Problematic Question Detection for Supporting Math Word Problem Design

Anonymous ACL submission

Abstract

When designing math word problems, teachers need to ensure the question is clear and precise. This is crucial to prevent ambiguity and misinterpretations, which can otherwise compromise consistent grading and teaching effectiveness. To help teachers overcome these hurdles, we offer support in designing math word problems that are clear, unambiguous, and formally structured. In this paper, we introduce MathError, a dataset of real-world math word problems with annotated error types, specifically to highlight the necessity of question correction. Our work explores the opportunities and challenges of using large language models (LLMs) to detect problematic questions for supporting math word problem design in real-world scenarios with limited data. We propose a framework for prompt refinement through self-optimization (PRO), a method by which to detect question descriptions that are ambiguous, informal, or that lack problem-solving information. A reflection mechanism is introduced to enhance the reasoning ability of LLMs by refining error type definitions and selecting proper examples for few-shot prompting. Experimental results show that PRO yields measurable gains in detecting error types. Nevertheless, this task remains challenging due to the diversity of linguistic expressions and the presence of implicit assumptions requiring contextual reasoning.

1 Introduction

When coming up with exam questions, teachers must ensure that the questions are clear and precise. This prevents students from misunderstanding the questions, which could lead to inconsistent grading standards and render the questions ineffective. This is particularly important for math word problems where there should usually be only one correct answer. However, when questions are formulated, there may be blind spots or minor oversights that lead to misinterpretations by students, or a lack of crucial details, making immediate comprehension difficult.

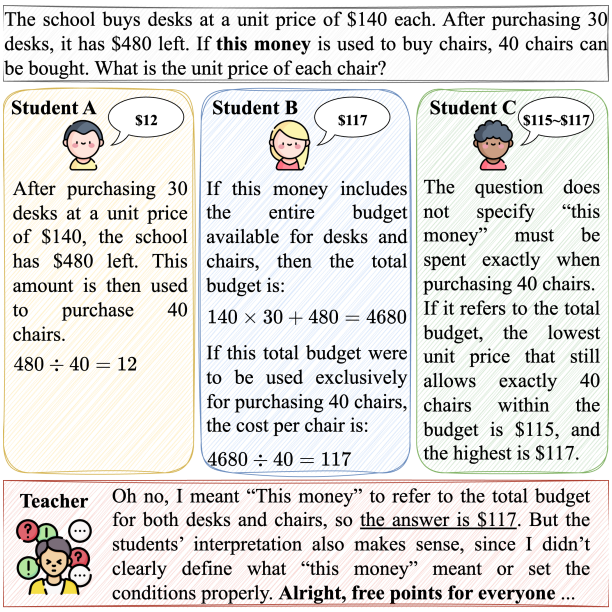

Figure 1: Example of how ambiguous wording can lead to multiple valid interpretations and answers.

Taking the question in Figure 1 as an example, the issue in this question lies in the interpretation of the phrase "this money", which can be understood in two different ways:

(1) "This money" refers to the remaining amount after purchasing the desks. Student A interprets it as the remaining amount after purchasing the desks, yielding a unit price of $12 per chair.

(2) "This money" refers to the total budget allocated for purchasing both desks and chairs. Student B interprets it as the total budget for both desks and chairs, leading to a unit price of $117. Student C likewise interprets it as the total budget but emphasizes that the problem statement does not require the budget to be exhausted exactly when purchasing 40 chairs. Under this reading, any price in the interval $115–$117 is admissible.[1]

[1] A unit price of $114 or lower is excluded, since at that

This question originates from the publicly available dataset Math23K (Wang et al., 2017).[2] The intended interpretation by the question designer assumes "this money" refers to the total budget and that it is exactly sufficient to purchase 40 chairs, leaving no remaining funds. However, students can arrive at different answers based on various reasonable interpretations and can justify their reasoning logically, making the problem invalid or contentious.

Additionally, teachers may also make errors like missing or extra words when designing questions. These do not cause ambiguity but make the question less formal. The importance of detecting such issues is evident in real-world contexts. For example, in the 2023 VCE exams in Australia,[3] multiple errors in math questions, such as incorrect units, missing terms, and mislabeled graphs, resulted in invalidated questions, in-exam corrections, and fairness concerns. Monash University mathematics professor Burkard Polster emphasized that such errors can cause students to waste time and lose focus during exams, and that awarding marks to all students is not truly fair because only those who attempted the flawed question experienced disrupted performance. These controversies illustrate the tangible consequences of imprecise or misleading language in assessment settings, and indicate the importance of early detection and quality control in question design.

Hence, we seek to support teachers in refining the clarity and precision of math word problem statements. This issue is language-dependent, as different languages convey conditions, value relationships, and precision differently. In the previous example, using the past perfect tense would indicate that the desks had already been bought, leaving only $480. However, tense usage varies across languages, and the complexity of error types depends on the difficulty of the math problems. To this end, this pilot study focuses on elementary-level math word problems in Chinese.

Similar issues have explored (Sun et al., 2024), emphasizing the importance of detecting problematic math word problems by constructing a dataset of unanswerable questions with predefined error types. However, their study focuses on error types specifically created for their experiment, our work shifts towards detecting naturally occurring errors in real math word problems. We extend the existing Chinese math word problem dataset—Math23K—with error type annotations.

Detecting such issues in natural language problem statements is challenging due to the diversity of linguistic expressions, implicit assumptions, varied contextual cues, and the need for both semantic understanding and mathematical reasoning. Recently, large language models (LLMs) have shown strong abilities in both semantic understanding and mathematical reasoning, motivating us to explore their capability in detecting problematic math word problems. Moreover, in real-world settings, there may be a lack of large annotated datasets for training, and even suitable examples for constructing effective few-shot prompts for LLMs may be scarce. We further investigate a self-optimizing approach that allows the model to learn from its mistakes, making effective use of the limited available data to capture the task as fully as possible. To sum up, the contributions of our work are threefold:

1. We introduce the task of detecting errors in math word problems to help teachers ensure clarity in question statements.
2. We present the MathError dataset,[4] which is designed for detecting errors in the statements of math word problems, to facilitate the investigation of the need for correcting problematic questions.
3. We explore a self-optimizing framework where the model iteratively refines its instructions. This approach simulates real-world scenarios where data is scarce by utilizing only a few examples, offering a preliminary solution to the challenge of error detection in math word problems. Experimental results show that the prompts refined by our reflection mechanism yield better performance.

2 Related Work

LLMs for Education Given the remarkable capabilities of large language models (LLMs) in language understanding and generation, recent studies use LLMs to generate test questions for student practice (Gonzalez et al., 2023; Feng et al., 2024; Song et al., 2023). The pedagogical ability of LLMs in mathematics education has also been

price the budget would suffice for 41 chairs rather than exactly 40.

[2]The provided rationale is presented Appendix A.

[3]https://en.wikipedia.org/wiki/2023_VCE_exam_period_controversies

[4]The dataset and code will be released upon acceptance. Sample data is available at: https://anonymous.4open.science/r/MathError-5276/.

studied (Yen and Hsu, 2023; Wang et al., 2024a; Daheim et al., 2024). Some works have investigated the role of LLMs in assisting teachers with tasks such as for math problem generation (Feng et al., 2024; Lee et al., 2024; Hunter McNichols et al., 2024; Christ et al., 2024). Liu et al. (2023) have shown that modeling ambiguity remains a significant challenge for LLMs, reinforcing the importance of developing methods to detect and address these problems. However, the capability of LLMs to recognize errors in math word problems and disentangle potential meanings is rarely explored.

Ambiguous Questions There are several types of ambiguity: lexical, syntactic, semantic, pragmatic, and anaphoric (Li et al., 2024). Numerous works address disambiguation using methods such as syntactic and semantic parsing (Tanaka et al., 2007; Koller et al., 2008) or coreference resolution (Kocijan et al., 2019). In question-answering applications, ambiguous queries lead to unanswerable queries. Methods have been developed to identify question answerability (Zhang et al., 2021; Yang et al., 2019) and generate clarification questions (Zamani et al., 2020; Krasheninnikov et al., 2022) or correct unanswerable questions (Yen et al., 2021). There has also been growing interest in addressing ambiguity in math problems. Sun et al. (2024) define five different categories of unanswerable questions. Curated annotators modified answerable questions into unanswerable ones based on the categories. By contrast, we construct a dataset by annotating error types of real-world questions. Additionally, we investigate more nuanced issues, such as unintended yet reasonable interpretations that lead to multiple answers, ultimately rendering the problem invalid.

Self-Optimization with LLMs LLMs have made significant advancements in producing coherent text and following given instructions (Wei et al., 2022; Ouyang et al., 2022). Recently, methods have been investigated that elicit feedback from LLMs on self-generated solutions, enabling iterative improvement of outputs based on the feedback. Several studies explore the use of LLMs for optimizing prompts.

Zhou et al. (2022) employ the LLM to create instructions, select the proper instructions based on accuracy, and instruct the LLM to generate a semantically similar variant. Pryzant et al. (2023) propose an approach to guide the LLM to provide textual feedback on how to revise an existing instruction at each step.

Methods have also been developed to use natural language feedback generated by LLMs to refine the model's output. Self-Debugging (Chen et al., 2023) is a method for code generation in which the model produces an initial program, explains its own errors in natural language, optionally incorporates execution results, and iteratively revises the code. Moral self-correction at test time (Ganguli et al., 2023) prompts models to first produce free-form critiques that invoke normative principles before generating the final response, thereby reducing unsafe or biased content in safety evaluations. Reflexion (Shinn et al., 2024) is an agent framework in which the model writes natural language reflections on its own failures or on external feedback, stores these reflections in episodic memory, and conditions subsequent attempts on this memory. Self-Refine (Madaan et al., 2024) is a framework in which the model generates task-specific feedback along dimensions such as factuality, helpfulness, and style, and then produces a revised output based on this feedback. For verification-oriented refinement, Chain-of-Verification (Dhuliawala et al., 2024) generates targeted verification questions, answers them independently, and integrates results to reduce hallucinations in knowledge-intensive tasks. For reasoning structure discovery, Self-Discover (Zhou et al., 2024) lets the LLM select and compose atomic reasoning modules for complex reasoning tasks PromptAgent (Wang et al., 2024b) frames prompt generation as a planning problem solved with Monte Carlo Tree Search, and PromptBreeder (Fernando et al., 2024) evolves both task prompts and mutation prompts to self-improve across generations. Inspired by these studies, we introduce a reflection mechanism to our framework to refine the LLM's prompts in detecting problematic questions in mathematics.

3 Dataset Construction

3.1 Error Type Definition

Identifying all math word problem error types immediately is challenging. To establish an initial set of error types, we randomly sampled 200 questions from Math23K, referred to as the initial set, and categorized the errors present in these questions.[5] We conducted a preliminary annotation of problematic descriptions, after which we consolidated these initial error types by merging similar ones. This process resulted in the definition of five error types:

[5] Math23K comprises 23,162 math word problems.

1. Interpretation Ambiguity (*INTPN*): The question appears formal and reasonable on the surface, but closer examination reveals that it may allow multiple interpretations which deviate from the intended meaning, leading to divergent yet reasonable answers.
2. Informal Wording (*Informal*): The wording of the question is informal. This may include missing or redundant words, unnecessary symbols, or awkward phrasing.
3. Unitless (*Unit*): The question does not specify the unit for the answer, which may lead to confusion about what measurement is expected.
4. Unclear Relationship (*Rel*): The question does not clearly define the logical relationship between numerical values, leading to potential misinterpretation.
5. Calculation Error (*Calc*): The problem uses imprecise words to describe a mathematical expression, for instance, making it unclear whether to perform multiplication or division before addition or subtraction; this can cause students to calculate in the wrong order.

If a question exhibits none of the issues mentioned above, it belongs to the *None* type. To ensure the completeness of these error definitions and the quality of the annotation, we established an iterative refinement annotation process.

3.2 Iterative Refinement Annotation

We invited native Chinese speakers as annotators, all of whom were from universities in the same country as the authors. Before beginning the annotation, we explained the purpose of the data and confirmed the compensation terms with the annotators. Compensation was calculated based on the minimum wage regulations of the authors' country; both parties agreed to these terms.

Three annotators were invited and we split the entire dataset into four parts, with each person responsible for the labeling of 5,815 samples.[6] We labeled one of these parts by ourselves. Unlike large-scale crowd annotation approaches, we relied on a small, trusted group of annotators with mathematical backgrounds. To ensure consistency and quality across the dataset, we conducted a quantitative evaluation of the annotators' labeling accuracy using our pre-annotated set of 100 samples, which we refer to as the *golden set*. Of these, 23 samples contained math word problems with error statements, whereas 77 samples had no errors. To evaluate the correctness of the annotators' labels, we divided the 100 samples from the golden set into five subsets, each containing 20 samples, ensuring a balanced representation of error types. These golden sets were inserted into the subsets that the annotators were to label. The annotation process was as follows.

[6]The annotation guideline is presented in Appendix B.

	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Overall
P	0.4546	0.7692	0.8335	0.9286	1.0000	0.7972
R	0.4166	0.4167	0.5556	0.5417	0.6000	0.5061
F	0.4348	0.5405	0.6667	0.6842	0.7500	0.6153

Table 1: Annotation agreement across five stages. P, R, and F denote precision, recall, and F-score, respectively.

Initial Calibration with Golden Set The annotators first labeled the initial subset of 20 samples from the golden set. These 20 samples were used to verify the annotators' labeling correctness. Since these samples had gold labels, we assessed the macro F-score of each annotator's results. Based on these results, we then provided further clarification and discussion on unclear aspects. Annotators are allowed to revisit and modify their previously labeled data if necessary. At this stage, they could also provide feedback and suggest adjustments to the error type definitions.

Iterative Calibration After confirming no immediate issues with the annotators' task, they continued labeling the next 20 samples from the golden set. This process was repeated using our pre-annotated gold labels to evaluate the annotators' performance for the 20 samples and identify discrepancies. This iterative process continued until all data in the golden set was annotated.

Table 1 presents the annotators' labeling correctness, evaluated using macro F-scores against the gold labels. The five stages correspond to macro F1 scores for subsets labeled at different stages, with increasing scores reflecting improved consistency and accuracy from iterative discussions and revisions. We also used the gold-standard set to assess agreement between each annotator and the gold labels, yielding Cohen's κ of 0.5688, 0.6970, and 0.5556 for the three annotators, respectively. Overall inter-annotator agreement across the three annotators, measured by Fleiss' κ, was 0.6038, indicating a moderate level of agreement.

Final Calibration and Annotation To ensure that annotators fully understood the task and applied consistent standards, we had them annotate the remaining 100 samples from the initial set, of which

Error type	Question	Reason
Interpretation Ambiguity	学校购买每张单价是140元的课桌，买了30张还多480元．如果用这笔钱买椅子，可以买40把．每把椅子的单价=多少元？ (The school buys desks at a unit price of $140 each. After purchasing 30 desks, it has $480 left. If this money is used to buy chairs, 40 chairs can be bought. What is the unit price of each chair?)	The question does not clearly specify whether "this money" refers to the total budget or the remaining funds after purchasing the desks.
Informal Wording	某书店购书一律0.95，小红买了一本书，比原价便宜6元，这本书原价多少元？(A certain bookstore charges 0.95 for books purchased. A girl bought a book and saved 6 yuan off the original price. What was the original price of the book?)	The problem statement might lead one to believe that all books in the bookstore are priced at 0.95 yuan. The intended meaning, however, is that a 5% discount is applied to the original price.
Unitless	一根木头，用32秒的时间分成了5段，以同样的速度将另一根木头分成7段，需要多少时间？(A piece of wood is cut into 5 sections in 32 seconds. At the same rate, how much time is needed to cut another piece of wood into 7 sections?)	The unit for the required time is not specified. For example, the question should ask, "How many seconds?"
Unclear Relationship	4平家电商场现有各种品牌的电视机240台，比电冰箱多(1/5)，商场现有电冰箱多少台？(An appliance store currently has 240 televisions of various brands, which is more than refrigerators by 1/5. How many refrigerators does the store have?)	The question does not clarify the actual meaning of "more by 1/5". It should describe that the number of televisions is 1/5 times more than the number of refrigerators.
Calculation Error	6000/59与35的差，商？(6000/59 and 35 difference, quotient?)	The question belongs to this type is either poorly worded or unclear in defining the operation. For this example, a more precise wording would be "What is the result of 6000 divided by 59, then subtracting 35?"
None	为灾区捐款，小华捐4.2元，比小丽多捐了0.4元，小华比小丽多捐几分之几？(Xiaohua donated 4.2 yuan, which is 0.4 yuan more than what Xiaoli donated. By what fraction did Xiaohua donate more than Xiaoli?)	The question is clear and correct.

Table 2: Error types in math word problems

21 were problematic math questions. The average inter-annotator agreement for this round reached a Fleiss' κ of 0.8103, representing substantial agreement. After ensuring a sufficient level of consistency in the annotation standards, we assigned the annotators to label their respective portions of the data. Each annotation was reviewed by the authors, and any discrepancies in the labels were discussed with the annotators to reach a consensus.

3.3 Dataset Analysis

Table 2 presents the examples for each error type. This process resulted in 23,162 math word problems, with an error-type distribution of *Interpretation Ambiguity*, *Informal Wording*, *Unitless*, *Unclear Relationship*, *Calculation Error*, and *None* errors of 136, 1,076, 416, 606, 67, and 20,916, respectively. Although "Multiple Interpretation," "Informal Wording," and "Unclear Relationship" all stem from imprecise or incomplete problem descriptions, their effects differ. "Multiple Interpretation" may lead to various interpretations, resulting in multiple possible solutions. On the other hand, "Informal Wording" and "Unclear Relationship" involve cases where the teacher's intended question can still be inferred, but the phrasing is informal and unsuitable for offical examinations. "Unclear Relationship" specifically refers to situations where numerical relationships are not clearly defined, affecting the clarity of the problem. As for "Calculation Error," it typically arises from mixed mathematical and verbal expressions. We aim to detect these five types of errors to assist teachers in refining problem descriptions, ensuring they are more comprehensive and formal, which is essential for official examinations. Note that a single math world problem may contain multiple errors. Therefore, detecting error types in math word problems is a multilabel classification task.

4 Prompt Refinement through Self-Optimization

As mentioned, prompt refinement methods have been explored. The selection of demonstrations in few-shot prompts has also been discussed (Luo et al., 2023). Figure 2 shows an overview of the proposed framework.[7]

PRO comprises a classifier $\mathcal{M}_c$, a definition reflector $\mathcal{M}_s$, an evaluator f_e, and a judge f_j. The

[7]The algorithm is provided in Appendix C.

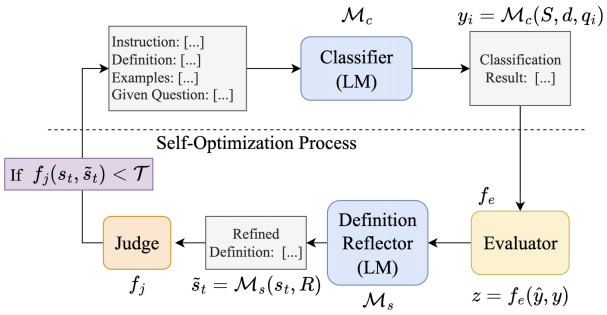

Figure 2: Overview of PRO framework

base models of $\mathcal{M}_c$ and $\mathcal{M}_s$ are the same LLM. All the prompt templates used in this work are presented in Appendix D. For the self-optimization process, we partition our training set into sets D and $D^{'}$ for demonstration and reflection, respectively. Instead of using human-written definitions, we employ an LLM $\mathcal{M}$ to generate the initial definition s_t of each error type $t \in T$. Specifically, given a set of questions $Q_t \in D$ that contains an error of t and its corresponding corrections $Q_t^{'}$, we obtain $s_t = \mathcal{M}(Q_t, Q_t^{'})$. To assess whether the definition requires refinement, we consult the classification results. Given the definitions S of T, the selected examples d in D, and the i-th question q_i in $D^{'}$, we obtain the possible error types in q_i: $y_i = \mathcal{M}_c(S, d, q_i)$.

After obtaining the initial classification results, we proceed to the self-optimization phase. A maximum number of trials is set to iteratively refine S through $\mathcal{M}_s$. f_e measures the classification results using the macro-averaged F-score to judge the model's performance in each round. For each t, incorrectly predicted questions are extracted for self-reflection. If the predicted type y_i^t does not match the ground truth $\hat{y}_i^t$, the question q_i^t is appended to reflection set R. Subsequently, the refined definition $\tilde{s}_t$ of t is obtained as $\tilde{s}_t = \mathcal{M}_s(s_t, R)$. $\tilde{s}_t$ will be used in the next round as s_t.

Finally, to determine whether the update to s_t has converged, f_j compares the difference in the reflection results between the previous and current rounds. We measure the difference between s_t and $\tilde{s}_t$ using ROUGE-1 (Lin, 2004). If the ROUGE-1 score $r = f_j(s_t, \tilde{s}_t)$ is greater than the threshold $\mathcal{T}$, we view it as convergence. In this case, s_t will not be refined in the next round. If S converges, self-optimization ceases. We measure the performance of using S in each round via $z = f_e(\hat{y}, y)$: S from the highest-scoring round is used to instruct $\mathcal{M}_c$ to detect error types in the test set. The number of max trials and $\mathcal{T}$ are set to 10 and 0.9, respectively.

5 Experiments

5.1 Experimental Setup

Since most questions in Math23k were of the *None* type, we randomly selected a subset of math word problems from this type. This resulted in a total of 4,766 math word problems in MathError. To simulate limited data availability, we identified patterns from sparse data to detect error types in math word problems. Specifically, the data used for demonstration and reflection consisted of 15 and 30 math word problems, respectively. The remaining 4,721 math word problems were used for testing. The demonstration set consisted of 3 questions for each error type, providing a foundation for inference. The reflection set, designed to refine and validate the model, included 2 questions for each error type and an additional 20 questions without errors. In the test set, the number of questions for *INTPN*, *Informal*, *Unit*, *Rel*, *Calc*, and *None* were 131, 1,071, 411, 601, 62, and 2,500, respectively.[8] We utilized OpenAI's API, in particular the `gpt-3.5-turbo-0125` and `gpt-4o-2024-05-13` models. The temperature was set to 0 to increase reproducibility.

5.2 Experimental Results

Table 3 shows the performance of PRO.[9] We use the macro-averaged F-score as the evaluation metric and report the F-scores for each error type. The top two rows show the baseline models using LLM for direct inference on the test set with few-shot prompting and no refinement. The baseline models are GPT-3.5 and GPT-4o. In Table 3, all the results are based on model-generated definitions and human-written examples, as this is the optimal setting. As different reasoning strategies can yield different results, we conducted an experiment to find the most suitable strategy for detecting error types. For PRO (GPT-4o), we further include results from four strategies for comparison.

Directly Classify (Direct): $\mathcal{M}_c$ is prompted to predict the error type with any reasoning steps.

Analyze → Classify (A → C): $\mathcal{M}_c$ is tasked to generate an analysis of the errors present in the question and then classify the error types.

[8]The sum of questions exceeds the total because some contain multiple error types.

[9]Supplementary experiments are reported in Appendix E.

Method	Strategy	Overall	Interpretation Ambiguity	Informal Wording	Unitless	Unclear Relationship	Calculation Error	None
GPT-3.5	Direct	0.2295	0.0938	0.3422	0.1850	0.2681	0.2255	0.2625
GPT-4o	Direct	0.3118	**0.1256**	**0.4362**	0.3081	0.3930	0.2269	0.3812
PRO (GPT-3.5)	Direct	0.2381	0.0242	0.1150	**0.4033**	0.2339	0.2208	0.4315
PRO (GPT-4o)	Direct	**0.3243**	0.0936	0.4271	**0.3109**	**0.4212**	**0.2538**	**0.4390**
	CoT	0.2367	0.0457	0.3368	0.2249	0.2097	0.0492	0.5534
	A → C	0.2563	0.0864	0.4106	0.2270	0.2878	0.0688	0.4574
	S → C	0.2420	**0.0980**	0.3942	0.1769	0.2934	0.1188	0.3708

Table 3: Results of error type detection with different inference strategies.

Solve → Classify (S → C): $\mathcal{M}_c$ solves the math word problem before classifying the error type. This strategy enables $\mathcal{M}_c$ to identify informal or missing information during problem-solving.

We also tested the above inference strategies for GPT-3.5, GPT-4, and PRO (GPT-3.5), the "Direct" strategy consistently performed best; hence, only "Direct" results are reported for these methods. Experimental results show that PRO based on GPT-4o with the "Direct" strategy outperforms other methods, and significantly outperforms GPT-3.5 ($p < 0.05$). Although it does not significantly outperform the remaining methods in Table 3, overall, PRO (GPT-4o) detects the most problematic questions. This may be because in error type detection, iteratively reflecting on the wrong prediction and coming up with a solution to refine the definitions that can strengthen the reasoning process requires an LLM with more powerful natural language understanding and generation abilities. Comparing the results between GPT-4o and "PRO (GPT-4o)", the proposed reflection mechanism enhances the ability to detect *Informal*, *Rel*, *Calc*, and *None* types. Moreover, both GPT-3.5 and GPT-4o struggle to detect *INTPN*. Detecting "Interpretation Ambiguity" problems remains a challenging task.

For PRO (GPT-4o), the results across different strategies reveal notable differences. In general, direct classification is the most effective for detecting problematic questions, whereas solving the problem and then classifying its error type is best for detecting *INTPN* errors. We also conducted a comparison using the CoT prompting method, which did not yield the best performance. Based on these results, we speculate that if an LLM is guided to reason, the stronger the LLM's inferencing ability, the more it may overlook issues in textual descriptions, leading to ineffective error detection.

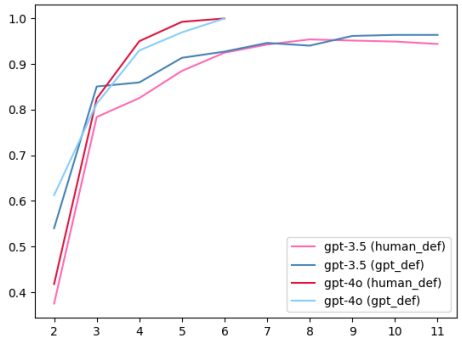

Figure 3: Refinement trajectories across rounds

6 Discussion

6.1 Model Sensitivity to Prompt Design

We speculate that model-generated prompts better align with the model's probability distribution, enhancing its semantic understanding and ability to detect error types. To this end, we evaluate LLaMA3 8B (AI@Meta, 2024) with human-written and model-generated definitions. Since probability distributions are difficult to obtain through the API, we conduct experiments with a locally deployed model. The overall macro F-score of LLaMA3 8B are 0.1191 with human-written definitions and 0.1479 with model-generated definitions. The higher score indicates that model-generated definitions yield better performance. The detailed results are shown in Figure 7 in Appendix F. Furthermore, we assess the model's perplexity on both human-written and model-generated definitions with examples, resulting in values of 0.7041 and 0.6499, respectively. This suggest that model-generated prompts are more predictable, implying the model handles this type of narrative better.

6.2 Effects on Refinement and Convergence

We further analyze the impact of different initial definition approaches and LLMs on refinement trajectories across rounds. Figure 3 presents the results of PRO based on GPT-3.5 and GPT-4o, using

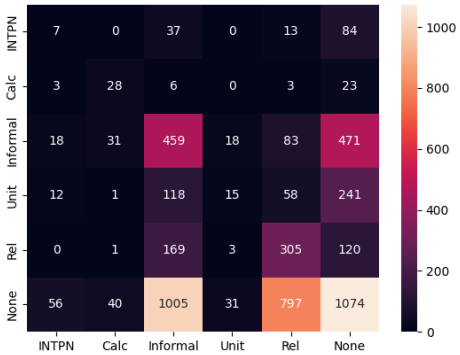

Figure 4: Confusion matrix for error type detection using "PRO (GPT-4o)"

either human-written or model-generated definitions, with all examples human-written. Specifically, we measured the ROUGE-1 score of definitions and examples before and after rounds of self-optimization. A higher score indicates greater similarity between definitions and examples across rounds. With this analysis we investigate the number of rounds required for convergence under different settings. The x-axis represents the round number, and the y-axis denotes the ROUGE score between the current and the previous round's definitions and examples. GPT-3.5 significantly refines descriptions, leading to considerable variation and making convergence difficult, often reaching the maximum of 10 rounds. GPT-4o, in turn, makes slight refinements, typically adding a few details to definitions and examples, and generally converging in around 5 rounds due to smaller refinements and fewer classification errors.

Additionally, Figure 3 shows a significant increase in ROUGE scores during the first three rounds, indicating substantial modifications to definitions and examples based on classification results. This trend levels off in later rounds, suggesting the model has exhausted useful information from $D^{'}$ for further refinement. Moreover, with human-written definitions, the ROUGE score increases more sharply in the first three rounds compared to model-generated definitions, indicating a greater initial disparity.

6.3 Error Analysis

Figure 4 presents the confusion matrix for error type detection results using the PRO (GPT-4o) method with model-generated definitions and human-written examples.[10] *Interpretation Ambiguity* and *Unitless* errors are often predicted as *None* or *Information Wording*. Additionally, questions exhibiting the *Unclear Relationship* error are frequently predicted as *Informal Wording*.

Interestingly, math word problems with no errors (belongs to *None*) are often predicted as *Informal Wording* and *Unclear Relationship*. We examined the results predicted by other methods, which similarly tended to predict *None* type questions as containing *Informal Wording* and *Unclear Relationship* errors. This may be due to using *Informal Wording* and *Unclear Relationship* errors as examples to demonstrate the response format. To examine this effect, we conducted an additional experiment (Appendix G) in which the response format demonstration was replaced with "Calculation Error." The results show that the model's bias toward predicting "Informal Wording" and Unclear Relationship decreased, confirming that the model is highly sensitive to the examples used in the demonstration. However, the underlying reason for this prediction behavior remains unclear and is left for future investigation.

7 Conclusion

This paper aims to detect errors in math word problem statements to assist teachers in designing clear and well-structured questions. These errors include ambiguous statements that deviate from the intended meaning, informal wording that affects readability, missing unit specifications, unclear numerical relationships, and imprecise expressions that lead to miscalculations. We construct MathError, the first human-annotated dataset, to explore the method of error type detection. We explore prompt refinement through self-optimization (PRO) to instruct an LLM to adapt to the given task. We investigate whether definitions for error types and corresponding few-shot examples are more effectively provided by humans or models. The results show that machine-generated definitions of error types, enhance effectiveness in error type detection. During the self-optimization process, we modify the definitions based on the classification errors of the data retained from the training set. However, accurately identifying the errors in a math word problem is still challenging; more advanced methods are left as future work. We also plan to explore methods for correcting problematic questions.

[10]The total numbers in the confusion matrix differ from the actual number of error types because our task is multilabel classification. An error type can be incorrectly identified as multiple different error types.

Limitations

In this work, we extend the Math23K dataset with error type annotation. While this is the first human-annotated dataset used for detecting errors in math word problems, our research was limited to Chinese math word problems and did not explore other languages. Additionally, our work currently relies on a single dataset comprising questions collected from an online educational platform, predominantly at the elementary school level, which may not offer sufficient diversity in problem types. Increased complexity and difficulty may necessitate different methods.

Moreover, we define only five error types within this dataset: these may not encompass all possible errors in math word problems. Note also that in the proposed self-optimization framework, we examine only GPT-3.5, GPT-4, and LLaMA 3, leaving several LLMs unexplored.

References

AI@Meta. 2024. Llama 3 model card.

Xinyun Chen, Maxwell Lin, Nathanael Schaerli, and Denny Zhou. 2023. Teaching large language models to self-debug. In *The 61st Annual Meeting of The Association for Computational Linguistics*.

Bryan Christ, Jonathan Kropko, and Thomas Hartvigsen. 2024. Mathwell: Generating educational math word problems using teacher annotations. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11914–11938.

Nico Daheim, Jakub Macina, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2024. Stepwise verification and remediation of student reasoning errors with large language model tutors. *arXiv preprint arXiv:2407.09136*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 3563–3578.

Wanyong Feng, Jaewook Lee, Hunter McNichols, Alexander Scarlatos, Digory Smith, Simon Woodhead, Nancy Ornelas, and Andrew Lan. 2024. Exploring automated distractor generation for math multiple-choice questions via large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3067–3082.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2024. Promptbreeder: self-referential self-improvement via prompt evolution. In *Proceedings of the 41st International Conference on Machine Learning*, pages 13481–13544.

Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas I Liao, Kamilė Lukošiūtė, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, and 1 others. 2023. The capacity for moral self-correction in large language models. *arXiv preprint arXiv:2302.07459*.

Hannah Gonzalez, Liam Dugan, Eleni Miltsakaki, Zhiqi Cui, Jiaxuan Ren, Bryan Li, Shriyash Upadhyay, Etan Ginsberg, and Chris Callison-Burch. 2023. Enhancing human summaries for question-answer generation in education. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 108–118.

Wanyong Feng Hunter McNichols, Jaewook Lee, Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2024. Automated distractor and feedback generation for math multiple-choice questions via in-context learning. *arXiv preprint arXiv:2308.03234*.

Vid Kocijan, Oana-Maria Camburu, Ana-Maria Cretu, Yordan Yordanov, Phil Blunsom, and Thomas Lukasiewicz. 2019. WikiCREM: A large unsupervised corpus for coreference resolution. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4303–4312. Association for Computational Linguistics.

Alexander Koller, Michaela Regneri, and Stefan Thater. 2008. Regular tree grammars as a formalism for scope underspecification. In *Proceedings of ACL-08: HLT*, pages 218–226.

Dmitrii Krasheninnikov, Egor Krasheninnikov, and David Krueger. 2022. Assistance with large language models. In *NeurIPS ML Safety Workshop*.

Jaewook Lee, Digory Smith, Simon Woodhead, and Andrew Lan. 2024. Math multiple choice question generation via human-large language model collaboration. *arXiv preprint arXiv:2405.00864*.

Margaret Y Li, Alisa Liu, Zhaofeng Wu, and Noah A Smith. 2024. A taxonomy of ambiguity types for NLP. *arXiv preprint arXiv:2403.14072*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah Smith, and Yejin Choi. 2023. We're afraid language models aren't modeling ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 790–807, Singapore. Association for Computational Linguistics.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023. Dr. icl: Demonstration-retrieved in-context learning. *arXiv preprint arXiv:2305.14128*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7957–7968.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Tian Song, Qinqin Tian, Yijia Xiao, and Shuting Liu. 2023. Automatic generation of multiple-choice questions for CS0 and CS1 curricula using large language models. In *International Conference on Computer Science and Education*, pages 314–324. Springer.

YuHong Sun, Zhangyue Yin, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Hui Zhao. 2024. Benchmarking hallucination in large language models based on unanswerable math word problem. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2178–2188.

Takaaki Tanaka, Francis Bond, Timothy Baldwin, Sanae Fujita, and Chikara Hashimoto. 2007. Word sense disambiguation incorporating lexical and structural semantic information. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 477–485.

Rose E. Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024a. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric Xing, and Zhiting Hu. 2024b. Promptagent: Strategic planning with language models enables expert-level prompt optimization. In *The Twelfth International Conference on Learning Representations*.

Yan Wang, Xiaojiang Liu, and Shuming Shi. 2017. Deep neural solver for math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 845–854.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32.

An-Zi Yen and Wei-Ling Hsu. 2023. Three questions concerning the use of large language models to facilitate mathematics learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3055–3069.

An-Zi Yen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021. Unanswerable question correction in question answering over personal knowledge base. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14266–14275.

Hamed Zamani, Gord Lueck, Everest Chen, Rodolfo Quispe, Flint Luu, and Nick Craswell. 2020. MIMICS: A large-scale data collection for search clarification. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 3189–3196.

Zhuosheng Zhang, Junjie Yang, and Hai Zhao. 2021. Retrospective reader for machine reading comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14506–14514.

Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V Le, Ed Chi, Denny Zhou, Swaroop Mishra, and Huaixiu Steven Zheng. 2024. Self-discover: Large language models self-compose reasoning structures. *Advances in Neural Information Processing Systems*, 37:126032–126058.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

Objective: To create a dataset of Chinese math word problems with error type annotations, where some problems contain errors that need to be annotated accordingly. The error types include misleading, unclear problem statement, etc.
Reminders: - We will regularly track your annotated results, so any questions can be addressed immediately. - If you find an error with a problem after reviewing it, select the appropriate category from the list below and label it. - Label the chosen error type with a lowercase "x" in the box following the problem. - Multiple annotations can be applied to a single problem.
Error Type Definition: *Described in Section 3.1*

Table 4: Annotation guidelines

A Example Problem and Its Rationale

Question: 学校购买每张单价是140元的课桌，买了30张还多480元．如果用这笔钱买椅子，可以买40把．每把椅子的单价=多少元？ (The school buys desks at a unit price of $140 each. After purchasing 30 desks, it has $480 left. If this money is used to buy chairs, 40 chairs can be bought. What is the unit price of each chair?)

Rationale: 设椅子的单价为x元 (Let the unit price of each chair be x.)

根据题意，课桌的总价为30 × 140 = 4200元。 (According to the problem statement, the total cost of the desks is: 30 × 140 = 4200)

再加上剩下的480元，那么可以用来买椅子的总价为4200 + 480 = 4680元。 (Adding the remaining amount of 480, the total amount available for purchasing chairs is: 4200 + 480 = 4680)

根据题意，可以买40 把椅子，所以$40x = 4680$。 (Since 40 chairs can be bought with this amount, we set up the equation: $40x = 4680$

解方程得到：$x = 4680/40 = 117$ 元。 (Solving for x: $x = \frac{4680}{40} = 117$

所以每把椅子的单价为117元。 (Thus, the unit price of each chair is 117.)

B Annotation Guideline

Table 4 presents the annotation guidelines provided to the annotators.

C Algorithm of PRO

Algorithm 1 presents our proposed method, outlining its key steps and design choices.

Algorithm 1: Refinement via self-optimization

```
Initialize M_c, M_s, f_e, f_j
Generate initial definition s_t ∈ S for each error type
  t ∈ T using D
for number of max trials do
    for q_i ∈ D' do
        y_i = M_c(S, d, q_i);
    end
    z = f_e(ŷ, y);
    for t ∈ T do
        Set reflection set R = {};
        for q_i^t ∈ D' do
            if y_i^t ≠ ŷ_i^t then
                R ← q_i^t;
            end
        end
        s̃_t = M_s(s_t, R);
        if f_j(s_t, s̃_t) > T then
            Stop refining s_t in next round;
        end
        s_t = s̃_t;
    end
end
```

D Input Formats

Tables 5, 6, and 7 contain templates of the "Directly classify", "Analyze → Classify", and "Solve → Classify" strategies, respectively. The templates for definition generation is shown in Table 8. The proposed definition refinement template is shown in Table 9. In the refinement prompt, we instruct the LLM to analyze why the error was not detected. However, we find that not including the result of the analysis is better. This may be because the LLM still struggles to complete the error type detection tasks. The content in the generated analysis may contain incorrect information.

E Additional Analyses

E.1 Multilabel Classification vs. Binary Classification

We also conducted binary classification for error type detection by querying whether a specific math word problem contains a particular error. This raises the question: Which method for error type detection is more suitable: multilabel classification or binary classification? Figure 5 shows the results when treating error type detection as multilabel or binary classification. We used the GPT-3.5 model with zero-shot prompting. We find that binary classification causes the model to tend to view every math word problem as erroneous. A possible reason is that the prompt for multi-label classification

Task Introduction:
Please analyze and identify whether the `[given problem]` contains any of the following errors based on the descriptions under `[error types]`: interpretation ambiguity, informal wording, unitless, unclear relationship, or calculation error. If any errors are identified, respond with the corresponding option(s). If the `[Given Problem]` does not contain any of these errors, select (F). The problem must match the definitions of the error types to be considered erroneous; note that a single problem may contain multiple errors.
`[Error Types]`
(A) Interpretation Ambiguity: {*Definition*} {*Examples*}.
(B) Informal Wording: {*Definition*} {*Examples*}.
(C) Unitless: {*Definition*} {*Examples*}.
(D) Unclear Relationship: {*Definition*} {*Examples*}.
(E) Calculation Error: {*Definition*} {*Examples*}.
(F) None of the above. Example: {*Example*}

Response Requirement:
Enter your predicted results under `[classification result]`. If multiple errors are present, separate them with a semicolon ";".
`Example response:`
`[Given Problem]` There are 120 chicken eggs, and the duck eggs are 1/6 more than the chicken eggs. How many eggs are there in total?
`[Classification Result]` (C) Informal Wording; (E) Unclear Relationship
`[Given Problem]` {*question*}

Table 5: Prompt template for event type detection (directly classify method).

Task Introduction:
{Same as that shown in Table 5}

Response Requirement:
Please provide an analysis of up to 150 words after `[Analysis]` and record your assessment results under `[Classification Result]`. If multiple errors are present, separate them with a semicolon ";".
`Example response:`
`[Given Problem]` There are 120 chicken eggs, and the duck eggs are 1/6 more than the chicken eggs. How many eggs are there in total?
`[Analysis]` The statement is unclear, rendering the meaning of "1/6" ambiguous. This could imply that the number of duck eggs is 1/6 that of the chicken eggs (which is logically inconsistent as it contradicts the statement that there are more duck eggs), or that the duck eggs exceed the number of chicken eggs by 1/6 of the chicken egg count.
`[Classification Result]` (C) Informal Wording; (E) Unclear Relationship
`[Given Problem]` {*question*}

Table 6: Prompt template for event type detection (Analyze→Classify method).

provides definitions of all types, which helps the model better understand the error types compared to binary classification.

E.2 Performance of LLMs with Different Parameter Sizes

We compared the performance of different sizes of LLaMA 3 (AI@Meta, 2024) with few-shot prompting using model-generated definitions and examples. The results are shown in Figure 6. LLaMA3 70B tends to classify questions as informal, often failing to recognize error-free cases. While larger models generally perform better, the task remains difficult for smaller LLMs, making strong semantic understanding crucial for detecting errors in math word problems.

Task Introduction:
{Same as that shown in Table 5}

Response Requirement:
Please solve the problem and generate your calculations and reasoning within 150 words under `[Calculation Process and Rationale]`. If the solution remains elusive, it is possible that the problem includes the above errors. Based on the difficulties encountered during the calculation, place your predicted results behind `[Classification Result]`. If multiple errors are identified, separate them with a semicolon ";".
`Example response:`
`[Given Problem]` There are 120 chicken eggs, and the duck eggs are 1/6 more than the chicken eggs. How many eggs are there in total?
`[Calculation Process and Rationale]` The statement "duck eggs are 1/6 more" is ambiguous: does it refer to 1/6 more than the number of duck eggs, or 1/6 more than the number of chicken eggs? The lack of a clear referent for the ratio leads to confusion. If it means 1/6 more than the number of duck eggs, the question lacks sufficient information to calculate the number of duck eggs. Conversely, if it refers to 1/6 of the chicken eggs, then the number of duck eggs would be 120 + 120/6 = 140.
`[Classification Result]` (C) Informal Wording; (E) Unclear Relationship
`[Given Problem]` {*question*}

Table 7: Prompt template for event type detection (Solve → Classify method).

Task Introduction:
You are tasked with writing a definition for the error type based on problematic questions and their corrected versions. Please use the following `[Problematic Question]` and corresponding `[Corrected Question]` to formulate your definitions. Please place the generated definition behind `[Definition]`.
`[Given Error Type]` {*error type*}
`[Problematic Question 1]` {*problematic question* 1}
`[Corrected Question 1]` {*corrected question* 1}
`[Problematic Question 2]` {*problematic question* 2}
`[Corrected Question 2]` {*corrected question* 2}
...
`[Definition]`

Table 8: Prompt template for generating definitions by model.

F Effect of Human-Written vs. Model-Generated Definitions on LLaMA 8B Performance

Figure 7 reports the detailed comparison between using human-written and model-generated prompts under LLaMA 3 8B.

G Error Analysis

To investigate the impact of the response format demonstration, we replaced the response format demonstration with *Calculation Error*. Figure 8 shows the confusion matrix of error type detection results after changing the format demonstration. The model's preference for predicting *Informal Wording* and *Unclear Relationship* errors became less pronounced, but there was no tendency to predict the *None* type as *Calculation Error*. From this result, we find that the model is highly sensitive to the examples used. However, it remains unclear why the prediction trend does not align with our original hypothesis that the model will predict the error types used in the format demonstration after

Task Introduction:
You are tasked with refining the given error type definition based on the results of erroneous classifications. The classification task involves analyzing the following `[Error Types]` to determine whether `[Given Problem]` contain errors such as interpretation ambiguity, informal wording, unitless, unclear relationship, or calculation error. If any errors are present, select the corresponding option. If the `[Given Problem]` contains no such errors, choose (F). The problem must match the definitions of the error types to be considered erroneous, and note that a single problem may contain multiple errors.
`[Error Types]`
(A) Interpretation Ambiguity: {*Definition*} {*Examples*}.
(B) Informal Wording: {*Definition*} {*Examples*}.
(C) Unitless: {*Definition*} {*Examples*}.
(D) Unclear Relationship: {*Definition*} {*Examples*}.
(E) Calculation Error: {*Definition*} {*Examples*}.
(F) None of the above. Example: {*Example*}

{*wrong predictions*}
Please refer to both the correctly classified results and the misclassified ones under "{target type}" error, along with the definition of this error type. Analyze why there was a wrong prediction or why the error was not recognized. Use this analysis to refine the definition of the `[Error Type]` "{target type}".
Please output the refined definition in the following format: `[Error Type Definition]`: [Revised Definition] [Analysis]

Table 9: Prompt template for definition refinement.

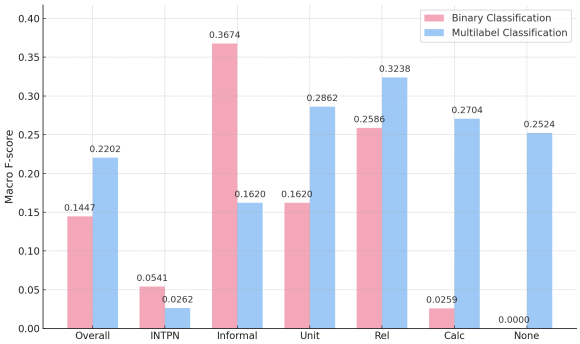

Figure 5: Results of different classification tasks using GPT-3.5 with zero-shot prompting

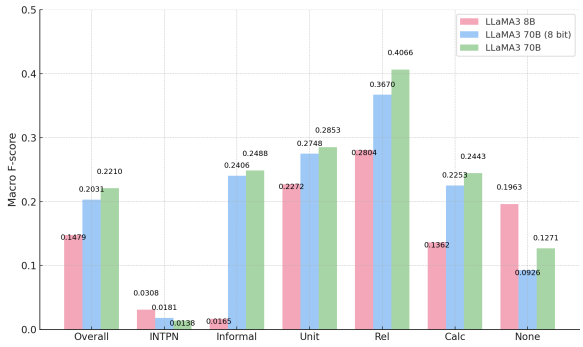

Figure 6: Results of different LLMs with few-shot prompting using model-generated definitions and examples

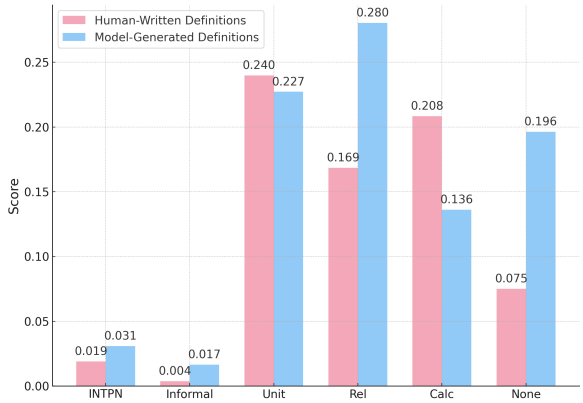

Figure 7: Comparison of LLaMA 8B performance across error categories using human-written and model-generated definitions

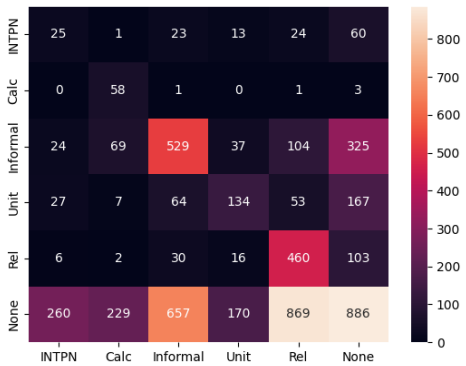

Figure 8: Confusion matrix for error type detection with format example changed to *Calc*

changing the example.

H Examples of Prompt Refinement Results

Tables 10, 11, 12, 13, and 14 show the examples of prompt refinement results for *INTPN*, *Informal*, *Unit*, *Rel*, and *Calc*, respectively. The following prompt refinement results in our study are presented in Chinese. To facilitate reader understanding, we used GPT-4o to translate them into English.

I GenAI Usage Disclosure

We utilized generative AI tools solely for grammar and language refinement. All content was reviewed and edited by the author(s), who take full responsibility for the final manuscript.

Error Type	Initial Definition and Example Generated by LLM	Refined By LLM in the 2nd Round
Interpretation Ambiguity	多解是指在数学题目中，由于题目描述不够明确或存在歧义，导致读者可以从不同的角度理解题意，从而得出多个不同的解答。这种情况下，题目本身并没有唯一的解答，或者解答的条件不够充分，无法确定唯一的正确答案。范例A. 学校购买每张单价是140元的课桌，买了30张还多480元．如果用这笔钱买椅子，可以买40把．每把椅子的单价=多少元? B. 一块平行四边形的街头广告牌，底是12.5米，高是6.4米．如果要油饰这块广告牌，每平方米用油漆0.6千克，需要多少千克油漆? 其中A 因为题目未提到这笔钱刚好只能买这些椅子，所以答案有无限多种可能。B 未提到油漆要涂双面还是单面，属于此类。(Interpretation Ambiguity in problem-solving occurs when the description of a mathematical problem is unclear or contains ambiguities, allowing readers to interpret the problem from different perspectives and arrive at multiple distinct solutions. In such cases, the problem itself does not have a unique solution, or the conditions provided are insufficient to determine a single correct answer.For example:A. A school purchases desks at a unit price of 140 units, buying 30 desks with an additional 480 units remaining. If this remaining amount is used to purchase chairs, they can buy 40 chairs. What is the unit price of each chair? B. A street advertisement board in the shape of a parallelogram has a base of 12.5 meters and a height of 6.4 meters. If painting this board requires 0.6 kilograms of paint per square meter, how many kilograms of paint are needed? In the case A, since the problem does not specify that the remaining money can only be used to buy these chairs, there are infinitely many possible answers.The problem of case B falls into the error type of Interpretation Ambiguity since it does not specify whether the paint is needed for both sides or just one side.)	多解是指在数学题目中，由于题目描述不够明确或存在歧义，导致读者可以从不同的角度理解题意，从而得出多个不同的解答。这种情况下，题目本身并没有唯一的解答，或者解答的条件不够充分，无法确定唯一的正确答案。具体来说，这可能包括但不限于以下情况：题目中缺少关键信息或条件，导致无法确定唯一解；题目中存在多种可能的解释或理解方式，导致多个解答；题目中使用了模糊或不明确的语言，导致解答者对题意产生不同的理解。范例A. 学校购买每张单价是140元的课桌，买了30张还多480元．如果用这笔钱买椅子，可以买40把．每把椅子的单价=多少元? B. 一块平行四边形的街头广告牌，底是12.5米，高是6.4米．如果要油饰这块广告牌，每平方米用油漆0.6千克，需要多少千克油漆? C. 小芳一家准备出去旅游，计划每天的支出如下：住宿费65元，交通费25元，伙食费60元，如果外出旅游一周（按7天算），至少应准备多少钱? 其中A 因为题目未提到这笔钱刚好只能买这些椅子，所以答案有无限多种可能。B 未提到油漆要涂双面还是单面，属于此类。C 属于此类。(Interpretation Ambiguity refer to situations in mathematical problems where the problem description is not clear enough or contains ambiguities, leading readers to interpret the problem from different perspectives and arrive at multiple different solutions. In such cases, the problem itself does not have a unique solution, or the conditions provided are insufficient to determine a single correct answer. Specifically, this may include, but is not limited to, the following situations: the problem lacks key information or conditions, making it impossible to determine a unique solution; the problem allows for Interpretation Ambiguity or understandings, leading to Interpretation Ambiguity; the problem uses vague or unclear language, resulting in different understandings by the solvers.Examples:A. A school purchases desks at a unit price of 140 units, buying 30 desks with an additional 480 units remaining. If this remaining amount is used to purchase chairs, they can buy 40 chairs. What is the unit price of each chair?B. A street advertisement board in the shape of a parallelogram has a base of 12.5 meters and a height of 6.4 meters. If painting this board requires 0.6 kilograms of paint per square meter, how many kilograms of paint are needed?C. Xiaofang's family plans to go on a trip, with the following daily expenses: accommodation 65 units, transportation 25 units, and food 60 units. If they plan to travel for a week (7 days), how much money should they prepare at least?In example A, the problem does not specify that the remaining money can only be used to buy these chairs, so there are infinitely many possible answers. In example B, it is unclear whether the paint is needed for both sides or just one side, making it fall into this error type. Example C also falls into this error type.)

Table 10: Example of prompt refinement for the multiple-interpretation type

Error Type	Initial Definition and Example Generated by LLM	Definition Refined By LLM in the 2nd Round
Informal Wording	当数学题目中缺少关键的叙述或符号，导致题意不明确或无法正确理解时，这种情况被称为「不明或缺失关键叙述」。这种错误会使读者无法准确地解读题目，从而影响解题过程和结果。 范例A. 一本书原__是18元，降价一半后售价是多少? B. 一个数字的1/3是另一个数字的1/4，这两个数字的比是((())/(())) 其中，A缺少部分叙述，让题目略显怪异，属于此类，正确的写法：一本书原价是18元。而B的括号为答案区，并非缺少资讯，因此不在此列。 (When a mathematical problem lacks essential descriptions or symbols, resulting in unclear or incorrect interpretation. Such errors prevent readers from accurately understanding the problem, thereby affecting the problem-solving process and outcomes. For example: A. The original price of a book was __18 units. What is the price after a 50% discount? B. One number is 1/3 of another number's 1/4. The ratio of these two numbers is ((())/(())). In example A, the lack of critical descriptions makes the problem somewhat peculiar, falling into this error type. The correct statement should be: "The original price of a book is 18 units." In example B, the parentheses indicate the answer area and do not represent missing information; therefore, it is not considered under this error type.)	当数学题目中缺少关键的叙述或符号，导致题意不明确或无法正确理解时，这种情况被称为「不明或缺失关键叙述」。这种错误会使读者无法准确地解读题目，从而影响解题过程和结果。常见的情况包括：题目中缺少必要的数据或条件，导致无法进行计算；题目叙述不完整或存在歧义，使得解题者无法确定应该如何解答。 范例A. 一本书原__是18元，降价一半后售价是多少? B. 一个数字的1/3是另一个数字的1/4，这两个数字的比是((())/(())) C. 四年级共有13个合作小组，每个小组有4名同学．已知全班同学本学期共得到312颗合作星，平均每个同学得到几颗合作星？D. 6（2）班有科技书240本，故事比科技书少(1/6)．故事书比科技书少多少书？其中，A缺少部分叙述，让题目略显怪异，属于此类，正确的写法：一本书原价是18元。而B的括号为答案区，并非缺少资讯，因此不在此列。C属于此类，D属于此类。 (When a mathematical problem lacks critical descriptions or symbols, making the problem statement unclear or difficult to interpret correctly, this situation is referred to as "Informal Wording" Such errors prevent readers from accurately interpreting the problem, thus affecting the problem-solving process and outcomes. Common issues include missing necessary data or conditions that make calculation impossible, or incomplete or ambiguous problem statements that leave solvers uncertain about how to approach the problem. Examples: A. The original price of a book was __18 units. What is the price after a 50% discount? B. One number is 1/3 of another number's 1/4. The ratio of these two numbers is ((())/(())). C. There are 13 cooperative groups in the fourth grade, each with 4 students. If the entire class earned a total of 312 cooperation stars this semester, how many stars did each student earn on average? D. Class 6(2) has 240 science books. The number of storybooks is less by (1/6). How many fewer storybooks are there compared to science books? In example A, the lack of a part of the statement makes the problem somewhat peculiar and falls into this error type. The correct statement should be: "The original price of a book is 18 units." Example B uses parentheses to denote the answer area and does not lack information; therefore, it is not considered under this error type. Examples C and D fall into this error type due to missing or unclear descriptions.)

Table 11: Example of prompt refinement for the informal wording type

Error Type	Initial Definition and Example Generated by LLM	Definition Refined By LLM in the 2nd Round
Unit-less	缺少单位'是一种常见的数学错误，指的是在数学题目或答案中缺少必要的单位标示，导致读者无法准确理解数值的具体含义。单位是用来表示数量的标准，如千米、米、千米/时、元等。在数学问题中，单位的缺失会使得问题或答案不完整，甚至可能引起误解或错误计算。 范例A. 长方体水槽长3公尺、宽2公尺、高5公尺，请问水槽的容量是? B. 工人一天可以赚10元，工作五天可以赚多少? C. 110-10*6,其中，A 会有其他可能性(立方公分, 立方公尺) 属于此类错误，正确的写法：立方公尺。而B，钱的单位默认是'元'，所以不算在此列。C 为单纯的数学题目，因此不在此列。 ("Unit-less" is a common mathematical error, referring to the absence of necessary unit indicators in a problem or its solution, which prevents the reader from accurately understanding the specific meaning of the values. Units are standards used to express quantities, such as kilometers, meters, kilometers per hour, currency units, etc. In mathematical problems, the absence of units can render the problem or solution incomplete, potentially leading to misunderstandings or calculation errors. Examples: A. A rectangular water tank has a length of 3, a width of 2, and a height of 5. What is the capacity of the tank? B. A worker can earn 10 per day. How much can they earn in five days? C. Calculate 110 - 10 * 6. In example A, the lack of units (e.g., cubic centimeters, cubic meters) makes it fall into this error type of error. The correct specification should be cubic meters. In example B, the unit of money is implicitly understood as "currency units," so it is not considered under this error type. Example C is purely a mathematical expression and does not fall under this error type either.)	"缺少单位"是一种常见的数学错误，指的是在数学题目或答案中缺少必要的单位标示，导致读者无法准确理解数值的具体含义。单位是用来表示数量的标准，如千米、米、千米/时、元等。在数学问题中，单位的缺失会使得问题或答案不完整，甚至可能引起误解或错误计算。特别是当题目涉及多个不同的数值或比例时，缺少单位会使得这些数值之间的关系不明确，从而影响解题的准确性。 范例A. 长方体水槽长3公尺、宽2公尺、高5公尺，请问水槽的容量是? B. 工人一天可以赚10元，工作五天可以赚多少? C. 110-10*6 D. 修路队修一条公路，第一天修了全长的(1/4)，第二天修了全长的20%，两天共修了3.6千米，这条公路有多长？其中，A 会有其他可能性(立方公分, 立方公尺) 属于此类错误，正确的写法：立方公尺。而B，钱的单位默认是'元'，所以不算在此列。C 为单纯的数学题目，因此不在此列。D 属于此类。 (Unit-less is a common mathematical error, referring to the absence of necessary unit indicators in a problem or its solution, which prevents readers from accurately understanding the specific meaning of the values. Units are standards used to express quantities, such as kilometers, meters, kilometers per hour, currency units, etc. In mathematical problems, the absence of units can render the problem or solution incomplete, potentially leading to misunderstandings or calculation errors. Particularly when problems involve multiple different values or ratios, missing units can obscure the relationships between these values, thus affecting the accuracy of the solution. Examples: A. A rectangular water tank is 3 long, 2 wide, and 5 high. What is the tank's capacity? B. A worker can earn 10 per day. How much can they earn in five days? C. Calculate 110 - 10 * 6. D. A road construction team repairs a road. On the first day, they repaired (1/4) of the total length, and on the second day, they repaired 20% of the total length. In two days, they repaired 3.6 kilometers. How long is the road? In example A, the lack of units (e.g., cubic centimeters, cubic meters) makes it fall into this error type of error. The correct specification should be cubic meters. In example B, the unit of money is implicitly understood as "currency units," so it is not considered under this error type. Example C is a purely mathematical problem and does not fall into this error type. Example D lacks units for the road length and falls into this error type.)

Table 12: Example of prompt refinement for the unit-less type

Error Type	Initial Definition and Example Generated by LLM	Definition Refined By LLM in the 2nd Round
Unclear Relationship	数值意义不明确是指在数学题目中，数值的描述不够清晰或具体，导致读者无法准确理解题意或进行正确的计算。这种错误通常发生在涉及比例、倍数或百分比的题目中，当数值的关系没有明确说明时，容易引起混淆。 范例A. 鸡蛋有120个，鸭蛋比鸡蛋多(1/6)，请问两种蛋共有几个？B. 百家村要挖一条540米的水渠，第一天挖了20%，第二天挖了(1/8)，两天共挖了多少米？C. 商店去年的营业额是240万．相当今年的(4/5)，今年的营业额是多少万？其中，A 描述数量关系时缺少'倍'字, B 缺少描述的主词而属于此类错误，正确的写法：多(1/6)倍、总长的(1/8)。而C，因为'相当今年的(4/5)' 就算不用'倍'字也知道它代表是一个倍数，所以不算在此列。例如，全部的1/6, 总长的3/7, 某某的x/x, 题目单位不会有分数..等等，也不属于此错误。 (Unclear relationship refers to instances in mathematical problems where the description of the values is not sufficiently clear or specific, making it difficult for readers to accurately understand the problem or perform correct calculations. This type of error often occurs in problems involving ratios, multiples, or percentages, where the relationships between values are not clearly stated, leading to confusion. Examples: A. There are 120 chicken eggs, and the number of duck eggs is more than the chicken eggs by (1/6). How many eggs are there in total? B. A village plans to dig a 540-meter canal. On the first day, they dug 20%, and on the second day, they dug (1/8). How many meters did they dig in total over the two days? C. The store's revenue last year was 2.4 million, which is (4/5) of this year's revenue. What is this year's revenue in million units? In example A, the description of the quantity relationship lacks the word "times" and thus falls into this error type of error; the correct phrasing should be "more by (1/6) times." In example B, the subject of the description is missing, making it another example of this error; the correct phrasing should be "1/8 of the total length." Example C, however, clearly states that "4/5" refers to a multiple of this year's revenue, so it does not fall under this error type. For instance, phrases like "1/6 of the total," "3/7 of the length," or "x/x of something," where units cannot be fractions, are also not considered this type of error.)	数值意义不明确是指在数学题目中，数值的描述不够清晰或具体，导致读者无法准确理解题意或进行正确的计算。这种错误通常发生在涉及比例、倍数或百分比的题目中，当数值的关系没有明确说明时，容易引起混淆。具体情况包括：数值之间的关系不明确、缺少必要的数值说明、或数值的描述存在歧义。 范例A. 鸡蛋有120个，鸭蛋比鸡蛋多(1/6)，请问两种蛋共有几个？B. 百家村要挖一条540米的水渠，第一天挖了20%，第二天挖了(1/8)，两天共挖了多少米？C. 商店去年的营业额是240万．相当今年的(4/5)，今年的营业额是多少万？D. 一桶啤酒倒出(2/3)，刚好倒出12千克．这桶啤酒原来重多少千克．E. 一列火车从上海开往天津，行了全程的(3/5)，距天津还有538千米．这列火车已经行了多少千米？其中，A 描述数量关系时缺少'倍'字, B 缺少描述的主词而属于此类错误，正确的写法：多(1/6)倍、总长的(1/8)。而C，因为'相当今年的(4/5)' 就算不用'倍'字也知道它代表是一个倍数，所以不算在此列。例如，全部的1/6, 总长的3/7, 某某的x/x, 题目单位不会有分数..等等，也不属于此错误。D 属于此类，E 不属于此类 (Unclear relationship refers to situations in mathematical problems where the description of the values is not sufficiently clear or specific, leading to difficulties in accurately understanding the problem or performing correct calculations. This type of error often occurs in problems involving ratios, multiples, or percentages, where the relationships between values are not clearly stated, causing confusion. Specific issues include: unclear relationships between values, missing necessary numerical descriptions, or ambiguous descriptions of values. Examples: A. There are 120 chicken eggs, and the number of duck eggs is more by (1/6). How many eggs are there in total? B. The village needs to dig a 540-meter canal. On the first day, they dug 20%, and on the second day, they dug (1/8). How many meters did they dig in total over the two days? C. The store's revenue last year was 2.4 million, which is (4/5) of this year's revenue. What is this year's revenue in millions? D. A barrel of beer is (2/3) empty, with 12 kilograms poured out. How much did the barrel originally weigh in kilograms? E. A train travels from Shanghai to Tianjin, covering (3/5) of the total distance, with 538 kilometers remaining to Tianjin. How many kilometers has the train already traveled? In example A, the description of the quantity relationship lacks the word "times," making it fall into this error type of error; the correct phrasing should be "more by (1/6) times." In example B, the subject of the description is missing, making it another example of this error; the correct phrasing should be "1/8 of the total length." Example C, however, clearly states that "4/5" refers to a multiple of this year's revenue, so it does not fall under this error type. Similarly, phrases like "1/6 of the total," "3/7 of the length," or "x/x of something," where units cannot be fractions, are also not considered this type of error. Examples D and E fall into this error type due to unclear numerical meanings.)

Table 13: Example of prompt refinement for the unclear relationship type

Error Type	Initial Definition and Example Generated by LLM	Definition Refined By LLM in the 2nd Round
Calculation Error	不当数学符号是指在数学题目中使用了不正确或不明确的符号，导致题意模糊或误解。这种错误通常会使解题者无法正确理解题目要求，从而影响解题过程和结果。常见的情况包括缺少括号、使用错误的运算符号、或符号位置不当等。 范例A. 3-2/6的商，再加上5，结果=? B. 1/2+1/3+1/4=? C. 1/2+1/3+1/4=((())/(())) 其中，A 会有其他可能性(3-2/6的商+5) 属于此类错误，正确的写法：3-(2/6)+5。而B 为正常数学题目，不在此列。C 的括号使用为答案区，因此不在此列。 (Calculation error refer to the use of incorrect or unclear symbols in mathematical problems, leading to ambiguous or misunderstood problem statements. These errors can prevent the solver from correctly understanding the problem's requirements, thus affecting the problem-solving process and outcomes. Common issues include missing parentheses, incorrect use of operators, or improper placement of symbols. Examples: A. The result of 3-2/6, plus 5, equals? B. 1/2 + 1/3 + 1/4 = ? C. 1/2 + 1/3 + 1/4 = ((())/(())) In example A, there is ambiguity due to the placement of the symbols, potentially leading to Interpretation Ambiguity (3 - (2/6) + 5). The correct expression should be written as 3 - (2/6) + 5. Example B is a standard mathematical problem and does not fall into this error type. Example C uses parentheses to denote the answer area and therefore is not considered an error in symbol usage.)	不当数学符号是指在数学题目中使用了不正确、不明确或不合适的符号，导致题意模糊或误解。这种错误通常会使解题者无法正确理解题目要求，从而影响解题过程和结果。常见的情况包括缺少括号、使用错误的运算符号、符号位置不当、符号不完整或符号格式错误等。 范例A. 3-2/6的商，再加上5，结果=? B. 1/2+1/3+1/4=? C. 1/2+1/3+1/4=((())/(())) D. (10/3)与(1/3)的和比(5/6)与(4/5)的和多多少？E. 从(9/7)的倒数里减去(6/7)/(6/5)的商，差=? 其中，A 会有其他可能性(3-2/6的商+5) 属于此类错误，正确的写法：3-(2/6)+5。而B 为正常数学题目，不在此列。C 的括号使用为答案区，因此不在此列。D 不属于此类，E 不属于此类。 (Calculation error refers to the use of incorrect, unclear, or inappropriate symbols in mathematical problems, resulting in ambiguous or misunderstood problem statements. These errors often prevent solvers from accurately understanding the problem's requirements, thereby affecting the problem-solving process and outcomes. Common issues include missing parentheses, incorrect use of operators, improper placement of symbols, incomplete symbols, or incorrect symbol formatting. Examples: A. The result of 3-2/6, plus 5, equals? B. 1/2 + 1/3 + 1/4 = ? C. 1/2 + 1/3 + 1/4 = ((())/(())) D. How much more is the sum of (10/3) and (1/3) compared to the sum of (5/6) and (4/5)? E. The difference when subtracting the quotient of (6/7) divided by (6/5) from the reciprocal of (9/7) is? In example A, there is ambiguity due to the placement of the symbols, potentially leading to Interpretation Ambiguity (3 - (2/6) + 5). The correct expression should be written as 3 - (2/6) + 5. Example B is a standard mathematical problem and does not fall into this error type. Example C uses parentheses to denote the answer area and therefore is not considered an error in symbol usage. Examples D and E are also not considered under this error type.)

Table 14: Example of prompt refinement for the calculation error type