Can LLMs Learn from Previous Mistakes? Investigating LLMs' Errors to Boost for Reasoning

Anonymous ACL submission

Abstract

Large language models (LLMs) have demonstrated striking reasoning capability. Recent works have shown the benefits to LLMs from fine-tuning golden-standard CoT rationales or using them as correct examples in few-shot prompting. While humans can indeed imitate correct examples, learning from our mistakes is another vital aspect of human cognition. Hence, a question naturally arises: *can LLMs learn and benefit from their mistakes, especially for their reasoning?* This study investigates this problem from both the prompting and model-tuning perspectives. We begin by introducing CoTErrorSet, a new benchmark with 609,432 questions, each designed with both correct and error references, and demonstrating the types and reasons for making such mistakes. To explore the effectiveness of those mistakes, we design two methods: (1) Self-rethinking prompting guides LLMs to rethink whether they have made similar previous mistakes; and (2) Mistake tuning involves finetuning models in both correct and incorrect reasoning domains, rather than only tuning models to learn ground truth in traditional methodology. We conduct a series of experiments to prove LLMs can obtain benefits from mistakes. Both of our two methods serve as potential low-cost solutions to utilize mistakes to improve reasoning abilities compared with the high cost of making hand-crafted references. We ultimately make a thorough analysis of the reasons behind LLMs' errors, which provides directions that future research needs to overcome. CoTErrorSet will be published soon on `Anonymity Link`.

1 Introduction

LLMs have recently proven strong capabilities across various reasoning tasks (Huang, 2022; Kojima et al., 2022). (Wei et al., 2022) proposed CoT prompting, guiding LLMs to think step by step, which becomes a new paradigm to align LLMs' reasoning with the human thinking process. Unfortunately, few studies have focused on fully understanding what kinds of intermediate errors occur in making CoT procedures and whether LLMs can learn from those mistakes like humans.

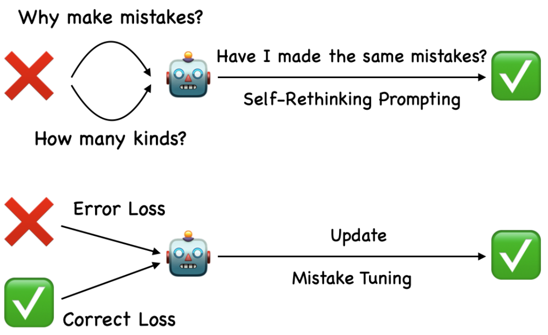


Figure 1: Our two proposed methods to utilize incorrect CoT rationales: self-rethinking prompting and mistake tuning. Our experiments demonstrate LLMs can consistently benefit from incorrect rationales.

Indeed, recognizing and correcting previous mistakes serves as a critical component for better learning and reasoning abilities for our humans (Mercer, 2008; Reich et al., 2023). In order to thoroughly explore whether LLMs have similar capabilities of utilizing their errors in LLMs' reasoning, we systematically collect a vast dataset of LLMs' reasoning outputs and built CoTErrorSet, which consists of 609,432 questions collected from 1060 tasks across various domains. Each is designed with both the hand-crafted correct reference and PaLM2-540B's incorrect rationales. Additionally, we provide the LLMs with the correct reference and annotate the type of each error made and the potential reasons behind them.

In this study, we introduce two possible solutions to investigate the potential benefits of mistakes: self-rethinking prompting and mistake tuning. For self-rethinking, we first provide LLMs with corresponding 8 incorrect rationales randomly selected from CoTErrorSet as the in-domain knowledge for all tasks in arithmetic reasoning. Similarly, for commonsense reasoning, another distinct set of 8 incorrect rationales in the same domain is employed for all questions in commonsense reasoning. Then, after each subsequent reasoning, we guide LLMs to self-rethink whether they make similar mistakes. If they recognize such errors in their output, they are then instructed to correct their reasoning based on the provided domain knowledge. To prevent excessive computational expenditure and avoid loops, we set a threshold to limit the number of times the model can perform self-rethink and corrections. We conduct a series of experiments and prove that LLMs can utilize mistakes through self-rethinking, which vastly and consistently outperforms self-consistency under the same computational costs.

We propose mistake tuning to offer another perspective investigating the potential benefits of utilizing mistakes to make LLMs better capable of reasoning. Mistake tuning incorporates the combinations of both correct references and incorrect rationales. We finetune two different sizes of Flan-T5 and reveal that tuning on the two domains leads to consistent improvement across various tasks compared with only tuning on the correct CoT rationale. Introducing prefixes [CORRECT RATIONALE] and [INCORRECT RATIONALE] before each corresponding rationale enables Flan-T5 to differentiate and contribute to making correct rationales. Our results have proven that learning from mistakes is beneficial to LLMs' reasoning in both prompting and finetuning.

We conduct comprehensive experiments and studies to prove the effectiveness of utilizing mistakes in LLMs' reasoning. Aligning self-rethinking to guide the PaLM2-540B model, there have been significant improvements with fewer computing resources observed in areas such as GSM8K (+6.75%) and LogiQA (+5.69%) compared to self-consistency. Finetuning Flan-T5-large (780M) with our principles of mistake tuning, there are notable improvements compared to finetuning on only correct rationales, such as GSM8K(+4.08%) and MathQA(+6.16%).

Overall, this study introduces the COTERRORSET, a comprehensive dataset of 609,432 questions with both correct and incorrect rationales across various domains. We underscore the benefits of learning from mistakes to further improve LLMs' reasoning abilities at low cost by proposing two strategies: self-rethinking prompting and mistake tuning, both of which have demonstrated remarkable improvements. These findings validate the utility of the COTERRORSET as a promising direction to advance LLMs' reasoning performance. Furthermore, we delve into COTERRORSET and conduct a comprehensive analysis of the errors made by CoT, as well as potential categorizations and reasons for those mistakes.

2 Related Work

Model Tuning. In the evolving landscape of Natural Language Processing (NLP), the concept of mistake tuning has emerged as a significant advancement in the instruction tuning of Large Language Models (LLMs). Our experiments with Flan-T5 demonstrate that finetuning models on a blend of correct and incorrect rationales, rather than solely on correct Chain-of-Thought (CoT) rationales, yields consistent improvements across a range of tasks. This approach marks a departure from traditional methods that mainly leverage human-crowdsourced tasks from sources like T0 (Sanh et al., 2021), FLAN (Wei et al., 2022), and NaturalInstructions (Mishra et al., 2021), or model-generated tasks. While human-crowdsourced tasks guarantee high quality, they are often limited in scope and require significant human labor. In contrast, model-generated tasks, which utilize the capabilities of advanced language models like PaLM2-540B, create extensive sets of instructions, inputs, and outputs from initial seed sets (Wang et al., 2022; Peng et al., 2023). Our approach, integrating insights from the COTERRORSET, applies mistake tuning to PaLM2-540B, aiming to improve the quality and expand the scope of instruction-following data by incorporating a nuanced understanding of both correct and incorrect reasoning processes. Although concurrent work (An et al., 2023) has done similar experiments on instruction tuning LLMs to learn why those rationales are incorrect with our mistake tuning, they make GPT4 as a teacher model to finetune the explanations for incorrect answers to downstream LLMs, which demands computational resources and associated costs.

CoT Rationales. In the realm of LLMs, the CoT prompting technique, particularly in zero-shot scenarios, has revolutionized complex reasoning by generating intermediary reasoning steps (Wei et al., 2022). This approach, initiated by simple prompts like "Let's think step by step", has been promising in enhancing the reasoning abilities of models like PaLM2-540B (Zhou et al., 2022b). Following this trend, Zelikman et al. (2022) employed GPT-J (Wang and Komatsuzaki, 2021) to produce rationales, selecting the most effective ones. Our study advances this concept using PaLM2-540B, focusing on complex logical reasoning and incorporating insights from the COTERRORSET to understand and correct reasoning errors.

LLMs with CoT Reasoning. The research conducted by (Wei et al., 2022) on the emergence of CoT reasoning in large models like PaLM2-540B has catalyzed new research directions. These capabilities have been explored across logical reasoning (Creswell et al., 2022; Pan et al., 2023; Lei et al., 2023), commonsense reasoning (Talmor et al., 2018; Geva et al., 2021; Ahn et al., 2022), and mathematical reasoning (Miao et al., 2021; Koncel-Kedziorski et al., 2016; Patel et al., 2021; Cobbe et al., 2021; Hendrycks et al., 2021). The exceptional performance of models like PaLM2-540B has motivated further exploration into optimizing CoT reasoning (Wang et al., 2022; Zhou et al., 2022a; Creswell and Shanahan, 2022; Li et al., 2023b; Lightman et al., 2023), particularly with a focus on error analysis and learning as emphasized by our COTERRORSET initiative.

Mathematical Reasoning. Considerable research efforts have been directed toward enhancing the capabilities of Large Language Models (LLMs) in solving mathematical problems. This enhancement has been approached from various innovative perspectives. Some studies have focused on employing voting or verification methods that utilize multiple reasoning paths to improve accuracy and reliability in solutions (Wang et al., 2022; Li et al., 2023b; Lightman et al., 2023). Another direction has involved the generation of executable programs or the integration of plug-in tools to enable the execution of external APIs during the reasoning process, thereby augmenting the LLMs' problem-solving capabilities (Jie and Lu, 2023; Wang et al., 2023a; Gou et al., 2023). Additionally, there has been a significant focus on data augmentation strategies. These include methods to expand the training datasets and provide external annotations, which enrich the LLMs' understanding and approach towards complex mathematical problems (Magister et al., 2022; Huang et al., 2022; Ho et al., 2022; Li et al., 2022; Yuan et al., 2023; Li et al., 2023a; Luo et al., 2023; Yu et al., 2023; Liang et al., 2023). Our work, in particular, leverages the MathQA benchmark of multiple-choice math problems (Amini et al., 2019). This benchmark provides a comprehensive and challenging set of mathematical problems, which serves as an excellent platform for refining and testing the enhanced problem-solving capabilities of LLMs through CoT reasoning and mistake tuning methods.

Logical Reasoning. Logical reasoning is a fundamental element in both human cognition and AI systems. Various methodologies have been pursued to enhance this capability in AI, including rule-based and symbolic systems (MacCartney and Manning, 2007), the finetuning of large language models (Wang et al., 2018), and a combination of neural and symbolic strategies (Li and Srikumar, 2019). This intricate, multi-step nature of logical reasoning tasks makes them suitable for CoT instruction tuning. Our work is novel in applying this technique to logical reasoning with PaLM2-540B, using a comprehensive dataset of reasoning chains from COTERRORSET to refine the model's reasoning abilities, thereby improving its performance in logical reasoning tasks.

3 A Novel Benchmark: COTERRORSET

In order to investigate whether incorrect rationales can also contribute to LLMs' reasoning performance, we introduce COTERRORSET, a novel benchmark based on the source of COT-COLLECTION (Kim et al., 2023), built upon various domains, including multiple-choice QA, extractive QA, closed-book QA, formal logic, natural language inference, and arithmetic reasoning. Those public available datasets are QASC (Khot et al., 2020), AQuA (Ling et al., 2017), GSM8K (Cobbe et al., 2021), QED (Lamm et al., 2021), StrategyQA (Geva et al., 2021), SenseMaking (Wang et al., 2019), CREAK (Onoe et al., 2021), e-SNLI (Camburu et al., 2018) and ECQA (Aggarwal et al., 2021). Each task within this collection is systematically organized to include a question and instruction section, followed by an answer paired with its golden rationale reference.

COTERRORSET diverges from traditional CoT datasets by employing PaLM2-540B's mistakes. We utilized PaLM2 to generate rationales for each question in the dataset, focusing specifically on collecting incorrect rationales. Then we provide PaLM2 with both correct references and its incorrect answers to demonstrate and reflect and demonstrate why it makes such mistakes. This systematic collection of incorrect rationales can make COTER-

RORSET a promising benchmark in providing future improvements from a different perspective.

4 Our Methodology

4.1 Self-rethinking Prompting

Self-rethinking introduces a unique strategy for prompting LLMs to rethink whether they have made the same previous mistakes. This process begins by presenting LLMs with several incorrect rationales and deducing the reasons for making such errors. The primary objective of this stage is to enable the LLM to introspectively deduce and categorize the nature of mistakes. For example, PaLM2-540B can recognize specific errors they made in GSM8K: application of percentage or ratio, making assumptions without basis, etc.

This explicit demonstration of errors, coupled with the question, golden reference, and incorrect rationales, is instrumental in enabling the LLM to recognize specific types of mistakes it tends to make. Following this, the LLM enters the forward reasoning phase, where it employs a CoT reasoning approach. Here, it logically progresses step-by-step to solve the problem, actively engaging in the reasoning process. The core of self-rethinking lies in the backward-checking stage. In this phase, the LLM reviews its reasoning chain, but with a specific focus on the error types it previously identified. This targeted review helps the LLM to not just correct random errors but to consciously avoid repeating the same types of mistakes it has made in the past. The process includes a loop for error correction and confirmation. If the LLM finds that it has repeated any of the previously identified mistakes, it revisits the reasoning process to correct them.

However, the iterative checking process should have a crucial repeating boundary, denoted as 'k' iterations. If the LLM's error-checking and correction cycles surpass this predefined threshold and errors still persist, the process concludes under the assumption that the issue at hand or the error detection might exceed the LLM's current capabilities. This constraint prevents the LLM from being caught in an endless loop of self-rethinking, ensuring the efficiency and practicality of the reasoning process. In this work, we set k equal to 1 in order to trade between the accuracy and computing resources.

Algorithm 1 self-rethinking

```
Correct & Incorrect Rationales = {...}
ErrorCounter ← 0
Prompt: Why you made the mistakes?
Mistakes ← Error Type, Demonstrations, Examples.
Stage1 Prompt: Let's think step by step.
Stage2 Prompt: Do you make the same mistakes in Mistakes?
while ErrorCounter < k do
    if Yes then
        go to Step2
        ErrorCounter ← ErrorCounter + 1
    else if No then
        get the answer
        break
    end if
end while
if ErrorCounter == k then
    Assume: Problem or error detection exceeds the model's capabilities.
end if
Prompt: So the final answer is:
```

4.2 Mistake Tuning

In order to fully investigate the potential utilization of incorrect rationales in COTERRORSET, we propose mistake tuning, instructing LLMs to memorize common mistakes, which can further improve their abilities to output correct rationales. By simply appending prefixes [CORRECT RATIONALE] and [INCORRECT RATIONALE] before corresponding rationales, mistake tuning is built upon the foundational conclusions of self-rethinking, where LLMs can differentiate the implicit reasons and types of mistakes they made in CoT reasoning. This process can be formulated as:

$$p = [Q \oplus S \oplus R], \tag{1}$$

$$\mathcal{L} = -\sum_{t=1}^{|pwod|} logP(p_t|p_{<t}), \tag{2}$$

Where Q, S and R represent the given question, special prefix and corresponding rationale respectively. $\oplus$ represents the operation of concatenation.

Mistake tuning presents a cost-effective, straightforward, and efficient alternative.

5 Experiments

In this section, we conducted a series of experiments to compare the proposed self-rethinking methods with the existing approach on both arithmetic and commonsense reasoning benchmarks.

5.1 Experiment Setup

We conduct comparisons between self-rethinking and several other baselines on multiple benchmarks.

Benchmarks: We consider the following existing math problems benchmarks designed with human rationale reference.

- GSM8K benchmark of math word problems (Cobbe et al., 2021).
- AQuA dataset of algebraic math problems (Ling et al., 2017).
- MathQA benchmark of multiple-choice math problems (Amini et al., 2019).
- Openbook benchmark modeled after open book exams for assessing human understanding of a subject (Mihaylov et al., 2018).
- LogiQA dataset sourced from expert-written questions for testing human logical reasoning (Liu et al., 2020).
- Critical Reasoning in MARB benchmark of several graduate admission tests, highlighting the reasoning to assumptions, conclusions and paradoxes in arguments (Tong et al., 2023).

Models: In order to evaluate self-rethinking's effects, we choose PaLM2-540B (Anil et al., 2023) and GPT4 (OpenAI, 2023) as the baseline model. PaLM2-540B is a dense left-to-right, decoder-only language model with 540 billion parameters. It is pre-trained on a high-quality corpus of 780 billion tokens with filtered webpages, books, Wikipedia, news articles, source code, and social media conversations. GPT4 is a large-scale multimodal SOTA model that exhibits human-level performance on various tasks.

For mistake tuning, we choose two different-sized Flan T5 (Chung et al., 2022), which are specifically designed for instruction tuning strategies. This model excels in understanding and generating human-like text, demonstrating remarkable performance across a wide range of natural language processing tasks. We choose the common settings(random seed=42, learning rate=1e-4) and finetune using the AdamW optimizer. Considering the vast number of data in AQuA, we only randomly select 10,000 of them to represent the differences in tuning on two different domains.

5.2 Self-rethinking Results

Table 1 presents PaLM2-540B's evaluation results on chosen benchmarks. The self-rethinking method shows superior performance with significant improvements, especially in GSM8K, AQuA, MathQA, and LogiQA, clearly outperforming self-consistency within the same computing budget. However, while our method surpasses CoT in performance on the OpenbookQA dataset, it falls short of achieving self-consistency results. This can be attributed to the nature of the tasks in this dataset, which are less focused on logical difficulty and more on assessing commonsense knowledge. Unlike the other datasets where logical reasoning and mathematical skills are paramount, OpenbookQA requires a strong understanding of general knowledge. Table 3 compares GPT4's performance of CoT and self-rethinking. The results demonstrate a notable improvement when using the self-rethinking method over CoT. These findings suggest that self-rethinking is a more effective approach for enhancing GPT-4's performance.

Table 2 presents the 8-shot examples of CoT and self-rethinking, using the PaLM2-540B model across four different tasks: GSM8K, AQuA, MathQA, and LogiQA. The experiment was designed with a common setting, employing a random seed of 42 and selecting 8-shot examples from the respective training sets. A key part of the process involved collecting PaLM2-540B's incorrect rationales for these examples, which were then used as learning demonstrations to rethink. The results show a clear advantage of the self-rethinking method over the standard 8-shot CoT approach. These results highlight the efficacy of the self-rethinking method in improving accuracy in few-shot learning scenarios for complex problem-solving tasks.

In conclusion, our self-rethinking method achieved remarkable accuracy improvements in most tests, particularly in scenarios that demand high logical rigor and offer the opportunity to learn from errors by identifying fixed logical patterns, especially in arithmetic reasoning tasks. It indicates self-rethinking effectiveness in tasks requiring strong logic and prone to minor errors. Additionally, the self-rethinking method proves partic-

Methods	GSM8K	AQuA	MathQA	OpenbookQA	LogiQA	CR
Standard (Kojima et al., 2022)	17.06	22.40	27.57	80.92	41.21	24.45
CoT (Wei et al., 2022)	56.29	32.11	30.89	82.66	41.05	51.98
Self-consistency (Wang et al., 2022)	58.38	42.80	41.37	**87.61**	42.88	22.58
Self-rethinking (Ours)	**65.13**	**44.72**	**43.95**	85.71	**49.12**	**54.53**

Table 1: PaLM2-540B's accuracy on Standard Prompting(Standard) (Kojima et al., 2022), Chain-of-Thought Prompting(CoT) (Wei et al., 2022), self-consistency (Wang et al., 2022) and our methods, self-rethinking prompting. In this experiment, we set the times of inference in self-consistency to 3, aligning the computing budget with our method. Our approach involves an initial zero-shot CoT inference, then rethinking whether this rationale has made similar errors. This leads to the final answer if no errors are found. If inaccuracies are detected, it combines a demonstration and the previously suspected erroneous answer for a third inference to arrive at the final answer. Hence, the overall inference times in our methods are between 2 and 3 times per question, which is still lower than self-consistency here.

Methods	GSM8K	AQuA	MathQA	LogiQA
8-shot CoT	64.56	30.65	36.21	29.57
8-shot self-rethinking	**70.15**	**34.80**	**40.56**	**33.64**

Table 2: PaLM2-540B's accuracy results on few-shot Chain-of-Thought(CoT) and our methods, self-rethinking. We select 8-shot examples from the corresponding trainset. Then we collect PaLM2-540B's incorrect rationales of those 8 examples. The part of the original correct reference is CoT's demonstrations. Those generated incorrect rationales serve as demonstrations for the rethink stage.

Methods	GSM8K	AQuA	OpenbookQA	CR
CoT	97.93	88.98	93.21	78.92
self-rethinking	**98.02**	**91.03**	**95.07**	**81.37**

Table 3: GPT4' results on zero-shot Chain-of-Thought(CoT) and our methods, self-rethinking.

Models	Methods	GSM8K	MathQA	AQuA
Flan-T5-large (780M)	Standard Finetuning	14.28	42.79	13.10
	Mistake Tuning	**18.36**	**48.95**	**18.07**
Flan-T5-xl (3B)	Standard Finetuning	23.81	47.24	17.81
	Mistake Tuning	**24.29**	**52.22**	**20.99**

Table 4: Accuracy of Standard Finetuning models (with only correct rationales) vs. our methods, mistake tuning (combined correct and incorrect rationales). Mistake tuning shows consistent and superior performance compared with only fine-tuned correct CoT rationales.

ularly beneficial in assisting LLMs in identifying and rectifying low-level mistakes or misunderstandings that are within the model's capabilities but have been previously overlooked. This capability indicates that self-rethinking can serve as a valuable tool in refining the accuracy and reliability of responses in LLMs, especially in complex problem-solving contexts.

5.3 Mistake Tuning Results

Table 4 showcases the performance of Flan-T5 models in the context of mistake tuning, highlighting the impact of combining correct and incorrect rationales. The data presented in Table 4 reveals significant insights into the performance of Flan-T5 models under mistake tuning, which involves integrating both correct and incorrect rationales. This approach is evident across different model scales, whether it's the smaller 780M version or the larger 3B variant. Notably, in the MathQA domain, Flan-T5-large(780M) tuned by our methods demonstrates superior performance compared to PaLM2-540B, achieving an accuracy of 48.95% versus 41.37%. This phenomenon suggests that LLMs can benefit from engaging with incorrect reasoning, thereby enhancing their problem-solving and reasoning capabilities. It extends beyond merely bolstering the model's grasp of correct CoT, to also encompassing the ability to identify and learn from incorrect rationales.

Furthermore, the expense of obtaining ground truth or hand-crafted references is significantly higher compared to generating and collecting incorrect rationales. This cost disparity underscores the practical value of our approach, offering a more cost-effective solution without compromising the quality of training data for machine learning models. All mentioned provides a direction for further work of reasoning, which involves not only enhancing the model's understanding and learning of correct CoT but also the ability to identify and learn from incorrect rationales.

6 Further Studies

6.1 Hyperparameter Analysis of Rethinking Iteration Times

In this section, we conduct experiments to assess the impact of different rethinking iterations, denoted as k, on the performance of our framework. We evaluate it on two mainstream benchmarks in the field of mathematics and commonsense reasoning, GSM8K and LogiQA. Figure 2 represents the detailed trend under varying re-thinking times. Notably, as k increases from 1 to 24, GSM8K represents a growth of 8.11% and 12.37% in LogiQA. It is evident as k increases, both LLMs' arithmetic and commonsense reasoning accuracy exhibit an upward trend. This trend suggests a positive correlation between the number of rethinking iterations and the overall reasoning abilities. These observations indicate self-thinking's potential benefits with more inference time.

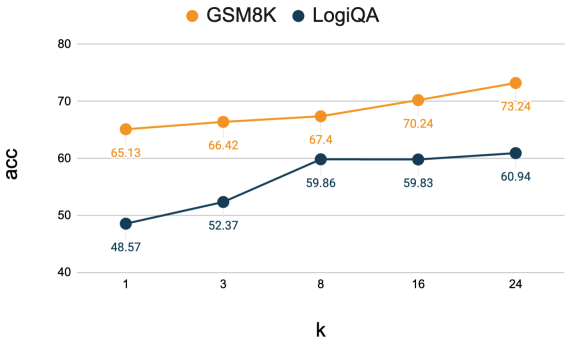


Figure 2: Accuracy of different re-thinking iterations(k). As the value of k increases, the overall prediction accuracy improves.

6.2 Ablation Study on Rethinking Process

In this ablation study, we examined the impact of various component combinations in promptings to guide LLMs to self-rethink. Table 5 shows the performance of different components. The results indicate that the inclusion or exclusion of different components has varying effects on PaLM2-540B's accuracy in domains of GSM8K and LogiQA. However, the overall performance across various components is relatively similar. It performs similarly well regardless of the specific combination of components, indicating good generalizability of the method. This study suggests our method's flexibility and stability in future usage.

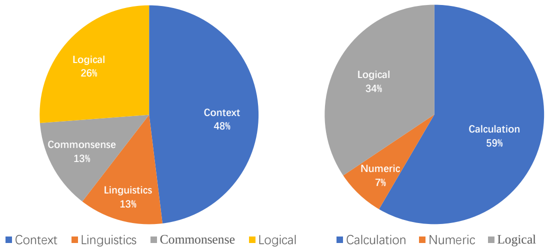
7 Unveiling LLM's Reasoning Errors

In this section, we delve into the detailed types and underlying reasons that lead to mistakes in LLMs's inference process. We sample mistake examples from GSM8K and LogiQA to conduct an in-depth analysis of both arithmetic and commonsense reasoning. We put some examples in Appendix A.

For commonsense reasoning, we find errors like the misinterpretation of facts or concepts usually arise due to the model's limitations in understanding and applying context accurately. This reveals current LLMs may still fall short of consistently recalling precise factual knowledge within a given context. Consequently, this underscores the imperative to advance toward the development of Retrieval-Augmented Generation(RAG) systems (Guu et al., 2020; Mallen et al., 2022), as they hold the promise of yielding more faithful and contextually aligned results. Additionally, errors stemming from logical fallacies or incorrect inferences reveal LLMs' reliance on pattern recognition over logical reasoning, sometimes leading them to make logically inconsistent or unsupported connections by the given facts.

CAT.	DEM.	COR.	INC.	**GSM8K**	**LogiQA**
✓				64.30	50.21
✓	✓			62.70	48.57
✓	✓	✓		65.70	51.01
✓	✓	✓	✓	65.13	49.21

Table 5: Impact of Component Combinations. CAT. stands for the previous mistakes' type name, DEM. are the reasons for making such mistakes, and COR. and INC. mean corresponding correct and incorrect rationale examples. All components here are generated by LLM itself before reasoning.



(a) Commonsense Reasnoing (b) Arithmetic Reasoning

Figure 3: PaLM2's error type distribution in the commonsense and arithmetic reasoning task.

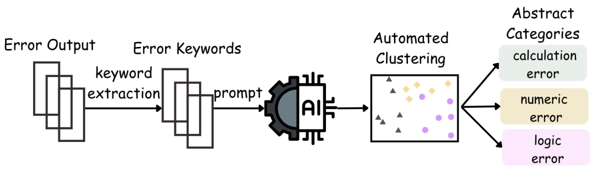


Figure 4: Our pipeline for clustering PaLM2-540B's mistakes.

Concerning arithmetic reasoning, we observe that the error types identified by LLMs are notably more intricate and diverse compared to those associated with commonsense reasoning. The complexity presents challenges in conducting a thorough analysis. Unfortunately, little previous work has explored what kinds of mistakes LLMs make during inference. In order to tackle this issue and gain a more overarching understanding of LLMs' error types, we utilize an LLM-based clustering mechanism shown in Figure 4 to match diverse error types into more general categories. To be specific, we begin by extracting the specific error type names from each output of LLM. Subsequently, we input all the extracted names into the LLMs and prompt them to generate more general categories that encompass the entire spectrum of error names. Following this automated clustering process, we meticulously review each cluster, making necessary adjustments to refine the matching results. Finally, we distill the diverse error types into several abstract categories, such as calculation error, numeric error, and logical error in domains of arithmetic reasoning. Detailed content is shown in Appendix B.

As shown in Figure 3, the most errors made by LLMs in arithmetic reasoning are about calculation. This can be attributed to the different nature of LLMs compared to other tools like calculators. To address this issue, Chen et al. (2022)'s suggestion using Program-of-Thought (PoT) is a promising approach to instruct LLMs to generate a segment of code to solve the given problem, resulting in more accurate calculation results. Furthermore, it's important to note that logical error is also a type of error that LLMs always suffer from. Compared with calculation errors and numeric errors, the causes of logical errors are more complicated and nuanced. For instance, errors like misinterpreting given data or misapplying arithmetic operations reveal a lack of depth in understanding mathematical relationships. This can result from the model's limitations in comprehending the nuances of mathematical concepts or its inability to correctly infer the needed function from the context of the question. In the future, more fine-grained analysis and methods are needed to address such complex logical errors in arithmetic reasoning.

For commonsense reasoning, results in Figure 3 demonstrate that context understanding is the most problematic issue. This problem is intricate and difficult to tackle since this can result from the model's limitations in comprehending the nuances of context information and the context's relationship with its own knowledge. Wang et al. (2023b) aligns reward models to enhance LLMs' understanding of context information and successfully enhances the model's reasoning abilities, which is consistent with our findings. Table 6 demonstrates one example where our methods guide PaLM2-540B to change their understanding of the context based on provided mistake examples.

True or False? When one consumes the leg meat of a swine, they are eating Ham.
CoT Reasoning: The leg meat of a swing is typically referred to as pork. Ham is specifically the cured or smoked hind leg of the pig. So the answer is False.
self-rethinking(our method): **Step1** ... So the answer is False. **Step2** Do I make the same mistakes in those examples:...? **Step3** Swine is pig and ham comes from pig's legs. The context here should not be so rigorous. So the final answer is: True

Table 6: An example where self-rethinking guides PaLM2-540B to learn from mistakes to better understand the context.

8 Conclusions and Future Work

In this work, we explore whether LLMs can learn from their mistakes. In order to investigate LLMs' abilities to differentiate and learn from mistakes, we introduce CoTErrorSet, a novel benchmark collecting both correct and incorrect CoT rationales across various domains and designed with demonstrations for making errors. We propose two possible solutions to expose the effects of mistakes from different perspectives: self-rethinking and mistake tuning. Both of them have achieved consistent and significant improvements, which demonstrates the potential benefits of learning from reasoning errors. In the last, we conduct a comprehensive and detailed analysis of LLMs' common mistakes in both arithmetic and commonsense reasoning. The findings will provide a clear direction for future improvements.

For future work, we envision proposing corresponding algorithms or loss functions to learn implicit information from mistakes. The primary intent of this work is to provide a new paradigm so there are still a lot of improvements that can be down following this work. For example, incorporating contrastive learning to differentiate correct references and errors is intuitive to make more improvements. Also, some memorization and retrieval-augmented skills can help models benefit from mistakes similar to each question.

Limitations

It is a pity that we can not do very large-scale mistake tunings considering the computational resources. As the scaling effects proposed in existing previous work, models over 100B tend to have leap effects after instruction tuning in the reasoning domain. Moreover, we surprisingly find that most LLMs' end-end unsupervised clustering abilities, especially API-based LLMs are still underexplored. This work still lacks large-scale clustering for all errors in COTERRORSET to broadly investigate and analyze error types.

References

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for commonsenseqa: New dataset and models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3050–3065.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2023. Learning from mistakes makes llm better reasoner. *arXiv preprint arXiv:2310.20689*.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

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

Antonia Creswell and Murray Shanahan. 2022. Faithful reasoning using large language models. *arXiv preprint arXiv:2208.14271*.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Minlie Huang, Nan Duan, Weizhu Chen, et al. 2023. Tora: A tool-integrated reasoning agent for mathematical problem solving. *arXiv preprint arXiv:2309.17452*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2022. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.

Po-Wei Huang. 2022. Domain specific augmentations as low cost teachers for large students. In *Proceedings of the first Workshop on Information Extraction from Scientific Publications*, pages 84–90, Online. Association for Computational Linguistics.

Zhanming Jie and Wei Lu. 2023. Leveraging training data in few-shot prompting for numerical reasoning. *arXiv preprint arXiv:2305.18170*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090.

Seungone Kim, Se June Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. 2023. The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. *arXiv preprint arXiv:2305.14045*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 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*, pages 1152–1157.

Matthew Lamm, Jennimaria Palomaki, Chris Alberti, Daniel Andor, Eunsol Choi, Livio Baldini Soares, and Michael Collins. 2021. Qed: A framework and dataset for explanations in question answering. *Transactions of the Association for computational Linguistics*, 9:790–806.

Bin Lei, Chunhua Liao, Caiwen Ding, et al. 2023. Boosting logical reasoning in large language models through a new framework: The graph of thought. *arXiv preprint arXiv:2308.08614*.

Chengpeng Li, Zheng Yuan, Guanting Dong, Keming Lu, Jiancan Wu, Chuanqi Tan, Xiang Wang, and Chang Zhou. 2023a. Query and response augmentation cannot help out-of-domain math reasoning generalization. *arXiv preprint arXiv:2310.05506*.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, et al. 2022. Explanations from large language models make small reasoners better. *arXiv preprint arXiv:2210.06726*.

Tao Li and Vivek Srikumar. 2019. Augmenting neural networks with first-order logic. *arXiv preprint arXiv:1906.06298*.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023b. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333.

Zhenwen Liang, Wenhao Yu, Tanmay Rajpurohit, Peter Clark, Xiangliang Zhang, and Ashwin Kaylan. 2023. Let gpt be a math tutor: Teaching math word problem solvers with customized exercise generation. *arXiv preprint arXiv:2305.14386*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

Bill MacCartney and Christopher D Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2022. Teaching small language models to reason. *arXiv preprint arXiv:2212.08410*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*.

Neil Mercer. 2008. Talk and the development of reasoning and understanding. *Human development*, 51(1):90–100.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2021. A diverse corpus for evaluating and developing english math word problem solvers. *arXiv preprint arXiv:2106.15772*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

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

Yasumasa Onoe, Michael JQ Zhang, Eunsol Choi, and Greg Durrett. 2021. Creak: A dataset for commonsense reasoning over entity knowledge. *arXiv preprint arXiv:2109.01653*.

OpenAI. 2023. Gpt-4 technical report.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. *arXiv preprint arXiv:2305.12295*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Taly Reich, Alex Kaju, and Sam J Maglio. 2023. How to overcome algorithm aversion: Learning from mistakes. *Journal of Consumer Psychology*, 33(2):285–302.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Yongqi Tong, Yifan Wang, Dawei Li, Sizhe Wang, Zi Lin, Simeng Han, and Jingbo Shang. 2023. Eliminating reasoning via inferring with planning: A new framework to guide llms' non-linear thinking. *arXiv preprint arXiv:2310.12342*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Ben Wang and Aran Komatsuzaki. 2021. Gpt-j6b: A 6 billion parameter autoregressive language model. https://github.com/kingoflolz/mesh-transformer-jax.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019. Does it make sense? and why? a pilot study for sense making and explanation. *arXiv preprint arXiv:1906.00363*.

Ke Wang, Houxing Ren, Aojun Zhou, Zimu Lu, Sichun Luo, Weikang Shi, Renrui Zhang, Linqi Song, Mingjie Zhan, and Hongsheng Li. 2023a. Mathcoder: Seamless code integration in llms for enhanced mathematical reasoning. *arXiv preprint arXiv:2310.03731*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yikun Wang, Rui Zheng, Haoming Li, Qi Zhang, Tao Gui, and Fei Liu. 2023b. Rrescue: Ranking llm responses to enhance reasoning over context. *arXiv preprint arXiv:2311.09136*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022a. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022b. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

A Reasoning Mistake Examples

B Definition for Error Categories

Table 7: Examples of Error Types in Arithmetic Reasoning. All contents are generated by PaLM2-540B itself.

Error name: Misinterpretation of Given Data
Error type: Logical

- *Example:* Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?
- *Correct Answer:* Natalia sold 48/2 = 24 clips in May. Natalia sold 48+24 = 72 clips altogether in April and May.
- *Incorrect Rationale:* Natalia sold 48 * 2 = 96 clips in May. Natalia sold 48+96 = 144 clips altogether in April and May.
- *Demonstration:* Mistaking multiplication for division led to a significant overestimate of the total clips sold.

Error name: Misapplication of Arithmetic Operation
Error type: Calculation

- *Example:* Weng earns $12 an hour for babysitting. Yesterday, she just did 50 minutes of babysitting. How much did she earn?
- *Correct Answer:* Weng earns 12/60 = $0.2 per minute. Working 50 minutes, she earned 0.2 x 50 = $10.
- *Incorrect Rationale:* Weng earns 12/60 = $2 per minute. Working 50 minutes, she earned 2 x 50 = $100.
- *Demonstration:* Confusing the rate per hour with the rate per minute led to a substantial overestimation of earnings.

Error name: Numerical
Error type: Numeric

- *Example:* The chicken crossed the road to get to the other side twice for the thrill of it. The first time, it had to dodge 23 speeding cars. The second time, a person tried to catch it and accidentally pulled out twice as many feathers as the number of cars the chicken had dodged. The chicken had 5263 feathers before its thrill-seeking road crossings. How many feathers did it have afterward?
- *Correct Answer:* The chicken lost 23 * 2 = «23*2=46»46 feathers on its second road crossing., it had 5263 - 46 = «5263-46=5217»5217 feathers after crossing the road twice.
- *Incorrect Rationale:* The chicken lost 23 * 2 = «23*2=46»46 feathers on its second road crossing., it had 5263 - 46 = «5263-52=5211»5211 feathers after crossing the road twice.
- *Demonstration:* 1. The correct answer is 5217, while your answer is 5211. 2. Your answer is wrong because you subtracted 52 instead of 46. 3. The type name of the incorrect answer is numerical.

Table 8: Examples of Error Types in Commonsense Reasoning. All contents are generated by PaLM2-540B itself.

Error name: Logical Fallacy or Incorrect Inference
Error type: Logical

- *Example:* "When standing miles away from Mount Rushmore"
- *Correct Rationale:* Objects appear smaller when viewed from a greater distance.
- *Incorrect Rationale:* "The mountains do not look smaller when standing miles away from Mount Rushmore. They look larger." (Logical fallacy)
- *Demonstration:* 1. The correct rationale is that objects appear smaller when viewed from a greater distance, whereas the incorrect rationale states the opposite. 2. This is a logical fallacy as it contradicts a basic principle of perception. 3. The type name of the incorrect rationale is logical.

Error name: Incorrect Assumptions or Generalizations
Error type: Logical

- *Example:* "Poison causes harm to which of the following?"
- *Correct Rationale:* Poison affects living organisms.
- *Incorrect Rationale:* "Robot do not get hurt by poison." (Incorrect generalization about the effects of poison)
- *Demonstration:* 1. The correct rationale is that poison affects living organisms, but the incorrect rationale generalizes that robots are immune to poison. 2. This is an incorrect generalization because robots, being non-living entities, are not subject to biological effects. 3. The type name of the incorrect rationale is logical.

Error name: Misunderstanding Literal vs. Metaphorical Language
Error type: Linguistics

- *Example:* "When food is reduced in the stomach"
- *Correct Rationale:* Digestion involves the breakdown of food by stomach acid.
- *Incorrect Rationale:* "Choice D is incorrect because it is not a fact." (Misunderstanding metaphorical language)
- *Demonstration:* 1. The correct rationale is about the literal process of digestion, whereas the incorrect rationale misinterprets the metaphorical language. 2. This demonstrates a misunderstanding of metaphorical language. 3. The type name of the incorrect rationale is linguistics.

Error name: Incorrect Application of Knowledge
Error type: Commonsense

- *Example:* "Stars are"
- *Correct Rationale:* Stars are massive celestial bodies made of gases.
- *Incorrect Rationale:* "Stars are not made of warm lights that float." (Incorrectly applying knowledge about stars)
- *Demonstration:* 1. The correct rationale states that stars are massive celestial bodies made of gases, but the incorrect rationale describes them as warm lights that float. 2. This is an incorrect application of knowledge, as it fails to accurately describe the nature of stars. 3. The type name of the incorrect rationale is commonsense.

Error name: Factual Inaccuracy
Error type: Commonsense

- *Example:* "You can make a telescope with a"
- *Correct Rationale:* A telescope requires specific optical elements to function.
- *Incorrect Rationale:* "A telescope needs a lens and a magnifying glass is a lens, so glass is a good choice." (Factually inaccurate about how telescopes are made)
- *Demonstration:* 1. The correct rationale is that a telescope requires specific optical elements, whereas the incorrect rationale assumes any lens, like a magnifying glass, can make a telescope. 2. This shows a factual inaccuracy in understanding how telescopes are constructed. 3. The type name of the incorrect rationale is commonsense.

Error type: Misunderstanding Context or Relevance
Error type: Context

- *Example:* "an inherited characteristic found on all mammals is"
- *Correct Rationale:* Inherited characteristics in mammals include features like fur.
- *Incorrect Rationale:* "Shoes are not found on all mammals" (Misunderstanding the context of biological characteristics)
- *Demonstration:* 1. The correct rationale focuses on relevant inherited physical traits like fur. 2. This error illustrates a clear lack of understanding of the context. 3. The type name of the incorrect rationale should be context.

Table 9: PaLM2-540B's Understanding and Definitions for Error Types. All contents are generated by itself after providing its mistakes and corresponding golden-standard references.

Error Type	Definition
Calculation	Mistakes or inaccuracies that occur during the process of performing mathematical calculations. These errors can arise from various sources and can occur at any stage of a mathematical problem-solving process.
Numeric	Numeric errors in the context of mathematical reasoning refer to inaccuracies that arise from the representation and manipulation of numerical values. These errors can occur at various stages of mathematical computations and can result from limitations in the precision of the representation of real numbers or mistakes in handling numerical data.
Logical	Logical errors involve mistakes in the overall reasoning or strategy used to solve a mathematical problem. This type of error may not be immediately apparent during the calculation process but can lead to incorrect final results. It could include using an incorrect formula or assumptions, misunderstanding the problem statement, or applying the wrong concept.
Linguistics	Errors in linguistics involve inaccuracies or mistakes in the use of language. These can include grammatical errors, misuse of vocabulary, incorrect syntax, or problems in semantics. Linguistic errors may arise from a lack of understanding of the rules of a language, misinterpretation of meaning, or the inability to effectively convey a message in a given language. Such errors can affect the clarity, coherence, and overall effectiveness of communication.
Commonsense	Commonsense errors refer to mistakes or inaccuracies that occur in the application of general world knowledge or everyday reasoning. These errors can arise from misconceptions, flawed logic, or misunderstandings of basic principles that are widely accepted as common knowledge. Commonsense errors often lead to conclusions or decisions that, upon closer examination, are illogical or inconsistent with general understanding of the world.
Context	Errors of misunderstanding context or relevance occur when there is a failure to correctly interpret or apply the relevant information in a given scenario. This type of error typically involves overlooking key aspects of a context, making inappropriate generalizations, or failing to distinguish between literal and metaphorical language. These errors can significantly alter the intended meaning or relevance of a response in reasoning tasks.