MathPrompter

Mathematical Reasoning using Large Language Models

Shima Imani, Liang Du, Harsh Shrivastava - Microsoft Research

ICLR 2023

Presented by Sudhansh

Introduction

MathPrompter

- Enhance performance of LLMs on arithmetic reasoning tasks
- LLMs struggle with “lack of confidence” due to generative nature
- Chain-of-thought also helps break down complex problems
- This work builds on top of zero-shot CoT to address validation and confidence concerns

Previous Works

Large Language Models are Zero-shot reasoners - Kojima et al. 2022

- Proposed a Zero-shot CoT on MultiArith dataset
- Improved SOTA results from 17.7% to 78.7%
- Key Insights
    - No check on validity of steps followed in CoT
    - Confidence in predictions is often not known

Motivation

MathPrompter

Authors studied techniques students use to verify their math solutions -

1. Compliance with known results - check with known problems to verify
2. Multi-verification - approach from multiple perspectives and compare results
3. Cross-checking - Verifying the correctness of intermediate steps
4. Compute Verification - Using a calculator to verify calculations

Method

Method

Prelude

- MultiArith dataset
    - 600 word problems with single numeric answers
- Ex:

  Q: Each chocolate bar in a box cost $4. If a box had 11 bars total and Vanessa sold all but 7 bars, how much money would she have made?

  A: 16

Proposed Method

MathPrompter

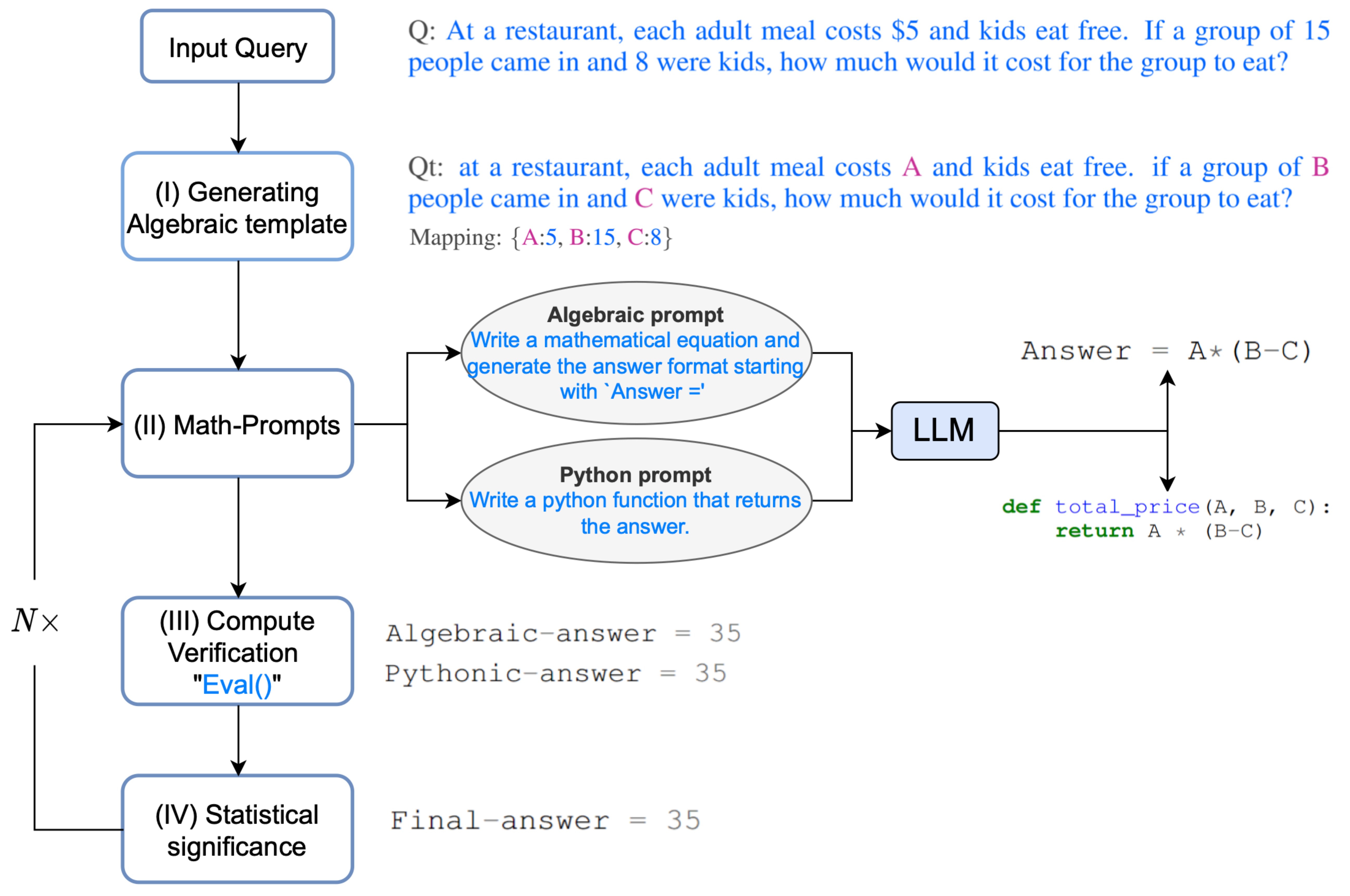

Input Query
Q: At a restaurant, each adult meal costs $5 and kids eat free. If a group of 15 people came in and 8 were kids, how much would it cost for the group to eat?
(I) Generating Algebraic template
Qt: at a restaurant, each adult meal costs A and kids eat free. if a group of B people came in and C were kids, how much would it cost for the group to eat?
Mapping: {A:5, B:15, C:8}
(II) Math-Prompts
Algebraic prompt
Write a mathematical equation and generate the answer format starting with `Answer ='
Python prompt
Write a python function that returns the answer.
LLM
Answer = A*(B-C)
def total_price(A, B, C):
    return A * (B-C)
N×
(III) Compute Verification "Eval()"
Algebraic-answer = 35
Pythonic-answer = 35
(IV) Statistical significance
Final-answer = 35

Figure 1: **MathPrompter flow.** We outline the MathPrompter process with an example alongside.

1. Generating Algebraic Templates

Convert the problem to an algebraic expression **Qt**

- Replace all the numeric values in the problem with algebraic symbols
- Ex:

  Q: At a restaurant, each adult meal costs $5 and kids eat free. If a group of 15 people came in and 8 were kids, how much would it cost for the group to eat?

  changes to

  Qt: at a restaurant, each adult meal costs A and kids eat free. if a group of B people came in and C were kids, how much would it cost for the group to eat?

  Mapping: {A:5, B:15, C:8}

1. Generating Algebraic Templates

Convert the problem to an algebraic expression **Qt**

- In the code, they perform this with basic regular expressions
  - Splits the input problem into numbers, words and operators
  - Checks for suffixes for each number to account for units and decimals
  - Assigns a variable to each numeric value found
  - Generates a new prompt

1. Generating Algebraic Templates

Convert the problem to an algebraic expression **Qt**

- Problems with this approach
    - If numbers are written as words, then it does not help the next steps much
    - Assumes numbers and variables do not occur in any other context
- Ex:

  Q: The sides of a triangle are 3cm, ten millimeters and 4 inches. What is the area of the triangle in square millimeters?

  Qt: The sides of a triangle are A cm, ten millimeters and B inches. What is the area of the triangle in square millimeters? {A: 3.0, B: 4.0}

2. Math Prompts

Multiple prompts to solve problem analytically in different ways

- The motivation is multi-verification and cross-checking mentioned previously
- Generate solutions

1. Algebraic way

   Prompt - Write a mathematical equation and generate the answer format starting with ‘Answer =’

2. Python code

   Prompt - Write a Python function that returns the answer.

2. Math Prompts

Algebraic Expression Generation

- Ex:

  At a restaurant, each adult meal costs $ A and kids eat free. If a group of B people came in and C were kids, how much would it cost for the group to eat?
  {A: 5.0, B: 15.0, C: 8.0}

  Expression: (A * (B - C))

- This implementation treats algebraic expressions as python code

2. Math Prompts

Algebraic Expression Generation

- The paper claims zero-shot but in practice requires few-shot to work correctly for the next steps
- Ex:

A company has A employees. Each employee works B hours per week. If the average hourly wage is C dollars and the company spends D percent of its revenue on salaries, what is the company's weekly revenue? {A: 10.0, B: 40.0, C: 30.0, D: 10.0}

Expression with zero-shot: \(\frac{A \cdot B \cdot C}{D/100}\)

Expression with few-shot: (A * B * C) / (D / 100)

2. Math Prompts

Python code

- Generates multiple functions to solve the problem
- Ex:

At a restaurant, each adult meal costs $ A and kids eat free. If a group of B people came in and C were kids, how much would it cost for the group to eat?
{A: 5.0, B: 15.0, C: 8.0}

Code:

```
def solution(A, B, C):
    return A * (B - C)
```

2. Math Prompts

Python code

- Relies heavily on code-generation ability of LLMs
- Code evaluated simply using `eval` method
  - LLM often generates wrong code (Tested it with Llama 1B)
  - The only fallback is repeating the steps with different seeds
  - Alternate could be to use a fine-tuned model for this part

2. Math Prompts

Multiple prompts to solve problem analytically in different ways

- These analytical versions of solutions show an *intermediate thought process* of the LLM
- Authors mention incorporating additional techniques will improve the accuracy and consistency
- Other ways according to me:
  - Embed other paradigms of reasoning
  - Ex: Chain of thought methods with self-consistency

3. Compute Verification

Evaluate the analytics solutions with multiple random values

- Evaluate the generated expressions with *randomized* key-value mappings
- If there is *consensus* among *k* different mappings for all the methods, generate the final answer
- Helps with validation of the model

4. Statistical Significance

Repeat across iterations, return the popular answer

- Repeats the previous steps and return the most frequent value for the answer
- Paper suggests doing it 5 times
- This can be parallelized for faster inference, but is still heavy compute
- Helps with the confidence of the answer

Experiments

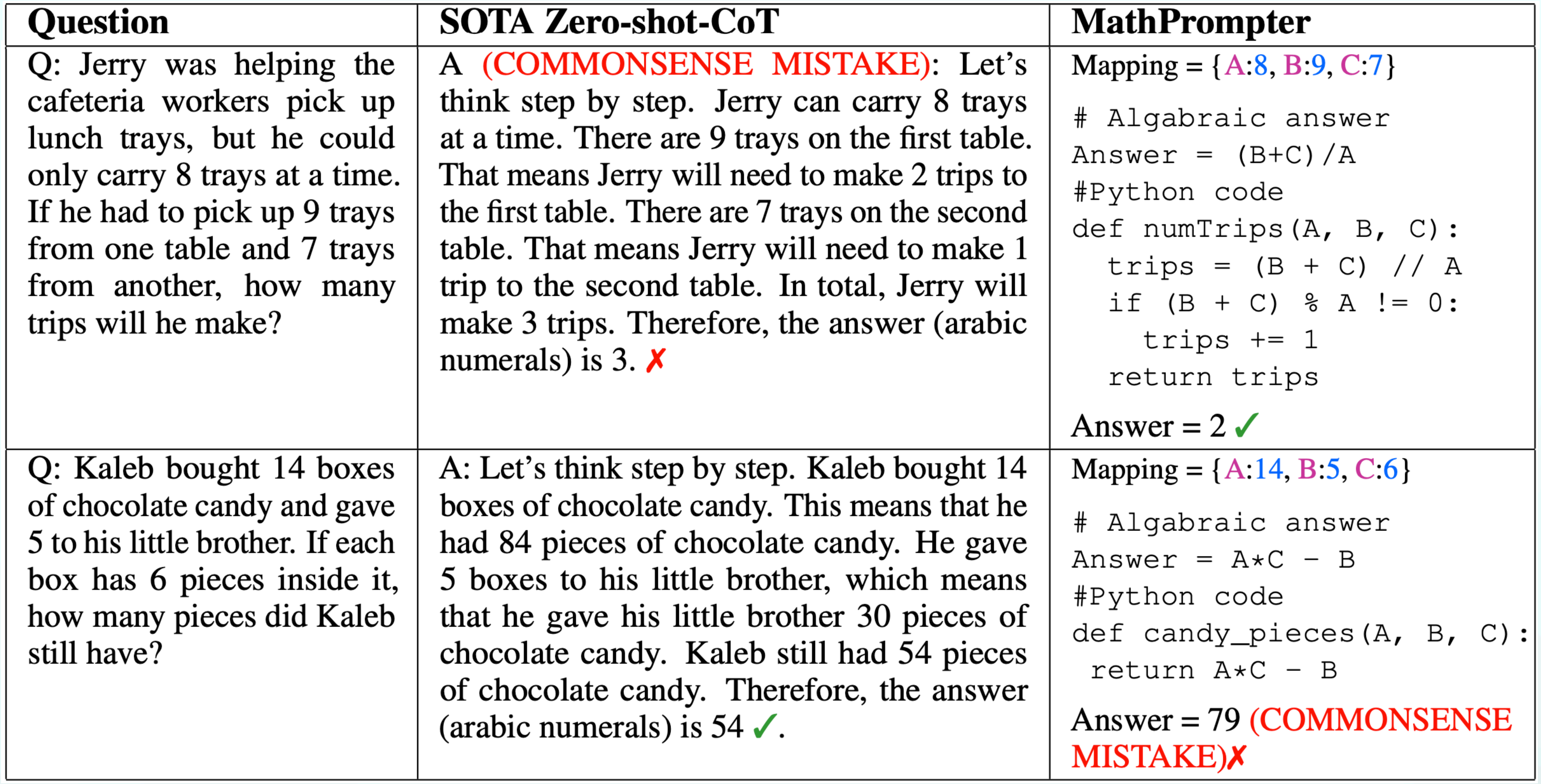
Experiment

Baselines

- Zero-shot model by Brown et al. (2020)
  - LLMs are Few-shot reasoners
  - Improve zero-shot by utilizing semantic relationships between classes
- Zero-shot-CoT by Kojima et al. (2022)
  - LLMs are Zero-shot reasoners
  - Reasoning and answer extraction steps

Experiment

Baselines



Question	SOTA Zero-shot-CoT	MathPrompter
Q: Jerry was helping the cafeteria workers pick up lunch trays, but he could only carry 8 trays at a time. If he had to pick up 9 trays from one table and 7 trays from another, how many trips will he make?	A (COMMONSENSE MISTAKE): Let's think step by step. Jerry can carry 8 trays at a time. There are 9 trays on the first table. That means Jerry will need to make 2 trips to the first table. There are 7 trays on the second table. That means Jerry will need to make 1 trip to the second table. In total, Jerry will make 3 trips. Therefore, the answer (arabic numerals) is 3. ✗	Mapping = {A:8, B:9, C:7} `# Algabraic answer` `Answer = (B+C)/A` `#Python code` `def numTrips(A, B, C):` `trips = (B + C) // A` `if (B + C) % A != 0:` `trips += 1` `return trips` Answer = 2 ✓
Q: Kaleb bought 14 boxes of chocolate candy and gave 5 to his little brother. If each box has 6 pieces inside it, how many pieces did Kaleb still have?	A: Let's think step by step. Kaleb bought 14 boxes of chocolate candy. This means that he had 84 pieces of chocolate candy. He gave 5 boxes to his little brother, which means that he gave his little brother 30 pieces of chocolate candy. Kaleb still had 54 pieces of chocolate candy. Therefore, the answer (arabic numerals) is 54 ✓.	Mapping = {A:14, B:5, C:6} `# Algabraic answer` `Answer = A*C - B` `#Python code` `def candy_pieces(A, B, C):` `return A*C - B` Answer = 79 (COMMONSENSE MISTAKE)✗

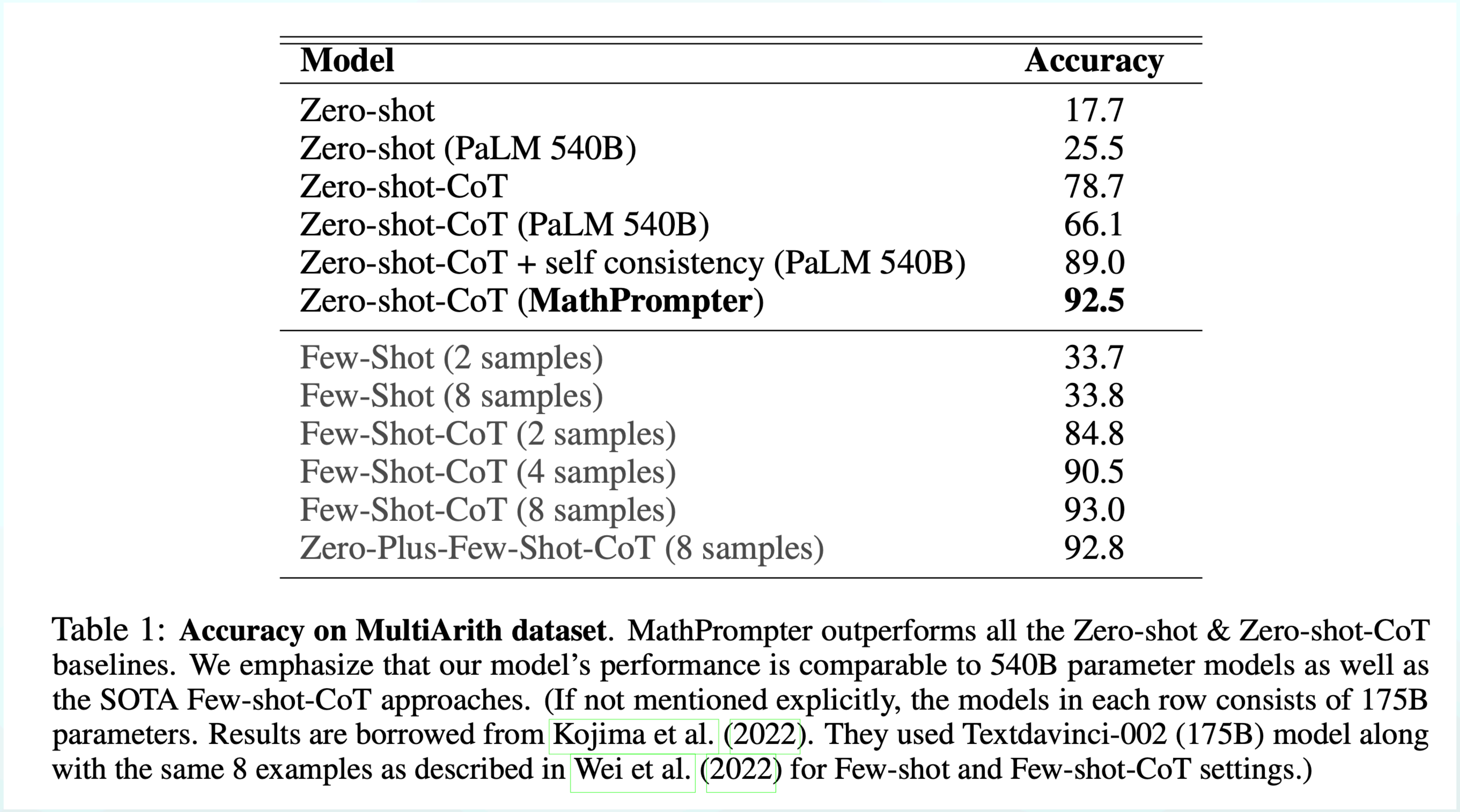
Experiment

Baselines - Observations

- One-step errors are remedied with consensus results
- Reasoning steps with CoT can be unnecessarily lengthy, but Python code can address with fewer tokens
- Reasoning steps may be correct, but the calculations can be incorrect
- Mathprompter fails sometimes due to lack of step-by-step reasoning

Results

Comparison with all baselines



Model	Accuracy
Zero-shot	17.7
Zero-shot (PaLM 540B)	25.5
Zero-shot-CoT	78.7
Zero-shot-CoT (PaLM 540B)	66.1
Zero-shot-CoT + self consistency (PaLM 540B)	89.0
Zero-shot-CoT (**MathPrompter**)	**92.5**
Few-Shot (2 samples)	33.7
Few-Shot (8 samples)	33.8
Few-Shot-CoT (2 samples)	84.8
Few-Shot-CoT (4 samples)	90.5
Few-Shot-CoT (8 samples)	93.0
Zero-Plus-Few-Shot-CoT (8 samples)	92.8

Table 1: **Accuracy on MultiArith dataset**. MathPrompter outperforms all the Zero-shot & Zero-shot-CoT baselines. We emphasize that our model's performance is comparable to 540B parameter models as well as the SOTA Few-shot-CoT approaches. (If not mentioned explicitly, the models in each row consists of 175B parameters. Results are borrowed from Kojima et al. (2022). They used Textdavinci-002 (175B) model along with the same 8 examples as described in Wei et al. (2022) for Few-shot and Few-shot-CoT settings.)

Discussion

Method and results

- Model addresses validation and certainty with probabilistic LLM generation
- Model does not work with other datasets
    - Units and word number limitations
    - Cannot work with more complicated problems that require multiple steps
- Unsure about the results provided since open-source implementation uses few-shot prompting

Limitations and Future Work

Limitations

MathPrompter

- Model lacks semantic understanding of the problem. Ex:

Q: If AAA_4 can be expressed as 33_b, where A is a digit in base 4 and b is a base greater than 5, what is the smallest possible sum A+b?

Qt: If AAA_ A can be expressed as B _b, where A is a digit in base C and b is a base greater than D what is the smallest possible sum A + b? {A: 4.0, B: 33.0, C: 4.0, D: 5.0}

- Expression generated: A + (D + 1)

Limitations

MathPrompter

- Does not make use of known mathematical results and other shortcuts. Ex:

  Q: What is the area under the curve y = x^2 + 1 from x = 0 to x = 2?

  Qt: What is the area under the curve y = x ^ A + B from x = C to x = D ? {A: 2.0, B: 1.0, C: 0.0, D: 2.0}

  Expression: ∫(C to D) (x^A + B) dx

  Code:

```
import sympy as sp
def solution(A, B, C, D):
    x = sp.symbols('x')
    function = x**A + B
    area = sp.integrate(function, (x, C, D))
    return area
```

Future Work

MathPrompter

- The error-checking steps can be enhanced to verify code and reasoning
- Using symbolic manipulations in algebraic expression generation
- Including other methods such as graph-based reasoning

Conclusion

MathPrompter

- Effectively addresses validation and confidence issues
- Achieves SOTA performance on MultiArith-like datasets
- Model works well for math reasoning problems in other domains
  - Experimented with problems in physics and chemistry
- Does not work well with more complicated math problems

Recent Updates

GSM-Symbolic

Understanding the Limitations of Mathematical Reasoning in Large Language Models

- Claims that LLMs primarily engage in probabilistic pattern matching than formal logic reasoning
- Study shows that LLMs are not robust to different instantiations
- Irrelevant information in the question also significantly affects performance
- Complexity deteriorates performance of models significantly
- Benchmark designed to provide more nuanced and reliable assessments

Thank You!

Any Questions?

References

MathPrompter

- https://github.com/arkilpatel/SVAMP
- https://huggingface.co/datasets/ChilleD/MultiArith?row=5