MultiLogicNMR(er): A Benchmark and Neural-Symbolic Framework for Non-monotonic Reasoning Tasks with Multiple Extensions

Anonymous ACL submission

Abstract

Non-monotonic reasoning is a classic paradigm widely used in daily life and legal reasoning. The δ-NLI and LogicNMR proposed in the existing work have only preliminary explored the non-monotonic reasoning ability of the pre-trained language models (LMs) in natural language. However, the performance of large language models (LLMs) on complex non-monotonic reasoning tasks with multiple extensions has not yet been explored. An extension can be interpreted as a set of plausible conclusions. In this paper, we automatically synthesized a non-monotonic reasoning dataset with multiple extensions, MultiLogicNMR. Then, we systematically evaluated prompt-based and fine-tuned LLMs using skeptical and credulous reasoning, respectively. Skeptical reasoning only believes in common facts in all extensions, while credulous reasoning believes in facts in any one extension. In addition, inspired by classic symbolic solvers, we propose a neural-symbolic framework, MultiLogicNMRer, to improve the model's non-monotonic reasoning ability. Experimental results show that the accuracy of MultiLogicNMRer based on ChatGPT3.5 is about 23.1% higher $(46.2\% \rightarrow 69.3\%)$ than the corresponding prompt-based LLMs. The proposed MultiLogicNMR dataset and MultiLogicNMRer framework are expected to promote the research of LLMs on non-monotonic reasoning in natural language.

1 Introduction

Non-monotonic reasoning has been extensively studied in the field of traditional artificial intelligence (McCarthy, 1986; McDermott and Doyle, 1980). Reiter (1980) proposed default logic to formalize non-monotonic reasoning. The key to non-monotonic reasoning is to find all extensions based on the default theory, where an extension is a set of plausible conclusions. However, different default rules in a default theory may lead to multiple extensions; for example, given a default theory, "John is a professor. John is a chair. The professor usually teaches. The chair usually does not teach.". This default theory can get two extensions; in one, John is a typical professor and so teaches; in the other, he is a typical chair and does not teach (van Harmelen et al., 2008). In addition, skeptical and credulous reasoning are usually used in multiple extension non-monotonic reasoning. Skeptical reasoning only believes in common facts in all extensions, while credulous reasoning believes in facts in any one extension (van Harmelen et al., 2008). Non-monotonic reasoning also has a wide range of application scenarios, such as daily decision-making (Szalas, 2019) and legal reasoning (Gordon, 1988). Recently, large language models (LLMs) have achieved excellent performance in many logical reasoning tasks (Parmar et al., 2024), so exploring their logical reasoning ability in multiple extension non-monotonic reasoning has positive significance.

Researchers have proposed some non-monotonic reasoning benchmarks and used the prompt-based and fine-tuned LLMs to evaluate the models'

Facts: Toby is not noisy. Toby is not intelligent. Toby is not alert. Toby is important. Toby is not gorgeous.
Default Rules: If someoneA is not intelligent then he is delicious, unless he is not drab. If someoneA is delicious and not alert then he is huge, unless he is grieving. If someoneA is not noisy and delicious then he is different, unless he is dead. If someoneA is not gorgeous then he is grieving, unless he is huge.

The number of Extensions is: 2
E1: [Toby is delicious. Toby is grieving. Toby is different.]
E2: [Toby is delicious. Toby is huge. Toby is different.]

Question	Answer
Q1: Toby is delicious.	**A1**: True
Q2: Toby is not huge.	**A2**: Unknown
Q3: Toby is not different.	**A3**: False

Figure 1: A simplified example of skeptical reasoning mode in the MultiLogicNMR dataset.

non-monotonic reasoning ability. Rudinger et al. (2020) initially used NLI datasets to build a non-monotonic reasoning benchmark δ-NLI through crowdsourcing. However, δ-NLI entangled non-monotonic reasoning with commonsense reasoning. Recently, Xiu et al. (2022) constructed a natural language non-monotonic reasoning benchmark LogicNMR based on automatic synthesis, thereby reducing the interference of commonsense knowledge. However, the LogicNMR only involves non-monotonic reasoning with a single extension. Recently, Parmar et al. (2024) proposed a comprehensive logic reasoning benchmark, including non-monotonic logic, LogicBench. They generated simple non-monotonic reasoning samples based on eight default rules: Reasoning with Priorities, Default Reasoning with Irrelevant Information, etc. However, the prompt-based and fine-tuned LLMs still showed great challenges in non-monotonic reasoning tasks. On the one hand, the prompt content affected the performance of the prompt-based LLMs. On the other hand, although the fine-tuned LLMs had better logical reasoning ability, they did not really master non-monotonic logical reasoning.

This paper first proposes a non-monotonic reasoning benchmark with multiple extensions, MultiLogicNMR. Figure 1 shows a simplified sample in skeptical reasoning mode from the MultiLogicNMR. The question "Toby is delicious" appears in all extensions E1 and E2, so the answer is "True". However, the question "Toby is not huge" only appears in extension E2, so this answer is "Unknown". Based on MultiLogicNMR, we explore the non-monotonic reasoning capabilities of open-source and closed-source LLMs in skeptical and credulous reasoning, respectively. To further evaluate the generalization of LLMs on multiple extension non-monotonic reasoning tasks, we also constructed a dataset MultiLogicNMR_OOD involving more extensions and default rules. In addition, inspired by the symbolic solver in Algorithm 2 in the Appendix A.2, we propose a neural-symbolic framework for non-monotonic reasoning with multiple extensions, MultiLogicNMRer. The main idea of the MultiLogicNMRer framework is based on a symbolic solution strategy, and each module in the framework uses prompt-based LLMs to perform reasoning to complete different operations, thereby computing all extensions and answering questions.

The main contributions of this paper include the following points:

- This paper automatically synthesized a non-monotonic reasoning dataset with multiple extensions, **MultiLogicNMR**, and systematically explored the non-monotonic reasoning capabilities of LLMs in skeptical and credulous reasoning modes, respectively.
- We propose a neural-symbolic framework, **MultiLogicNMRer**, for multiple extension non-monotone reasoning task. MultiLogicNMRer can generate all the extensions to answer the question.
- Experiment results show that the proposed MultiLogicNMRer performs better than the prompt-based LLMs and even exceeds the fine-tuning LLMs.

2 Related Work

2.1 Preliminaries

Similar to LogicNMR (Xiu et al., 2022), we use default logic (DL) as a formal language for MultiLogicNMR in this work. The default rule is of the form: $\alpha : \beta_1, \beta_2, \ldots, \beta_m / \gamma$, where α, β_i and γ are formulas in first-order logic, α is called the prerequisite, $\beta_1, \beta_2, \ldots, \beta_m$ the justifications, and γ the conclusion. This default rule can be interpreted as if the prerequisite fact α is believed, and each justifications $\beta_1, \beta_2, \ldots, \beta_m$ can be consistently believed, then γ can be deduced. A default theory Γ consists of a pair of fact sets and default rule set $\Gamma = \langle D, W \rangle$, where D is the set of default rules and W is the set of facts. A set of sentences E is an extension of $\Gamma = \langle D, W \rangle$ iff for every sentence π, E satisfies $\pi \in E$ iff $W \cup \Delta \models \pi$, where $\Delta = \{\gamma | \alpha : \beta / \gamma \in D, \alpha \in E, \neg \beta \notin E\}$. So, an extension E is the set of entailments of $\{W \cup \gamma\}$, where the γ are consequents from D. For example: the default rules $D = \{prof(x) : teaches(x) \,/\, teaches(x), chair(x) : \neg teaches(x) \,/\, \neg teaches(x)\}$, fact set $W = \{prof(J), chair(J)\}$, this default theory has two extensions $E_1 = \{prof(J), chair(J), teaches(J)\}$, $E_2 = \{prof(J), chair(J), \neg teaches(J)\}$. Any such extension will be interpreted as an acceptable set of beliefs that one may hold about the incompletely specified world W (Reiter, 1980).

2.2 Benchmarks and Approaches

Existing work has proposed some natural language non-monotonic reasoning datasets. Recently,

Kazemi et al. (2023b) proposed a defeasible reasoning benchmark, BoardGameQA, which implements defeasible reasoning on conflicting knowledge bases through rule-based priority. However, the defeasibility in BoardGameQA is caused by conflicting rules, while the non-monotonicity in the proposed multiLogicNMR is caused by default conditions. In addition, Antoniou and Batsakis (2023) conducted a preliminary evaluation of the non-monotonic reasoning capabilities of the ChatGPT3.5 through the prompt-based method on ten classic examples. However, the ChatGPT3.5 still has a big gap with symbolic solvers. In this paper, the samples in MultiLogicNMR often involve multiple non-monotonic reasoning rules and may generate multiple extensions.

The prompt-based LLMs have been proven to have good logical reasoning capabilities. Zero-shot prompting directly instructs LLMs to perform a task without giving any examples, while few-shot prompting provides explicit examples (Brown et al., 2020; Kojima et al., 2022). In addition, Wei et al. (2022) proposed that the chain-of-thought prompting, including intermediate reasoning steps, can improve the reasoning ability of LLMs on complex tasks. Inspired by the chain-of-thought prompting, there have also proposed prompting methods involving more topological structures, such as tree of thoughts (Yao et al., 2023), graph of thoughts (Besta et al.), etc. The prompt-based methods do not require updating model parameters, but their reasoning performance is limited.

The fine-tuning method usually refers to fine-tuning model parameters in a fully supervised manner based on samples from downstream tasks. However, the model parameters of LLMs are enormous, and fine-tuning the model requires many training samples and computing resources. To solve these problems, on the one hand, a large number of samples are constructed through automatic synthesis (Wang et al., 2024; Guo and Chen, 2024); For example, Clark et al. (2020) used the synthetic deductive reasoning dataset RuleTaker to fine-tune the pre-trained LMs, verifying that the LMs can perform soft reasoning on natural language. On the other hand, some parameter-efficient fine-tuning methods are proposed, such as adapter tuning (Hu et al., 2022) and partial parameter tuning (Zaken et al., 2022) and so on. This paper mainly evaluates the generated MultiLogicNMR benchmark by prompting and fine-tuning LLMs.

Neural-symbolic framework have been widely used in deductive reasoning tasks, mainly including based on search and external solvers (Kautz, 2022). Search-based neural-symbolic methods usually use LMs for single-step reasoning in classical search algorithms (Hong et al., 2022). For example, Kazemi et al. (2023a) combines the capacity of LMs to handle naturalistic text input with the backward chaining algorithm, to solve the logical reasoning task. Recently, Hao et al. (2023) combined LLMs and Monte Carlo tree search (Browne et al., 2012) to achieve logical reasoning. Although the search algorithm can enhance the planning ability of the LLMs, the cost of calling LLMs is very high. In addition, The neural-symbolic solving based on external solvers is to use external solvers to solve the formal language translated by LLMs (Olausson et al., 2023). For example, Pan et al. (2023) combines a symbolic solver and LLMs, providing an effective way to achieve faithful logical reasoning. However, The formal language translated by LLMs is prone to grammatical errors and information loss, while the symbolic solver requires strict and correct formal input (Xu et al., 2024). In this paper, we build a neural-symbolic framework MultiLogicNMRer for non-monotonic reasoning based on the classic answer set programming (ASP) algorithm.

3 MultiLogicNMR Benchmark

This paper automatically synthesizes the MultiLogicNMR dataset[1] to explore the non-monotonic reasoning capabilities of LLMs in skeptical and credulous reasoning. The algorithm 1 in Appendix A.1 gives the process of generating the MultiLogicNMR dataset. The construction of the MultiLogicNMR mainly includes the following steps:

Generate Default Theory: The constants and variables are called terms, and $P(t_1, \ldots, t_n)$ is an atom if P is an n-ary predicate symbol and $t_1, \ldots, t_n$ are terms. A literal is an atom or the negation of an atom. We first randomly select predicates from the predicate pool to generate prerequisite literal, justification literal, and conclusion literal. It is required in generating the default rule that the α is the conjunction of at most two literals, justifications is a set containing at most two literals β_i, and the γ is a literal. Consistent with the predicate list in LogicNLI, the predicate pool contains unary predicates and binary predicates Tian et al. (2021). To ensure the default theory has multiple extensions, the literals in the prerequisite and con-

[1]URL link will be added after anonymous reviewing

clusion in different default rules may be the same, and the literals in the justifications and conclusions in different default rules may negate each other. After generating all the default rules, we randomly select entities from the entity pool to instantiate the prerequisite literals that are not included in the conclusion into facts W.

Convert DL to ASP: Baral and Gelfond (1994) pointed out that when α is a conjunction of literals, $\beta_1, \beta_2, \ldots, \beta_m$ and γ are all literals, such a default rule can be translated into an answer set rule γ :- $\alpha, not \neg \beta_1, \ldots, not \neg \beta_m$. To ensure the correctness of the answers, we convert the default rule into an ASP and then call the ASP solver for reasoning.

Solving: To ensure the correctness of the answer of the questions, we use the symbolic solver[2] to solve. First, all the extensions of the default theory are generated based on the symbolic solver, and to generate the correct answer of questions based on the extensions in the skeptical and the credulous reasoning, respectively. Given a default theory Γ and questions Q. Δ are all extensions. $E \in \Delta$ represents an extension. Equation 1 shows the answer to the question $A(\Gamma, Q)$ in the skeptical reasoning, and equation 2 shows the answer to the question $A(\Gamma, Q)$ in the credulous reasoning. The answer for the question may be True (T), False (F), and Unknown (M).

$$A(\Gamma, Q) = \begin{cases} T, \text{ if } \forall_E E \vdash Q \\ F, \text{ if } \forall_E E \vdash \neg Q \\ M, \text{ if } \exists_E E \nvdash Q, \exists_E E \nvdash \neg Q \end{cases} \quad (1)$$

$$A(\Gamma, Q) = \begin{cases} T, \text{ if } \exists_E E \vdash Q \\ F, \text{ if } \exists_E E \vdash \neg Q \\ M, \text{ if } \forall_E E \nvdash Q, E \nvdash \neg Q \end{cases} \quad (2)$$

Translate into Natural Language: Translate the default theory into natural language according to the template. For example, the rule "$\neg\, gorgeous(x)$:- $grieving(x), not\, huge(x)$." is translated into "If someoneA is grieving, then he is not gorgeous, unless he is huge".

We generated MultiLogicNMR datasets in skeptical and credulous reasoning modes, respectively, and Table 6 in Appendix A.1 shows the statistics of the MultiLogicNMR. Specifically, the training, development, and test sets contain 5000, 500, and 500 samples, respectively, and each sample contains three questions with different answer labels. The number of samples on the number of extensions $E = \{1, 2, 3, 4, 5\}$ is equal. Figure 1 shows a simplified example from the MultiLogicNMR dataset in skeptical reasoning. In addition, to measure the generalization of LLMs, we also generated the non-monotonic reasoning dataset MultiLogicNMR-OOD with more default rules and a more significant number of extensions.

[2]https://pypi.org/project/clyngor/

4 MultiLogicNMRer Framework

Inspired by the ASP symbolic solver in Appendix A.2 (Gebser et al., 2012), we propose a neural-symbolic framework for multiple extension non-monotonic reasoning, MultiLogicNMRer. Figure 2 shows the framework of the MultiLogicNMRer, which includes five modules: Grounding Module, Upper and Lower Bound Initialization Module, Reduction Module, Reasoning Module, and Selection Module. Firstly, the upper and lower bounds initialization modules are used to generate the initial upper and lower bounds fact sets of extension based on the instantiated default rules, and the reduced rules are generated based on the upper and lower bounds fact sets, respectively. Then, the upper and lower bounds facts are updated using the conclusions generated by the reduced rule. If the updated lower bound fact set is still a subset of the upper bound fact set, a fact is selected from the upper bound fact set and added to the lower bound set. Then, the reduction rules are inferred based on the updated upper and lower bound fact sets, and the upper and lower bound fact sets are updated again using the generated conclusions. Finally, the process is iterated until the upper and lower bounds facts are consistent and an extension is found. Appendix A.5 shows the full prompts used in different modules of the framework MultiLogicNMRer.

Grounding Module: The MultiLogicNMRer framework implements the grounding module using prompt-based LLMs. It is mainly used to reference the resolution of rules. For example, the rule "If someoneA is handsome and not intelligent then he is delicious, unless he is not drab." is instantiated as "If Toby is handsome and not intelligent then Toby is delicious ,unless Toby is not drab.".

Upper and Lower Bound Initialization Module: This module extracts facts from the rules through fact extraction prompts to initial upper and lower bound fact sets. Since the extension must contain the original facts in the default theory, the lower bound fact set is initialized with the original

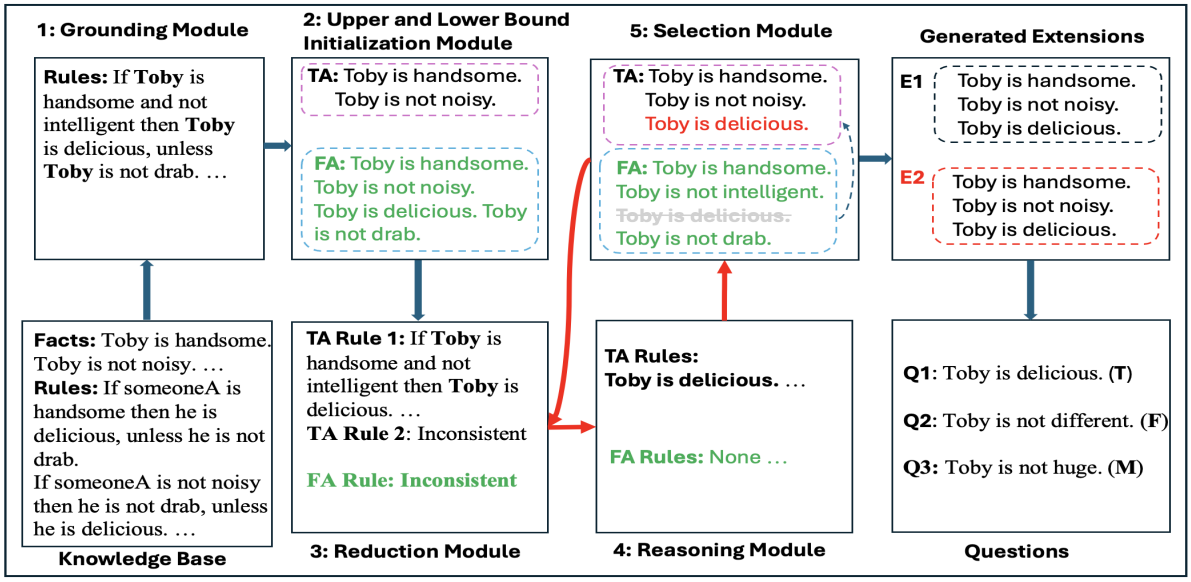

Figure 2: The framework of the proposed neural-symbolic method MultiLogicNMRer.

facts. In addition, the upper bound fact set should also include all the facts extracted from the rules.

Reduction Module: The reduction module obtains the reduction rules generated by the default rules under the upper and lower bound fact sets, respectively. For a default rule $\alpha : \beta_1, \beta_2, \ldots, \beta_m/\gamma$, if all the justification β_i does not appear in the lower bound set, the default rule can be reduced to α/γ. Otherwise, the rule cannot be reduced.

Reasoning Module: The reasoning module uses reasoning prompting to require the LLMs to reason about the reduced rules under the lower and upper bound fact sets, respectively, thereby generating conclusions. Then, the upper and lower-bound fact sets are updated using the conclusions generated by reduction rules. Specifically, the conclusions generated by the reduced rules under the upper bound fact set should be included in the lower bound fact set. In comparison, the upper bound fact set should only contain the conclusions generated by the reduced rules in the lower bound fact set.

Selection Module: When the updated lower-bound fact set is still a subset of the upper-bound fact set, select a fact from the upper-bound set to add to the lower-bound fact set, and the selected fact is removed from the upper-bound fact set. Then, the updated lower and upper bound facts are used to reason for the reduction rules. In the selection module, we use cosine similarity[3] to select facts from the upper bound fact set.

[3]https://huggingface.co/sentence-transformers/distilbert-base-nli-stsb-mean-tokens

5 Experiments

This section first explores the non-monotonic reasoning capabilities of prompt-based and fine-tuning LLMs using skeptical and credulous reasoning on the MultiLogicNMR. Then, the proposed neural-symbolic framework MultiLogicNMRer is evaluated. In addition, we also explore the generalization of these methods on the out-of-domain dataset MultiLogicNMR_OOD.

5.1 Experimental Settings

We systematically evaluates the non-monotonic reasoning capabilities of open-source LLMs (gpt-3.5-turbo (ChatGPT3.5) (Brown et al., 2020), gpt-4o OpenAI (2023), claude-3-sonnet-20240229[4] (Claude3), gemini-pro[5] (Gemini-pro), DeepSeek-chat (Bi et al., 2024)) (DeepSeek) and closed-source LLMs (Meta-LLAMA3-8B-Instruct (LLAMA3) (Touvron et al., 2023), gemma-7b-it[6] (Gemma), Mistral-7B-Instruct_v0.2 (Mistral) (Jiang et al., 2023)) in skeptical and credulous reasoning, respectively. The zero-shot and few-shot in-context learning strategies were used to evaluate the prompt-based LLMs, and the LoRA fine-tuning method was used to fine-tune the open-source LLMs. In addition, we use the accuracy as the evaluation metric[7].

[4]https://www.anthropic.com/news/claude-3-family
[5]https://blog.google/technology/ai/google-gemini-ai/
[6]https://ai.google.dev/gemma?hl=zh-cn
[7]https://pypi.org/project/scikit-learn/

5.2 Experimental Results

5.2.1 Results of Prompting-based LLMs

Tables 1 show the results of LLMs based on the zero-shot prompting on MultiLogicNMR in skeptical and credulous reasoning, respectively. The zero-shot prompting is shown in Appendix A.3. According to the results in Table 1, the non-monotonic reasoning capabilities of LLMs are relatively limited, among which LLAMA3 has the best performance among open-source LLMs. The average accuracy on all number of extensions can reach 40.4%. Among the closed-source LLMs, GPT-4o and Claude3 perform best and are equivalent, with the average accuracy reaching 54.5%. The results also show that closed-source LLMs still have more advantages than open-source LLMs in non-monotonic reasoning tasks. Secondly, the results of models in credulous reasoning are slightly higher than those in skeptical reasoning. This may be because credulous reasoning only needs a particular extension to get the correct answer, while skeptical reasoning needs to find all extensions to the correct answer. Third, regardless of whether in skeptical or credulous reasoning, the model's performance does not change much as the number of extensions increases. This illustrates that the number of extensions is not the main challenge for LLMs in non-monotonic reasoning tasks.

Table 1: Results of LLMs based on the zero-shot prompting on MultiLogicNMR in skeptical and credulous reasoning, respectively.

Model	Mode	Test (Accuracy (%))					
Extension		1	2	3	4	5	Avg.
Gemma	Skep -tical	37.0	38.0	29.3	32.6	31.0	33.6
Mistral		39.3	39.3	36.7	36.7	38.6	38.1
LLAMA3		40.3	39.0	41.7	35.0	42.3	39.7
Gemini-pro		42.3	45.0	49.6	41.6	52.3	46.2
Claude3		53.4	52.3	55.0	50.3	55.0	53.5
ChatGPT3.5		41.3	46.7	44.6	45.6	49.6	45.6
DeepSeek		38.0	37.0	40.0	41.0	45.3	40.2
GPT-4o		**57.0**	**58.3**	**56.3**	**53.6**	**58.3**	**56.7**
Gemma	Cred -ulous	32.0	35.0	36.0	38.0	36.7	35.5
Mistral		40.0	40.6	39.0	44.0	43.0	41.3
LLAMA3		36.7	41.6	43.0	42.3	42.3	41.2
Gemini-pro		44.6	45.6	45.6	48.3	48.0	46.5
Claude3		53.3	49.0	57.6	56.7	58.6	55.1
ChatGPT3.5		45.3	44.3	47.3	47.3	49.8	46.8
DeepSeek		41.0	42.0	39.6	36.3	38.3	39.5
GPT-4o		**59.3**	**56.0**	**58.3**	**62.7**	**62.7**	**59.8**

Table 2: Results of LLMs based on few-shot prompting on the MultiLogicNMR in skeptical and credulous reasoning.

Model	Mode	Test (Accuracy (%))					
Extension		1	2	3	4	5	Avg.
Gemma	Skep -tical	34.0	32.0	29.6	32.3	30.0	31.6
Mistral		36.3	40.6	36.0	36.0	38.0	37.4
LLAMA3		36.0	34.6	33.3	35.0	39.3	35.7
Gemini-pro		47.3	48.7	47.3	45.6	46.0	47.0
Claude3		56.3	59.3	57.6	57.4	**66.7**	**60.7**
ChatGPT3.5		39.3	36.7	33.3	30.3	40.6	36.1
DeepSeek		51.7	48.0	49.3	45.3	50.3	48.9
GPT-4o		**64.2**	**62.5**	**58.3**	**58.3**	58.3	60.3
Gemma	Cred -ulous	31.3	32.0	31.7	34.0	33.3	32.5
Mistral		35.3	36.0	40.3	40.0	38.3	38.0
LLAMA3		33.0	35.3	36.0	37.0	33.7	34.9
Gemini-pro		46.7	49.3	49.7	49.7	58.0	49.9
Claude3		56.7	55.7	56.3	60.0	63.3	58.4
ChatGPT3.5		43.3	40.6	39.6	42.6	41.6	41.6
DeepSeek		55.6	51.0	52.6	52.0	55.3	53.3
GPT-4o		**60.0**	**57.0**	**66.0**	**62.0**	**65.0**	**62.0**

Tables 2 show the results of LLMs based on few-shot prompting in skeptical and credulous reasoning, respectively. The few-shot prompting shown in Appendix A.3 includes three samples. Compared with models based on zero-shot prompting, the results of three open-source LLMs based on the few-shot prompting have declined. The possible reason is that the examples in the prompts interfere with the model's reasoning. Among the closed-source LLMs, gpt-4o, claude3, and Deepseek have all improved, among which gpt-4o performed the best, with an average accuracy of 60.3% and 60.5% in skeptical reasoning and credulous reasoning, respectively. The Claude3 model also shows close results to gpt-4o based on few-shot prompting. Surprisingly, the ChatGPT3.5 showed a similar trend to the open-source LLMs, with a drop of nearly 7%. In general, the few-shot prompting can further improve the non-monotonic reasoning capabilities of LLMs with strong reasoning performance.

5.2.2 Results of Fine-tuning LLMs

To further evaluate the non-monotonic reasoning ability of the LLMs, we use the LowRank Adaptation (LoRA) to fine-tune the open-source LLMs based on the training set. All details of the fine-tuning experiments are described in Appendix A.4.

Figure 3 shows the results of the fine-tuned LLMs on MultiLogicNMR. First, the average accu-

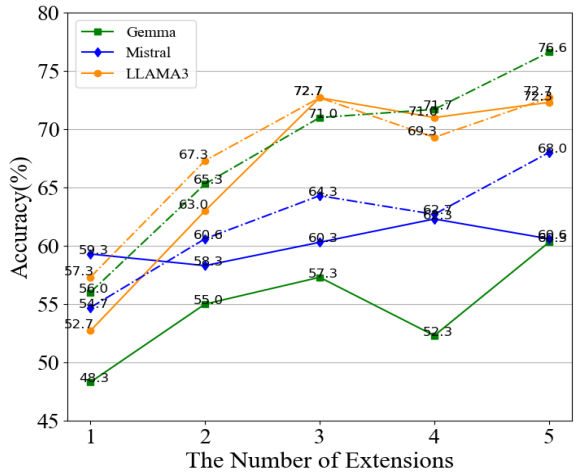

Figure 3: Results of fine-tuning models. The solid and dashed lines represent the results of models in skeptical and credulous reasoning, respectively.

racy of fine-tuned LLAMA3, Mistral, and Gemma models on all the number of extensions in skeptical reasoning is 66.3%, 60.2%, and 54.7%, respectively, which are all higher than the corresponding prompt-based LLMs. Second, the fine-tuned LLAMA3 performed better than the Mistral and Gemma. In addition, as the number of extensions increases, the accuracy of the fine-tuned models gradually increases. The possible reason is that the fine-tuned models do not learn non-monotonic reasoning but make predictions based on some correlation, and the increase in the number of extensions increases the correlation between context and question, resulting in the fine-tuned models having a greater probability of answering the question correctly. Finally, It is worth noting that the results of the Gemma in skeptical reasoning are much lower than those in credulous reasoning, indicating that the model is unstable and has not mastered non-monotonic reasoning.

5.2.3 Results of the MultiLogicNMRer

To evaluate the proposed MultiLogicNMRer, we use closed-source LLMs (ChatGPT3.5, DeepSeek-chat) and open-source LLMs (LLAMA3, Mistral) as the basic model in the MultiLogicNMRer and evaluate them on the MultiLogicNMR test set.

Table 3 shows the results of the MultiLogicNMRer based on different basic models. The results show that MultiLogicNMRer has a more significant improvement than the corresponding prompt-based basic model and is close to or even exceeds the

Table 3: Results of the proposed MultiLogicNMRer, abbreviated as MLNMRer, on the MultiLogicNMR dataset. The model in parentheses indicates the basic model used.

Models(Ours)	Mode	Test (Accuracy (%))					
Extension		1	2	3	4	5	Avg.
MLNMRer **(ChatGPT3.5)**	Skep-tical	70.6	69.3	72.0	67.3	72.0	70.3
MLNMRer **(DeepSeek)**		75.0	72.2	74.3	75.3	77.3	74.8
MLNMRer **(LLAMA3)**		71.0	71.3	72.3	70.3	76.3	72.3
MLNMRer **(Mistral)**		63.3	55.8	55.6	45.8	59.1	55.9
MLNMRer **(ChatGPT3.5)**	Cred-ulous	73.7	68.0	63.0	70.6	65.7	68.2
MLNMRer **(DeepSeek)**		**80.0**	**69.3**	**73.3**	**71.0**	**74.0**	**73.5**
MLNMRer **(LLAMA3)**		73.3	62.7	64.0	67.3	65.7	66.7
MLNMRer **(Mistral)**		60.8	52.8	51.7	60.0	53.3	55.7

results of the fine-tuning methods. For example, when DeepSeek was the base model, the average accuracy increased from 51.1% to 74.1%; when LLAMA3 was the base model, the average accuracy increased from 40.5% to 69.5%. These verify the effectiveness of the MultiLogicNMRer. In addition, the results of the MultiLogicNMRer on different numbers of extensions are consistent, which illustrates the stability and reliability of MultiLogicNMRer on the multiple extension non-monotonic reasoning. It is worth noting that although the results of the MultiLogicNMRer method based on the Mistral have declined to a certain extent, they are still higher than the prompt-based Mistral. The possible reason for the decline is that the basic model Mistral has poor logical reasoning capabilities, which leads to poor performance of some modules in the MultiLogicNMRer framework, thus affecting the overall performance.

5.3 Generalization

To explore the generalization of the LLMs on non-monotonic reasoning tasks, we evaluate the model's performance on the out-of-domain dataset MultiLogicNMR_OOD. Compared with the MultiLogicNMR dataset, the MultiLogicNMR_OOD contains more default rules and extensions. Table 6 in Appendix A.1 shows the statistical information

Table 4: Results of methods on the out-of-domain dataset MultiLogicNMR_OOD.

Model Extension	Met -hod	Mode	Test (Accuracy) 6	8	10	12	16	Avg
Gemma	Zero Shot	Skep -tical	34.3	30.7	35.0	31.3	33.0	32.9
Mistral			33.7	34.6	36.3	34.3	33.3	34.5
LLAMA3			37.0	35.0	34.7	40.7	32.7	36.0
Gemma		Cred -ulous	32.3	31.7	31.0	31.0	34.3	32.1
Mistral			30.3	31.3	33.0	34.3	30.0	31.8
LLAMA3			36.7	34.0	32.6	38.6	35.0	35.4
Gemma	Fine Tune	Skep -tical	67.7	75.0	75.0	70.0	69.6	71.5
Mistral			48.0	54.3	53.6	55.7	51.0	52.5
LLAMA3			66.0	64.7	67.0	63.3	62.6	64.7
Gemma		Cred -ulous	66.0	61.3	64.3	59.0	63.0	62.7
Mistral			66.7	64.0	65.3	63.3	66.3	65.1
LLAMA3			66.7	71.0	71.3	70.3	73.0	70.5
MLNMRer (LLAMA3)	ours	Skep -tical	59.7	65.0	72.3	69.3	65.3	66.3
		Cred -ulous	54.7	52.7	51.0	53.6	55.0	53.4

of MultiLogicNMR_OOD.

Table 4 gives the results of methods on the MultiLogicNMR_OOD. The results of the prompt-based LLMs are still poor, which further reveals the limitations of the prompt-based LLMs. The results of the fine-tuned LLMs only slightly decreased on MultiLogicNMR_OOD. The average accuracy of three fine-tuned LLMs in the skeptical reasoning on MultiLogicNMR_OOD datasets was 62.9%, respectively, indicating that increasing the number of rules and the number of extensions will not bring more challenges to the fine-tuned models. In addition, the result of MultiLogicNMRer in skeptical reasoning is 66.3%, which is still higher than the fine-tuning LLMs. Although the result of MultiLogicNMRer in credulous reasoning is 53.4%, it is still higher than prompt-based LLMs. These results show that the proposed MultiLogicNMRer framework has a certain degree of generalization.

5.4 Analysis and Case Study

To analyze the challenges of LLMs on the MultiLogicNMR, Figure 4 shows the distribution of answers generated by the models in skeptical reasoning. First, the results show that the zero-shot prompt-based ChatGPT3.5 and LLAMA3 have the lowest accuracy for questions with Unknown, especially ChatGPT3.5, which only answers correctly 35/500. In addition, although the fine-tuning methods can further improve the model's accuracy on questions with True and False, it still performs poorly on questions with Unknown. The possible reason is that the models need to find all the extensions to answer questions with Unknown correctly in skeptical reasoning, and it is still challenging for LLMs to find all the extensions. Moreover, Figure 4 shows that MultiLogicNMRer based on LLAMA3 can correctly answer 290/500 questions with Unknown while maintaining high accuracy for questions with other labels. These results further illustrate that the MultiLogicNMRer has more advantages and effectiveness than the prompting and fine-tuning methods.

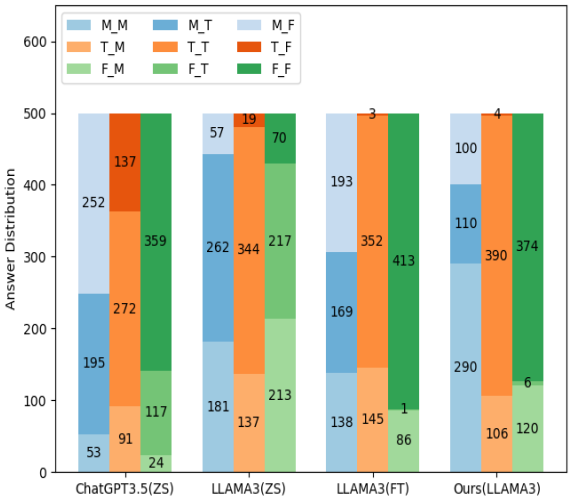

Figure 4: The distribution of answers generated by models in skeptical reasoning. The ZS and FT represent zero-shot prompting and fine-tuning, respectively.

6 Conclusions

In this paper, we automatically synthesize a non-monotonic reasoning dataset with multiple extensions, MultiLogicNMR, and propose a neural-symbolic framework, MultiLogicNMRer, for non-monotonic logical reasoning in natural language. MultiLogicNMR can be used to explore the non-monotonic reasoning capabilities of LLMs. Our work reveals that the prompt-based LLMs still face significant challenges on the non-monotonic reasoning task, and the fine-tuned LLMs do not understand non-monotonic reasoning. However, the performance of the MultiLogicNMRer can not only close and even exceed the fine-tuned models, but the reasoning process is also more reliable. The proposed MultiLogicNMR(er) takes a new step towards achieving a reliable logical reasoning approach with LLMs on non-monotonic reasoning.

7 Limitations

Although the proposed MultiLogicNMR and neural-symbolic framework MultiLogicNMRer can effectively explore and improve the non-monotonic reasoning ability in LLMs, there are some limitations. First, although the automatic synthesis method can generate a large number of samples according to constraints and ensure the correctness of answer labels, the MultiLogicNMR is translated from a formal language through a template, which makes the generated sentences relatively simple and still has a certain distance from the real natural language sentences. In addition, real logical reasoning scenarios often involve massive premise facts. However, the number of facts and rules in the MultiLogicNMR is usually small, which makes it challenging to evaluate the non-monotonic reasoning ability of LLMs in real scenarios based on MultiLogicNMR. Moreover, although the basic model in the proposed neural-symbolic framework MultiLogicNMRer does not need to be fine-tuned, it is necessary to iteratively call LLMs in multiple modules, which increases the cost of the MultiLogicNMRer framework.

References

Grigoris Antoniou and Sotiris Batsakis. 2023. Defeasible reasoning with large language models - initial experiments and future directions. In *Proceedings of the 17th International Rule Challenge and 7th Doctoral Consortium @ RuleML+RR 2023 co-located with 19th Reasoning Web Summer School (RW 2023) and 15th DecisionCAMP 2023 as part of Declarative AI 2023, Oslo, Norway, 18 - 20 September, 2023*, volume 3485 of *CEUR Workshop Proceedings*.

Chitta Baral and Michael Gelfond. 1994. Logic programming and knowledge representation. *J. Log. Program.*, 19/20:73–148.

Maciej Besta, Nils Blach, and Ales Kubicek et al. Graph of thoughts: Solving elaborate problems with large language models. In *AAAI 2024, pages = 17682–17690, year = 2024, url = https://doi.org/10.1609/aaai.v38i16.29720, doi = 10.1609/AAAI.V38I16.29720,*.

Xiao Bi, Deli Chen, and Guanting Chen et al. 2024. Deepseek LLM: scaling open-source language models with longtermism. *CoRR*, abs/2401.02954.

Tom B. Brown, Benjamin Mann, and Nick Ryder et al. 2020. Language models are few-shot learners. In *NeurIPS 2020, December 6-12, 2020, virtual*.

Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. In *IJCAI 2020*, pages 3882–3890. ijcai.org.

Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. 2012. *Answer Set Solving in Practice*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Thomas F Gordon. 1988. The importance of nonmonotonicity for legal reasoning. *Expert systems in law: Impacts on legal theory and computer law*, pages 111–126.

Xu Guo and Yiqiang Chen. 2024. Generative AI for synthetic data generation: Methods, challenges and the future. *CoRR*, abs/2403.04190.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 8154–8173.

Ruixin Hong, Hongming Zhang, Xintong Yu, and Changshui Zhang. 2022. METGEN: A module-based entailment tree generation framework for answer explanation. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 1887–1905.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Albert Q. Jiang, Alexandre Sablayrolles, and Arthur Mensch et al. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Henry A. Kautz. 2022. The third AI summer: AAAI robert s. engelmore memorial lecture. *AI Mag.*, 43(1):93–104.

Mehran Kazemi, Najoung Kim, Deepti Bhatia, Xin Xu, and Deepak Ramachandran. 2023a. LAMBADA: backward chaining for automated reasoning in natural language. In *ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6547–6568.

Mehran Kazemi, Quan Yuan, Deepti Bhatia, Najoung Kim, Xin Xu, Vaiva Imbrasaite, and Deepak Ramachandran. 2023b. Boardgameqa: A dataset for natural language reasoning with contradictory information. In *NeurIPS 2023*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *NeurIPS 2022*.

John McCarthy. 1986. Applications of circumscription to formalizing common-sense knowledge. *Artif. Intell.*, 28(1):89–116.

Drew V. McDermott and Jon Doyle. 1980. Non-monotonic logic I. *Artif. Intell.*, 13(1-2):41–72.

Theo Olausson, Alex Gu, Benjamin Lipkin, Cedegao E. Zhang, Armando Solar-Lezama, Joshua B. Tenenbaum, and Roger Levy. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *EMNLP 2023, Singapore, December 6-10, 2023*, pages 5153–5176.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 3806–3824.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. Towards systematic evaluation of logical reasoning ability of large language models. *arXiv preprint arXiv:2404.15522*.

Raymond Reiter. 1980. A logic for default reasoning. *Artif. Intell.*, 13(1-2):81–132.

Rachel Rudinger, Vered Shwartz, Jena D. Hwang, Chandra Bhagavatula, Maxwell Forbes, Ronan Le Bras, Noah A. Smith, and Yejin Choi. 2020. Thinking like a skeptic: Defeasible inference in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4661–4675.

Andrzej Szalas. 2019. Decision-making support using nonmonotonic probabilistic reasoning. In *Intelligent Decision Technologies 2019 - Proceedings of the 11th KES International Conference on Intelligent Decision Technologies (KES-IDT 2019), Volume 1, Malta, June 17-19, 2019*, volume 142 of *Smart Innovation, Systems and Technologies*, pages 39–51.

Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, and Yaohui Jin. 2021. Diagnosing the first-order logical reasoning ability through logicnli. In *EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3738–3747.

Hugo Touvron, Thibaut Lavril, and Gautier Izacard et al. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Frank van Harmelen, Vladimir Lifschitz, and Bruce W. Porter, editors. 2008. *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*.

Zifeng Wang, Chun-Liang Li, Vincent Perot, Long T. Le, Jin Miao, Zizhao Zhang, Chen-Yu Lee, and Tomas Pfister. 2024. Codeclm: Aligning language models with tailored synthetic data. *CoRR*, abs/2404.05875.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS 2022*.

Yeliang Xiu, Zhanhao Xiao, and Yongmei Liu. 2022. Logicnmr: Probing the non-monotonic reasoning ability of pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3616–3626.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024. Faithful logical reasoning via symbolic chain-of-thought. *arXiv preprint arXiv:2405.18357*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *(Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1–9.

A Appendix

A.1 MultiLogicNMR Dataset

Algorithm 1 describes the generation process of the MultiLogicNMR dataset. Lines 2 to 9 in Algorithm 1 show how to generate the default rules in each sample iteratively. Specifically, Lines 3, 5, and 6 in Algorithm 1 randomly select the prerequisite literal r_l^{pre}, justifications literal r_l^{jus}, and conclusion literal r_l^{con} of the default rule from the predicate pool P_P. In a default rule, we require that the prerequisite α is the conjunction of at most two literals r_l^{pre}, justifications is a set containing at most two literals r_l^{jus}, and the conclusion γ is a literal r_l^{con}. In addition, to be more capable of generating default theories with multiple extensions, the literals in the prerequisite and conclusion in different default rules may be the same, and the justifications and conclusions in different default rules may negate each other. After generating all the default rules, line 10 shows that we randomly select entities from the entity pool E_P to instantiate the prerequisite literals not included in the conclusion into facts W. In addition, we generate questions Q based on the conclusion literals r_l^{con} that are not included in the prerequisites. To generate answers to questions, we first call the symbolic solver[8] to generate all extensions of the default theory (line 12), and then use the skeptical and credulous reasoning to reason about the questions based on the extensions to generate answers (line 13). Finally, we translate all facts W, and default rules R, questions Q, answers A, and extensions E into natural language based on the template (line 14).

Table 6 gives the statistical information of the generated MultiLogicNMR dataset. Among them, the samples in MultiLogicNMR_OOD have more rules and involve more extensions than MultiLogicNMR. They are used to measure the generalization of LLMs on extended non-monotonic reasoning tasks.

Algorithm 1: MultiLogicNMR Dataset Generation Algorithm

Input: Predicates Pool P_P, Entity Pool E_P, Iterative Number T.
Output: Facts W, Default Rules R, Questions Q, Extensons E, Answers A.

1 Initialization: $iter = 1, T = 10, W = \varnothing, R = \varnothing, Q = \varnothing$;
2 **while** $iter \leq T$ **do**
3 **Generate prerequisite literal** r_l^{pre} based on predicate r_p^{pre} from P_P or R_P^{con};
4 $R_P^{pre} = R_P^{pre} \cup r_P^{pre}$;
5 **Generate justifications literal** r_l^{jus} based on predicate r_p^{jus} from P_P or R_P^{con}.
$R_P^{jus} = R_P^{jus} \cup r_P^{jus}$;
6 **Generate conclusion literal** r_l^{con}, predicate r_p^{con} from P_P or R_P^{con}.
$R_P^{con} = R_P^{jus} \cup r_P^{con}$;
7 **Generate default Rule** $R_i : \frac{R_l^{pre}:R_l^{jus}}{R_l^{con}}$, R_l^{pre} is the conjunction of at most two literals r_l^{pre}, R_l^{jus} at most two literals r_l^{jus}, and $R_l^{con} = r_l^{con}$;
8 $R = R \cup R_i, iter = iter + 1$;
9 **end**
10 **Generate fact literal** W_l^{fact} based on predicate $R_P^{fact} \in R_P^{pre} \setminus R_P^{con}, W = W \cup W_l^{fact}$;
11 **Generate question literal** R_l^{ques} based on predicate $R_P^{ques} \in R_P^{pre} \cap R_P^{con}, Q = Q \cup R_l^{ques}$;
12 **Generate default theory extensions** E through symbolic solvers[9]. ;
13 **Generate answers** A to questions Q based on the expansion E. ;
14 **Convert the facts, default rules and questions** into natural language ;

[8] https://pypi.org/project/clyngor/

A.2 Answer Set Solver

Algorithm 2 gives the symbolic solver for answer set programming. The idea of the proposed neural-symbolic framework MultiLogicNMRer is consistent with algorithm 2. In Algorithm 2, first, the input default theory is instantiated (line 16), and then the upper bound set U and the lower bound set L of the default theory are respectively calculated (line 17). Finally, the function $expand_p$ is called to update the upper and lower bound fact sets to generate the expansion (lines 1-7). Specifically, the lower bound set should contain the conclusions of the reduction rules under the upper bound set (line 5), while the upper bound set should only contain the conclusions of the reduction rules under the upper bound set (line 6). The lower bound is returned as an extension when the upper and lower bounds are consistent (line 11). If the updated lower bound set is included in the upper bound set, the extension search fails, and failure is returned (line 10); if the updated upper bound is still a superset of the lower bound, a random literal is selected from the upper bound set to be added to the lower bound set. The literal is deleted from the upper bound set. The upper and lower bounds set will be updated and searched again. All the upper and lower bound set pairs are searched, and all the extensions in the default rules are found.

It is worth noting that this symbolic solver only applies to normal logical program rules. On the one hand, the default rules in MultiLogicNMR generated under specific constraints can be converted into equivalent logic programs; on the other hand, although the proposed MultiLogicNMR dataset involves classical negation $\neg$, it is impossible to include atoms and the negation of atomic in the same extension at the same time. Hence, the symbolic solver's solution idea still applies to the proposed non-monotonic reasoning benchmark MultiLogicNMR.

Algorithm 2: Classic ASP solving algorithm

1 $expand_p$(L,U) ;
2 repeat ;
3 $L' \leftarrow L$;
4 $U' \leftarrow U$;
5 $L \leftarrow L' \cup Cn(P^{U'})$;
6 $U \leftarrow U' \cap Cn(P^{L'})$;
7 Until ($L = L'$) or $L \nsubseteq U$;
8 $solver_p$(L, U) ;
9 $(L, U) \leftarrow expand_p(L, U)$;
10 if $L \nsubseteq U$ then failure ;
11 if $L = U$ then output L ;
12 else $a \leftarrow choose(U \setminus L)$;
13 $solve_p(L \cup \{a\}$, U);
14 $solve_p(L, U \setminus \{a\}$);
15 main() ;
16 $P \leftarrow ground(input)$;
17 init(L, U) ;
18 $solve_p$(L, U) ;

A.3 Zero/Few-Shot Prompts

Zero-Shot Prompting in Skeptical Reasoning

Task Description: Given contexts and question, You need to generate answer labels for questions in a given context. The answers to the questions are labeled “True”, “False” and “Unknown”.
If the question can be inferred under all reasoning paths based on the context, and the negation of the question cannot be inferred under all reasoning paths based on the context, the answer label of the question is: “True”;
If the negation of the question can be inferred under all reasoning path based on the context, and the question cannot be inferred under all reasoning path based on the context, the answer label of the question is: “False”;
If the question and the negation of the question cannot be deduced under a certain reasoning path based on the context, the answer label of the question is: “Unknown”.
You must generate answer labels for the question.
The input format is: Context: “ ”. Question:“ ”.
The output format is: The answer label of the question is:“ ”.

Note that you only need to generate the answer label for the question without giving an explanation or justification. Please read the context carefully and answer the questions.

Zero-Shot Prompting in Credulous Reasoning

Task Description: Given contexts and question, You need to generate answer labels for questions in a given context. The answers to the questions are labeled “True”, “False” and “Unknown”.
If the question can be inferred under a certain reasoning path based on the context, the answer label of the question is: “True”;
If the negation of the question can be inferred under a certain reasoning path based on the context, the answer label of the question is: “False”;
If the question and the negation of the question both cannot be deduced under all reasoning path based on the context, the answer label of the question is: “Unknown”.
You must generate answer labels for the question.

The input format is: Context: “ ”. Question: “ ”.
The output format is: The answer label of the question is: “ ”.

Note that you only need to generate the answer label for the question, without giving an explanation or justification. Please read the context carefully and answer the questions.

Few-Shot Prompting in Skeptical Reasoning

Task Description: Given contexts and question, You need to generate answer labels for questions in a given context. The answers to the questions are labeled “True”, “False” and “Unknown”.
If the question can be inferred under all reasoning paths based on the context, and the negation of the question cannot be inferred under all reasoning paths based on the context, the answer label of the question is: “True”;
If the negation of the question can be inferred under all reasoning path based on the context, and the question cannot be inferred under all reasoning path based on the context, the answer label of the question is: “False”;
If the question and the negation of the question cannot be deduced under a certain reasoning path based on the context, the answer label of the question is: “Unknown”. Each context has a question, you must generate answer labels for each question.
The input format is: Context: “ ”. Question:“ ”.
The output format is: The answer label of the question is:“ ”.
Example 1: Context: Basil is not innocent. Basil is not wooden. Basil is discreet. Basil is not petite. Basil is comprehensive. Basil is nutty. Basil is historical. ... If someoneA is historical then he is red, unless he is not lively or he is not big. If someoneA is nutty and steep then he is miniscule, unless he is not weary or he is outstanding. If someoneA is not petite then he is brave, unless he is sticky or he is psychological. If someoneA is not wooden and miniscule then he is psychological, unless he is brave. ...
If the question is: Basil is red. **Then the answer label for the question is:** “True”;
If the question is: Basil is miniscule. **Then the answer label for the question is:** “Unknown”;
If the question is: Basil is not ashamed. **Then the answer label for the question is:** “False”.
Note that you only need to generate the answer label for the question, without giving an explanation or justification. Please read the context carefully and answer the questions.

Few-Shot Prompting in Credulous Reasoning

Task Description: Given contexts and question, You need to generate answer labels for questions in a given context. The answers to the questions are labeled “True”, “False” and “Unknown”.
If the question can be inferred under a certain reasoning path based on the context, the answer label of the question is: “True”; If the negation of the question can be inferred under a certain reasoning path based on the context, the answer label of the question is: “False”; If the question and the negation of the question both cannot be deduced under all reasoning path based on the context, the answer label of the question is: “Unknown”. Each context has three questions, You must generate answer labels for each question.
The input format is: Context: “ ”. Question:“ ”.
The output format is: The answer label of question is: “ ”.
Example 1: Context: Cecil is acceptable. Cecil is uptight. Cecil is not good tempered. Cecil is not severe. Cecil is not messy. Cecil is not self disciplined. Cecil is not logical. Cecil is not right. Cecil is careful.... If someoneA is not logical then he is not visible, unless he is not harsh. If someoneA is not messy and careful then he is not outstanding, unless he is not uptight. If someoneA is uptight and not severe then he is not successful, unless he is similar or he is not good. If someoneA is not visible then he is serious, unless he is not outstanding. If someoneA is not self disciplined then he is not fantastic, unless he is emotional or he is serious. ...
If the question is: Cecil is good. **Then the answer label for the question is:** “False”; **If the question is:** Cecil is not visible. **Then the answer label for the question is:** “Unknown”; **If the question is:** Cecil is similar. **Then the answer label for the question is:** “True”.
Note that you only need to generate the answer label for the question, without giving an explanation or justification. Please read the context carefully and answer the questions.

A.4 Experimental setup for fine-tuning LLMs

We use the LoRA fine-tuning method to fine-tune the open-source LLMs (LLAMA3-8B-Instruct[10], gemma-7b-it[11], Mistral-7B-Instruct_v0.2[12]), respectively. The parameters of the fine-tuned model are shown in Table 5. All fine-tuning experiments are completed on a single NVIDIA 4090 GPU based on the unsloth[13] framework.

Table 5: Fine-tuning experimental parameters of open-source LLMs.

Parameter	Value
per_device_train_batch_size	4
gradient_accumulation_steps	4
warmup_steps	10
max_steps	100
weight_decay	0.01
optim	Adamw_8bit
seed	3407

[10]https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[11]https://huggingface.co/google/gemma-7b-it

[12]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

[13]https://github.com/unslothai/unsloth

A.5 MultiLogicNMRer Prompts

Groundding Prompting

Task Description: Given a set of facts and a rule, you need to instantiate the rules based on given facts. Instantiation requires the replacement of pronouns in rules with individuals from the fact.

Example 1: Godwin is not sour. Godwin is short. Godwin is scared. Godwin is wild. Godwin is expensive. Godwin is not bad. Godwin is not straightforward. Godwin is anxious. Godwin is not stubborn. Godwin is not zany. Godwin laugh Connor.Godwin esteem Connor. Godwin is not immediate. Godwin is persistent. The rule is: If someoneA laugh someoneB and he is not stubborn then he is old, unless he is not poor.
The output is: If Godwin laugh Connor and Godwin is not stubborn then Godwin is old, unless Godwin is not poor.

The output format is: The output is:“ ”.

Note that you need to output all instantiation rules.

Fact Extraction Prompting

Task Description: Given a set of facts and a rule, you need to extract all instantiated facts in the rule.

Example 1: The rule is: If Godwin laugh Connor and Godwin is not stubborn then Godwin is old, unless Godwin is not poor or Godwin is unhappy.
The output is: Godwin laugh Connor. Godwin is not stubborn. Godwin is old. Godwin is not poor. Godwin is unhappy.

The output format is: The output is: “ ”.

Note that you only need to output all instantiated facts in the rule, do not print the contents of the prompt, and don’t output the same facts repeatedly.

Split Rule Prompting

Task Description: Given a rule, The rule format is: If A then B, unless C. The A is the prerequisite, the B is the conclusion, and the C is the justification. You need to output all prerequisite, conclusions, and justifications in this rule.

Example 1: The rule is: If Brice is emotional then Brice is beige, unless Brice is sufficient.
The output is: prerequisite: “Brice is emotional.”, conclusion:“Brice is beige. ”, justification: “Brice is sufficient.”.

Example 2: The rule is: If Cadman is historical and Cadman is emotional then Cadman is swift, unless Cadman is smart or Cadman is happy.
The output is: prerequisite: “Cadman is historical. Cadman is emotional.”, conclusion: “Cadman is swift.”; justification: “Cadman is smart. Cadman is happy. ”.

The output format is: The output is: prerequisite: “ ”, conclusion:“ ”, justification: “ ”.

Reasoning Prompting

Task Description: Given facts and a rule. You need to reason about the rules based on facts. The rule format is usually: If A then B. The A is the prerequisite, the B is the conclusion. If the prerequisite A is in the facts, you can deduce conclusion B. If the prerequisite A is not in the facts, then you can not deduce the conclusion B, so your output is: None.

Example 1: The input facts are: Godwin is not sour.Godwin is short.Godwin is scared. Godwin is wild. Godwin is expensive. Godwin is not bad. Godwin is not straightforward. Godwin is anxious. Godwin is not sour. Godwin is not zany. Godwin laugh Connor. Godwin esteem Connor. Godwin is immediate. Godwin is persistent. The rules are: If Godwin is not sour and immediate then Godwin is not lovely.
The output is: Godwin is not lovely.

Example 2: The facts are: Juliana is not old. Juliana is not anxious. Juliana is asleep. Juliana is giant. Juliana is not short. Juliana is comfortable. Juliana is not fearless. Juliana is aggressive. Juliana is not hot. Juliana is not southern. Juliana is not technical. Juliana is not educational. Juliana is not octagonal. Juliana is low. Juliana is not poor. The rule is: If someoneA is not short and not low then Juliana is persistent.
The output is: None.

The output format is: The output is:“ ”.

Note that you only need to output rule conclusions that can be inferred, not facts and reasoning processes.

A.6 Analysis and Case Study

Figure 5 also shows the distribution of answers generated by different methods in credulous reasoning, and a similar conclusion can be drawn as in Figure 5 in skeptical reasoning. Questions with Unknown are still very challenging to prompt and fine-tune methods. At the same time, the proposed neural symbolic framework MultiLogicNMRer has significantly improved the performance of questions with Unknown. To further explain this phenomenon, Figure 4 shows an example of the answer and explanation generated by GPT-4o in skeptical reasoning. According to the experimental results, it can be found that the model gave the correct answer to questions 1 and 3 with True and False answers, and GPT-4o generated a reasonable and correct reasoning path, respectively, which corresponds to a certain extension generated in the context. However, the model made a wrong prediction for question 2 with the answer Unknown. First, according to the model’s explanation, the answer generated for this question should be False, not True. This shows that the GPT-4o has inconsistencies in the reasoning process. In addition, the model explanation only contains one extension, so it cannot correctly implement non-monotonic reasoning.

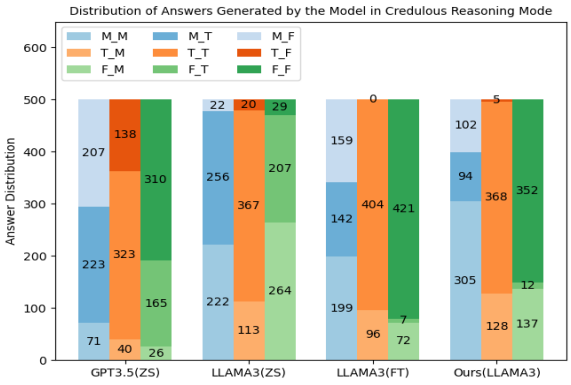

Figure 5: The distribution of answers generated by models in skeptical reasoning. The ZS represents the zero-shot prompt-based model and FT represents the fine-tuned model.

Table 6: Statistical information for MultiLogicNMR datasets.

Dataset	Mode		#Num.	#F.Avg	#R.Avg	#Extensions (1:1:1:1:1)	Label (T:F:M)
MultiLogicNMR	Skeptical	Train	5000	12	10	[1,2,3,4,5]	1:1:1
		Dev	500	12	10	[1,2,3,4,5]	1:1:1
		Test	500	12	10	[1,2,3,4,5]	1:1:1
	Credulous	Train	5000	12	10	[1,2,3,4,5]	1:1:1
		Dev	500	12	10	[1,2,3,4,5]	1:1:1
		Test	500	12	10	[1,2,3,4,5]	1:1:1
MultiLogicNMR_OOD	Skeptical	Test	500	22	10	[6,8,10,12,16]	1:1:1
	Credulous	Test	500	22	20	[6,8,10,12,16]	1:1:1

The #Num. represents the number of samples in the generated dataset, #F. Avg represents the average number of facts in the dataset, #R. Avg represents the average number of rules in the dataset, and #Extension represents the number of extensions.

Facts: Toby is not noisy.Toby is not intelligent.Toby is handsome.Toby is not alert.Toby is honest.Toby is important.Toby is young.Toby is not pessimistic.Toby scorn Benjamin.Toby is passionate.Toby is not gorgeous.Toby is scared.
Rules: If someoneA is handsome and not intelligent then he is delicious ,unless he is not drab.If someoneA is delicious and not alert then he is huge,unless he is grieving or he is not nasty.If someoneA is not pessimistic and scared then he is not foreign,unless he is not sensitive or he is not bossy.If someoneA is not noisy and delicious then he is different ,unless he is dead.If someoneA is not gorgeous and honest then he is grieving ,unless he is huge.If someoneA is different then he is not numerous,unless he is not confused or he is not financial.If someoneA scorn someoneB and someoneA is not grieving then he is not financial ,unless he is not distinct or he is not proud.If someoneA is important then he is not distinct ,unless he is not financial.If someoneA is passionate and not huge then he is not proud,unless he is not financial or he is not spotless.If someoneA is not numerous and young then he is pleasant ,unless he is not financial.

Question 1: Toby is delicious. **Answer 1: True** **Generate Answer 1: True**
Explanation 1: Toby is handsome and not intelligent, which triggers the rule that if someoneA is handsome and not intelligent then **he is delicious** unless he is not drab. Since there is no evidence in the context indicating that Toby is not drab, we conclude that Toby is delicious.

Question 2: Toby is not huge. **Answer 2: Unknown** **Generate Answer 2: True**
Explanation 2: Toby is handsome and not intelligent, making him delicious. **Toby is delicious and not alert**, with no mention of him grieving or being nasty, so **he is huge.** Hence, being huge implies he cannot be not huge, so the statement Toby is not huge is False, making the question "Toby is not huge" answered as True for the opposite.

Question 3: Toby is not different. **Answer 3: False** **Generate Answer 3: False**
Explanation 3: Toby is not noisy. Toby is handsome and not intelligent, which makes **him delicious.** Being delicious and not alert makes **Toby huge.** Furthermore, still being delicious makes **Toby different** because he is not dead, thus confirming that Toby is different. Therefore, the statement "Toby is not different" is false.

Figure 6: An example of reasoning by the GPT-4o in skeptical reasoning mode.