

チーム dentaku

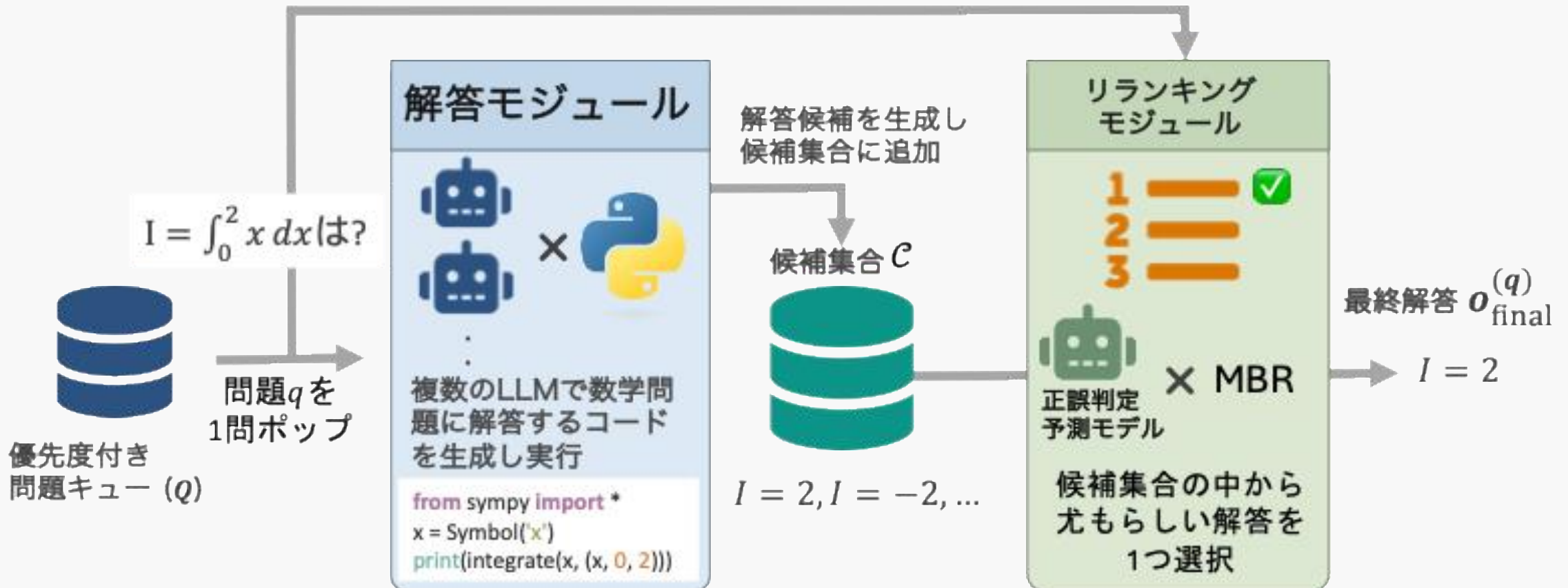
FT-LLM 2026 (数学タスク) システム紹介

工藤 慧音^{1,2}, 日高雅俊³, 塩野 大輝¹, 鴨田 豪^{4,5}, 葉 夢宇¹, 高橋 良允^{1,2},
青木 洋一^{1,2}, 矢野 一樹¹, 谷口 雅弥^{2,1}, 伊藤 拓海^{1,3}, 坂口 慶祐^{1,2}, 鈴木 潤^{1,2,6}

¹東北大学, ²理化学研究所, ³Machine Learning Solutions 株式会社,
⁴総合研究大学院大学, ⁵国立国語研究所, ⁶国立情報学研究所 LLMC

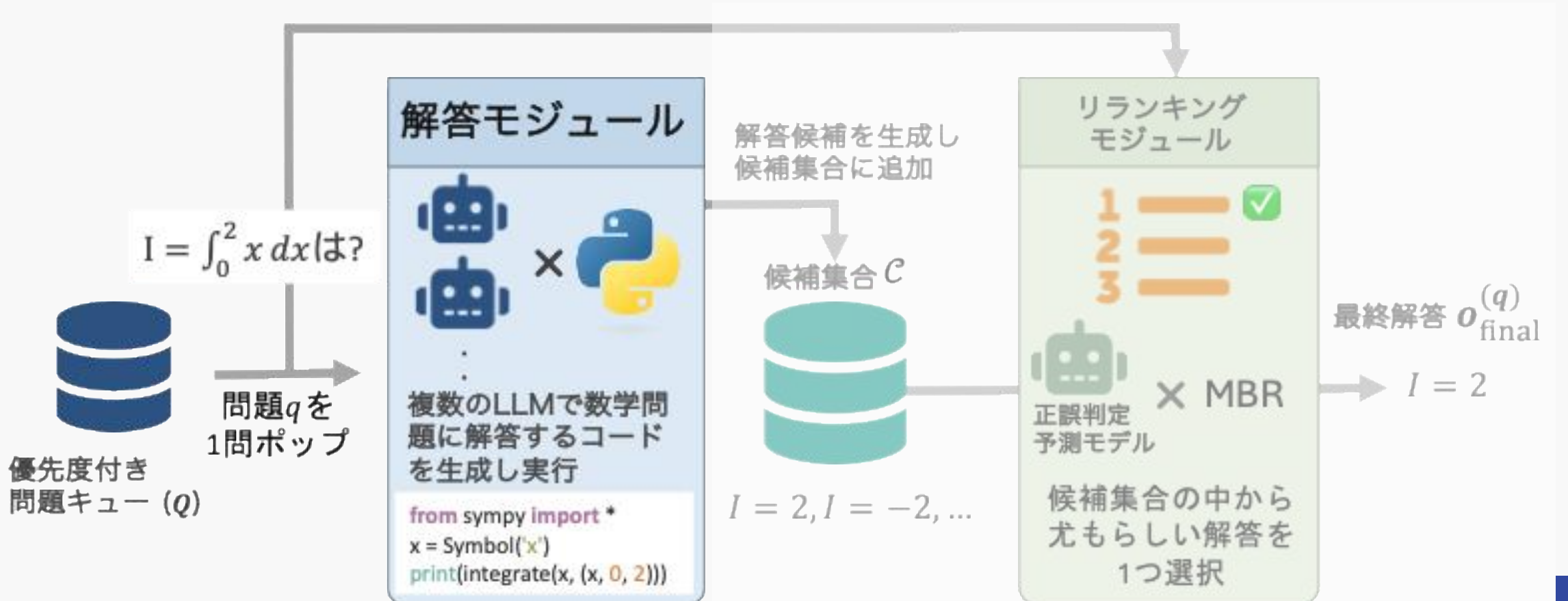
目標 / システム概要

- 目標: LLM-jp-v4 8Bによるgpt-oss-120b [OpenAI. gpt-oss-120b & gpt-oss-20b Model Card. 2024]
出力の模倣 + α



目標 / システム概要

- 目標: LLM-jp-v4 8Bによるgpt-oss-120b [OpenAI. gpt-oss-120b & gpt-oss-20b Model Card. 2024] 出力の模倣 + α



解答生成モジュール

- コードとして数学問題への解法を生成

[Chen et al. Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks. 2023]

問題:

次の式で与えられる数列がある。

$$a_n = \left(\frac{2}{n}\right)^n$$

この数列の極限值 $\lim_{n \rightarrow \infty} a_n$ を求めなさい。



思考過程 (thinking):

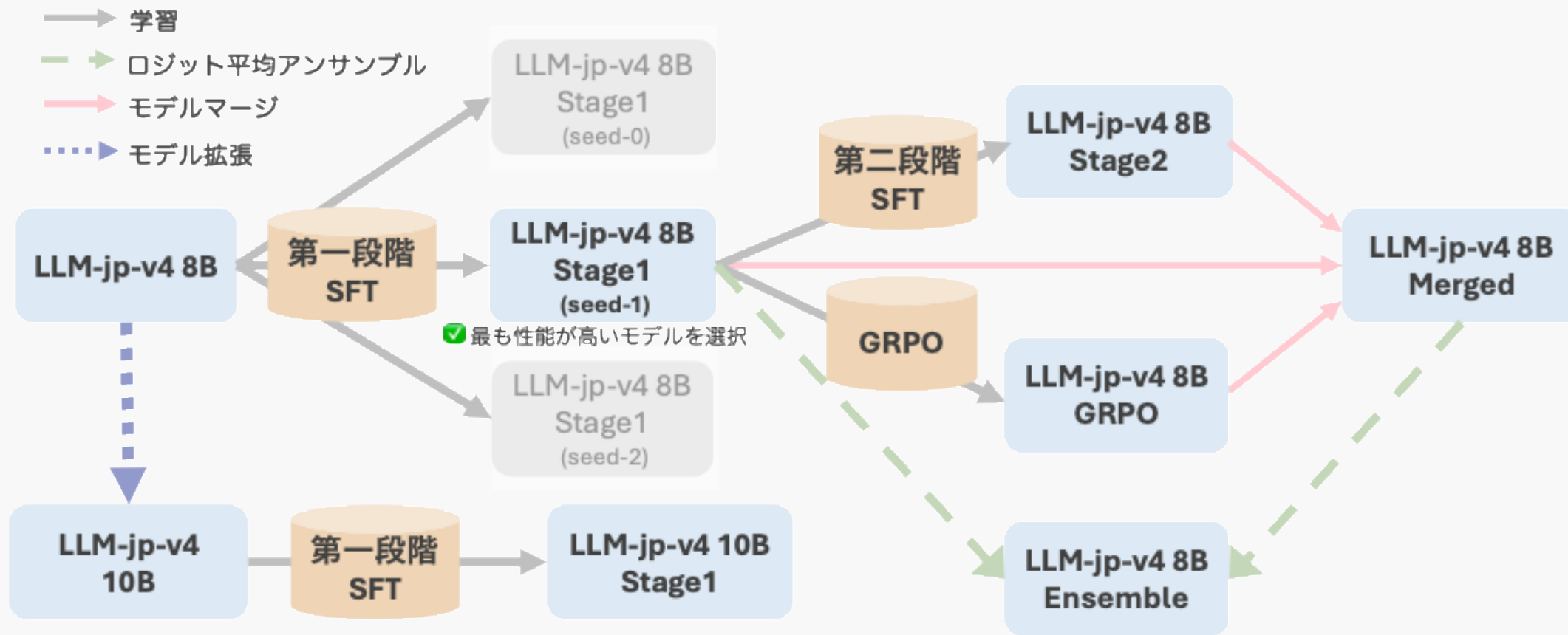
The user wants Python code that uses sympy to compute the limit of $a_n = (2/n)^n$ as $n \rightarrow \infty$. The limit is 0. Need to produce code that defines symbols, computes limit, outputs latex....

+

解法コード:

```
from sympy import latex, limit, oo, symbols
n = symbols("n", integer=True, positive=True)
a_n = (2 / n) ** n
L = limit(a_n, n, oo)
latex_answer = latex(L)
print(f"Answer: {latex_answer}")
```

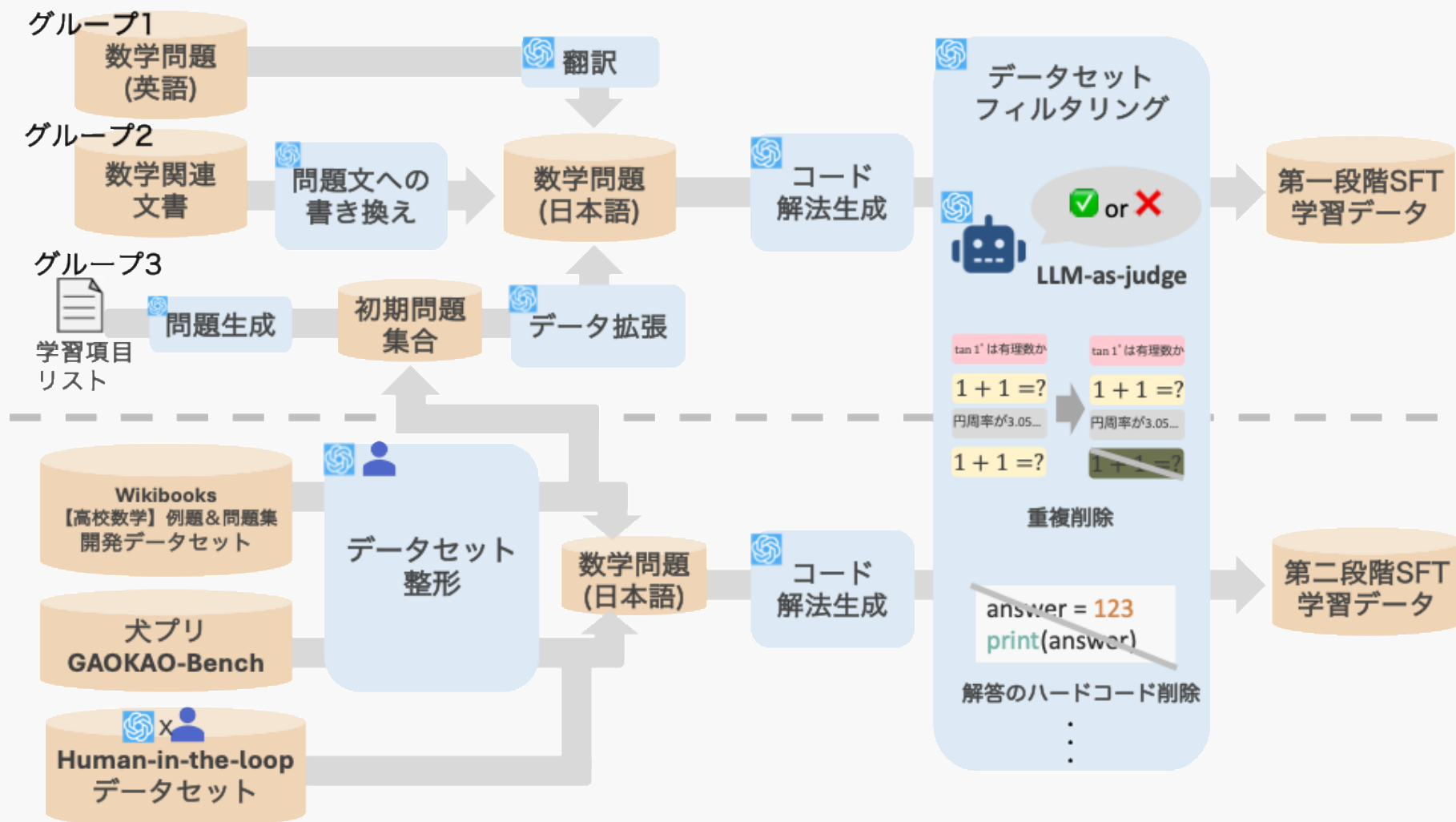
モデル構築



- **モデル拡張:** LLaMA Proによる層の追加
- **ロジット平均アンサンブル:** 複数モデルが出力する確率分布を平均

データセット構築の流れ

- gpt-oss-120bを教師モデルとして知識蒸留



リランキングモジュール

2つのコンポーネントのスコアの組み合わせ

- 正誤判定予測モデル
- 最小ベイズリスク復号 (MBR)



正誤判定予測モデル

- 問題と候補のコードから正誤判定を予測
 - gpt-oss-120bの知識蒸留により構築

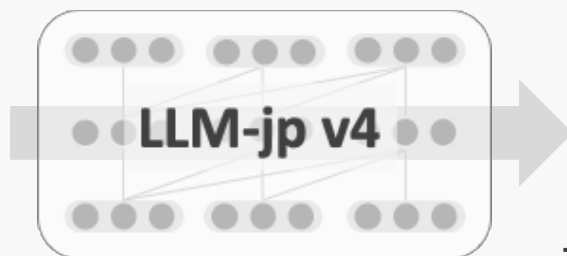
問題: 正の数 (a, b, c) が等比数列をなすとする.
すなわち

$$b = ar, c = ar^2$$

このとき, 積 $abc = 216$ かつ和 $a + b + c = 21$ が成り立つ. a の値を求めなさい
(ただし $a < b < c$ とする).

コード:

```
from sympy import *
r = symbols("r", positive=True, real=True)
eq = Eq(21 * r**2 * (1 + r + r**2), 216)
solutions_r = solve(eq, r)
r_val = [
    sol for sol in solutions_r if sol.is_real and
sol > 1
][0]
a = simplify(6 / (1 + r_val + r_val**2))
latex_answer = latex(a)
print(f"Answer: {latex_answer}")
```



思考過程 (thinking):

We need to evaluate if the code correctly solves the problem. Let's analyze problem:

+

最終判断:

Step by Step Rationale

...

- Consequently, the solution method is logically flawed and does not solve the given problem.

Decision

Invalid

MBR

- 本システムの範疇では実質的に (ほぼ) 多数決

math-evalを用いて候補集合 C をリランキング

$$s_i = \sum_{j \neq i} \text{MathEval}(o_i, o_j).$$

- s_i : i 番目の候補のスコア
- $o_i, o_j \in C$: i, j 番目のコードの実行結果

問題: $\int_0^2 x dx$ を解け

擬似参照

	$o_0 = "2"$	$o_1 = "2"$	$o_2 = "4"$	
$o_0 = "2"$		T	F	✓
$o_1 = "2"$	T		F	✓
$o_2 = "4"$	F	F		

事後評価設定

- データセット: 本コンペ評価データセット498問
- 評価指標: Math-Eval (devブランチ)
- 評価対象
 - 提出システム
 - 複数のモデルからの出力を集めてリランキング
 - 各段階のモデル
 - 教師モデル (gpt-oss-120b)

※ コンペの評価時と完全に同等の環境ではない & ある程度のランダム性があるため、コンペの評価結果とは一致しない可能性があります

事後評価結果

	math-eval↑
a. LLM-jp-v4 8B (Stage1)	86.75
b. LLM-jp-v4 8B (Stage 2 + GRPO)	92.57
c. LLM-jp-v4 8B (Stage 2 + GRPO) + リランキング	93.98
d. LLM-jp-v4 10B (Stage1)	89.96
e. LLM-jp-v4 8B (ロジットアンサンブル)	90.96
f. 提出システム	91.16
g. LLM-jp-v4 8B (Stage1, 思考過程なし)	78.31
h. gpt-oss-120b	92.57

事後評価結果

	math-eval↑
a. LLM-jp-v4 8B (Stage1)	86.75
b. LLM-jp-v4 8B (Stage 2 + GRPO)	92.57
c. LLM-jp-v4 8B (Stage 2 + GRPO) + リランキング	93.98
d. LLM-jp-v4 10B (Stage1)	89.96
e. LLM-jp-v4 8B (ロジットアンサンブル)	90.96
f. 提出システム	91.16
g. LLM-jp-v4 8B (Stage1, 思考過程なし)	78.31
h. gpt-oss-120b	92.57



単一モデルでgpt-oss-120bと同等の性能

事後評価結果

	math-eval↑
a. LLM-jp-v4 8B (Stage1)	86.75
b. LLM-jp-v4 8B (Stage 2 + GRPO)	92.57
c. LLM-jp-v4 8B (Stage 2 + GRPO) + リランキング	93.98
d. LLM-jp-v4 10B (Stage1)	89.96
e. LLM-jp-v4 8B (ロジットアンサンブル)	90.96
f. 提出システム	91.16
g. LLM-jp-v4 8B (Stage1, 思考過程なし)	78.31
h. gpt-oss-120b	92.57



リランキングによりさらに性能向上

事後評価結果

	math-eval↑
a. LLM-jp-v4 8B (Stage1)	86.75
b. LLM-jp-v4 8B (Stage 2 + GRPO)	92.57
c. LLM-jp-v4 8B (Stage 2 + GRPO) + リランキング	93.98
d. LLM-jp-v4 10B(Stage1)	89.96
e. LLM-jp-v4 8B (ロジットアンサンブル)	90.96
f. 提出システム	91.16
g. LLM-jp-v4 8B (Stage1, 思考過程なし)	78.31
h. gpt-oss-120b	92.57



複数のモデルを組み合わせた提出システムは単一モデルに劣る…

システムの限界 1

Sympyに依存しているため数学的には等価だが不自然な解答を出力する場合がある

問題: 複素数平面上の点 $z = 1 + 2i$ を原点を中心に反時計回りに $\pi/6$ 回転させた点を求めなさい。

コード:

```
from sympy import I, exp, latex, pi, simplify
z = 1 + 2 * I
theta = pi / 6
z_rot = simplify(z * exp(I * theta))
latex_answer = latex(z_rot)
print(f"Answer: {latex_answer}")
```

出力: $\sqrt[6]{-1} (1 + 2i)$

システムの限界 2

Sympyに依存しているため数学的には等価だが不自然な解答を出力する場合がある

問題: $\frac{1}{x}$ の不定積分を求めよ。ただし、積分定数を C とする。

コード:

```
from sympy import integrate, latex, symbols
x = symbols("x")
C = symbols("C")
F = integrate(1 / x, x) + C
latex_answer = latex(F)
print(f"Answer: {latex_answer}")
```

出力: $C + \log(x)$

まとめ

- gpt-oss-120bの知識蒸留により， LLM-jp v4を基盤とした解法コード生成モデルを学習
- 単体モデルでは， 教師モデルであるgpt-oss-120bと同等以上の性能を達成
- 複数モデルの組み合わせ方には改善の余地がある
- 本システムにはSymPyに強く依存しているため， 表現力には限界がある