

研究論文

情報可視化システム開発における 大規模言語モデルプロンプトの定量的調査

平林 晴馬¹ 宮崎 翔¹ 矢谷 浩司¹

受付日 2025年1月15日, 採録日 xxxx年0月xx日

概要：自然言語処理技術に基づく大規模言語モデル (LLM) の進化により、高い生成能力を示す AI 技術がシステム開発やプログラミングの分野にも影響を与えている。本研究では対話型のテキスト生成 AI の利用に関して、情報可視化システムの開発におけるユーザの行動パターンと利用目的に焦点を当てた定量的な分析を行う。情報検索におけるセッション分析の観点を取り入れ、連続して入力されたプロンプト同士の関係性にも注目して AI の利用実態を考察する。さらに、これらの利用実態の調査結果からユーザの生成 AI を用いたシステム開発を支援するツールの設計について考察を行う。分析の結果、ユーザの対話型 AI の利用が AI を主軸としてコードを積極的に記述させる段階と、AI をコード記述の支援に利用してユーザが主体となってコードを記述する 2 つの段階に分類されることが確認された。また、ユーザと AI 間でのシステム開発の進捗状況の同期の難しさが存在することが確認され、支援の必要性が確認された。

キーワード：AI コーディング、定量的分析、ヒューマン・コンピュータ・インタラクション、生成 AI、ソフトウェア・エンジニアリング

A Quantitative Investigation on Use of LLM in Information Visualization System Development

HARUMA HIRABAYASHI¹ KAKERU MIYAZAKI¹ KOJI YATANI¹

Received: January 15, 2025, Accepted: xx xx, xxxx

Abstract: Advances in large language models (LLMs) based on natural language processing have boosted the generative capabilities of AI, influencing software development and programming. This study focuses on how users interact with conversational text-generation AI when they develop information visualization systems. It incorporates a session-based analysis approach from information retrieval, which examines the relationships among consecutive prompts, and explores actual usage patterns. It then considers how to design tools that support system development with generative AI, based on these findings.

The analysis reveals two distinct phases in users' interactions with conversational AI. In the first phase, users rely heavily on AI to generate code, placing AI at the core of coding activities. In the second phase, users take the lead in writing code and use AI as an assistant. The introduction of generative AI leads users to write code while also evaluating and modifying the AI's output. This shift highlights the importance of sharing development progress between users and AI, but also reveals challenges that require further support.

Keywords: AI-coding, quantitative analysis, Human Computer Interaction, generative AI, software engineering

1. はじめに

自然言語処理技術に基づく大規模言語モデル (LLM) の飛躍的な進化は、AI 技術の領域において革新的な発展を

もたらしている [8]。膨大なデータに基づいて学習されたモデルはテキストによる自然な会話だけでなく、画像の生成やプログラミング・コードの生成など様々な用途で利用が進んでおり、幅広い分野に変化や影響を与えている。

特にソフトウェア・エンジニアリングの分野では AI の

¹ 東京大学 Interactive Intelligent Systems Laboratory

コード生成能力を用いたシステム開発が活発に行われるようになり、ChatGPT^{*1}やGitHub Copilot^{*2}のようなサービスを利用したコードの生成やコードのレビュー^{*3}がすでに実用化され多くの開発者によって用いられている。

システム開発における生成 AI の利用が広まるにつれて Human Computer Interaction (HCI) の分野では AI の導入がユーザのプログラミング・コードの記述やシステム開発にどのような影響を与えるかを調査し、より効果的な AI を用いたシステム実装を促進するために、AI コーディングにおける生成 AI の利用実態の調査や、その支援手法の検討が活発に行われている。ただ、先行研究の調査対象としては実際のシステム開発の代替として幾つかのアルゴリズム課題を解くコーディング・タスクが選定されることが多いっている [4]。実際のコーディングによるシステム開発の多くはシステム設計から実際の実装、修正と長期的かつ段階的なものであり、それらを包括的に観察し、分析することでより有効な知見が得られると考えられる。そこで本研究では東京大学で工学部の三年生に向けて開講される「電気電子情報実験・演習第二：情報可視化とデータ解析」を利用し、およそ一ヶ月に渡って行われる 1000 件以上のデータを用いた Web 可視化システムの開発を観察し、より実際のシステム開発に近い環境での AI 利用の実態の調査を行った。また、本研究ではこれまで主にインタビューなど質的な調査に基づいて考察されていたシステム開発における AI の利用に対し、定量的な観点からアプローチし、調査を行った。加えて、AI の利用と Web 検索との類似性に着目し、情報検索 (IR) 分野の研究を参考に実験参加者が目的を達成するまでにどのようなプロンプトを入力し（目的別分類）、AI と何度も取り取りをしたか（セッション別分類）を調査することで新たな定量的観点でシステム開発における AI の利用を考察していく。

本研究の研究的議題としては、

1. 定量的な観点によってシステム開発における対話型 AI の利用の実態はどのように説明されるか
2. 対話型 AI の導入によってシステム開発の実態はどのように変化するか
3. システムの開発における AI 支援はどのように行われるべきか

となっており、これらの観点から利用者の生成 AI の利用を観察し、その実態や影響と支援の手法について考察する。

2. 関連研究

目覚ましい発展を遂げる AI の生成技術は、その汎用性の高さから、様々な用途で取り入れられ、それぞれの分野

に大きな変化をもたらしている。AI の利用が拡大するにつれ、特に Human Computer Interaction (HCI) の分野では、生成 AI の利用の実態や、それがもたらす人間への影響、効果的な利用を目指す支援方針の検討など生成 AI の利用に焦点を当てた研究が活発に行われている。

2.1 生成 AI の利用実態の調査

生成 AI の利用に焦点を当てた研究では、ユーザがどのように生成 AI に指示を与えるか、その入力（プロンプト）に着目したものが多く、White ら [9, 10] はインターネット上で公開された生成 AI の利用を観察し、目的に応じて分類を行なっている。プロンプトを階層的に分類することで、ユーザの目的意識を抽出し、実際にどのように生成 AI に指示を与えていているのか、その実態を調査している。このような利用の実態を調査することで、効果的な生成 AI の利用を考察し、実際にそれを支援するシステムを提案する研究も多く、Wu [11] らは複雑なタスクを一度に生成 AI に指示するのではなく、より単純なタスクに分解することでより高い精度の出力結果を得られることを明らかにし、実際に入力された目的を自然言語処理によって分解し、より単純なタスクを課すプロンプトを繋ぎ合わせることでユーザの目的の分解とその入力を自動化するシステムを作成した。これまで言及してきたテキストベースの生成 AI の利用の他に、画像生成 AI を用いた調査では、生成 AI に入力されるプロンプトとそれに対する出力結果の観察から、より精度の高い出力が得られるプロンプトの特徴を考察し、ユーザの入力を改善するシステムを提案している。このような利用実態の調査は生成 AI がもたらすユーザへの影響と、それに基づく支援の方向性を決定するための基盤調査として重要であるが、生成 AI の利用においては、その目的や扱う AI 技術によってその実態が大きく異なるため、それぞれの利用に応じた調査と考察が必要となっている。

2.2 情報検索分野の類似性

生成 AI による出力に基づく情報の取得はインターネット検索と類似する点も多く、実際に Xie ら [12] は、情報検索 (IR) の分野で行われた研究を参考に画像生成におけるプロンプトの分類と分析を行なっている。IR 分野では目的的情報を得るまでに入力されたクエリ群をセッションとして定義し、ユーザの情報検索における利用の実態を調査する研究が行われており [2]、生成 AI のプロンプト分析においても同様の概念を導入した調査が有効であると考えられる。

2.3 生成 AI によるコード生成とシステム開発

実際、生成 AI がどのようにコーディング時に利用されるかをプロンプト・エンジニアリングの視点で調査した研究

^{*1} <https://openai.com/chatgpt>

^{*2} <https://github.com/features/copilot>

^{*3} <https://github.com/marketplace/actions/ai-code-review-action>

として, Kazemitabaar [4] らは, 初学者による生成 AI を用いた Python のコーディング課題の実施を観察し, プロンプト入力時の状況や内容の分類と内訳を示した。また, 生成 AI の利用戦略を 4 つに分類し, 戦略と課題実施後の学習効果の関係性から生成 AI を頻繁に利用したり, 全く利用しなかった場合に比べて, 生成 AI によるコード生成とユーザ自身によるコード記述を統合的に行った場合の方が Python の記法に関する学習効果が高かったことが報告されており, 生成 AI とユーザが共同でプログラミングを行うことを促進する支援デザインの可能性を示唆している。

また, Jiang ら [3] は HTML と JavaScript を用いた Web アプリケーションの作成においてコードの中に入力されたプロンプトからコードを生成する独自の LLM システム GenLine を作成し, その利用の観察とデザイン改善の考察を行っている。利用の観察ではコーディング経験の有無によって入力手法に異なる傾向があることを報告している。またシステムの出力の精度を出力の採用率によって数値化し, 入力手法と精度の関係性を分析することで, システムの出力精度を向上させるプロンプトへの変換機能の可能性を示した。

システム開発における生成 AI の利用実態の調査としてはインタビューによる質的調査も行われており, Barke ら [1] は, GitHub Copilot を用いたコーディング課題を実施し, 実施後のインタビューから利用実態の質的分析を行っている。その中で, 生成 AI を利用したコーディングでは生成 AI を用いながらも利用者が積極的にコードを書いていく accelerate mode と, 現状の解決策が明らかになっていない状態で生成 AI にコードを記述させていく exploration mode の 2 種類の利用形態が存在することを報告しており, 実際に利用者がタスク中にどのようにこの 2 形態の利用を往来するか定量的観点で示している。また, コード支援に関する考察では, これらの利用形態を入力内容から判別することでユーザが求める支援を提供することが重要であると考察している。

このようなコーディングにおける AI 技術の利用に関してはこれまで単純なコーディング課題の実施やインタビューによる定性的な評価が主となっており, 実際のシステム開発のプロトタイピングを長期的に観察・調査した研究やデータに基づく定量的な分析は多く行われていない。そこで本研究では単純なコード記述に留まらず, システムの設計や構想段階も含めた包括的な観察を行い, 取得されたプロンプトデータに対する定量的なアプローチからシステム開発における生成 AI の利用の実態とその支援の方向性を考察する。

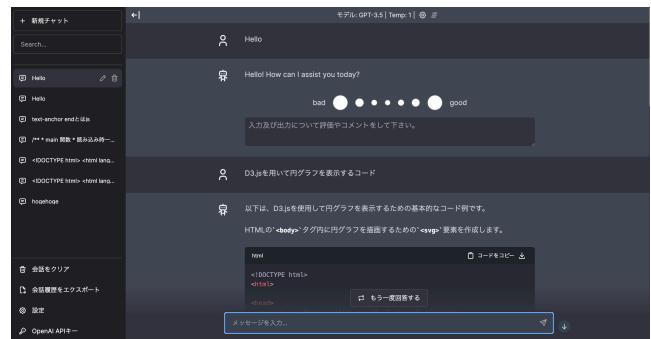


図 1: ChatBot システムの UI. ChatGPT の UI を模した仕様となっている。左下の「会話履歴をエクスポート」のボタンを押すだけで全てのログデータが容易に取得できるようにした機能などを追加している。

Fig. 1 The user interface of the ChatBot system, designed to resemble ChatGPT's UI. Additional features have been implemented, such as a button in the lower-left corner labeled "Export Conversation History," which allows users to easily obtain all log data with a single click.

3. データ収集

3.1 AI 利用実験の概要

本研究では, 対話型 AI を用いたシステム開発におけるプロンプトデータを収集するため, 東京大学の学生向けに行われる講義「電気電子情報実験・演習第二: 情報可視化とデータ解析」(以下, 情報可視化実験)においてデータ収集を実施した。情報可視化実験では, HTML や JavaScript などを用いてブラウザ上で動作するデータ可視化システムの実装を行う。参加生徒は 2 人ないし 3 人のチームで 10 日に渡る講義の時間とその他の時間を利用しておよそ一ヶ月間で開発を行う。データの可視化には統一して D3.js^{*4} と呼ばれる JavaScript の情報可視化に特化したライブラリが用いられる。

また, 本研究のデータ収集のために OpenAI が発行する GPT モデルの API キーを利用した図 1 のような ChatBot を利用してもらった。この ChatBot は Web 上で公開されている Chatbot UI^{*5}をベースに,

- ログデータ(プロンプトや応答, タイムスタンプなど)出力機能の追加
- チャット履歴の削除機能の追加
- ワークスペース機能^{*6}やモデルの temperature 機能^{*7}などの不要な機能の削ぎ落とし

を行った。追加した機能に関しては, 実験参加者ごとにデータの提供に関して提供が拒まれるものは削除できるよ

^{*4} <https://d3js.org/>

^{*5} <https://www.chatbotui.com/ja>

^{*6} Chatbot UI ではチャットルームとは別に一階層上の概念としてワークスペースが存在し, ワークスペースごとにチャット設定を行うことなどが可能となる機能

^{*7} ChatGPT の応答の特性を変化させるためのパラメータ

表 1: 実験参加者の Web 関連のシステム実装経験に関する構成。

Table 1 Composition of the participants' experience in web-related system implementation.

HTML や JavaScript に対する事前知識	人数
Web 関連は全く触ったことがない	13
HTML くらいは触ったことがあるが、JavaScript はない	15
多少触ったことがあり、簡単な Web サービスなら実装できる	8

うにしつつ、容易にデータ提供作業を行えるようにデザインを行い、追加した。また、機能の削ぎ落としに関しては本研究の関心の対象であるプロンプト以外のチャットボットの機能の利用の仕方に実験参加者ごとに差異が生まれることを避け、データの分析を簡素化する目的で実施した。

3.2 研究参加への同意確認

講義に参加する生徒に対して研究参加の同意確認を行い、44人から本研究への参加の同意が得られた。

3.3 情報可視化実験内の生成 AI の利用とデータの収集

講義内では、実験に関わる全てのタスクにおいて生成 AI の利用を可能とし、積極的な利用を促した。講義最終回終了後、講義中に利用した ChatBot システム内の全てのログデータを収集した。本研究への参加を同意した参加者の中には実験期間中にこちらが提供した ChatBot システムを利用しなかった人や、個人で別のシステムを利用した人が数名おり、全員からデータを収集するには至らず、最終的に36人(表1)から5912件のプロンプトと生成AIの応答のデータを収集できた。

3.4 講義最終回後のアンケート調査の実施

講義の最終回後には、情報可視化システムの実装において利用した ChatBot の利用目的や、詳細な利用の実態について問うアンケートを行った。本研究ではデータに基づいた対話型AIの利用を分析していくが、得られた結果に関して、アンケートへの回答と照らし合わせ、考察を行っていく。

4. データの選別・分類

本節では取得したプロンプト・データの分析対象の選別と分類の実施について記述する。

4.1 分析対象データの選別

データの収集段階では ChatBot システム内の入力(プロンプト)データや生成AIの出力など全てのデータを収集した。収集したデータの中には実験前半で行われるチュート



図 2: データの分類のために作成した分類用のシステム画面。左側に前後のデータを表示し、関連性を確かめながら分類を行った。右側にはセッションと目的別それぞれの分類項目が表示されている。

Fig. 2 The system interface developed for data classification.

The left side displays previous and subsequent data for verifying relevance during classification. The right side shows classification items for both sessions and objectives.

リアル課題に取り組むために利用されたものや実験と関連性が低いと見られるデータが含まれていた。本研究ではプログラミングによるシステム開発における生成AIの利用方法を調査するため、実験内で行われるチュートリアルなどに関わるプロンプトデータなどは分析対象外とし、最終的に制作する情報可視化システムに関するやりとりのみを分析対象データとした。その結果、4400件のデータが分析対象として選定された。

4.2 データの分類

データの分類においてはプロンプト入力の文脈を理解し、正確に分類を行うため、前後のプロンプト同士の関連性を加味しながら分類を行っていく必要がある。そこで、図2に示すような分類用のアプリケーションを作成し、画面内で前後のデータを確認しながら、その関連性を加味して分類を行った。分類観点としては、以下に示す二つの観点で分類を行った。

- セッション単位での初期プロンプトと後続プロンプトの分類
- 入力目的別の分類

それぞれの分類に関する説明と分類の結果をまとめる。

4.2.1 入力プロンプトのセッションごとの分類

情報検索の分野におけるセッションの概念をプロンプト分析にも導入し、分類を行った。情報検索の分野では目的の情報を取得するまでに入力されるクエリ群をセッションと定義しており、これをプロンプト分析にも導入するため、Jansenら[2]が行ったようにプロンプトをセッションの開始に入力される「初期プロンプト」とその後に続く「後続プロンプト」に分類を行った。二つの分類の定義に関しては4.2.2内で後述する分類項目の検討段階で適切な定義を検討し、それぞれ以下のように設定した。

- 初期プロンプト … 以前のやりとりを参照せず、独立した目的を持った入力

表 2: セッションを構成する初期プロンプトと後続プロンプトの分類結果。出現回数の総計と初期プロンプトの出現回数から平均的なセッション長はおよそ 5.1 回であると算出できる。

Table 2 The classification results of initial prompts and subsequent prompts that form a session. Based on the total number of occurrences and the occurrence count of initial prompts, the average session length is calculated to be approximately 5.1.

分類	出現回数
初期プロンプト	856
後続プロンプト	3544

- 後続プロンプト … 以前のやりとりを参照しており、同一または関連した目的を持った入力

以上の分類をもとに「初期プロンプト」とそれに続く「後続プロンプト」を一つのセッションとして定義した。分類の結果、表 2 に示すように、プロンプトが分類され、セッション内では平均しておよそ 5.1 回のプロンプト入力が行われたことがわかる。

4.2.2 入力プロンプトの目的別の分類

プロンプトデータの目的ごとの分類では、より詳細にユーザの目的意識を抽出し、適切なカテゴリを作成するためにデータに基づいた概念抽出を目的とした木下による修正版グラウンデッド・セオリー・アプローチ [13] と呼ばれる質的分析の手法を参考に分類項目の作成を行った。実際の分類項目の作成手順を以下に示す。

1. いくつかのデータを基に分類項目を作成する
2. 作成された分類ごとにワークシートを作成し、その定義や具体例を記入する
3. その他のデータに対しても、作成された分類項目に基づいて分類を行う
4. 分類中に例外が発生したら分類項目の再検討を行う
5. 分類項目が定着し、分類が終了するまで 2 から 4 を繰り返す

以上の手法を用いて理論的飽和の状態^{*8}となるまで分類作業と分類項目の検討を繰り返した。分類段階では、ユーザの複雑な目的意識を抽出するために、一つのプロンプトに対し、複数の分類(マルチラベリング)を許容した。また、本研究で得られる知見を可視化システムの実装におけるものに留めず、より一般化されたシステム開発に適用されるものとするために目的ラベルは本研究の実施状況に依存しない一般性の高いものとなるように作成した。

分類の作業を進める中で複数の分類ラベルの該当箇所の重複や分類項目の定義との不一致があった場合には分類項目の再構成や再定義を行い、最終的に表 3 に示す 11 の分類項目が作成された。^{*9}

^{*8} 新たな分類や概念が生成されなくなり、分析上の問題点や例外処理がなくなった状態

^{*9} 以降、11 のラベル名を表 3 内、上から順に「アイディア創出」、

表 3: 実験参加者によって入力されたプロンプトの目的別分類。複数ラベリングを許容し、前後のプロンプトの関係性も加味して分類を行った。分類ラベル名は日本語表記と括弧で示された英語表記を定義した。

Table 3 Objective-based classification of prompts entered by the experiment participants. Multiple labels were allowed, and the relationships between preceding and subsequent prompts were taken into account during the classification process. Each classification label is shown in Japanese, followed by its English translation in parentheses.

分類	定義	回数
アイディア創出 (idea)	複数の選択肢の列挙の要求	66
システム設計の構想		
・実装方針の検討 (plan)	システム作成の方針の検討	212
注目対象の共有		
・現状の同期 (condition)	コードの添付や視覚的情報など現状の伝達	1755
コード生成要求 (generation)	コードの仕様伝達と生成の要求	1849
出力の改善要求 (improvement)	要件理解・コードなどの出力の改善要求	999
エラー・問題解決 (error)	エラー文や問題状況の共有と解決要求	551
知識の獲得・質問 (question)	知識向上のための質問	151
作業の代行 (labor)	機械的作業の代行	149
出力・コードの説明要求 (explain)	取得した情報の説明要求	319
理解・進捗の確認 (check)	ユーザの理解や現状の進捗の確認	71
その他 (others)	誤送信や謝辞などの例外的入力	89

これらの分類項目の作成においては客觀性を担保するため、研究実施者二名での議論を行い、項目決定後の分類は筆頭著者が一人で行った。

5. データの分析

5.1 セッション長に基づく二つの利用形態

セッションごとの調査においては、利用の傾向を観察するため、十分な利用がみられたデータのみを抽出し、分析を行うために、セッション数が 100 件以上であった 18 名の利用者のデータを用いて個人ごとに分析を行った。まず、生成 AI を用いたシステムの作成の過程においてセッションの長さがどのように変化するか調査を行った。する

「実装方針の検討」、「現状の同期」、「コード生成」、「質問」、「出力の改善」、「エラー処理」、「機械作業」、「出力の説明」、「現状の確認」、「その他」と表記する。

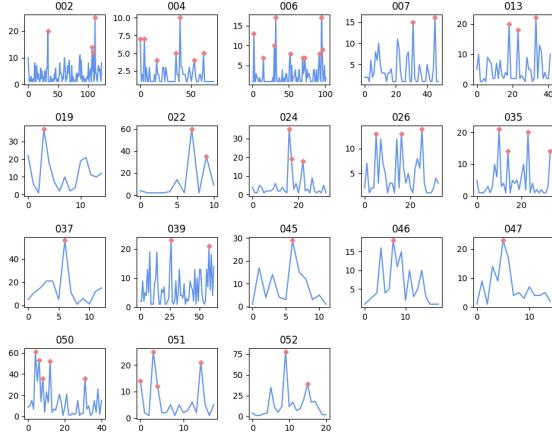


図 3: データ数が 100 件以上であった利用者の時系列順のセッション長。横軸はセッションのインデックスを表す。赤の菱形はセッションの長さが外れ値として検出されたセッション。

Fig. 3 The time-series session lengths of users with over 100 data points. The horizontal axis represents the session index. Red diamonds indicate sessions whose lengths were detected as outliers.

と、図 3 に赤の菱形で示したように長さが一時的に非常に大きくなるセッションが存在する傾向にあることが確認できた。このセッション長が増大する期間の生成 AI の利用について、その他のセッションとの利用方法の差異を調査し、その原因や特性を探る。増大したセッション長の抽出には、以下に示す各データの第三四分位数 (Q_3)、四分位範囲 (IQR) による外れ値判定を利用した。

$$\text{outlier} > Q_3 + 1.5 \times IQR \quad (1)$$

以降、異常値判定により抽出されたセッショングループをグループ I、その他のセッショングループをグループ II として表記する。セッショングループ I, II についてその利用目的にどのような違いがあるかを分析するために、Dirichlet 回帰を行なった。本データ分析においては初め、ベータ回帰の利用を検討したが、同一のプロンプトに対して適用される複数のラベルの連関や、連続したプロンプトでの目的ラベルの相関などラベル同士の今日分散構造を適切に取り込み、分析を行う手法として Dirichlet 回帰を採用した。実際に、モデル比較のために 5-fold クロスバリデーションで平均負の対数尤度 (NLL) を算出したところ、表 4 に示すように Dirichlet 回帰モデルの方が大幅に低い NLL を示し(尤度が高く)データの適合度が高いことを確認した。

Dirichlet 回帰を用いて各ラベルの回帰を用いて、各ラベルのセッション (グループ I vs. グループ II) がその相対的出現比率に及ぼす影響を定量的に評価した。モデルは対数リンク関数を用い、係数 $\beta > 0$ はセッション・グループ I において当該ラベルの比率が相対的に増加することを示す。検定の多重性に対応するために、各ラベルについて推定された β と Benjamini-Hochberg 法による FDR 補正後の p 値をもとに、5% 水準で有意な増減を判定した。表

表 4: Dirichlet 回帰と個別ベータ回帰のモデル適合度比較 (平均負の対数尤度)。Dirichlet 回帰はより小さい (より負の大きい) 平均負の対数尤度を示し、データへの適合度が高いことがわかる。

Table 4 Comparison of model fit between Dirichlet regression and individual beta regressions in terms of mean negative log-likelihood (NLL). The Dirichlet regression yields a lower (more negative) NLL, indicating a better fit to the data.

モデル	平均負の対数尤度 (NLL)
Dirichlet 回帰	-32.15
個別ベータ回帰	14.06

表 5: Dirichlet 回帰によるラベル別係数と有意性。係数が大きいほどセッション・グループ I で相対的出現比率が高いことを示し、コード生成や AI への確認などのラベルで増加が見られる。有意性は*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$ を表す。

Table 5 Regression coefficients and significance for each label from Dirichlet regression. A larger coefficient indicates a relatively higher proportion in session group I; labels related to code generation and AI verification show increases. Significance levels are indicated by *: $p < 0.05$, **: $p < 0.01$, and ***: $p < 0.001$.

ラベル名	係数 (β)	p 値 (FDR 補正後)	有意性
アイディア創出	+0.1599	0.242	—
実装方針の検討	+0.1923	0.218	—
現状の同期	+1.4098	7.76×10^{-31}	***
コード生成	+1.5536	9.30×10^{-38}	***
質問	+0.1735	0.228	—
出力の改善	+1.2051	3.73×10^{-20}	***
エラー処理	+0.8106	2.87×10^{-9}	***
機械作業	+0.1725	0.228	—
出力の説明	+0.4076	4.13×10^{-3}	**
現状の確認	+0.4434	2.04×10^{-3}	**
その他	+0.4726	1.11×10^{-3}	**

5 に示すように長いセッションでは「現状の同期」「コード生成」「出力の改善」「エラー処理」といった、コード記述に直結するラベルの相対的出現比率が有意に上昇しており、また「出力の説明」「現状の確認」など、AI からコードの内容や開発状況を教えてもらうためのラベルも同様に有意な増加を示した。

表 6 に示す実測平均割合を見ると、「コード生成」など表 5 の Dirichlet 回帰で得られた有意性を確認できるほど割合が増加しておらず、「エラー処理」などに関しては割合で見るとグループ I よりもセッションが短いグループ II の方が割合が大きくなっているが、これは表 5 で確認した各

表 6: セッション・グループ I および II における各ラベルの平均割合。Dirichlet 回帰で統計的に有意だったラベルでも、実際の平均割合の増減が必ずしも一致しないことが分かる。

Table 6 Mean proportions of each label in session groups I and II. Even labels that were statistically significant in the Dirichlet regression do not necessarily exhibit corresponding changes in the observed mean proportions.

ラベル名	グループ I	グループ II
アイディア創出	0.0022	0.0112
実装方針の検討	0.0201	0.0629
現状の同期	0.2599	0.2494
コード生成	0.3289	0.3093
質問	0.0132	0.0598
出力の改善	0.1687	0.0987
エラー処理	0.0810	0.1038
機械作業	0.0207	0.0172
出力の説明	0.0512	0.0694
現状の確認	0.0298	0.0072
その他	0.0243	0.0111

ラベルの相対的出現比率が同時に増加したことによるものであり、セッションが長引くにつれて実際の平均的なシェア変化は小さくとも、対話全体の構成比において「コード記述関連の行為」や「AIへの確認を行う行為」が優先される傾向にあると言える。

このように、一つの目的を達するために AI が盛んに利用されたグループ I では、主にコード記述に関連した目的のプロンプトが入力され、セッション長が増大するグループ I では生成 AI によるコーディングが活発に行われていると捉えることができる。AI のコード生成が活発で、セッションの長さが長くなっているという特徴は Barke ら [1] が報告している AI コーディングにおいて AI が主導権を持ってコーディングを出力していく exploration mode に共通しており、セッションの観点から分類された AI コーディングにおける利用形態を barke らの報告した二形態と表 7 のように対応づけることができると考えられる。

5.2 目的別ラベルに基づく AI コーディングのモデル化

AI コーディングにおける AI の利用目的を 11 の項目に分類し、マルチラベリングを行ったが、これらの利用目的同士がどのように関連を持っているかを調査するために、同一セッション内で前後の関係にあるラベル同士のデータを取得した。このデータを元に連続して利用される目的からコーディング時の AI 利用の概観を調査するためにネットワーク図を作成し、各ラベルがランダムに接続される際の値よりも強い接続を示すもの（各ラベル・ノードからの出次数をその他の 10 ラベルへの接続数で平均した値より大きいもの）のみをプロットした（図 4）。なお、該当する

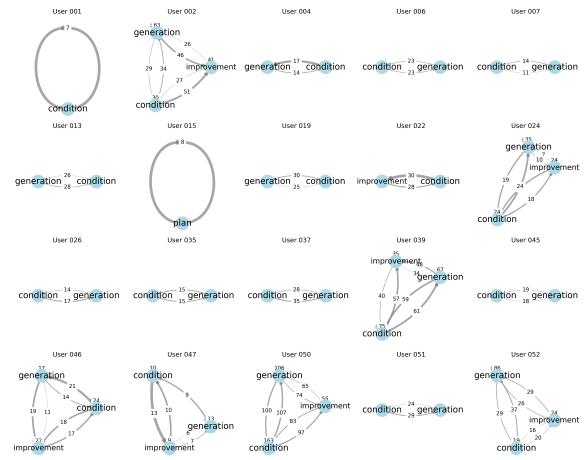


図 4: 目的別ラベルごとの前後関係の中で強い関連性を持ったノード同士の接続を表したネットワーク図。11 のラベルそれぞれの出次数から他の 10 のノードにランダムに接続した際の平均値より大きな値を示したエッジのみを表示している。該当するノードの組みが一つもなかった実験参加者のデータに関しては表示していない。可読性の観点で英語ラベルを利用している。日本語ラベルとの対応は表 3 を参照。

Fig. 4 A network diagram showing connections between nodes with strong relationships in the sequential context of objectie-specific labels. Only edges with values greater than the average calculated from randomly connecting each of the 11 labels to the other 10 nodes are displayed. Data from participants without any such node pairs are excluded. For readability, we use the English labels; the correspondence with the Japanese labels is provided in Table 3.

ラベルが一つも存在しなかった 7 名（表 8）のデータに関してはプロットしていない。プロット結果を表にまとめると表 8 のようになり、多くのユーザがコードやシステムの現状を AI に伝達する「現状の同期」を行いながら、「出力の改善」や「コード生成」を行っていることがわかる。

6. 考察

6.1 AI コーディングにおける利用形態

セッションの観点を導入し、実験参加者の AI 利用を分析したことで、十分な利用があった全ての実験参加者と共にセッション長が局所的に大きくなる段階が存在することがわかった。5.1 で分析したように、セッション長が長くなる利用段階にはコード生成に関連した目的ラベルの利用が有意に大きくなっている、AI に積極的にコード生成を行わせていることがわかった。この結果は表 7 で示したように、先行研究の中で報告された二つの利用形態とも合致しており、セッションを導入した定量的な観点から同様の結果を得られたものと考えられる。この結果から AI コーディングにおいてはユーザが主体的にコードを書きながらサポート・ツールとして AI を利用していく段階（加速的利用）と AI が主導権を持ってコードを生成していく段階（探索的利用）が存在すると考えられ、システム開発の中で

表 7: 本研究におけるセッション観点による利用形態の分類と Barke ら [1] による GitHub Copilot を利用したコーディングにおける利用形態の分類の対応と各利用形態の特徴。

Table 7 The correspondence between the classification of usage patterns based on session perspectives in this study and the classification of coding usage patterns with GitHub Copilot by Barke et al. [1], along with the characteristics of each usage pattern.

本研究における分類	barke らが行った分類	特徴
グループ I	exploration mode	AI が積極的にコード生成を行い、利用時間が長い傾向
グループ II	accelerate mode	AI を支援ツールとして利用しつつ人間主導で作業を進める

表 8: 連続して利用された目的別ラベルにおいて特に利用が多かったラベルの組み合わせ。図 4 に示すように、11 の目的のプロンプトをランダムに利用した時と比べて利用量が多かったものだけを抽出した。

Table 8 Combinations of objective-specific labels that were frequently used consecutively. As shown in Figure 4, only those combinations with higher usage compared to random usage of the 11 purpose-specific prompts were extracted.

ラベルの組み	該当ユーザ数
なし	7 名
「システム設計の構想」	1 名
「現状の同期」	1 名
「現状の同期」・「出力の改善」	1 名
「現状の同期」・「コード生成」	10 名
「現状の同期」・「コード生成」・「出力の改善」	7 名

ユーザと AI がそれぞれ主導権を持ってコーディングしていくことが分かった。このようにコード記述者がユーザと AI の複数体制になる AI コーディングにおいてはその主導権が状況によって変化しながらそれぞれの実装を伝え合うことが必要になると考えられ、実際「注目対象の共有・現状の同期」のように実験参加者が現状を AI に伝える目的ラベルや「出力・コードの説明」や「理解・進捗の確認」のように AI にコードや現状の説明を求める目的ラベルが観察された。こちらもコード記述に直結するラベルとともに、5 で確認したように Dirichlet 回帰を用いた分析によりセッションが長くなるグループ I において有意な増加が認められた。特に後者に関しては AI が主体的にコードを生成していく中で利用者が開発の現状を把握する必要が生じることにより、利用の相対的な出現比率が増加したものと考えられる。

以上の考察をもとに、AI コーディングにおける利用の実態を概略的に説明したモデル図を作成した(図 5)。本研究は情報可視化システムの開発を観察し、得られたデータ

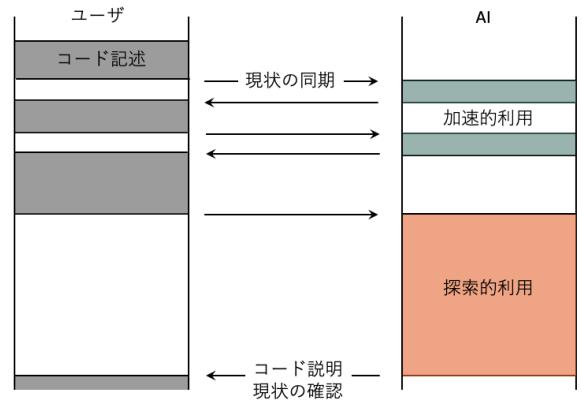


図 5: 本研究における分析をもとに作成された AI コーディングの実態を概略的に説明した図。AI と人間が交互に主導権を持ちながら現状共有を挟むことで一つのシステムを開発していくフローを説明している。

Fig. 5 A diagram illustrating the observed practices in AI-assisted coding based on the analysis in this study. It describes a flow where AI and humans alternately take the lead, incorporating periodic sharing of progress to collaboratively develop a single system.

をもとに分析・考察を行なっているため、本モデル図の妥当性も開発対象が情報可視化システムであるという制約が含まれる。特に「注目対象の共有・現状の同期」ラベルを割り当てられたプロンプトに関しては、ブラウザ上で表示される UI の共有のように可視化システムに限られる利用ケースも確認され、他の開発場面と比較して、視覚情報の共有を行う必要がある点において、現状の同期が難しいことが他のラベルと比べて重要性が大きくなった要因として考えられる。ただ、最も観測された利用ケースは実際のコードの提示のような一般的なシステム開発でも行われるものであり、先行研究で言及された利用段階との関連性や 4.2.2 で言及したような利用場面と依存のない目的別ラベルの作成によって、一定の制約を持つつも、より広くシステム開発における AI 利用のモデル図として一般化可能なものと考える。

6.2 AI 利用によるシステム開発の変化

5.2において目的別のラベル同士の関連性を分析し、現状の共有を行いながら AI によるコードの生成とその改善を繰り返す実験参加者が多く観察された。また、実験終了後のアンケートの中でも

ChatGPT の利用において自身が変化したと思うことがあれば教えてください

といった質問に対して

ChatGPT がエラーのない、期待する挙動を示す完璧なコードを書くまで対話を続けるのではなく、あくまで ChatGPT の書いたコードをベースにして自身でコードをリファクタしたり、あるいは必要な部分だけを抜き出して自身のコードに入れるようになった。
(P23)

といった回答や

コーディングがコーディングというよりもコードレビューに近くなったと思う。自分でどんどん書いていくというよりは、出力が正しいか、無駄がないかというのを気にする時間が多かったと思う。
(P2)

のように AI の出力結果を評価し、さらに改善を行わせたり、自分で書き換えていくといったコード記述よりもコードレビューに近い姿勢でシステム開発を行っている参加者が確認できた。このように、コードを「書く」ことよりも「確認、評価」するという認知的プロセスが重要になってきていることは関連研究の中でも言及されている [6].

6.3 現状の AI コーディングにおける課題と支援の可能性

6.3.1 探索的な利用

先行研究とともに本研究の定量的な観察からも確認された、探索的利用はユーザに目的達成の手法が明確でない際に AI に主体的に作業を行わせている段階と報告されており [1]、加速的利用に対して目的達成まで多くの時間を要することがわかっている。この特徴は本研究の中でもセッション長が長くなることで確認されており、より短い時間で目的の実装を行う上では探索的な利用を効果的に支援することが重要となると考えられる。先行研究においては Wu ら [11] は、LLM を用いてユーザのタスクを適切に分解し、より単純化されたタスクを複数回に分けて AI に実行させることで効果的に期待した出力を得るシステムを考案しており、このようにユーザ側の知識の不足を補填し、効果的な AI を利用を促進する手法の更なる検討が必要であると考えられる。

6.3.2 AI からユーザへの現状の共有

利用の実態と AI 導入によって生じた変化を考察する中で、ユーザと AI の間で主導権が変化していく AI コーディ

ングにおける現状の共有の重要性や AI が出力したコードをレビューするプロセスの存在を確認できた。特に生成 AI のコードを評価するにあたっては、生成 AI にコードを説明させる「出力・コードの説明」やユーザ自身の理解を深める「理解・進捗の確認」のような AI からユーザに現状を共有することが重要であると考えられる一方で、表 3 からもわかるようにこれらのラベルの利用頻度は他のラベルと比べて高くななく、考察された重要性とは裏腹な結果となっている。

この点においては、実験参加者へのアンケートの中で ChatGPT のコードで正しく動いた場合はあまりコードの意味を確認しないままになってしまふこともあった。(P33)

全体像を自分で把握していない。(P29)

といった回答が見られ、実際に現状の把握が十分行えていないと推察できる。実際、Kazemitabaar [5] らは AI を用いたコーディングの容易さが実装者のプログラミング・スキルの学習の妨げになっている問題を取り上げており、AI の素早いコード生成がそのコードの意味を理解しないままに作業を進めてしまうことに繋がると考えられる。

このように、システムの開発状況を理解せずに作業を進めてしまうと、エラーが生じた際にその原因の対応がユーザ自身で行えないなどの問題につながると考えられ、その対応も AI を用いて対応することになる。このようなケースでは探索的な利用に繋がり、結果的に実装効率が低下したり、ユーザ側では対処できないコードが増えてしまうなどの問題が考えられる。

このように AI を用いることで素早いタスク遂行が可能になる一方で、長期的に見ると AI を利用することで総合的なパフォーマンスが低くなってしまう可能性がある。実際、AI との共同作業において AI なしで作業した場合と比べてパフォーマンスが悪化する事例なども報告されており、有効な活用には適切な利用とそれに対する支援が必要になる [7]。本課題に関しても所望のシステムの挙動が得られた際にもユーザに AI の出力したコードに対する理解を促すことで素早いタスク遂行とシステムの現状の適切な把握の両立を目指すことが重要であると考えられる。

7. まとめ

本研究では、情報可視化システムの制作を取り上げ、システムのプロトタイピングの初期段階から実装までの生成 AI の利用を観察し、プロンプト・データの分析を行った。目的別分類では 11 種のラベルによってマルチラベリングを行い、4400 件のデータセットが得られた。こちらは今後の更なる研究や AI を用いたコーディング開発の支援システムに有効に利用されると考えられる。また、情報検索の研究分野で用いられるセッションの観点を取り入れ、新

たな視点による分析と AI の利用に異なる段階が存在することを確認した。また、定量的な分析結果とインタビュー結果を照らし合わせ、システム開発における AI とユーザの状況の同期が不十分であることが確認され、支援の必要性が考察された。

8. 制約

本研究は東京大学の学生を対象に AI を用いた情報可視化システムの開発を実施してもらい、AI の利用データをもとに分析・考察を行なったものであり、本研究の知見をより広いユーザ層や利用場面へ一般化するためには更なる調査が必要になる。また、本研究においては、セッションの定義として、4.2.1 で示したようにプロンプトの「目的」の関連性をもとに定義を行ったが、対話型 AI の利用においては、対話の中で更なる目的が派生することも多く、目的の連續性や同一性をどのように判断するか、セッションをどのように定義するかに関しては今後、更なる議論と分析が必要になると考える。

謝辞 本稿の推敲に協力してくださった研究室のメンバーの皆さん、実験に参加してくださった方々に深くお礼申し上げます。

参考文献

- [1] Barke, S., James, M. B. and Polikarpova, N.: Grounded copilot: How programmers interact with code-generating models, *Proceedings of the ACM on Programming Languages*, Vol. 7, No. OOPSLA1, pp. 85–111 (2023).
- [2] Jansen, B. J., Spink, A., Bateman, J. and Saracevic, T.: Real life information retrieval: A study of user queries on the web, *Acm sigir forum*, Vol. 32, No. 1, ACM New York, NY, USA, pp. 5–17 (1998).
- [3] Jiang, E., Toh, E., Molina, A., Olson, K., Kayacik, C., Donsbach, A., Cai, C. J. and Terry, M.: Discovering the syntax and strategies of natural language programming with generative language models, *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pp. 1–19 (2022).
- [4] Kazemitaabari, M., Hou, X., Henley, A., Ericson, B. J., Weintrop, D. and Grossman, T.: How novices use LLM-based code generators to solve CS1 coding tasks in a self-paced learning environment, *Proceedings of the 23rd Koli calling international conference on computing education research*, pp. 1–12 (2023).
- [5] Kazemitaabari, M., Huang, O., Suh, S., Henley, A. Z. and Grossman, T.: Exploring the design space of cognitive engagement techniques with ai-generated code for enhanced learning, *Proceedings of the 30th International Conference on Intelligent User Interfaces*, pp. 695–714 (2025).
- [6] Sarkar, A., Gordon, A. D., Negreanu, C., Poelitz, C., Ragavan, S. S. and Zorn, B.: What is it like to program with artificial intelligence?, *arXiv preprint arXiv:2208.06213* (2022).
- [7] Steyvers, M. and Kumar, A.: Three challenges for AI-assisted decision-making, *Perspectives on Psychological Science*, Vol. 19, No. 5, pp. 722–734 (2024).
- [8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I.: Attention is all you need, *Advances in neural information processing systems*, Vol. 30 (2017).
- [9] White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J. and Schmidt, D. C.: A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT (2023).
- [10] White, J., Hays, S., Fu, Q., Spencer-Smith, J. and Schmidt, D. C.: ChatGPT Prompt Patterns for Improving Code Quality, Refactoring, Requirements Elicitation, and Software Design (2023).
- [11] Wu, T., Jiang, E., Donsbach, A., Gray, J., Molina, A., Terry, M. and Cai, C. J.: Promptchainer: Chaining large language model prompts through visual programming, *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pp. 1–10 (2022).
- [12] Xie, Y., Pan, Z., Ma, J., Jie, L. and Mei, Q.: A prompt log analysis of text-to-image generation systems, *Proceedings of the ACM Web Conference 2023*, pp. 3892–3902 (2023).
- [13] 木下康仁: 修正版グラウンデッド・セオリー・アプローチ (M-GTA) の分析技法, 博士論文, University of Toyama (2007).