Beyond Accuracy: Alignment and Error Detection across Languages in the Bi-GSM8K Math-Teaching Benchmark

Anonymous ACL submission

Abstract

Recent advancements in Large Language Models (LLMs) have significantly improved mathematical problem-solving, with models like GPT-4 achieving human-level performance. However, proficiently solving mathematical problems differs fundamentally from effectively teaching mathematics. To bridge this gap, we introduce the Bi-GSM8K benchmark, a bilingual English-Korean dataset enriched with authentic teacher-generated solutions, student-generated solutions, and annotations marking students' initial errors. This dataset facilitates a comprehensive evaluation of how closely LLM-generated solutions align with human educators' reasoning and the precision of LLMs in detecting initial student errors. Our experiments showed alignment exact match rates between student and teacher solutions of 74.4% for English and 75.0% for Korean. We further evaluated various commercially available and open-source LLMs, highlighting GPT-4o's superior accuracy in initial error detection while recognizing open-source models' computational efficiency advantages. Our key contributions include the open-source release of Bi-GSM8K, novel evaluation metrics, and comparative analyses of LLM performance across languages.

1 Introduction

Recent advancements in LLMs have led to significant progress in mathematical problem-solving tasks. Notably, GPT-4 has achieved accuracy rates of 97% and 86% on GSM8K and MMLU benchmarks, respectively, demonstrating performance comparable to expert human levels. Additionally, OpenAI's o1 model has attained an accuracy of 74.4% (pass@1) on the AIME problems, further evidencing its advanced reasoning capabilities (Zhong et al., 2024; Achiam et al., 2023; OpenAI, 2024). These results indicate that LLMs have evolved from mere language-understanding tools into sophisticated instruments capable of logical reasoning and computational problem-solving.

However, effectively solving mathematical problems and proficiently teaching mathematics to students constitute fundamentally distinct tasks. Prior research has established that the competencies involved in effectively solving mathematical problems, termed Common Content Knowledge (CCK), differ significantly from the Mathematical Knowledge for Teaching (MKT) required for effective pedagogical practice (Understand, 1986; Ball et al., 2008). Within educational contexts, this implies that merely providing correct answers is insufficient; it is crucial to understand the student's thought processes and diagnose their errors accurately (Copur-Gencturk and Tolar, 2022; Daheim et al., 2024; Sonkar et al., 2024).

This perspective is increasingly prevalent in contemporary research on mathematics education using LLMs. In particular, there is a growing consensus that aligning LLMs to think like experienced educators rather than merely serving as answer-generating machines maximizes educational effectiveness. Recent studies have emphasized that, similar to human teachers, LLMs must engage in diagnosing errors and providing feedback based on students' solution processes when imparting Pedagogical Content Knowledge (PCK) (Jiang et al., 2024; Hu et al., 2025).

In this context, from a learning efficiency perspective, we emphasize the necessity for evaluation metrics that assess Large Language Models (LLMs) beyond simply providing direct answers. Specifically, these metrics should evaluate: (1) how closely an LLM-generated solution aligns with the solution processes of human educators, and (2) the accuracy with which an LLM identifies the initial point of error in student-generated responses. To facilitate such evaluation, it is essential to first establish evaluation datasets that include authentic solution processes generated by teachers. However, existing datasets, such as GSM8K, contain only mathematical problems accompanied by solved answers, lacking genuine teacher-generated solution processes (Cobbe et al., 2021). To address this gap, we augment the GSM8K dataset by incorporating real teacher-generated solutions, thus creating the Bi-GSM8K benchmark. Furthermore, we extend this benchmark to include student-generated solution processes annotated explicitly with labels marking students' initial errors. Finally, we translate the augmented dataset into a bilingual Korean-English corpus, enabling the analysis of linguistic differences in mathematical problem-solving.

Using this benchmark, we conducted experiments evaluating the alignment accuracy of various LLM-generated and student-generated solutions compared to authentic teacher solutions. Results indicate alignment accuracy rates of 74.4% for English and 75.0% for Korean in matching student solutions with teacher solutions. We further compared multiple commercially available open-source LLMs with GPT-4o, assessing both accuracy and latency to determine practical applicability. Since the Bi-GSM8K benchmark includes annotations marking students' initial errors, it enabled evaluation of LLMs' proficiency in detecting these initial mistakes. In this evaluation, GPT-4o demonstrated significantly superior performance. However, certain open-source models exhibited faster error analysis capabilities than larger commercial models, highlighting their potential advantage in practical educational contexts.

- Construction and open-source release of Bi-GSM8K, a mathematical educational benchmark dataset including authentic human-generated solution processes

- Proposal of novel evaluation metrics to assess the accuracy of mathematical solution processes

- Comparative analysis of alignment accuracy between LLM-generated and teacher-generated answers, utilizing various LLMs and examining differences across languages

2 Related Work

In this section, we review existing research related to mathematical education and LLMs by analyzing the characteristics influenced by problem-solving approaches, instructional methods, and linguistic differences.

2.1 Utilization and Limitations of LLMs in Mathematics Education

Recently, there has been growing interest in leveraging LLMs within mathematics education, primarily focusing on problem-solving and tutoring. However, their performance differs significantly between these domains. For instance, while LLMs achieve an accuracy of 85.5% in algebraic problem-solving tasks, their accuracy in generating educational dialogues as tutors is limited to 56.6%, highlighting frequent errors in tutoring scenarios (Gupta et al., 2025). Integrating stepwise error detection models has significantly enhanced the accuracy, clarity, and educational validity of LLM-generated feedback compared to standard methods (Daheim et al., 2024).

Furthermore, LLMs fine-tuned on mathematical tutoring datasets such as MATHDIAL can provide more equitable and accurate feedback; however, limitations remain in error prevention, correction, and fully replacing human (Macina et al., 2023). Although LLMs exhibit human-like performance in replicating intelligent tutoring feedback or automatic grading, their effectiveness diminishes significantly with novel errors, student cognition diagnosis, or irrelevant content avoidance (Gupta et al., 2025; Daheim et al., 2024; Macina et al., 2023; McNichols et al., 2024; Jin et al., 2025; Baral et al., 2024; Jin et al., 2024).

2.2 Challenges in Non-English Educational Environments

The performance of LLMs on mathematical problems formulated in non-English languages remains a significant challenge requiring further improvement. For example, Wei et al. analyzed the performance of various LLMs using the CMATH dataset, which consists of elementary mathematics problems in Chinese, and found that GPT-4 achieved an accuracy above 60%, whereas most other LLMs showed considerably lower performance on non-English mathematics tasks. Additionally, Nguyen et al. applied ChatGPT to Korean middle and high school mathematics problems, reporting an accuracy rate of 66.7%. Although this performance is lower compared to English problems, the study suggests that LLMs remain useful in evaluating student responses. Moreover, multimodal-based assessments such as those using the KoNET dataset have revealed substantial performance degradation of AI models when applied within Korean high school educational settings (Park and Kim, 2025). These studies clearly demonstrate ongoing performance limitations of LLMs in non-English educational contexts.

2.3 Research Trends of LLMs for Educational Purposes

Research on employing LLMs for educational purposes broadly divides into two streams. The first stream emphasizes enhancing mathematical problem-solving skills, utilizing LLMs to generate customized mathematical problems or solution methods to fine-tune student models and boost learning effectiveness (Liang et al., 2023). The second stream employs LLMs as tutors to simultaneously enhance students' learning outcomes and feedback generation capabilities (Scarlatos et al., 2025). Approaches include training LLMs using students' learning outcomes as reward signals, as well as employing schema-based strategies and role-based prompts to generate structured and pedagogically beneficial feedback (Dixit and Oates, 2024; Hu et al., 2025; Scarlatos et al., 2025).

These studies indicate the potential utility of LLMs in diverse pedagogical practices within mathematics education, such as problem generation, problem-solving, feedback provision, and instructional design. Nevertheless, concerns remain regarding the accuracy of generated outputs, the support for autonomous learning, and overall research reliability, underscoring ongoing areas for improvement.

3 The Bi-GSM8K Dataset

What considerations are necessary for LLMs to effectively teach mathematical problem-solving in a manner comparable to human educators? A fundamental requirement for addressing this issue is to systematically devise evaluation methodologies capable of assessing how closely LLM-generated responses resemble authentic teacher-generated solution processes. To facilitate the training and evaluation of the proposed mathematical tutoring system, we constructed the Bi-GSM8K dataset, which comprises elementary-level mathematical problem-solving processes. This dataset plays a critical role in detecting student errors and analyzing their learning states. Specifically, Bi-GSM8K contains structured solution procedures and includes both correct and incorrect solution examples. The dataset was entirely developed in Korean by domestic domain experts, with an English version provided via automatic translation.

We constructed the Bi-GSM8K dataset by automatically translating the original English GSM8K dataset into Korean, subsequently adapting it to align with the Korean educational curriculum, and meticulously correcting translation errors. The dataset consists of a total of 500 items, each provided in JSON format. Each item includes curriculum-aligned fields indicating the educational domain and unit, as well as detailed sections labeled problem, solution, the teacher-generated correct solution (correct_solution), and the student-generated erroneous solution (error_solution).

```
{
  "area": "problem",
  "problem": "Byeongjin went fishing with his family
yesterday. Byeongjin caught 4 fish, his wife caught 1,
the eldest son caught 3, the younger son caught 2, and
the youngest daughter caught 5. Unfortunately, 3 of the
fish were too small and were released back. If each fish
yields 2 fillets, how many fillets can Byeongjin's family
make?",
  "solution": "Four hats with 3 stripes each have a total
of 4×3=«43=12»12 stripes.\n Three hats with 4 stripes
each have a total of 3×4=«34=12»12 stripes.\n Six hats
with no stripes have 6×0=«60=0»0 stripes.\n And two hats
with 5 stripes each have 2×5=«25=10»10 stripes.\n The
total number of stripes on Byungjin's hats is
12+12+0+10=«12+12+0+10=34»34 stripes.\n #### 34",
  "correct_solution": {
    "step_1": "Byung-jin's family caught 4 + 1 + 3 + 2 +
5 = «4+1+3+2+5=15»15 fish.",
    "step_2": "15 - 3 = «15-3=12» I stored 12 fish.",
    "step_3": "Since you can obtain 2 fillets from each
fish, for 12 fish, you have 12 fish * 2 fillets per fish
= 24 fillets."
  },
  "error_solution": {
    "step_1": "4+1+3+2+5 = 15",
    "step_2": "15 * 2 = 30",
    "step_3": "Answer: 30 fillets"
  }
}
```

Table 1: In the proposed Bi-GSM8K dataset, examples are expanded beyond the original GSM8K format, which included only "Problem" and "Solution" fields. Bi-GSM8K additionally provides the teacher-generated correct solution (`correct_solution`) and an erroneous student-generated solution (`error_solution`). Furthermore, the Bi-GSM8K dataset is offered as a bilingual Korean-English corpus.

Given that the problems were translated from English, we performed an extensive validation process. Proper nouns, units, and expressions were adjusted to fit the standards of the Korean curriculum, and errors originating from machine translation were carefully revised. Teacher-generated solutions were then added, and their accuracy was thoroughly reviewed, applying necessary corrections. Finally, student-generated solutions were manually created to intentionally reflect realistic student errors while closely adhering to the teacher-generated correct solutions. Mathematical expressions were formatted following the GSM8K standard by enclosing them within "« »" symbols. Following this meticulous process, the Bi-GSM8K dataset was finalized. Table 1 illustrates a representative example from the Bi-GSM8K dataset.

The Bi-GSM8K dataset possesses the following key features:

- **Automated Translation and Review**: The original GSM8K dataset was translated into Korean via automated translation, then revised to align with the Korean curriculum, with corrections made for proper nouns, units, and translation errors. Correct solutions underwent multiple reviews for accuracy.

- **Inclusion of Student Solutions**: By incorporating varied student errors into student-generated solutions and aligning them with correct solutions, the dataset supports error tracking, learning state analysis, and customized feedback generation.

- **Initial Error Annotation**: Each item provides both the teacher's correct solution and the student's erroneous solution, explicitly annotating the initial error point within the student's solution, thereby assisting accurate error diagnosis and remediation by the model.

The Bi-GSM8K dataset provides a structured data format enabling detailed analysis of errors occurring within student-generated solutions. By tracking specific error points, the dataset facilitates the identification of student error patterns and enhances the precision of error diagnosis.

4 Evaluation Metric

The Bi-GSM8K benchmark proposed in this study provides data enabling the evaluation of solution processes generated by LLMs by directly comparing them with those produced by experienced human teachers. To achieve this comparative evaluation, appropriate assessment methodologies are necessary. Specifically, this study introduces two similarity-based evaluation methods: (1) Ground Truth Alignment (GTA), which assesses how closely generated solutions align with correct teacher-generated solutions, and (2) Solution Error Detection (SED), which identifies the initial error point within student-generated solutions, thereby systematically evaluating the capabilities of language models.

4.1 Ground Truth Alignment

Evaluating narrative-style solutions of mathematical problems quantitatively is inherently challenging. Therefore, we propose GTA, which assesses solution quality based on the similarity between the predicted and ground-truth solutions. The similarity measurement involves two key steps: (1) assessing semantic similarity using measures such as Cosine Similarity, Pearson Correlation, SemScore, and BERTScore (Aynetdinov and Akbik, 2024; Zhang et al., 2019), and (2) determining structural similarity using the Needleman-Wunsch (NW) algorithm to align solution sequences needleman1970general. This dual-step process identifies discrepancies and optimizes the similarity of reasoning paths.

Algorithm 1 NW Algorithm with Semantic Similarity for Ground Truth Alignment

Input: $s1, s2, sim\_m, sim\_th$
Output: x_aln, y_aln
$m \leftarrow \text{len}(s1), n \leftarrow \text{len}(s2)$
Initialize bt_table of size $(m+1, n+1)$
for $i = 0$ to m **do**
  $bt\_table[i][0] \leftarrow 1$ ▷ From up
end for
for $j = 0$ to n **do**
  $bt\_table[0][j] \leftarrow 2$ ▷ From left
end for
for $i = 1$ to m **do**
  for $j = 1$ to n **do**
    $m\_sc \leftarrow score[i-1][j-1] + sim\_m[i-1][j-1]$
    $gap\_p \leftarrow gap\_u \times (1 - sim\_m[i-1][j-1])$
    $u\_sc \leftarrow score[i-1][j] - gap\_p$
    $l\_sc \leftarrow score[i][j-1] - gap\_p$
    $bt\_table[i][j] \leftarrow \text{argmax}(m\_sc, u\_sc, l\_sc)$
  end for
end for
$i \leftarrow m, j \leftarrow n$
Initialize x_aln, y_aln
while $i > 0$ or $j > 0$ **do**
  if $bt\_table[i][j] = 0$ **then**
    if $sim\_m[i-1][j-1] \geq sim\_th$ **then**
      Append aligned values to x_aln, y_aln
    end if
  end if
  Update indices i and j
end while
Reverse x_aln, y_aln
Return x_aln, y_aln

In this study, we modified the conventional NW algorithm specifically for aligning mathematical solution processes. The NW algorithm numerically quantifies the similarity between two strings and identifies the most similar alignment, making it suitable for static similarity comparisons of lengthy texts. Typically, when comparing two texts, the NW algorithm assigns fixed penalty scores for character insertions or deletions. However, mathematical solution processes inherently involve both sequential

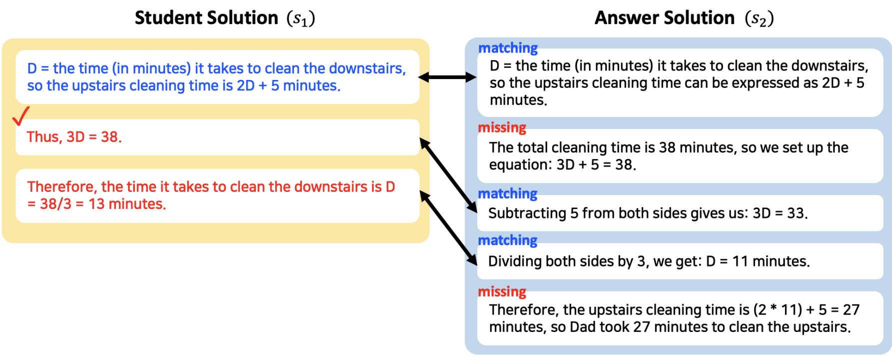

Figure 1: Upon submission of a student's solution to the system, an initial comparison and alignment with the correct solution is performed. This alignment employs Language Models (LMs), similarity functions, and the NW algorithm to systematically analyze omitted steps or extraneous information in the student's solution. Subsequently, an independent LLM-based error detection model operates separately from the alignment process to precisely identify the initial point of error within the student's solution.

order and detailed semantic content. Therefore, we adjusted gap penalties according to semantic similarity metrics (e.g., Cosine similarity, BERTScore), assigning smaller penalties to semantically similar sentences and larger penalties to dissimilar ones. Additionally, matching scores are computed based on substring similarities, enabling natural and precise alignment between solution steps. This allows clear identification of differences between two solutions and facilitates alignment consistent with the problem-solving flow.

Algorithm 1 describes the proposed procedure for computing alignment scores between two solutions (examples in Figure 1). Here, s_1 represents the teacher-generated solution with m steps, and s_2 denotes the student-generated solution with n steps. The inputs are two sequences (s_1, s_2), a similarity matrix (sim_m) indicating semantic similarities between solution steps, and a similarity threshold (sim_th). A backtracking table bt_table for dynamic programming is initialized (line 4) with dimensions $(m + 1) \times (n + 1)$, storing directional moves for reconstructing optimal alignments. The first row and column of bt_table are initialized to represent leftward and upward movements, respectively.

Next, a backtracking table bt_table for dynamic programming is initialized (line 4). The table has dimensions $(m+1)\times(n+1)$, with each cell recording the optimal move direction for backtracking. The first row and first column of bt_table are initialized with left (represented by 2) and upward (represented by 1) movements, respectively.

Subsequently, the remainder of bt_table is filled using two nested loops. At each cell (i, j), values for three possible movements are calculated, and the maximum value is selected and recorded in $bt\_table[i][j]$. Diagonal movements (matches) add the value from the diagonal cell and $sim\_m[i-1][j-1]$; upward movements (deletions) add penalties to values from the cell above; leftward movements (insertions) add penalties to values from the cell to the left. Once the table is fully populated, an optimal alignment between the sequences is determined by backtracking through bt_table.

During the backtracking phase, alignments are determined according to each cell's recorded movement direction. For diagonal movements (represented by 0), if the similarity is below the threshold sim_th, the element $s_1[i-1]$ is aligned as an "omission". If the similarity meets or exceeds sim_th, the algorithm checks for duplicate alignments. Specifically, if $s_2[j-1]$ is already aligned with another element in y_aln, the algorithm compares similarity scores between the existing and current alignments. If the existing alignment has a higher similarity score, the current element $s_1[i-1]$ is aligned as an "omission"; otherwise, the existing alignment is replaced with the current one. If no duplication occurs, the two elements are directly aligned. For upward movements (represented by 1), the element $s_1[i-1]$ is aligned as an "omission". For leftward movements (represented by 2), an "unnecessary" is aligned with the element $s_2[j-1]$. After backtracking completes, the aligned sequences $x\_aln.reverse()$ and $y\_aln.reverse()$, along with the similarity matrix sim_m, are returned in the correct order.

Compared with conventional similarity-based scoring (NW or cosine similarity), GTA aligns each intermediate reasoning step with its ground-truth counterpart, producing a fine-grained score that captures logical coherence rather than superficial lexical overlap. This step-aware alignment not only yields stronger correlations with true solution quality but also pinpoints where a learner's reasoning diverges, enabling interpretable diagnostics and targeted feedback.

4.2 Solution Error Detection

Identifying the initial error step is particularly crucial in personalized tutoring scenarios. Precisely pinpointing the moment when a student deviates from the correct problem-solving strategy enables the system to gain deeper insights into the student's conceptual understanding. This module aims to accurately identify the initial erroneous step in a student's mathematical problem-solving process.

```
Your goal is to compare the correct solution to the student solution and output the steps the student needs to review again. If the student's solution is correct, output 0.

Problem: Sohee feels bored with her current game and decides to play a new one. In the new game, 80% of the 100 hours of gameplay consists of repetitive and boring stages. However, through an expansion pack, she can add 30 hours of enjoyable stages. Including the expansion pack, how many hours of enjoyable stages can Sohee play?
Answer Solution: {
  "step_1":"There are 1000.8=«1000.8=80»80 hours of boring stages in the game.",
  "step_2":"The enjoyable gameplay time is 100-80=«100-80=20»20 hours.",
  "step_3":"With the expansion pack, the enjoyable gameplay time increases to 20+30=«20+30=50»50 hours."
}
Student Solution: {
  "step_1": "There are 1000.8=«1000.8=80»80 hours of boring stages in the game.",
  "step_2":"The enjoyable gameplay time is 100-80=«100-80=20»20 hours.",
  "step_3":"The expansion pack has 300.8=«300.8=24»24 hours of boring stages.",
  "step_4":"The enjoyable gameplay time in the expansion pack is 30-24=«30-24=6»6 hours.",
  "step_5":"The total enjoyable gameplay time is 50+6=«50+6=56»56 hours."
}

Q: Is the Student Solution incorrect? Write only the step number with the first error or 0 if no error is found.
A: 2

Problem: {problem}
Answer Solution: {answer_solution}
Student Solution: {student_solution}
Q: Is the Student Solution incorrect? Write only the step number with the first error or 0 if no error is found.
A:
```

Table 2: Prompt Template for Solution Error Detection

The initial error detection module proposed in this study leverages open-source LLMs and employs a few-shot learning approach, enabling accurate error identification with only a small number of examples. Specifically, we adopt a 3-shot learning setup, where each example comprises a mathematical problem, an Answer Solution, a Student Solution, and a Q(Question) for determining the presence of an error. The model compares the correct solution with the student's solution and outputs the step number corresponding to the first incorrect reasoning; it returns 0 if no error is detected.

Table 2 presents an example of the prompt structure used in this task. Initially, the prompt provides task instructions, followed by three example problems along with their correct and student-generated solutions. Subsequently, it includes a query prompting the determination of whether an error has occurred in the student's solution and, if so, to identify the initial erroneous step.

5 Experiment

5.1 Experiment Settings and Models

In this study, we evaluated various open-source LLMs using the mathematical Bi-GSM8K dataset. The models selected for experimentation were assessed comprehensively based on multilingual processing capabilities, mathematical reasoning abilities, response consistency, and computational efficiency. Models ranging from 7B to 8B parameters were specifically chosen due to their balance between performance and practicality, making them suitable for real-world educational contexts. Additionally, for the GTA module, BERTScore was employed to measure nuanced semantic similarity between the generated texts and ground-truth solutions. This was facilitated by adopting multiple pretrained transformer models. The primary hyperparameters for the open-source models were set to a temperature of 1, a top_p of 0.75, a top_k of 40, and num_beams of 4. For GPT-4o, the temperature was adjusted to 0.75 to enhance response consistency and stability.

5.2 Ground Truth Alignment

In this section, we evaluate the alignment performance between student-generated solutions produced by the GPT-4o model and the teacher-generated correct solutions. Table 3 summarizes evaluation results of the GTA system across various similarity metrics, models, and thresholds. It reports exact match accuracy for English (EN) and Korean (KO) datasets, along with computational efficiency metrics such as latency (seconds) and peak memory usage (MB). Except for the exact match accuracy, all other values in Table 3 represent averages of results obtained from the En-

Similarity	Model	Threshold	Exact Match EN	Exact Match KO	Latency (sec)	Peak Memory (MB)
-	**GPT-4o**	-	**68.4**	**82.0**	3.372	-
Cosine Pearson Semscore	Llama-3.1-8B-Instruct	0.6	66.6	66.8	0.427	28681.48
	DeepSeek-R1-Distill-Llama-8B	0.5	67.4	67.4	0.429	28681.48
	DeepSeek-llama3.1-Bllossom-8B	0.6	67.2	66.2	0.389	28658.35
	Qwen2.5-7B-Instruct	0.5	63.2	65.2	0.406	27178.01
	DeepSeek-R1-Distill-Qwen-7B	0.7	66.2	60.8	0.373	27178.21
	DeepSeek-R1-Distill-Qwen-7B-Multilingual	0.7	65.4	61.2	0.398	27162.12
	Mistral-7B-Instruct-v0.3	0.7	66.8	67.2	0.501	27221.69
	Phi-4-mini-instruct	0.5	62.8	65.4	0.183	14728.41
BertScore	bart-large	0.6	71.8	72.0	**0.070**	**703.91**
	bert-large-uncased	0.5	**74.4**	72.8	0.113	1010.21
	deberta-v2-xlarge-mnli	0.7	72.6	**75.0**	0.213	2683.69
	roberta-large	0.8	65.6	66.6	0.109	1040.56

Table 3: Comparative evaluation of Ground Truth Alignment scores, thresholds, and memory usage for each model

glish and Korean datasets. For the alignment performance evaluation between student-generated and teacher-generated solutions, 500 problems from the Bi-GSM8K dataset were utilized. Similarity thresholds of $[0.5, 0.6, 0.7, 0.8, 0.9]$ were applied to each similarity metric. Detailed alignment examples are provided in Appendix C.

The large-scale commercial API GPT-4o, used as a baseline, achieved the highest performance with accuracies of 68.4% on the English dataset and 82.0% on the Korean dataset, demonstrating its advanced capabilities in language understanding and reasoning.

In comparisons across similarity measurement methods, BertScore-based models consistently outperformed vector similarity-based models (Cosine, Pearson, Semscore). Specifically, the bert-large-uncased model recorded an accuracy of 74.4% on the English dataset, approximately 6 percentage points higher than GPT-4o. On the Korean dataset, the deberta-v2-xlarge-mnli model exhibited the best performance with 75.0% accuracy. These results suggest that the contextualized embeddings utilized in BERT-based models excel at detecting semantic similarities, enabling more precise alignment between student-generated and teacher-generated solutions. Consequently, the importance of context-aware embeddings in solution alignment tasks is emphasized.

From a computational efficiency standpoint, larger transformer models such as Llama-3.1-8B and the DeepSeek series had an average latency of approximately 0.4 seconds, while bart-large and bert-large-uncased models demonstrated significantly faster inference speeds at 0.07 seconds and 0.11 seconds, respectively. Memory usage showed similar patterns; LLM-based models required substantial memory exceeding approximately 27,000 MB, whereas BERT-based models utilized significantly less, generally under 3,000 MB. Notably, the bart-large model operated with only around 700 MB of memory, making it viable even in resource-constrained environments. These experimental results clearly illustrate a trade-off between accuracy and computational efficiency. GPT-4o delivered the highest accuracy but faced limitations in real-world tutoring applications due to computational overhead and associated costs. Among more accessible models, bert-large-uncased and deberta-v2-xlarge-mnli models, utilizing BertScore, demonstrated the most effective balance between performance and efficiency.

5.3 Solution Error Detection

In this section, we evaluated the accuracy of various LLMs in identifying the initial errors made by students during mathematical problem-solving. Table 4 summarizes the results of the SED task across multiple LLMs, reporting accuracy for both English (EN) and Korean (KO) datasets, along with computational efficiency metrics (throughput: requests/sec, latency: seconds, memory usage: MB). The evaluation utilized 500 items from the Bi-GSM8K dataset, with detailed alignment examples presented in Appendix D.

GPT-4o, employed as a baseline commercial API, demonstrated the highest accuracy, achieving 94.4% for English and 95.8% for Korean, indicating top-tier performance. Particularly notable is GPT-4o's superior accuracy on Korean data, suggesting strong suitability for Korean-language mathematics tutoring applications. Moreover, GPT-4o achieved relatively rapid response times, with a throughput of 0.1309 requests/sec and latency of 8.39 seconds, highlighting both its sophisticated linguistic reasoning capabilities and computational

Model	Accuracy		Throughput	Latency	Peak Memory
	EN	KO	(requests/sec)	(sec)	(MB)
GPT-4o	**94.4**	**95.8**	**0.1309**	**8.39**	-
Llama-3.1-8B-Instruct	80.4	75.2	0.0014	711.60	10896.85
DeepSeek-R1-Distill-Llama-8B	55.4	57.8	0.0014	720.06	10898.43
DeepSeek-llama3.1-Bllossom-8B	63.2	62.0	0.0014	716.86	10851.40
Qwen2.5-7B-Instruct	79.4	**86.0**	0.0013	783.91	11279.73
DeepSeek-R1-Distill-Qwen-7B	82.8	80.4	0.0013	801.63	11328.93
DeepSeek-R1-Distill-Qwen-7B-Multilingual	**86.4**	83.4	0.0012	847.68	11281.79
Mistral-7B-Instruct-v0.3	67.0	67.0	0.0009	1168.91	15120.72
Phi-4-mini-instruct	76.6	78.6	**0.0029**	**355.01**	**8075.12**

Table 4: Comparative evaluation of Solution Error Detection scores, thresholds, and memory usage for each model

efficiency. Nonetheless, multilingual models based on Qwen demonstrated relatively stronger performance compared to single-language and basic models. For example, DeepSeek-R1-Distill-Qwen-7B-Multilingual achieved commendable accuracy of 86.4% in English and 83.4% in Korean, while the single-language counterpart, Qwen2.5-7B-Instruct, attained 86.0% accuracy in Korean.

Conversely, lightweight models based on the Llama series exhibited lower accuracy. Llama-3.1-8B-Instruct achieved moderate accuracy levels of 80.4% in English and 75.2% in Korean, whereas DeepSeek-R1-Distill-Llama-8B performed poorly with accuracies of 55.4% and 57.8% in English and Korean, respectively. Notably, the DeepSeek-llama3.1-Bllossom-8B model, trained to think in English and output in the input language, demonstrated negative performance impacts upon integrating Korean data. This suggests that English-centric cognitive strategies were inadequately adapted for the Korean context or that the semantic quality of training data was compromised, ultimately hindering mathematical reasoning and error detection capabilities.

In terms of computational efficiency, the Phi-4-mini-instruct model exhibited exceptional resource efficiency with a throughput of 0.0029 requests/sec, latency of 355.01 seconds, and memory usage of 8075.12MB; however, its low accuracy limits its practicality for real-time tutoring requiring immediate feedback. Most open-source models showed comparable or higher throughput and shorter latencies than GPT-4o, advantageous for real-time responsiveness, yet their performance remains limited in complex problem-solving and deep linguistic comprehension.

In this experiment, we evaluated various LLMs on their ability to detect initial errors in students' mathematical solution processes. Open-source models generally underperformed commercial models in accuracy and computational efficiency, with multilingual models exhibiting notable performance variability across languages. Particularly, Llama series models trained primarily on English data experienced performance degradation when incorporating Korean data, underscoring the necessity for language-specific optimization and enhanced data quality for complex mathematical reasoning tasks.

6 Conclusion

In this study, we introduced Bi-GSM8K, a bilingual English-Korean benchmark dataset for mathematical problem-solving, constructed using student and teacher solution processes.

This dataset provides foundational resources for mathematics education, facilitating comprehensive evaluation of alignment accuracy between student-generated and teacher-generated solutions, as well as assessing initial error detection performance across various LLMs. Specifically, we introduced an analytical approach combining similarity metrics, GPT, and the Needleman-Wunsch algorithm to measure the semantic similarity between student-generated and teacher-generated solutions.

Evaluation results indicate that certain recent advanced open-source models exhibit performance levels comparable or superior to commercial models such as GPT-4o in terms of accuracy and computational efficiency. Specifically, BERT-based models utilizing BertScore demonstrated high alignment accuracy relative to computational efficiency, effectively distinguishing agreement between learner-generated and teacher-generated solutions. Furthermore, separate experiments conducted for English and Korean datasets revealed distinct performance variations across languages. In future research, we aim to validate the educational effectiveness of the system through interactive experiments with real learners and educators.

Limitations

Limited Reflection of Authentic Learning Environments in Dataset

The dataset utilized in this study comprises elementary-level mathematical problems and tutoring dialogues, thus limiting the scope to specific grades and difficulty levels, which constrains the representation of diverse problem types. Additionally, the student-generated mathematical solutions were created by domain experts, potentially insufficiently reflecting the varied cognitive processes and errors exhibited by real students. Consequently, future research should focus on developing datasets that encompass a broader range of problem types, languages, and authentic learner-generated data.

Necessity for Improved Performance in Complex Mathematical Reasoning

The open-source LLMs employed in this study demonstrate performance degradation on tasks requiring complex mathematical reasoning. This limitation is likely due to insufficient training on high-dimensional reasoning tasks or inherent difficulties in processing mathematical expressions in certain languages. To overcome these limitations, future research should emphasize domain-specific training and integrate Retrieval-Augmented Generation techniques to enhance reasoning capabilities.

Evaluation in Real Educational Environments

While this research evaluates system performance through quantitative data-driven analyses, direct verification of its applicability in actual classroom or tutoring environments has not been performed. In real educational settings, factors such as student responses, class dynamics, and teacher interventions significantly influence system effectiveness. Therefore, comprehensive evaluations of system practicality and educational efficacy require implementation in authentic educational contexts. Future studies should conduct experiments involving teachers and students to measure educational effectiveness and derive feedback mechanisms and interface improvements that meet real-world demands.

Limitations and Improvement Directions for Solution Alignment Representation

In this study, we performed sentence-level alignment between student-generated and correct (teacher-generated) solutions. However, this approach exhibits limitations in alignment accuracy, as a single sentence may often encompass multiple solution steps. To precisely model students' reasoning processes and effectively capture the intricate relationships between student and teacher solutions, a hierarchical and multi-layered representational approach is required. Consequently, developing novel alignment methods capable of reflecting such complex structures is proposed as an important direction for future research.

Need for Improvement in Model Efficiency and Practicality

Most models evaluated in this study exhibited a trade-off between accuracy and efficiency. Larger models provided high accuracy at substantial computational costs, whereas smaller models offered higher efficiency but lower performance. Future research should prioritize enhancing small-model performance and optimizing computational efficiency using hardware acceleration technologies, facilitating the development of real-time feedback systems and improving model practicality.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ansar Aynetdinov and Alan Akbik. 2024. Semscore: Automated evaluation of instruction-tuned llms based on semantic textual similarity. *arXiv preprint arXiv:2401.17072*.

Deborah Loewenberg Ball, Mark Hoover Thames, and Geoffrey Phelps. 2008. Content knowledge for teaching: What makes it special?

Sami Baral, Eamon Worden, Wen-Chiang Lim, Zhuang Luo, Christopher Santorelli, and Ashish Gurung. 2024. Automated assessment in math education: A comparative analysis of llms for open-ended responses.

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*.

Yasemin Copur-Gencturk and Tammy Tolar. 2022. Mathematics teaching expertise: A study of the dimensionality of content knowledge, pedagogical content knowledge, and content-specific noticing skills. *Teaching and Teacher Education*, 114:103696.

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*.

Prakhar Dixit and Tim Oates. 2024. Sbi-rag: Enhancing math word problem solving for students through schema-based instruction and retrieval-augmented generation. *arXiv preprint arXiv:2410.13293*.

Adit Gupta, Jennifer Reddig, Tommaso Calo, Daniel Weitekamp, and Christopher J MacLellan. 2025. Beyond final answers: Evaluating large language models for math tutoring. *arXiv preprint arXiv:2503.16460*.

Bihao Hu, Jiayi Zhu, Yiying Pei, and Xiaoqing Gu. 2025. Exploring the potential of llm to enhance teaching plans through teaching simulation. *npj Science of Learning*, 10(1):7.

Zhuoxuan Jiang, Haoyuan Peng, Shanshan Feng, Fan Li, and Dongsheng Li. 2024. Llms can find mathematical reasoning mistakes by pedagogical chain-of-thought. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 3439–3447. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Hyoungwook Jin, Yoonsu Kim, Dongyun Jung, Seungju Kim, Kiyoon Choi, Jinho Son, and Juho Kim. 2025. Investigating large language models in diagnosing students' cognitive skills in math problem-solving. *arXiv preprint arXiv:2504.00843*.

Hyoungwook Jin, Yoonsu Kim, Yeon Su Park, Bekzat Tilekbay, Jinho Son, and Juho Kim. 2024. Using large language models to diagnose math problem-solving skills at scale. In *Proceedings of the Eleventh ACM Conference on Learning@ Scale*, pages 471–475.

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*.

Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. Mathdial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. *arXiv preprint arXiv:2305.14536*.

Hunter McNichols, Jaewook Lee, Stephen Fancsali, Steve Ritter, and Andrew Lan. 2024. Can large language models replicate its feedback on open-ended math questions? *arXiv preprint arXiv:2405.06414*.

Phuong-Nam Nguyen, Quang Nguyen-The, An Vu-Minh, Diep-Anh Nguyen, and Xuan-Lam Pham. 2025. On the robustness of chatgpt in teaching korean mathematics. *arXiv preprint arXiv:2502.11915*.

OpenAI. 2024. Learning to reason with large language models. `https://openai.com/index/learning-to-reason-with-llms/`.

Sanghee Park and Geewook Kim. 2025. Evaluating multimodal generative ai with korean educational standards. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 671–688.

Alexander Scarlatos, Naiming Liu, Jaewook Lee, Richard Baraniuk, and Andrew Lan. 2025. Training llm-based tutors to improve student learning outcomes in dialogues. *arXiv preprint arXiv:2503.06424*.

Shashank Sonkar, Kangqi Ni, Sapana Chaudhary, and Richard G Baraniuk. 2024. Pedagogical alignment of large language models. *arXiv preprint arXiv:2402.05000*.

Those Who Understand. 1986. Knowledge growth in teaching. *Educational Researcher*, 15(2):4–14.

Tianwen Wei, Jian Luan, Wei Liu, Shuang Dong, and Bin Wang. 2023. Cmath: Can your language model pass chinese elementary school math test? *arXiv preprint arXiv:2306.16636*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Qihuang Zhong, Kang Wang, Ziyang Xu, Juhua Liu, Liang Ding, and Bo Du. 2024. Achieving> 97% on gsm8k: Deeply understanding the problems makes llms better solvers for math word problems. *arXiv preprint arXiv:2404.14963*.

A Analysis of Problem Lengths

In this appendix, we present a quantitative statistical analysis of problem lengths in the Bi-GSM8K dataset utilized in our experiments. The dataset comprises mathematical problems presented in both Korean and English, each containing 500 items. A statistical summary of the problem lengths is provided in Table 5.

The average length of English items was approximately 237 characters, significantly longer than the average length of Korean items (122.88 characters). This discrepancy likely arises from the automatic translation of Korean data into English, resulting in generally more detailed explanations or structurally longer sentences in the English items. Additionally, the standard deviation of English problems (90.91) is more than double that of Korean problems (44.50), indicating a broader

Metric	English	Korean
Count	500.000	500.000
Mean	237.004	122.876
Std	90.912	44.501
Min	62.000	36.000
25%	173.000	90.000
50%	220.000	117.000
75%	281.000	145.250
Max	592.000	304.000

Table 5: Statistical Summary of Problem Lengths

and more varied length distribution in the English dataset.

Regarding minimum lengths, English problems contain at least 62 characters, while Korean problems have a minimum of 36 characters, suggesting that English texts generally include more information. The maximum length further highlights this difference, with English problems reaching up to 592 characters compared to the 304 characters of Korean problems.

Quartile-based metrics exhibit similar trends, consistently showing higher values for English problems compared to their Korean counterparts, thereby reinforcing the structural length disparity across the entire dataset range. Notably, the median length of English problems is 220 characters, approximately 1.88 times greater than the Korean median of 117 characters.

These results might reflect the explicit and detailed sentence structures often required by the English language during translation, as well as potential adjustments made by generative models to accommodate stylistic differences between languages. Such findings underscore the necessity of considering language-specific perceptions of difficulty and interpretative approaches in subsequent analyses.

B Analysis of Solution Lengths

In this appendix, we present a quantitative statistical analysis of the lengths and steps of student-generated and correct solutions from the Bi-GSM8K dataset used in our experiments. The dataset comprises problems provided in two languages, Korean and English, each consisting of 500 items.

B.1 Analysis of Solution Length by Steps

This section analyzes the lengths of complete solutions written in Korean and English for both correct and student-generated solutions within the dataset. The first analysis considers each solution as an integrated unit without dividing it into individual steps.

Metric	Correct Solution	Student Solution
Count	500.000	500.000
Mean	278.596	181.296
Std	136.449	117.530
Min	36.000	8.000
25%	174.750	99.750
50%	249.500	169.000
75%	353.000	244.000
Max	1006.000	689.000

Table 6: Statistical Summary of Solution Lengths (English)

The statistical analysis results for English solutions are summarized in Table 6. The average length of correct (teacher-generated) solutions is approximately 278.60 characters, roughly 1.54 times longer than student-generated solutions, which average 181.30 characters. Notably, the standard deviation for correct solutions (136.45) is considerably higher compared to that for student solutions (117.53), indicating that correct solutions not only tend to be lengthier but also exhibit greater variability in structural complexity and explanatory depth.

Metric	Correct Solution	Student Solution
Count	500.000	500.000
Mean	185.456	122.008
Std	82.344	75.418
Min	46.000	5.000
25%	123.000	70.000
50%	168.000	112.500
75%	237.000	166.000
Max	631.000	389.000

Table 7: Statistical Summary of Solution Lengths (Korean)

The statistical results for Korean solutions are summarized in Table 7, displaying trends similar to those observed in the English solutions. The average length of correct (teacher-generated) solutions in Korean is approximately 185.46 characters, approximately 1.5 times longer than student-generated solutions, which average 122.01 characters. The standard deviation of correct solutions (82.34) is slightly greater than that of student solutions (75.42), suggesting more variability due to detailed explanatory content. Furthermore, the maximum length of correct solutions (631 characters) substantially surpasses that of student solutions (389 characters).

Overall, correct solutions consistently exhibit greater length compared to student solutions, reflecting their more detailed and stepwise explanatory nature. This pattern is consistent across both languages, reinforcing the observation that teacher-generated solutions generally present higher complexity and explanatory completeness.

The subsequent analysis separately examines solution lengths at the individual step level.

Metric	Correct Solution	Student Solution
Count	1724	1340
Mean	80.80	67.65
Std	34.27	32.12
Min	6	6
25%	57	48
50%	74	63
75%	99	83
Max	265	282

Table 8: Statistical Summary of Step Lengths (English)

Metric	Correct Solution	Student Solution
Count	1724	1340
Mean	53.79	45.53
Std	17.22	17.57
Min	7	5
25%	43	36
50%	51	44
75%	64	55
Max	135	157

Table 9: Statistical Summary of Step Lengths (Korean)

Tables 8 and 9 provide statistical summaries of step-by-step solution lengths in English and Korean, respectively.

In English, the average step length of correct solutions (80.80 characters) exceeds that of student solutions (67.65 characters). Additionally, the standard deviation and maximum-minimum values show a broader distribution for correct solutions, indicating that these solutions may contain more detailed and complex explanations.

Similar trends appear in Korean solutions. Correct solutions have an average step length of 53.79 characters, longer than the 45.53 characters for student solutions. Standard deviations for both groups were comparable, while maximum lengths were higher in student solutions, indicating the existence of some student solutions with extensive explanations.

Generally, correct solutions have longer and more detailed step explanations than student solutions, although exceptions exist regarding maximum-minimum lengths and range distributions.

Step	Correct Solution	Student Solution
1	82.47	64.92
2	79.81	66.95
3	79.90	71.71
4	81.15	70.64
5	83.47	70.63
6	74.74	74.47
7	73.58	71.00
8	69.75	-

Table 10: Average Step Length per Step Number (English)

Step	Correct Solution	Student Solution
1	54.20	43.04
2	53.32	45.39
3	53.84	48.37
4	54.81	48.73
5	53.26	47.54
6	50.98	49.94
7	51.50	54.00
8	53.75	-

Table 11: Average Step Length per Step Number (Korean)

Tables 10 and 11 display the average solution length by individual steps for English and Korean, respectively.

For English solutions, the average length of correct solutions consistently surpasses that of student solutions across the initial five steps, with the largest discrepancy observed in step 1 (82.47 vs. 64.92 characters). Differences decrease in subsequent steps, with steps 6–7 showing minimal divergence, and step 8 lacking student solution data, suggesting that correct solutions tend to provide longer, more detailed explanations early in the solution process.

Korean solutions reveal a similar pattern, with correct solutions typically longer than student solutions across most steps, though student solutions exceed correct solutions at step 7. Similar to English data, step 8 lacks student-generated solution data. Compared to English, differences in length per step are generally smaller in Korean solutions.

Overall, in both languages, the difference in length per step decreases as solutions progress, with correct solutions consistently providing more comprehensive explanations.

B.2 Analysis of Solution Steps

This section analyzes the number of steps in the Correct Solutions and Student Solutions within the dataset. The number of solution steps is a crucial metric indicating how granularly the solution process is articulated, serving as an essential factor for evaluating the detail and complexity of the solutions.

Metric	Correct Solution	Student Solution
Count	500	500
Mean	3.448	2.680
Std	1.349	1.309
Min	2	1
25%	2	2
50%	3	3
75%	4	3
Max	8	7

Table 12: Statistical Summary of Step Counts by Solution Type

Table 12 summarizes statistical measures for the step counts of both groups. The analysis was conducted on 500 solution samples from each group.

The Correct Solutions exhibited an average of 3.448 steps, noticeably higher than the Student Solutions, which averaged 2.680 steps. This indicates a tendency for Correct Solutions to provide more detailed and finely segmented explanations. The standard deviation of step counts was similarly around 1.3 for both groups, suggesting comparable variability in the number of solution steps.

Regarding the minimum number of steps, Student Solutions included instances starting from a single step, whereas Correct Solutions always began from at least two steps. Additionally, the 75th percentile step count was 4 for Correct Solutions and 3 for Student Solutions, reinforcing the observation that Correct Solutions generally involve more steps.

For maximum step counts, Correct Solutions extended up to 8 steps, whereas Student Solutions reached a maximum of 7 steps, indicating slightly less granularity in student-generated explanations.

These results suggest that Student Solutions tend to provide briefer explanations or omit certain steps compared to Correct Solutions. Thus, step count analysis serves as a valuable measure for assessing the completeness and structural detail of student-generated solutions.

C Detailed Examples of Ground Truth Alignment

In this appendix, we present illustrative examples of the GTA results proposed in this paper. The "Predicted Alignment" column in the table shows the alignment results from each model, with segments highlighted in red indicating misaligned sections.

Problem

Yeona has a 2L water bottle next to her desk. She takes a sip every 5 minutes, and each sip is 40ml. How many minutes does it take to finish one bottle of water?

Answer Solution

step_1: First, find the total ml of the bottle: 2L * 1000ml/L = «2*1000=2000»2000ml
step_2: Then divide the total ml by the amount consumed per sip: 2000ml / 40ml = «2000/40=50»50 sips.
step_3: Then, multiply the number of sips by the time per sip to find the time it takes to drink the bottle: 50 sips * 5 minutes/sip = «50*5=250»250 minutes.

Student Solution

step_1: Yeona's water bottle is 200ml.
step_2: Divide 200ml by the amount consumed per sip. 200ml / 40ml = «200/40=5»5 sips.
step_3: To find the time it takes to drink the bottle, multiply the number of sips by the time per sip: 5 sips * 5 minutes/sip = 25 minutes

Predicted Alignment by GPT-4o

Student Solution	Answer Solution
_	First, find the total ml of the bottle: 2L * 1000ml/L = «2*1000=2000»2000ml
Yeona's water bottle is 200ml.	_
Divide 200ml by the amount consumed per sip. 200ml / 40ml = «200/40=5»5 sips.	Then divide the total ml by the amount consumed per sip: 2000ml / 40ml = «2000/40=50»50 sips.
To find the time it takes to drink the bottle, multiply the number of sips by the time per sip: 5 sips * 5 minutes/sip = 25 minutes	Then, multiply the number of sips by the time per sip to find the time it takes to drink the bottle: 50 sips * 5 minutes/sip = «50*5=250»250 minutes.

Predicted Alignment by BERTScore + bert-large-uncased + NW algorithm

Student Solution	Answer Solution
Yeona's water bottle is 200ml.	First, find the total ml of the bottle: 2L * 1000ml/L = «2*1000=2000»2000ml
Divide 200ml by the amount consumed per sip. 200ml / 40ml = «200/40=5»5 sips.	Then divide the total ml by the amount consumed per sip: 2000ml / 40ml = «2000/40=50»50 sips.
To find the time it takes to drink the bottle, multiply the number of sips by the time per sip: 5 sips * 5 minutes/sip = 25 minutes	Then, multiply the number of sips by the time per sip to find the time it takes to drink the bottle: 50 sips * 5 minutes/sip = «50*5=250»250 minutes.

Reference Alignment

Student Solution	Answer Solution
_	First, find the total ml of the bottle: 2L * 1000ml/L = «2*1000=2000»2000ml
Yeona's water bottle is 200ml.	_
Divide 200ml by the amount consumed per sip. 200ml / 40ml = «200/40=5»5 sips.	Then divide the total ml by the amount consumed per sip: 2000ml / 40ml = «2000/40=50»50 sips.
To find the time it takes to drink the bottle, multiply the number of sips by the time per sip: 5 sips * 5 minutes/sip = 25 minutes	Then, multiply the number of sips by the time per sip to find the time it takes to drink the bottle: 50 sips * 5 minutes/sip = «50*5=250»250 minutes.

Table 13: Example Results of Ground Truth Alignment using GPT-4o and BERTScore with bert-large-uncased and NW Algorithm (English)

Problem
연아는 책상 옆에 2L짜리 물병을 두고 있습니다. 5분마다 한 모금씩 마시는데, 한 모금당 물의 양은 40ml입니다. 물 한 병을 다 마시는 데 몇 분이 걸리나요?
Answer Solution
step_1: 먼저 병의 총 ml 수를 찾습니다: 2L * 1000ml/L = «2*1000=2000»2000ml step_2: 그런 다음 총 ml 수를 한 모금당 마시는 양으로 나눕니다: 2000ml / 40ml = «2000/40=50»50 모금 step_3: 그런 다음 모금 횟수에 모금당 시간을 곱하여 병을 마시는 데 걸리는 시간을 찾습니다: 50 모금 * 5분/모금 = «50*5=250»250분
Student Solution
step_1: 연아의 물병은 200ml입니다. step_2: 200ml를 한 모금당 마시는 양으로 나눕니다200ml / 40ml = «200/40=5»5 모금 step_3: 모금 횟수에 모금당 시간을 곱하여 병을 마시는 데 걸리는 시간을 찾습니다: 5 모금 * 5분/모금 = «5*5=25»25분
Predicted Alignment by GPT-4o

Student Solution	Answer Solution
_	먼저 병의 총 ml 수를 찾습니다: 2L * 1000ml/L = «2*1000=2000»2000ml
연아의 물병은 200ml입니다.	_
200ml를 한 모금당 마시는 양으로 나눕니다200ml / 40ml = «200/40=5»5 모금	그런 다음 총 ml 수를 한 모금당 마시는 양으로 나눕니다: 2000ml / 40ml = «2000/40=50»50 모금
모금 횟수에 모금당 시간을 곱하여 병을 마시는 데 걸리는 시간을 찾습니다: 5 모금 * 5분/모금 = «5*5=25»25분	그런 다음 모금 횟수에 모금당 시간을 곱하여 병을 마시는 데 걸리는 시간을 찾습니다: 50 모금 * 5분/모금 = «50*5=250»250분

Predicted Alignment by BERTScore + bert-large-uncased + NW algorithm

Student Solution	Answer Solution
연아의 물병은 200ml입니다.	먼저 병의 총 ml 수를 찾습니다: 2L * 1000ml/L = «2*1000=2000»2000ml
200ml를 한 모금당 마시는 양으로 나눕니다200ml / 40ml = «200/40=5»5 모금	그런 다음 총 ml 수를 한 모금당 마시는 양으로 나눕니다: 2000ml / 40ml = «2000/40=50»50 모금
모금 횟수에 모금당 시간을 곱하여 병을 마시는 데 걸리는 시간을 찾습니다: 5 모금 * 5분/모금 = «5*5=25»25분	그런 다음 모금 횟수에 모금당 시간을 곱하여 병을 마시는 데 걸리는 시간을 찾습니다: 50 모금 * 5분/모금 = «50*5=250»250분

Reference Alignment

Student Solution	Answer Solution
_	먼저 병의 총 ml 수를 찾습니다: 2L * 1000ml/L = «2*1000=2000»2000ml
연아의 물병은 200ml입니다.	_
200ml를 한 모금당 마시는 양으로 나눕니다200ml / 40ml = «200/40=5»5 모금	그런 다음 총 ml 수를 한 모금당 마시는 양으로 나눕니다: 2000ml / 40ml = «2000/40=50»50 모금
모금 횟수에 모금당 시간을 곱하여 병을 마시는 데 걸리는 시간을 찾습니다: 5 모금 * 5분/모금 = «5*5=25»25분	그런 다음 모금 횟수에 모금당 시간을 곱하여 병을 마시는 데 걸리는 시간을 찾습니다: 50 모금 * 5분/모금 = «50*5=250»250분

Table 14: Example Results of Ground Truth Alignment using GPT-4o and BERTScore with bert-large-uncased and NW Algorithm (Korean)

D Detailed Examples of Solution Error Detection

In this appendix, we provide illustrative examples of SED results within the solutions proposed in this paper. In the provided tables, red-colored sections indicate incorrect responses, while blue-colored sections represent correct responses.

Problem
Three cats were sitting on a fence meowing at the moon. The first cat meowed 3 times per minute. The second cat meowed twice as often as the first cat. The third cat meowed at 1/3 the frequency of the second cat. What is the total number of times the three cats meowed in 5 minutes?
Answer Solution
step_1: The second cat meowed twice as often as the first cat, which meowed 3 times per minute, resulting in a total of 2*3=«2*3=6»6 meows per minute. step_2: The third cat meowed at 1/3 the frequency of the second cat, resulting in a total of 6/3=«6/3=2»2 meows per minute. step_3: Therefore, the three cats meow 3+6+2=«3+6+2=11»11 times per minute. step_4: In 5 minutes, three cats meow 5*11=55 times.
Student Solution
step_1: The second cat meows 2*3=«2*3=6»6 times. step_2: The third cat meows 3*1/3=«3*1/3=1»1 time. step_3: The three cats meow 3+6+1=«3+6+1=10»10 times per minute. step_4: For 5 minutes, the three cats meow 10*5=«10*5=50»50 times.
GPT-4o
step_2
Llama-3.1-8B-Instruct
step_2
DeepSeek-R1-Distill-Llama-8B
step_3
DeepSeek-llama3.1-Bllossom-8B
step_3
Qwen2.5-7B-Instruct
step_3
DeepSeek-R1-Distill-Qwen-7B
step_1
DeepSeek-R1-Distill-Qwen-7B-Multilingual
step_1
Mistral-7B-Instruct-v0.3
step_3
Phi-4-mini-instruct
step_2
Answer
step_2

Table 15: Example Results of Solution Error Detection using GPT-4o and Open LLMs (English)

Problem
고양이 세 마리가 울타리에 앉아 달을 향해 야옹거리고 있었습니다. 첫 번째 고양이는 분당 3번 야옹거렸습니다. 두 번째 고양이는 첫 번째 고양이보다 두 배 더 자주 야옹거렸습니다. 그리고 세 번째 고양이는 두 번째 고양이의 1/3 빈도로 야옹거렸습니다. 고양이 세 마리가 5분 동안 야옹거리는 총 횟수는 얼마입니까?
Answer Solution
step_1: 두 번째 고양이는 첫 번째 고양이가 분당 3번 야옹하는 것보다 두 배 더 자주 야옹하여 분당 총 2*3=«2*3=6»6번의 야옹을 했습니다. step_2: 세 번째 고양이는 두 번째 고양이의 1/3 빈도로 야옹거렸으므로 분당 총 6/3=«6/3=2»2번의 야옹거림을 보였습니다. step_3: 따라서 세 마리의 고양이는 분당 3+6+2=«3+6+2=11»11번 야옹거립니다. step_4: 5분 동안 고양이 세 마리는 5*11=«5*11=55»55번 야옹거립니다.
Student Solution
step_1: 두번째 고양이는 2*3=«2*3=6»6번 야옹거립니다. step_2: 세번째 고양이는 3*1/3=«3*1/3=1»1번 야옹거립니다. step_3: 세 고양이는 분당 3+6+1=«3+6+1=10»10번 야옹거립니다. step_4: 5분동안 세 고양이들은 10*5=«10*5=50»50번 야옹거립니다.
GPT-4o
step_2
Llama-3.1-8B-Instruct
step_2
DeepSeek-R1-Distill-Llama-8B
step_3
DeepSeek-llama3.1-Bllossom-8B
step_3
Qwen2.5-7B-Instruct
step_2
DeepSeek-R1-Distill-Qwen-7B
step_2
DeepSeek-R1-Distill-Qwen-7B-Multilingual
step_2
Mistral-7B-Instruct-v0.3
No errors
Phi-4-mini-instruct
step_2
Answer
step_2

Table 16: Example Results of Solution Error Detection using GPT-4o and Open LLMs (Korean)