Исследование возможностей файн-тюнинга LLM для решения задачи повышения читабельности декомпилированного кода на языке Си

Кислов Константин Александрович*, Божко Артем Александрович, Ременяко Владислав Денисович, Лялин Максим Андреевич
НИЯУ МИФИ,
*e-mail: *kostik_kislov@list.ru*

Аннотация

В большинстве случаев декомпилированный программный код трудно поддается анализу: названия переменных и функций лишены изначального заложенного смысла и трудно прослеживается логика программы. В ходе работы над проектом был обучен адаптер для языковой модели CodeLlama, предназначенный для улучшения декомпилированного кода на языке Си: приближения к исходному коду программы и упрощения для человеческого восприятия. Также исследованы возможности адаптера и проведена оценка его эффективности при решении данной задачи.

Ключевые слова: декомпиляция, языковые модели, fine-tuning.

Классическая проблема в сфере реверс-инжиниринга – это проблема восстановления исходного кода из машинного. Для выполнения данной задачи используются промышленные программы-декомпиляторы. Однако они генерируют код, который, во-первых, нельзя повторно компилировать (лишь у небольшого числа декомпиляторов нет данной проблемы), во-вторых, содержащаяся в нем информация требует от человека значительного времени для анализа. Стремительно развивающиеся технологии в сфере NLP позволяют приблизится к решению данных двух проблем. Из существующих эффективных вариантов для обработки вывода декомпиляторов можно выделить: 1) DIRTY [1] – seq2seq модель на базе трансформера для восстановления изначальных имен и типов переменных; 2) LmPa [2] – система для отправки запросов в ChatGPT с задачей изменить названия переменных и функций, 3) DecGPT [3] – проект, использующий в основе GPT-3.5 для исправления в коде ошибок, возникающих при повторной компиляции.

Приведенные выше и многие другие решения для упрощения анализа кода направлены на внесение небольших изменений в результат работы декомпилятора (например, новых имен переменных и функций) и слабо

затрагивают возможности дообучения существующих мощных LLM для более кардинального его улучшения, вплоть до написания программы со структурой кода, близкой к исходной. В связи с этим возникла идея разработать адаптер на основе предобученной языковой модели для решения задачи интеллектуальной обработки (в нашем случае – повышения читабельности) декомпилированного кода, а также оценить его работоспособность.

В качестве LLM была выбрана модель CodeLlama-7b [4] – дообученная модель Llama 2 для написания, завершения и исправления кода. Обучение адаптера с 1,05 млн. обучаемых параметров (0,015 % от всех параметров модели) проводилось на основании датасета, состоящего из 176 тыс. примеров из исходного кода на Си (часть из которых взята из датасета FormAI Dataset) и соответствующего результата работы декомпилятора Hex-Rays (версия - 8.3.0.230608; компилятор – GCC 11.4.0).

Проведенные тесты показывают, что модель с адаптером, несмотря на относительно небольшие для NLP объем датасета и количество эпох, достигает приличных результатов, в том числе при обработке кода, декомпилированного при помощи программ, примеров вывода которых не было в обучающей выборке (например, RetDec). В дальнейшие планы работы над проектом входят: продолжение обучения модели на датасете большего объема и с примерами работы других декомпиляторов, проведение более масштабного тестирования как с классическими для задачи seq2seq метриками (BLEU, AED и т.д.), так и с оценкой при помощи опроса специалистов и с проверкой возможности перекомпилирования результатов работы нейросети.

Список литературы

1. Qibin Chen, Jeremy Lacomis, Edward J Schwartz, Claire Le Goues, Graham Neubig, and Bogdan Vasilescu. Augmenting decompiler output with learned variable names and types. In 31st USENIX Security Symposium (USENIX Security 22), pages 4327–4343, 2022.
2. Xu Xiangzhe, Zhang Zhuo, Feng Shiwei, Ye Yapeng, Su Zian, Jiang Nan, Cheng Siyuan, Tan Lin and Zhang Xiangyu. LmPa: improving decompilation by synergy of large language model and program analysis. arXiv preprint arXiv:2306.02546v1 (2023).
3. Wai Kin Wong, Huaijin Wang, Zongjie Li, Zhibo Liu, Shuai Wang, Qiyi Tang, Sen Nie and Shi Wu. Refining Decompiled C Code with Large Language Models. arXiv preprint arXiv:2310.06530v2 (2023).
4. CodeLlama-7b-hf. [Электронный ресурс] – URL: https://huggingface.co/codellama/CodeLlama-7b-hf.