生成系 AI によるコードの自動修正とその解説が与える学習効果

椋　泰聖

1. はじめに

近年，人工知能(AI)技術の発展が著しく，多様な分野で活用されている．吉田ら[1]は自然言語処理に優れた ChatGPT を用いたコード生成に着目し，人手によるバグ修正の作業量を軽減するシステムを構築している．具体的には ChatGPT が生成したコードと同時に生成したテストコードを用いて実行エラーなどを読み取り，そのエラーと生成されたコードの修正を再度 ChatGPT に依頼する．

本研究では，コード作成の学習支援を狙い，吉田らのシステム加え，修正箇所の提示とその解説を行う機能を実装した構成がもつ学習への効果を検証することを目的とする．

2. 実験システムの設計と実装

本システムは生成系 AI（ChatGPT）を用いて，コードの生成，コードの修正，コードのテスト，修正箇所の提示，解説の生成を行う．図 1 はシステムの外観である．上段の入力欄は依頼文を入力する箇所であり，生成コード欄と修正箇所欄は ChatGPT が生成したコードや修正箇所を表示する．入力欄に生成したいコードの内容を入力し，生成・テスト実行ボタンを押す．本システムは以下の手順で動作する．

図 1　システムの外観

① ユーザーはシステムに生成して欲しいコードの内容を入力する．
② システムは入力された内容を ChatGPT API を利用しコードとテストコードの生成を依頼する．
③ ChatGPT は依頼に沿ってコードを生成するとともにテストコードの生成を行う．依頼が修正の場合ここで修正箇所と解説を生成する．
④ システムはテストコードを用いて生成されたコードのコンパイルを行う．
⑤ エラーが発生した場合，システムは再度 ChatGPT にエラー内容とともにコードの修正の依頼と，テストコードの生成を依頼する．エラーが発生しなかった場合は⑦へ進む．
⑥ ③～⑤の手順を繰り返し行う．
⑦ システムは，ユーザーに最終のコードと修正箇所，解説を提示する．
⑧ ユーザは，エラーに対しての不明点を解明しながら学習に役立てる．

3. 生成系 AI へのプロンプト設計

ChatGPT API に依頼するプロンプトの冒頭を以下に記載する．以降に入力欄へ入力された内容が続く．

```
Always generate only valid Java source code and a Junit5 test class.  The main code must always contain a public static void main(String[] args) method. Do not include any explanetions, comments, or markdown formatting. Output only two plain code blocks (main and test) separated by "#####". The main class must be named App, and the test class must be named AppTest.
```

図 2　プロンプト設計

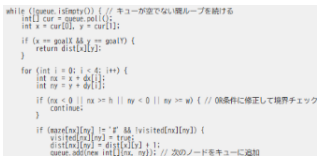

図 3　生成コード

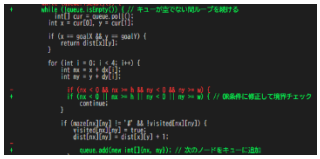

図 4　修正箇所

4. 実験と結果

本システムの動作を確認するため，幅優先探索を処理するコードの生成を行う．図 3 は生成されたコードの一部である．この例ではエラーが出なかったため，手動で関数の削除や if 文の条件の改変を行い，エラーを挿入した．図 4 はシステムが修正した箇所とその解説である．エラーは適切に修正され，修正箇所の提示も行えた．

5. おわりに

本稿はエラーを故意に挿入したコードによる修正とその解説の生成を確認したため，より複雑なコードの生成での評価が必要である．さらに学習効果の検証も今後の予定である．

［参考文献］

[1] 吉田　隼, 佐藤　央, 近藤　羽音, 橋浦　弘明, 櫨山 淳雄：ChatGPT を用いた生成プログラムの自動修正システムの開発, 電子情報通信学会, KBSE2023-68，2024

[担当教員] 石原　真紀夫