

Улучшенный вывод типов в Python с использованием глубокого обучения сходству на основе графового представления кода*

Г.А. Арутюнов, С.М. Авдошин, А.М. Литвиненко
НИУ ВШЭ, ул. Мясницкая, д. 20, г. Москва, Россия

Целью данной работы стало исследование альтернативных подходов к представлению кода, таких как графы абстрактного синтаксического дерева и потока данных. Для этого была выбрана задача определения аннотаций типов в исходном коде программ на языке Python.

В предыдущей работе была разработана модель GraphTyper [1]. Модель является чистым Трансформером без каких-либо изменений в исходную архитектуру. Исходный код в графовом представлении переводится в последовательность благодаря вычислению вложений от собственных векторов и значений матрицы смежности, которые позволяют закодировать структуру графа [2]. Тем не менее, у первой версии модели был ограничен размер словаря. Увеличение размера словаря привело бы к увеличению количества параметров, а значит и веса модели.

Для решения проблемы во второй версии модели был использован метод глубокого обучения сходству [3]. Для обучения модели был собран датасет из 100 репозиторий на языке Python. Исходный код был преобразован в граф посредством метода, описанного в работе Typilus [3]. Далее, граф был преобразован в формат, который может быть обработан моделью TokenGT [2]. Кроме того, была адаптирована триплетная функция потерь из работы Typilus. Наконец, был имплементирован метод предсказания аннотаций типов из полученных векторных вложений с использованием k ближайших соседей. Улучшенная модель GraphTyper продемонстрировала точность 67.8% для типов с параметрами и 81.2% при отбрасывании параметров типов против 50.7% и 64.58% соответственно для первой версии модели.

Улучшения модели GraphTyper позволили разработать универсальную архитектуру модели, которая может обучаться на графовом представлении кода. Данное достижение позволит перейти в будущих исследованиях к более сложным задачам, а именно к применению данной модели для поиска и классификации уязвимостей в исходном коде программ и смарт-контрактов.

Литература

1. Arutyunov G.A., Avdoshin S.M. GraphTyper: Neural Types Inference from Code Represented as Graph // Proceedings of the Institute for System Programming of the RAS. 2024. Vol. 36, no. 4. P. 69–80. DOI: 10.15514/ispras-2024-36(4)-6.
2. Kim J., Nguyen T.D., Min S. et al. Pure Transformers are Powerful Graph Learners // CoRR. 2022. Vol. abs/2207.02505. arXiv: 2207.02505 [cs.LG]. DOI: 10.48550/arXiv.2207.02505.
3. Allamanis M., Barr E.T., Ducousso S., Gao Z. Typilus: Neural Type Hints // Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation. ACM, 2020. P. 91–105. DOI: 10.1145/3385412.3385997.

*Исследование выполнено при поддержке вычислительных ресурсов HPC кластера НИУ ВШЭ.