УДК 004.4:004.8

ВИКОРИСТАННЯ ГЕНЕРАТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ СТВОРЕННЯ АРТЕФАКТІВ ТЕСТУВАННЯ ПРОЄКТІВ З ПОБУДОВИ ПРОГРАМНИХ ПРОДУКТІВ

Павленко Є. В., аспірант, Pavlenko.Yeh.V@nmu.one, НТУ «Дніпровська політехніка»
Гнатушенко В. В., д.т.н., професор, Hnatushenko.V.V@nmu.one, НТУ «Дніпровська політехніка»

Протягом останньої пари років спостерігається значний розвиток генеративного штучного інтелекту (ШІ) на основі великих мовних моделей (LLM) з перетворювачами (transformers), та сервісів, побудованих на них, таких як ChatGPT від OpenAI, Gemini від Google, Claude від Anthropic, GitHub Copilot, Amazon CodeWhisperer та інші.

Тестування має важливу роль у розробці програмного забезпечення (ПЗ). Тестування не тільки життєво необхідне для забезпечення якості ПЗ, але також відіграє значну роль у рефакторингу або переході між різними типами проєктів (proof of concept, MVP тощо). Наявність тестового покриття особливо важлива при роботі над проєктами, що використовують гнучку методологію (Agile). Під час рефакторингу та міграцій важливо переконатися, що функціонал та дані залишаються коректними протягом усього процесу. Недостатнє тестове покриття може призвести до обмежень можливостей вносити зміни, що, у свою чергу, впливає на розвиток проєкту. Це також може вимагати широкого та всеосяжного ручного тестування, тим самим уповільнюючи прогрес проєкту. Створення належного тестового покриття для проєкту, особливо підтримка працездатності тестів під час постійних змін проєкту на етапі активної розробки, є трудомістким завданням. Попри те, що тестування програмного забезпечення відіграє вирішальну роль, багато проєктів тестуються неналежним чином. Це пов'язано зі складністю тестування як такого.

Все це визначає актуальність розробки прикладних методів, теоретичної та алгоритмічної бази для створення артефактів проєктів у сфері тестування програмного забезпечення, для побудови програмних продуктів із використанням генеративного штучного інтелекту на основі великих мовних моделей із перетворювачами. На нашу думку генеративний ШІ дозволить проводити більш комплексне тестування без збільшення ресурсів, виділених на тестування. Ця робота присвячена визначенню основних вимог, обмежень та можливостей застосування генеративного штучного інтелекту для створення артефактів тестування проєктів з побудови програмних продуктів.

Генеративний ШІ на великих мовних моделях – це нова технологія, ми лише починаємо розуміти його можливості та потенційні сфери застосування. Часто тести та тестові дані, дуже схожі між собою. Під час активної розробки проєкту та значних змін у його кодовій базі потрібні значні зусилля та час для

підтримки працездатності тестів. Така робота видається придатною для автоматизації. Однак, автоматизація на основі заздалегідь визначених процедур не дає бажаного ефекту. Є сподівання, що генеративний ШІ на основі великих мовних моделей із перетворювачами міг би допомогти у розв'язанні наступних завдань: 1) аналіз вимог; 2) генерація тест-планів; 3) генерація сценаріїв тестів; 4) генерація моків (mocks); 5) генерація тестових даних; 6) розширення та доповнення існуючих наборів тестів; 7) підтримка актуальності тестів; 8) виявлення та прогнозування дефектів; 8) оптимізація тестів; 9) написання звітів про помилки; 10) створення документації.

Проведений нами аналіз відповідної літератури [1-3] дозволяє визначити можливі (очікувані) переваги застосування генеративного ШІ: "самолікування" наборів тестів при внесенні змін до вихідного коду; генерація тестів у фоновому режимі або без використання ресурсів, призначених для розробки; аналіз інтерфейсу користувача; підвищення продуктивності праці спеціалістів-людей; генерація тестів із моделей або мов опису даних; генерація тестів з неформальних описів, документації, написаної природними мовами, або аудіо описів; зниження витрат на обслуговування шляхом усунення надлишкових тестів.

Однак на цьому етапі розвитку генеративного ШІ спроби його використання пов’язані з наступними труднощами: розроблені підходи та інструменти можуть застаріти до їх завершення з публікацією більш досконалих моделей; висока мінливість у генерованих тестах; неправильна генерація коду; проблеми з продуктивністю через залежність від зовнішніх систем; витрати, пов'язані з використанням сервісів генеративного ШІ; розкриття коду проєкту під час використання зовнішніх сервісів, що може бути неприйнятним у деяких випадках.

Висновок. Таким чином розвиток використання генеративного ШІ для тестування підвищить рівень тестового покриття, покращить співвідношення ціна-ефективність тестування та розширить упровадження практик тестування серед компаній і проєктів. Наші подальші дослідження будуть присвячені розробці комплексу автоматизованих та автоматичних методів використання генеративного штучного інтелекту для вирішення задачі створення артефактів тестування проєктів з побудови програмних продуктів.

Список використаних джерел

1. Nguyen-Duc Anh, Cabrero-Daniel Beatriz et al. Generative Artificial Intelligence for Software Engineering – A Research Agenda. https://doi.org/10.48550/arXiv.2310.18648
2. Fan Angela, Gokkaya Beliz, Harman Mark, Lyubarskiy Mitya, Sengupta Shubho, Yoo Shin, Zhang Jie M. Large Language Models for Software Engineering: Survey and Open Problems. https://doi.org/10.48550/arXiv.2310.03533
3. Wang Junjie, Huang Yuchao, Chen Chunyang, Liu Zhe, Wang Song, Wang Qing. Software Testing with Large Language Model: Survey, Landscape, and Vision. https://doi.org/10.48550/arXiv.2307.07221