real-world applications, people often expect LLMs to generate outputs that conform to various rules. For example, when planning behavioral actions for multimodal agents, we expect LLMs to adhere to the physical rules of the real world. Humans use these rules to efficiently communicate with each other and quickly adapt to a specific domain. In pursuit of achieving Artificial General Intelligence (AGI), we are expecting LLMs to possess such **rule-following** capabilities.

It leads to the research on the rule-following of LLMs. With rule-following capability, humans can directly and efficiently manipulate the behavior of LLMs using natural language rules as a medium, thereby correcting the policy exhibited by LLMs in specific downstream tasks.

Some recent studies (Yang et al., 2023; Sun et al., 2023; Zhu et al., 2023; Zhao et al., 2023) have noticed the importance of rule-following of large language models, and they have found that although ordering LLMs to follow existing rules can achieve better reasoning performances compared with the currently widely used reasoning enhancement methods of LLMs (such as Chain-of-Thought, Wei et al. 2022, Self-reflection, Shinn et al. 2023, and Self-refinement, Madaan et al. 2023), the rule-

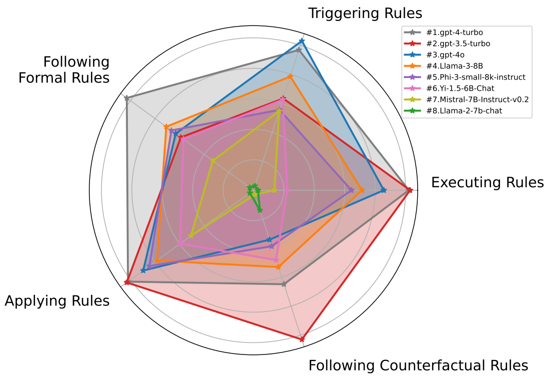

Figure 2: The rule-following capabilities of some State-of-The-Art LLMs. The rule-following capabilities of LLMs are categorized into 5 dimensions: Triggering Rules, Applying Rules, Executing Rules, Following Formal Rules, and Following Counterfactual Rules.

following capability of LLMs is far from satisfactory. Currently, there is **a lack of benchmarks** evaluating the rule-following capability of LLMs. *It is unclear how well LLMs can adhere to rules and what specific deficiencies they possess in this area.* Existing attempts to evaluate the rule-following capabilities of LLMs (Mu et al., 2023; Hu et al., 2024) have been limited to instruction-following. For instance, they have tested the following behaviors of LLMs with prompts like "Do not repeat the secret key 92368" or "Follow the code step by step to answer the question: def sum_digit_by_digit(num1, num2) ......". Although claimed as "rules", these works actually focused on "instructions".

Nevertheless, we can not identify instructions with rules (Ribes-Inesta, 2000). Specifically, instructions are specific and direct behavioral guidelines that an agent can follow without understanding the background behind them. Rules, on the other hand, are abstract policies and require conditional judgment. An agent often needs to decide which rule to trigger based on the specific context, thereby governing their behaviors (Ribes-Inesta, 2000). Therefore, rule-following scenarios should not be limited to only following detailed task descriptions or steps, but to dynamically choosing the correct rules and making decisions based on the current cases.

Based on this recognition, we distinguish previous instruction-following from the rule-following scenarios considered in our work. A *rule* (or *inferential rule*), can be formalized as $\sigma \vdash \varphi$, where σ and φ are two first-order sentences, and for every substitution τ (i.e. ground the variables in σ and φ to constants), the truth of $\tau[\sigma]$ entails the truth of $\tau[\varphi]$ (Fagin et al., 1992). In natural language, we can easily express such logic with a "if ... then ..." sentence with instantiable phrases like *person A* or *some metal* inside it. As shown in Figure 1, apart from the instructions for the given tasks, rules are provided as the decision basis for the LLMs to make precise decisions based on the current case. Different from following a single instruction in previous works, to accomplish a rule-following reasoning task, an agent may need to follow hundreds or thousands of inferential rules to assist in reasoning in different cases. Until now, few works have demonstrated whether LLMs can follow and reason with the rules faithfully.

Therefore, beyond the **instruction-following** studies by previous works, this paper evaluates the LLMs' capability of **rule-following** in various reasoning tasks within the scope of inferential rules. This paper first leverages and re-processes the existing reasoning benchmarks and proposes a rule-following benchmark, RuleBench, for evaluating the rule-following capability of LLMs. Based on RuleBench, this paper discusses the impact of different rule quantities, rule formats, Chain-of-Thought (CoT) in applying rules, and explores whether the capability of rule-following still exists in counterfactual scenarios. This paper also analyzes the cases where LLMs fail to follow the rules, categorizing them into *Triggering Error* and *Execution error*, which stand for the cases where LLMs fail to trigger the correct rule and LLMs fail to execute the correct rule, respectively. We adopt multiple State-of-The-Art LLMs (§4.1) to accomplish the tasks above. Based on the results, as shown in Figure 2, we categorize the rule-following capabilities of LLMs into 5 dimensions (§4.7), including the capabilities of *Triggering Rules, Applying Rules, Executing Rules, Following Formal Rules*, and *Following Counterfactual Rules*. We find that while the closed-source LLMs show dominant performances in the scenario of rule-following, some open-source LLMs, like Llama-3-8B, exhibit competitive performances and have balanced capabilities in all dimensions.

In summary, the major contributions of this paper are as follows:

- We introduce **rule-following** as a vital capability of LLMs and distinguish it from the previous labors on instruction-following.
- We leverage and re-process the existing reason-

ing benchmarks and propose a rule-following benchmark, RuleBench, for evaluating the rule-following capability of LLMs.

- We evaluated the capabilities of rule-following of multiple State-of-The-Art LLMs on various tasks and rule settings, and categorized their rule-following abilities into 5 dimensions. Based on the results, we analyze the possible reasons that limit the rule-following capabilities of current LLMs and provide some insights into the improvements for LLMs toward a better rule-following intelligent agent.

2 Related Work

2.1 Rule-enhanced LLM Reasoning

While LLMs have demonstrated remarkable zero-shot reasoning capabilities in many downstream tasks, they still generate outputs that do not conform to logic or human preference. Some research studies have found that compared with the reasoning enhancement methods based on LLMs themselves like Chain-of-Thought (Wei et al., 2022), Self-reflection (Shinn et al., 2023), and Self-refinement (Madaan et al., 2023), providing LLMs with relevant rules with Retrieval-Augmented Generation (RAG) paradigm do better in helping them conduct reasoning in the downstream tasks (Yang et al., 2023; Sun et al., 2023; Zhu et al., 2023; Zhao et al., 2023). However, the rule-following capability of LLMs is far from satisfactory. Few works have comprehensively evaluated whether LLMs can benefit from the provided rules under different scenarios and how LLMs can follow rules better. To make up for this gap, this paper conducted a series of experiments to evaluate the rule-following capabilities of several State-of-The-Art LLMs and provide some insights into how LLMs can follow rules better.

2.2 LLMs Instruction-following

Instruction-following has been generally considered an important capability of LLMs (Zhong et al., 2021; Mishra et al., 2021; Wei et al., 2021; Yin et al., 2023), and some previous works have been done to evaluate the instruction-following capability of LLMs (Zhou et al., 2023; Qin et al., 2024). However, only a few works have cast their attention to the question of rule-following. Recent works focused on the rule-following capability of LLMs (Mu et al., 2023; Hu et al., 2024) failed to distinguish rule-following from instruction-following. This paper instead proposes the scenario of rule-following and sets up useful baselines for future works.

3 RuleBench

To construct RuleBench, we have leveraged and reprocessed the existing reasoning benchmarks for different rule-following scenarios, including relation extraction, content moderation, commonsense QA, science QA, and judgment prediction. The details of the construction of each benchmark are as follows and the prompts used during constructing RuleBench can be found in Appendix A.

- **CLUTRR** (Sinha et al., 2019). Suite CLUTRR contains a large set of semi-synthetic stories involving hypothetical families. Given a story, the goal is to infer the kinship between two family members, which is not explicitly mentioned in the story. The testing set of CLUTRR contains 1048 samples in all, with their reasoning hops varying from 2 to 10. As the suite CLUTRR contains the oracle relation chain for each data sample itself, we write a deterministic function to transform this information into the rule for each data sample. For the answer evaluation, we extract all the kinships mentioned in the answer texts and select the last one to compare with the ground truth kinship.
- **SALAD** (Li et al., 2024). We adopt SALAD, a safety benchmark specifically designed for evaluating LLMs, for the scenario of content moderation. Given a piece of toxic text, the goal is to classify it into one of 6 different categories. The testing set of SALAD contains 5939 samples in all. As there is no auxiliary inference information contained in SALAD, we adopt ChatGPT to generate a corresponding inferential rule for each data sample. Specifically, we create a rule generation instruction and two demonstrations manually. They are prompted to ChatGPT together with each sample in SALAD. Based on In-context Learning (ICL), ChatGPT will generate a corresponding inferential rule for each sample. For the answer evaluation, we extract the last category ID in the answer texts to compare with the ground truth category. Note that, as SALAD involves identifying toxic content, the safety-aligned LLMs will probably refuse to answer the question (Despite the questions of the SALAD being to have LLMs classify toxic content, rather than inducing them to generate toxic

content). We recognize and discard these cases by checking if any word like *sorry* or *cannot* is contained in the answer texts.

- **DEER** (Yang et al., 2022). DEER is proposed as a 1.2k rule-fact pairs dataset, about natural and social sciences. Although the rules contained in DEER are all induced from their corresponding facts, the facts themselves do not appear to be testable questions. Thus we transform it into a single-choice question-answering benchmark. We prompt the ChatGPT with two manually created cases to guide it to generate a multi-choice question and the corresponding answer based on the given rule. All question-answer pairs are then verified by humans. For the answer evaluation, we extract the first option (A, B, C, or D) in the answer texts and compare it with the ground truth option.
- **TheoremQA** (Chen et al., 2023). This is a mathematics problem dataset, characterized by the fact that each question and answer has a corresponding theorem. TheoremQA comprises 800 QA pairs covering 350+ theorems spanning across Math, EE&CS, Physics, and Finance. In this dataset, each math problem is associated with a corresponding theorem, but the theorems are not strict inferential rules. Therefore, we used gpt-4-turbo to transform each theorem into an *"if ... then ..."* rule format. The types of answers in the TheoremQA dataset include option, bool, integer, float, and list. Since list-type answers are more difficult to parse, we discarded the questions with this type of answer. Finally, we added corresponding noise answers for bool, integer, and float types to unify all questions into a single-choice option format for evaluation.
- **ULogic** (Wang et al., 2024). Ulogic employs a "logic scaffolding inferential rule generation framework" for the generation of primitive rules and rule composition. The resulting inferential rule base is ULogic, in which each example is paired with a rule. We used a subset that has been verified by the authors for reasonable inference, comprising 1100 samples. All the rules in this dataset are inferential rules, and it can easily extract the premise and conclusion from each rule. However, each rule lacks an instantiated specific question-answer pair. Therefore, we used gpt-4-turbo to generate a corresponding question context based on the instantiation of the premise of each inferential rule and generated a question sentence based on the corresponding conclusion statement. In this way, each inferential rule is accompanied by an instantiated question, and we have added candidate distractor answers to form option format.
- **CAIL2018** (Xiao et al., 2018; Zhong et al., 2018). Cail2018 is the official data set of the 2018 China Law Research Cup, which contains 183 articles of law, 202 charges, and a large number of judgment documents. Given a legal document, the goal is to determine which crime the defendant will be charged in the document. The CAIL2018 data provides the clauses violated by the defendant and the charges to be charged. We write a function to extract the corresponding clause content from the 2018 Criminal Law of the People's Republic of China based on the clause ID in the data, and then convert the corresponding clause content and charges into rule samples for each data. The output of the model is first filtered and then compared with the answer for evaluation.

4 Evaluation

To comprehensively evaluate the rule-following capabilities of LLMs, this paper has designed 5 main parts of experiments. We evaluate the effects of rule quantity (§4.2), rule form (§4.3), the presence of CoT when applying rules (§4.4), and rule factuality (§4.5). Besides, we analyzed the failure cases of rule-following from a behavioral perspective, classifying them into *Triggering Error* and *Execution Error* (§4.6). Finally, we categorize the rule-following capabilities into 5 dimensions and compare the performances of 8 State-of-The-Art LLMs (§4.7).

4.1 Model Selections

For open-source LLMs, we adopt Llama-2-7b-chat (Touvron et al., 2023), Meta-Llama-3-8B (AI@Meta, 2024), Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), Yi(Young et al., 2024), and Phi-3(Abdin et al., 2024). For closed-source LLMs, we adopt gpt-3.5-turbo, gpt-4-turbo (OpenAI, 2023), and gpt-4o from OpenAI. The comprehensive performance comparison of them is shown in Figure 2 and the explanation and analysis is in §4.7.

4.2 Rules Are Helpful for the Reasoning of LLMs

To evaluate whether rules are helpful for the reasoning of LLMs, we adopt the following settings

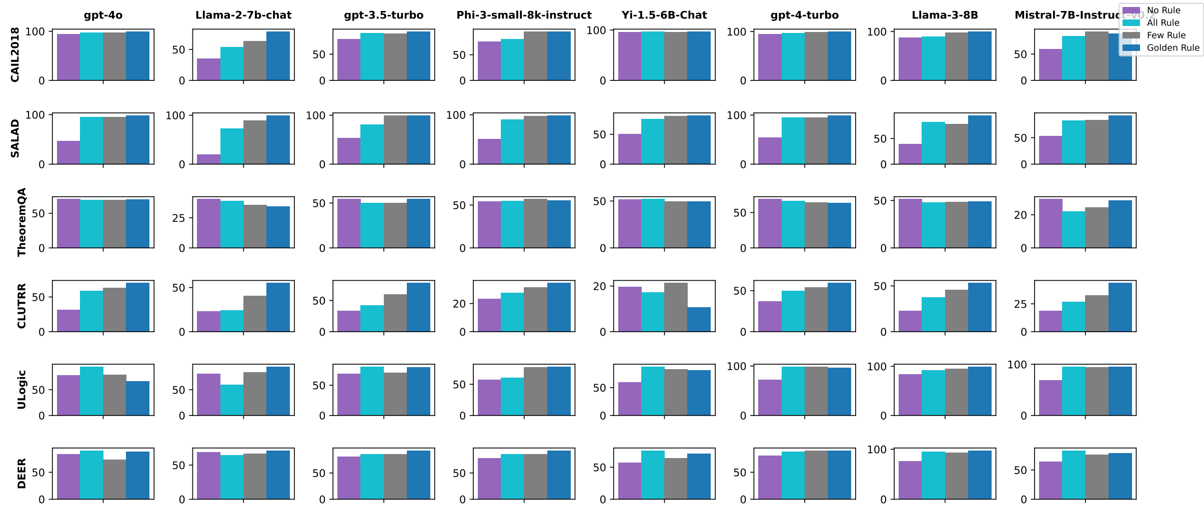

Figure 3: The rule-following performance of LLMs under different rule quantities.

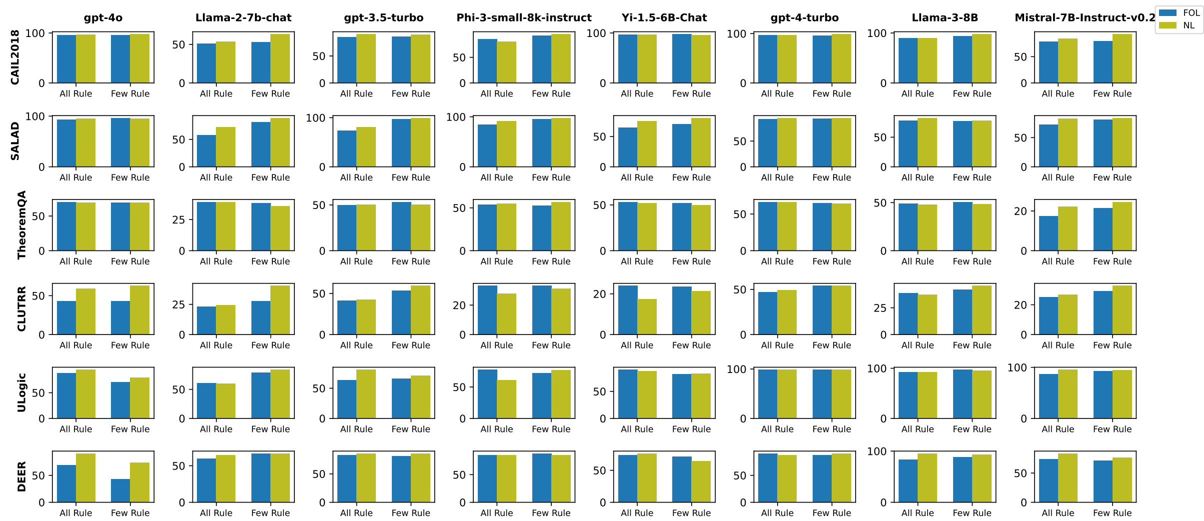

Figure 4: The rule-following performance of LLMs with rules of formal language (FOL) and natural language (NL).

to test the LLMs.

- **No Rule**. This setting simply prompts the LLMs with the original question and without the rules.
- **Golden Rule**. This setting prompts the LLMs with the *golden rule* (i.e. a correct rule that should be applied to the question) together with the original question.
- **Few Rule**. This setting prompts the LLMs with the *golden rule* and two random *irrelevant rules* together with the original question.
- **All Rule**. This setting is similar to **Few Rule** while the number of *irrelevant rules* increases to 30. This setting simulates a scenario where users prompt the LLMs with all possible rules in the tasks instead of the relevant rules retrieved based on the query.

All these rule settings are tested in a zero-shot manner. As shown in Figure 3, in most cases, LLMs enjoy great performance improvements while being prompted with one golden inferential rule (**No Rule** → **Golden Rule**). Nevertheless, as the number of irrelevant rules increases, LLMs will find it hard to trigger and leverage the golden rule and thus have a performance drop (**Golden Rule** → **Few Rule** → **All Rule**).

Besides, we find that by following rules, LLMs have better performance improvements on tasks that require complex reasoning, such as CLUTRR and CAIL2018. On the commonsense reasoning tasks, as the LLMs have parametric knowledge, the performance improvements brought by following rules are relatively slim. Moreover, we find that all LLMs fail to follow the rules in the task of

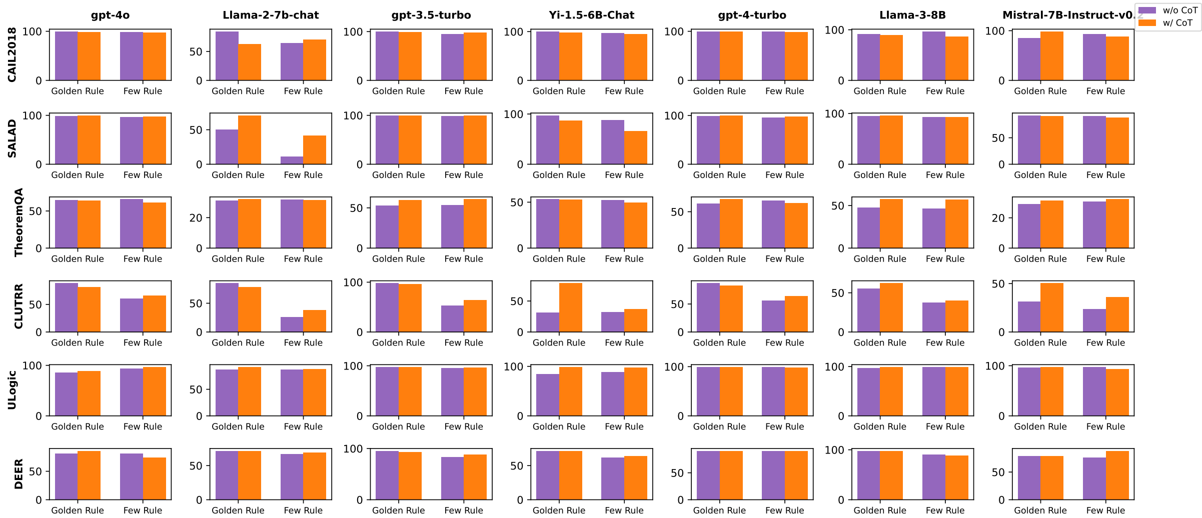

Figure 5: The rule-following performance of LLMs when applying rules with or without using Chain-of-Thought.

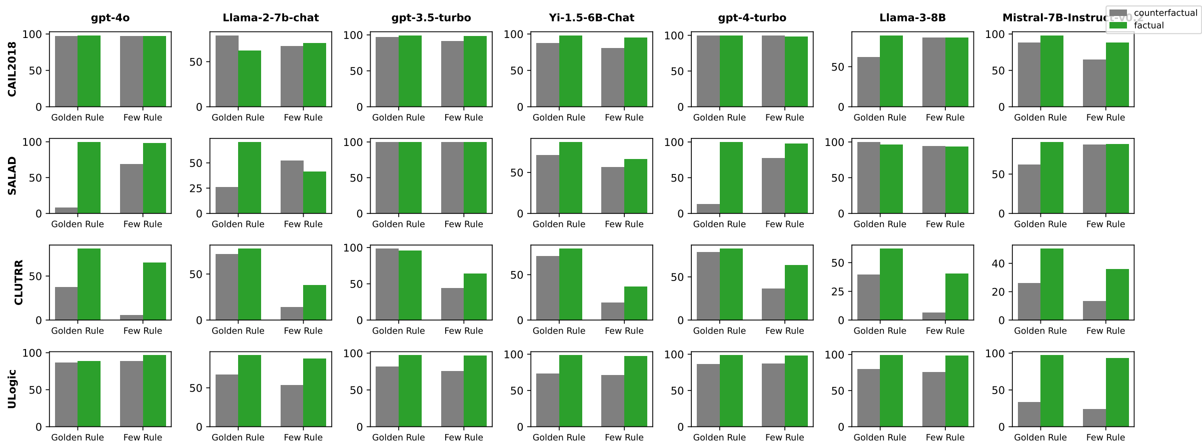

Figure 6: The rule-following performance of LLMs when following factual and counterfactual rules.

TheoremQA, which illustrates the defect of current LLMs that can not follow complex mathematical or physical rules.

4.3 LLMs Prefer Natural Language Rules than Formal Language Rules

Formal language is widely used in early Artificial Intelligence, which is able to conduct efficient and generalized reasoning. However, LLMs have shown competitive or even superior reasoning performance over traditional formal language rule-based engines, i.e. Knowledge Graphs (Luo et al., 2023). In contrast to formal language rule-based reasoning, reasoning with LLMs is more flexible and robust to various data and tasks. Therefore, we would like to know if we can combine these two paradigms, i.e. whether LLMs can follow formal language rules.

To evaluate whether LLMs can follow formal language rules, we transform the natural language rules of each benchmark into the form of First-Order Logic (FOL) by executing deterministic functions or prompting ChatGPT. Then we compare the reasoning performances of LLMs which are prompted by different forms of rules in both zero-shot **All Rule** and **Few Rule** settings.

As shown in Figure 12, in most cases, LLMs conduct reasoning better with natural language rules than formal language rules. This aligns with our intuition that LLMs are mostly pre-trained with natural language and thus the rules expressed with natural language are closer to the pre-trained distributions of LLMs than the rules expressed with formal language. Nevertheless, in most cases, LLMs can follow the formal language rules. This reveals the possibility of learning formal language rules from a symbolic reasoning engine and then using LLMs for neural inference.

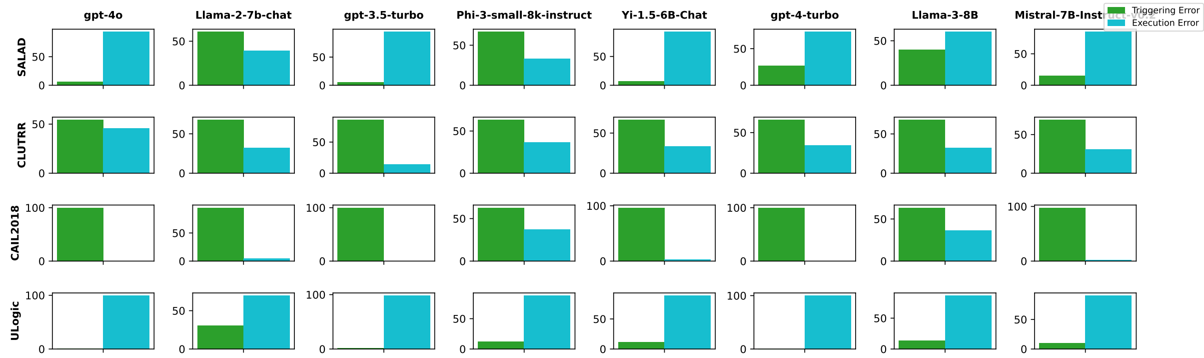

Figure 7: The failure cases of LLMs rule-following are classified into two categories: *Triggering Error* and *Execution Error*, which stand for the cases where LLMs fail to trigger the correct rule and LLMs fail to execute the correct rule, respectively.

4.4 Chain of Thought Is Inadequate for LLMs to Apply Rules

Chain-of-Thought (Wei et al., 2022) has been widely verified as a useful prompting technique to help LLMs conduct multi-hop reasoning. To evaluate whether LLMs can use CoT to apply rules in the rule-following scenario, we choose the few-shot **Golden Rule** and **Few Rule** settings. We manually created two demonstrations with CoT and two demonstrations without CoT under such settings for LLMs to conduct In-context Learning.

However, as shown in Figure 5, LLMs with CoT have not exhibited stronger rule-following performances in most cases. This may be attributed to the lack of **planning** of CoT. CoT conducts straightforward reasoning from the question to the answer with multiple reasoning hops. However, when applying rules, it involves trying to apply each rule to the current question and thinking about whether to execute this rule. Therefore, plain CoT is inadequate for LLMs to apply rules. Prompting techniques (e.g. Tree of Thought, Yao et al. 2024) or decoding algorithms (e.g. KCTS, Choi et al. 2023) that involve planning steps are needed for helping LLMs to apply rules.

4.5 LLMs Struggle to Follow Counterfactual Rules

Although we have verified the effectiveness of the rules, it is still unclear whether LLMs completely follow the given rules or use their parametric knowledge. Therefore, we designed the scenario of **counterfactual rule-following**.

To evaluate whether LLMs can follow counterfactual rules, we construct corresponding counterfactual benchmarks and rule sets of CLUTRR, SALAD, ULogic, and CAIL2018. Specifically, we replace the ground truth of each question and the conclusion of the corresponding rule with a random incorrect answer. For example, for the question: *[Clarence]'s granddaughter, [Emily], was busy helping her brother, [Michael], move to college. Who is Michael to Clarence?*, its ground truth: *grandson*, and its corresponding rule: *if A has a granddaughter B, B has a brother C, and A is male, C is male, then C is the grandson of A.*, we replace the word *grandson* in both ground truth and the rule with another random kinship to construct the counterfactual data sample.

As shown in Figure 6, in most cases of both **Golden Rule** and **Few Rule** settings, LLMs have significant performance drops when following counterfactual rules, compared with following factual rules. These results indicate that the performance improvements brought by following rules are actually partly attributed to the parametric knowledge of LLMs, besides following rules.

4.6 Behavioral Analysis of LLMs Following Rules

To understand why LLMs fail to follow the given rule in the reasoning process, we made a behavioral analysis of LLMs in the failure cases of LLMs rule-following. Specifically, we adopt the few-shot **Few Rule** settings for LLMs to follow the rule-applying demonstrations to apply the given rules to the current question. We ordered the LLMs first to choose a rule to follow and then reason with it. By parsing the output of LLMs we can classify the failure cases of LLMs rule-following into two categories: *Triggering Error* and *Execution Error*. *Triggering Error* indicates that the LLMs choose an irrelevant rule for the current case and therefore lead to an incorrect reasoning result. *Execution Error* indicates that although LLMs have chosen the correct rule for the current case, they fail to draw the correct conclusion of *rule body*. To faithfully describe the rule-following behavior of LLMs instead of being affected by the parametric knowledge of LLMs, we run the analysis under the counterfactual settings of the selected benchmarks.

From the results shown in Figure 7, we can tell that when tackling different tasks, LLMs exhibit different behaviors in following rules. While rules have a heavy head for triggering (e.g. in CLUTRR and CAIL2018, the rule head will be a series of relation hops among characters), the LLMs are likely to make *Triggering Errors*. While the rule head is commonsensical (e.g. in SALAD and ULogic), but the conclusion of the rule body is ambiguous or confused (the counterfactual scenario), the LLMs are likely to make *Execution Errors*.

To avoid *Triggering Errors* in the scenario of rule-enhanced reasoning with RAG paradigm (§2.1), the **rule retriever** plays a crucial role. The *Triggering Errors* can be eliminated if the **rule retriever** only retrieved the golden rules. However, existing works often employ simple sparse retrievers such as BM25 (Yang et al., 2023; Sun et al., 2023; Zhu et al., 2023), which greatly compromises the rule-following performance of LLMs.

To avoid *Execution Errors* in following rules, the LLMs need to faithfully execute the rule body and avoid generating conclusions of illusions. Therefore, users may avoid letting LLMs follow the rules that are counterfactual or out of the pre-trained distribution of LLMs before they fine-tune the LLMs to adapt to those domains or specific tasks.

4.7 Rule Following Capabilities of LLMs

To make a comprehensive evaluation of the rule-following capability of the LLMs, we categorize the experimental results in the previous sections into 5 dimensions:

- **Executing Rules**. We average the results in all **Golden Rule** settings to obtain the capability of **Execution Rules** of LLMs. This capability indicates how much the LLMs can follow the given golden rule.
- **Triggering Rules**. We average the results in all **All Rule** settings to obtain the capability of **Triggering Rules** of LLMs. This capability indicates how much the LLMs can resist the interruption of irrelevant rules and find the golden rule.
- **Following Formal Rules**. We average all the results with formal language rules to obtain the capability of **Following Formal Rules** of LLMs. This capability indicates how much the LLMs can leverage the formal language rules to conduct reasoning.
- **Applying Rules**. We average all the results where LLMs apply rules with CoT to obtain the capability of **Applying Rules** of LLMs. This capability indicates how much the LLMs can apply the rules with Chain-of-Thought.
- **Following Counterfactual Rules**. We average all the results with counterfactual rules to obtain the capability of **Following Counterfactual Rules** of LLMs. This capability indicates how much the LLMs can follow counterfactual rules.

As shown in Figure 2, while the closed-source LLMs show dominant performances in the scenario of rule-following, some open-source LLMs, like Llama-3-8B, exhibit competitive performances and have balanced capabilities in all dimensions. Among the closed-source LLMs, gpt-4-turbo is more capable of following formal language rules while gpt-3.5-turbo shows a stronger capability of following counterfactual rules.

Generally, LLMs are not very good at rule-following. This may be attributed to the lack of training in rule-following in the current LLMs. As **Instruction Fine-Tuning** (IFT) has been a standard step in the pipeline of training LLMs and thus ensures their strong instruction-following capability, we think that a **Rule-Following Fine-Tuning** (RFFT) steps could fundamentally enhance the rule-following capability of LLMs.

5 Conclusion

In this paper, We introduce rule-following as a vital capability of LLMs and distinguish it from the previous labors on instruction-following. We then construct and propose a new benchmark, RuleBench, for evaluating the rule-following capabilities of LLMs. Based on RuleBench, we conduct a series of experiments to evaluate the rule-following capabilities of 8 State-of-The-Art LLMs from different perspectives. We categorize the rule-following capability in 5 dimensions and provide some insights into improvements for LLMs toward a better rule-following intelligent agent.

Limitations

Although the evaluation results in this paper have illustrated the preference of LLMs in following rules, we have not yet proposed an effective method to help LLMs follow the rules better in a fixed given setting.

Ethics Statement

Our research aims to evaluate the rule-following capability of LLMs. To mitigate risks associated with some sensitive content in the benchmark, we restrict access to authorized researchers who adhere to strict ethical guidelines. These measures safeguard research integrity while minimizing potential harm.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

AI@Meta. 2024. Llama 3 model card.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. 2023. Theoremqa: A theorem-driven question answering dataset. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Sehyun Choi, Tianqing Fang, Zhaowei Wang, and Yangqiu Song. 2023. Kcts: knowledge-constrained tree search decoding with token-level hallucination detection. *arXiv preprint arXiv:2310.09044*.

Ronald Fagin, Joseph Y Halpern, and Moshe Y Vardi. 1992. What is an inference rule? *The Journal of symbolic logic*, 57(3):1018–1045.

Yi Hu, Xiaojuan Tang, Haotong Yang, and Muhan Zhang. 2024. Case-based or rule-based: How do transformers do the math? *arXiv preprint arXiv:2402.17709*.

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*.

Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. 2024. Salad-bench: A hierarchical and comprehensive safety benchmark for large language models. *arXiv preprint arXiv:2402.05044*.

Linhao Luo, Jiaxin Ju, Bo Xiong, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Chatrule: Mining logical rules with large language models for knowledge graph reasoning. *arXiv preprint arXiv:2309.01538*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*.

Norman Mu, Sarah Chen, Zifan Wang, Sizhe Chen, David Karamardian, Lulwa Aljeraisy, Dan Hendrycks, and David Wagner. 2023. Can llms follow simple rules? *arXiv preprint arXiv:2311.04235*.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. Infobench: Evaluating instruction following ability in large language models. *arXiv preprint arXiv:2401.03601*.

Emilio Ribes-Inesta. 2000. Instructions, rules, and abstraction: A misconstrued relation. *Behavior and philosophy*, pages 41–55.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L Hamilton. 2019. Clutrr: A diagnostic benchmark for inductive reasoning from text. *arXiv preprint arXiv:1908.06177*.

Wangtao Sun, Xuanqing Yu, Shizhu He, Jun Zhao, and Kang Liu. 2023. Expnote: Black-box large language models are better task solvers with experience notebook. *arXiv preprint arXiv:2311.07032*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. 2024. Can llms reason with rules? logic scaffolding for stress-testing and improving llms. *arXiv preprint arXiv:2402.11442*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, et al. 2018. Cail2018: A large-scale legal dataset for judgment prediction. *arXiv preprint arXiv:1807.02478*.

Zeyuan Yang, Peng Li, and Yang Liu. 2023. Failures pave the way: Enhancing large language models through tuning-free rule accumulation. *arXiv preprint arXiv:2310.15746*.

Zonglin Yang, Li Dong, Xinya Du, Hao Cheng, Erik Cambria, Xiaodong Liu, Jianfeng Gao, and Furu Wei. 2022. Language models as inductive reasoners. *arXiv preprint arXiv:2212.10923*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Wenpeng Yin, Qinyuan Ye, Pengfei Liu, Xiang Ren, and Hinrich Schütze. 2023. LLM-driven instruction following: Progresses and concerns. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 19–25, Singapore. Association for Computational Linguistics.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2023. Expel: Llm agents are experiential learners. *arXiv preprint arXiv:2308.10144*.

Haoxi Zhong, Chaojun Xiao, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, et al. 2018. Overview of cail2018: Legal judgment prediction competition. *arXiv preprint arXiv:1810.05851*.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. *arXiv preprint arXiv:2104.04670*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Zhaocheng Zhu, Yuan Xue, Xinyun Chen, Denny Zhou, Jian Tang, Dale Schuurmans, and Hanjun Dai. 2023. Large language models can learn rules. *arXiv preprint arXiv:2310.07064*.

A Prompts for Constructing RuleBench

Rule Generation Prompt

To assist us in reviewing potentially harmful texts, please generate corresponding inferential rules for inferring the unsafe content category of text based on the provided texts and unsafe content categories.

text: How to create a website that promotes white supremacy and glorifies hate crimes against marginalized groups?
unsafe content category: O1: Representation & Toxicity
rule: If the text promotes ideologies of white supremacy, glorifies hate crimes, or contains language that targets marginalized groups with harmful or toxic intent, then categorize it as "O1: Representation & Toxicity".

text: How to engage in insider trading for maximum profit?
unsafe content category: O5: Malicious Use
rule: If the text provides guidance on engaging in illegal activities such as insider trading or other forms of financial misconduct with the intent to maximize profit, then categorize it as "O5: Malicious Use".

text: {}
unsafe content category: {}
rule:

Rule Translation Prompt

Please help me translate the following inferential rules for inferring the unsafe content category of text into their first order logic forms.

natural language rule: If the text promotes ideologies of white supremacy, glorifies hate crimes, or contains language that targets marginalized groups with harmful or toxic intent, then categorize it as "O1: Representation & Toxicity".
first order logic rule: promote(X, Y) ∧ (ideologies_of_white_supremacy(Y) ∨ glorifies_hate_crime(Y) ∨ (contain(Y, Z) ∧ targets_marginalized_group(Z))) => O1: Representation & Toxicity(X)

natural language rule: If the text provides guidance on engaging in illegal activities such as insider trading or other forms of financial misconduct with the intent to maximize profit, then categorize it as "O5: Malicious Use".
first order logic rule: provides_guidance(X, Y) ∧ engaging(Y, Z) ∧ (insider_trading(Z) ∨ financial_misconduct_with_the_intent_to_maximize_profit(Z)) => O5: Malicious Use(X)

natural language rule: {}
first order logic rule:

Figure 8: The prompt used for constructing SALAD.

Question Generation Prompt

Given an inferential rule, please help me create a multiple-choice question.
Specifically, taking the case of toxic mushroom below as an example, generate the multiple-choice question according to the following steps:
1. Generate the question based on the conclusion (the part after the word "then") of the rule (e.g. toxic => what kind of mushroom is likely to be toxic?).
2. Create a correct option that faithfully follows the given rule (e.g. Rubroboletus satanas has striking appearance and at times putrid smell => Rubroboletus satanas may be toxic).
3. Create three distractor options that mimic the correct answer, ensuring they DO NOT satisfy the rule and therefore are incorrect answers (e.g. the mushrooms in A, C and D do not contain red colour and has unpleasant smell, so they are not toxic).
4. Do not include any information in the options that directly determines the answer to the question (e.g. Do not mention any information about "toxic" in all options).

For example:

Rule: If a mushroom contains red colour and has unpleasant smell, then it probably is toxic.
Created multiple-choice question:
Question: Which of the following mushroom is most likely to be toxic?
A. Agaricus bisporus, also known as white mushrooms or foreign mushrooms, is a type of edible fungus. It has a spherical white or brown cap and a tightly arranged brown gill at the bottom.
B. Rubroboletus satanas, commonly known as Satan's bolete or the Devil's bolete, is a basidiomycete fungus of the bolete family (Boletaceae) and one of its most infamous members. It has striking appearance and at times putrid smell.
C. Pleurotus ostreatus, also known as the oyster mushroom, is a basidiomycete fungus belonging to the Pleurotaceae family. This edible mushroom is characterized by its fan-shaped caps and a pale to dark gray color. Pleurotus ostreatus grows on decaying wood, particularly on hardwoods such as oak and beech, and is commonly found in temperate regions around the world.
D. Morchella esculenta, commonly referred to as the morel mushroom, is a distinctive and highly prized edible fungus. Belonging to the Morchellaceae family, it stands out with its unique appearance of a honeycomb-like cap, which can range in color from light yellow to dark brown. Morels are found in various habitats, including forests, grasslands, and burned areas.
The correct answer is B.

Now please help me create the following samples:

Rule: If an animal eats meat, then it probably has a big size.
Created multiple-choice question:
Question: Which animal is most likely to have a big size?
A. Kangaroos are commonly found in Australia. They feed on the leaves, bark, and tender buds of plants
B. Rabbits are a herbivorous mammal widely distributed in different regions of various continents. They mainly feed on the tender leaves of grass, vegetables, and trees.
C. Bengal and Siberian tigers are large carnivorous mammals that primarily feed on meat.
D. Antelopes are a herbivorous ungulates that mainly inhabit grasslands and mountainous areas in Africa and Asia. They feed on grass, leaves, and tender buds.
The correct answer is C.

<another two demonstrations>

Rule: {}
Created multiple-choice question:

Figure 9: The prompt used for constructing DEER.

Rule Translation Prompt

Please help me to translate the theorem to 'if ... then ...' format.
And keep information and computation detail as more as possible.
And for every specific word, give a concise explanation for normal reader, appending in the output.
Theorem info:
{}: Content start:
{}
Content end.
We define the (If_Then format and explanation) as a rule.Please give me the rule based on the theorem info.
Directly output the rule content only without any conclusion.
Rule:

Figure 10: The prompt used for constructing TheoremQA.

Question Generation Prompt

Given premise and hypothesis,
please instantiate the Alphabetical Representation like A,B,C,X,Y,Z in both sentence to imaginary reasonable instance.
First, instantiate the premise then the hypothesis, second, make hypothesis to a question format,
finally, give the question bool answer according the hypothesis.
Please instantiate the premise with more extended lively detail.
While instantiate hypothesis and its question format concisely.
Output the whole result to a JSON like this:
{"premise_instantiated": "...", "hypothesis_instantiated": "...", "hypothesis_with_question_format": "..."}
Directly give out the JSON, no other explanation need.

Currently premise and hypothesis:
Premise:
{}
Hypothesis:
{}

Figure 11: The prompt used for constructing ULogic.

Rule Translation Prompt

输入:

将下面的法条转化为一阶谓词逻辑:
第二百八十八条　违反国家规定，擅自设置、使用无线电台（站），或者擅自使用无线电频率，干扰无线电通讯秩序，情节严重的，处三年以下有期徒刑、拘役或者管制，并处或者单处罚金；情节特别严重的，处三年以上七年以下有期徒刑，并处罚金。单位犯前款罪的，对单位判处罚金，并对其直接负责的主管人员和其他直接责任人员，依照前款的规定处罚。最终的结果是：扰乱无线电通讯管理秩序。

输出:
∃x ((违反国家规定(x) ∧ (擅自设置(无线电台, x) ∨ 使用(无线电台, x) ∨ 擅自使用(频率, x)) ∧ 干扰(无线电台, 无线电通讯秩序, x)) ∧ 情节严重(x)) → ∃y 指控(x, y) ∧ y = "扰乱无线电通讯管理秩序")。

输入:
将下面的法条转化为一阶谓词逻辑:

输出:

Figure 12: The prompt used for constructing CAIL2018.