

生成 AI を活用したプログラミング演習支援のための バーチャル TA の提案

佐々木 虎太郎 伊藤 恵

2022 年頃より生成 AI の活用が社会的に注目され、プログラミング教育においても利用されている。生成 AI を用いたプログラミング支援システムによって、学生だけでなく教員や TA (Teaching Assistant) も恩恵を受けている。しかし、それらのシステムではプログラムコードを生成し学生に提供されており、学習者の主体性が損なわれている。また、従来のプログラミング教育では、教員や TA は学生ごとにつまづきやすいポイントや理解ができていない概念を特定し、それに基づいた効果的な学生指導を行うのが難しい。本研究ではプログラミング支援システムの効果を高める手段の一つとして生成 AI を活用しバーチャル TA となるシステムを開発することで、教員や TA の業務負担を軽減させ効果的な学生指導を行いやすくすることを目的とする。

1 はじめに

近年、大学におけるプログラミング教育は大きな変革を遂げ、多様化している。多くの大学では、プログラミングが必修科目として導入され、情報工学やコンピュータサイエンスを専攻しない学生にも基礎的なプログラミング教育が提供されている。大学や高専等ではプログラミングなどの学習のために、PC を使った演習を伴う講義が多く導入されている。この演習講義は、多数の学生が個別に異なるタイミングで課題に取り組むため、学生が抱える多様な学習上の問題に対して教員がきめ細かく対応することが困難である [7][2]。そのため授業補助者として大学院生などの TA (Teaching Assistant) を配置することで、学生の学習状況の把握や個別指導を行う取り組みが多く採用されている [4]。特に、学生の学習能力や進捗状況には個人差があり、全ての学生に対して適切な支援を提供することは容易ではない。そのため、TA や教員は常に効率的な指導方法や効果的なサポート体制の

構築に努める必要がある。

2022 年頃より、生成 AI の活用が社会的に注目されている。桑田ら [3] は、ChatGPT^{†1} をプログラミングに利用する学生の形態を分類した。具体的には、学習者のプログラミング言語やアルゴリズムなどの詳細についての知識を必ずしも前提としない「仲介モデル」と、これらの知識を前提とする「支援モデル」に分類している。また、プログラミング演習のエラー解消に注目し、ChatGPT による支援の評価を実施した。その結果、多くの場合、ChatGPT は有益な情報を提供するが、場合によっては誤った情報を出力することが分かった。プログラミングを学習中の学生が利用すると誤解を招く可能性があるため、質問の方法などを工夫する必要がある。

小原ら [5] は、ChatGPT を Google Colab^{†2} に統合したプログラミング演習支援システム KOGI について、開発状況と初期の実践の報告を行った。2023 年前期のプログラミング演習における事例を紹介し、KOGI 導入により教員や TA への軽微なエラーに関する問い合わせが大きく減少したことを明らかにした。

近年、生成 AI が急速に進化し、最近では GPT-4o

A Proposal for a Virtual Teaching Assistant Utilizing Generative AI to Support Programming Exercises.

Kotaro Sasaki, 公立はこだて未来大学システム情報科学部情報アーキテクチャ学科, School of Systems Information Science and Department of Media Architecture, Future University Hakodate.

^{†1} <https://openai.com/chatgpt/>

^{†2} <https://colab.research.google.com/>

がリリースされた。Achiam, et al [1] によると GPT-4.0 モデルは GPT-3.5 モデルに比べて試験成績の面で専門性の高いものに強く、言語による正確さも性能が高い。コードの生成についても GPT-3.5 モデルよりも約 10 % 精度が高いため、このモデルを使用してプログラムの支援を行うと良い結果がでると思われる。

本研究では、システムの効果を高める手段の一つとして生成 AI を活用しバーチャル TA となるシステムを開発することで、プログラミング授業における教員や TA の業務負担を軽減させることを目指す。本稿においてバーチャル TA とは、プログラミング教育の場において教員や TA の役割を補完・支援するものである。具体的には、教員や TA の代わりに学生の質問に対応することを想定している。さらに、授業中に TA や教員がプログラムのどこを教えたかをリアルタイムで収集し、システムが特にどこに力を入れて教えるべきかを学習させる機能を組み込む。この機能により、システムは学生がつまづきやすいポイントや理解が難しい概念を特定し、それに基づいた効果的な学生指導を行うことが可能となる。また、既存のシステムではコードの生成が行われるが、本システムでは回答を制限し、学生に言葉でのヒントのみを提供するように設計する。これにより、学生は課題解決力を養い、より深い理解を得ることができる。また、ChatGPT のモデルは現在の最新モデルの GPT-4o を使用する予定である。

2 関連研究

2.1 生成 AI のプログラミングエラーへの対応支援に関する研究

桑田ら [3] によると、ChatGPT をプログラミングに利用する場合の利用法には、以下の 3 つのモデルがある。

生成 AI を使わない現在のプログラミングモデル

生成 AI を使わず、作成するプログラムの動作を理解していることを前提に、ドキュメントを参照しながらプログラムを作成する。

仲介モデル 学習者のプログラミング言語やアルゴリズムなどの詳細についての知識を必ずしも

前提とせず、ChatGPT と対話しながらプログラムを作成する。

支援モデル 学習者がプログラミング言語やアルゴリズムなどの知識を前提とし、ChatGPT と対話しながらプログラムを作成する。

「支援モデル」は「現在のプログラミングモデル」の発展であり、プログラミング作成者は ChatGPT の誤りに気づいて指摘する必要があると述べられている。さらに、プログラミング演習における Syntax Error, Name Error, Type Error の解消に注目して ChatGPT による支援の評価を実施した結果、エラーの情報のみでその他プログラムに関する情報の提供をせずに評価した場合でも概ね正しい指摘が得られることを確認したが、誤解を招く回答もあり、プログラミングを学習中の学生が利用すると誤解を招く可能性があることが指摘されている。

Phung et al. [6] は、GPT-4 を用いて、失敗したテストケースや修正情報を含むプロンプトを使用することで、質の高いプログラミングヒントを生成する手法、GPT4Hints-GPT3.5Val を開発した。この手法では、GPT-4 が「チューターモデル」としてヒントを生成し、その後 GPT-3.5 が「学生モデル」として生成されたヒントの有用性を自動的に評価する。具体的には、Python プログラミング入門の「基本的なアルゴリズム」、「データ正規表現」、「データ分析」の異なる領域のデータセットを用いて評価を行った。その結果、GPT4Hints-GPT3.5Val は全データセットで一貫して 90 % 以上の精度レベルを達成した。この精度は、「学生モデル」がヒントを利用してプログラムを修正できる割合を示しており、GPT-4 による「チューターモデル」が出したヒントを用いて、GPT-3.5 による「学生モデル」がそれぞれのデータセットに対して 90 % 以上の確率でプログラムを修正できたことを示している。

2.2 生成 AI を活用したプログラミング演習支援システムに関する研究

小原ら [5] は、ChatGPT を Google Colab に統合したプログラミング演習支援システム KOGI について、開発状況と初期の実践の報告を行った。KOGI

は、プログラミング学習においてエラー診断とヒント提供を中心に、専門用語の解説からコード修正支援、対話的な UI まで幅広い機能を提供し、Jupyter Notebook や Google Colab などの環境で実装され、学習者の理解とスキル向上をサポートする AI ツールである。小原らは、担当するプログラミング演習の講義で活用し、ユーザーの実行したコード、発生したエラー、ユーザーの入力、ChatGPT からの回答を全てログとして記録し、それらを分析し、学生の動向を見た。その結果、学生から TA へ質問される内容が単純なエラーに対する質問は減り、プログラミングする際のロジックの組み方や考え方への質問に変わった。KOGI 導入により軽微なエラーへの問い合わせが大きく減少したことを明らかにした。

3 提案システム

本研究では、プログラミング教育における教員や TA を対象に、業務負担の軽減と効果的な学生指導を目的とした、生成 AI を活用したバーチャル TA システムを提案する。

3.1 システムの概要

システムの全体像を図 1 に示す。図 1 の各部分に対応する具体的な機能は 3.2 節で説明する。このシステムは、OpenAI 社の ChatGPT の API を使用し、最新モデルである GPT-4 を利用する予定である。学習者と生成 AI のやり取りは、Chatbot のような対話型で行う。提案システムの動作は以下の通りである。まず、学習者が支援を必要とする内容を入力する。回答となるプログラムコードが生成されないようにするため、学生が入力したものに「あなたは一流のプログラマーです。コードは生成しないでください」という文章を自動的に追加し、送信する。また、教員や TA は授業中に学生がつまずきやすいポイントや理解が難しい概念を特定するために提案システムを使用し、教えた内容をデータベースにリアルタイムで保存し、生成 AI に学習させる。これにより、生成 AI が学生の理解度や学習状況に応じた適切なヒントや指導を提供できるようになる。

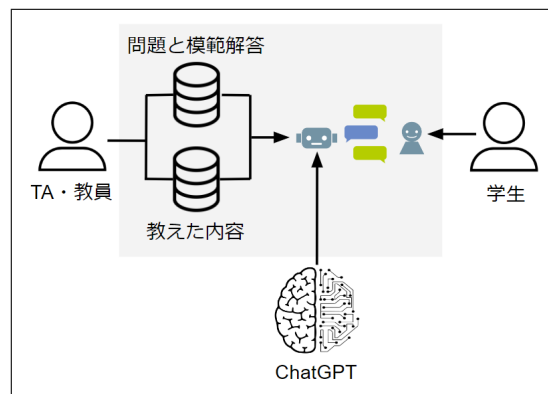


図 1 システム概要

3.2 システムの機能

提案システムは以下のような機能を持つ。

1. 問題と模範コードの登録機能

教員や TA が授業前に問題と模範コードをデータベースに登録できる機能。この機能により、授業で使用する問題と模範コードを一括で管理し、必要に応じて編集、追加、削除が可能である。

2. 授業中の指導内容の記録機能

教員や TA が授業中に学生に教えた内容をリアルタイムでデータベースに保存する機能。これにより、学生がつまずきやすいポイントを特定し、その情報を生成 AI に学習させることができる。

3. 学習者向け対話機能

学習者が生成 AI と対話しながら学習を進めるための Chatbot 機能。この Chatbot は、起動時に問題と模範コードのデータベースの値を生成 AI に学習させることで、迅速に問題の内容に応じた支援を提供できる。かつ、上記 2 の機能で記録された指導内容から学生がつまずきやすいポイントを学習することで、つまずきやすいポイントに特化した学習支援が可能になる。

3.3 回答の制限と効果

提案システムでは、学生が入力した質問内容に「あなたは一流のプログラマーです。コードは生成しないでください」というプロンプトを自動的に追加することで、生成 AI が直接的なコードを提供しないように制限している。これにより、学生が課題解決力を養うこ

```

You: 分からないのですが、コードを教えてください
Edit
Bot: おっけい、JAVAの素数判定プログラムのコードを以下に示します。

```java
import java.util.Scanner;

public class Main {
 // 素数判定関数
 public static boolean isPrime(int number) {
 // 1以下の数は素数ではない
 if (number <= 1) {
 return false;
 }
 // 2は素数

```

図 2 制限なしの出力結果

とができる。

制限しない場合、生成 AI は学生に直接的なプログラムコードを提供することが多い。これにより、学生は問題解決のための具体的なコードをすぐに得ることができるため、短期的には課題を迅速に解決することができる。しかし、この方法では学生が自ら考える機会が減少し、問題解決のプロセスを十分に経験できないため、深い理解や応用力の育成が阻害される可能性がある。学生が生成 AI に依存することで、自己解決能力や批判的思考力が低下し、長期的な学習効果が減少するリスクがある。

一方で、回答を制限することで、生成 AI は直接的なコードを提供せず、代わりに学生に対してヒントや方向性のみを示すようになる。これにより、学生は自らの考えで問題を解決する必要が生じるため、試行錯誤を通じて学習を深めることができると考えられる。このプロセスは、学生の論理的思考力や課題解決能力を向上させ、自律的な学習姿勢を育むのに効果的である。また、学生が自らの力で解決した経験は、自信を持つことにもつながり、さらなる学習意欲の向上が期待できる。

実際に、回答を制限しない場合と制限する場合の比較を行った結果を図 2 と図 3 に示す。

### 3.4 現在の開発状況と今後の開発予定

現在、以下の機能が実装されており、Web 上で操作できるようになっている。

1. 問題と模範コードのデータベースの編集・新規

```

You: 分からないのですが、コードを教えてください
Edit
Bot: プログラムは、与えられた数が素数かどうかを判定するものです。偶数や1以下の数は素数ではないという特性を利用しています。また、与えられた数が2で割り切れない場合、その数以下の奇数で割り切れるかどうかをチェックして素数かどうかを判定します。

```

図 3 制限ありの出力結果

イズとパス

図 4 問題と模範コードの入力画面

作成・削除 (図 4)。

2. 授業中の指導内容のリアルタイム保存 (図 5)。
3. 学習者向け画面の動作確認 (図 6, 図 7)。

今後の予定としては、バーチャル TA のプロンプト生成機能を強化し、実際の授業中に TA がどのように振る舞うかを再現することを目指す。そのために、TA が学生に対してどのように振る舞っているのかを把握するためのアンケート調査を実施し、授業中にどのような内容の質問を受けているのか、質問に対する TA の行動を調査する。また、調査結果を基に、バーチャル TA のフィードバック精度を向上させるための機能を追加開発する予定である。

## 4 演習時の質問対応調査

実際の演習で、TA が学生に対してどのように振る舞っているのかを収集するため、プログラミング教育を受講中の学生が、授業中にどのような内容を質問し、そして質問に対する TA がどのような行動をする

図 5 学生に教えた内容の入力画面

図 6 学習者向け画面

図 7 問題について聞いた結果

のかを調べるアンケートを実施した。

#### 4.1 調査目的

調査の目的は三つある。一つ目は、授業中に課題 1 から徐々に難しくなっていくが、特にどの段階での質問が多いのかを調査する。二つ目は、TA が学生の進捗状況や理解度に応じてどのように対応しているのかを調査すること。三つ目は、学生が授業中にどのような内容の質問が多いのかを調査する、アンケートは、Google フォームで作成し、回答してもらった。

#### 4.2 対象科目と調査対象者

アンケートの対象者は、著作らが所属する大学の学部 2 年前期必修科目で、Java プログラミングを扱っている情報処理演習 I(2024) の TA を行っている B4 以上の学生 18 名を対象にする。情報処理演習 I(2024) を受講している学生は 200 名である。今回対象としている演習科目では全 13 回の授業で、各回に 1 つまたは複数の課題が出される。これらの課題は段階的に難易度が上がるように設計されており、学習者は授業中に課題 1 から順次取り組んでいく。

#### 4.3 質問項目

質問項目は以下のように用意した

- 氏名
- 担当クラス
- 対応した課題番号
- 質問内容
- 教えた内容

基本情報を収集するため、氏名と担当クラスを質問項目に入れた。氏名選択欄は、TA の名前を選択式にしており、「その他」の項目を追加することで、教員の教えた内容も収集できるようにした。担当クラスも同様に選択式で、こちらも「その他」の項目を設け、教員や TA の担当外のクラスに対応できるようにしている。また、対応した課題番号も選択式にしており、これは 1 回の演習の複数の課題のうちの何番目かを選ぶ形式である。ここにも「その他」の項目を追加し、授業に関係のない質問に関してはこの項目で選別できるようにした。質問内容と教えた内容については自由記述とし、アンケートは複数回回答できるようにしている。

#### 4.4 結果

本稿の執筆時点では、39 件の回答が集まっている。学習者が授業前および授業中に、特にどの段階での質問が多いのかを調査した。結果は表 1 に示す。過半数以上が課題 1 への質問になり、課題 2 への質問はその他への質問と変わらない結果から、最初何もわからず TA に頼るが、課題 1 をクリアすると課題 2 は自力で解けるようになり、最初の理解を深めることで後の課題に大きく影響してくると考えられる。また、授業開始するとほとんどの学生は課題 1 から同時にスタートし、この結果からかなりの質問が最初に偏ることが分かり、そうすると TA が全員の質問対応していくと、最初に対応した学生と最後に対応した学生で進捗に差が出てしまうことになってしまおうと考えられる。それと共に、TA の業務負担もそこに集中するのではないかと考えられる。

TA が学生の進捗状況や理解度に応じてどのように対応しているのか、学生が授業中にどのような内容の質問が多いのかを調査した。質問内容は自由記述で収

集し、後に分類を行った。結果は表 2 に示す。質問内容として、「エラー内容の説明」と「アドバイスを求める」と「その他」に分類した。「エラー内容の説明」は「無限ループが起きてしまう」と具体的な内容で質問されていたが、「アドバイスを求める」はかなり抽象的で「正規表現の \ が二つある理由はなんですか」と用語の解説を求められている質問もあったが、「～が分からない」と質問されてコードを読んでいくとエラーが原因であることが分かり、実質「エラー内容の説明」に分類されるものもある。具体的には、学生が「出力結果が違いその理由がわからない」と質問した場合、自分のコードと見比べながら、return の位置と if の条件が足りないことを指摘し、なぜこの実装になるのか理由を説明する事例があった。このようなケースでは、初めは抽象的な質問として「アドバイスを求める」に分類されるが、コードの詳細を確認する過程で具体的なエラー内容の説明に繋がるため、実質的には「エラー内容の説明」に分類されることとなる。したがって「アドバイスを求める」という領域に「エラー内容の説明」という内容が存在しているということが考えられる。

また、TA の教えた内容についても以下の 5 種類に分類した。

**直接指示** 学生に対して具体的な解決方法やコードの書き方を直接伝えるものであり、例えばエラー内容から問題箇所を指摘することが含まれる。

**確認とフィードバック** 学生のコードを確認し、その内容をフィードバックするものであり、例えば自分が書いたコードを 1 行ずつ丁寧に説明しながら実装のヒントを与えることが含まれる。

**逆質問と考えさせる方法** 学生に対して質問を投げかけ、自ら考えさせる方法。

**リソース参照** 関連資料や参考文献を示すもの。

**教員に報告** 特定の問題を教員に報告する対応。分類した結果は表 3 に示す。

分類した結果、「直接指示」と「確認とフィードバック」の二つの項目が過半数以上を占めた。「直接指示」と比較して「確認とフィードバック」は、コードレビューから具体的なアドバイスまで行うため、質問した一人の学生に対して時間がかかると考えられる。

表 1 課題内容の分類と合計

課題 1	26
課題 2	7
その他	6
合計	39

表 2 質問内容の分類と合計

エラー内容の説明	10
アドバイスを求める	23
その他	6
合計	39

表 3 教えた内容の分類と合計

直接指示	21
確認とフィードバック	12
逆質問と考えさせる方法	2
リソース参照	2
教員に報告	2
合計	39

「直接指示」と「確認とフィードバック」と比較して少数ではあるが「逆質問と考えさせる方法」と「リソース参照」という項目がある。「直接指示」と「確認とフィードバック」は結果的に TA が学生に答えを教えてしまうが、これでは本研究の目的である効果的な学生指導が達成されないのではないかと考えられる。しかし、「逆質問と考えさせる方法」と「リソース参照」に関しては学生がどこまで理解しているのかを TA が理解し、それに合わせた用語やアルゴリズムの解説をすることによって直接の答えは教えずに、正解までの過程をヒントとして学生に示し、答えにたどりつけるようにしている。「逆質問と考えさせる方法」と「リソース参照」が増えることによって効果的な学生指導が達成されるのではないかと考えられる。この結果を踏まえて、学生の理解度に応じて適切なヒントやリソースを提供できるようにすることを検討している。具体的には、リソース参照機能を拡充し、関連する資料や参考文献を自動的に提案することで、学生が自ら学びを深めるためのサポートを行う。

## 5 まとめと今後の予定

本研究では、プログラミング教育における教員や TA を対象に、業務負担の軽減と効果的な学生指導を目的とした、生成 AI を活用したバーチャル TA システムの提案を行った。プログラミング教育を受講中の学生が、授業中にどのような内容を質問し、そして質問に対する TA がどのような行動をするのかを調査したアンケートの結果から、授業中に特に多くの質問が集中する段階や、学生がよく質問する内容について明らかになったため、提案システムで学生からの質問内容に応じて過去の授業資料を参照し、第何回の資料を見ればわかるといった具体的なガイドを提供できる機能の開発を進める。今後はこの機能を試作し、評価実験を行うことで、提案システムの効果について検証する予定である。

## 参考文献

- [1] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Al-tenschmidt, J., Altman, S., Anadkat, S., et al.: Automating Human Tutor-Style Programming Feedback: Leveraging GPT-4 Tutor Model for Hint Generation and GPT-3.5 Student Model for Hint Validation, *Proceedings of the 14th Learning Analytics and Knowledge Conference*, March 2024, pp. 12–23.
- [2] 加藤利康, 石川孝: プログラミング演習支援システムにおける学習状況把握機能の提案, 情報処理学会研究報告. コンピュータと教育研究会報告, Vol. 2013, No. 2, 2013, pp. 1–8.
- [3] 桑田喜隆, 石坂徹, 政谷好伸, 横山重俊, 浜元信州, 谷沢智史: 生成系 AI を活用した Jupyter Notebook のプログラミング演習支援に関する検討, 人工知能学会第二種研究会資料, Tokyo, 人工知能学会, 2023, pp. 06.
- [4] 文部科学省: ティーチング・アシスタントについて, [http://www.mext.go.jp/b\\_menu/shingi/chukyo/chukyo4/003/gijiroku/07011713/001/002.htm](http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo4/003/gijiroku/07011713/001/002.htm), 2013. Accessed: 2013-11-12.
- [5] 小原有以, 佐藤美唯, 倉光君郎: KOGI: ChatGPT を Colab に統合したプログラミング演習支援, *No Title*, 2023, pp. 141–148.
- [6] Phung, T., Pădurean, V. A., Singh, A., Brooks, C., Cambroner, J., Gulwani, S., ..., and Soares, G.: Automating human tutor-style programming feedback: Leveraging gpt-4 tutor model for hint generation and gpt-3.5 student model for hint validation, *Proceedings of the 14th Learning Analytics and Knowledge Conference*, March 2024, pp. 12–23.
- [7] 西澤泰彦: 多人数講義における問題点と教育方法, 名古屋高等教育研究, Vol. 6(2006).