

KOGI: ChatGPTをColabに統合した プログラミング演習支援

小原 有以^{1,a)} 佐藤 美唯² 倉光 君郎^{1,b)}

概要:

プログラミングは創造的な活動である一方で、学習者は予期せぬエラーに絶えず直面し、経験豊富な教員やTAからの助言は極めて有用である。一方、経験のある教員やTAへのアクセスは限られており、些細な疑問すら解決できない状況が生まれている。我々は、教員やTAの代わりにAIシステムを導入することで、学生の質問や疑問に即座に応えるプログラミング演習環境の構築を目指している。

KOGIは、クラウド上の演習環境 Google Colab に対話 AI を統合した学習支援システムである。大規模言語モデルとして ChatGPT を採用することで、学生からの質問への回答能力は大幅に向上した。本論文では、KOGIの開発状況と教室での実践結果について報告する。これらの先行経験を通じて、大規模言語モデルの教育用 AI 開発の可能性について展望する。

A Preliminary Report on Programming Education on Colab with ChatGPT

1. はじめに

デジタル時代の到来に伴い、将来の多くの職業においてデジタルスキルが求められるようになってきている。特にプログラミングは、高度なデジタルスキルを習得するための入り口となり、データサイエンスやデジタル創作活動などの人気な職業で重要な役割を果たしている。そのため、多くの若い学生や保護者がプログラミングスキルの習得に高い関心を寄せている。

しかし、プログラミングスキルの習得は容易ではない。今日、Web上にはさまざまな興味深いチュートリアルが存在しているが、ほとんどの学生はそれらを試してみるだけで終わることが多い。もし、少しでも書き直してみようと試みた場合、自分では解決できないエラーが発生し、前へ

進むことが困難になることも少なくない。

このような状況の解決策は、経験豊富な教員やTAによる個別のアドバイスである。プログラミング学習中に生じる疑問は、学生や状況によって変化し、千差万別である。これらの疑問に的確に応えるためには、コンピュータ、オペレーティングシステム、コンパイラ、アルゴリズムなど幅広い基礎知識が必要となる。しかしながら、コンピュータサイエンスの学位を持った教員は少ない。また、経験豊富な教員やTAがいても、受講する学生数が多いため、個別に対応できる時間も限られている。

本研究の目的は、学生が十分なアドバイスを受けられない課題に対し、AIによる支援を提供することである。我々は、2021年頃から大規模言語モデルを活用したプログラミング学習支援システム KOGI を開発してきた。高校生から大学3年生、更に大学院生までの幅広いレベルの学生を対象に、KOGIを提供し学生からのフィードバックを得てきた。本論文では、ChatGPTを採用したKOGIを報告する。

ChatGPTは、OpenAI社が2022年にリリースした対話型の大規模言語モデルである^{*1}。自然言語のテキスト処理

¹ 日本女子大学理学部数物情報科学科
Department of Mathematics, Physics, and Computer Science, Japan Women's University, 2-8-1 Mejirodai, Bunkyo-ku, Tokyo 112-8681, Japan

² 日本女子大学大学院理学研究科数理・物性構造科学専攻
Division of Mathematical and Physical Sciences, Graduate School of Science, Japan Women's University, 2-8-1 Mejirodai, Bunkyo-ku, Tokyo 112-8681, Japan

a) m2016026oy@ug.jwu.ac.jp

b) kuramitsuk@fc.jwu.ac.jp

*1 <https://openai.com/blog/chatgpt>

用に構築された言語モデルであるが、コードも学習対象に含まれており、プログラミングに関する質問にも対応することができる。我々は ChatGPT をプログラミング演習環境である Google Colab に統合することで、学生のプログラミング演習中の疑問に回答しやすい環境を開発した。

本論文では、KOGI の開発状況を紹介し、クラスルーム利用における効果をアンケートとログ分析から報告する。本論文の残りの構成は以下の通りである。2 節では、プログラミング演習環境である Google Colab を紹介し、現在のプログラミング教育に残る課題点について説明する。3 節では、ChatGPT とプログラミングに関する能力について解説する。4 節では、ChatGPT をプログラミング環境 Google Colab に統合する方法について説明する。5 節では、学生の反応を通じた ChatGPT の経験について述べる。6 節では関連研究を概説し、7 節で最終的な結論を述べる。

2. 現代プログラミング教育と課題

現在のプログラミング教育の対象は、従来のコンピュータサイエンスを学ぶ大学生に限らず、中高生や文系大学生など異なる背景や関心をもった学習者が増えている [1], [2]。そのため、プログラミング教育の課題も少しずつ変化してきている。

2.1 Google Colab

プログラミング学習を始める際の最初の壁は、従来はプログラミング演習環境の構築にあった。現在は、クラウドコンピューティングが普及し、Web ブラウザ経由で手軽にプログラミングが始められる環境が多く提供されている。

我々も、Google 社が提供する Google Colab をプログラミング演習環境に採用し、中学生から大学生までプログラミング演習の授業や講習を提供してきた。Google Colab は Jupyter ノートブックのクラウド版であるが、Google 社の安定したクラウド環境が利用できるため、多くの大学やプログラミング教室で採用されている。以下に Google Colab を利点を示す。

- 十分な計算機資源：CPU(GPU)、メモリ、ストレージと十分な計算機資源が提供される。
- 整備された環境：Python 処理系だけでなく、最新のデータサイエンス/AI 開発に必要なライブラリが設定済みである。
- 教材活用：ノートブック形式は、テキストや画像も手軽に組み合わせられ、それ自体が教材になる。
- ノートブックの共有：学生は、教員や学生同士でコードや実行結果を共有し、進捗確認や質問応答に活用できる。

Google Colab は、本格的な AI 開発も可能なプロフェッショナルな環境でありながら、ブラウザの操作経験さえあれば、誰でも使い始められる。しかし、プログラミングス

```
[2] df = pd.read_csv('pollen.csv')
NameError                                Traceback (most recent call last)
<ipython-input-2-803f2727015c> in <cell line: 1>()
----> 1 df = pd.read_csv('pollen.csv')
NameError: name 'pd' is not defined
(原因①) pandasのimportし忘れ

[4] df = pd.read_csv("pollen.csv")
File <ipython-input-4-f8d90b8d6fb9>, line 1
df = pd.read_csv("pollen.csv")
                ^
SyntaxError: unterminated string literal (detected at line 1)
(原因②) 一重引用符と二重引用符の間違い

[5] df = pd.read_csv('pollen.csv')
NameError                                Traceback (most recent call last)
<ipython-input-5-4c73b9cb8f4c> in <cell line: 1>()
----> 1 df = pd.read_csv('pollen.csv')
NameError: name 'read_csv' is not defined
(原因③) カンマとピリオドの打ち間違い

[6] df = pd.read_csv('pollen.csv ')
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-6-17cbd0a9e93b2> in <cell line: 1>()
----> 1 df = pd.read_csv('pollen.csv ')
6 frames
/usr/local/lib/python3.10/dist-packages/pandas/io/common.py in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
    854     if ioargs.encoding and "b" not in ioargs.mode:
    855         # Encoding
-> 856         handle = open(
    857             handle,
    858             ioargs.mode,
FileNotFoundError: [Errno 2] No such file or directory: 'pollen.csv '
(原因④) ファイル名に余分な空白を入れた
```

図 1 初学者のよくあるエラー例

キル習得が容易になるかといえば、そうではない。プログラミングスキル習得を阻む障壁は大きく存在している。

2.2 エラーという壁

エラーという壁について、近年人気のデータサイエンス演習を例に考えてみる。下記は、学生が最初に共通して実行する、ファイルからデータを読み込むコード例である。

```
df = pd.read_csv('file.csv')
```

教員と一緒にやる Hands-on 形式の演習では、学生がこのコードを入力するだけでも、複数の学生がエラーを発生させてしまう。図 1 は、僅かな間違いが非常に多様なエラーを引き起こす例を示している。1 つ目の例は、事前に `import pandas as pd` を実行していないため、発生している。2 つ目の例は、引用符 `"` と `'` が異なる例である。3 つ目の例は、エラーメッセージは `read_csv` が未定義と記述されているが、実際は `.` を、と間違えている。4 つ目の例は、よくある File Not Found のエラーである。この場合はファイル自体は存在し、ファイル名の後に不要な空白があるため発生している。

エラーが壁になる理由は、以前まではエラーメッセージが英語であるため、英語力不足が問題であると考えられた。しかし、英語のエラーメッセージを解読できたところで、エラーメッセージが解決策のヒントにならないことも多い。

初学者は、エラーの原因を理解するための基礎知識が不十分なまま取り組むことが多く、エラーを解決するのは困難とされている [3], [4]。エラーの発生後、エラーを自力で直せないため挫折したり、過度の苦手意識をもってしまいうことも少なくない。

3. 対話 AI 技術

我々の問題意識は、「大規模言語モデルを用いた最新の対話 AI 技術を導入することで、第 2 節で述べたプログラミング演習における困難な状況を支援することができるか？」である。

3.1 ChatGPT

ChatGPT は、OpenAI 社が 2022 年 11 月にリリースした大規模言語モデル GPT-3.5 に基づく対話 AI である。その技術基盤となった GPT(Generative Pretrained Transformer) は、テキストコーパスにおける単語の共起性や統計的なパターンを学習し、それを元に文の生成を行う大規模言語モデルである。ChatGPT は、人間との対話用途に特化させて、差別を助長する発言、バイアスに基づく表現、その他望ましくない回答をしないように、GPT を強化学習によって調整されている。

ChatGPT の興味深い点は、プログラミングに関する質問に回答する能力がある点である。これは、学習したテキストコーパス内にコードサンプルやプログラミングの質問応答などが含まれるためである。この能力は、プログラミングの質問応答やコードサンプルの生成、コード修正などのソフトウェア工学タスクも対応でき、プログラミング学習支援に応用できると期待される。

ChatGPT は、**幻覚 (Hallucination)**[5] と呼ばれる現実には存在しない回答をするリスクがある。これは、訓練データから確率的にありえる情報を生成するために発生する。現状、生成された文の正確性を検証する手段を持たず、追加の検証手段が必要な場合がある。

3.2 プロンプト

プロンプトは、ChatGPT などの大規模言語モデルに与えられるテキストの入力や指示のことである。プロンプトの役割は、以下の通り大別される。

- **タスク指定:** プロンプトは、ChatGPT に対して実行したいタスクや要求を指定する役割を果たす。例えば、要約、翻訳、質問応答などの特定のタスクを実行するために、プロンプトに対応するタスク指定することができる。
- **文脈の提供:** プロンプトは、対話や文書の文脈を提供することができる。これにより、ChatGPT は与えられた文脈に基づいて応答を生成することができる。
- **応答形式の指定:** プロンプトは、ChatGPT に対して応答の形式や話調を指定できる。これにより、応答の長さや柔らかさを変更できる。

大規模言語モデルの回答は、プロンプトによって大きく品質が異なることが知られている。コンテキストを適切に与えることは、より利用者の求める回答を得る重要な要素と

プロンプト 1

次のエラーの原因を教えてください。

```
コード: df = pd.read_csv('pollen.csv ')
FileNotFoundError: [Errno 2] No such file or directory:
'pollen.csv '
```

ChatGPT からの回答

このエラーは、指定したファイルパスに存在しないファイルが見つかったことを示しています。エラーメッセージに`pollen.csv` (末尾にスペースが含まれています) と表示されているため、ファイル名に余分なスペースが含まれている可能性があります。

プロンプト 2

以下のコードを直してください。

```
df = pd.read_csv('pollen.csv')
df.head()
```

ChatGPT からの回答

以下の修正したコードをお試しください。

```
import pandas as pd
df = pd.read_csv('pollen.csv')
print(df.head())
```

修正後のコードでは、`pd.read_csv()` 関数の呼び出しを修正しています。コード内でコンマ (,) がドット (.) になっているため、それを修正しました。

なる。

3.3 予備調査

我々は、ChatGPT のプログラミング演習環境への統合を進める。具体的な統合を説明する前に、ChatGPT の能力をプロンプトの実例とともに調査した結果を示す。なお、temperature で回答にランダム性を与えているため、回答の再現性は保証されない。

プロンプト 1 は、発生したエラーとエラーの原因を問い合わせたプロンプトの例である。エラーメッセージの直訳だけでなく、ファイル名の誤りを正しく指摘できていることがわかる。プロンプト 2 は、エラーが発生したコードを修正する例である。エラーの原因を的確に指摘し、コード修正ができている。一方、言語モデルの特性から `df.head()` は間違っていないが、より一般的な `print(df.head())` に変更される様子も確認できる。

プロンプト 3、プロンプト 4 はエラー対応以外にどのような支援が可能か調べた例である。学生は、授業で提示されたコードの意味がわからなくなる状況も少なくないが、ChatGPT は解説やコメントを適切に与えることができることがわかる。また、キーワードや用語の疑問にも説明は多少冗長であるが応えることができている。

このようにプロンプトを与えることで多くの支援が可能になることがわかる。一方、的確にプロンプトを与えるこ

プロンプト 3

‘pollen.csv’ からデータフレームを読むコードを教えてください。

ChatGPT からの回答

Python の場合、pandas ライブラリを使用して CSV ファイルを読み込むことができます。以下のコードを使用して、‘pollen.csv’ ファイルをデータフレームとして読み込むことができます。

```
'''
import pandas as pd
# CSV ファイルを読み込む
df = pd.read_csv('pollen.csv')
# データフレームの内容を確認する
print(df)
'''
```

プロンプト 4

データフレームって何？

ChatGPT からの回答

データフレームは、表形式のデータを扱うためのデータ構造です。特に、pandas ライブラリで提供されています。

データフレームは、行と列からなる 2 次元のデータ構造であり、Excel のスプレッドシートや SQL のテーブルに似ています。各列は異なるデータ型を持つことができ、データフレーム内の異なる行や列にはラベルやインデックスが付けられます。データフレームは、データの整理や操作、分析を容易にするための多くの機能やメソッドを提供します。例えば、データのフィルタリング、集計、ソート、結合などの操作を行うことができます。また、データフレームはデータの可視化や統計分析にも使用されます。

pandas ライブラリを使用すると、CSV や Excel ファイル、データベースからデータを読み込んでデータフレームとして扱ったり、データフレームをファイルに書き込んだりすることができます。

簡単に言えば、データフレームは構造化されたデータを表形式で管理するための強力なツールです。データの取り扱いや分析において広く使われています。



図 2 対話画面の起動: KOGI への質問

との難しさがある。

4. KOGI

KOGI は、Google Colab に ChatGPT が統合されたプログラミング学習支援システムである。本節は、KOGI のデザイン・コンセプト、アーキテクチャ設計、実装につい



図 3 KOGI からの回答

て述べる。

4.1 アーキテクチャ

まず、KOGI の全体的なアーキテクチャを説明する。KOGI は Python モジュールとして提供されており、Google Colab 上で pip を使用してインストールし、import 文でアクティベートできる。一旦、アクティベートされると、後述で詳細を説明する通り、Jupyter カーネルとシームレスに統合される。

KOGI の基本動作は、対話である。対話は、Google Colab の実行セルを実行しエラーが発生したときに、図 2 のようにエラー情報を示すことで始まる。

エラーが発生したときのスタックトレースからエラーの情報を取り出して、学生の書いたコードと比較してハイライトする。これを実行時情報のコンテキストに追加し、ChatGPT に送るプロンプトを生成する。そして、ChatGPT から回答を受け取り、ユーザに返す。

なお、KOGI は、研究室の TA が飼っているロボット犬 (デジタルペット) として位置付けられている。これは、KOGI 開発プロジェクトが始まった当初、我々の独自開発を行っていた大規模言語モデル [6], [7] は、50%程度の正解率に過ぎず、明らかな間違いも多かった。そのため、心理的な抵抗感を下げるため、ユーモアがあって学生に親しみやすい KOGI のキャラクターが開発された。

KOGI は図 4 に示す通りの手順で動作する。

- (1) Jupyter カーネルのテーブルから実行時情報 (コードや変数) を読み取る。
- (2) ユーザの問い合わせに、実行時情報のコンテキストとして追加してプロンプトを生成する。
- (3) ChatGPT へ OpenAI API を経由してプロンプトを送る。
- (4) ChatGPT から回答を受け取り、ユーザに返す。
- (5) AWS S3 ストレージにログとして、コードやプロンプト、回答を記録する。

図 3 は、ChatGPT のレスポンスを KOGI が回答している例である。将来は、KOGI が ChatGPT のレスポンスを検証して一部加工することも予定しているが、現在は、フォーマットの調整の加工に留め、回答している。



図 4 KOGI アーキテクチャ



図 5 エラー診断: どうしたらいいの?



図 7 コード修正: 直して

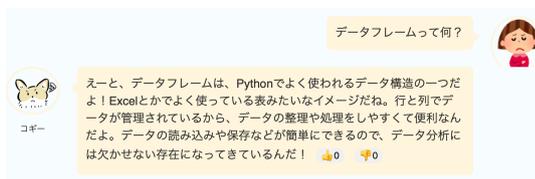


図 6 用語解説



図 8 キーワード引数の設定

4.2 KOGIのAI支援機能

ユーザからの目線でKOGIの使い方をもう少し説明する。

KOGIのAI支援は、エラー診断と呼ばれる機能が中心になっている。これは、発生したエラーに対して、原因と解決のためのヒントを提示することを期待した機能である。エラーに対して図5のように「どうしたらいいの?」と質問すると、期待する回答が得られる。

KOGIでは、講義で習った用語だけを使うように制限することは難しいため、「原因」や「解決策」を未知の専門用語を用いて解説されることがある。このような場合は、図6のように、「〇〇って何?」のように用語解説を求めることができる。

ユーザは、解決のためのヒントを提示されても、エラーを自力で解決することが難しいことがある。KOGIでは、図7のように、より直接的に「直して」と、コード修正を要求することができる。初心者がするような初歩的な誤り

はコードの修正案を得ることができる。

コード修正の支援は、実は非常に強力である。例えば、図8のようなキーワード引数の書き方がわからない場合、コード内に「文字化けしたくない」と実行したい内容を入力することで、求めるキーワード引数の書き方が得られる。

最後に興味深い機能を支援を紹介しておきたい。学生は、サンプルで提供されたコードの意味がわからなくなり、コメントを求めることが多い。ChatGPTは、ソースコードにコメントを付けることができるため、KOGIにも期待されるが、非常に自然に統合されている。図9は実行セル上に「コメントをつけて」と先頭に書いて実行すると、(発生したエラーからKOGIが中継することで)コメント生成



図 9 コメント

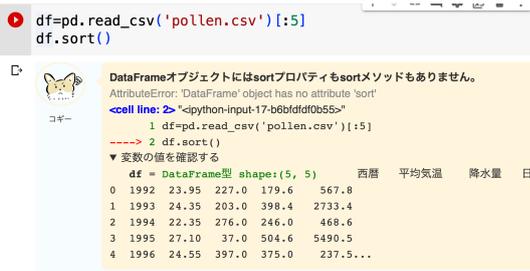


図 10 変数を見る

が行えた例である。KOGIは、少しでも口調になっている。

4.3 シームレスな統合

最後に、興味のある読者のために KOGI 実装について述べる。KOGI は、Python モジュールとして提供され、通常の Python モジュールのインストール手順で使うことができる。Jupyter カーネルが提供するマジックセルなどの標準的な拡張手段は使わず、Jupyter Notebook (IPython) カーネルを書き換えて、実行セルが実行されたタイミングで割り込めるようにしている。これは、Python に不慣れな学生が余計な設定をしなくても、すぐに利用可能にするためである。

KOGI の割り込み処理は、前処理と後処理に大きく分けられる。

前処理は、実行する前のソースコードをスキャンして、その結果から後続の処理を変えることができる。KOGI は、前処理の機構を用いて、競技プログラミングサイトの AtCoder のサンプルデータを獲得し、テストを実行するなどの機能がある。

後処理は、基本的にエラーが発生したとき、割り込みが発生する。KOGI は、Jupyter や Google Colab が提供する標準的なスタックトレースの代わりに独自のスタックトレースでエラーを分析している。図 10 のように、エラーが発生した時点での変数の値を見ることができる。

対話型チャットの UI 起動し、ユーザの入力から実行を行っている。対話型チャットなどの UI は、HTML と JavaScript で実装行っている。現在は、Jupyter カーネルサーバと通信方式が、Jupyter Lab, Colab, Visual Studio

Code (ipython-kernel モード) と全て異なるため、Google Colab のみのサポートとなっている。ただし、原理的には全ての Jupyter で動作できる。

5. 実践報告

我々は、前節で述べた KOGI を 2023 年度 4 月より、著者らが担当するプログラミング演習の講義で活用した。本節では、KOGI の教育実践における経験について述べる。

5.1 母集団

我々は、2023 年春より、著書らが教員・TA を務める以下の講義において KOGI を提供した。

- アルゴリズムとデータ構造（日本女子大学理学部数物情報科学科 2 年生，対面形式，2023 年春）
- 機械学習（日本女子大学理学部数物情報科学科 3 年生，対面形式，2023 年春）

双方の講義で、60 分以上のプログラミング演習が含まれている。アルゴリズムとデータ構造は、1 年次に学習した Python の文法知識をベースに、AtCoder の問題を学生自身のペースで解く演習形式である。機械学習は、データ可視化や回帰分析などの概念を学んだ後に、pandas や matplotlib を用いたプログラミング演習が含まれている。

5.2 ログ解析

KOGI は、4.1 節で述べた通り、ユーザの実行したコード、発生したエラー、ユーザの入力、ChatGPT からの回答を全てログとして記録している。本節では、2023 年 5 月 1 日から 5 月 31 日の KOGI のログを分析し、学生の動向を見ていく。

表 1 各クラスと KOGI への質問数

クラス	演習内容	件数
A	繰り返し，文字列，リストなど	399
D	基礎統計量，データ可視化，回帰分析	367

ここからは、便宜上、クラス A を、アルゴリズムとデータ構造、クラス D を機械学習の演習に取り組んだ学生を表すものとする。授業時間外の自主的な演習のログも含まれているため、コードや課題からどちらのクラスに属するか人手判断しラベルを付けた。表 1 は、各クラスと学生から KOGI への質問数を集計している。延べ利用者数は、クラス A が 62 人、クラス D が 65 人となった。

我々は質問の傾向を把握するため、次のように質問の種類を分類した。

- **解説:** 語彙やキーワードの解説を求める: 例. ○○ (用語) とは、○○って何ですか?、○○について教えてください
- **コード:** コードの例示を求める: 例. ○○するコードを教えてください、○○ (コード実行) したい

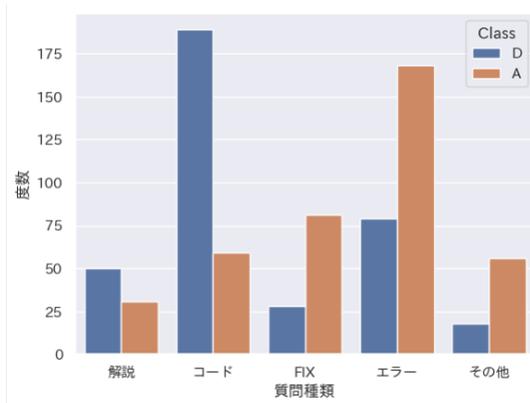


図 11 質問種類と度数分布

- **FIX**: 書いたコードの修正を求める: 例. 直して
- **エラー**: エラーに対する原因やヒントを求める: 例. どうしたらいいの?
- **その他**: 感謝 (例. ありがとう) や怒り (例. お前頭悪) を表現する言葉など

図 11 は質問種類とその回数をクラスごとに集計したヒストグラムである。クラス A は、「エラー」や「FIX」に関する質問が多いが、クラス D は、「エラー」に関する質問より「コード」に関する質問が増える。このことから、クラスによって AI を最も必要とする質問の種類が異なることを示している。

これは、著者らの想定に反しない傾向であるが、原因の分析はまだ十分に行えている状況ではない。ここでは、AI への利用状況として 2 つの仮説を述べるに留めたい。

- (習熟度) クラス D の方が 1 学年分だけ、エラーの対応を含めたプログラミングへの習熟度が全体的に高い。学生の関心は、エラーよりも別の内容に移っている。
- (質問しやすさ) クラス A は、AtCoder の ABC 問題が中心で、論理的問題を解くことが多い。KOGI にコードの例示を求めるには、KOGI に問題のロジックを正確に説明をする必要があるため、「コード」に関する質問はしにくい。そのため、KOGI にコード生成をさせるよりも、自分でコードを作成し、KOGI にエラーの原因やヒント、コード修正の支援をもらった方が早いと学生が考えた可能性がある。したがって、「エラー」や「FIX」に関する質問はしやすい。一方、クラス D では NumPy, Matplotlib, Pandas などのライブラリにある関数を利用することでデータ分析を行う。自分のさせたい動作についての関数を Web で検索するより KOGI に直接質問する方が早いと学生が考えた可能性がある。そして、データの前処理やグラフ作成など、自分のさせたい動作について説明しやすいため、「コード」に関して KOGI に質問しやすい。

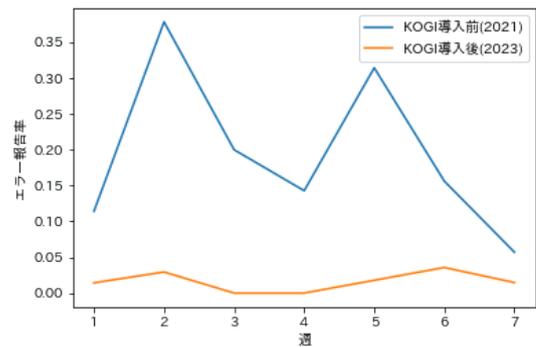


図 12 自由感想文におけるエラー相談件数の推移

5.3 AI 導入の効果

我々は、担当する教員や TA の体感としても、学生から質問される内容が単純なエラーに対する質問は減り、プログラミングとしてのロジックの組み方や考え方への質問に変わったと感じている。これらを定量的に裏付ける比較を最後に示したい。

我々は、毎回プログラミング演習の後に自由感想文によって演習の進捗や疑問、困っていることの見を集めている。図 12 は、自由感想文からエラーが発生して困った相談件数を数え、全体の感想に占める割合を各週ごとに比較したものである。対象は、クラス A に相当する 2 年生向けの AtCoder による演習である。なお、我々は 2022 年には、KOGI の前世代版の導入を始めていたため、AI を導入する前の 2021 年との比較になっている。

KOGI 導入の 2021 年は、多い週で 30%以上の学生がエラーについての苦勞を報告している。これは、KOGI 開発の動機にもなっていたが、エラーの発生で課題が進められないという深刻な報告も少なくなかった。一方、KOGI 導入後は、毎回、1,2 人のエラーについて述べるだけである。しかも、深刻な状況を感じさせる相談は 1 件も存在しなくなった。

公平さのため注意：2021 年はコロナ下でオンラインで実施されたプログラミング演習であったため、比較には注意が必要である。我々は、Zoom のブレイクアウトルームを活用し、対面形式と同じように友人に相談できる環境を提供し、TA の数も同数であった。

5.4 得られた教訓

最後に、KOGI を教室で運用してきて得られた経験をまとめておく。

ChatGPT は大規模言語モデルであり、教育 AI ではない。

ChatGPT の可能性は大きいテキストコーパスを学習した大規模言語モデルである。ただ導入するだけでは、教員の代わりに学生の応対が可能になるような特別な訓練はされていない。学生の疑問や困難、理解度を類推し、学習効果が高まるような応答の実現が重要になる。

AIの介入は、学習意欲や持続性を向上させる。我々が実装したKOGIは、キャラクター設定もあり、多くの学生から好評を得ている。教員やTAが些細な質問で呼び出される回数は減り、演習時間中に他のことに意識を取られる学生も減っている。

学生に積極的にAIを活用させるのも難しい。我々はAIを活用して、学生が積極的により難しい課題に取り組むことを願うが、残念ながらAIを使いこなせず、諦める学生も少なからず存在する。まずはAIの活用方法を学ばせる必要がある。

学生の倫理観を高め、信用しよう。AIによる不正を恐れるあまり、教員はAIの活用に制限をかけがちである。不正を行う学生は、AIがなくても別の手段で不正を行うため、もう少し学生の倫理観を信じて、AIによる新たな可能性に期待をしよう。

6. 関連研究

AIの教育利用は、Intelligent Tutoring Systems(ITS)の時代から盛んに検討されてきた。近年、深層学習、大規模言語モデルと技術革新が続き、より身近にAIを活用した講義や演習が可能になってきた[8]。ここでは、AIによるプログラミング演習に焦点をあてて、関連研究を紹介する。

OpenAI Codexの強力なコード生成能力は、研究者らの関心を引きつけ、大規模言語モデルの教育活用を調査してきた。オークランド大 James Finnie-Ansley[9]らは、プログラミング入門問題の解答能力を調べ、学生の平均能力を上回っている事実を明らかにした。フィンランドのアルート大の Juho Leinonen[10]らは、Codexを用いてエラーメッセージの解説がより初学者にわかりやすくなることを示した。米テンプル大の Stephen MacNeilら[11]は、GPT-3を用いてさまざまなコード解説を生成する試みを行っている。Majeed Kazemitabaar[12]は、Codexを活用し、初学者のコード作成を助ける支援システムの構築を行っている。

7. まとめ

本論文では、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報告を行った。KOGIは、Jupyterカーネルのエラー処理と統合され、ユーザからは完全にシームレスに活用することができる。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。

参考文献

[1] Kölling, M., Brown, N. C. C., Hamza, H. and McCall, D.: Stride in BlueJ – Computing for All in an Educational IDE, *Proceedings of the 50th ACM*

Technical Symposium on Computer Science Education, SIGCSE '19, New York, NY, USA, Association for Computing Machinery, p. 63–69 (online), DOI: 10.1145/3287324.3287462 (2019).

[2] 萩谷昌己: 情報教育の格差と、情報学分野の参照基準: 情報教育の基盤となる学問としての情報学, 情報管理, Vol. 59, No. 7, pp. 472–478 (2016).

[3] Munson, J. P. and Schilling, E. A.: Analyzing Novice Programmers' Response to Compiler Error Message s, *J. Comput. Sci. Coll.*, Vol. 31, No. 3, p. 53–61 (2016).

[4] McCall, D. and Kölling, M.: A New Look at Novice Programmer Errors, *ACM Transactions on Computing Education*, Vol. 19, pp. 1–30 (online), DOI: 10.1145/3335814 (2019).

[5] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A. and Fung, P.: Survey of Hallucination in Natural Language Generation, *ACM Comput. Surv.*, Vol. 55, No. 12 (online), DOI: 10.1145/3571730 (2023).

[6] Akinobu, Y., Obara, M., Kajiura, T., Takano, S., Tamura, M., Tomioka, M. and Kuramitsu, K.: Is Neural Machine Translation Approach Accurate Enough for Coding Assistance?, *Proceedings of the 1st ACM SIGPLAN International Workshop on Beyond Code: No Code*, BCNC 2021, New York, NY, USA, Association for Computing Machinery, p. 23–28 (online), DOI: 10.1145/3486949.3486966 (2021).

[7] Obara, M., Akinobu, Y., Kajiura, T., Takano, S. and Kuramitsu, K.: A Preliminary Report on Novice Programming with Natural Language Translation, *IFIP WCCE 2022: World Conference on Computers in Education* (2022).

[8] Joyner, D. A.: ChatGPT in Education: Partner or Pariah?, *XRDS*, Vol. 29, No. 3, p. 48–51 (online), DOI: 10.1145/3589651 (2023).

[9] Finnie-Ansley, J., Denny, P., Becker, B. A., Luxton-Reilly, A. and Prather, J.: The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming, *Proceedings of the 24th Australasian Computing Education Conference*, ACE '22, New York, NY, USA, Association for Computing Machinery, p. 10–19 (online), DOI: 10.1145/3511861.3511863 (2022).

[10] Leinonen, J., Hellas, A., Sarsa, S., Reeves, B., Denny, P., Prather, J. and Becker, B. A.: Using Large Language Models to Enhance Programming Error Messages, *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, SIGCSE 2023, New York, NY, USA, Association for Computing Machinery, p. 563–569 (online), DOI: 10.1145/3545945.3569770 (2023).

[11] MacNeil, S., Tran, A., Mogil, D., Bernstein, S., Ross, E. and Huang, Z.: Generating Diverse Code Explanations Using the GPT-3 Large Language Model, *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 2*, ICER '22, New York, NY, USA, Association for Computing Machinery, p. 37–39 (online), DOI: 10.1145/3501709.3544280 (2022).

[12] Kazemitabaar, M., Chow, J., Ma, C. K. T., Ericson, B. J., Weintrop, D. and Grossman, T.: Studying the Effect of AI Code Generators on Supporting Novice Learners in Introductory Programming, *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, Association for Computing Machinery, (online), DOI: 10.1145/3544548.3580919 (2023).