Abstract

GUI testing on Android app is a critical process in ensuring software quality, but it often requires significant manual effort in both designing test cases and implementing test procedures. This manual dependency leads to inefficiencies, high maintenance costs, and limited scalability, especially in rapidly evolving user interfaces.

To address these challenges, several researches focus on test generation and automation and hope to build a system to create test cases with only Android app as input. However, there’s still a limit in current researches.

Instead of focusing on test generation, this research presents a new approach to GUI testing using Large Language Model for testers to automate their tests based on natural language instructions. The system accepts user-described step-by-step actions and expected outcomes in a conversational format, and automatically transforms them into executable test scripts with corresponding result verifications.

Unlike traditional image-based verification methods, the system interprets flexible descriptions of expected results, enabling it to operate robustly across different devices, screen sizes, and application versions. Experimental results demonstrate that the proposed approach significantly reduces the need for manual intervention while maintaining high accuracy in both test execution and result evaluation, offering a more adaptable and efficient solution for modern GUI testing tasks.

Acknowledgement

I would like to sincerely thank my advisor, Professor Yi-Yu Liu, for their invaluable guidance, encouragement, and support throughout the course of my research. Their professional insight and patient mentorship have been fundamental to the completion of this thesis.

I am also deeply grateful to Heroic Faith Medical Science, where I conducted part of this research. In particular, I would like to thank Yuan-Ren Cheng, Chief Executive Officer at Heroic Faith Medical Science, for their support and recognition of the academic value of this work. Their encouragement allowed me to explore the practical relevance of my research in a real-world setting.

My sincere thanks also go to the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, for providing a supportive academic environment and the necessary resources throughout my graduate studies.

Last but not least, I am truly thankful to my family and friends for their continuous support, understanding, and motivation throughout this journey.

Introduction

Graphical User Interface (GUI) testing plays a critical role in mobile applications, ensuring the functionality and usability of modern software applications. However, with the proliferation of mobile applications, there’s an increasing demand of advanced functionality. Operations on mobile applications became more complex day by day, and leads to inevitable introduction of bugs.

Traditional GUI testing processes often rely heavily on manual efforts, such as designing test cases and implementing test scripts. These tasks are time-consuming, error-prone, and difficult to scale, especially when dealing with frequent GUI updates or multiple platform versions. Moreover, manual interpretation of expected results makes the testing process subjective and inconsistent.

To address the challenges in GUI testing, most of existing researches focus on building an automated tool to detect crash bugs [], and few of them focus on detecting non-crash functional bugs []. Various methods have been developed to generate test cases rely only on system under test, but comparing research shows that these approaches were not mature enough to detect most of bugs.

Due to immaturity of test automation researches, there should be an approach between fully automated tools and fully manual approach. In this paper, we proposed Testing & standard development automation procedure (TSDAP), with introduction of human test instruction, we can avoid the problem of low test case quality and coverage. To overcome inconsistence of manual test, we delegated the missions of operating GUI and asserting test result to Large Language Model (LLM), providing an ability to detect specific objects on GUI instead of whole screenshot.

Background

GUI testing

Traditional approach & result

Recent approach & result

Formal paper with simpler assertion method

References