

生成 AI を活用したデバッグ機構を通じて 誤りに気づかせるコーディング学習支援環境

Awareness-support environment for learning error debugging with generative AI

矢代 涼^{*1}, 松浦 健二^{*1}, 竹内 寛典^{*1}
Ryo YASHIRO^{*1}, Kenji MATSUURA^{*1}, Hironori TAKEUCHI^{*1}
^{*1}徳島大学
^{*1}Tokushima University
Email: c611901481@tokushima-u.ac.jp

あらまし: プログラミング演習などの集団授業において、誤りの解消を自動で支援するシステムの必要性が高まっている。しかし、従来のデバッグ支援ではコンパイルエラーの表示されない誤りに対応できないという問題があった。そこで本研究では、プログラムの解析やデバッグなどが行える生成 AI をデバッグ機構として活用することでこの問題に対応した。そして、修正済みのプログラムに学習者が誤りに気づきやすくなるような加工を施すことで、学習者のデバッグを支援するシステムを開発した。

キーワード: プログラミング初学者、デバッグ支援、生成 AI、ChatGPT

1. はじめに

2020 年から小学校でのプログラミング教育が必修化されており、プログラミング教育の重要度がさらに高まっている。

一方で、多くのプログラミングの授業が教員 1 対生徒多数でおこなわれ、教師が一人一人学習者の誤りを支援することは困難である。誤りに陥っている学習者に対して、自動的に誤りの解消を支援するシステムの必要性が高まっている。

2023 年頃から多くの企業によってリリースされている生成 AI は、テキストや画像の生成だけでなく、プログラムの生成、解析、デバッグなど多岐にわたるタスクにおいても優れた能力を発揮している。

本研究では生成 AI をデバッグ機構として活用して、初学者が誤りに陥った際に誤りがある場所の特定と修正を支援するシステムを開発する。

2. 初学者の支援に関する研究

後藤ら⁽¹⁾は、一斉授業時のデバッグ支援を円滑に行うために、シェルスクリプトを用いて、学習者のプログラムのコンパイルエラーを解析して、エラーに関する補助的なメッセージや修正済みのプログラムを自動で提示するデバッグ支援システムを開発した。しかし、この提案手法はエラーの表示されない誤りに対して支援が行えないという課題が存在する。

桑田ら⁽²⁾は Jupyter Notebook を活用したプログラミング演習に、学習者がエラーに陥った際に ChatGPT を利用して、素早くエラーの解消をする支援モデルを提案した。しかし、この提案手法は学習者が ChatGPT に対してプログラムの説明やプログラムの添付を行う必要があり、学習者の負担が大きくなってしまいう課題がある。また、ChatGPT から正しいプログラムが出力されても、学習者が自身のプログラムの誤りを見つける必要があり、デバッグの効率が悪くなる課題もある。

3. 生成 AI を活用したデバッグ支援機構

生成 AI とは与えられた入力や条件に基づいて新しいデータを生成する機械学習の分野で、文章や画像などのコンテンツを出力することが出来る人工知能である。生成 AI を使用すると文章や画像の生成だけでなく、プログラムがどのような処理を行うかを分析したり、任意のプログラムを生成したり、デバッグを行うことができる。これらの生成 AI の機能をデバッグ機構として活用すると、前章で述べた、エラーの表示されない不具合の支援を行うことができない、既存の学習支援システムの問題に対応できる。

3.1 使用するプログラミング言語

今回使用する ChatGPT は最も得意なプログラミング言語について公開されていないため、Web 上のドキュメントなども豊富にあり、初学者でも利用しやすい Python を使用することにした。

3.2 適応対象の誤り

深町⁽³⁾や Brown ら⁽⁴⁾の初学者が陥りやすい誤りをまとめた報告から、以下の 4 つのエラーが表示されない誤りを選定した。

1. 3 つ以上の条件を持つ if 文
2. if 文のネスト利用
3. 2 重 for 文
4. 配列を入れ替える際の index 操作

3.3 学習者に支援を行う方法

江木ら⁽⁵⁾はデバッグをする際の手順を示したデバッグ過程モデルを示した。本研究ではデバッグ過程モデルにおける仮説の段階において、デバッグに役立つ情報を提供することで学習者のデバッグを支援する。江木らのデバッグ過程モデルに本システムが行う支援を追記し、図 1 に示す。

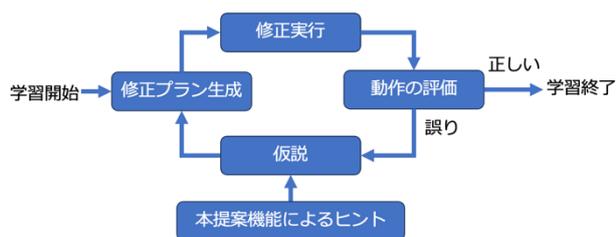


図1 支援を行うモデル

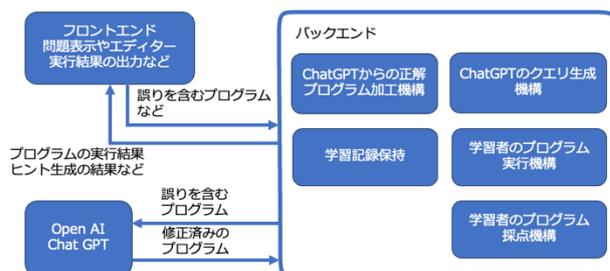


図2 システムの構成図

3.4 学習者に表示するヒント

ヒントは全部で3つのレベルに分かれており、学習者がヒントを求めると、レベル1からレベル3まで段階的に表示される。レベル3まで表示したあとは、レベル1に戻る。

ヒントレベル1 誤り箇所の高ライト表示

誤りがある行を赤色でハイライト表示することで、学習者にプログラム全体から誤りのある行を気づかせることが目的である。

ヒントレベル2 正解プログラムの選択表示

正解プログラムを1行ごとにマスクして、学習者が表示したい行を選択する。正解プログラムを提示することで、学習者に誤りのある行の中から詳細な誤りの場所を気づかせることが目的である。

ヒントレベル3 正解のプログラムを表示

最後に、正解のプログラムを提示して、学習者が答え合わせを行う。

4. 学習支援システムの設計及び開発

4.1 システムの概要

学習開始後、学習者は問題に沿ってプログラムの実装を行い、誤りに陥った際に、最初は学習者自身で修正を行う。学習者自身で修正ができない場合に、本研究提案機能のヒントを表示する。学習者は表示されたヒントを確認してプログラムの修正を行う。修正したプログラムを実行し、期待する動作ができていないかの確認を行う。すべてのテストケースを通過した場合に回答を提出して学習を終了する。

4.2 システムの構成

システムは Web ブラウザで稼働するフロントエンド部とサーバー側で稼働するバックエンド部の2つに分かれている。バックエンド部は学習者のプログラム実行機構、プログラム採点機構、生成 AI に問い合わせを行うクエリ生成機構、正解プログラム加工機構、学習者イベント記録機構などの複数の機構から構成されている。クライアント部では、問題やプログラムの実行結果の表示、ヒントの加工表示、プログラムの編集などが行えるようになっている。生成 AI との通信は、OpenAI によって提供されている ChatGPT の API を通して行われる。システムの構成図を図2に示した。

4.3 ヒントの実装

学習者がヒントを求めると、学習者のプログラムと問題の ID がバックエンド部へと送られる。バックエンド部のクエリ生成機構では、問題 ID から問題の説明文が取得され、学習者のプログラムと問題の説明文をあらかじめ作成しておいた生成 AI の出力形式を指定するテンプレートに当てはめることで、クエリが生成される。このクエリを生成 AI へと送信してデバッグを行い、生成 AI から返ってきた修正済みのプログラムと学習者の書いたプログラムを比較して、生成 AI の修正した箇所を特定する。2つのプログラムの差分を検出するために、Python 標準ライブラリの `difflib` を用いた。修正箇所の情報がフロントエンド部へと送られ、学習者に加工表示される。

5. おわりに

本研究では、生成 AI をデバッグ機構として活用し、エラーの表示されない誤りのデバッグを行い、修正内容から学習者が誤りに気づきやすくなるような加工を施すことで、誤りの発見と修正を支援するシステムを開発した。

今後の展望としてはプログラミング初学者を対象に、本システムの有用性を評価したいと考える。

参考文献

- (1) 後藤孔、藤中透: “プログラミング教育におけるデバッグ支援”、システム制御情報学会論文誌、pp. 249-255、(2019)
- (2) 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 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、(2023)
- (3) 深町 修一: “文系の学生に対するコンピュータプログラミング教育の一考察”、(2010)
- (4) Neil CC Brown、Amjad Altadmri: “Novice java programming mistakes: Large-scale data vs. educator beliefs”、ACM Transactions on Computing Education (TOCE)、pp. 1-21、(2017)
- (5) 江木鶴子、竹内章: “プログラミング初心者にトレースを指導するデバッグ支援システムの開発と評価”、日本教育工学会論文誌、pp. 369-381、(2009)