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	**AI より生成されたソースコードの機械学習による識別手法の研究**				修士課程

1. 研究の背景

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、人々に多くの利便性を提供している。しかしそれに伴い、いくつかの避けられない問題も起こっている。

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（その可能性は否定できない）、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。オンライン面接が一般化する中、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ことが多くなっている。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。プログラミングにはデバッグが不可欠であるため、ほぼすべての企業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、不正行為者がChatGPTを用いた不正行為を行うことを可能としている。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。

こうした状況を踏まえ、AI 生成テキスト識別に関する研究が複数なされている。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。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[1]のみである。

前述した通り、業界には AI 生成コードを識別する有効なツールが存在しておらず、関連する研究も非常に不十分な状況である。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。こうした状況の中、AI 生成コード識別の研究やその実用化が急がれている。

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 はコードを一定の正確性をもって AI 生成と人間編集に分類でき、その精度は 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 研究目的

本研究の目的としては、高精度なモデルを構築し、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ソースコードを有効に分類し、教育もしくは入社テストでの AI 不正応用の検出を目指すことである。

3.2 研究の特色

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より大規模なデータに基づいて構築すること（現状では 4000 組以上のデータセットを想定）。

2. 調査で得られた生成コードの特徴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 研究アプローチ

本研究は以下のプロセスを従い、順次行う予定である。

1. 関連研究が限られており、まだ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は、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の初期的検証をすること。

2. 調査や文献から以下の仮説を設定し、これらの仮説を特徴として、機械学習を通して特徴の重要性分析を行い、AI 生成ソースコードと人間で書いたソースコードの違いを探求すること。

i. AI 生成のコードは通常、一貫したコードスタイルに従い、類似した構造や書式のものが多い。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、異なるスタイルの違いがあると考えられる。

ii. 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少ない可能性がある。

iii. 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。

3. 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、高精度な AI 生成コード識別モデルを構築すること。

3.4 初期的検証

2 章で述べたように、筆者は参考文献が構築したデータセットは小規模、コード要件が単純であるため、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、本研究は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[3]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答例の 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。

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。

前述したように、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を示している。

3.5 得たい結論

初期的検証を踏まえ、本研究は ChatGPT で生成した異なる言語でのソースコードと人間で書いたソースコードが設定した特徴での違いが存在する、そして精度が参考文献を超え、もしくは実用レベルの AI 生成コード検出モデルを構築し、教育もしくは入社テストなどの場合での AI 不正利用検出が可能であることを結論として得たいと考えている。

3.6 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 3.3 で述べたように、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を順次に行う予定である。

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、モデルの汎化性能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、今後はモデルの汎化性能向上のため、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と考えられる。

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 the Association for Computational Linguistics: EMNLP 2020, 1536–1547.

[3] S. -J. Hwang, S. -H. Choi, J. Shin and Y. -H. Choi, "CodeNet: Code-Targeted Convolutional Neural Network Architecture for Smart Contract Vulnerability Detection," in IEEE Access, vol. 10, pp. 32595-32607, 2022.