

マルチターンで学生の問いを解決するプログラミング学習支援チャットボットの試作

メタデータ	言語: Japanese 出版者: Musashino University Smart Intelligence Center 公開日: 2024-03-25 キーワード (Ja): チャットボット, 教育支援, マルチターン, 対話分析 キーワード (En): 作成者: 上地, 泰彰, 田丸, 恵理子, 渡邊, 紀文 メールアドレス: 所属:
URL	https://mu.repo.nii.ac.jp/records/2000202

一般論文

マルチターンで学生の問いを解決する プログラミング学習支援チャットボットの試作

Towards a Programming Learning Chatbot Utilizing Multi- turn Problem Solving Methodology

上地泰彰†‡, 田丸恵理子‡, 渡邊紀文‡

Forbital† 武蔵野大学‡

概要

プログラミング学習を支援するためにチャットボットの開発を進めているが、プログラミングは多段階の問題解決プロセスであり、一問一答型のチャットボットでは学習支援としては不十分である。本文では、問題解決に向けた対話を進めるマルチターン方式による学習支援チャットボットの開発について展望を述べる。まずは学生の質問内容や教員との対話データを活用し、より適切な回答を生成できるシングルターンのチャットボットを作成・運用し、学生とチャットボットとの対話の特徴や課題を抽出した。さらに今後のマルチターン型チャットボットの開発へ向けて、学生と教員のオンラインセッションのトランスクリプトの分析に着手し、教員が学生を問題解決に導く対話の構造に関し、初期的な示唆を得た。

キーワード: チャットボット, 教育支援, マルチターン, 対話分析

1. はじめに

近年、AI（人工知能）を用いたチャットボットが、さまざまな分野で活用されている。特に教育分野では、学生の疑問解消や学習支援を目的としたチャットボットの導入が進んでいる。プログラミング教育においては学生の問題解決スキルを向上させることが重要とされており、単一の質問応答だけでなく問題解決プロセス全体を支援するチャットボットの開発が求められている。

プログラミング学習における問題解決は多段階的なプロセスを必要とする。具体的には、問題の明確化、抽象化、アルゴリズムの導出、具象化というステップが含まれる。従来の質問応答型チャットボットではこれらの多段階的なプロセスに対応することが難しく、またそれぞれのステップで必要な支援内容も異なるため、効果的な学習支援を行うことが困難であった。

そこで本プロジェクトでは、マルチターン方式によるステートフルな会話形式のチャットボットを提案する。マルチターン方式とは、一問一答だけでなく複数回のやり取り（ターン）を通じてボットが能動的に情報を獲得し、問題解決に向けた対話を進める方式である。学生自身も複数ターンを通して自分の抱えている課題をより深く理解し、それらに対する

対処法を見つけだすきっかけとなる。また、ステートフルとは、各学生の以前の会話履歴や周辺情報をボットが取得し、回答に役立てるという意味である。これは、以前の会話履歴が参照されないステートレスに対置する概念である。これらにより、学生の問いに対してより適切な支援が可能と考えられる。

本プロジェクトの主な目的は、マルチターン方式のステートフルなチャットボットが、一問一答形式のステートレスなチャットボットよりも学生の理解促進と問題解決に効果的であるという仮説を検証し、実際の学習支援において有効であることを示すことである。本稿ではそれらの機能の有効性を確認するために、主に対話データの分析と、質問応答チャットボットを開発した、具体的には、プログラミング授業における学生と教員間のオンラインセッションを分析し、会話のデータセットを作成する。さらに分析から得られた対話のパターンをもとに、武蔵野大学の「プログラミング基礎」の授業に対応した質問応答方式のチャットボットを開発し、その性能を実際のクラスで運用することで評価する。

2. 教育分野におけるチャットボットの活用

2.1 既存のチャットボットの事例

教育分野においても AI 及びチャットボットの活用が進んでいる。オンライン学習プラットフォームを運営する Khan Academy は、生成 AI を活用したチャットボット「Khamingo」[1]を開発し、自力では解けない難しい問題にぶつかった学生をサポートする機能を提供している。また、厳密にはチャットボットではないが、駿台予備校が提供する教育支援システム「スルメ」[2]では、膨大なデータベースと機械学習を活用して学生一人ひとりに最適化された「アダプティブヒント」を提供することで、すぐにヒントに頼ることなく自分の力で問題解決を進める手助けを行っている。

プログラミング教育においては、学生が問題解決能力を身につけることが重要であり、そのためには学生の質問に直接的な答えを与えるのではなく、どのように問題を解決すれば良いか考える力を引き出す指導が必要である（魚ではなく、釣り方を教える）。さらに、プログラミング教育ではエラーやバグといった問題に直面することも多いため、それらの失敗を恐れない姿勢や挑戦し続ける精神も育てるべきである。

このように、プログラミング教育と一口に言ってもその方法論とカバレッジは多様である。教育エンジニアリングの答えとは、単なる Q&A ボットなどといった局所解で満足されるものではなく、より核心的な「学生の問題解決」という命題に AI 活用と織り交ぜて答え続けていく、より大きな営みの先にある。この時流の中で取り扱われるトピックの中に、駿台のアダプティブヒントニングや、本プロジェクトが目指すマルチターン形式による学生の理解支援がある。

2.2 シングルターンとマルチターンの違い

シングルターンとマルチターンとは、チャットボットの対話形式を表す用語である。シングルターンとは、一問一答のやり取りが基本となる形式であり、学生が質問を投げかけるとチャットボットがその質問に対する答えを返す、という流れを基本とする。一方、マルチターンはその名の通り複数回のやり取り（ターン）で情報を蓄積し、問題解決に向けた対話を進める形式である。例えば学生が「このエラーメッセージが理解できない」と投げかけた場合、シングルターン方式のチャットボットは「それは〇〇のエラーです。△△することで解消できます」と直接的な解答を与えてしまう。

しかしマルチターン方式では、「エラーメッセージは何ですか?」、「どのような操作を行った時にそのエラーが出ましたか?」等、学生の状況や課題理解を深めるために追加的な質問を行う。そしてそれらの回答に基づいて最適なアドバイスを提供することで問題解決を図る。また、学生はこれらの質問に回答する過程で自分の抱えている問題をより深く分析する機会が与えられる。このようにマルチターン方式は一問一答形式よりも深い対話が可能となり、学生の問題解決に対する理解と支援がより具体的に行える可能性がある。

2.3 マルチターンチャットボットの先行研究

LLM のマルチターン会話性能を客観的に評価するベンチマーク及びデータセットに MT Bench [3]がある。このデータセットにはユーザーの質問とそれに対するアシスタントの解答、それに対するユーザーのフォローアップ質問とそれに対する解答が会話の1単位として含まれている。マルチターン性能は質問解答ペア（ターン）ごとに人間が数値評価をし、最終評価は2つのターンの平均値で評価される。このベンチマークを用いることで、LLM のマルチターン性能をある程度評価・比較することが可能となる。

3. 武蔵野大学 AI 副専攻でのプログラミング教育とチャットボットの活用

3.1. AI 副専攻における教育とサポート体制

武蔵野大学では、データサイエンス学科を除く全学科を対象とした、データサイエンスおよび人工知能の活用について学ぶ副専攻（AI 活用エキスパートコース：AI 副専攻）を開講している。さまざまな学科の学生が履修しているため、各学生のデータサイエンスや人工知能に関する知識やスキルはそれぞれ異なっており、それらの学生をサポートする体制の構築をこれまで進めて来た[4]。特にプログラミング科目は、2020年度の授業開講時より、授業の説明および演習すべてをオンラインで実施し、コラボレーションツールの一つである Microsoft Teams を利用して学生が気軽に質問をしたり、個々の学生が自身の画面を共有し

ながら、教員のサポートを受けることができる Web 会議システムを導入し、個々の学生のサポートを充実させる環境を構築した[5-6]。これらのサポート体制を更に充実させたものとするため、今回提案するマルチターン型のチャットボットを授業に取り入れることとした。

3.2. AI 副専攻でのプログラミング教育

AI 副専攻では、従来の教育で行われてきた知識やスキルを獲得することを重視した教育に加えて、そこで得たものを利用して自らのアイデアを実現するアウトプット型の教育を重視している。そこでアウトプットをするツールの一つとして、プログラミングを教育している。

AI 副専攻を履修する 1 年生後期科目「プログラミング基礎」は、授業を週 2 コマまで開講し、1 コマは知識やスキルを自ら獲得するオンデマンド型の授業を、1 コマは Web 会議システムを利用してオンライン上で教員のサポートを受けながら同時双方向型の授業を実施している。オンデマンド授業では、教員が作成した單元ごとの授業資料で、プログラミングに関する問題解決手法（問題の分解、パターンの発見、抽象化、手順化など）について学ぶ。更にプログラミングの開発環境として、ブロック型言語で開発が可能な Microsoft Makecode を利用した「Minecraft Education」を用いて、実際にプログラムを作成し、動作を確認しながら学ぶ。更に同時双方向授業では、オンデマンドで獲得した知識やスキルに基づいて、教員が作成した單元に基づいた演習を実施する。演習では、個々の学生の躓きを、実務経験のある非常勤講師を含めて 20 人の学生に 1 人の体制でサポートをし、更に授業後には任意参加で、オンライン上で個別に非常勤講師に質問をしたり、アドバイスを受けることができる体制を構築した。

3.3. プログラミング科目でのオンラインサポート体制

オンラインでの学生とのやりとりは、学生の表情や声などのコミュニケーション不足により、サポートが困難であると考えられているが、一方でオンラインならではのサポートも可能となる。その一つが、サポート内容の教員間の共有である。

AI 副専攻で開講した「プログラミング基礎」では、授業時間外での教員による学生のサポートを Microsoft Teams の Web 会議システムを利用した。Web 会議では、学生に PC の画面を共有させながら口頭でサポートをし、そこでの様子をすべて録画した。更に Teams のトランスクリプト機能を利用し、すべての対話データを学生と講師とで話者分離をした形で文字起こしをした。これらの対話データを利用することで、授業を担当する教員は、学生がどのようなサポートを講師から受けたのかを確認することが可能となり、その内容を他の学生、また次回の授業でフィードバックをすることが可能となった。



図 1. Web カルテの検索結果

またサポートをした非常勤講師は、学生に対してどのようなサポートをしたのかを要約し、教員および他の非常勤講師とも情報を共有した。具体的には図 1 のような Web カルテを用意し、学生の質問やそれに対する回答、また学生がどのようなところに躓いていたのか、更にその場では解決できなかった場合は、それもコメントをし、授業を担当する教員や他の非常勤講師がフィードバックできるようにした。これらの情報共有により、非常勤講師がサポートをする前に、現在どのような質問が出ているのか、また個々の学生がどのような学習状況にあるのかを確認してサポートをすることができるようになった。

3.4. チャットボットを利用した学生支援

オンラインでのサポートは、個々の学生の質問を直接聞くことができるため、学生にとってはすぐに問題を解決することが可能であり、講師にとっては学生と対話をすることで解決に導くことが可能である。しかし直接の対話では、24 時間いつでもサポートをするという事は困難である。また講師に質問することに戸惑いを持ち、積極的に利用できない学生もいる。そこで、AI 副専攻のプログラミング科目では、授業内容や課題について質問が可能なチャットボット開発を進めることとした。

このチャットボット開発において、オンラインサポートで得られたトランスクリプトおよびカルテは重要である。これらのデータには、学生がどのような質問をし、それに対して非常勤講師がどのような回答をしたのかが書かれているため、それを利用したシングルターンでのチャットボットを作成することが可能である。特に授業は段階を追って単元が進

むため、学生が質問をする内容は時系列で変化し、質問は特定の内容に集中する。そのため、特定の単元や課題の情報が含まれているカルテ、またトランスクリプトの中でも単元に含まれるキーワードや、課題の番号などが含まれているため、これらのデータを利用してよりの確な回答をすることが可能となる。

またマルチターン型のチャットボット開発においては、オンラインで非常勤講師がサポートをした、トランスクリプトの内容が重要となる。講師は、学生の質問に対して直接回答をするのではなく、学生が抱える問題を解きほぐし、段階を追って問題を解決して回答に導く。これらのプロセスや内容がトランスクリプトに含まれているため、それらの構造やポイントを明らかにすることで、マルチターンでの回答を生成することが可能となる。また学生のカルテも、個々の学生の質問の内容の変遷を学習することで、長期的なターンでの回答も可能になると考えられる。

これらの目的を元に、プロトタイプとして開発したチャットボットの2023年度「プログラミング基礎」科目での活用事例について、次章で説明する。

4. プログラミング基礎でのチャットボットの活用事例

2023年度は、チャットボットという媒体が学生のプログラミング学習に対してどれほど効果的なのか、どのような影響を与えるのかを検討するために、武蔵野大学で開講した「プログラミング基礎」において、生成AIを活用した質問応答チャットボット「SimpleQA」を開発・運用した。学生が使用した期間は2023年12月21日から2024年2月12日であり、主に配列の課題やミニプロジェクトと呼ばれるプログラムの最終成果物の作成時に利用するように案内をした。本章では、システムの大まかな構成と、実際に運用して得られた知見を紹介する。

4.1 SimpleQA の概要

SimpleQA はチャットインターフェースを備えた Web サービスであり、武蔵野大学の Google アカウントを持つユーザーのみがアクセスできる。SimpleQA の構成図を図2に示す。全体的には関数呼び出し機能を持つ LLM を用いた RAG (Retrieval Augmented Generation) 型チャットボットである。「プログラミング基礎」の講義資料がベクトルデータベースにインデックスされている。LLM はユーザーの質問の内容によってデータベースを参照すべきかを判断する。データベースを参照すべきと判断した場合は、最適な検索クエリを自ら考えデータベースから関連する資料の断片を取得し、その結果をベースに最終的な回答を出力する。このように、いつデータベースを参照するのか、あるいは参照しないのかを LLM 自身に任せるやり方を関数呼び出し (Function Calling) と呼び、外部データベースの知識を用いて回答の正確性を担保する方式を RAG と呼ぶ。

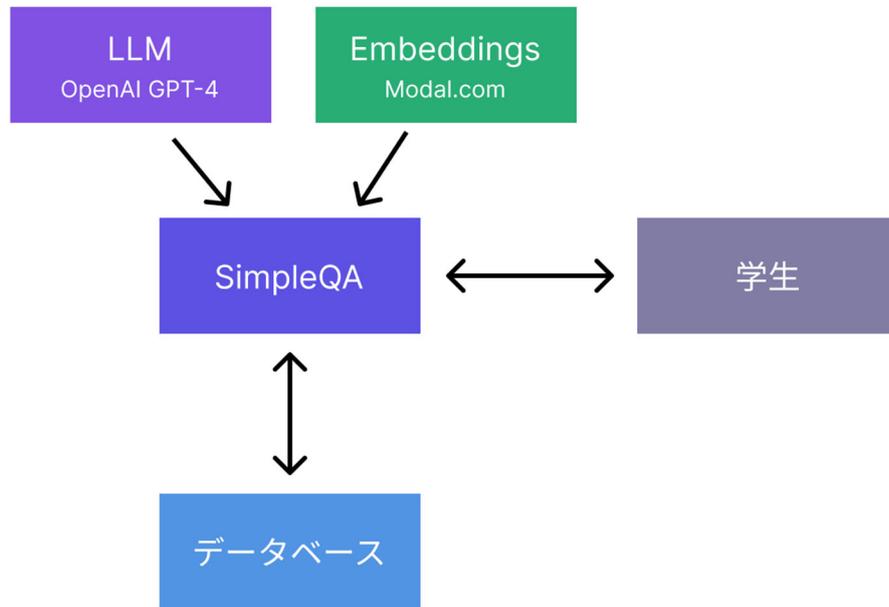


図 2. SimpleQA のシステム構成

1) システムプロンプト

SimpleQA の最終的なシステムプロンプトをリスト 1 に示す。プロンプトの内容は、「クラスの説明」「想定質問の列挙」「期待されるペルソナの指定」「関数呼び出しの説明」「検索ノイズに対する注意」「ハルシネーション対策のプロンプト」で構成されている。

図 3 に以上のシステムプロンプトを利用したボットによる回答例を示す。

あなたは *MakeCode for Minecraft Education* を利用した武蔵野大学のプログラミング基礎クラスのアシスタントです。プログラミング初心者の学生からのブロックプログラミング、講義、課題に関する質問に対して TA として正確かつ適切な対応をなささい。授業固有の情報や課題に関する質問には必ず資料検索をして、資料名を引用して回答してください。ただし、検索結果には関係ない情報が混入している場合があるので慎重に検討なささい。正確な回答ができない場合には「Teams のオンライン質問チームで聞いてね」と返事なささい

リスト 1. SimpleQA のシステムプロンプト

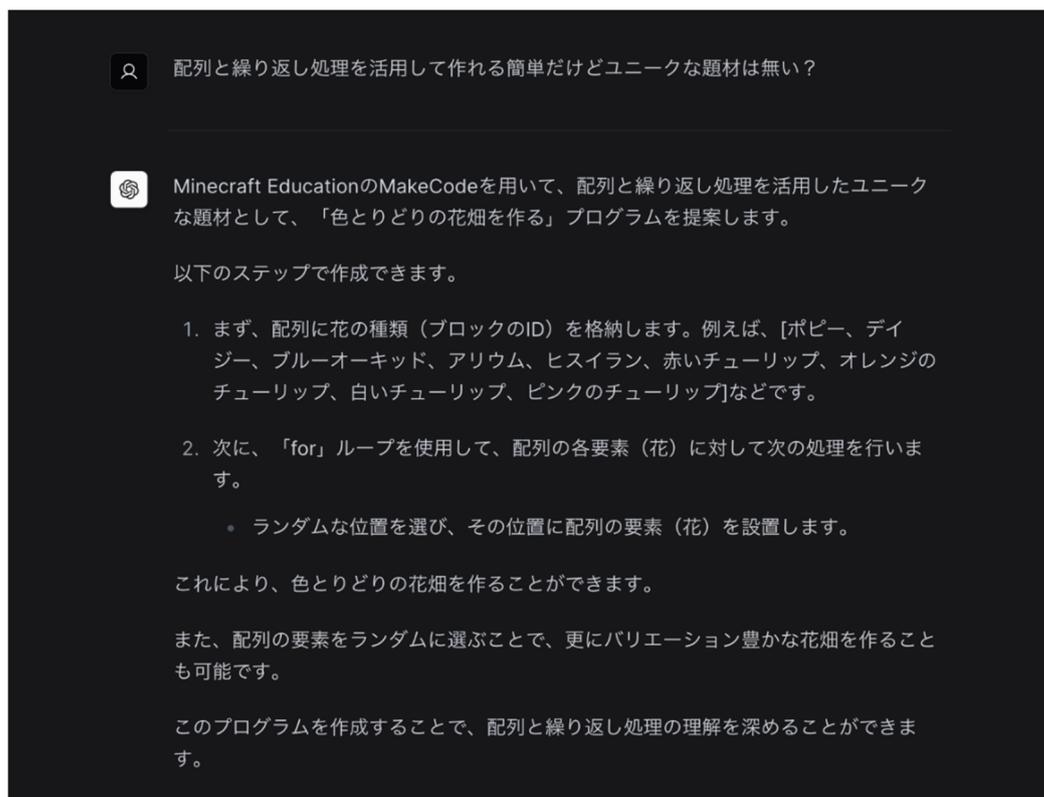


図 3. SimpleQA の回答例

2) 逆質問

2023年12月21日にチャットボットの運用を始めてから一週間データを収集した結果から分析を行った結果、学生の質問はLLMにとって回答しやすい形式ではないことが多いことがわかった。つまりコンテキストが完備された質問ではないということである。例えば、「山の作り方を教えてほしい」と学生がチャットボットに質問をした場合、具体的にどのような形状の山なのか、どのような意図で山を作りたいのか、といったコンテキストについての情報が必要である。

そこで、「文字数が極端に少ない質問」に対しては特殊な対応をすることにした。当初はシステムプロンプトに「曖昧または漠然とした質問に対しては逆質問をせよ」等という指示を挿入したが、このようなやり方でLLMが実際に逆質問をするケースは一件も観測されなかった。試行錯誤の結果、システムプロンプトではなく、ユーザーの質問そのものを編集するやり方が効果的であることを発見した。そのコードをリスト2に示す。

短い(isShort)質問には「下の質問がありますが、自分でもどのような段階を踏めば良いのかわからないです。質問の回答に必要な情報を尋ねてください。」という文章を先頭に追記し、そうではない質問には単に「質問があります。」という文章を追記している。この方式を採用した後に観測された実際の逆質問の例を図4に示す。

```
const realPrompt: ChatCompletionMessageParam[] = [
  {
    role: 'system',
    content: systemPrompt
  },
  ...previousMessages,
  {
    role: 'user',
    content: (isShortReply
      ? '以下の質問がありますが、自分でもどのような段階を踏めば良いのか
      わからないです。質問の回答に必要な情報を尋ねてください。 ¥n' +
      userInput
      : '質問があります。 ¥n' + userInput) as string
  }
]
```

リスト 2. 学生の質問の長さに応じて、質問内容を編集している例

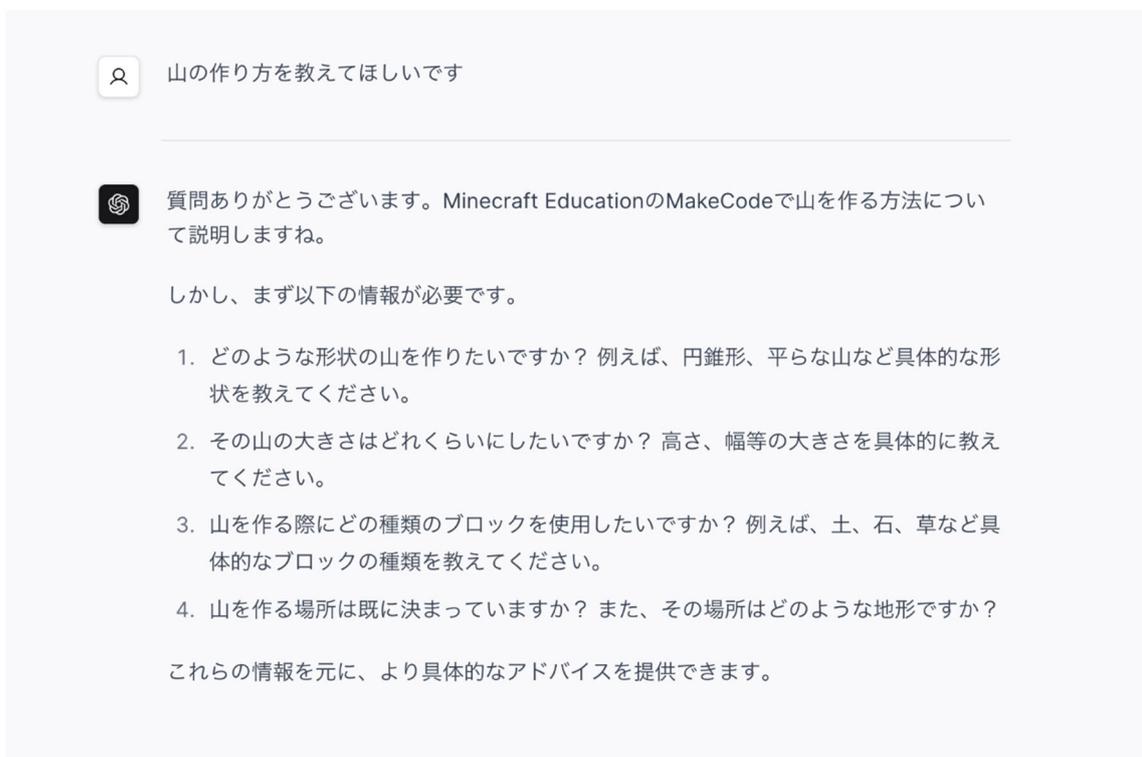


図 4. チャットボットによる逆質問の例

4.2 学生の行動解析

表 1 に示すように、前述の利用期間で合計 17 人の学生による利用があり、29 セッションにまたがり合計 98 回の発言が確認された。

表 1. 収集した対話データの基本データ

合計セッション数	29
合計発言数	222
合計学生発言数	98
学生利用者数	17
学生あたりの平均対話数	13.1
平均対話数の標準偏差	17.5

学生あたりの平均対話数は 13.1 であるが、その標準偏差は 17.5 と大きい。これは Zipf の法則がここにおいても現れているということである。具体的にはトップ利用者 2 名の対話数がそれぞれ 62 回と 53 回、合計 115 回となっていて、全体の 52% を占めている。

最も持続したセッションでは、33 回、30 回、29 回の対話が行われ、どれも上位 2 名によるものである。

1) 学生のとる行動の分類

チャットボットを利用するユーザーの行動を構造化する試みには、Nielsen Norman グループによるものがある[7]。筆者らは前節で示した学生の対話データを分析し、学生の質問タイプを以下の 3 グループに分類した。

■ブレインストーミング型（ファンネリング型）

- ・ 漠然としたアイデアからより粒度の細かいアイデアへと深化させる試み
例: 「配列や繰り返しを使って、スロットを題材にしたプログラムを作りたい」「スロットの各リールがすべてそろった時に、特殊な演出をしたいからアイデアを教えて」

■情報検索型

- ・ 講義資料やインターネット上にある特定情報をピンポイントで索引する試み
例: 「配列のコマンドについて知りたいです」「座標について理解したいんだけど、どの動画を見れば良い？」

■ショートサーキット型

- ・ 問題文そのもの、または課題のコアとなるタスクをそのまま LLM に解かせようとする試み
例: 「エージェントが土台 10*10 ブロックから始まる、葉っぱブロックのピラミッドを作るプログラムを作って」

4.3 チャットボットの運用における課題

1) ハルシネーション

LLM の先天的な欠点として「嘘をつく」という性質がある。これを緩和する策として RAG が登場したのであるが、データベースから関連する情報が返却されなかった場合は、やはり嘘をついてしまう。これを緩和するために、筆者らはシステムプロンプトに「正確な回答ができない場合は～」という節を導入したのであるが、実際に運用した結果、残念ながらこれが機能する場面はほとんどなかった。

Air Canada が、自社の AI チャットボットによるハルシネーションによってユーザーに被害を与えた事件[8]は、大企業であっても如何にハルシネーションの阻止が困難かつ非自明なチャレンジであるかを物語っている。

2) 直接回答

上記で述べたショートサーキット型質問に対して、SimpleQA は直接回答を与えようとしてしまい、学生の学習機会を奪ってしまうことがある。これに対抗するために、当初はシステムプロンプトに「課題の質問には直接回答せず、ヒントを与えよ」等と記述した。しかし、結果としてチャットボットはどのような質問に対しても回答を拒否してしまい、上記の情報検索型質問に対して有効な回答ができなくなってしまった。

3) 返答時間

今回のチャットボットは図 2 に示したように適切な資料を適時データベースから検索する方式を取っていた。その結果、質問内容によっては情報検索と精査に 1 分以上要する状況がまれに発生することがあり、その場合学生は回答を待ちきれずにページをリロードしたと思われる事例が 4 件確認された。ただし、その判定条件は 1) 前セッションと後セッションの間隔が 1 分以内であり、2) 前セッションと後セッションで質問内容が一致しているものとする。

回答生成中にアニメーションを表示したり、生成に時間がかかることを案内したりすることで不安感を解消するよう努めているが、根本的な問題は関数呼び出し型かつデータベース参照型ボットは LLM との往復回数が複数回に及ぶことである。LLM 部分が外部サービスに依存していることもあって、LLM 応答時間自体の改善は難しいため、キャッシュ等を用いた関数呼び出し型の改善、または全く異なる手法の模索が必要である。

5. トランスクリプトの構造分析

本章では、オンライン質問対応における教員と学生の対話の記録であるトランスクリプトの分析について述べる。筆者らは、質問応答型のチャットボットではなく、プログラミングの学習を支援する対話型のチャットボットの構築を目指している。プログラミングを学

ぶということは、単にプログラムコードを書けるようになることではない。達成したい目的があり、解決のための手段を検討し、それを実現するロジックを考え、そのロジックを手続きに変換してプログラミング言語で表現するという、多段階の問題解決を学ぶことである。このような問題解決能力やプログラミングスキルの学習を支援するためには、学生の状態や文脈を適切に把握し、対話を通じて段階的に問題解決へと導いてくれる助言者のような存在が有益であろう。本文ではこのようなマルチターン方式での対話型チャットボットを提案しているが、実際に問題解決に至るための適切な対話とはどのようなものであろうか。

本学のプログラミング授業では、課題で困難を生じた学生のオンラインの質問に教員が回答する学習支援の仕組みを提供している。ここでは、トラブルに陥った学生は教員に問い合わせを行うが、教員はこの質問に直接回答するのではなく、学生の基本知識レベルや課題の理解状況、進捗状況などを把握しつつ、学生の理解が促進するような質問や助言を与え、段階的に問題解決まで導いている。学習支援チャットボットを構築するにあたり、人間の教員がどのような段階を踏み、どのような対話を通じて助言を与え、学生の問題解決を支援しているかを理解することは、マルチターン型のチャットボットを構築する上で、有益な示唆を与えてくれるであろう。そこでプログラミング基礎という授業でのオンライン質問のトランスクリプトに関して、対話の構造や解決に導く適切な助言など対話の特徴に関して分析を行っている。現在はまだ分析途中であり、今回は速報レベルの報告となるが、現段階で見えてきた対話の特徴について報告する。

5.1 分析データについて

今回分析対象としたのは、プログラミング基礎という科目で Minecraft のプログラミング環境を使いながら、プログラミングの基礎を学ぶ科目である。全 7 週の中でも特に学生が躓きやすい、繰り返しや条件文が出てくる第 3 週の課題に絞り、そのトランスクリプトを分析した。教員は 1 コマ 100 分の対応時間の中で複数人の学生をサポートするが、これらが 1 つのトランスクリプトファイルとして記録される。したがってトランスクリプトファイルは学生毎に分割し、1 つのファイルには 1 人の学生との対話が記録されるようにデータ化した。助言を受けた後、一旦オンライン会議室を退出するが、同一時間帯内に再訪して再度助言を受けた場合は、同一対話として 1 ファイルに記録した。一方、一連のトラブルが継続しているが、時間帯を跨ぎ別の教員に対応が引き継がれた場合は、別対話として扱った。2021 年から 2022 年度にかけて蓄積されたデータを分析中であるが、今回の報告は、そのうち 11 件分のデータに関する結果である。

5.2 分析方法

学生の問い合わせに対する対話構造の特徴を理解するために、トランスクリプトの対話

データに対して次のようなラベリング作業を実施した。ラベリングは、実際に質問対応を行った教員2名に依頼した。分析品質を一定に保つため、本人が担当した分だけではなく、他の教員が担当した対話に関してもこの2名でラベリングを実施してもらった。

1) チャンク化と要約

トランスクリプトをよく読みいくつかの意味のある単位に分割した。分割単位はかたまりの大きさ順に「セクション」→「ブロック」→「センテンス」である。表2に各チャンクの概要を示す。さらに各チャンクに関する要約を100～200文字程度で作成した。

2) ラベリング

対話のブロックとセンテンスに対してラベルを付与した。ブロック単位のラベリングでは「問いの明確化」と「問いの解消」の2つのラベルを基本とし、その他は分析者に任意につけてもらった。一方センテンス単位のラベリングに関しては、一連の対話のセンテンスの中でも、問題解決にとって重要なセンテンスを抽出し、そのセンテンスが問題解決の中でどのような意味のある発話かに関してラベルを付与した。ラベルは表3の7種類を準備した。ラベリングの分析例を図5に示す。

表2. チャンクの種類

チャンク名	概要説明
セクション	1つの問題解決に関わる一連の対話のかたまり。1つの対話中で複数の問題解決が行われたり、複数のサブ問題に分割した上で、問題解決が行われたりする。
ブロック	1つの問題もしくはサブ問題を解決するためには、問題を明確化する段階、問題を解消する段階など、段階的に対話が進められる。ここでは段階毎の1つの意味を成す対話シーケンスのかたまりをブロックと呼ぶ。1つのセクションは複数のブロックから構成される。
センテンス	通常は意味を成す1つの文章を指すが、ここでは Teams によって自動的に区切られた1行分のデータを指す。

表3. ラベルの種類

ラベル名	概要説明
問い合わせ	学生からの質問や相談内容に関わるセンテンス
問題解決	最終的に問題が解決された箇所のセンテンス
問題の明確化	問い合わせ内容に対して対話を通じて問題を明確化していく過程にかかわるセンテンス
明確化された問題への回答	明確になった問題に対しての回答が示されたセンテンス
教員の問いかけ/助言	対話シーケンスの中で、問題解決に繋がった、あるいは学生の理解を促進させた重要な問いかけや助言などのセンテンス
学生の質問/回答	上記に対応する形での問題解決に繋がった、あるいは理解が進んだ学生の質問や回答/反応に関わるセンテンス
その他	上記ラベルに該当しないが問題解決に重要と評価されたセンテンス

表 4. 対話の構成概要

ファイル名	対話数	ラベル数	セクション数	ブロック数
ファイル 1	652	390	5	12
ファイル 2	212	76	2	3
ファイル 3	97	43	2	4
ファイル 4	98	35	2	3
ファイル 5	150	54	2	4
ファイル 6	515	183	5	8
ファイル 7	391	182	3	7
ファイル 8	507	197	5	11
ファイル 9	362	238	4	8
ファイル 10	281	112	2	5
ファイル 11	393	207	3	7
平均	332.5	156.1	3.2	6.5

1つの問いに対して一気に解決に至るのではなく、教員が問題解決のステップをいくつか分割し、段階的に学生を問題解決に導いていることを示唆している。変型パターンとしては、「問いの明確化」が発生しないケースがあり、これは授業中の課題の取り組みからオンライン質問へと継続して実施したケースや、同じ質問に対する再訪のパターンなどがある。さらに「問いの明確化」が続くパターンも見られた。これは大きな問題を明確化した後、もう一段階詳細な問題を明確化する対話が始まるようなパターンである。

その他のラベルでは、問い合わせに関する問題解決はなされたものの、この先躓くかもしれないという箇所に対する助言などが行われ、教員が学生の「問いに回答する」というだけではなく、学生が「課題を達成できる」ための助言をしている様子が見られた。

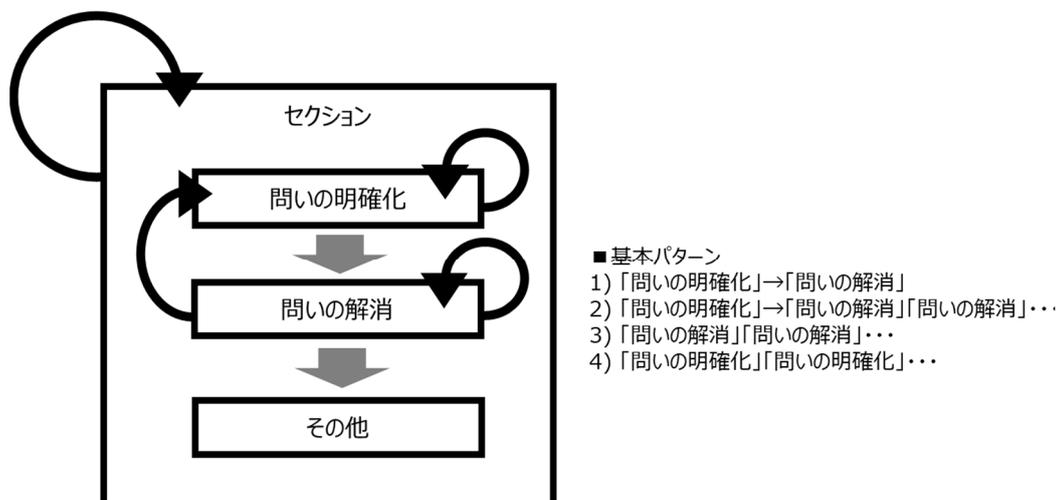


図 6. 対話の構造

3) 教員と学生の対話の特徴

問いを明確化する過程と問いを解消する過程で教員と学生の対話の特徴に差異があるかを見るため、ブロック種別毎に「教員の問いかけ/助言」「学生の質問/回答」を比較した結果を表5に示す。

表5. ブロック種別の教員と学生発話数

ブロック種別	教員の問いかけ/助言		学生の回答/反応	
	総数	割合	総数	割合
問いの明確化	196	61.1%	125	38.9%
問いの解消	668	66.0%	344	34.0%

どちらのブロックでも教員のほうが重要なセンテンスの割合が高い傾向があり、基本的には問題解決は教員がリーディングしている傾向がみられる。図7にブロック種別毎の「教員の問いかけ/助言」「学生の質問/回答」のラベル数の散布図を示す。単純な総数の割合では問いの明確化と解消のブロック種別の間には極端な差異は見られなかったが、散布図からは、問いの明確化においては、教員と学生の対話が対等なグループと教員がリーディングして問いを明確化していくグループに分かれた。一方で、問いの解消に関しては、教員と学生が対等な対話もあるものの、全体としては教員が対話をひっぱり、かつそのバラツキが大きい傾向が見られた。

これらの対話から「教員と学生が対等な対話」「教員がリーディングしている対話」の具体例をいくつか示す。

【対話例1】問いの明確化：教員と学生が対等

教員と学生のラベル数が26:30で、両者がほぼ対等に対話を行っているケースである。課題はMinecraftで色違いのブロックを15個並べるプログラムを作ることだが、同じ色が並んでしまうというトラブルの問い合わせである。この課題はstep1~3まであり、問い合わせはstep2に関わるものである。教員は当初step1のコードを応用してstep2を作らせよう

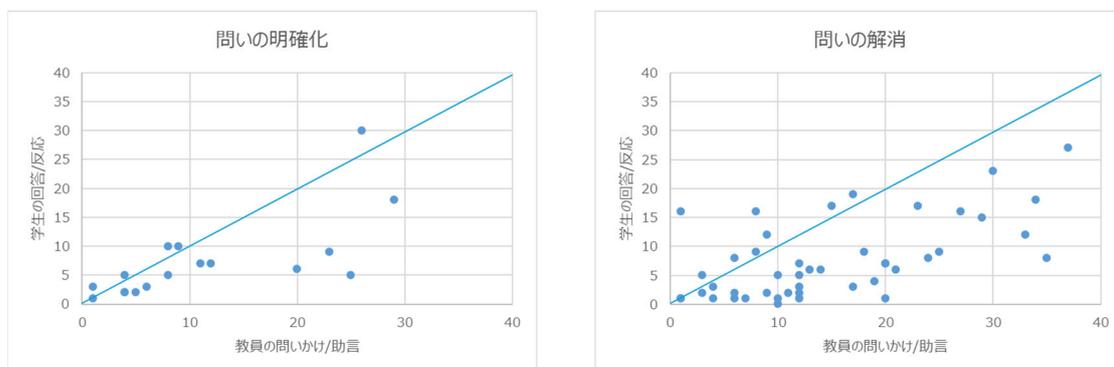


図7. ブロック種別毎の教員と学生の対話の特徴

としたが、step1 のコードを学生なくしてしまいコードの編集ができないことが判明。問題解決の方針を転換し、現状の step2 のコードを修正して問題解決を図ることとした。このように学生のトラブル状況の確認から学生の課題の進捗状況に応じた問題解決の方向性を見定めるために適宜質問をし、学生が都度回答するという対話が行われた。

【対話例 2】 問いの解消：教員と学生が対等

教員と学生のラベル数が 30:23 で、両者がほぼ対等に対話を行っているケースである。Reminder という関数を使ってエージェントが何歩か歩く毎にブロックの色を変えるというプログラムがうまく動かないという学生からの問い合わせである。これに対して、まずは学生の理解レベルを確認したうえで、学生に問いかけをしながら学生に答えさせ、あるいは手を動かしてプログラムの挙動を確認する対話を繰り返し問題解決まで至った。学生はこの対話の中で自分のわからない点を教員に伝えたり、自身の状況の認識や次にどうすべきかに関して、自分なりの意見を積極的に説明した。

【対話例 3】 問いの解消：教員が問題解決をリーディング

教員と学生のラベル数が 35:8 で、教員が問題解決をリーディングしている対話パターンである。対話例 1 と同様に step2 に関わる問い合わせである。教員は学生のコードを確認し、コードをどのように修正したら良いかをステップバイステップで説明及び指示をする。これに対して学生は「はい」などの応答はするが、積極的に自分の考えなどを発言することはほとんどなく、指示されたとおりに変数を追加したりコードの修正を行うなどして問題解決に至った。

以上、現段階での初期的な結果を示した。同じ問いの解消の対話でも、対話の特徴が異なることがわかる。現状ではこれらの特徴が何に起因しているかを抽出するに至ってはいないが、今後は分析例を増やしつつ詳細な分析をすすめ、学習支援のための教員と学生の対話の特徴やあり方を明確にしていく予定である。

6. おわりに

本文では、プログラミング学習を支援するマルチターン型のチャットボットの開発に向けて、現状の取り組みとその課題について述べた。武蔵野大学の AI 副専攻プログラミング科目において、学生のオンライン質問対応を通して蓄積されたトランスクリプトやカルテデータに基づき、単一質問応答型の SimpleQA チャットボットを開発し、初期的な運用を行った。その過程で明らかになった課題として、ハルシネーションの抑制や、学生にヒントを与えるのではなく直接回答を与えてしまう問題などが挙げられた。

更に、トランスクリプトデータの対話構造分析により、教員が学生の理解を促進するため

の対話のあり方について初期段階での示唆を得た。学生の単一の質問に一気に回答を与えるのではなく、複数のブロックに分割した上で段階的にアプローチしていること、問題を明確化する過程と問題を解決する過程で教員と学生の関わり方が異なることなどが明らかとなってきた。

今後はこれらの分析結果を基に、効果的な学習支援対話を生成できるマルチターン型チャットボットを開発し、その運用を通じて有効性を検証していく予定である。さらに個々人の過去の対話履歴や学習履歴も活用したステートフルなチャットボットの開発も目指しており、学生毎に最適な個別指導に対応できるようなチャットボットの開発を目指していきたい。

謝辞 本研究は、授業に関わられた多くの先生方のご支援の下に進めて参りました。武蔵野大学プログラミング基礎非常勤講師の先生方には、授業でのチャットボットの案内や学生のサポートを中心にご担当頂きました。プログラミング基礎サブ講師の先生方には、学生カルテの作成や、オンラインでの学生の質問、またそこでのトランスクリプトデータの収集にご協力いただきました。また、これらトランスクリプトのデータから、教員と学生との対話の構造を分析していただきました、サブ講師の中原香純先生、鈴木志歩先生に感謝いたします。先生方の分析により、対話データの価値を大きく高めることができました。本研究に関わった先生方に改めてお礼申し上げます。

参考文献

- [1] Academy, K., 「Meet Khanmigo, Khan Academy's AI-powered teaching assistant & tutor」
[Online]. Available: <https://www.khanmigo.ai/>
- [2] 「記述式問題対策 AI 学習教材「スルメ」 [Online]. Available: <https://prtimes.jp/main/html/rd/p/000000098.000042400.html>. [Accessed: 17-2 月-2024]
- [3] Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., と Stoica, I., 2023, 「Judging LLM-as-a-Judge with M T-Bench and Chatbot Arena」
- [4] 渡邊紀文, 上地泰彰, 田丸恵理子, 圓崎祐貴, 岡田龍太郎, 糸田孝太, 岡田真穂, 守谷元一, 宮田真宏, 2020, 「コラボレーションツールを利用した学生の個別サポートにおける情報共有環境の構築」, Musashino University Smart Intelligence Center 紀要, 4, pp. 73-82
- [5] 渡邊紀文, 2021, 「問題解決を重視したプログラミング教育とオンラインでの実践」, Musashino University Smart Intelligence Center 紀要, 2, pp. 60-67
- [6] 渡邊紀文, 横山誠, 圓崎祐貴, 岡田龍太郎, 宮田真宏, 2022, 「プログラミング科目でのオンラインでの個別指導と情報共有環境の構築」, Musashino University Smart Intelligence Center 紀要, 3, pp. 78-86

[7] Budiu, R., Liu, F., Cionca, E., & Zhang, A., 2023, 「The 6 Types of Conversations with Generative AI」
[Online]. Available: <https://www.nngroup.com/articles/ai-conversation-types/>

[8] Cecco, L., 「Air Canada ordered to pay customer who was misled by airline's chatbot | Canada [The
Guardian」 [Online].

Available: <https://www.theguardian.com/world/2024/feb/16/air-canada-chatbot-lawsuit>