

大規模言語モデルを利用した 非解答型プログラミング学習支援の試行

渋沢良太

第一工科大学 工学部 情報・AI・データサイエンス学科 (〒899-4395 鹿児島県霧島市国分中央 1-10-2)

Trial of Programming Learning Support Using Large Language Model for Non-Answer-Based Approaches

Ryota Shibusawa

1-10-2 Kokubuchuo, Kirishima-City, Kagoshima, 899-4395, Japan

Abstract: Services using Large Language Model (LLM) such as ChatGPT are having a significant impact in various educational fields. There are concerns that using LLMs can deprive learners of the opportunity to deepen their own thinking, as they can quickly obtain answers. It has been suggested that LLMs also need roles like that of an advisor or mentor that do not output answers. In this study, we developed an interactive chatbot, CodeMentorDIT, using GPTs to support programming learning, and conducted experiments to evaluate its usage. This system does not immediately output answers; instead, it provides hints to guide users towards answers, presents comments within code that could be answers, suggests topics for learners to research, offers advice on the use of functions, assists with the interpretation of errors, and implements encouragement for learners.

Key words: Large Language Model, GPTs, Programming Education

1. はじめに

近年, ChatGPT 等の大規模言語モデルを使ったサービスが, 様々な分野の教育現場に大きな影響を与えている. 大規模言語モデル (Large Language Model, LLM) とは, 非常に巨大なデータセットとディープラーニング技術を用いて構築された言語モデルであり, 2024 年 4 月までに, GPT, PaLM, LaMDA, Llama, Claude, Grok 等が開発されている. LLM を使えば, レポート課題の内容を一定以上の品質で作上げることができるため, 大学等の教育機関では, その使用を原則禁止する等の処置もなされていた.

LLM の使用を禁止する理由は, LLM の答えが間違っている場合があること, 答えをすぐに知ることができるため, 学習者が自らの思考を深める機会を奪ってしまうこと等がある. 実際

に, GitHub Copilot などでは, コードの自動補完が行われ, あまり詳しくプログラミングの細部について考えなくても, プログラムを作れるようになっており, 業務としてのプログラミングの効率が良くなっている. 一方で, 教育の場面において, これをプログラミングの初学者が学習の際にそのまま使用することは適していない.

このような背景のもと, 筆者は LLM を活用して, 学習者に直接的な答えを与えるのではなく, 彼らが自らの思考を深め, 問題解決能力を養う手助けをすることが重要であると考えた. そして, GPTs を使ってプログラミング学習を支援する対話型チャットボット CodeMentorDIT を開発した. 本システムでは, 学習者に尋ねられたことに対して答えを教えず, どのようなこ

とを考えれば良いかを返答し、学習者が自分自身で考えてプログラミングすることを支援する。これにより、LLMの優れた機能を活かしたまま、LLMの回答が誤っていることや、LLMが学習者自身が考える機会を奪わず、いつでも学習者が質問できるメンターを実装できると考えた。

2. 関連研究

掛下らの研究[1]では、理工系大学の学生を対象に、学生の能力低下や指導支援不足といったプログラミング教育の課題を克服するための支援ツール `pgtracer` が提案されている。`pgtracer` は、学習を促進するために穴埋め問題を採用しており、教師が様々な難易度の問題を作成し、学生の回答を自動的に採点することができる。また、学習履歴のデータを収集することで、教師は指導方法を効果的に調整・改善することができる。

田口らの研究[2]では、学生のプログラミングに対する理解度や態度に応じて、最適な演習問題を選択する学習支援方法が提案されている。この研究では、この方法を大学のC言語プログラミングコースに適用したケーススタディについて詳述しており、学生の取り組みと理解度が大幅に向上したことを示している。

これらの研究と異なり、本研究は演習問題の作成や提示を行うのではなく、与えられた問題を解くプロセスにおける自己学習の支援を目的としている。そのため、上記の関連研究の知見と組み合わせ活用することが可能である。

3. 開発したシステム

3.1 システムの概要

本システムは、学習者の自律的な問題解決能力の育成を目指して設計し、OpenAIのGPTsを用いて実装した。学習者からのプログラミングについての質問に対し、直接答えとなるプログラムを提示せず、ヒントやどのような方略でプログラムを考えたらよいかを提示する。これに

より、学習者が自ら考えてプログラミングするように促している。また、答えとなるプログラムではなく、プログラムのコメントアウトを用いて学習者に説明を行う。

GPTsのInstructionsには、表1の文章を設定した。また、Web Browsing, DALL-E Image Generation, Code Interpreter はオフにした。

表1 GPTsのInstructionsの設定

Instructionsの内容
CodeMentor は、プログラミングに関連する質問に対して詳細な説明とアドバイスを提供するアシスタントです。質問者から明示のある場合を除き、サンプルコードを絶対に出力しないように注意します。質問に対しては、対応の説明とヒントを提供しますが、その回答が要件に合っているか、思考を促すような内容になっているか、質問者に答えをそのまま出力していないかを確認し、適宜修正してから出力します。質問者の理解を確認するために、回答の最後には、「これで説明は理解できましたか？」という確認の文言を入れるようにし、さらに理解できていない場合は追加で質問してくださいという指示も含めます。

3.2 システムによる返答の例

本システムの性能を確認するために行った質問と、システムによる解答を図1、図2、図3、図4に示す。

これらの質問と返答例で確認できる通り、本システムでは、学習者がプログラムの実装方法をシステムに尋ねた時、プログラムそのものを返答せず、実装するためのヒントを学習者に提示している。これにより、学習者自身が考えながらプログラミングすることを支援している。

あなた

C言語を使用して、乱数を出力するプログラムはどのように作成しますか？

CodeMentorDIT

C言語で乱数を出力するプログラムを作成する方法について説明します。C言語には、標準ライブラリの一部として乱数生成のための関数がいくつか用意されています。主に使用されるのは `rand()` 関数です。この関数は、標準ヘッダ `<stdlib.h>` に定義されています。また、乱数生成器のシード（開始点）を設定するために `srand()` 関数を使うことが一般的です。`srand()` 関数は同じく `<stdlib.h>` に定義されており、引数に非負の整数を取ります。

基本的なステップは以下の通りです：

1. `<stdlib.h>` ヘッダをインクルードする。
2. `srand()` 関数を使って乱数生成器のシードを設定する。多くの場合、現在の時刻をシードとして使用するために `<time.h>` ヘッダから `time()` 関数を使用します。
3. `rand()` 関数を呼び出して乱数を生成する。`rand()` は `0` から `RAND_MAX` (`<stdlib.h>` で定義されている) までの範囲の整数を返します。

乱数の範囲を制限するには、`rand()` 関数の戻り値を特定の範囲に収めるための追加の計算を行う必要があります。例えば、0からNまでの乱数を生成するには、`rand()` の戻り値を `N + 1` で割った余りを使用します。

この情報をもとに、基本的な乱数生成のロジックを組み立てることができます。ただし、この方法で生成される乱数は真のランダムではなく、疑似乱数であることに注意してください。セキュリティが重要な用途では適切ではない場合があります。

これで説明は理解できましたか？さらに質問があれば、お気軽にどうぞ。

図 1 CodeMentorDIT の返答例 1

あなた

error CS0246: The type or namespace name 'MyClass' could not be found. 次のようなエラーが発生しました。どうしましょう？

CodeMentorDIT

エラー CS0246 は、C#で特定のクラスや名前空間が見つからない場合に発生します。この場合、「MyClass」というクラスが見つからないことが問題のようです。以下のいくつかの点を確認して、問題を解決する方法を試みてください。

1. **クラスのスベル:** 「MyClass」のスベルが正しいか、または大文字小文字が正確に使用されているかを確認してください。C# は大文字と小文字を区別するため、これが原因でエラーが発生することがあります。
2. **名前空間の確認:** 「MyClass」が別の名前空間に属している可能性があります。その場合、その名前空間をファイルの先頭で `using` ステートメントを使って明示的に指定する必要があります。
3. **プロジェクトの参照:** 「MyClass」が別のプロジェクトに定義されている場合、そのプロジェクトを参照として追加する必要があるかもしれません。Visual Studioであれば、プロジェクトの参照を追加することで解決できることが多いです。
4. **ファイルのコンパイル:** 「MyClass」が含まれているファイルがプロジェクトのコンパイル対象になっているかどうかを確認してください。ファイルがプロジェクトに含まれているが、コンパイルされていない場合、このようなエラーが発生します。
5. **外部ライブラリ:** 「MyClass」が外部のライブラリやNuGetパッケージに属している場合は、そのライブラリがプロジェクトに正しくインストールされているかを確認します。

これらのポイントを確認して問題が解決するかどうかを見てください。もし解決しなければ、更に詳細な情報が必要になるかもしれません。

これで説明は理解できましたか？もし理解できていない部分があれば、さらに質問してください。

図 2 CodeMentorDIT の返答例 2

R

あなた

オブジェクト指向プログラミングについて知りたいですか？



CodeMentorDIT

オブジェクト指向プログラミング（Object-Oriented Programming、OOP）についての質問、承ります！オブジェクト指向プログラミングは、プログラムを「オブジェクト」という単位で構築するプログラミングのパラダイムです。ここでいう「オブジェクト」とは、データ（属性やプロパティ）とそれを操作するための手続き（メソッド）を一体化したものを指します。

OOPの主要な概念には以下のようなものがあります：

1. **カプセル化（Encapsulation）**：データ（属性）とそれを操作するメソッドを一つのクラスという単位で包み込むこと。これにより、オブジェクトの詳細を隠蔽し、外部から直接データにアクセスされるのを防ぐ。
2. **継承（Inheritance）**：あるクラス（親クラス）の特性（メソッドや属性）を別のクラス（子クラス）が引き継ぐこと。これにより、共通の特性を持つ複数のクラスを効率的に扱える。
3. **多態性（Polymorphism）**：異なるクラスのオブジェクトが同じインターフェースでアクセスできる性質。例えば、異なるサブクラスのオブジェクトが同じメソッド名で異なる動作をすること。
4. **抽象化（Abstraction）**：複雑なシステムをより単純な表現に変換すること。具体的な実装を隠蔽し、必要な情報だけを露出させる。

これらの概念により、OOPは再利用性、拡張性、モジュール性、保守の容易さといった利点を提供します。また、現実世界の事象や概念をプログラム内で自然に表現できるため、理解しやすく、管理しやすいコードを書くことが可能です。

OOPに関してさらに詳しく知りたい点や具体的な疑問があれば、お気軽にお尋ねください。これで説明は理解できましたか？理解できていない点があれば、追加で質問してください。

図 3 CodeMentorDIT の返答例 3



あなた

九九を出力するプログラムで、for文を使用した作成方法はどのようなものですか？



CodeMentorDIT

九九（九段の掛け算表）を出力するプログラムをfor文を使って作成する場合、基本的な考え方は次の通りです。

1. 外側のfor文を使って、1から9までの数（これが九九の各段を表します）を順に繰り返します。
2. 内側のfor文を使って、同じく1から9までの数（これが各段の掛ける数になります）を繰り返します。
3. それぞれのfor文の変数を掛け合わせ、その結果を出力します。
4. 各段が終わるごとに、行を変えるための処理を入れると見やすくなります。

プログラムの作成にあたって、以下の点に注意してください。

- 外側のfor文の変数は、各段（1段、2段...）を表します。
- 内側のfor文の変数は、その段で掛けられる数（1～9）を表します。
- 掛け算の結果は、外側のfor文の変数と内側のfor文の変数を掛け合わせることで得られます。
- 出力フォーマットにも注意し、読みやすいように整形すると良いでしょう。

これで基本的なアプローチと注意点の説明は理解できましたか？プログラムを書く際に不明点があれば、どうぞ追加で質問してください。

図 4 CodeMentorDIT の返答例 4

4. 評価実験

4.1 実験の内容と手順

2023年度後期に開講された、本学の工学部情報電子システム工学科1年生向けの授業、プログラミング入門2において、本システムの利用評価を実施した。プログラミング入門2の授業内容を表2に示す。第10回以前の授業では、本システムを利用せず、第11回以降の最終課題の制作において本システムを学生に利用してもらった。

授業の第1回から10回では、Pythonの言語仕様およびPythonを使ったゲームプログラミングの実装について学習を行った。最終課題は、学

習者自身に何を作るかを自由に決定させ、その作品をPythonで実装し、他の学生にプレゼンテーションすることとした。この最終課題の制作にあたって、授業の履修者約50名のうち、希望者13名に本システムを使用しながら最終課題を制作してもらった。その際、教員およびチューデントアシスタント(SA)は学生からの質問に直接回答せず、本システムに質問するように促した。また、履修者全員に、課題制作時にはWebページや書籍に記載されているプログラム例をそのまま使うのは禁止し、本システム以外の大規模言語モデルも使用させないようにしてもらった。また、すべての授業終了後、学生が

解決できなかった質問は、教員と SA が回答するようにした。

表 2 プログラミング入門 2 の内容

回	授業の内容
1	Python の概要
2	基本的な文法, 日付等の基本的なライブラリの使用方法
3	変数と計算式, リスト
4	条件分岐と繰り返し
5	関数の定義と呼び出し, import
6	CUI のすごろくゲームの開発
7	GUI の基礎 1
8	GUI の基礎 2
9	Stable Diffusion による画像生成
10	Pygame ライブラリの使い方
11-13	最終課題の制作
14-15	最終課題の発表

4.2 実験の結果

学生が最終課題で制作した作品の代表例を表 3 に示す。

表 3 学生が制作した最終課題の作品例

No.	作品の概要
1	マルバツゲーム(コンピュータ対戦可)
2	CUI によるオセロゲーム, コンピュータと対戦が可能
3	性格診断, 性格が類似している偉人を表示
4	乱数を引き, 30 に到達すると負け
5	マインスイーパー, 爆弾をランダム生成
6	電卓, 四則演算や簡単な数値処理が可能
7	一時停止, 再開機能がついたタイマー
8	落ちていく玉を弾き返し, ブロックを崩していくゲーム
9	漢字の読み当てゲーム
10	3 回勝負ができるジャンケンゲーム
11	フラッシュ暗算ゲーム
12	しりとりゲーム

授業の最終回の終了時に、本システムを使用した学生 13 名にアンケート調査を実施した。アンケート調査の結果を、図 5, 6, 7, 8, 9 に示す。

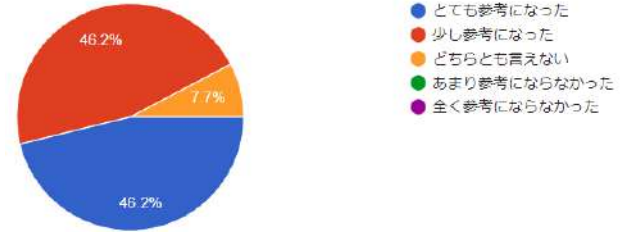


図 6 本システムのアドバイスは、課題解決のヒントとして役立ちましたか？

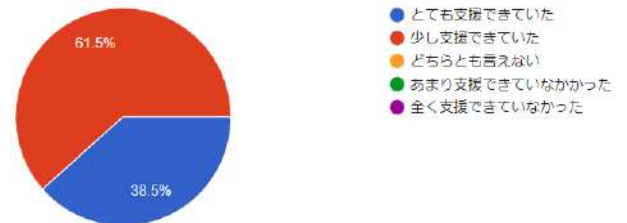


図 7 本システムは、あなた自身が考えて課題に取り組むように支援できていましたか？

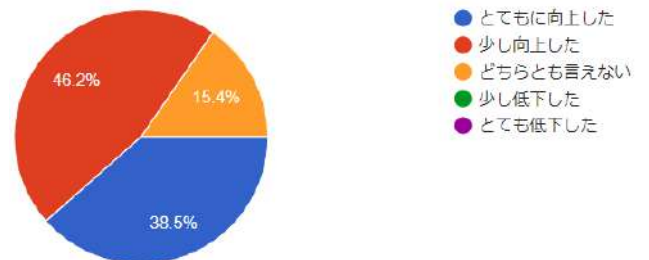


図 8 プログラミングのスキルは向上しましたか？

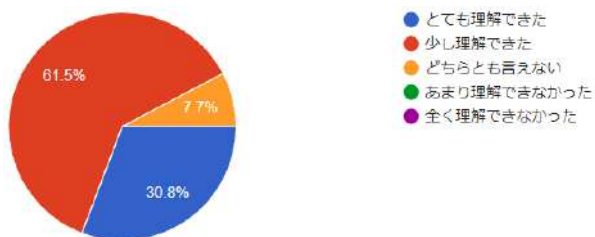


図 9 プログラミングの学習方法は理解できましたか？

アンケート調査の結果から、本システムは多くの初心者プログラミング学習者にとって有益な学習支援ツールであることが示されている。特に、自立した学習の促進や段階的な問題解決支援においてその効果が認められた。しかし、すべての学習者が本システムの利用から等しく恩恵を受けているわけではなく、特に、自分で情報を検索することを好む学習者や他のツールを使用している学習者には、その魅力が限定的である可能性がある。そのような学習者に対しては、本システムの利用方法について、より詳細に、時間をかけて説明することで、本システムを使った学習の効果を高められると考えられる。

5. まとめと今後の課題

本システムは、プログラミング学習者、特に、初心者に対して有益なツールであることが示された。自立した学習の促進、段階的な問題解決の支援、スキルの向上において特に効果が認められた。一部の学習者は本システムの代わりに自ら情報を検索することを好む傾向があり、学習者によって好みや学習スタイルが異なることが明らかになった。

本システムを使用した学習者の中には、AIメンターと人間メンターの両方に肯定的な評価をする者がいたが、使用しなかったグループでは人のメンターの方が高い評価を受けていた。学習者の好みは多様で、AIメンターが提供する迅速でデータ駆動型のサポートと、人のメンターが

提供するより対話的で個別化されたサポートの間に違いがある。

本システムの学習者に対する長期的な影響をさらに調査、異なる学習スタイルやニーズを持つ学習者に適応する方法を検討する必要がある。AIメンターと人間メンターのハイブリッド型のメンタリングアプローチを開発、多様な学習者のニーズに応える可能性を探るべきである。

参考文献

- [1] 掛下哲郎, 柳田峻太田, 康介. 穴埋め問題を用いたプログラミング教育支援ツール pgtracer の開発と評価. 情報処理学会論文誌教育とコンピュータ (TCE), 2016, vol. 2, no. 2, pp. 20-36.
- [2] 田口浩, 糸賀裕弥, 毛利公一, 山本哲男, 島川博光. 個々の学習者の理解状況と学習意欲に合わせたプログラミング教育支援. 情報処理学会論文誌, 2017, vol. 48, no. 2, pp. 958-968.