RMath: A Logic Reasoning-Focused Datasets Toward Mathematical Multistep Reasoning Tasks

Ziyi Hu[1], Jun Liu[2], Zhongzhi Liu[1], Yuzhong Liu[1], Zheng Xie[1], Yiping Song[1]*

[1]National University of Defense Technology, Changsha, China
[2]Sun Yat-sen University, Zhuhai, China

Abstract

Mathematical reasoning ability objectively reflects a language model's understanding of implicit knowledge in contexts, with logic being a prerequisite for exploring, articulating, and establishing effective reasoning. Large language models (LLMs) have shown great potential in complex reasoning tasks represented by mathematical reasoning. However, existing mathematical datasets either focus on common sense reasoning, assessing the model's knowledge application ability, or arithmetic problems with fixed calculation rules, evaluating the model's rapid learning capability. There is a lack of datasets that require solving problems solely through logical reasoning. As a result, the performance of LLMs in accurately understanding the implicit logical relationships in problems and deriving conclusions based solely on given conditions is hindered. To address this challenge, we construct a dataset specifically for multiple step reasoning tasks: Reasoning-Math (RMath). This dataset focuses on evaluating logical reasoning ability with mathematical reasoning problems, covering typical problem types, including direct reasoning problems, hypothetical reasoning problems, and nested reasoning problems. Additionally, we design a standardized annotation scheme that transforms natural language descriptions of conditions into formal propositions. Other annotation contents include problem categories, proposition truth values, and proposition relationship types. This not only reduces biases caused by semantic misunderstandings during problem-solving, but also facilitates the incorporation of theoretically grounded logical reasoning methods to enhance reasoning ability. Furthermore, we propose a normalization problem-solving framework based on propositional logic for RMath and design the problem-solving process for prompt tuning to guide LLMs to absorb mathematical logical theories and improving reasoning ability. Finally, we evaluate RMath on several popular LLMs and present the corresponding results.

Introduction

With the rapid increase in parameter sizes and training data, LLMs have gained emergent capabilities in various tasks, among which, powerful reasoning is one of the core capabilities for "intelligent emergency" of LLMs. Logical reasoning is a form of thinking in which premises and relations between premises are used in a rigorous manner to infer conclusions that are entailed (or implied) by the premises and the relations (Nunes 2012). In this context, the premises refer to known conditions, the relations refer to connective relations, and all the premises and conclusions are presented in the form of propositions. Mathematics is a subject in heavy reliance on multistep logical reasoning, and the ability to solve mathematical problems objectively reflects a model's logical and reasoning ability.

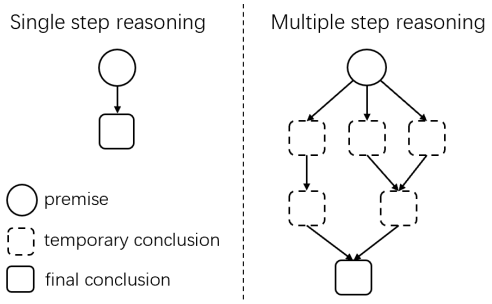

Figure 1: The comparison of single step reasoning of existing logical datasets and multiple step reasoning of RMath.

In recent years, there has been a significant surge in the development of LLMS for solving mathematical problems. Recent LLMs such as ToRA (Gou et al. 2023), DeepSeek-Math (Shao et al. 2024), and Llama3 (Meta 2024) have demonstrated new advancements in their ability to solve mathematical problems. However, whether these models truly possess powerful reasoning capabilities remains to be studied. Current datasets for measuring a model's reasoning ability can be roughly divided into two categories: one is based on common sense reasoning, where models tend to directly use stored world knowledge to solve problems, like MMLU (Hendrycks et al. 2020), StrategyQA (Geva et al. 2021), COPA (Brassard et al. 2022) and so on. They mainly focus on the model's ability to select and integrate relevant knowledge. MMLU is a benchmark designed to evaluate the performance of language models on multiple tasks. Models must have extensive knowledge of the world and problem-solving skills. The other is based on computational reasoning, for example, for some mathematical datasets like MATH (Hendrycks et al. 2021), GSM8K (Cobbe et al.

*Corresponding author

2021), SVAMP (Patel, Bhattamishra, and Goyal 2021), ASDIV (yun Miao, Liang, and Su 2020) and so on, where the problem statements mostly follow a fixed format, and the model tends to imitate the training set's procedural treatment of data to calculate result. MATH (Hendrycks et al. 2021) is a dataset of 12,500 challenging math competition problems with each problem having a complete step-by-step solution, which can be used to teach the model to generate answer derivations and explanations. However, these two types of datasets cannot accurately evaluate the model's true non-knowledge-based, non-imitation logical reasoning ability.

Currently, some research on datasets has begun to explore logical reasoning capabilities of LLMs from the perspective of mathematical logic. For example, LogicAsker (Wan et al. 2024) attempts to use the axiomatic laws of logical reasoning to generate some relatively simple single-step reasoning example, inputting these examples into LLMs to reveal weaknesses and improve reasoning. FOLIO (Han et al. 2022) is equipped with first-order logical annotations, consisting of 1430 examples, each paired with one of 487 sets of premises, which are used to deduce the validity of each conclusion through deductive reasoning. GloRE (Yunpeng Chen and et al. 2018) is a general logical reasoning evaluation benchmark that covers 12 datasets with a total of 72,848 instances and assembles three types of logical reasoning tasks. The three tasks cover a wide range of logical reasoning phenomena, which can evaluate the logical reasoning ability of LLMs across multiple logical reasoning tasks. However, these datasets are either based on first-order logical axioms to formulate simple one-step reasoning examples, or based on the truth value to formulate simple binary reasoning example, or integrated multiple reasoning tasks. But ultimately, all of them rely on single-step reasoning tasks based on logical reasoning, as shown in Figure 1, and have not attempted to solve some more complex multi-step reasoning tasks with LLMs. Multistep reasoning tasks are often more in line with real-world needs and can better reflect the logical reasoning capabilities and the ability to solve complex reasoning tasks of LLMs.

In this paper, we construct RMath[1], a dataset specifically for multistep reasoning tasks, where each problem in RMath requires no common sense or specialized knowledge and only reason based on given conditions. We also design a comprehensive and standardized annotation scheme, annotating each problem with propositions with truth values, propositions' relation, and problem categories. Thus, natural language is transformed into propositions and their relation, which can be further represented by symbols. The symbolic formulas build an foundation for propositional logic reasoning system, and help to prevent misinterpretation of the problem statement. Based on the annotation, we develop a standardized problem-solving framework with propositional logic reasoning theory: based on a set of prespecified answers, iteratively applying nested assumptions to each answer in the set and checking for contradictions with known conditions. If a contradiction is found, the prespecified answer is deemed invalid, and the next answer is tested. Continuing this process until the correct answer is found. Based on RMath, we apply the problem-solving framework to specific reasoning tasks, organize a standardized problem-solving process, and train the LLMs through prompt tuning to achieve better logic reasoning ability.

Our contributions are as follows:

- We construct a dataset called RMath specifically for multistep reasoning tasks, containing 200 problems that cover three types of problems. RMath can be used to evaluate and enhance the logical reasoning ability of LLMs.
- The dataset is meticulously annotated with problem categories, proposition truth values, and proposition relationship types. The annotation transforms the natural language to propositions with the aim of representing the implicit logical relationships by symbols.
- Based on the annotation, we propose a proposition-based problem-solving framework for the dataset. This framework aids LLMs in solving complex reasoning problems, thereby enhancing their logical reasoning ability with mathematical logic theory. We also test multiple LLMs on RMath and present the corresponding results.

Related Work

Logical Reasoning

Logical reasoning refers to argumentation and deduction that are in accordance with logic, representing a rigorous form of thinking. It can help to clarify thoughts, make correct judgments and decisions, and is the foundation of various fields in computer science and mathematics. There has been a lot of research exploring neuro-symbolic methods, which combine neural networks with symbolic reasoning (Mao et al. 2019; Pryor et al. 2022; Tian et al. 2022). However, these methods lack generality in specialized module designs. In contrast, LLMs show stronger generalization capabilities in logical reasoning. Propositional logic is the most fundamental part of logical reasoning, dealing with statements that can be assigned truth values. In the context of propositional logic parsing, (Tomasic1 et al. 2021) fine-tuned off-the-shelf GPT-2 and GPT-3 language models to simulate the propositional logic resolution of non-recursive rules that combine conjunction, disjunction, and negation connectives. Logical reasoning is now used in many exploratory tasks on pre-trained LLMs and applied to downstream tasks such as question answering and dialogue systems (Beygi et al. 2022; Shi et al. 2021). However, they primarily focus on simple reasoning in mathematical problems, neglecting multistep reasoning based on mathematical logic. They are limited to situations such as determining the truth of unknown propositions based on given propositions, or directly drawing conclusions based on reasoning formulas, without deeply exploring the relationships between conditions and the truth values of propositions.

[1]Our dataset and code are available at: https://github.com/huziyi19/RMath

Mathematical Dataset

In recent years, the ability to solve mathematical problems has been considered an important indicator for measuring the logical reasoning ability of language models (Romera-Paredes et al. 2023; Liu et al. 2020). At present, there are four types of mathematical problems that researchers are more focused on:

Arithmetic. This category of problems entails pure mathematical operations and numerical manipulation, devoid of the need for the model to interpret contextual elements (Ahn et al. 2024). MATH-140 (Yuan et al. 2023) contains 401 arithmetic expressions to test large language models. It focuses on the use of various operators in equations but not on reasoning.

Math Word Problems. These problems are arithmetic problems solved in conjunction with real-world application scenarios, which requires models to recognize the mathematical information contained in problems and formulate equations or expressions to address problems. The representative datasets are MATH (Hendrycks et al. 2021), GSM8K (Cobbe et al. 2021) and et al. While some complex problems in these datasets may involve reasoning, the focus is generally on single-step reasoning.

Automated Theorem Proving. These problems primarily assess a model's ability to perform deduction based on hypotheses, its mastery and use of formal languages, and its capacity to access and utilize extensive knowledge bases. Minif2f (Zheng, Han, and Polu 2021) is intended to provide a unified cross-system benchmark for neural theorem proving. But these problems do not focus on reasoning.

Math in Vision-language Context. These problems primarily assess a model's ability to solve mathematical problems in multimodal scenarios. MATHVISTA (Lu et al. 2023) is a benchmark designed to combine challenges from diverse mathematical and visual tasks. But these problems do not focus on reasoning.

The datasets about this four types of math problems cover various areas of mathematics, assessing the model's ability to integrate knowledge, apply formal languages and theorems, and master the rules of data procedural processing. While our dataset aims to evaluate the multistep reseaning ablitiy of LLMs without the assistance of external knowledge.

Data Collection and Analysis

In this work, we focus on mathematical reasoning tasks and construct a dataset about mathematical reasoning problems to enhance the long-chain reasoning ability of LLMs. As shown in Figure 2, each problem consists of a piece of description noted as D and a question noted as Q. The main sources of data collection are the internet and relevant books, but the vast internet-sourced data often suffers from poor quality. To ensure the quality of the dataset, we assemble several undergraduate students majoring in mathematics to systematically review the collected data. During this process, we intentionally skip reasoning problems about common sense to prevent errors caused by models when analyzing or understanding some common sense knowledge. Additionally, we exclude complex reasoning problems that are challenging even for humans. We focus on assisting models first with problem-solving of simple multi-step reasoning and will introduce more complex problems in future work.

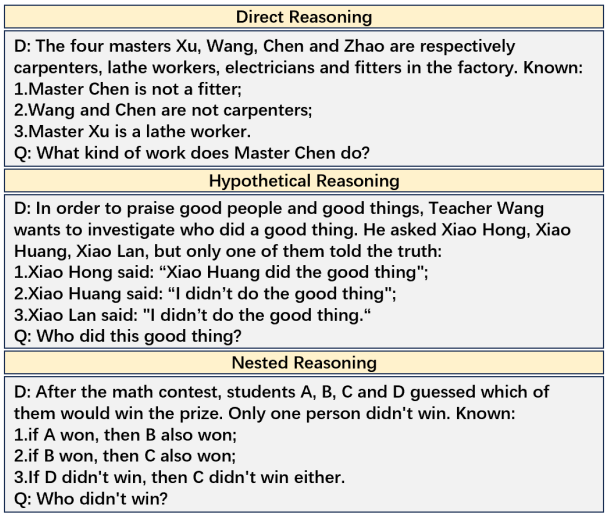

Figure 2: Examples of three types of reasoning problems.

The dataset includes three types of reasoning problems: direct reasoning problems, hypothetical reasoning problems, and nested reasoning problems. These three types of problems effectively assess the model's understanding of implicit logical relationships in the problems and ability to draw conclusions based on given conditions. The examples are shown in Figure 2.

- **Direct reasoning Problems.** These problems refer to reasoning problems where the given conditions in the problem are known to be true or false, and the solution can be directly inferred based on these conditions.
- **Hypothetical Reasoning Problems.** These problems refer to reasoning problems where some of the conditions given in the problem are true and some are false with the numbers of true and false conditions being certain. When solving these problems, the truth values of the given conditions should be hypothesized first based on the certain number of true and false conditions for further reasoning.
- **Nested Reasoning Problems.** These problems refer to reasoning problems where each condition given in the problem implies a simple reasoning relationship in the form of"If . . . , then . . . ". When solving these problems, the truth value of the "if" part in each condition should be hypothesized and judged firstly. Then based on this, the truth value of the"the" part can be deduced.

In mathematics, many reasoning problems have multiple answers. To simplify the problem-solving process, we preprocess the dataset to ensure each problem with only one answer. This helps LLMs to avoid errors in judgment caused by the failure to accurately understand the logical relationship between conditions when solving problems.

Proposition-based Problem Solving Framework

To assist LLMs to solve mathematical reasoning problems and understand the logical relationships in problems, we design a problem-solving framework based on propositional logic. Propositional logic is the most fundamental and simplest part of mathematical logic and focuses on reasoning mainly based on the relationships between propositions without the need to decompose them into their constituent non-propositional components. Importing the problem solving process from the framework by prompt tuning can help LLMs clarify and process the logical relationships implied between propositions during reasoning, thereby improving LLMs' ability to derive conclusions based on given conditions without auxiliary knowledge. The framework consists of three processes:

- **Proposition Transformation.** Proposition transformation first converts known conditions into propositions that can be judged as true or false, and then preliminarily distinguishes the implicit logical between propositions, deepening the model's understanding of the logical relationships in the problem during problem-solving.
- **Proposition Expansion.** Proposition expansion adds basis and conditions for reasoning and judgment, akin to data augmentation.
- **Proposition Connection and Judgment.** Proposition transformation and proposition expansion are prepared for proposition connection and judgment. The purpose of proposition connection and judgment is to assess whether the hypotheses about uncertain propositions are consistent with the requirements in problems by using the known propositions and then reason the truth value of the uncertain propositions.

Proposition Transformation

Proposition transformation transforms all the conditions of the problem into atomic propositions. Propositions are judgmental or declarative sentences with true or false meanings, divided into atomic propositions and compound propositions. The truth value of a proposition is its meaning of being true or false. Atomic propositions, also known as simple propositions, are propositions without any logical connectives and cannot be decomposed into other simple statements. Within the problem-solving framework, the subjects of our connection and judgment are all atomic propositions.

Generally, reasoning problems are described in natural language and the conditions in the problem are not necessarily in the form of atomic propositions. Therefore, before solving the problem, we transform all conditions into atomic propositions, which are unambiguous and easy for the large model to understand. During the transformation, as can be seen from the previous analysis, different types of problem imply different logical relationships. Thus, we categorize the transformed propositions into three classes: Class A, Class B, and Class C.

- Class A: Class A is the proposition from the problem description D whose truth values are determined. These propositions can be directly used as evidence to determine the truth values of other propositions.
- Class B: Class B is the proposition from the problem description D whose values are uncertain and should be determined by inference.
- Class C: Class C is the proposition derived from assumptions in the problem Q and does not exist in the description D. They are also uncertain and should be determined by inference. In Class C, there is only one true proposition with the other false.

Proposition Expansion

Proposition expansion expands propositions based on their truth values. Specifically, based on propositions in Class A and their truth values, start with the set of propositional objects and expand conditions by adding some additional propositions as "known condition". For example, for the Class A proposition "Xiao Hong is not from Beijing" (a true proposition), knowing that the locations in the propositional object set include Beijing, Shanghai, and Hangzhou, we can expand the initial condition as: Xiao Hong is from Shanghai or Xiao Hong is from Hangzhou. And this compound proposition is true. Further processing should be breaking down the compound proposition into two atomic propositions and classifying the propositions into the three categories based on the connection relationships and truth values between the two propositions.

Proposition Connection and Judgment

Initially, it should be known that the basis for judging the true answer is that when the assumption based on the answer is consistent with the requirements in problems, it can be determined as the true answer. What does"consistent" mean? First, the proposition describing the answer is true. And it does not contradict the propositions obtained from the problem transformation; it does not contradict the proposition deduced from the transformed propositions. What is "contradiction"? Contradiction refers to the situation where the semantics expressed by the combination of the two propositions and their truth values are contradictory and in conflict with each other when the truth values of them are determined.

The flow chart of proposition connection and judgment is shown in Figure 3, and the process is divided into nine steps. Initially, input the three types of propositions(step 1). Then starting with propositions in Class C, assume their truth values and connect them with propositions in Class A to check for contradictions. If there is a contradiction, revise the assumption for the propositions in Class C; if not, based on this assumption and according to the requirements in problems about the numbers of true or false propositions, hypothesize the truth values for all Class B propositions one by one. Check if the hypotheses of propositions in Class B contradict with each other. Connect them respectively with propositions in Class A (Loop-A-Contradict-B : step 4-6) and Class C and check for contradictions. If there is a contradiction, check if all the hypothesis combinations of truth values of propositions in Class B are cycled;if not, re-assume

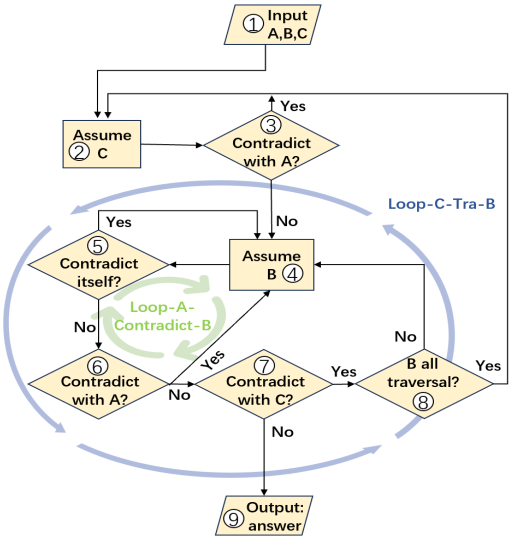

Figure 3: The flowchart for proposition connection and judgment.

for propositions in Class B; otherwise, re-assume for propositions in Class C (Loop-C-Tra-B:step 4-8), output the correct answer, the proposition in Class C that is true and consistent with the requirements in problems.

- Step 1: Input propositions in Class A, B, and C.
- Step 2: Assume truth or false value for propositions in Class C. Based on the sample space of the possible answer, make assumptions cyclically for the truth values of the propositions in Class C. Specifically, for each assumption (a total of n propositions), assume that the i-th proposition is true, and the remaining $n-1$ propositions are false.
- Step 3: Based on the assumption in Step 2, connect all propositions in Class C with propositions in Class A and check for contradictions. Specifically, connect all propositions with assumed truth values in Class C and propositions in Class A, and determine if there is any proposition in Class C that conflicts with propositions in Class A. If there is, then the assumption in Step 2 does not consistent with the requirements in problems, return to Step 2 and re-assume. Otherwise, proceed to Step 4.
- Step 4: Based on the assumptions in steps 2 and 3, further assume the truth values of all propositions in Class B according to the requirements in the problem for the numbers of true or false propositions. Generate the sample space of all possible combinations of truth values for propositions in Class B, and then iterate through these combinations.
- Step 5: Based on the assumptions in Step 4, check if the hypothesises of propositions in Class B contradict with each others. If there is, then the assumption in Step 4 does not consistent with the requirements in problems, return to Step 4 and re-assume. Otherwise, proceed to Step 6.
- Step 6: Check whether the assumptions in Step 4 are consistent with the requirements in problems. Based on the assumptions in Step 4, connect all propositions in Class B with propositions in Class A and determine if there is any proposition in Class B that conflicts with propositions in Class A. If there is, then the assumption in Step 4 does not consistent with the requirements in problems, return to Step 4 and re-assume. Otherwise, proceed to Step 7.
- Step 7: Check for contradictions between the assumptions for propositions in Class B in Step 4 and propositions in Class C in Step 2. Based on the assumptions in previous steps, connect all propositions in Class B with propositions in Class C and determine if there is any contradiction between propositions in Class B and C . If there is, proceed to Step 8. Otherwise, proceed to Step 9.
- Step 8: Check whether all possible combinations of truth values for propositions in Class B have been traversed. If they have all been traversed, then in this cycle, the assumptions for the truth values of propositions in Class C in Step 2 are contradictory to all combinations of truth values of propositions in Class B. That means the assumption for propositions in Class C in Step 2 does not consistent with the requirements in problems, then return to Step 2 and re-assume. Otherwise, return to Step 4.
- Step 9: Output the true propositions in Class C as the correct answer.

It is worth noted that for different problems, Class A and Class B may sometimes be empty. For example, for many direct reasoning problems, the set of Class B is empty, and in this case, the step of proposition connection and judgment for Class B is directly omitted, that is, Steps 4 to 7. Therefore, when the assumption for propositions in Class C is consistent with propositions in Class A and no contradiction occurs, it can be determined as the correct answer. For some hypothetical reasoning problems, the set of Class A is often empty. At this time, after the assumption for propositions in Class C, there is no need for the proposition connection and judgment between Class C and A. Then directly assume for Class B and connect propositions in Class B and Class C. As shown in the Figure 3 that is, skip the step three and step six.

Annotation

After data collection, we designed a standardized annotation scheme to annotate the problems. The annotation content includes problem categories, propositions, proposition object sets.

Problem Categories Annotation. Based on the different characteristics of the problems in the dataset, problems are classified and annotated to distinguish different types of problems: direct reasoning problems, hypothetical reasoning problems and nested reasoning problems.

Proposition Annotation. The annotation of propositions is divided into two parts: propositions' truth values and propositions' relationship. To label the true or false states of each proposition, we design three states: "true", "uncertain", "false". Proposition classification annotation means dividing the propositions into the three categories (Class A, Class B, Class C) based on the connectives contained in the problems

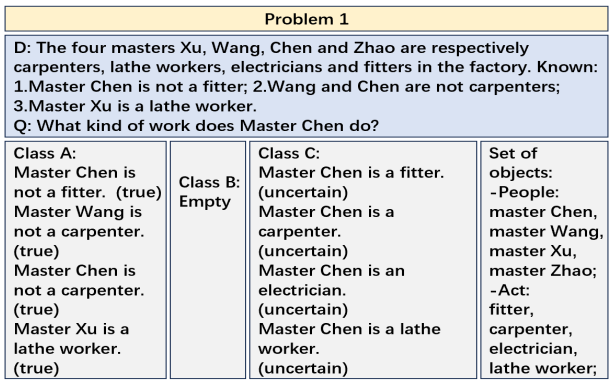

Figure 4: One example of annotation.

Problem category	Amount
Direct reasoning	34
Hypothetical Reasoning	137
Nested Reasoning	29
All	200
Proposition category	**Amount**
Class A	151
Class B	1179
Class C	642
Average	9.86

Table 1: Dataset Statistics

or the implied logical relationships. Propositional connectives refer to words that can link simple propositions to form compound propositions, which is the focus for this work.

Proposition Object Set Annotation. Proposition object set annotation refers to the extraction of key objects involved in each problem, such as people, places, actions, etc. These objects are extracted for proposition expansion, which is helpful for the hypothesis construction of propositions and subsequent reasoning and judgment.

One example of annotation is shown in Figure 4. Table 1 is a statistical table of the dataset after annotations and lists the number of each types of problems and the number of propositions for each category.

Prompt Tuning with the Framework on RMath

Based on RMath, we apply the problem-solving framework to specific reasoning tasks, organize a standardized problem-solving process and train the LLMs through prompt tuning. This can enhance the model's ability to handle logical relationships between propositions and standardize problem-solving, thereby improving the model's reasoning ability.

Experiments

LLMs on RMath

Experimental setting. We use RMath to train and test a range of large models with parameter sizes ranging from 7 billion, 8 billion, 13 billion, to 70 billion. We also demonstrate the performance of these LLMs on various datasets related to mathematical problems, including GSM8K (Cobbe et al. 2021), MATH (Hendrycks et al. 2021), SVAMP (Patel, Bhattamishra, and Goyal 2021), ASDIV (yun Miao, Liang, and Su 2020), MAWPS (Koncel-Kedziorski et al. 2016) and TabMWP (Lu et al. 2022). We use accuracy for the evaluation of logical reasoning performance. In the experiments, to ensure the accuracy of the results, we conduct repeated experiments and average the results.

Results. Table 2 shows the performance of various LLMs on our dataset RMath and other datasets related to mathematical problems, which assess the abilities of LLMs from different perspectives.

The reasoning ability of large models need improvement. As can be seen in the table, the accuracy of LLMs on RMath is generally not high. Even for models that perform exceptionally well on other math datasets, such as DeepSeekMath, the accuracy on RMath does not exceed 50%, despite maintaining a leading position. For some powerful LLMs, such as the recently released Llama3 8b and Llama3 70b, the performance on RMath is also unsatisfactory—with accuracy rates not even reaching 40%. Surprisingly, Llama2 7b and Llama2 13b have an accuracy rate of 0 on RMath. Even though there is a certain element of chance in how LLMs solve math reasoning problems, sometimes getting them right and sometimes wrong, their performance should not be that poor. The contrasting performance of some LLMs on RMath and other math datasets reflects that the reasoning ability of current LLMs to deduce conclusions based on known conditions need to be enhanced.

Existing datasets cannot comprehensively evaluate the reasoning ability of LLMs. It is obvious from the table that the ToRA models perform very well on other datasets, especially ToRA 70b, which has an accuracy rate higher than 80% on the GSM8K, SVAMP and ASDIV datasets, and even reaches 93.8% on MAWPS. Even on the Math dataset, one of the most challenging math datasets, ToRA has a good performance and demonstrates strong comprehensive abilities in solving problems. However, the performance of the ToRA models on RMath is unsatisfactory. Among them, ToRA 70b, which performs well on other datasets, has only 31% accuracy. From this phenomenon, we can see that existing datasets cannot comprehensively evaluate the reasoning ability of LLMs, and the construction of RMath has also made contributions to the comprehensive evaluation of the reasoning ability of LLMs.

Prompt Tuning on RMath

Experimental setting. Based on the proposed standardized problem-solving framework, we construct a training dataset, RMath-train, from RMath. Specifically, we develop detailed problem-solving processes according to the proposed framework to build RMath-train for LLM's training. Our baseline models include the base LLMs llama2 (7b, 13b, 70b) (Touvron et al. 2023), llama3 (8b, 70b) (Meta 2024), and LLMs oriented to solving mathematical problems with SFT or RLHF, WizardMath (7b, 13b, 70b) (Luo et al. 2023), MetaMath (7b, 13b, 70b) (Yu et al. 2023) and ToRA (7b, 13b, 70b) (Gou et al. 2023).

Model	Size	GSM8K	MATH	SVAMP	ASDIV	MAWPS	TabMWP	RMath (our)
Llama2	7b	13.3	4.1	38.0	50.7	60.9	31.1	**0**
Llama3	8b	79.6	30.0	–	–	–	–	**23.0**
WizardMath	7b	54.9	10.7	57.3	59.1	73.7	38.1	**19.5**
MetaMath	7b	66.6	20.7	68.8	72.5	86.9	43.8	**17.0**
ToRA	7b	68.8	40.1	68.2	73.9	88.8	42.4	**15.5**
DeepSeekMath	7b	63.3	32.3	73.2	82.9	92.4	68.6	**45.0**
Llama2	13b	24.3	6.3	43.1	56.3	70.4	39.5	**0**
WizardMath	13b	63.9	14.0	64.3	65.8	79.7	46.7	**47.5**
MetaMath	13b	71.0	23.2	71.9	75.7	87.0	52.8	**32.5**
ToRA	13b	72.7	43.0	72.9	77.2	91.3	47.2	**26.0**
Llama2	70b	57.8	14.4	73.6	76.0	92.4	57.5	**39.5**
Llama3	70b	93.0	50.4	–	–	–	–	**33.0**
WizardMath	70b	81.6	22.7	80.0	76.2	86.2	49.8	**54.5**
MetaMath	70b	82.0	27.2	85.8	84.0	95.4	63.4	**64.0**
ToRA	70b	84.3	49.7	82.7	86.8	93.8	74.0	**31.0**

Table 2: Results of several LLMs on datasets

Model	Size	RMath (our)	Model	Size	RMath (our)
Llama2	7b	0	MetaMath	13b	32.0
Llama2-RMath	7b	**18.5**	MetaMath-RMath	13b	**54.0**
WizardMath	7b	19.5	ToRA	13b	26.0
WizardMath-RMath	7b	**24.0**	ToRA-RMath	13b	**30.5**
MetaMath	7b	17.0	Llama2	70b	39.5
MetaMath-RMath	7b	**20.0**	Llama2-RMath	70b	**43.5**
ToRA	7b	15.5	Llama3	70b	33.0
ToRA-RMath	7b	**21.0**	Llama3-RMath	70b	**64.0**
Llama3	8b	23.0	WizardMath	70b	54.5
Llama3-RMath	8b	**41.0**	WizardMath-RMath	70b	28.5
Llama2	13b	0	MetaMath	70b	64.0
Llama2-RMath	13b	**23.0**	MetaMath-RMath	70b	42.0
WizardMath	13b	47.5	ToRA	70b	31.0
WizardMath-RMath	13b	31.0	ToRA-RMath	70b	**35.5**

Table 3: The comparison of LLMs on RMath before and after prompt tuning

Results. Table 3 shows the performance of the models on the RMath before and after prompt tuning. We conduct prompt tuning on multiple LLMs based on RMath-train and compared the performance of the models before and after training on RMath. It can be found from the table that the accuracy of the models after prompt tuning on RMath has generally improved, with an average increase of about 5.3%, among which the accuracy of llama3 70b improved the most, up to 31%. This indicates that the problem-solving framework we proposed helps to enhance the logical reasoning ability of LLMs when solving mathematical reasoning problems.

At the same time, some discordant numbers appear in the table. The accuracy of WizardMath 13b and MetaMath 70b on RMath after training is not as good as that of the original models. Why some LLMs after prompt tuning do not perform as well on RMath as the original models? We conduct a preliminary analysis and speculation: it may be that for some LLMs with better comprehensive performance, using prompt tuning to improve performance on a specific task might have less noticeable effects.

Conclusions

In this paper, we construct a dataset specifically for multiple step reasoning tasks called RMath. It is a mathematical reasoning dataset focused on assessing logical reasoning ability, containing 200 problems that include direct reasoning problems, true-or-false reasoning problems, and nested reasoning problems. Additionally, we design an annotation scheme to reduce semantic bias during problem-solving with LLMs and introduce theoretically grounded logical reasoning methods to enhance the reasoning ability of LLMs. Furthermore, we develop a standardized problem-solving framework for RMath based on propositional logic and use the problem-solving process formed with the framework in the prompt tuning for LLMs to guide them to assimilate mathematical logic theory and improve their reasoning ability. We evaluate multiple LLMs with RMath and present the corresponding results. Finally, we train the LLMs with RMath through prompt tuning. From the results, the problem-solving framework we proposed does help to enhance the logical reasoning ability of LLMs when solving mathematical reasoning problems.

Acknowledgments

This paper is supported by National Natural Science Foundation of China (NSFC Grant No. 62106275).

References

Ahn, J.; Verma, R.; Lou, R.; Liu, D.; Zhang, R.; and Yin, W. 2024. Large Language Models for Mathematical Reasoning: Progresses and Challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, 225–237. St. Julian's, Malta: Association for Computational Linguistics.

Beygi, S.; Fazel-Zarandi, M.; Cervone, A.; Krishnan, P.; and Jonnalagadda, S. 2022. Logical Reasoning for Task Oriented Dialogue Systems. In *Proceedings of the Fifth Workshop on e-Commerce and NLP*, 68–79. Dublin, Ireland: Association for Computational Linguistics.

Brassard, A.; Heinzerling, B.; Kavumba, P.; and Inui, K. 2022. COPA-SSE: Semi-structured Explanations for Commonsense Reasoning. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 3994–4000. Marseille, France: European Language Resources Association.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.

Geva, M.; Khashabi, D.; Segal, E.; Khot, T.; Roth, D.; and Berant, J. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. In *Transactions of the Association for Computational Linguistics*, 346–361. Cambridge, MA: MIT Press.

Gou, Z.; Shao, Z.; Gong, Y.; Shen, Y.; Yang, Y.; Huang, M.; Duan, N.; and Chen, W. 2023. Tora: A tool-integrated reasoning agent for mathematical problem solving. arXiv:2309.17452.

Han, S.; Schoelkopf, H.; Zhao, Y.; Qi, Z.; Riddell, M.; Zhou, W.; Coady, J.; Peng, D.; Qiao, Y.; Benson, L.; Sun, L.; Wardle-Solano, A.; Szabo, H.; Zubova, E.; Burtell, M.; Fan, J.; Liu, Y.; Wong, B.; Sailor, M.; Ni, A.; Nan, L.; Kasai, J.; Yu, T.; Zhang, R.; Fabbri, A. R.; Kryscinski, W.; Yavuz, S.; Liu, Y.; Lin, X. V.; Joty, S.; Zhou, Y.; Xiong, C.; Ying, R.; Cohan, A.; and Radev, D. 2022. FOLIO: Natural Language Reasoning with First-Order Logic. arXiv:2209.00840.

Hendrycks; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020. Measuring Massive Multitask Language Understanding. In *The Ninth International Conference on Learning Representations*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. arXiv:2103.03874.

Koncel-Kedziorski, R.; Roy, S.; Amini, A.; Kushman, N.; and Hajishirzi, H. 2016. MAWPS: A Math Word Problem Repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1152–1157. San Diego, California: Association for Computational Linguistics.

Liu, J.; Cui, L.; Liu, H.; Huang, D.; Wang, Y.; and Zhang, Y. 2020. LogiQA: A Challenge Dataset for Machine Reading Comprehension with Logical Reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 3622–3628. Yokohama: International Joint Conferences on Artificial Intelligence.

Lu, P.; Bansal, H.; Xia, T.; Liu, J.; Li, C.; Hajishirzi, H.; Cheng, H.; Chang, K.-W.; Galley, M.; and Gao, J. 2023. MathVista: Evaluating Mathematical Reasoning of Foundation Models in Visual Contexts. In *The Twelfth International Conference on Learning Representations*.

Lu, P.; Qiu, L.; Chang, K.-W.; Wu, Y. N.; Zhu, S.-C.; Rajpurohit, T.; Clark, P.; and Kalyan, A. 2022. Dynamic Prompt Learning via Policy Gradient for Semi-structured Mathematical Reasoning. arXiv:2209.14610.

Luo, H.; Sun, Q.; Xu, C.; Zhao, P.; Lou, J.; Tao, C.; Geng, X.; Lin, Q.; Chen, S.; and Zhang, D. 2023. WizardMath: Empowering Mathematical Reasoning for Large Language Models via Reinforced Evol-Instruct. arXiv:2308.09583.

Mao, J.; Gan, C.; Kohli, P.; Tenenbaum, J. B.; and Wu, J. 2019. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In *The Seventh International Conference on Learning Representations*.

Meta. 2024. Introducing Meta Llama 3. https://ai.meta.com/research/publications/the-llama-3-herd-of-models/. Https://openai.com/index/chatgpt/.

Nunes, T. 2012. Logical Reasoning and Learning. *Encyclopedia of the sciences of learning*, 2066–2069.

Patel, A.; Bhattamishra, S.; and Goyal, N. 2021. Are NLP Models really able to Solve Simple Math Word Problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2080–2094. Online: Association for Computational Linguistics.

Pryor, C.; Dickens, C.; Augustine, E.; Albalak, A.; Wang, W.; and Getoor, L. 2022. NeuPSL: Neural Probabilistic Soft Logic. In *International Joint Conference on Artificial Intelligence*.

Romera-Paredes, B.; Barekatain, M.; Novikov, A.; Balog, M.; Kumar, M. P.; Dupont, E.; Ruiz, F. J. R.; Ellenberg, J. S.; Wang, P.; Fawzi, O.; Kohli, P.; Fawzi, A.; Grochow, J.; Lodi, A.; Mouret, J.-B.; Ringer, T.; and Yu, T. 2023. Mathematical discoveries from program search with large language models. *Nature*, 625: 468 – 475.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: pushing the limits of mathematical reasoning in open language models. arXiv:2402.03300.

Shi, J.; Ding, X.; Du, L.; Liu, T.; and Qin, B. 2021. Neural Natural Logic Inference for Interpretable Question Answering. In *Proceedings of the 2021 Conference on Empirical*

Methods in Natural Language Processing, 3673–3684. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Tian, J.; Li, Y.; Chen, W.; Xiao, L.; He, H.; and Jin, Y. 2022. Weakly Supervised Neural Symbolic Learning for Cognitive Tasks. In *The 36th AAAI Conference on Artificial Intelligence*, 5888–5896. Vancouver, Canada: Association for the Advancement of Artificial Intelligence.

Tomasic1, A.; Romero1, O. J.; Zimmerman1, J.; and Steinfeld1, A. 2021. Propositional Reasoning via Neural Transformer Language Models. In *International Workshop on Neural-Symbolic Learning and Reasoning (NeSy)*, 104–119.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Ferrer, C. C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288.

Wan, Y.; Wang, W.; Yang, Y.; Yuan, Y.; tse Huang, J.; He, P.; Jiao, W.; and Lyu, M. R. 2024. LogicAsker: Evaluating and Improving the Logical Reasoning Ability of Large Language Models. arXiv:2401.00757.

Yu, L.; Jiang, W.; Shi, H.; Yu, J.; Liu, Z.; Zhang, Y.; Kwok, J. T.; Li, Z.; Weller, A.; and Liu, W. 2023. MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models. arXiv:2309.12284.

Yuan, Z.; Yuan, H.; Tan, C.; Wang, W.; and Huang, S. 2023. How well do Large Language Models perform in Arithmetic tasks? arXiv:2304.02015.

yun Miao, S.; Liang, C.-C.; and Su, K.-Y. 2020. A Diverse Corpus for Evaluating and Developing English Math Word Problem Solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 975–984. Online: Association for Computational Linguistics.

Yunpeng Chen, M. R.; and et al., Z. Y. 2018. Graph-Based Global Reasoning Networks. In *Proceedings of the arxiv18*.

Zheng, K.; Han, J. M.; and Polu, S. 2021. MiniF2F: a cross-system benchmark for formal Olympiad-level mathematics. In *The Tenth International Conference on Learning Representations*.