

一般情報教育における AI チャットボットによる プログラミング支援手法の検討

藤原 康宏†

兵庫医科大学 医学部†

1. はじめに

高等教育において、数理・データサイエンス・AI を適切に理解し、それを活用する基礎的な能力を育成することが求められ、多くの高等教育機関が教育プログラム認定制度に参加している。データ処理や機械学習を学ぶ際には、プログラミングの知識が必要であるが、大学入学時点で習得している者は少ない。今後、新課程の「情報 I」でプログラミングを経験した学生が入学してくるが、基礎的なプログラミング教育の必要性はなくなり、個々の学生の学習履歴に配慮した教育が必要となる。

初学者へのプログラミング教育の難しさは、従来から指摘されており、多くの調査が行われている。例えば、Lahtinen らは、500 人規模の国際的な調査を実施し、学生は自学 (studying by themselves) が、講義、演習、実習よりも有用と感じていると分析した。プログラミング学習は自律的に試行錯誤を繰り返す必要があるが、プログラミングの初学者には、エラーメッセージの解釈、人への質問を躊躇するなどの理由で、最初の段階で学習に行き詰まり、苦手意識を持つことも少なくない。

初学者のプログラミング時の試行錯誤を促進させるために、学習者が作成したプログラムに対する生成 AI が出力した助言の有効性について、初期段階の検討を行った。

2. 授業の概要

本システムは、著者が担当する医学部 1 年生を対象とする「医情報学実習」(1 単位、16 コマ、受講生 63 名程度、2 クラス開講)での利用を想定している。本授業の各回の学習内容のうち、プログラミングの実習に関する内容を表 1 に示す。本授業の目的は、情報化に対応できる医師として必要な基礎的なスキルの習得である。ただし、プログラミングのスキルの習得は目標ではなく、本授業の後半(第 9 回~14 回)で実施するデータ処理を行う手段として位置付けている。また、

Programming Support Method by AI Chatbot in General Information Education

†Yasuhiro Fujihara, School of Medicine, Hyogo Medical University

別の科目で、機械学習の実習でもプログラミングの知識が必要となる。

今回試作した、学習者のプログラムへの助言ツールは、表 1 のうち個別演習に取り組む際に使用され、課題の提示・提出の機能を含んでいる。受講生は、各受講生のコンピュータにインストールされた Jupyter Notebook または、Google Colaboratory を用いて、プログラムを実行し、行き詰ったときや、完成したときに、本システムにソースコードを貼り付ける。

表 1 授業内容

| 回 | 内容 | 時間 |
|--------|----------------|-------|
| 1, 2 | 基本事項の説明と操作練習 | 45 分 |
| | 個別演習(変数、入出力) | 30 分~ |
| 3, 4 | 小テスト(変数、入出力) | 10 分 |
| | 個別演習(条件分岐) | 30 分~ |
| 5, 6 | 小テスト(条件分岐) | 10 分 |
| | 個別演習(繰り返し) | 30 分~ |
| 7, 8 | 小テスト(繰り返し) | 10 分 |
| | 個別演習(配列、関数) | 30 分~ |
| 9, 10 | 小テスト(配列、関数) | 10 分 |
| | データ処理の説明と操作練習 | 65 分 |
| | 個別演習(回帰分析) | 45 分~ |
| 11, 12 | 個別演習(母平均の差の検定) | 45 分~ |
| | 個別演習(分割表の分析) | 45 分~ |
| 13, 14 | アルゴリズムの説明と操作練習 | 75 分 |
| | 個別演習(並び替え) | 45 分~ |

※プログラミングの実習に関する内容のみを記載

3. システムの概要

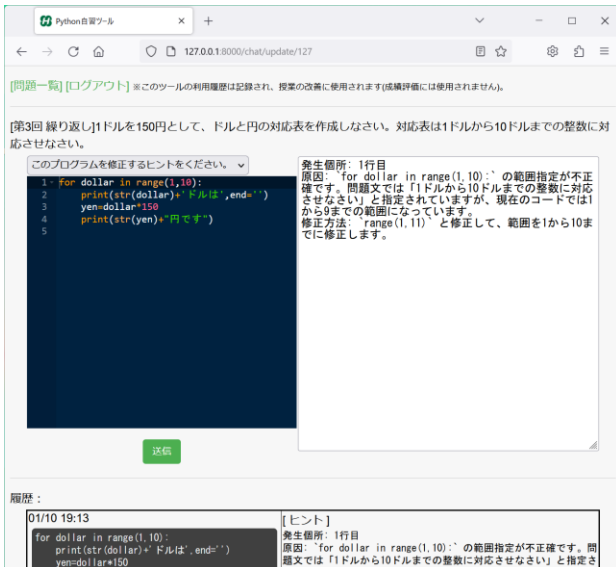
本稿では、学習者のソースコードに対して生成される助言について述べる。システムにログインし、取り組む課題を選んだ後、図 1 に示すメイン画面が表示される。学習者は左側のフィールドにソースコードを記入し、それに対する助言が右側のフィールドに表示される。画面の下部には、これまでにこの課題に対して提出したプログラムとその助言が表示される。

本システムは、Python の Web アプリケーションフレームワーク Flask で開発した。ソースコードの評価には、言語モデル統合フレームワー

クである LangChain、OpenAI 社の ChatGPT API (gpt-3.5-turbo)を用いた。API を呼び出す際には、次の情報及び制約を与えている。

- (1) 学習者が書いたソースコード
- (2) 結果の表示フォーマット：評価・助言の出力形式を統一するために指定する。学習者のコードに問題がある場合、場所、原因、修正方法、修正後のコードの順に出力するように指定した。
- (3) 演習課題：プログラムの文法エラーのみに助言するためであれば省略できるが、課題に正確に答えているかを含めた助言をするために必要となる。
- (4) 解答条件：使用するプログラム言語及びバージョン、既習の文法、利用可能なモジュールを指定する。さらに、課題ごとに助言に考慮すべき特別な観点があれば記載する。
- (5) 模範解答：文脈にあった助言を生成し、現在の授業の進行からずれた助言の抑制を期待している。その効果は現時点では明らかでない。

API の戻り値は、修正後のコードが含まれる。学習者が助言を理解することなく、修正後のコードをそのまま提出させないために、戻り値から修正後のコードを取り除いて、助言として出力することにした。



※コードエディタには Ace(Ajax.org) を使用
図 1 画面例

4. 出力結果の検証

実際に授業で使用する前に、助言の妥当性について、検討を行った。入力データは、過去の受講生の小テスト(表 1 の 3~10 回目で実施)での誤答 68 件を使用した。前項で述べた「(5) 模範解答」を API に入力する場合としない場合の両方の出力結果について、著者が評価した。評価結果と基準を表 2 に示す。

表 2 出力の有用性の評価

| 問題 | 問題 1 変数 | | 問題 2 条件分岐 | | 問題 3 繰り返し | | 問題 4 配列 | | |
|----|------------|------------|--------------|-------------|--------------|-------------|------------|------------|-------------|
| | あり | なし | あり | なし | あり | なし | あり | なし | |
| 評価 | 4 | 9 (64%) | 4 (29%) | 12 (86%) | 11 (79%) | 15 (88%) | 9 (53%) | 3 (13%) | 2 (9%) |
| | 3 | 4 (29%) | 3 (21%) | 2 (14%) | 2 (14%) | 1 (6%) | 3 (18%) | 8 (35%) | 2 (9%) |
| | 2 | 1 (7%) | 2 (14%) | 0 (0%) | 0 (0%) | 1 (6%) | 5 (29%) | 6 (26%) | 10 (43%) |
| | 1 | 0 (0%) | 5 (36%) | 0 (0%) | 1 (7%) | 0 (0%) | 0 (0%) | 6 (26%) | 9 (39%) |

評価基準

4:説明が適切であり、すぐに修正できる
 3:説明が分かりにくい、修正の参考になる
 2:説明に誤りは無いが、修正の参考にならない
 1:説明に誤りがある

問題 2,3 では、条件分岐、繰り返しに直接かかわる部分を出題したため、学習者の意図が読み取りやすく、出力は良好であった。問題 1(2つの乱数の値の合計)、問題 4(リストの要素をランダムに表示)では、新しい学習項目を複数含むため、著者が見ても、学習者が何を表現したいのか分からないプログラムが多く、適切な助言が出力されにくかった。

全誤答 68 件に対して、4 と評価された割合は、模範解答ありの場合で 39 件、なしの場合で 26 件であった。3 以上と評価された割合も、それぞれ 54 件、36 件であった。入力に模範解答を使用した方が適切な場合が 30 件、しない方が適切な場合は 1 件であった。特に、学習者の意図が不明なプログラムに対して、誤りの個所の正解を考慮することで理解を促す出力がされた。一方、学習者の誤りの根本的な原因が説明されないため、学習者は理解しないまま、プログラムを完成させる可能性も否定できない。

5. おわりに

AI が生成した助言を検証した結果、演習課題や解答条件、模範解答が出力に良好な影響を及ぼしていることが示唆された。一方、言語モデルには、学習者の誤りのデータがないことから、誤りのデータを学習させる必要がある。今後、学習者の理解状態の使用、理解度を確認するクイズの利用可能性について、教育への AI 利用のガイドラインを考慮しながら検討を行う。

参考文献

E. Lahtinen, K. Ala-Mutka, and H. Jarvinen: A Study of the Difficulties of Novice Programmers, Proc. of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, pp.14-18 (2005)