**研 究 計 画 書**

学籍番号5223C038　　　提出日 12/15/2023

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 氏　名 | 馮 天時 | 専　　　攻 | 経営システム工学 | 指導教員 | 岸 知二 　印 | |
| 研究指導 | ソフトウェア工学研究 |
| 研　究  題　目 | **機械学習を用いたAI生成ソースコードの識別** | | | | | 修士課程 |

1.　研究の背景と目的

ChatGPTは優れたコード生成能力により、ソフトウェア開発、研究、教育などさまざまな分野で広く活用され、人々に多くの便利を提供している。しかしそれに伴い、避けられない問題もいくつ起こしている。

例えば教育分野では、学生は教師が設定した課題の問題を要件としてChatGPTに入力するだけで、問題を満たすソースコードを簡単に得られる。もし学生が思考せず、そのままAIで生成したコードを自分の成果として提出すれば（可能性高く）、教師は学生が課題を通じてそれに関する能力を身につけたかどうかを判断できなくなり、これは教育の本質に反することである。

まだIT企業でのエンジニア職は一般的にプログラミングスキルが求められ、面接ではコーディングインタビューがよく含まれている。オンライン面接のトレンドに伴い、これらのテストはオンラインテストプラットフォームで行われるようになっている。しかし、ほとんどのプラットフォームはオンラインでのデバッグ機能を提供しないが、プログラミングにはデバッグが不可欠であるため、ほぼすべてのプラットフォームは候補者が自分の環境でコードを編集してデバッグし、その後プラットフォームにコピーして正確性を検証することを許可している。筆者の知る限りでは書き込み中の不正行為の検出はなく、これは不正行為者に大きな余地を与える。従来の不正行為者はプログラミング経験が豊富な第三者を探して自分を支援すると考えられるが、ChatGPTのリリースに伴い、そのコストは現在非常に低くなっている。ChatGPTのコード生成能力は経験豊富な開発者に匹敵し、日本でのオンラインコーディングテストを90％以上簡単に通過できる。ChatGPTの不適切な応用は、オンラインでのコーディングテスト全体を事実上無意味にするといっても過言ではない。

実際に、多くの学者がこれらの問題に気づき始めている。学界には、AI生成テキスト識別に関する研究がたくさん存在している。GPTZeroやOpenAI Text ClassifierのようなAIテキスト検出ツールが次々と実現され、大きな進歩を遂げている。しかし、ほぼすべてのAIテキスト検出ツールは、入力がコードの場合、検出結果が混乱し、精度が極めて低くなっている。主な要因としては、モデルは入力がコードのシナリオに対してファインチューニングされていないと考えられる。

AI生成テキスト識別に比べ、AI生成コード識別に関する研究は非常に不足であり、筆者の知る限りでは参考文献［１］のみであり、中の「関連研究」セクションもAI生成コード識別に直接関連する研究を言及していないである。

前述通り、業界にはAI生成コードを識別する有効なツールが存在おらず、学界も関連する研究が非常に不足、しかし、教育やオンラインテスト業界では効果的なAI生成コード検出ツールへの需要は日々増加している。結論として、AI生成コード識別の研究価値は十分ある。

本研究の目的としては、ChatGPTで生成した主流となるプログラミング言語でのソースコードと人間で書いたコードを有効に分類し、高精度かつ汎用性高いモデルの実現を目指すことである。

2.　従来研究

Phuong T. Nguyenらは、AIで生成したコード断片を判別する実証的研究を行い［１］、大規模ｐre-ｔrainedモデルCodeBERT［２］を基に、識別手法「GPTSniffer」を提案した。これは、AI生成コードの識別任務の実現可能性を初歩的に検証し、識別能力に影響を与える要因を調査している。結果として、GPTSnifferはコードを正確にAI生成と人間編集に分類でき、精度はGPTZeroおよびOpenAI Text Classifier、二つのベースラインより優れている。さらに、論文は同じソリューションに対してAI生成、人間編集のペアとなるコード断片サンプルの存在が分類の性能向上に役立つことを示している。

データセットとして、Phuong T. NguyenらはJava教科書の演習問題やその他の情報源から約1000サンプルデータを収集し、データセットを構築した。しかし筆者は、データセットのコード要件が単純で、本研究の応用シナリオには適していないと考えている。

3.　研究内容

3.1　研究アプローチ

本研究は、二つの側面から行う予定である。

一つ目は、AI生成コードと人間編集コードの本質的な違いを明らかにするために、ChatGPTを用いてコード生成を行った経験から得られたコモンセンスや調査に基づいて、AI生成コード判別タスクに関連する特徴の重要性分析を行う。筆者はChatGPTを使用したコード生成の経験に基づいて、以下のコモンセンスや仮説を述べる：

1. AI生成のコードは通常、一貫したコードスタイルに従い、品質が高い。一方、人間で書かれたコードには、変数名、インデント、コメントなど、異なるスタイルや品質の違いがあると考えられる。
2. 人間で書いたコードには通常、コードのロジックや目的を説明するためのコメントやドキュメンテーションが含まれる。AI生成のコードにはこれらのコメントやドキュメンテーションが不足の可能性がある。
3. 人間で書いたコードは、異常処理の考慮が多い場合がある。一方、AI生成のコードはこれらの側面を無視する可能性がある。

本研究では、これらの特徴やその他調査中の特徴を活用して、ランダムフォレストモデルをトレーニングして特徴の重要性を分析し、AI生成コードと人間で書いたコードの本質的な違いを探求する。

二つ目の側面では、CodeBERTなどの大規模事前学習モデルを活用、関連分野の大規模データセットを構築、および上記の特徴重要性分析から得られた結論を参考し、高精度かつ高い汎用性を持つAI生成コード識別モデルを構築する。

3.2　研究の特色（新規性，独創性）

1. 本分野での研究インフラを補足するため、参考文献より大規模かつ高品質なラベル付きデータセットを4000組の人間編集コードとChatGPT生成コードを収集して構築したこと。

2. コモンセンスとドメイン知識を用いたモデルをファインチューニングするアプローチを提案し、AI生成コードと人間編集コードの本質的な違いに関する調査を行い、後続の研究者に参考価値のある結論を提供すること。

3.3　研究経過

関連研究が限られており、かつ参考文献が正式に発表されていないため、本研究の最初のステップとして、必要の基本的なインフラストラクチャを実装し、AI生成コードの検出可能性を初期的検証することである。

前述通り、筆者は参考文献が構築したデータセットは小規模、コード要件が単純と認識し、オンラインテストプラットフォームでのAIで生成したコードを検出する用途に適していないと考えている。したがって、まず一定の規模を持つデータセットを構築する必要がある。CodeNetは、C、C++、Python、Javaなどの言語で構成されて、1400万のコードサンプルが含まれており、それぞれがプログラミング競技プラットフォームAtCoderから抽出された4000のプログラミング問題の解決策の1つである。筆者はCodeNetから全部4000の競技プログラミング問題に対してJavaでのソリューションをOpenAIが提供したAPIを用いて自動生成し、元のCodeNetに含まれるJavaソリューションと共に、本研究のデータセットを構築した。

その後、筆者はデータセットの2300組の2000組をトレーニングセット、大規模事前訓練モデルCodeBERTを基づいてトレーニングとファインチューニングを行い、そして300組のテストセットでモデルを評価した。結果はセクション3.4に示している。

特に注意すべき点では、参考文献ではコードサンプルからコメントとインデントを削除したが、筆者はコメントが本識別タスクに役立つと考え、インデントはコードスタイルの一部であり、ある形式で保持すべきだと考えている。

3.4　得たい結論

先述通り、実験は2000組のAI生成コードと人間編集コードをトレーニングセットとして、CodeBERTに基づくモデルをトレーニングし、その他300組のサンプルをテストデータとしてモデルの性能を評価した。その結果、平均精度が0.96であり、参考文献で実現された精度を達成、あるいは一部を超えていることを示している。これは、より大規模なデータセットの影響を受けている可能性が高く、実験を行う際、データセットは完全に構築されていないため、精度はまだ向上の余地があると考えられる。実験結果により、人間編集コードとAI生成コードの間に潜在的な違いが存在する可能性があることを示している。

3.5　今後の計画

本研究はまだ初期段階であり、必要な基盤を構築し、タスクの実現可能性を検討した上で、今後の研究計画は章3.1で述べたように、二つの継続的なプロセス：特徴の重要性分析およびモデルのファインチューニングに分けて順次行う予定である。

なお、予備実験では、データセットの構成は競技プログラミング問題に限られている、モデルの汎用性に対する評価はまだ行っていないだが、その結果は期待できないと考えられる、今後はモデルの汎用性向上のため、まだ違う分野のコードサンプルを収集し、モデルをファインチューニングする必要がある。

参考文献

1. Nguyen, P.T. et al. (2023). Is this Snippet Written by ChatGPT? An Empirical Study with a CodeBERT-Based Classifier. arXiv:2307.09381 [cs.SE].
2. Feng, Z. et al. (2020). CodeBERT: A Pre-Trained Model for Programming and Natural Languages. Findings of EMNLP 2020.