

АНАЛІЗ МЕТОДІВ СЕМАНТИЧНОГО ПОРІВНЯННЯ ТА ІНТЕЛЕКТУАЛЬНОГО СИНТЕЗУ ПРОГРАМНОГО КОДУ

Вінницький національний технічний університет

Анотація

У роботі проведено аналіз сучасних методів аналізу та формування програмного коду з використанням нейронних мереж. Розглянуто підходи до виявлення структурних розбіжностей у вхідних текстів програм, що базуються на побудові абстрактних синтаксичних дерев (AST) та застосуванні трансформерних архітектур для генерації рішень. Встановлено, що традиційні рядкові методи порівняння (line-based diff) є недостатньо ефективними для вирішення складних логічних конфліктів злиття. Найбільш перспективним напрямком є використання контекстних ембедінгів кодів, що дозволяють автоматизувати процес синтезу коду та знизити кількість ручних втручань при інтеграції програмних модулів.

Ключові слова: семантичний аналіз коду, нейронні мережі, конфлікти злиття, трансформатори.

Abstract

An analysis of modern methods for analyzing and generating program code using neural networks has been performed. Approaches to detecting structural discrepancies in source texts based on building abstract syntax trees (AST) and using transformer architectures for generating solutions are considered. It is established that traditional line-based diff methods are insufficient for resolving complex logical merge conflicts. The most promising direction is the use of contextual code embeddings, which allow automating the code synthesis process and reducing manual intervention during software module integration.

Keywords: semantic code analysis, neural networks, merge conflicts, transformers.

Вступ

У сучасному циклі розробки програмного забезпечення (SDLC) швидкість інтеграції змін є критичним фактором. Використання систем контролю версій (VCS) дозволяє командам працювати паралельно, проте етап злиття гілок коду часто супроводжується виникненням конфліктів, які потребують ручного вирішення [1]. Стрімкий розвиток великих мовних моделей (LLM) та спеціалізованих нейронних мереж для роботи з кодом відкриває нові можливості для автоматизації цих процесів.

Аналіз існуючих підходів

Сучасний стан систем контролю версій демонструє певну обмеженість у використанні алгоритмів порівняння, що базуються на знаходженні найдовшої спільної підпоследовності, оскільки такі методи ігнорують логічну структуру програмного забезпечення та сприймають вихідний код як набір непов'язаних текстових рядків. Це призводить до виникнення значної кількості технічних конфліктів при злитті гілок, які насправді не порушують цілісність програми, але потребують ручного втручання розробника.

Запропонована інформаційно-аналітична система вирішує цю проблему шляхом переходу від лінійного до ієрархічного аналізу через побудову абстрактних синтаксичних дерев. Використання бібліотек швидкого парсингу, зокрема tree-sitter, дозволяє трансформувати вихідний текст у структурований граф, що відображає синтаксичні зв'язки між функціями, класами та змінними незалежно від стилю форматування. На основі отриманого дерева формується розширений граф семантичних залежностей, який інтегрує в себе потоки керування та даних, що дозволяє системі відстежувати вплив змін у віддалених частинах проекту [2].

Центральним компонентом аналітичного ядра системи є нейромережева модель на базі архітектури трансформерів, яка перетворює вузли графа у високорозмірні векторні представлення – контекстні ембедінги. Застосування попередньо тренованих моделей типу CodeBERT дозволяє коду бути представленим у семантичному просторі, де близькість векторів визначається не схожістю написання, а ідентичністю функціонального призначення [3]. Процес класифікації конфліктів реалізується через обчислення косинусної подібності між ембедінгами різних версій коду, що дає змогу математично розділити зміни на тривіальні, які можуть бути синтезовані автоматично, та критичні, що потребують візуалізації для прийняття рішення людиною.

Висновки

Проведений аналіз показав, що інтеграція нейромережевих моделей у процеси формування та синтезу коду дозволяє значно підвищити рівень автоматизації розробки. Перехід від текстового до семантичного аналізу на основі графів та контекстних ембедінгів є ключовим для створення інтелектуальних систем підтримки прийняття рішень, які здатні зменшити когнітивне навантаження на розробників при вирішенні конфліктів злиття.

СПИСОК ВИКОРИСТОНОЇ ЛІТЕРАТУРИ

1. Continuous Integration: Improving Software Quality and Reducing Risk [[Continuous Integration: Improving Software Quality and Reducing Risk \(Addison-Wesley Signature Series \(Fowler\)\)](#)]
2. A Survey of Machine Learning for Big Code and Naturalness [[A Survey of Machine Learning for Big Code and Naturalness](#)]
3. CodeBERT: A Pre-Trained Model for Programming and Natural Languages [[CodeBERT: A Pre-Trained Model for Programming and Natural Languages](#)]

Бігас Олександр Сергійович — студент групи ІПІ-25м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: alex.bigas.2004@gmail.com

Bihás Oleksandr — Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: alex.bigas.2004@gmail.com