

ChatGPTを用いたJavaプログラミングエラーメッセージの学習者向け解説の生成 Generating beginner-friendly explanations of Java programming error messages using ChatGPT.

岡田 祐花¹⁾ 来住 伸子¹⁾ 中野 美由紀¹⁾

Yuka Okada Nobuko Kishi Miyuki Nakano

要旨

本研究では、Javaのプログラミングエラーメッセージ(以降、PEMと表記)の、プログラミング初学者向けの解説をChatGPTを用いて生成した。ChatGPTへのプロンプトを変更して複数の解説を生成した。生成された解説を比較し、プログラミング初学者のためのチャットボットの作成の観点から検討を行った。

1 研究背景

プログラミングにおいて、プログラムの誤りを探して修正するデバッグは重要な技術である。デバッグには、プログラミング言語処理系が生成するプログラミングエラーメッセージ(PEM)の意味を理解し、適切な修正を行なうことが必要である。しかし、プログラミングを学び始めた初学者が、PEMの意味を理解することは難しい。プログラムの構文や機能に関する用語が分からない、あるいは、プログラムのエラーが起きる箇所と、修正が必要な箇所が異なることに対応できないなど、さまざまな原因が考えられる。

そこで、PEMを分かりやすくし、初学者がデバッグすることを学びやすくするための研究が行われてきた。最近では、LLMを利用したPEMの解説生成が行われている。先行研究[1]では、Debugging C Compiler (DCC) [2]が出力するPEMについて、大規模言語生成モデル(以下LLM)を利用して初学者向け解説を生成した。プロンプトを利用し、「コード」を解説に含めないという条件をつけて、各種PEMの解説を生成した。

PEMの解説に「コード」を含めないという条件をつけるのは、初学者に自分でコードを作成するという経験をもらうためである。解説にコードが含まれると、エラーの原因やPEMの意味を考えずにコードをコピーする可能性がある。熟練者がプログラミングの効率化を目指している場合は、LLMが生成するコードを利用することは問題ない。しかし、初学者はプログラムの構文や機能について理解を深めるため、自分でコードを作成するという経験を積むべきである。

本研究では、初学者向けのPEMの解説をLLMで生成する方法について実験し、比較、分類を行った。今後は、生成されたPEMの解説を利用して、プログラムのエラーの修正方法を対話的に学べるチャットボットの作成を目指している。

2 先行研究

Taylorらは、初心者向けのコンパイラエラーメッセージを生成する`dcc --help`を提案し、評価を行った[1]。`dcc --help`は、“Debugging C Compiler(DCC)”[2]とLLMを統合したシステムである。DCCは、プログラミング学習者向けに、`gcc`[3]より丁寧なPEMを生成するC言語処理系であり、`dcc --help`は、DCCの改良版という位置付けである。学習者が、エラーのあるプロ

グラムを`dcc --help`でコンパイルすると、PEMと、LLMが生成するその解説が表示される。コードは含まれない。学習者はPEMと解説を読んで、プログラムの修正を行う。

`dcc --help`の評価は、プログラミング熟練者4名が行った。コンパイルエラーメッセージに対して生成された解説を、正確さなどについてランクづけを行った。

3 提案手法

本研究で作成中の、初学者にPEMを解説するチャットボット(対話システム)は、図1に示す要素で構成される。使用したアプリケーションのフレームワークはStreamlit[4]で、ChatGPTのモデルはgpt-4を使用した。

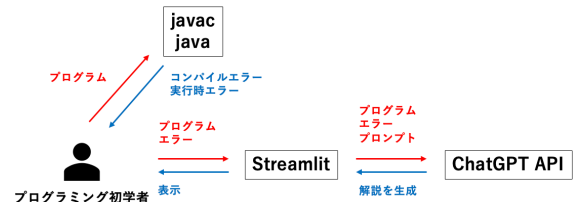


図1 システム概要

初学者は、プログラムをプログラミング言語処理系(例: `javac` または `javac` コマンド)に入力した後、PEM(コンパイルエラーまたは実行時エラー)があった場合、プログラムとPEMをチャットボットに入力する。チャットボットはStreamlitを利用して作成されており、ChatGPT APIを利用してエラーの解説を生成する。

現在は、先行研究の行われた時期と比べると、ChatGPTではコード生成への対応が高度化している。エラーのあるプログラムを入力すると、エラーのない、修正されたプログラムを生成することが多い。そこで、提案手法では、各種のプロンプトを作成し、修正されたプログラムを含まないPEMの解説を生成することを目指すことにした。適切な解説が実際に生成できるか、実験によって調べた。

4 PEMの解説の生成実験

解説の生成には、エラーのあるプログラム、JDKの生成したPEM、プロンプトの3つの要素を使用した。

4.1 エラーのあるプログラム

初学者向けの演習課題でよく見られるエラーを再現した短いプログラムを作成し、使用した。エラーの発生するプログラムの例として次の`eja04.java`を示す。

```
1 public class eja04 {
2     static int sum(int[] table, int n) {
3         int total = 0;
4         for (int i = 0; i < 5; i++) {
5             total = total + table[i];
6         }
7         return total;
8     }
9     public static void main(String[] args) {
```

1) 津田塾大学 Tsuda University

```

10     int[] a = {1, 4, 5, 6};
11     System.out.println(sum(a, 4));
12     int[] b = {1, 4, 5, 6, 3};
13     System.out.println(sum(b, 5));
14 }
15 }

```

4.2 PEM

上記のエラーを含むプログラム 4.1 に対し、JDK の Java 仮想機械 (java コマンド) は次のメッセージを生成する。

```

java eja04
Exception in thread "main"
java.lang.ArrayIndexOutOfBoundsException:
Index 4 out of bounds for length 4
    at eja04.sum(eja04.java:5)
    at eja04.main(eja04.java:11)

```

4.3 プロンプト

システムのプロンプトとして次のものを使用した。

1. "" (なにも設定しない)
2. “プログラムのエラーを説明してください。コードは含めないでください”
3. “プログラムのエラーを丁寧に説明してください。コードは含めないでください。”
4. “プログラムのエラーの解説と、エラーの修正方法を、2段階に分けて説明してください。コードは含めないでください。”
5. “プログラムのエラーを説明してください。”
6. “プログラムのエラーを丁寧に説明してください。”
7. “プログラムのエラーの解説と、エラーの修正方法を、2段階に分けて説明してください。”

プロンプト 1 は、ユーザーのプロンプトとしてエラーのあるプログラムと JDK の生成した PEM のみを使用した。プロンプト 2 から 7 は、エラーのあるプログラム、JDK の生成した PEM に “プログラムのエラーを説明してください。”という文を追加して使用した。

4.4 生成結果の集計

上記のプログラムと PEM と、7 種のプロンプトを組み合わせて、ChatGPT に与えると、7 通りの解説が得られた。7 通りの解説の字数や内容をまとめると次のようになる。エラーの出るプログラムは 4.1 を使用した。「解説の言語」の英/日は、生成された解説の使用言語が英語か日本語かを示している。

プロンプト	1	2	3	4	5	6	7
全文字数	724	507	548	421	588	988	843
コード字数	395	0	0	0	143	468	396
コード以外字数	329	507	548	421	445	520	447
コード有無	有	無	無	無	有	有	有
解説の言語	英	日	日	日	日	日	日

4.5 生成結果の例

集計表にある、最も字数の少ない解説はプロンプト 4 で、次のようになった。

エラーの解説: このエラーは “ArrayIndexOutOfBoundsException” という種類で、「Index 4 out of bounds for length 4」というメッセージが表示されています。これは配列の範囲外にアクセスしようとしたときに発生するエラーで、

この場合は 4 番目 (0 から数えて) のインデックスにアクセスしようとしたときに 4 要素しかない配列 (長さ 4 の配列) にアクセスしようとしたため発生しています。

エラーの修正方法: このエラーを修正するためには、配列の範囲外にアクセスしようとするコードを修正する必要があります。「sum」メソッドの for ループ中で配列にアクセスしているため、ここを中心に見直しましょう。具体的には、for ループの終了条件を固定の数字 (この場合は 5) ではなく、引数として渡される配列の長さ (n あるいは table.length) にすることで、配列の範囲外にアクセスすることを防ぐことができます。

4.6 生成結果の考察

ユーザーのプロンプトを、エラーの出るプログラムと PEM のみにすると、英語で解説が生成された (プロンプト 1)。プロンプトに “コードは含めないでください” が使用されると、実際にコードが含まれないことが多いことが確認できた (プロンプト 2, 3, 4)。解説を “丁寧に” と指定すると、長い解説を生成するが、必ずしも全ての学習者にとって分かりやすい表現にならないことが観察できた (プロンプト 3, 6)。“プログラムのエラーの解説と、エラーの修正方法を、2段階に分けて” という指示で生成された解答は、4.5 のように 2 段落で構成され、視覚的に読みやすい解説となった (プロンプト 4, 7)。

プログラミング初学者は、プロンプト 5 のような、エラーの出ないコードが含まれていて、説明文が少ない解説を求めていると予想する。しかし、プログラミング教育者の視点で考えると、学習者の理解度によって提示する解説を変えることが適切だと考える。

例えば、理解度の高い学生には、コードが含まれない、短い解説を提示する。短い説明ではエラーを理解することが難しい学生には丁寧な解説を提示し、それでもコードの修正方法がわからない場合にコードが含まれている解説を提示する。このように、提示する解説を段階的に変えて使用することは、PEM の理解度向上を促すと考える。

本実験では、プロンプトを変更すると文字数やコードの有無などに違いのある解説を生成できることがわかった。この結果は、学習者の理解度に対応する質問応答システムの今後の開発に有用と見込まれる。

5 まとめと今後の研究

ChatGPT を用いて、プログラミング初学者向けの PEM の解説を生成した。今後、より多くのエラーについて調査し、学習者の理解度に応じて適切な PEM の解説を応答するチャットアプリの開発を目指す。

参考文献

- [1] Taylor A. et al. Dcc-help: Transforming the role of the compiler by generating context-aware error explanations with large language models. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, pp. 1314–1320, 2024.
- [2] Taylor A. et al. Foundations first: Improving c’s viability in introductory programming courses with the debugging c compiler. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, pp. 346–352, 2023.
- [3] Free Software Foundation. Gcc, the gnu compiler collection. <https://gcc.gnu.org/>.
- [4] Streamlit. *Streamlit Documentation*, 2024. <https://docs.streamlit.io/>.