大規模言語モデルを用いた逆コンパイル手法の検討

門谷 拓能

1 はじめに

現代の情報社会で, ソフトウェアの信頼性とセキュリティは重要な課題となっており, 逆コンパイル技術が注目されている. 逆コンパイルとは, バイナリコードを高水準言語に変換し, ソフトウェアの詳細な分析やバグの修正, セキュリティ診断を可能にする技術である. しかし, 逆コンパイルしたコードではそのままコンパイルは難しい.

逆コンパイルツールでは限界がある中, 近年は大規模言語モデル (LLM) の活用注目されている. 本研究では,GPT-4o を用いて, 精度の高い逆コンパイルを実現する. このモデルは膨大なデータセットから学習し, プログラムコードの構造理解と生成能力に優れており, 機械命令から高級言語を生成することが期待されている [1].

具体的には,PolyBench を用いたデータセットで, アセンブリコードから C コードへの変換モデルを開発する. これにより, 正確でコンパイル可能な C コードの生成を目指す.

本研究の目的は, バイナリコードから C コードを出力する大規模言語モデル（LLM）の利用を検討し, コンパイル可能かつ元のソースコードと処理内容が正確なコードを生成することである. さらに, この研究は, 逆コンパイルによる信頼性の高い解析を実現し,LLM の逆コンパイル適用可能性を考える.

2 逆コンパイル手法

本研究ではバイナリコードからアセンブリコードの生成が不可能だったため,LLM によるアセンブリコードから C コードへの変換を行う.LLM は GPT-4o を用いる.GPT-4o はコード数が少なく簡易なアセンブリコードに関しては正しく C コードに変換することができたが, コード数が多く難解なコードに対しては期待されるコードが得られなかった.

ここでファインチューニングを GPT-4o に行う. ファインチューニングの意義は, リソース集約的に訓練された汎用モデルを, ニッチで専門家されたタスクに適用される点にある. データセットは PolyBench を用いて, アセンブリコードとそれに対応する C コードで作成する.PolyBench は, 数値計算に特化したベンチマーク集であり, 多様なアルゴリズムとその実装を含むため, 逆コンパイルに適した多様性を備えている.

また, 入力を PolyBench のアセンブリコードで, 出力を期待される C コードである.

3 生成された C コードの評価

アセンブリコードから C コードへ変換するモデルの性能評価を行う. まず, 生成された C コードのコンパイル可否を検証した. その結果, 逆コンパイルツールと比較して, 本手法により生成された C コードはコンパイル成功率が向上していることが確認された. これは,LLM を用いることで, 適切な構文や変数の一貫性が確保され, エラーの少ないコードが生成されたためである. 次に, 元のソースコードと比較し, 処理内容の一致度を測定した.PolyBench の各プログラムについて, オリジナルのソースコードと生成された C コードの出力結果を比較したところ, 多くのケースで同一の結果が得られた. しかし, 一部のケースでは, 不要な計算ステップが追加されるといった問題がみられた. 表 1 は, 構文エラー修正後の再評価の結果である. これより 30 個ある C コードのうち,18 個がコンパイル可能であった C コードである. また, そのうち,8 個が元のコードと等しくなった.

これらの評価結果からコンパイル成功率や結果の一致度をさらに高めるために, 型安全性とデータ構造の管理を強化する必要がある. 識別子の衝突と型ミスマッチを検出する機能の強化が求められ, これにより生成コードの信頼性の高い逆コンパイルを実現するためには, モデルの構文エラーチェック及び補完の強化, 数値演算精度向上のための訓練, および関数間の整合性チェックの統合が必須である.

表 1: 実験結果

プログラム名	コンパイル	元のコードとの結果比較
2mm	x	x
3mm	○	○
adi	○	x
atax	x	x
bicg	x	x
cholesky	○	x
correlation	x	x
covariance	x	x
deriche	x	x
doitgen	○	○
durbin	○	x
fdtd-2d	○	○
floyd-warshell	○	○
gemm	x	x
gemver	○	○
gesummv	○	○
gramschmidt	○	x
heat-3d	○	x
jacobi-1d	○	x
jacobi-2d	○	x
lu	x	x
ludcmp	x	x
mvt	x	x
nussinov	○	○
seidel-2d	○	○
symm	○	x
syr2k	x	x
syrk	x	x
trisolv	○	x
trmm	○	x

4 おわりに

本研究は, 大規模言語モデル GPT-4o を用いた逆コンパイル手法を提案した. アセンブリコードから C コードへの変換で, ある程度の成果が得られたが, コンパイルエラーや結果の相違も多く見られた. 今後, モデル精度向上に注力し, ソフトウェア開発やセキュリティ分野での実用化を目指す.

参考文献

[1] Dylan Manuel, Nafis Tanveer Islam, Joseph Khoury, Ana Nunez, Elias Bou-Harb, and Peyman Najafirad, “enhancing reverse engineering: Investigating and benchmarking large language models for vulnerability analysis in decompiled binaries”, *arXiv preprint arXiv:2411.04981*, 2024.