**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。

まだコロナの流行により、面接は徐々にオンライン化し、多くのIT企業がその変化を積極的に受け入れている。IT企業のエンジニア職で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せず、しかし、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、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では、ChatGPTは90％以上のテストを簡単に通過できる。ChatGPTの不適切な応用は、オンライン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［５］のみ、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されていないである。

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、学界で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生成コードの識別に関する研究は展開する価値が十分である。

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［４］、大規模ｐre-ｔrainedモデルCodeBERT［５］を基に、識別手法「GPTSniffer」を提案し、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、そして識別能力に影響を与える要因を調査した。結果として、GPTSnifferはコードを正確にAI生成または人間編集と分類でき、精度はGPTZeroおよびOpenAI Text分類器、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、2つの側面から行う予定である。

一方、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提出する：

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. AI生成のコードは、機械的であり、人間がプログラミング際の創造性を欠いていると考えられる。一方、人間で書いたコードは通常、より深い思考や意思決定プロセスが反映される。
4. 人間で書いたコードは、エラー処理や異常処理の考慮が多い場合がある、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の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。最終的に、識別規模と想定する

3.2　研究の特色（新規性，独創性）

1. この分野では研究インフラが不足しているため、本研究では、参考文献に比べてより大規模で高品質な注釈付きデータセットを構築した。これは、4000組の人間が書いたコードとChatGPTが生成したコードから成る。

2. この論文は、モデルの微調整に常識とドメイン知識を利用するディープラーニング方法を提供する。このプロセス中にAI生成コードと人間が書いたコードの本質的な違いを議論する予定であり、それが後続の関連研究の参考資料として役立つ。

3.3　研究経過

この章では、これまでの研究進捗と研究経過について簡単に説明する：

関連する研究が限られており、そして参考文献[5]が公式に発表されていないため、本研究の最初のステップは、AIによるコード生成の検出可能性を初めて検証し、必要な基本的なインフラストラクチャを実装することである。この段階では、参考文献[5]と類似したモデルと手法：事前訓練された大規模モデルCodeBERTを基にしたディスクリミネーターを使用する。

参考文献[5]では、データセットの構築に関する情報は提供されているが、具体的なデータセットの場所は示されておらず、さらに筆者は参考文献[5]で構築されたデータセットは小規模、コード要件が単純と考え、オンラインテストプラットフォームでAIが生成したコードを検出する用途には適していないと考えられる。

したがって、この段階での主要なタスクは、研究の用途に合致し、一定の規模を持つデータセットを構築すること。筆者は、IBMが構築したCodeNetデータセットをベースに選択した。CodeNetデータセットには、C、C++、Python、Javaから構成される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ム設計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期待解決策の1つである。筆者はこのデータセットから4000のプログラミング問題すべてに対応するJavaコードをOpenAIのAPIを用いて自動生成し、元のCodeNetデータセットに含まれるJavaソリューションと共に、本研究に使用する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（4000の問題のサンプル全体はまだ構築中）を、トレーニングセット2000組とテストセット300組に分割し、事前訓練された大規模モデルCodeBERTを基にトレーニングとファインチューニングを行い、AIによるコード生成の検出タスクの分類器を実装した。

特に注意すべき点では、参考文献[5]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、筆者はコメントが分類器の識別に役立つと考え、インデントはコードスタイルの一部であるため、ある形式で保持すべきだと考えている。

最後に、筆者は300組のサンプルでモデルを評価し、その結果はセクション3.4に示されている。

3.4　得たい結論

先述したように、本稿では2000組のAI生成コードと人間が書いたコードをトレニンーグデータとして、CodeBERTに基づくモデルをトレーニングし、その他の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、平均精度は0.96であり、参考文献[5]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、データセットが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。これは、人間によって書かれたコードとAIによって生成されたコードの間には潜在的な違いが存在する可能性があることを示唆している。

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示すとともに、今後の研究計画を二つの継続的なプロセスに分けると考えられる。

プロセスA：特徴分析

本研究では、主旨となるAI生成コードと人間で書かれたコードの本質的な違いに関する研究がまだ行われていないであり、データセットの構築が完了し、データクリーニングが終了した後、関連する作業がスケジュールに組み込まれる予定である。

プロセスB：モデルの精緻化

その後、特徴分析の結果は、高精度かつ高汎化性の高いAI生成コード判別モデルを実現するために、データベースとモデルの精緻化を指導する入力として活用されると想定している。

また、データセットの構築に関して、将来的にはスケールを拡大し、さまざまなプログラミング言語のサンプルを豊富に取り入れ、応用ナリオをより広く対応するために、さらに多くのAIモデルによる生成コードを導入する予定である。

参考文献

1. ChatGPT <https://openai.com/blog/chatgpt.>
2. GPTZero <https://gptzero.me/.>
3. OpenAI Text <https://platform.openai.com/ai-text-classifier.>
4. Nguyen, P.T., Di Rocco, J., Di Sipio, C., Rubei, R., Di Ruscio, D., & Di Penta, M. (2023). Computer Science > Software Engineering [Submitted on 18 Jul 2023 (v1), last revised 7 Aug 2023 (this version, v2)]. Is this Snippet Written by ChatGPT? An Empirical Study with a CodeBERT-Based Classifier. arXiv:2307.09381 [cs.SE] <https://doi.org/10.48550/arXiv.2307.09381>.
5. Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D., & Zhou, M. (2020). CodeBERT: A Pre-Trained Model for Programming and Natural Languages. [Findings of EMNLP 2020].