

MÔ HÌNH TRANSFORMER CHO BÀI TOÁN TỰ SINH TESTCASE TRONG KIỂM THỬ PHẦN MỀM

Nguyễn Thành Trung -20520831

Tóm tắt

- Lớp: CS519.011
- Link Github của nhóm: <https://github.com/nttrung309/CS519>
- Link YouTube video:
- Ảnh + Họ và Tên của các thành viên



Nguyễn Thành Trung - 20520831

Giới thiệu

- **Thống kê hiện tại:** Dữ liệu năm 2023 từ Global App Testing cho thấy 2/3 công ty sử dụng tỷ lệ kiểm thử 75:25 hoặc 50:50 (kiểm thử thủ công:tự động), với chỉ 9% thực hiện kiểm thử hoàn toàn thủ công.
- **Thách thức của phương pháp kiểm thử truyền thống:** Trong môi trường phát triển phần mềm ngày nay, phương pháp kiểm thử truyền thống gặp khó khăn trong việc đáp ứng tính đa dạng của quá trình phát triển.
- **Lợi ích của tự động hóa kiểm thử:** Tự động hóa kiểm thử mang lại giảm công sức và tăng hiệu quả, nhằm đối mặt với sự phức tạp ngày càng tăng trong quá trình kiểm thử.
- **Thách thức trong tự động hóa:** Phương pháp tự động hóa thường gặp khó khăn khi đối mặt với dự án lớn và sự biến động nhanh chóng trong mã nguồn, dẫn đến sự không linh hoạt và chi phí cao.
- **Mục tiêu của đề tài nghiên cứu:** Đề tài hướng đến việc cải thiện khả năng tự động hóa trong quá trình tạo trường hợp kiểm thử, sử dụng mô hình Transformer linh hoạt, ít tốn công sức hơn.

Mục tiêu

- Mục tiêu chính của đề tài là phát triển một hệ thống tự động hóa có khả năng tạo ra các trường hợp kiểm thử từ mã nguồn và tài liệu liên quan. Chúng tôi đặt ra bài toán làm thế nào có thể sử dụng mô hình Transformer và các kỹ thuật học máy để hiệu quả và tự động hóa quá trình này. [3]
- Tối ưu hóa mô hình và quy trình tự động hóa để đảm bảo rằng các testcase không chỉ đáp ứng yêu cầu chức năng mà còn tối ưu hóa thời gian và công sức kiểm thử.
- Tích hợp liên tục và tự động hóa quy trình kiểm thử, tăng cường khả năng triển khai và kiểm thử liên tục trong quá trình phát triển phần mềm.

Nội dung và Phương pháp

- **Nội dung:**

- Biến thể và cải tiến của mô hình Transformer: Đi sâu vào cơ chế Attention, bao gồm self-attention và multi-head attention. Áp dụng xử lý ngôn ngữ tự nhiên với BERT để tăng cường khả năng sinh test case từ mã nguồn và tài liệu liên quan.
- Tự sinh test case từ mã nguồn và tài liệu: Nghiên cứu về phương pháp tự sinh test case và tích hợp thông tin từ tài liệu đặc tả yêu cầu. Áp dụng kiểm thử dựa trên rủi ro để ưu tiên trường hợp kiểm thử dựa trên mức độ rủi ro.
- Fine-tuning và quy trình tự động hóa linh hoạt: Sử dụng kỹ thuật Fine-tuning để điều chỉnh mô hình Transformer cho kiểm thử phần mềm. Nghiên cứu về quy trình tự động hóa linh hoạt, có khả năng thích nghi với sự thay đổi nhanh chóng trong mã nguồn.

Nội dung và Phương pháp

- **Phương pháp:**

- Tiến hành đánh giá và so sánh giữa các biến thể của mô hình Transformer. Nghiên cứu về transfer learning và ensemble learning trong ngữ cảnh kiểm thử tự động.
- Nghiên cứu kỹ thuật Active Learning để mô hình có thể tận dụng sự tham gia của người kiểm thử và áp dụng học tăng cường để tương tác và học từ môi trường kiểm thử thực tế.
- Thực hiện thử nghiệm trên nhiều dự án phần mềm để đánh giá hiệu suất và sự chuyên sâu của mô hình. Sử dụng kỹ thuật fine-tuning và ensemble learning để cải thiện kết quả thử nghiệm.
- Sử dụng kết quả thử nghiệm để tối ưu hóa mô hình và điều chỉnh quy trình tự động hóa. Áp dụng kỹ thuật auto-configuration để mô hình có thể thích ứng với môi trường và yêu cầu cụ thể.

Kết quả dự kiến

- Tìm hiểu cơ chế Attention, bao gồm self-attention và multi-head attention. Áp dụng xử lý ngôn ngữ tự nhiên với BERT để tăng cường khả năng sinh test case từ mã nguồn và tài liệu liên quan.
- Nghiên cứu về phương pháp tự sinh test case và tích hợp thông tin từ tài liệu đặc tả yêu cầu. Áp dụng kiểm thử dựa trên rủi ro để ưu tiên trường hợp kiểm thử dựa trên mức độ rủi ro.
- Sử dụng kỹ thuật Fine-tuning để điều chỉnh mô hình Transformer cho kiểm thử phần mềm. Nghiên cứu về quy trình tự động hóa linh hoạt, có khả năng thích nghi với sự thay đổi nhanh chóng trong mã nguồn.

Tài liệu tham khảo

- [1]. Yan Xiao, Xinyue Zuo, Lei Xue, Kailong Wang, Jin Song Dong, Ivan Beschastnikh: Empirical Study on Transformer-based Techniques for Software Engineering. 2023: 1-10
- [2]. Shaukat Ali; Lionel C. Briand; Hadi Hemmati; Rajwinder Kaur Panesar-Walawege: A Systematic Review of the Application and Empirical Investigation of Search-Based Test Case Generation. IEEE 2010: 1-50
- [3]. Chengcheng Wan, Shicheng Liu, Sophie Xie, Yifan Liu, Henry Hoffmann, Michael Maire, Shan Lu: Automated testing of software that uses machine learning APIs. ICSE '22: 1–20

Tài liệu tham khảo

- [4]. Mitesh Soni: End to End Automation on Cloud with Build Pipeline: The Case for DevOps in Insurance Industry, Continuous Integration, Continuous Testing, and Continuous Delivery. IEEE 2015: 1-50
- [5]. Julia Ling, Erin Antono, Saurabh Bajaj, Sean Paradiso, Maxwell Hutchinson, Bryce Meredig, Brenna M. Gibbons: Machine Learning for Alloy Composition and Process Optimization. ASME 2018: 1-10