ChatGPTを用いたプログラム生成におけるプログラミング言語間の精度比較

Comparison of accuracy between programming languages in program generation using ChatGPT

榎本 創太 † 島 和之 †
Sohta Enomoto † Kazuyuki Shima †
† 広島市立大学 情報科学研究科

1 まえがき

ChatGPTをはじめとした生成系AIサービスが登場し，幅広い分野で利用され始めている．これらのサービスは個人のPCやスマートフォンで手軽に利用することができるため，創作に必要な知識や技術を持たないユーザであっても，文章や画像を生成することが可能となった．しかし，ChatGPTの回答は大量の学習データから深層学習によって構築した言語モデルから統計的にそれらしい応答が出力されるため，出力される回答には誤りが含まれている可能性が常に存在する．参考文献 [1] は文部科学省が発行した初等中等教育における生成AI利用のガイドラインであり，生成AIの教育利用について述べられている．文書中では，生成AIは新たな情報技術であり，生産性の向上に活躍している生成AIの仕組みを理解し，将来的に使いこなすための力を育てる必要があるとする一方，利用時の危険性，創造性や学習意欲への影響を考慮し，限定的な利用から始めることが適切であるとしている．また，ChatGPTは文章だけでなく，プログラムを生成することも可能なため，プログラミング学習にも活用される．ChatGPTは，学習者にサンプルプログラムの生成，プログラム解説，プログラム修正等を提供する．しかし，ここで出力される回答も全て正しい情報であるとは限らない．学習段階のユーザにとって，回答に含まれる誤りを見つけることは非常に困難である．

ChatGPTによるプログラム生成の不得意分野を把握することで, 学習者にとって出力されたプログラムの信用性を判断する有益な材料となることが期待できる．そこで本研究では，ChatGPTによるコード生成性能向上のための初期調査として,ChatGPTによるプログラミング言語間のプログラム生成精度について調査する．

2 関連研究

参考文献 [2] では，ChatGPTが大学のプログラミング授業 (Python言語) に及ぼす影響を評価している．実験では，大学1年生を対象とした2つのプログラミング授業 (基礎，応用) の授業課題をChatGPTにプロンプトとして与え，出力されたプログラムを編集することなく受講者が提出すると想定し，どの程度正解できるか調査している．また，[2] ではChatGPTを用いて解くのが難しい課題の傾向として，入出力トークン数に着目し考察されている．

実験の結果，基礎授業課題ではChatGPT-3.5を用いて91%(=32/35)，応用授業課題では35%(6/17)の課題で正解と判定された．応用問題についてChatGPT-4を用いて追加実験を行った結果，76%(13/17)の課題で正解と判定された．

実験の結果から [2] では，ChatGPT-4を用いれば，どちらの授業であっても十分に合格することができると結論づけられている．また，ChatGPT-4でのみ正解と判定された課題と両モデルで正解と判定された課題の累計出題長や解答長に明確な差はないとされ，両モデルで非正解と判定された課題はChatGPT-4のトークン制限が影響したのではないかと述べられている．

参考文献 [3] では, プロンプトとして日本語，英語，中国語のデータセットを使用して，GitHub Copilotによるコード生成を行っている．実験の結果, 日本語，英語，中国語の順に正答率が高く，英語と中国語には基礎問題において約11.5%の差が生じたことが確認されている．

3 実験

3.1 実験方法

本研究では，関連研究の結果を踏まえ，Python以外のプログラミング言語でプログラム生成を行い，正答率にどのような違いがあるのかを調査する．

実験を行うにあたって，次のような条件を設定し実験を行った．

- 利用したChatGPTのモデル：ChatGPT-3.5(2023/9/28時点のもの)
- 使用問題：競技プログラミングの鉄則 演習問題集 A問題 66問
- 使用言語 (入力)：日本語
- 使用言語 (出力)：C++, Java
- 採点に利用したサイト：AtCoder

AtCoderは競技プログラミングコンテストを開催しているWebサイトであり，演習問題が公開されている．本実験ではAtCoder内で提供されている競技プログラミングの鉄則演習問題と対応する自動採点システムを利用した．演習問題には問題文，制約条件，入出力形式，入出力例が与えられる．これをテキスト形式で記述しChatGPTにプロンプトとして与え，出力したプログラムを編集することなくAtCoderの自動採点システムに提出し採点を行った．

また，使用した問題集の問題は大きく 9 つのテーマに問題が分類されており，ChatGPT が苦手とする分野の調査も重ねて行った．

3.2 前提条件

実験を行うにあたって次のような前提条件を設定を行った．

- ChatGPT は同じ入力であっても，出力が異なるため一つの問題につき 5 回程度プログラム生成を行い，一度でも正解と判定された問題を正解として扱う．
- ChatGPT は前の入力を引き継いで出力文を生成するため，問題ごとに新規チャットを立ち上げ実行する．
- ChatGPT がテキスト入力しか受け付けないため，問題に図や表が含まれる場合，テキストで表現できるものはテキストに変換し，そうでないものは省略し実行した．
- AtCoder の採点結果は正解，不正解，コンパイルエラー，実行時エラー，実行時間超過の五種類に判別される．実行時間超過と判定された問題について，テストケースのサイズによって判定が変化したため，本実験では，正解と判定された問題のみを正解とし，他の 4 つを非正解として扱う．

3.3 実験結果

実験の結果は次のようになった．C++では 45.5%(30/66)，Java では 48.5%(32/66) の問題で正解と判定された．不正解と判定された問題には，アルゴリズムや計算式が適切でない，出力形式が条件を満たしていない，例外処理がされていないなど様々だった．両言語とも正解と判定された問題は，34.8%(23/66) であった．

3.4 考察

関連研究の結果とは異なり両言語とも正答率 50%を切る結果となった．このような結果となった理由の一つとして，使用問題の性質が影響したのではないかと考えられる．ChatGPT はパターンマッチで出力する内容を決定しているため，資料の少ないものの回答には不正確なものが多く含まれる傾向にある．競技プログラミングの問題は性質上，競技者自身でアルゴリズムを考え解くよう工夫されているため，ChatGPT がプログラム生成する際，パターンマッチで適合したものが少なかったと考えられる．実際，上記の実験で不正解と判定された問題の多くは，考察テクニック，グラフアルゴリズムに分類されるオリジナル性の高い問題だった．

本研究は，ChatGPT を用いたプログラム生成の言語間性能調査を目的とするため,C++と JAVA の両言語で正解と判定された問題に着目し, さらに別の言語で追加の実験を行った．

3.5 追加実験

自動採点システムに対応した言語のうち，検索エンジンのデータを基にしたプログラミング言語ランキング「TIOBE」のランキング 30 位以下 (2023 年 11 月時点) の 6 言語で同様の実験を行った．

- 利用した ChatGPT のモデル： ChatGPT-3.5(2023/10/16 時点のもの)
- 使用問題 : 両言語とも正解と判定された問題 5 問 [表 1]
- 使用言語 : Lua，Dart，Julia，Haskell，Crystal，なでしこ

表 1: 使用問題内容

言語	問題内容
A01	入力された自然数の平方数を出力
A04	入力された自然数を 2 進数へ変換し出力
A11	配列の要素を二分探索
A19	ナップサック問題
A27	入力された 2 つの自然数の最大公約数を出力

3.6 追加実験結果

実験の結果は [表 2] のようになった．Dart のみ全問正解と判定され，Lua，Julia，Haskell はそれぞれ 4 問ずつ正解と判定された．Crystal，なでしこは一問も課題を満たすプログラムを生成しなかった．Crystal についてプログラムのエラー文を ChatGPT に引き続き与えると正解と判定されるプログラムを出力したが，なでしこは正解と判定されるプログラムを生成しなかった．

表 2: 実験結果

言語＼問題	A01	A04	A11	A19	A27
Lua	○	×	○	○	○
Dart	○	○	○	○	○
Julia	○	○	○	×	○
Haskell	○	○	○	×	○
Crystal	×	×	×	×	×
なでしこ	×	×	×	×	×

4 ChatGPT が生成したなでしこプログラム

プログラミング言語「なでしこ」は日本語で記述する日本語プログラミング言語である．以下は ChatGPT に最大公約数を求める問題を与えた際に出力したプログラムの関数部分を抜き出したものとなでしこで記述したプログラムである．

```
FUNC 最大公約数 (整数 A, 整数 B)
    IF B == 0
        RETURN A
    ELSE
        RETURN 最大公約数 (B, A % B)
    ENDIF
END FUNC
```

```
●最大公約数（A の,B の）
    もし,B が 0 ならば
        A で戻る
    違えば,
        最大公約数（B,A%B）で戻る
    ここまで
ここまで
```

アルゴリズム自体は最大公約数を求めるものであるが， なでしこでの記述方式が ChatGPT の学習内容に含まれていないためか， ChatGPT はなでしこの文法ではなく， 最大公約数を求める疑似コードを出力した．出力されたコードとなでしこで記述されたプログラムに明確な差があるため, プログラム初学者であっても間違った出力がされているとわかりやすい．しかしこれは, 珍しいケースであり非正解と判定されたプログラムの多くは使用言語形式に沿っているように見えるコードであり, どこにエラーが含まれているか判別するのは困難であった．

5 まとめと今後の予定

本研究では， ChatGPT によるコード生成性能向上のための初期調査として, ChatGPT によるプログラミング言語間のプログラム生成精度比較についての調査を行った．実験の結果， ChatGPT のプログラム生成には生成する言語によって生成能力に差があり, 不正解と判定された問題の多くは， 考察テクニック， グラフアルゴリズムに分類されるオリジナル性の高い問題だった．有名でない言語を用いてプログラム生成をした際には, 指定言語の代わりに擬似コードが出力された．今後の予定として， ChatGPT に与えるプロンプト形式の変更によるプログラム性能調査が挙げられる．

参考文献

[1] 文部科学省初等中等教育局,”初等中等教育段階における生成 AI の利用に関する暫定的なガイドライン”文部科学省初等中等教育局,2023.

[2] 鈴木智也, 神谷年洋,”ChatGPT によるプログラミング授業の課題の解答生成の評価”信学技報 ,Vol.123,No.123,pp.55-60,2023.

[3] 小柳慶, 野口広太郎, 王棟, 近藤将成, 亀井靖高, 鵜林尚靖,”GitHub Copilot を用いたコード推薦における入力言語の影響調査” *FOSE2023*,2023.