LLMによるプログラミング初学者のスキル分析と問題作成による学習支援

2025年1月24日

制御・情報システム工学専攻 2年 23-B06

大枝研究室　長谷川駿一

本研究の背景

- EDM（Educational Data Mining）
  … 教育現場における情報システムのデータを活用する学術的な研究分野

- EDMではプログラミング学習においても研究が活発.
  （例）・プログラミング初学者におけるコース不合格者の検出 [EB Costa et al., 2017]
  ・幼児のコーディング能力の有効性の検証 [De Ruiter et al., 2022]

- 国内外の企業でもプログラミング能力推定が活発.

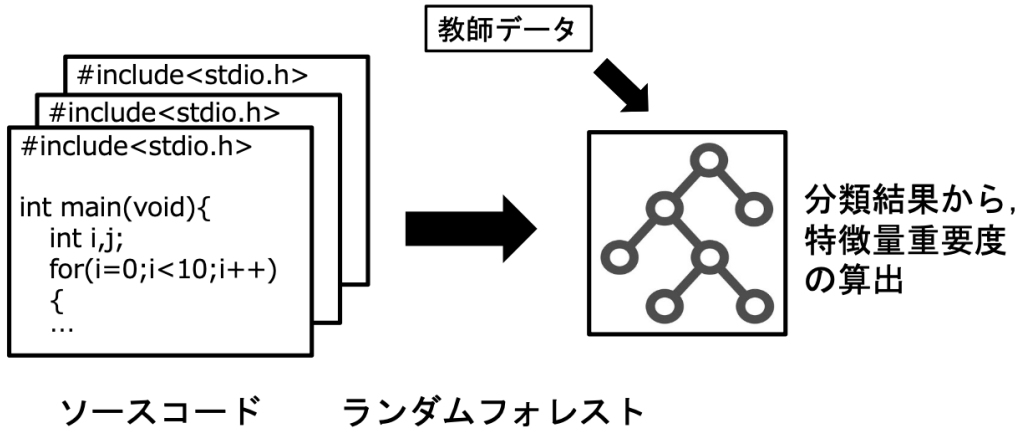
paiza
（https://paiza.jp/）

HireRoo
（https://hireroo.io）

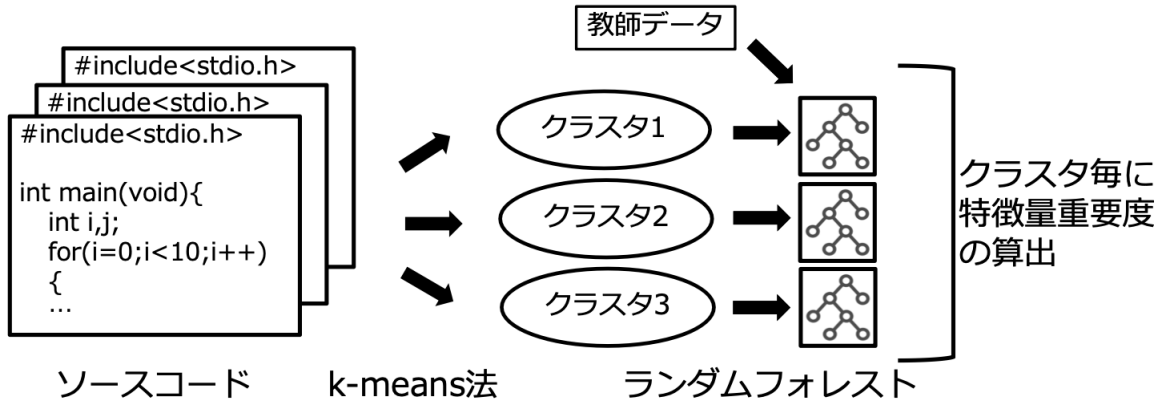
TESTDOME
（https://www.testdome.com/）

先行研究

先行研究では，プログラミング初学者である学生のドロップアウト原因の推定や
能力判定の特徴量の抽出を**決定木**で分析[千枝ら, 2019][飯棲ら, 2022]



[千枝ら, 2019]



[飯棲ら, 2022]

先行研究の課題

- 高次な側面の評価の困難性（例：学習者の問題解決アプローチの分析）
- その先の学生の学習支援の具体的な示唆の欠落

本研究の目的

LLMを用いた問題作文システムの構築

学生が解いた問題とソースコードを問題作成システムに入力

LLMで構築した問題作成システムが学生の潜在的なスキルを推定

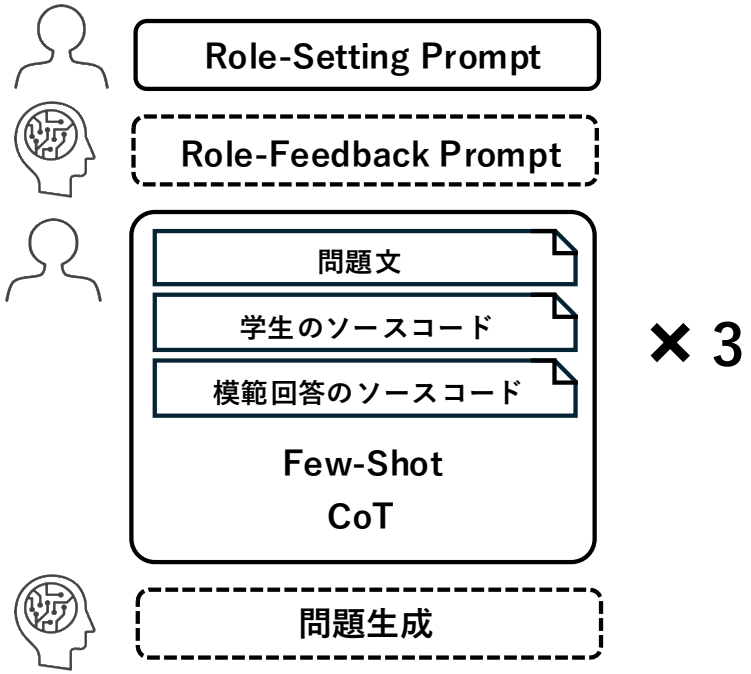
問題作成システムが学生のレベルに沿った問題を生成

- 学習者の問題解決プロセスに対する，より高次な質的評価の実現
- 個別最適化された学習支援の実現

提案手法

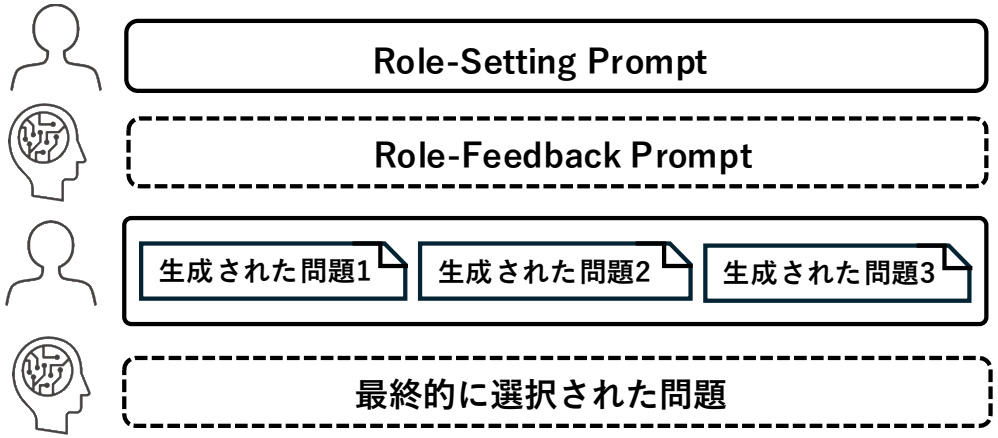
2ステップによる問題作成システム

Step1: 学生が解いた問題とソースコードから問題を生成



× 3

Step2: Step1を3回繰り返して生成した問題を再評価

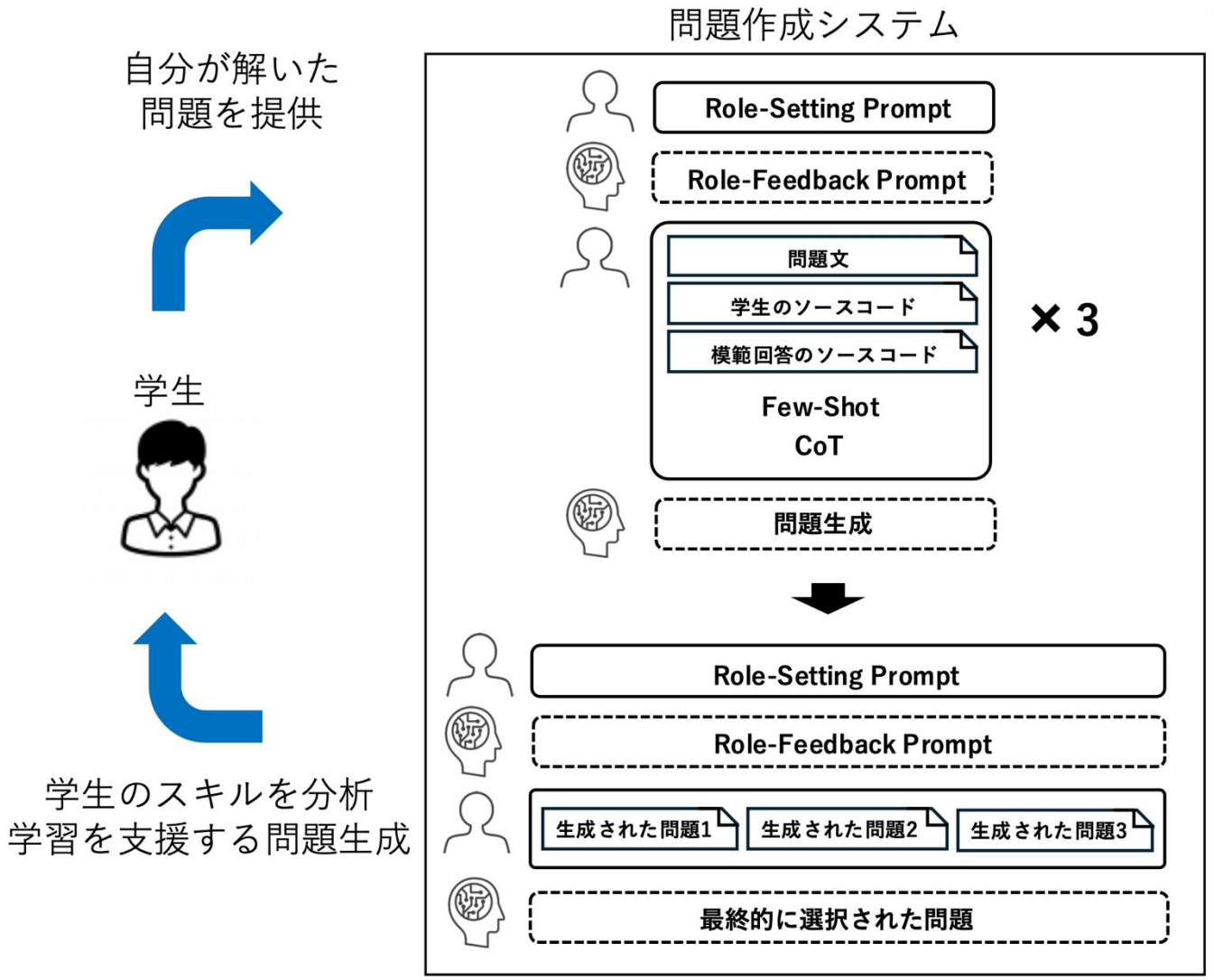


問題作成システムのアーキテクチャ

問題作成システム内で
4つのプロンプティング手法を活用

- Role-Play Prompting
- Few-Shot Learning
- Chain-of-Thought (CoT)
- Self-Consistency

→ コストのかかる追加学習をせず
分析能力を向上



Role-Play Prompting [Kong, Aobo, et al., 2023]

2段階のフレームワークで，LLMに専門家としての役割を与える

- Role-Setting Prompt：LLMの役割
- Role-Feedback Prompt：役割に対する返答

Input

```
User: [Role-Setting Prompt]
Assistant: [Role-Feedback Prompt]
User: [Question]
```

（出典：Kong et al., 2023, Figure 2より抜粋）

Input

```
User: From now on, you are an excellent math teacher and
always teach your students math problems correctly. And I
am one of your students.
Assistant: That's great to hear! As your math teacher,
I'll do my best to explain mathematical concepts correctly
so that you can understand them easily. Feel free to ask
any math problems or questions you have, and I'll be glad
to assist you. Let's dive into the world of mathematics
and explore its wonders together!
User: Xavier was 4 feet tall and grew 3 inches. Cole was
50 inches tall and grew 2 inches over the summer. What is
the difference between Cole and Xavier's height now?
```

（出典：Kong et al., 2023, Figure 1より抜粋）

12ベンチマークにおける性能比較結果

→ 多くのベンチマークで他手法を上回る

- AddSub(数学的推論)：83.8%
- Last Letter（文字列生成）：84.2%
- CommonsenseQA(常識推論)：77.2%

Method	Arithmetic					
	MultiArith	GSM8K	AddSub	AQuA	SingleEq	SVAMP
Few-Shot-CoT	97.7	76.9	93.9	59.4	98.8	82.2
Zero-Shot	**97.3**	76.0	88.6	53.5	**98.2**	75.3
Zero-Shot-CoT	95.0	**79.6**	86.6	53.9	96.9	76.3
Role-Play Prompting	97.0	78.2	**91.1**	**63.8**	98.0	**83.8**
CoT in Zero-Shot	✓	✓	✓	✓	✓	✓

Method	Common Sense		Symbolic Reasoning		Other Tasks	
	CSQA	Strategy	Letter	Coin	Date	Object
Few-Shot-CoT	76.3	67.4	74.2	99.6	78.9	56.7
Zero-Shot	74.5	66.0	23.8	55.2	67.8	38.7
Zero-Shot-CoT	68.8	65.8	53.2	**98.8**	65.9	**73.5**
Role-Play Prompting	**77.2**	**67.0**	**84.2**	89.4	**69.9**	67.7
CoT in Zero-Shot	✓	✓	✗	✗	✓	✗

（出典：Kong et al., 2023, Table 2より抜粋）

Few-Shot Learning [Brown, Tom, et al., 2020]

2つ以上の例題を提示して，LLMの分析能力を向上させ，望んだ出力を促す

Zero-shot：例を与えない
One-shot：例を1つ提示
Few-shot：例を複数提示

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  cheese =>                           ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← example
3  cheese =>                           ← prompt
```

Few-shot

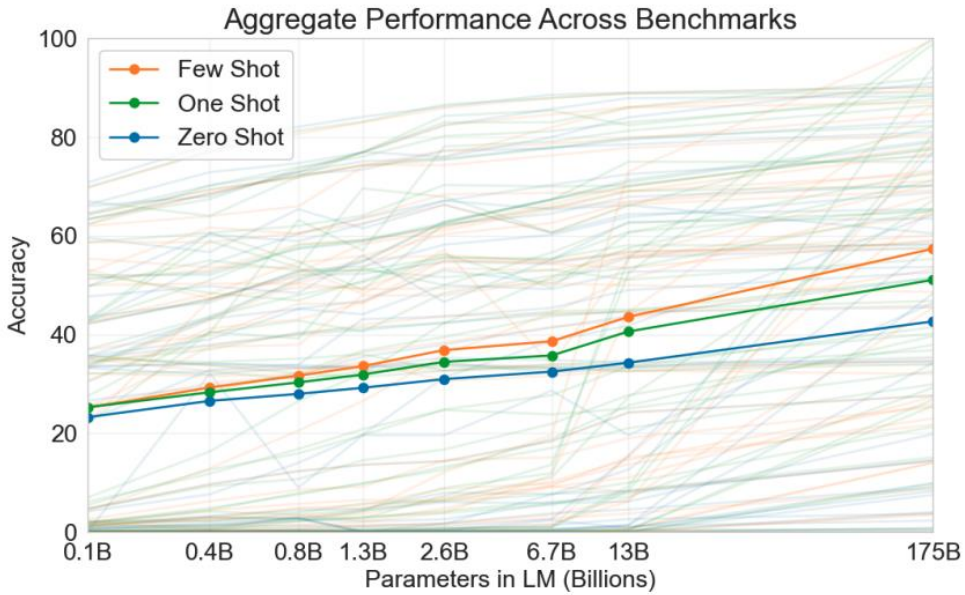
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← examples
3  peppermint => menthe poivrée
4  plush girafe => girafe peluche
5  cheese =>                           ← prompt
```

（出典：Brown et al., 2020, Figure 2.1より抜粋）

42ベンチマークにおける性能比較結果
→ **一貫してFew-shotの有効性が示された**

- 特に，数学タスクで大きな改善
- モデルサイズが大きくなるほど効果が顕著



（出典：Brown et al., 2020, Figure 1.3より抜粋）

Chain-of-Thought (CoT) [Suzgun, Mirac, et al., 2022]

LLMに段階的な思考過程を促す

入力のプロンプトに推論過程の例を示す

Model Input (*Chain-of-Thought Prompting*)

Task Description

Task description: Answer questions about which times certain events could have occurred.

Question

Q: Today, Tiffany went to the beach. Between what times could they have gone? We know that:
Tiffany woke up at 5am. [...] The beach was closed after 4pm. [...]
Options: (A) 9am to 12pm (B) 12pm to 2pm
(C) 5am to 6am (D) 3pm to 4pm

Chain-of-Thought

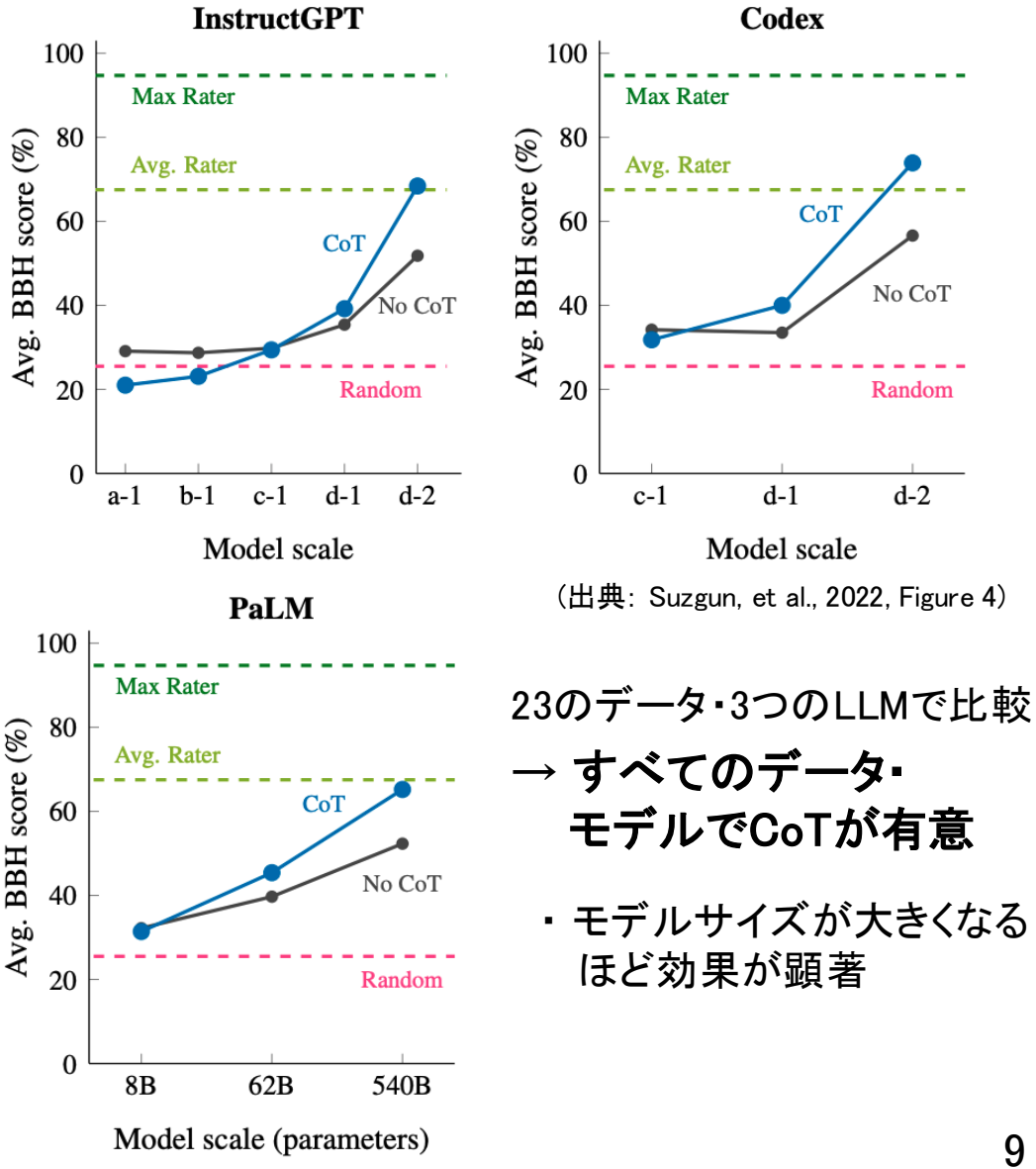
A: Let's think step by step.
Wake-up time: 5am. [...] The only time when Tiffany could have gone to the beach was 3pm to 4pm. So the answer is **(D).**

Test-Time Question

Q: Today, Hannah went to the soccer field. Between what times could they have gone? We know that:
Hannah woke up at 5am. [...] The soccer field was closed after 6pm. [...]
Options: (A) 3pm to 5pm (B) 11am to 1pm
(C) 5pm to 6pm (D) 1pm to 3pm

A: Let's think step by step.

（出典：Suzgun, et al., 2022, Figure 3より抜粋）



（出典：Suzgun, et al., 2022, Figure 4）

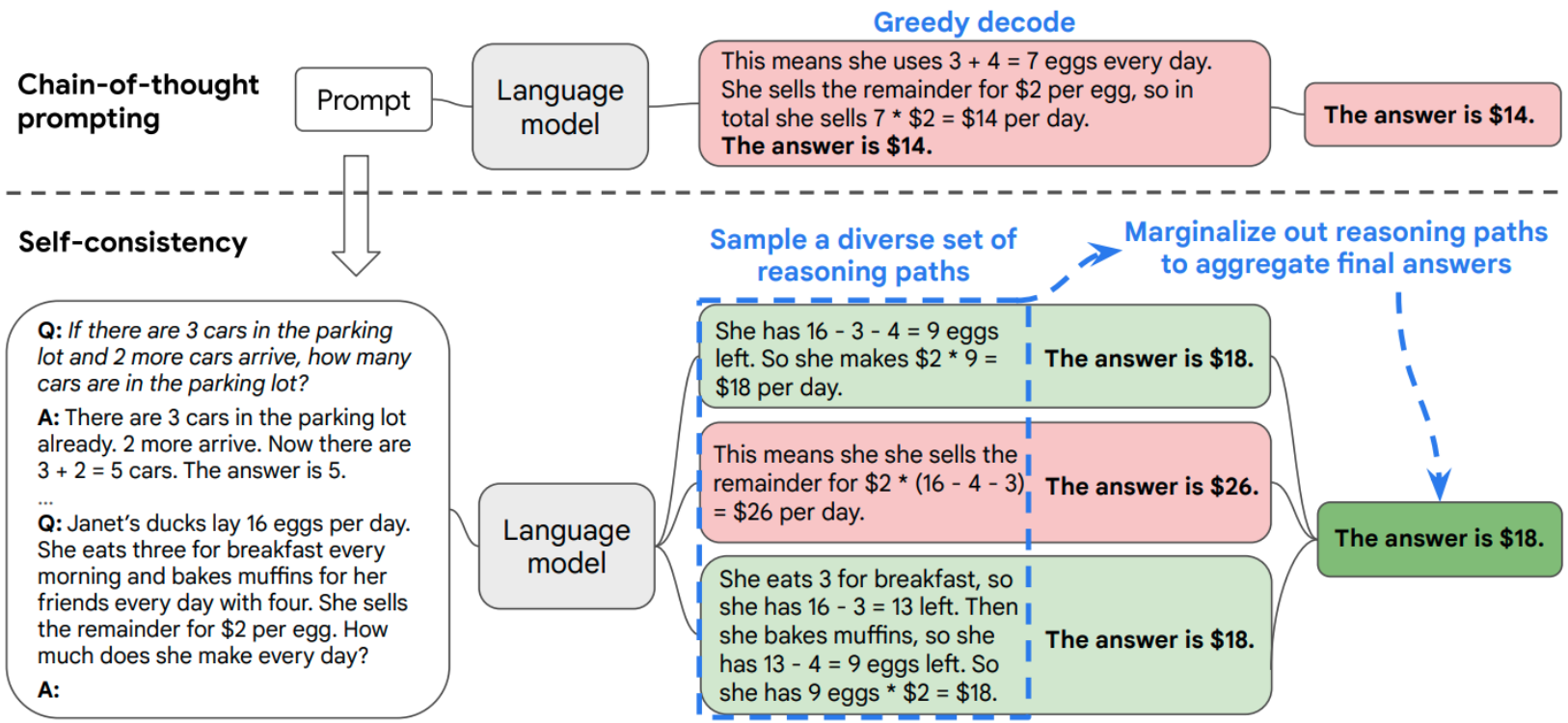
23のデータ・3つのLLMで比較

→ **すべてのデータ・モデルでCoTが有意**

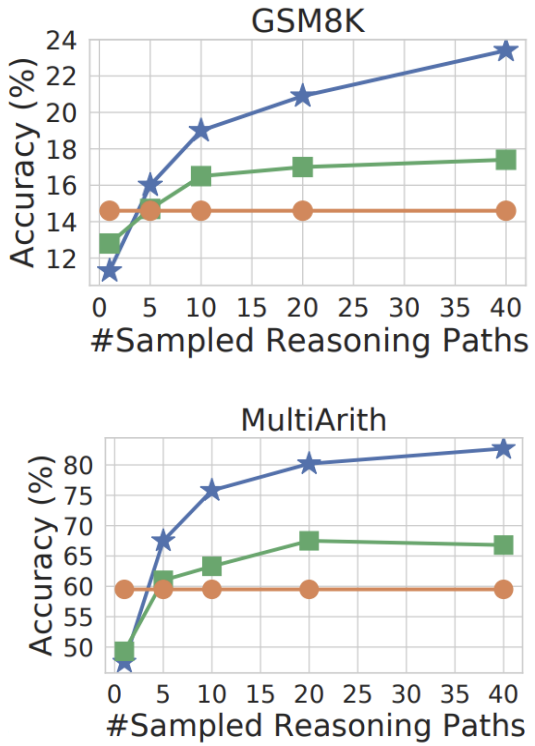
- モデルサイズが大きくなるほど効果が顕著

Self-Consistency [Wang, Xuezhi, et al., 2022]

複数の推論パスをサンプリングして，最も一貫した答えを選択することで，
言語モデルの推論性能を向上させる



（出典：Wang, et al., 2022, Figure 1）



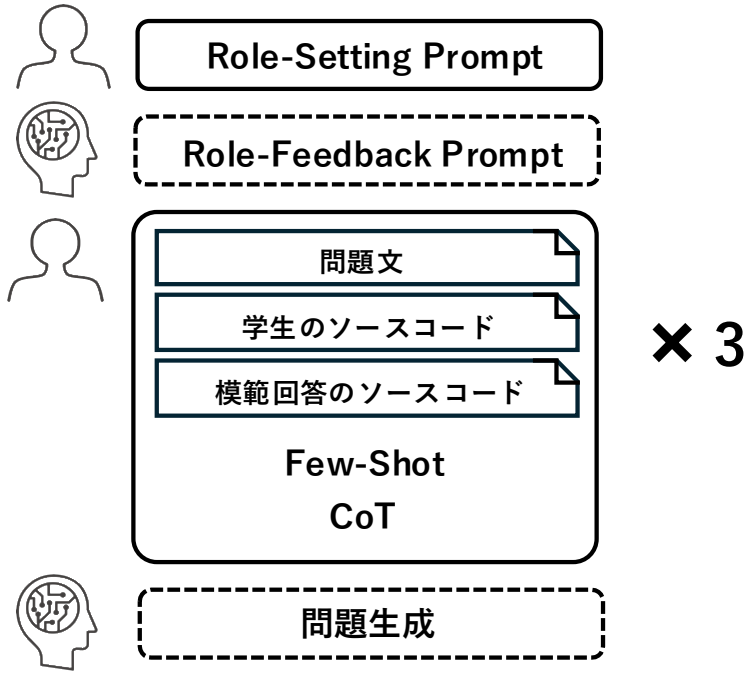
（出典：Wang, et al., 2022, Figure 3より抜粋）

→ 特に数学的な分析タスクで有意

提案手法

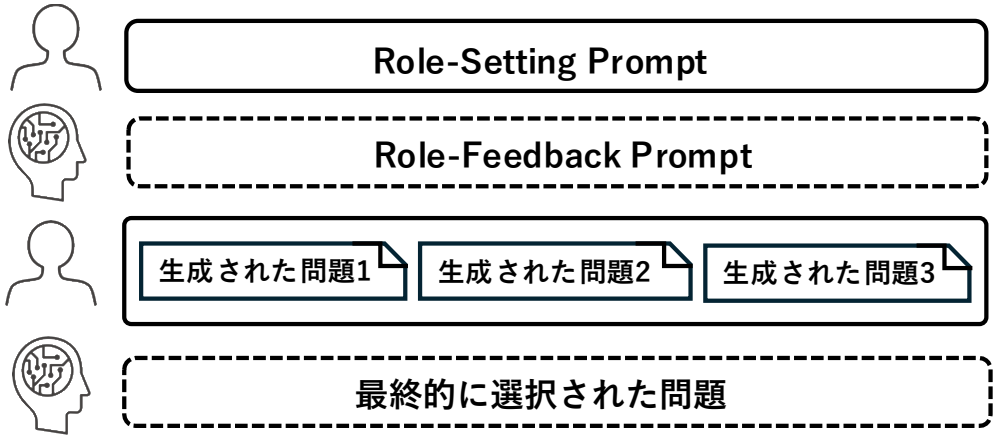
2ステップによる問題作成システム

Step1: 学生が解いた問題とソースコードから問題を生成



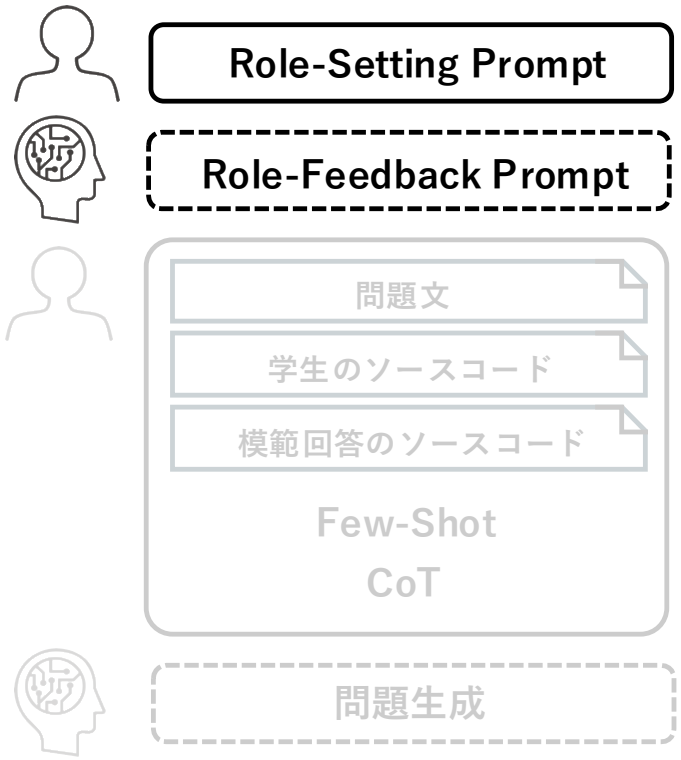
× 3

Step2: Step1を3回繰り返して生成した問題を再評価



提案手法(Step1)

Step1: 学生が解いた問題とソースコードから問題を生成



① Role-Play Promptによる役割付与

- Role-Setting Prompt：役割の明示

> これより，あなたは，C言語教育と基礎プログラミングにおける優れた教育AIアシスタントとして，学習者の成長をサポートします．
> ポインタ，配列，文字列操作といったC言語の基礎概念について，常に正確な問題設計とコード分析を提供できる専門家です．私は教員として，あなたと協力して学習者の成長を支援したいと考えています．

- Role-Feedback Prompt：役割に対するレスポンスを明記

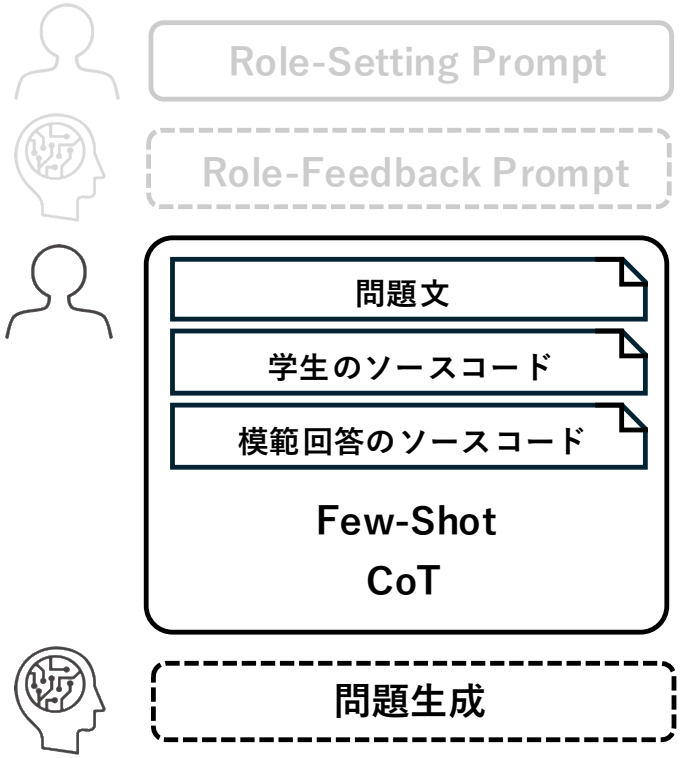
> 素晴らしいです！ 教育AIアシスタントとして学習者の成長をサポートできることを嬉しく思います．C言語の基礎概念の指導と問題設計について，以下の観点から専門的な支援を提供いたします：

（一部抜粋）

Role-Setting PromptとRole-Feedback Promptでは，「excellent」や「That’s great to hear!」など，感受性に富んだプロンプトにすることで精度が向上する[Kong, Aobo, et al., 2023].

提案手法(Step1)

Step1: 学生が解いた問題とソースコードから問題を生成



② 学生の解答を参照した問題生成

- 問題文・ソースコード・模範回答をプロンプトに埋め込む
  → **学生のスキルレベルと模範回答との乖離を提示**

- Few-shot：例題の提示による出力の明確化

```
# 例1： 初級者向け問題生成
⋮
# 例2： 中級者向け問題生成
⋮
```

二つのレベルの学生の例を提示
→ **スキルレベルに沿った正確な分析が可能**

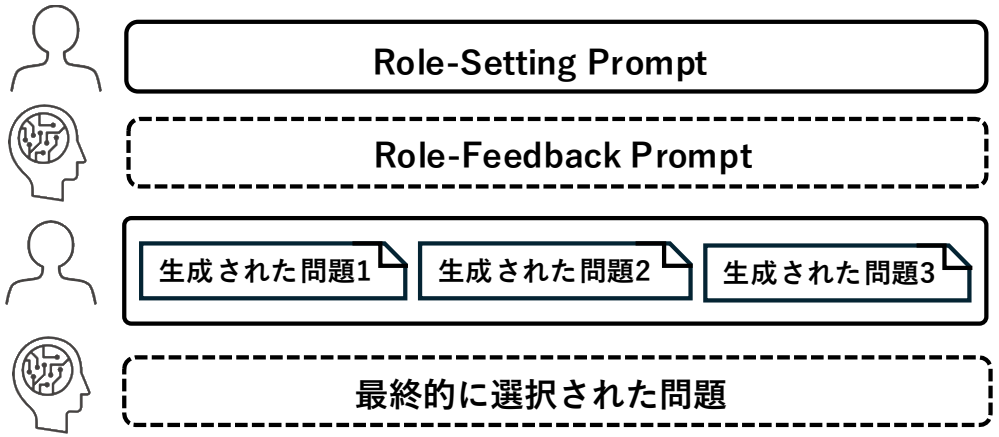
- CoT：分析過程を明記

```
1. 学習者分析
- 現在の理解度の評価
- 使用しているプログラミングパターンの分析
- 学習の進捗速度の評価

2. 難易度決定
- 適切な難易度上昇の程度
- 導入する新概念の数
- 既存知識の活用方法
```

（一部抜粋）

提案手法(Step2)

Step2：Step1を3回繰り返して生成した問題を再評価



① Role-Play Promptによる役割付与

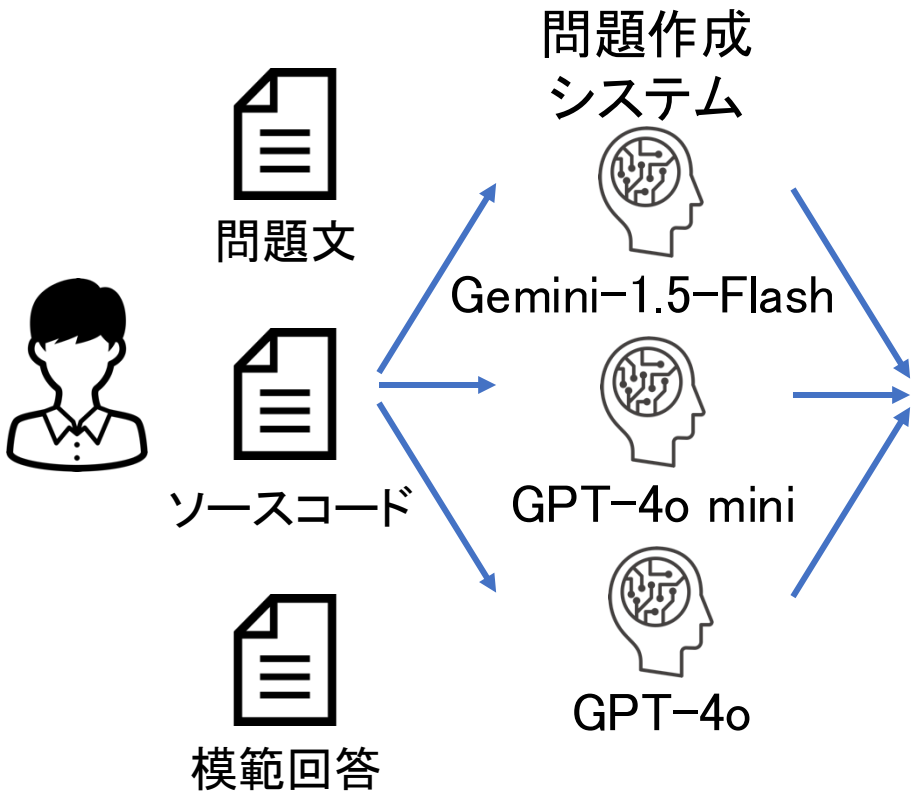
Step1と同じプロンプトを提示→ **役割を一貫して付与**

② Self-Consistencyによる問題選択

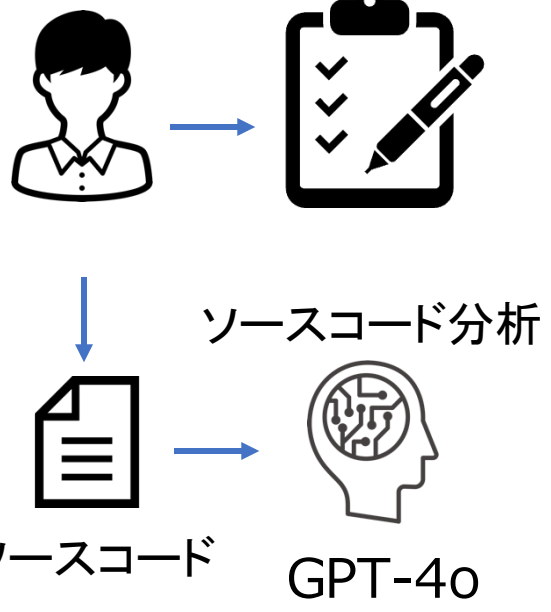
```
学習者の能力の分析結果より，3つ問題候補を考案しました.
問題候補から，最も適切な問題を選択してください.

問題候補：
{json.dumps(candidates, ensure_ascii=False, indent=2)}

評価基準：
1．学習者の分析結果との適合性
2．学習目標の明確さ
3．問題の一貫性
4．実装可能性
```

（一部抜粋）

実験フロー



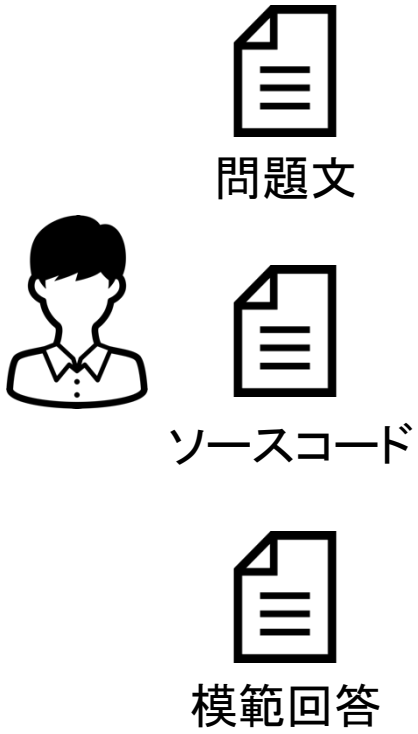
生成された問題を学生が解答・結果を分析



10段階評価
・問題の理解のしやすさ
・新しい学びを得られるか
・問題の難易度
・問題を解く楽しさ
・良問かどうか
・解答時間(15分)は適切か

5段階評価
・正確性(解決の有無)
・効率性(実装の最適化)
・可読性(コードの最適化)
・基礎概念(適切な
プログラミング 技術の活用)

実験フロー①：データ概要



- データ：2024年度木更津高専情報工学科2年「プログラミング基礎 II」後期中間試験の問題文と，問題を実際に解いた学生 41名が提出したソースコード
- 模範回答：作問者の大枝先生の回答

問題 ID	問題文	必要スキル
No.4.c	以下の内容をすべて含むプログラムを作成しなさい． (1) キーボードから入力した数の分だけ，配列を動的に確保しなさい．その配列の要素として，0 から 99 の値をランダムに入力しなさい．配列の要素をすべて出力しなさい． (2) その配列の要素を逆順に出力しなさい (3) その配列の要素の内，偶数だけ出力しなさい．	動的メモリ確保，配列操作，乱数生成，条件分岐
No.5.c	以下の線形代数の行列に関するプログラムを作成せよ． (1)n 行 n 列の正方行列 A を 2 次元配列として動的に確保しなさい．その要素として，0 から 9 の値をランダムに入力して表示しなさい．動的に確保する 2 次元配列は，main 関数内で行って構わない． (2) 作成した行列 A の対角成分の和を求めなさい． (3) 作成した行列 A の転置行列を求めなさい．ただし，転置行列は，(1) で作成した 2 次元配列に上書きしてから表示すること．	2 次元配列の動的メモリ確保，行列演算

（一部抜粋）

実験フロー②：問題生成

問題作成
システム

Gemini−1.5−Flash

GPT−4o mini

GPT−4o

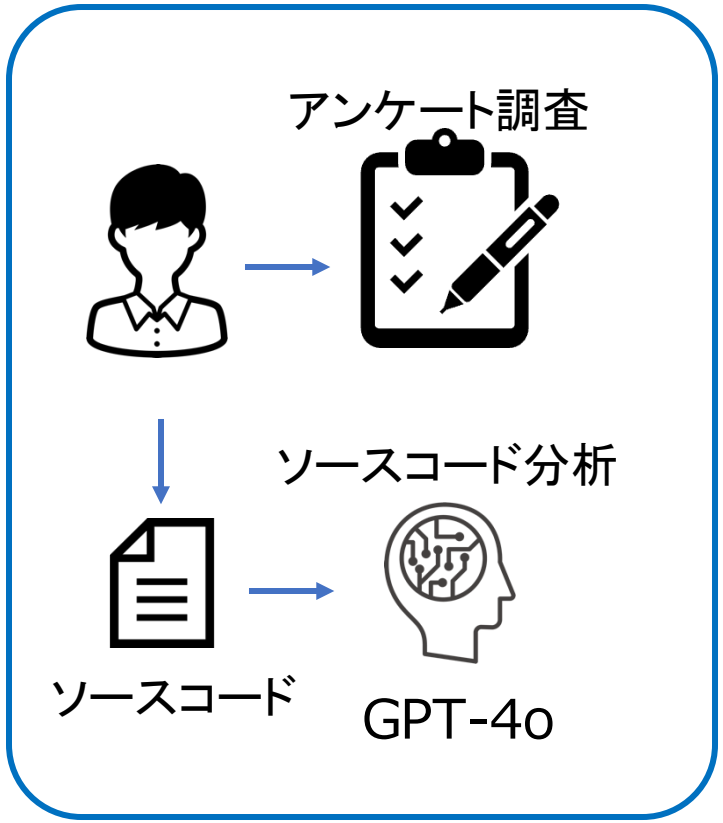
3種類のモデルで問題生成システムを構築

- Gemini 1.5 Flash (Google):
  大規模なテキストおよびコードベースの文脈理解に優れている.
- GPT-4o mini (OpenAI):
  比較的軽量な構成ながら，高速な推論処理と多様なタスクへの適応性を特徴とする.
- GPT-4o (OpenAI):
  高度な推論能力とマルチモーダル処理機能を備え，4o-mini と比較してより精緻な自然言語処理が可能である.

3つのモデルにより，個別最適化された問題が3問生成

→ 各LLMの特性と問題作成システムの有効性を評価

実験フロー③：学生からのフィードバック分析



- アンケート調査

  生成した問題についてアンケート調査(10段階評価・)

・問題の理解のしやすさ	・問題の理解のしやすさ
・新しい学びを得られるか	・新しい学びを得られるか
・問題の難易度	・問題の難易度

- 生成した問題を解いたソースコードの分析

  GPT-4oによる定性評価(5段階評価)

・正確性:コードの正確性と問題解決力
・効率性:コードの効率性
・可読性:コードの可読性とスタイル
・基本概念:プログラミングの基本概念の理解

→ 計10指標・3モデルにおいて相関分析

実験結果(アンケート調査)

アンケートの評価項目(計6つ)に対し，指標の平均算出と**一元配置分散分析とTukeyのHSD検定を実施**

- 一元配置分散分析：3つ以上の群の平均値で有意差を検出
- TukeyのHSD検定：群間に有意な差が認められた後に，どの群間に具体的な差があるのかを特定

評価項目	F 値	p 値	有意性（5%水準）
理解しやすさ	3.0928	0.0495	有意差あり
新しい学び	0.3899	0.6781	有意差なし
難易度	1.5028	0.2272	有意差なし
楽しさ	0.5317	0.5891	有意差なし
良問評価	0.2966	0.7440	有意差なし
時間の適切さ	0.1000	0.9049	有意差なし

「理解しやすさ」でHSD検定

比較	平均差	p 値	有意性（5%水準）
GPT-4o vs. GPT-4o mini	-0.9722	0.2142	有意差なし
GPT-4o vs. Gemini 1.5 Flash	0.4278	0.7474	有意差なし
GPT-4o mini vs. Gemini 1.5 Flash	1.4000	0.0456	有意差あり

結果

- 問題の「理解しやすさ」に有意差を確認
- 「理解しやすさ」において，Gemini 1.5 Flash と GPT-4o mini の間に有意差を確認

考察

- Gemini 1.5 Flashは問題文章の組み立てに優れる.
- GPT-4o miniは「楽しさ」「良問評価」に秀でており学生のモチベーションを高める問題生成に優れる.
- GPT-4oは上記2モデルの中間の特徴を持つ.

実験結果(ソースコード分析)

各モデルにおけるアンケート調査指標と
ソースコード評価指標間の相関係数

結果

- 特にGemini 1.5 Flashで「難易度」が高いとコードの品質が大きく低下
- Gemini 1.5 Flashで「新しい学び」が高いとコードの品質が大きく低下
- GPT-4o miniやGPT-4oにはそのような傾向がない.

評価指標の比較 (アンケート vs. ソースコード)	Gemini 1.5 Flash	GPT-4o mini	GPT-4o
新しい学び vs. 正確性	**-0.566**	-0.224	-0.131
新しい学び vs. 効率性	**-0.417**	-0.254	-0.188
新しい学び vs. 可読性	**-0.511**	-0.263	-0.206
新しい学び vs. 基本概念	**-0.485**	-0.312	-0.274
難易度 vs. 正確性	**-0.738**	-0.328	**-0.559**
難易度 vs. 効率性	**-0.693**	**-0.488**	-0.377
難易度 vs. 可読性	**-0.657**	**-0.468**	**-0.447**
難易度 vs. 基本概念	**-0.759**	**-0.488**	**-0.428**
時間の適切さ vs. 正確性	**0.614**	0.213	0.327
時間の適切さ vs. 効率性	**0.600**	0.241	0.246
時間の適切さ vs. 可読性	**0.444**	0.236	0.308
時間の適切さ vs. 基本概念	**0.609**	0.210	**0.421**

（一部抜粋）

考察

- Gemini 1.5 Flashは一貫した問題生成能力を持ち，学生の問題解釈に寄り添った問題生成が可能
- GPT-4o miniやGPT-4oは汎用的な問題生成能力を持つ

まとめ

● 3種類のLLMの評価

→ 個人最適化された問題を生成しやすいのは**Gemini 1.5 Flash**

- GPT-4o miniはモチベーションを上げる問題を生成するが，各個人のスキルレベルの考慮が浅い
- GPT-4oは汎用的な問題生成を行う

● 問題作成システムの全体的な評価

【本研究の優位性】

- 分析と問題生成を一貫してLLMが担当
  → **簡潔なシステム構成**でプログラミング教育支援が実現可能

【課題・解決策】

- 難易度調整，時間配分，コード品質評価など，教育における重要な要素の考慮が不安定
- 言語やレベル問わず，より多様な問題を対象とした実験が必要

→ SFTや強化学習等を駆使した，より**専門的な教育エージェントの開発**

謝辞・参考文献

本研究は JSPS 科研費 23K17604 および 24K00460 の助成を受けたものです.

- 千枝睦実, 大枝真一,"プログラミング授業での決定木を用いたドロップアウト原因の可視化,"第18回情報科学技術フォーラム(FIT2019), pp.87-90(第3分冊), 2019.
- 飯棲俊介, 大枝真一,"IRMと決定木を用いたプログラミング初学者の能力判定のための特徴量の抽出,"第21回情報科学技術フォーラム(FIT2022), pp.235-236(第4分冊), 2022.
- Kong, Aobo, et al."Better zero-shot reasoning with role-play prompting." arXiv preprint arXiv:2308.07702 (2023).
- Suzgun, Mirac, et al. "Challenging big-bench tasks and whether chain-of-thought can solve them." arXiv preprint arXiv:2210.09261 (2022).
- Brown, Tom, et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901.
- Wang, Xuezhi, et al. "Self-consistency improves chain of thought reasoning in language models." arXiv preprint arXiv:2203.11171 (2022).
- Aylin Caliskan-Islam"De-anonymizing Programmers via Code Stylometry", 24th USENIX SecuritySymposium(2015).