组会分享

Case-Based or Rule-Based: How Do Transformers Do the Math?
Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement

韩子坚

华中师范大学计算机学院

2024 年 10 月 5 日

Content

1. Case or Rule
   - case-based and rule-based 的原理
   - Leave-Square-Out method
   - rule-based setting
   - 实验结论
2. RFFT(Rule-Following Fine-Tuning)
3. Qwen-2.5-Math

1. Case or Rule
   - case-based and rule-based 的原理
   - Leave-Square-Out method
   - rule-based setting
   - 实验结论
2. RFFT(Rule-Following Fine-Tuning)
3. Qwen-2.5-Math

case-based and rule-based 的原理

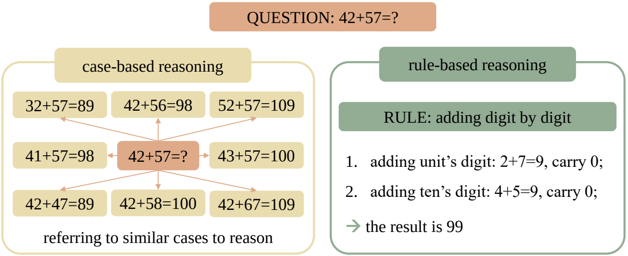


图 1: Illustrations of case-based and rule-based reasoning.

case-based and rule-based 的原理

case-based 依赖训练时的语料库，如果语料库中没有需要推理的这个问题，则准确度会大幅下降。

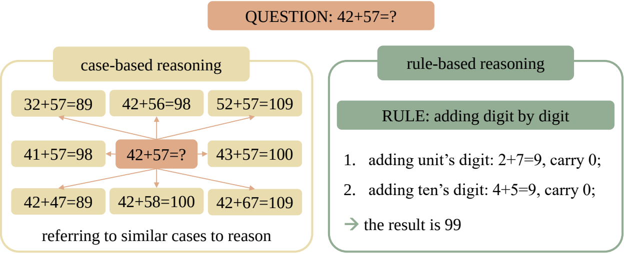


图 1: Illustrations of case-based and rule-based reasoning.

case-based and rule-based 的原理

case-based 依赖训练时的语料库，如果语料库中没有需要推理的这个问题，则准确度会大幅下降。

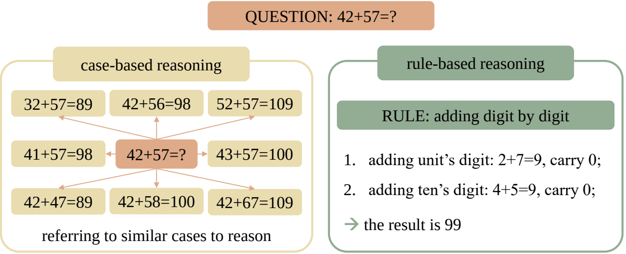


图 1: Illustrations of case-based and rule-based reasoning.

rule-based 依赖数学规则，即使语料库中没有这个问题，也可以根据从语料库中学习到的数学规则推理出正确的答案。

1. Case or Rule
    - case-based and rule-based 的原理
    - Leave-Square-Out method
    - rule-based setting
    - 实验结论
2. RFFT(Rule-Following Fine-Tuning)
3. Qwen-2.5-Math

Leave-Square-Out method

Leave-Square-Out method (留方法，LSO) 是作者提出的一种交叉验证（cross-validation）方法，用于评估机器学习模型的性能。它是留一法（Leave-One-Out）的扩展。与留一法相比，Leave-Square-Out 方法不是每次只留一个样本进行测试，而是每次留出 k^2 个样本进行测试，其中 k 是一个正整数。当数据集规模较大时，这种方法可以更好地评估模型的泛化能力。

Leave-Square-Out method

Leave-Square-Out method (留方法，LSO) 是作者提出的一种交叉验证（cross-validation）方法，用于评估机器学习模型的性能。它是留一法（Leave-One-Out）的扩展。与留一法相比，Leave-Square-Out 方法不是每次只留一个样本进行测试，而是每次留出 k^2 个样本进行测试，其中 k 是一个正整数。当数据集规模较大时，这种方法可以更好地评估模型的泛化能力。

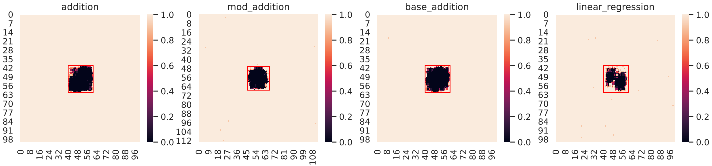


图 2: Accuracy of Leave-Square-Out method

Leave-Square-Out method

Leave-Square-Out method (留方法，LSO) 是作者提出的一种交叉验证（cross-validation）方法，用于评估机器学习模型的性能。它是留一法（Leave-One-Out）的扩展。与留一法相比，Leave-Square-Out 方法不是每次只留一个样本进行测试，而是每次留出 k^2 个样本进行测试，其中 k 是一个正整数。当数据集规模较大时，这种方法可以更好地评估模型的泛化能力。

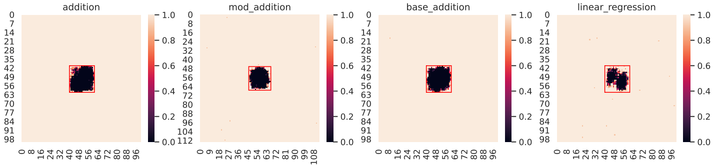


图 2: Accuracy of Leave-Square-Out method

The appearance of holes in the figure indicates that the test samples away from the boundary of the training set are hard for the models to correctly infer.

1. Case or Rule
   - case-based and rule-based 的原理
   - Leave-Square-Out method
   - rule-based setting
   - 实验结论
2. RFFT(Rule-Following Fine-Tuning)
3. Qwen-2.5-Math

rule-based setting

rule based 的重要性

Rule-based reasoning is essential for models to achieve systematic and length generalization so that they can be applied to new, unseen scenarios without re-training.

rule-based setting

rule based 的重要性

Rule-based reasoning is essential for models to achieve systematic and length generalization so that they can be applied to new, unseen scenarios without re-training.

rule based 应注意的事情

training set should always provide the necessities for the model to learn the underlying rule. For example, the training set should at least cover all the tokens used in the test set in order to develop a systematic rule that applies to the whole dataset.

1. Case or Rule
   - case-based and rule-based 的原理
   - Leave-Square-Out method
   - rule-based setting
   - 实验结论
2. RFFT(Rule-Following Fine-Tuning)
3. Qwen-2.5-Math

实验结论

- test squares 的位置不会影响实验结果

实验结论

- test squares 的位置不会影响实验结果
- test squares 的大小会影响实验结果（the hole disappears when the test square shrinks to less than a small size）

实验结论

- test squares 的位置不会影响实验结果
- test squares 的大小会影响实验结果（the hole disappears when the test square shrinks to less than a small size）
- scratchpad cannot teach transformers to perform rule-based reasoning. (why? scratchpad fine-tuning fails to teach transformers the actually applied "rule" behind each step. This is like teaching children addition only by showing them examples, without telling them the rationales behind each step.)

实验结论

- test squares 的位置不会影响实验结果
- test squares 的大小会影响实验结果（the hole disappears when the test square shrinks to less than a small size）
- scratchpad cannot teach transformers to perform rule-based reasoning. (why? scratchpad fine-tuning fails to teach transformers the actually applied "rule" behind each step. This is like teaching children addition only by showing them examples, without telling them the rationales behind each step.)
- 模型和数据集的增大几乎不会影响实验结果，“holes”仍然存在

1. Case or Rule

2. RFFT(Rule-Following Fine-Tuning)
   - RFFT 的步骤
   - RFFT 结果分析
   - 误差分析

3. Qwen-2.5-Math

direct answer

input:
59+13=

output:
72

scratchpad

input:
59+13=

output:
```
59+13, , C:0
# added 9+3+0=2
5+1, 2, C:1
# added 5+1+1=7
+, 72, C:0

72
```

rule following

input:
Follow the code step by step to answer the question:
[5, 9]+[1, 3]=
```
def sum_digit_by_digit(num1, num2):
    result=[]
    carry=0
    while num1 or num2:
        digit1=num1.pop() if num1 else 0
        digit2=num2.pop() if num2 else 0
        total=digit1+digit2+carry
        result.insert(0,total%10)
        carry=total//10
    if carry:
        result.insert(0,carry)
    return result
```

output:
```
num1 = [5,9]
num2 = [1,3]
1. Initialize Result and Carry
result = []
carry = 0
2. Loop Through Each Digit
```while num1 or num2:```
2.1 check the stop criterion
num1 = [5,9]
num2 = [1,3]
bool(num1) = True
bool(num2) = True
num1 or num2 = True
enter the loop
2.2 one iteration
```digit1=num1.pop() if num1 else 0```
num1 = [5,9]
bool(num1) = True
num1.pop()
num1 = [5]
digit1 = 9
```digit2=num2.pop() if num2 else 0```
num2 = [1,3]
bool(num2) = True
num2.pop()
num2 = [1]
digit2 = 3
```total=digit1+digit2+carry```
total = digit1+digit2+carry =
9+3+0 = 12
```result.insert(0,total%10)```
result = []
total%10 = 12%10 = 2
result = [2]
```carry=total//10```
carry = 12//10 = 1
2.3 back to the start of the loop
...
3. Check Remaining Carry
```if carry: result.insert(0,carry)```
result = [7,2]
carry = 0
bool(carry) = False
pass
result = [7,2]
4. Return Result
```return result```
result = [7,2]
```

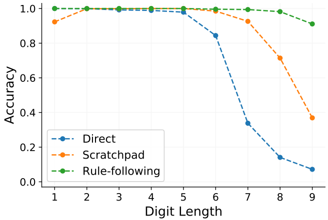
图 3: Examples of input-output sequence of question $59 + 13$

Step 1: Explicitly list the rules for solving a given task in the input.

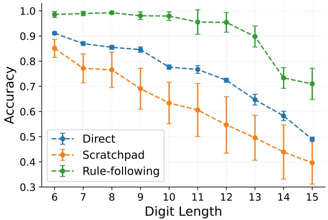
Step 2: Finetune the model to follow the rules step by step.

可以有不用的方式阐述规则， including programs, pseudo-code, first-order logic, natural language, etc.

RFFT 结果分析



(a) Accuracy of Llama-7B fine-tuned with three methods tested on addition with 1-9 digits.



(b) Accuracy of GPT-3.5 fine-tuned with three methods tested on addition with 6-15 digits.

图 4: Accuracy of Llama-2-7B and GPT-3.5-turbo fine-tuned with direct answer, scratchpad and rule following on addition.

Llama-2-7B: RFFT: 91.1% acc with 9-digit sums

scratchpad: less than 40% acc

GPT-3.5-turbo: over 95% acc on 12-digit addition (only 100 training samples)

1. Case or Rule
2. RFFT(Rule-Following Fine-Tuning)
   - RFFT 的步骤
   - RFFT 结果分析
   - 误差分析
3. Qwen-2.5-Math

误差分析

- RFFT 并不能带来 100% 的准确性，作者发现大模型在计算时的每一步总能找到正确的规则，但是在一些基本的运算中会出现失误的现象，这可能是由于大模型幻觉或者是大模型处理长文本的局限性。

误差分析

- RFFT 并不能带来 100% 的准确性，作者发现大模型在计算时的每一步总能找到正确的规则，但是在一些基本的运算中会出现失误的现象，这可能是由于大模型幻觉或者是大模型处理长文本的局限性。
- RFFT as a Meta Learning Ability：stronger models indeed need less examples to learn rules.

误差分析

- RFFT 并不能带来 100% 的准确性，作者发现大模型在计算时的每一步总能找到正确的规则，但是在一些基本的运算中会出现失误的现象，这可能是由于大模型幻觉或者是大模型处理长文本的局限性。
- RFFT as a Meta Learning Ability：stronger models indeed need less examples to learn rules.
- Given detailed rules, LLMs have certain abilities to follow the rules, which allows the models to show some reasoning ability on unfamiliar tasks. However, they do not gain a competitive edge from the rules in tasks already familiar to them.

1. Case or Rule
2. RFFT(Rule-Following Fine-Tuning)
3. Qwen-2.5-Math
   - 横向对比其他模型的得分表现
   - Self-improvement techniques
   - Qwen 2.5 math 的训练流程
   - pre-training
   - post-training
   - CoT and TIR
   - 实验结果

1 Case or Rule

2 RFFT(Rule-Following Fine-Tuning)

3 Qwen-2.5-Math
- 横向对比其他模型的得分表现
- Self-improvement techniques
- Qwen 2.5 math 的训练流程
- pre-training
- post-training
- CoT and TIR
- 实验结果

横向对比其他模型的得分表现

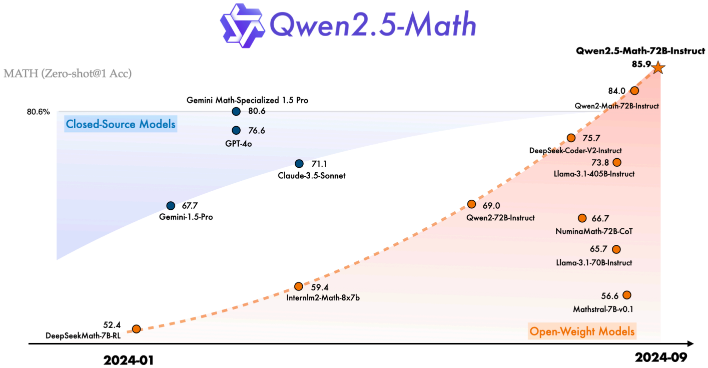


图 5: The pass@1 performance of Qwen2.5-Math-72B-Instruct on MATH by the Chain-of-Thought reasoning.

1. Case or Rule
2. RFFT(Rule-Following Fine-Tuning)
3. Qwen-2.5-Math
   - 横向对比其他模型的得分表现
   - Self-improvement techniques
   - Qwen 2.5 math 的训练流程
   - pre-training
   - post-training
   - CoT and TIR
   - 实验结果

Self-improvement techniques

- In pre-training, we employ Qwen2-Math-Instruct to synthesize math queries and corresponding responses on a large scale to enrich the pre-training corpus of Qwen2.5-Math.

Self-improvement techniques

- In pre-training, we employ Qwen2-Math-Instruct to synthesize math queries and corresponding responses on a large scale to enrich the pre-training corpus of Qwen2.5-Math.
- In post-training, we train a reward model on massive sampling from previous models and apply it to the iterative evolution of data in supervised fine-tuning.

Self-improvement techniques

- In pre-training, we employ Qwen2-Math-Instruct to synthesize math queries and corresponding responses on a large scale to enrich the pre-training corpus of Qwen2.5-Math.
- In post-training, we train a reward model on massive sampling from previous models and apply it to the iterative evolution of data in supervised fine-tuning.
- Use Qwen2.5-Math-RM in reinforcement learning and best-of-N sampling during inference.

Qwen 2.5 math 的训练流程

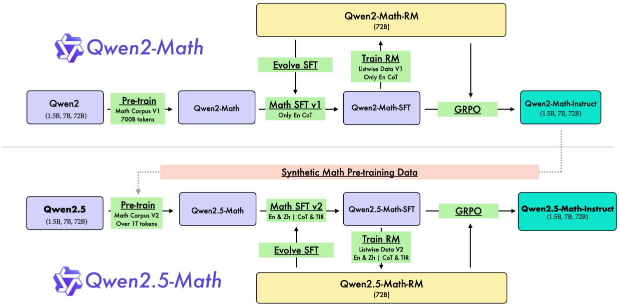


图 6: The development pipelines of Qwen2-Math and Qwen2.5-Math.

1. Start -> Qwen Math Corpus v1 (700B tokens) -> Qwen2-Math Base Models
2. Qwen2-Math-72B -> Qwen2-Math-RM -> SFT Data -> Qwen2-Math-Instruct
3. Qwen2-Math-72B-Instruct -> Additional Data -> Qwen Math Corpus v2 (1T tokens)
4. Qwen Math Corpus v2 -> Qwen2.5-Math Models
5. Qwen2.5-Math-RM -> Qwen2.5-Math-Instruct

1. Case or Rule
2. RFFT(Rule-Following Fine-Tuning)
3. Qwen-2.5-Math
   - 横向对比其他模型的得分表现
   - Self-improvement techniques
   - Qwen 2.5 math 的训练流程
   - pre-training
   - post-training
   - CoT and TIR
   - 实验结果

data

quantity:

1. Train a FastText classifier utilizing high-quality mathematical seed data and general text data.

data

quantity:

1. Train a FastText classifier utilizing high-quality mathematical seed data and general text data.
2. Leverage meta-information from the recalled data to expand the data pool for mathematical data retrieval.

data

quantity:

1. Train a FastText classifier utilizing high-quality mathematical seed data and general text data.
2. Leverage meta-information from the recalled data to expand the data pool for mathematical data retrieval.

quality:

1. Utilize the Qwen2-0.5B-Instruct model, augmented with prompt engineering, to evaluate the quality of potential data entries.

data

quantity:

1. Train a FastText classifier utilizing high-quality mathematical seed data and general text data.
2. Leverage meta-information from the recalled data to expand the data pool for mathematical data retrieval.

quality:

1. Utilize the Qwen2-0.5B-Instruct model, augmented with prompt engineering, to evaluate the quality of potential data entries.
2. Employ the Qwen2-72B-Instruct model to synthesize a large amount of mathematical pre-training corpus.

1. Case or Rule

2. RFFT(Rule-Following Fine-Tuning)

3. Qwen-2.5-Math
   - 横向对比其他模型的得分表现
   - Self-improvement techniques
   - Qwen 2.5 math 的训练流程
   - pre-training
   - post-training
   - CoT and TIR
   - 实验结果

post-training

- Aggregate more high-quality mathematical data, especially in Chinese, sourced from web documents, books, and code repositories across multiple recall cycles. Qwen Math Corpus v1(700B tokens) $->$Qwen Math Corpus v2(over 1T tokens)

post-training

- Aggregate more high-quality mathematical data, especially in Chinese, sourced from web documents, books, and code repositories across multiple recall cycles. Qwen Math Corpus v1(700B tokens) $->$Qwen Math Corpus v2(over 1T tokens)
- Leverage the Qwen2.5 series base models for parameter initialization instead of initializing from the Qwen2 series.

1. Case or Rule
2. RFFT(Rule-Following Fine-Tuning)
3. Qwen-2.5-Math
   - 横向对比其他模型的得分表现
   - Self-improvement techniques
   - Qwen 2.5 math 的训练流程
   - pre-training
   - post-training
   - CoT and TIR
   - 实验结果

CoT and TIR

Chain-of-Thought Dataset Synthesis

- Content: Comprises 580K English and 500K Chinese mathematical problems, collected from sources like GSM8K, MATH, and NuminaMath, and enriched with K-12 Chinese problems to enhance the Qwen2.5-Math model.

CoT and TIR

Chain-of-Thought Dataset Synthesis

- Content: Comprises 580K English and 500K Chinese mathematical problems, collected from sources like GSM8K, MATH, and NuminaMath, and enriched with K-12 Chinese problems to enhance the Qwen2.5-Math model.
- Problem Complexity: A **difficulty-scoring model** is used to ensure a balanced distribution of problem complexities.

CoT and TIR

Chain-of-Thought Dataset Synthesis

- Content: Comprises 580K English and 500K Chinese mathematical problems, collected from sources like GSM8K, MATH, and NuminaMath, and enriched with K-12 Chinese problems to enhance the Qwen2.5-Math model.
- Problem Complexity: A **difficulty-scoring model** is used to ensure a balanced distribution of problem complexities.
- Response Construction: Utilizes iterative approaches with rejection sampling and reward modeling to refine responses, incorporating majority voting for synthesized problems without definitive answers. An additional refinement iteration is conducted for Qwen2.5-Math.

CoT and TIR

Tool-Integrated Reasoning Data Synthesis

- Objective: To overcome CoT prompting challenges related to computational accuracy and complex algebraic problem-solving by integrating a Python interpreter as a reasoning aid.

CoT and TIR

Tool-Integrated Reasoning Data Synthesis

- Objective: To overcome CoT prompting challenges related to computational accuracy and complex algebraic problem-solving by integrating a Python interpreter as a reasoning aid.
- Dataset Content: Contains 190K annotated and 205K synthesized problems from datasets like GSM8K, MATH, and CollegeMath. An additional 75K problems are translated into Chinese to bolster proficiency in the language.

CoT and TIR

Tool-Integrated Reasoning Data Synthesis

- Objective: To overcome CoT prompting challenges related to computational accuracy and complex algebraic problem-solving by integrating a Python interpreter as a reasoning aid.
- Dataset Content: Contains 190K annotated and 205K synthesized problems from datasets like GSM8K, MATH, and CollegeMath. An additional 75K problems are translated into Chinese to bolster proficiency in the language.
- Response Construction: Employs online **Rejection Fine-Tuning (RFT)** to generate reasoning paths that align with reference answers. Nucleus sampling, deduplication, and majority voting techniques are used to ensure a diverse and accurate dataset for model fine-tuning.

1. Case or Rule
2. RFFT(Rule-Following Fine-Tuning)
3. Qwen-2.5-Math
   - 横向对比其他模型的得分表现
   - Self-improvement techniques
   - Qwen 2.5 math 的训练流程
   - pre-training
   - post-training
   - CoT and TIR
   - 实验结果

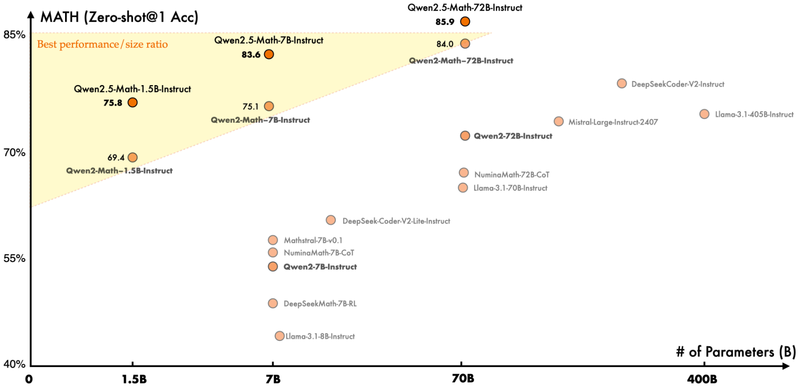


图 7: The Performance of Qwen2.5-Math-1.5/7/72B-Instruct on MATH by CoT compared to models of the same size.

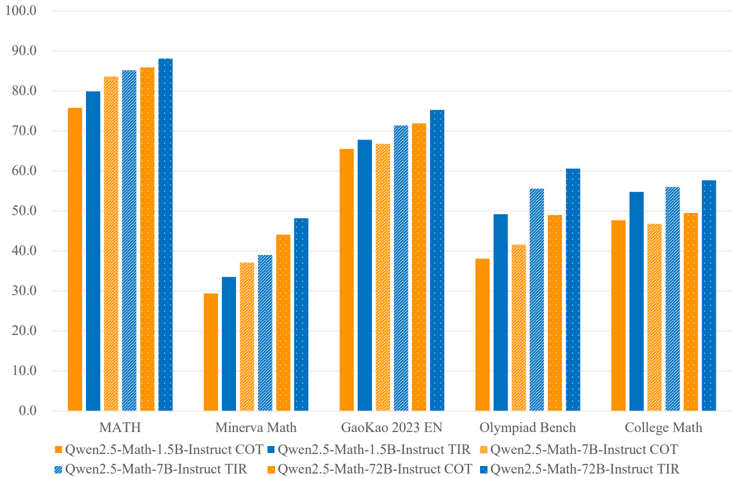


图 8: The Performance of Qwen2.5-Math-Instruct by using TIR compared to using CoT.

Thank you!