Leveraging Web-Crawled Data for High-Quality Fine-Tuning

Anonymous ACL submission

Abstract

Most large language models are fine-tuned using either expensive human-annotated data or GPT-4 generated data which cannot guarantee performance in certain domains. We argue that although the web-crawled data often has formatting errors causing semantic inaccuracies, it can still serve as a valuable source for high-quality supervised fine-tuning in specific domains without relying on advanced models like GPT-4. To this end, we create a paired training dataset by aligning web-crawled data with a smaller set of high-quality data. By training a language model on this dataset, we can convert web data with irregular formats into high-quality ones. Our experiments show that training with the model-transformed data yields better results, surpassing training with only high-quality data by an average of 9.4% in Chinese elementary school math problems. Additionally, our 7B model outperforms several open-source models larger than 30B and surpasses well-known closed-source models such as GPT-3.5 and Claude-2, highlighting the efficacy of our approach.[1]

1 Introduction

Large Language Models (LLMs) have attracted much attention over the past year and high-quality data has been a crucial factor in achieving their excellent performance. Currently, two primary methodologies are employed for data acquisition. The first approach involves leveraging GPT-4 (OpenAI, 2023) or other LLMs for distillation, such as Alpaca (Taori et al., 2023), ORCA (Mukherjee et al., 2023), and WizardLM (Xu et al., 2023), to enhance the capabilities of smaller models. The second approach (Zhou et al., 2023a; Databricks, 2023; Köpf et al., 2023) use human annotation or selection to further enhance model performance, emphasizing the significance of data quality over data quantity. However, in certain domains like mathematics, even the state-of-the-art model GPT-4 fails to achieve outstanding performance (Dong et al., 2023; Mitra et al., 2024; Yuan et al., 2023), while obtaining a large volume of human-annotated data within a short timeframe is not only challenging but also costly. Conversely, web-crawled data tends to have a larger volume despite being prone to noise and formatting errors. Leveraging processed web-crawled data for training can significantly alleviate the challenges associated with data collection in specific domains.

We focus on mathematical reasoning, which requires a deep understanding of mathematical concepts and proficient reasoning abilities. Previous studies (Dong et al., 2023; Mitra et al., 2024) have demonstrated the benefits of enhancing datasets with synthetic data. Typically, these studies (Luo et al., 2023; Mitra et al., 2024) rely on the excellent performance of GPT-4 on English mathematical datasets to generate simulated data for distillation to smaller models. In contrast, we explore the potential to acquire high-quality data without depending on additional powerful LLMs such as GPT-4.

We identified two significant advantages of web-crawled data: it (1) has a large volume and (2) contains most of the necessary information to solve specific problems, despite its poor formatting. Drawing on the intuition that rewriting data is comparatively simpler for the LLM than performing intricate reasoning tasks, we propose a method that augments the dataset by converting web-crawled data into high-quality ones. Our approach begins by automatically aligning low-quality web-crawled data with high-quality seed data to generate <low-quality, high-quality> data pairs. We subsequently utilize these pairs to fine-tune an LLM, developing a model specifically designed to transform low-quality web-crawled data into high-quality data. Our experiments demonstrate that this approach significantly improves data

[1]We have released our code in the attachment.

quality and boosts model performance, surpassing traditional rule-based methods. The key contributions of our work are as follows:

1. We propose a simple and effective method for transforming web-crawled data into high-quality data without relying on additional LLMs like GPT-4.
2. Our approach improves the performance of two representative open-source models, with an average improvement of 9.4%.
3. We revealed that formatting errors could lead to semantic inaccuracies and analyzed the reasons behind the effectiveness of our method.

2 Related Work

2.1 Large Language Models for Mathematical Reasoning

Complex reasoning has become a critical capability for LLMs, and a series of benchmarks have been developed to assess this ability using mathematical word problems. Notable English benchmarks include GSM8K (Cobbe et al., 2021) and SVAMP (Patel et al., 2021), while Ape210K (Zhao et al., 2020) and CMATH (Wei et al., 2023) are prominent benchmarks in the Chinese language.

"Chain of Thought" (CoT) (Wei et al., 2022; Zhou et al., 2023b; Kojima et al., 2022; Fu et al., 2023) enhances the model's reasoning capability by predicting the step-by-step reasoning process before arriving at the answer. Wang et al. (2023) further enhances the model's performance using majority voting techniques. Additionally, the "Tree of Thoughts" (ToT) (Yao et al., 2023) approach explores reasoning paths through self-evaluation by the LLM to facilitate global decision-making. Moreover, equipping the model with tools such as calculators (Cobbe et al., 2021) or programs (Gao et al., 2023a; Chen et al., 2022; Imani et al., 2023; Yue et al., 2023) can also contribute to improved problem-solving abilities. In our paper, we concentrate on improving the data quality for CoT, because it forms the foundation of the model's reasoning capability.

2.2 Is GPT4 Generated Data Enough?

Utilizing synthetic data generated by strong LLMs (Taori et al., 2023; Mukherjee et al., 2023; Gunasekar et al., 2023; Wang et al., 2024) for training has proven effective in enhancing model performance. In mathematics, studies (Luo et al., 2023; Mitra et al., 2024; Yuan et al., 2023; Yu et al., 2023) emphasize that utilizing a powerful LLM (GPT3.5/GPT4) to generate diverse and challenging datasets can significantly improve model performance.

However, the data generated by LLMs has inherent limitations. Although models have a certain degree of fault tolerance (Yu et al., 2023), relying solely on synthetic data generated by strong LLMs can limit the upper bound. For instance, in domains where the best LLM performs poorly, the quality of generated data may not be guaranteed. Our method leverages the knowledge in web-crawled data, along with the powerful information extraction and format standardization capabilities of LLMs, thereby obviating the need for additional LLMs to generate answers from scratch.

3 Methods

3.1 Settings

Training Data Sets. We acquired a meticulously annotated dataset from an educational institution, along with a web-crawled collection of mathematical problems. Due to their distinct origins, these two datasets are not independently and identically distributed (i.i.d.). This web-crawled dataset has been filtered with rules, thus almost all the data used in this paper are mathematical problems with detailed solving processes. The manual-annotated seed dataset consists of 84,095 instances, while the web-crawled dataset comprises 573,960 instances.

3.2 A Close Look on Web-Crawled Data

Misleading Caused by Formatting Issues. Although our preprocessing efforts have enhanced the quality of the web-crawled data, there still remain numerous format errors and non-standard formatting issues. An example is shown in Figure 1, the expression $3.14 \times 6^2 = 3.14 \times 36$ is represented as $3.14 \times 62 = 3.14 \times 36$ in the crawled data, which is mathematically incorrect. Due to the extensive combinatorial nature of mathematical formulas, these errors can result in expressions that *appear to be intact in terms of formatting but completely misrepresent the underlying physical meaning*. Consequently, training with these errors can mislead the model, particularly in complex scenarios. We summarize the most significant errors of web-crawled data in Table 1 and show corresponding examples in Table 8.

It is quite difficult to correct those errors using rule-based methods, which we will explain in Sec-

Error Type	Detailed Description
Fraction Format Errors	The fraction are not in latex format. $\frac{x}{y}$ may be in the form of "x\ny" or "xy".
Super/Subscripts Errors	The positional information of special characters such as superscripts and subscripts may be lost.
Missing Line Breaks	Occasionally, the line breaks ("\n") between different lines are missing.
Non-standard formula	Some symbols are displayed in non-standard form, such as "×" being typed as "X".
Garbled Characters	Severe formatting disruptions were observed in a tiny subset of samples due to OCR.

Table 1: Typical error types in web-crawled data. The fraction format errors and superscripts/subscripts errors are the most common in our data.

Data with Errors

Question: The radius of a small circle is 2 cm, and the radius of a large circle is times that of the small circle. What is the area of the large circle?

Answer:

The radius of the large circle: 2 × 3=6 (cm)

The area of the large circle: 3.14 × 62 = 3.14 × 36 = 113.04 (cm2) ❌

Correct Format

Question: The radius of a small circle is 2 cm, and the radius of a large circle is **3** times that of the small circle. What is the area of the large circle?

Answer:

The radius of the large circle: 2 × 3=6 (cm)

The area of the large circle: 3.14 × $\mathbf{6^2}$ = 3.14 × 36 = 113.04 (**cm^2**) ✅

Figure 1: An example of a web-crawled sample with "local errors" and "global errors". The "local errors" are denoted in blue, and the "global errors" are denoted in red.

tion 3.2. Utilizing these flawed samples for training may not only introduce inconsistent output formats but also affect the model's understanding of mathematical concepts. However, if we discard samples with errors entirely, it would significantly reduce the information content in the training data, thereby affecting the model's performance.

The Drawbacks of Rule-Based Methods In data preprocessing, rule-based methods often hold significant importance. However, it is important to note that while certain errors can be resolved using rule-based methods, others may not be amenable to such approaches in principle. To state it more clearly, we define two distinct types of errors: local errors and global errors.

- **Local errors** refer to errors that can be corrected by examining a few consecutive words.
- **Global errors** refer to errors that can only be rectified if the method comprehends the entirety of the example, including both the question and the answer.

The primary limitation of rule-based methods is that they can only solve "local errors" but not "global errors". Figure 1 illustrates an example, with the "local errors" highlighted in blue and the "global errors" marked in red. In this instance, the crucial information of "3 times" is missing from the question, making it impossible to fill in the blank without consulting the answer. Additionally, determining whether "62" represents "6^2" or simply "62" poses a challenge for rule-based approaches, as both interpretations are prevalent in the corpus. Consequently, these two instances are classified as global errors. Conversely, the third scenario involving "cm2" commonly denotes "cm^2" in most cases, making it a "local error" that can be easily addressed using rules. Another drawback of rule-based methods is the requirement to analyze numerous cases and handle various boundary situations when constructing rules. This process is not only highly challenging but also significantly increases people's workload.

Feasibility of Model-based Methods After careful examination of the web-crawled samples, we believe that despite the presence of numerous formatting issues in the crawled data, the data itself still contains a substantial amount of valuable information. We arrived at the following findings:

1. Despite the vast array of mathematical problem types, the types of formatting errors tend to be relatively uniform. Consequently, by fine-tuning a model, it should be capable of learning the correct paradigms efficiently with a limited number of samples.
2. Compared to performing complex reasoning tasks, it is easier for the LLM to rewrite the data. In other words, modifying the format of questions and answers to obtain training data is much simpler than generating answers for questions from scratch.
3. Different from rule-based methods focusing on local considerations, LLMs are good at combining all the information in the sample.

Therefore, we recommend utilizing the informa-

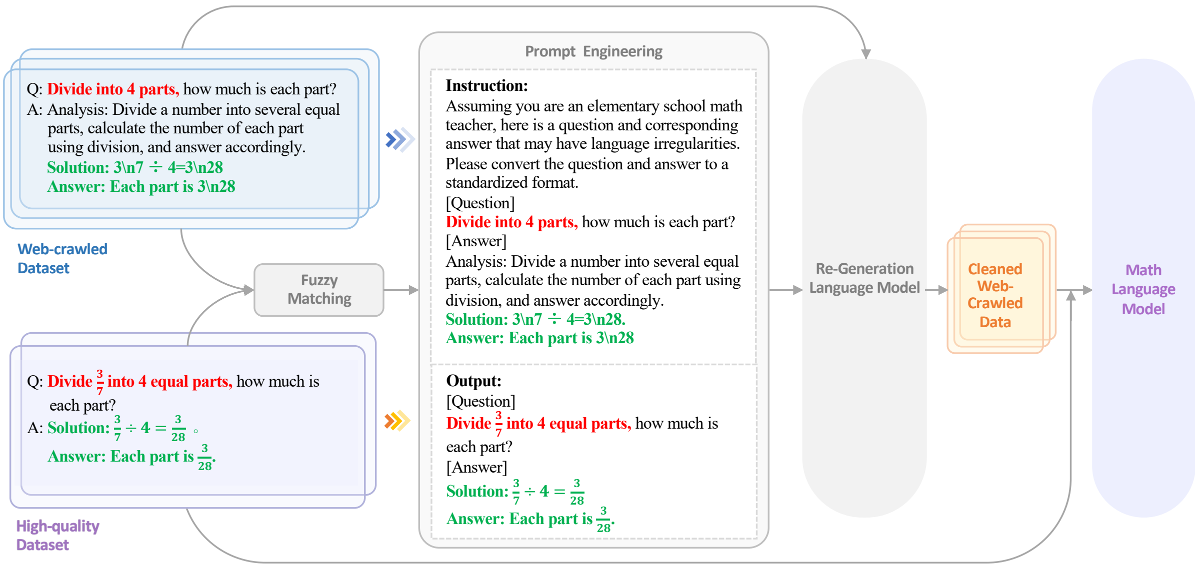

Figure 2: An illustration of our proposed data transforming architecture. The answer coloured in green is matched, resulting in a <web-crawled, high-quality> data pair. The text in red is originally wrong and needs to be corrected. We then prompt the paired data to train a re-generation language model to convert the web-crawled data into high-quality ones. Finally, we train a Math LLM using both the high-quality data and the cleaned web-crawled data.

tion in the web-crawled data and leveraging the excellent understanding and languaging capabilities of neural networks to construct high-quality training data. This is related to the core idea of Retrieval-Augmented Generation (RAG), which we will discuss later in Section 5.

3.3 A Simple and Effective Method for Data Cleaning

Based on the analysis above, we propose a simple and effective method to enhance the quality of web-crawled data. This approach leverages the linguistic capabilities of LLMs alongside the inherent knowledge within web-crawled data to refine and standardize its format, thereby effectively reducing the occurrence of erroneous expressions.

Our method involves the following four steps as shown in Figure 2:

1. Constructing format converter training data by pairing web-crawled data with high-quality data using fuzzy matching.
2. Train an LLM with the constructed data to enable it to transform raw web-crawled examples into high-quality examples.
3. Use the trained LLM to convert the web-crawled data into high-quality format.
4. Train another LLM to solve mathematical problems using both the high-quality data and the converted web-crawled data.

Formally, given a high-quality problem set $\mathrm{D}_{\text{high}} = \{(q_i, a_i)\}_i$ where q_i is a math question and a_i is the corresponding answer, along with a large web-crawled dataset $\mathrm{D}_{\text{crawl}} = \{(q_j, a_j)\}_j$, we can derive a matched dataset in the following manner:

$$\mathrm{D}_{\text{train}} = \{([q,a],[q',a'])|(q,a) \in \mathrm{D}_{\text{high}}, \\ (q',a') \in \mathrm{D}_{\text{crawl}}, \mathrm{match}(q,q') \vee \mathrm{match}(a,a')\}.$$

Typically, the size of the matched dataset $\mathrm{D}_{\text{train}}$ is smaller than that of the high-quality dataset and web-crawled dataset, i.e., $|\mathrm{D}_{\text{train}}| < \min(|\mathrm{D}_{\text{high}}|, |\mathrm{D}_{\text{crawl}}|)$. [2] Subsequently, we fine-tune an LLM g using the constructed dataset $\mathrm{D}_{\text{train}}$ and use this model to process the web-crawled data. For each sample $[q,a]$, the model generates an output in a predefined concatenated format "formatted($[a',q']$)". Afterwards, we apply rules to extract the question and answer from the output, resulting in the final mathematical problem-solving training dataset $\mathrm{D}_{\text{cleaned}} = \{q'_i, a'_i\}_i$. Samples that do not conform to the predefined output format are discarded. Finally, we fine-tune an LLM on both the high-quality data D_{high} and the cleaned data $\mathrm{D}_{\text{cleaned}}$ to improve the model performance in mathematical reasoning.

[2] We have further augmented our dataset with samples containing severe formatting errors, prompting the model to recognize these instances and output a "syntax error" indication. The relative number of those dropped examples is small, and we have verified that the dropped examples are not the main reason for our improvement in effectiveness.

	ChatGLM2-6B		Qwen1.5-7B-Chat	
	Ape210K	CMATH	Ape210K	CMATH
W.o. Training	38.7	62.8	55.4	72.5
SFT w. D_{high}	55.6	76.2	68.2	81.8
PT w. D_{crawl} + SFT w. D_{high}	59.4	77.2	69.0	83.2
SFT w. $D_{cleaned}$	72.1	84.5	**74.2**	**87.3**
SFT w. $D_{cleaned}$ + D_{high}	**73.9**	**84.8**	74.1	86.5

Table 2: Performance comparison among different language models on the Ape210K and CMATH. "SFT w. D_{high}" denotes fine-tuning with human-annotated high-quality data only. "PT w. D_{crawl} + SFT w. D_{high}" denotes first post-training the model with web-crawled data and then fine-tuning the model with high-quality data. "SFT w. $D_{cleaned}$ + D_{high}" denotes fine-tuning the model with model-converted and human-annotated data together. Best results are denoted in **Black**.

4 Experiments

4.1 Experimental Setup

4.1.1 Test Datasets and Evaluation Method

Because all our training data are about Chinese elementary school math, following ChatGLM-Math (Xu et al., 2024), we evaluate our performance on two Chinese math datasets, Ape210K (Zhao et al., 2020) and CMATH (Wei et al., 2023). Different from the works that utilize LLM as the verifier (Zheng et al., 2023; Xu et al., 2024), we wrote an automatic evaluation script in Python. Our auto-evaluation script exhibits an evaluation accuracy of 95% on Ape210K. For CMATH, we utilize the evaluation script [3] provided in the paper.

4.1.2 Models

We experiment on two most widely used Chinese open-source models, i.e., ChatGLM (Du et al., 2022; Zeng et al., 2023) and Qwen (Bai et al., 2023), specifically, ChatGLM2-6B and Qwen1.5-7B-Chat. We employ fully parameterized supervised fine-tuning (SFT) in all our experiments. Due to time constraints, we did not conduct hyperparameter searches; instead, all experiments were performed once using a pre-determined, stable hyperparameter set. During the training process, we employed a batch size of 128 for both models, a learning rate of 5e-5 for ChatGLM, and a learning rate of 5e-6 for Qwen. We do not use early stopping, but instead train all data for three epochs.

4.2 Main Results

Our results are shown in Table 2. The conventional approach of post-training with noisy, web-crawled data only marginally improves model performance by an average of 1.8%. In contrast, fine-tuning the model with both high-quality and our cleaned data significantly enhances performance by an average of 9.4%, demonstrating the effectiveness of our method. This improvement can be attributed to the following reasons: (1) The presence of errors in the web-crawled data hinders the learning of mathematical reasoning. (2) The format disparity between web-crawled (pure text) and high-quality data (LaTeX) makes it challenging to integrate both paradigms of information. (3) Two-stage fine-tuning makes it more susceptible for the model to forget the data used in the post-training stage, indicating that SFT exhibits superior data efficiency compared to post-training.

An intriguing observation that deviates from common sense is the comparable performance of SFT with $D_{cleaned}$ to that of SFT with both $D_{cleaned}$ and D_{high}. We propose two potential explanations for this phenomenon. Firstly, the cleaned data is generated by the model, which results in the distillation of certain information from the high-quality data into the cleaned web-crawled data during the cleaning process. Secondly, our high-quality dataset encompasses various types of mathematical problems, not limited to just mathematical word problems, which could potentially influence the distribution of the data.

Although we focus on improving data utilization rather than brushing rankings, we still achieved outstanding performance on small models within 10B. Comparison between different representative models is in Table 3. Our performance with the 7B model surpasses several models larger than 30B, including Yi-Chat (Young et al., 2024), DeepSeek-Chat (Bi et al., 2024), and ChatGLM3. Additionally, our results exceed some well-known closed-

[3] https://github.com/XiaoMi/cmath

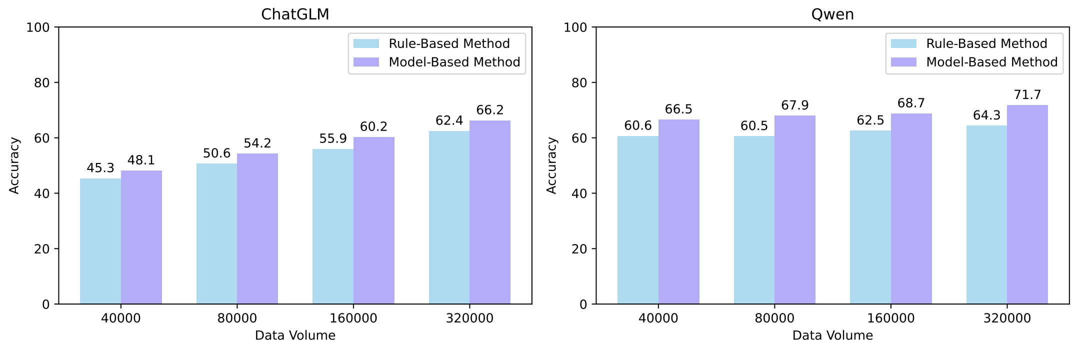

Figure 3: Comparison between rule-based and model-based method on Ape210K, as training data grows. The figure left is the results on ChatGLM and the figure right is the results on Qwen. The horizontal axis represents the amount of SFT data, and the vertical axis represents the accuracy on Ape210K.

source models like GPT-3.5 (OpenAI, 2023) and Claude-2 (Anthropic, 2023).

	#params	Ape210K	CMATH	Avg.
GPT-4-1106-Preview†	N/A	84.2	89.3	86.8
GPT-4-0613†	N/A	83.6	86.5	85.1
GPT-3.5-Turbo-0613†	N/A	70.4	76.8	73.6
Claude-2†	N/A	72.8	80.5	76.7
GLM-4†	N/A	93.5	89.0	91.3
Yi-Chat†	34B	65.1	77.7	71.4
DeepSeek-Chat†	67B	76.7	80.3	78.5
Qwen-Chat†	72B	77.1	88.1	82.6
ChatGLM3-SFT†	32B	78.0	79.8	79.8
Ours (ChatGLM2)	6B	73.9	84.8	79.4
Ours (Qwen1.5)	7B	74.2	87.3	80.8

Table 3: Performance comparison among different language models on the Ape210K and CMATH. Results denoted by † are reported by Xu et al. (2024). "#params" denotes the number of parameters, and "Avg." denotes the average performance.

4.3 More Analysis of the Effectiveness

To better compare the effectiveness of the traditional process pipeline (rule-based) and our model-based method, we eliminated the influence of high-quality data by solely employing the cleaned web-crawled data for SFT. We develop a refined rule-based data cleaning strategy to transform the web-crawled data into the SFT format. Details are in Appendix A.3, and results are in Figure 3.

Model-Based vs. Rule-Based From Figure 3, we can see that under various models and different data volumes, the model-based cleaning method consistently outperforms the rule-based one. Specifically, with ChatGLM2, the model-based method demonstrates an average improvement of 3.6% over the rule-based method, whereas with Qwen, the gap widens to an average improvement of 6.7%. This leads us to conclude that a better base model benefits more from our model-based re-generation strategy.

The Influence of the Number of SFT Data We conducted an investigation into the impact of increasing data volume on model performance. Remarkably, we observed a linearly increasing trend in the model's effectiveness as the data doubled, suggesting a log-linear relationship. This finding aligns with previous research (Yuan et al., 2023; Dong et al., 2023). On ChatGLM, there is an approximate 5% improvement in performance for every doubling of data volume. However, in the case of Qwen, doubling the data volume only leads to a 2% improvement. This discrepancy may be attributed to the nature of the data encountered during the pre-training phase. Specifically, the more limited exposure to mathematical-related data during pre-training, the more notable the performance gains with increased data volume.

4.4 Impact on Questions Across Grades

We further explore the impact of the cleaning method on questions across different grade levels. Typically, as students progress through higher grades, the knowledge required becomes more complex and often necessitates more intricate thinking processes. We classify and analyze the samples directly based on the grade labels provided in the CMATH dataset. Results are in Table 4.

Compared with the rule-based method, we can see that the model-based re-generation strategy can improve the performance of questions across different grades, with the greatest improvement observed for the fifth- or sixth-grade questions. The notable

Model	G1	G2	G3	G4	G5	G6
Rule-ChatGLM	92	87	84	82	60	71
Model-ChatGLM	94 (+2)	94 (+7)	90 (+6)	84 (+2)	75 (+15)	70 (-1)
Rule-Qwen	92	89	92	85	72	68
Model-Qwen	94 (+2)	93 (+4)	92 (+0)	86 (+1)	80 (+8)	79 (+11)

Table 4: Performance on different grades. G1, G2, ..., and G6 respectively represent grades 1 to 6. "Rule" denotes the rule-based data cleaning strategy, and "Model" denotes our model-based data cleaning strategy.

improvement is primarily attributed to the mitigation of global errors. The significant improvement observed in the fifth-grade or sixth-grade questions could be attributed to their higher complexity and greater demand for data accuracy. Additionally, these questions predominantly assess concepts related to fractions or geometry, which have a higher probability of errors in the original data.

4.5 Robustness w.r.t. the Number of High-Quality Data

In our experiments, we utilized a corpus of high-quality seed data consisting of 84,095 instances. This extensive dataset subsequently yielded 24,336 paired instances for training the generator, indicating that approximately 28.9% of the high-quality data could be successfully paired. However, it might not be possible for others to collect such a large number of high-quality data. Therefore, we conduct experiments to explore the relationship between the performance with the number of high-quality data (paired data).

Dataset	Rule	M-10k	M-20k	M-40k	M-All
Ape210K-C	50.6	52.6	53.2	53.8	54.2
CMATH-C	69.3	72.8	75.0	74.5	74.3
Ape210K-Q	60.5	66.1	67.9	67.8	67.9
CMATH-Q	79.2	82.5	82.7	82.8	82.0

Table 5: Performance w.r.t. different amounts of high-quality data. "10k", "20k", "40k", "All" respectively represent the number of high-quality seed data. "Rule" denotes the rule-based data cleaning strategy, and "M" denotes our model-based data cleaning strategy. "C" denotes ChatGLM and "Q" denotes Qwen.

We conducted experiments by varying the number of high-quality data and comparing the performance of both rule-based method and model-based methods. Owing to time constraints, our SFT experiments were conducted on a subset of 80,000 samples. The results are summarized in Table 5. Notably, even with a limited set of 10,000 high-quality data instances (yielding 2,990 pairs), our method significantly outperforms the rule-based approach. This demonstrates the robustness and practicality of our method in real-world scenarios.

4.6 The Quality of Data Cleaning

We evaluated the quality of 100 random data entries, and the corresponding results are presented in Table 6. It can be observed that the rule-based rewriting method surpasses the baseline by 5 points, while ChatGLM surpasses it by 13 points, and Qwen surpasses it by 18 points. These results demonstrate the effectiveness of our method. However, none of these methods achieves an accuracy above 90% due to the limitations in the mathematical capabilities of LLMs, making it challenging to handle complex scenarios. One potential direction is to conduct additional validation on the generated samples and facilitate the synchronous improvement of both model quality and data quality through techniques such as self-training.

Origin	Rule	Model-GLM	Model-Qwen
71%	76%	84%	89%

Table 6: The data quality under different methods. We sampled and assessed the quality of 100 data entries. "Rule" denotes the rule-based method. "Model-GLM" denotes the accuracy of the samples generated by ChatGLM2-6B. "Model-Qwen" denotes the accuracy of the samples generated by Qwen1.5-7B-Chat.

Table 7 presents a converted case using Qwen1.5-7B-Chat. In this case, the model (1) accurately identifies and converts fraction errors in the sentence into a LaTeX format, and (2) fills in missing numbers in the question by comprehending the context, which cannot be realized through rule-based methods. Additional cases can be found in Appendix A.4. From these cases, it can be concluded that our method significantly improves data quality in various error types.

	Original (Chinese)	Translated (English)
Web-Crawled	**Q:** 光明养鸡场今年养鸡2400只，比去年增加，去年养鸡多少只? **A:** 试题分析：把去年养鸡的只数看作单位“1”，求单位“1”的量，用除法计算，数量2400除以对应的分率（1+ 1 5）. 试题解析：去年养鸡的只数：2400÷（1+1 5），=2400÷6 5，=2400×5 6，=2000（只）．答：去年养鸡2000只.	**Q:** Guangming Chicken Farm raised 2400 chickens this year, an increase from last year. How many chickens did it raise last year? **A:** Analysis: Consider the number of chickens raised last year as unit “1”, and calculate the quantity of unit “1” using division. Divide the quantity 2400 by the corresponding fraction(1+ 1 5). Solution: Number of chickens raised last year: 2400÷(1+1 5), =2400÷6 5, =2400×5 6, =2000 (chickens). Answer: There were 2000 chickens raised last year.
Model-Cleaned	**Q:** 光明养鸡场今年养鸡2400只，比去年增加$\frac{1}{5}$，去年养鸡多少只? **A:** 解：2400÷（1+$\frac{1}{5}$） =2400÷$\frac{6}{5}$ =2000（只） 答：去年养鸡2000只.	**Q:** Guangming Chicken Farm raised 2400 chickens this year, an increase of $\frac{1}{5}$ from last year. How many chickens did it raise last year? **A:** Solution: 2400÷(1+$\frac{1}{5}$) =2400÷$\frac{6}{5}$ =2000 (chickens) Answer: There were 2000 chickens raised last year.

Table 7: Case of our model transformed examples. Our data are all Chinese elementary school math problems. For ease of understanding, we have provided an English translation on the right.

5 Discussions

Relationship with RAG The widely discussed RAG (Gao et al., 2023b; Komeili et al., 2022; Thoppilan et al., 2022; Schick et al., 2023) technology is conducted during the inference period. Providing references to the model and allowing the model to refer to these references in generating answers, helps the model reduce “hallucinations”, especially for knowledge-intensive tasks. Our method can be seen as RAG during the training process. Distilling the model’s unknown knowledge into the training data can further enhance the model’s capabilities. The injection of knowledge can also positively impact the model’s generalization in related domains.

Possible Applications in Other Domains A core idea of our paper is that: the effective use of appropriate data formats and instructions, derived from pretraining datasets, can facilitate the efficient SFT. Therefore, our method can be extended to various scenarios. Numerous open-source high-quality datasets can be used to create paired data through alignment with web-crawled resources. For instance, by aggregating relevant Wikipedia entries for specific QA datasets, one can train a model to generate pertinent questions and answers corresponding to those entries. Furthermore, in niche scenarios featuring unique personal corpora, it is feasible to initiate training with a small amount of seed data to produce high-quality SFT data, thereby integrating this knowledge into the model.

Future Directions Our training data for the transforming method is automatically constructed using fuzzy matching, which presents both benefits and challenges. While this approach enables the generator to produce correct answers even when the original answers are incorrect, it can also lead to errors in instances where the original answers are accurate. In such cases, employing additional verifiers could be helpful. Furthermore, implementing self-training methods may be valuable to concurrently improve the model’s mathematical capabilities and the quality of the transformed data.

6 Conclusion

We observed that in mathematical problems, format errors in the web-crawled data not only cause confusion in the output format but also result in semantic inaccuracies. Building on this insight, we propose a simple and efficient method that leverages the abundant information in web-crawled data and the strong understanding capabilities of LLMs. Our method enables the transformation of web-crawled data into high-quality ones without additional language models such as GPT-4. Experiments demonstrate the superiority of our method. In the future, it is worth exploring how to extend this method to enhance data quality in various other scenarios.

7 Limitations

Although our method greatly improved the model performance without relying on specific annotation or additional LLMs, for some special scenarios when it's difficult to construct suitable pairs, a certain amount of annotation is still needed as a cold start. Moreover, the cleaning process could introduce new errors in the data, thus additional methods that could enhance the data quality are still a problem worth exploring.

References

Anthropic. 2023. Introducing claude. https://www.anthropic.com/news/introducing-claude.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. arXiv preprint arXiv:2309.16609.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, Alex X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. 2024. Deepseek LLM: scaling open-source language models with longtermism. CoRR, abs/2401.02954.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. CoRR, abs/2110.14168.

Databricks. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2023. How abilities in large language models are affected by supervised fine-tuning data composition. CoRR, abs/2310.05492.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 320–335.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. In ICLR. OpenReview.net.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023a. PAL: program-aided language models. In ICML, volume 202 of Proceedings of Machine Learning Research, pages 10764–10799. PMLR.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023b. Retrieval-augmented generation for large language models: A survey. CoRR, abs/2312.10997.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need. CoRR, abs/2306.11644.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. In ACL (industry), pages 37–42. Association for Computational Linguistics.

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

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In ACL (1), pages 8460–8478. Association for Computational Linguistics.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations - democratizing large language model alignment. In NeurIPS.

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

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. 2024. Orca-math: Unlocking the potential of slms in grade school math. CoRR, abs/2402.14830.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of GPT-4. CoRR, abs/2306.02707.

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

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2080–2094, Online. Association for Computational Linguistics.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In NeurIPS.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Agüera y Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications. CoRR, abs/2201.08239.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. CoRR, abs/2401.00368.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In ICLR. OpenReview.net.

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

Tianwen Wei, Jian Luan, Wei Liu, Shuang Dong, and Bin Wang. 2023. CMATH: can your language model pass chinese elementary school math test? CoRR, abs/2306.16636.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. CoRR, abs/2304.12244.

Yifan Xu, Xiao Liu, Xinghan Liu, Zhenyu Hou, Yueyan Li, Xiaohan Zhang, Zihan Wang, Aohan Zeng, Zhengxiao Du, Wenyi Zhao, Jie Tang, and Yuxiao Dong. 2024. Chatglm-math: Improving math problem-solving in large language models with a self-critique pipeline. CoRR, abs/2404.02893.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In NeurIPS.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai. CoRR, abs/2403.04652.

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

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. CoRR, abs/2308.01825.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. CoRR, abs/2309.05653.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130b: An open bilingual pre-trained model. In The Eleventh International Conference on Learning Representations (ICLR).

Wei Zhao, Mingyue Shang, Yang Liu, Liang Wang, and Jingming Liu. 2020. Ape210k: A large-scale and template-rich dataset of math word problems. CoRR, abs/2009.11506.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In NeurIPS.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. LIMA: less is more for alignment. In NeurIPS.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023b. Least-to-most prompting enables complex reasoning in large language models. In ICLR. OpenReview.net.

A Appendix

A.1 Datasets

The web-crawled data often appear in rich text format (a mixture of texts and images). We apply Optical Character Recognition (OCR) to extract text from images on the webpage and then we employ rules to further discard low-quality samples, obtaining a portion of relatively high-quality samples. Although these samples already have relatively high quality, there are still many format errors and cases of non-standard formatting, which are difficult to process using rules. Ultimately, we obtain 84,095 high-quality seed data and 573,960 web-crawled data.

A.2 Format Error Examples of Web-Crawled Data

Examples of typical format errors are shown in Table 8, including fraction format errors, superscripts/subscripts errors, missing line errors and other non-standard formats.

A.3 Rule-based Methods

It should be noted that the web-crawled data we mentioned in the article has already been filtered through specific rules, yet numerous errors persist. We revised the data using rule-based methods as described in Section 4.3, applying the following rules.

1. Develop a series of templates to extract only the corresponding detailed answer parts as answers to the questions.
2. Correct fraction related errors, such as replacing "NUM1\nNUM2" with "NUM1/NUM2".
3. Correct equation related non-standardize expressions, such as replacing ",=" with "=" and replaceing ",≈" with "≈".

However, many format errors, while simple for humans, prove challenging for traditional rule-based systems. Firstly, it is impossible to enumerate all the rules comprehensively. Secondly, some global errors can not be fixed using rule-based methods. Crucially, cleaning one format might introduce errors in another. For instance, in the rule replacing NUM1\nNUM2 with NUM1/NUM2, where NUM1 and NUM2 are digits and "\n" denotes a line break, an accurate replacement is difficult without affecting other data. A case is shown in Table 9. However, neural networks can address this issue more effectively.

A.4 Case Study

In addition to the examples presented in the main text, we show two additional model-transformed cases with Qwen1.5-7B-Chat in Table 10. In the first case, the superscript is erroneously formatted as "2n+1" instead of "$2^n + 1$". Our model succeeds in detecting and correcting it. In the second case, the missing line break between two equations results in confusion and misinterpretation. By inserting appropriate line breaks, our model transforms the text into a more readable format. In both cases, our model accurately extracts the crucial elements of the sample instead of merely copying the entire analysis.

Error Type	Original Web-Crawled Data (Chinese)	Translated Data (English)
Fraction Format Errors	**Q:** 光明养鸡场今年养鸡2400只，比去年增加，去年养鸡多少只? **A:** 试题解析：2400÷（1+1 5），=2400÷6 5，=2400×5 6，=2000（只）．答：去年养鸡2000只．	**Q:** Guangming Chicken Farm raised 2400 chickens this year, an increase from last year. How many chickens did it raise last year? **A:** Solution: 2400÷(1+1 5),=2400÷6 5,=2400×5 6,=2000 (chickens).Answer: There were 2000 chickens raised last year.
Super/ Subscripts Errors	**Q:** 将一根绳子对折一次后从中间剪一刀,绳子变成3段;对折两次后从中间剪一刀,绳子变成5段:将这根绳子对折n次后从中间剪一刀,绳子变成___段. **A:** 根据分析可得:将一根绳子对折1次从中间一刀,绳子变成3段;有21+1=3.将一根绳子对折2次,从中间一刀,绳子变成5段;有22+1=5.依此类推,将这根绳子对折n次后从中间剪一刀,绳子变成(2n+1)段.	**Q:** After folding a rope in half once and cutting it in the middle, the rope becomes 3 segments. After folding it twice and cutting it in the middle, the rope becomes 5 segments. If we fold the rope n times and cut it in the middle, the rope will become ___ segments. **A:** According to the analysis, folding a rope once and cutting it in the middle results in 3 segments, which can be represented as 21+1=3. Folding the rope twice and cutting it in the middle results in 5 segments, represented as 22+1=5. Following this pattern, if we fold the rope n times and cut it in the middle, the rope will be divided into (2n+1) segments.
Missing Line Breaks	**Q:** 一辆汽车为灾区运送救灾物资，原计划每小时行驶60千米，12小时到达目的地。由于气候原因，实际每小时比计划少行驶10千米。这辆汽车实际用多少小时到达灾区?(用比例解) **A:** 解：设这辆汽车实际用x小时到达灾区，（$60-10$）$\times x = 60 \times 1250x = 60 \times 1250x = 72050x \div 50 = 720 \div 50x = 14.4$答：这辆汽车实际用14.4小时到达灾区.	**Q:** A car is transporting disaster relief supplies to a disaster area. The original plan was to travel 60 kilometers per hour and reach the destination in 12 hours. Due to weather conditions, the actual travel distance per hour is 10 kilometers less than planned. How many hours will it take for the car to reach the disaster area in reality? (Solve using proportions) **A:** Solution: Assuming that this car actually arrived at the disaster area in x hours,$(60-10)\times x = 60 \times 1250x = 60 \times 1250x = 72050x \div 50 = 720 \div 50x = 14.4$Answer: This car actually took 14.4 hours to reach the disaster area
Non-standard Formula	**Q:** 鸡兔同笼,共有11个头,有26条腿,鸡和兔各有多少只? **A:** 设鸡有x只，兔有y只 x十y=20, (1) 2x十4y=46 (2) 将(1)X2, 得 2x十2y=40 (3) (2) - (1), 得 2y=6 y=3 所以x=20-3=17 答: 鸡有17只，兔有3只。	**Q:** Chickens and rabbits are in the same cage, there are a total of 11 heads and 26 legs. How many chickens and rabbits are there respectively? **A:** Let's say there are x chickens and y rabbits. x十y=20, (1) 2x十4y=46 (2) (1)X2, we get 2x十2y=40 (3) (2) - (1), we get 2y=6 y=3 Therefore, x=20-3=17 Answer: There are 17 chickens and 3 rabbits.

Table 8: Typical error types and their corresponding instances. Our data are all Chinese elementary school math problems. For ease of understanding, we have provided an English translation highlighted in blue.

ID	Web-Crawled Examples	Rule Converted Examples
Case 1	**Q:** 光明养鸡场今年养鸡2400只，比去年增加，去年养鸡多少只? **A:** 试题分析：把去年养鸡的只数看作单位“1”，求单位“1”的量，用除法计算，数量2400除以对应的分率（1+ 1 5）． 试题解析：去年养鸡的只数：2400÷（1+1 5），=2400÷6 5，=2400×5 6，=2000（只）．答：去年养鸡2000只． **Q:** Guangming Chicken Farm raised 2400 chickens this year, an increase from last year. How many chickens did it raise last year? **A:** Analysis: Consider the number of chickens raised last year as unit “1”, and calculate the quantity of unit “1” using division. Divide the quantity 2400 by the corresponding fraction(1+ 1 5). Solution: Number of chickens raised last year: 2400÷(1+1 5), =2400÷6 5, =2400×5 6, =2000 (chickens). Therefore, there were 2000 chickens raised last year.	**Q:** 光明养鸡场今年养鸡2400只，比去年增加，去年养鸡多少只? **A:** 去年养鸡的只数：2400÷（1+1/5）=2400÷6/5=2400×5/6=2000（只）．答：去年养鸡2000只． **Q:** Guangming Chicken Farm raised 2400 chickens this year, an increase from last year. How many chickens did it raise last year? **A:** Solution: Number of chickens raised last year: 2400÷(1+1/5)=2400÷6/5=2400×5/6=2000 (chickens). Therefore, there were 2000 chickens raised last year.
Case 2	**Q:** 工人把10.5立方米的黄沙铺在一个长6米，宽3.5米的长方体沙坑里，可以铺多厚？（用方程解） **A:** 设可以铺x米， 6×3.5×x=10.5 21x=10.5 x=10.5÷21 x=0.5 答：可以铺0.5米． **Q:** How thick can workers lay 10.5 cubic meters of yellow sand in a rectangular sand pit that is 6 meters long and 3.5 meters wide? (Using equations to solve) **A:** Assuming that the layer can be laid to a thickness of x meters, 6×3.5×x=10.5 21x=10.5 x=10.5÷21 x=0.5 Therefore, the layer can be laid to a thickness of 0.5 meters.	**Q:** 工人把10.5立方米的黄沙铺在一个长6米，宽3.5米的长方体沙坑里，可以铺多厚？（用方程解） **A:** 设可以铺x米， 6×3.5×x=10.5/21x=10.5 x=10.5÷21 x=0.5 答：可以铺0.5米． **Q:** How thick can workers lay 10.5 cubic meters of yellow sand in a rectangular sand pit that is 6 meters long and 3.5 meters wide? (Using equations to solve) **A:** Assuming that the layer can be laid to a thickness of x meters, 6×3.5×x=10.5/21x=10.5 x=10.5÷21 x=0.5 Therefore, the layer can be laid to a thickness of 0.5 meters.

Table 9: Case of our examples using rule-based methods. The translation is marked in blue. In the first case, “NUM1\n NUM2” is correctly transformed into “NUM1/NUM2”. However, in the second case, the raw web-crawled data is correct, but the rule-based method incorrectly alters the expression.

Error Types	Web-Crawled Examples	Model Converted Examples
Super/ Subscripts Errors	**Q:** 将一根绳子对折一次后从中间剪一刀,绳子变成3段;对折两次后从中间剪一刀,绳子变成5段:将这根绳子对折n次后从中间剪一刀,绳子变成___段. **A:** 根据分析可得:将一根绳子对折1次从中间一刀,绳子变成3段;有21+1=3.将一根绳子对折2次,从中间一刀,绳子变成5段;有22+1=5.依此类推,将这根绳子对折n次后从中间剪一刀,绳子变成(2n+1)段. [详解]解:∵对折1次从中间剪一刀,有21+1=3 对折2次,从中间剪一刀,有22+1=5. ∴对折n次,从中间剪一刀全部剪断后,绳子变成(2n+1)段. 故答案为:(2n+1). **Q:** After folding a rope in half once and cutting it in the middle, the rope becomes 3 segments. After folding it twice and cutting it in the middle, the rope becomes 5 segments. If we fold the rope n times and cut it in the middle, the rope will become ___ segments. **A:** According to the analysis, folding a rope once and cutting it in the middle results in 3 segments, which can be represented as 21+1=3. Folding the rope twice and cutting it in the middle results in 5 segments, represented as 22+1=5. Following this pattern, if we fold the rope n times and cut it in the middle, the rope will be divided into (2n+1) segments. [Detailed Solution] From the question, it can be inferred that if you fold it once and cut it in the middle, there will be 21+1=3. If you fold it twice and cut it in the middle, there will be 22+1=5. Therefore, if you fold it n times and cut it all in the middle, the rope will be divided into (2n+1) segments. Therefore, the answer is: (2n+1)	**Q:** 将一根绳子对折一次后从中间剪一刀，绳子变成3段；对折两次后从中间剪一刀，绳子变成5段：将这根绳子对折n次后从中间剪一刀，绳子变成( )段. **A:** 解：由题意得：对折1次从中间剪一刀，有2^1+1=3， 对折2次，从中间剪一刀，有2^2+1=5， …， 所以对折n次，从中间剪一刀全部剪断后，绳子变成（2^n+1）段， 故答案为：（2^n+1） **Q:** After folding a rope in half once and cutting it in the middle, the rope becomes 3 segments. After folding it twice and cutting it in the middle, the rope becomes 5 segments. If we fold the rope n times and cut it in the middle, the rope will become ( ) segments. **A:** From the question, it can be inferred that if you fold it once and cut it in the middle, there will be $2^1 + 1 = 3$. If you fold it twice and cut it in the middle, there will be $2^2 + 1 = 5$. Therefore, if you fold it n times and cut it all in the middle, the rope will be divided into $(2^n + 1)$ segments. Therefore, the answer is: $(2^n + 1)$
Missing/ Line Breaks	**Q:** 三个三角形+两个圆形=78，两个三角形+三个圆形=72，三角形和圆形各代表多少? **A:** 解：设三角形代表x 圆形代表y 由题意有：3x+2y=782x+3y=72解得x=18 y=12答：三角形代表18 圆形代表12. **Q:** Three triangles + two circles = 78, two triangles + three circles = 72, how much do triangles and circles represent respectively? **A:** Solution: Let a triangle represent x, a circle represents y, therefore: 3x+2y=782x+3y=72. The solution is x=18 y=12. Answer: A triangle represents 18, and a circle represents 12.	**Q:** 三个三角形+两个圆形=78，两个三角形+三个圆形=72，三角形和圆形各代表多少? **A:** 解：设三角形代表x，圆形代表y， 由题意有： 3x+2y=78 2x+3y=72 解得： x=18 y=12 答：三角形代表18，圆形代表12. **Q:** Three triangles + two circles = 78, two triangles + three circles = 72, how much do triangles and circles represent respectively? **A:** Solution: Let a triangle represent x, a circle represents y, therefore: 3x+2y=78 2x+3y=72 The solution is x=18 y=12 Answer: A triangle represents 18, and a circle represents 12.

Table 10: Case of our model transformed examples. The translation is marked in blue.