Semantic Matching of GUI Events for Test Reuse:

AreWe There Yet?

...Test reuse approaches automatically migrate human-designed GUI tests from a source app to a target app that shares similar functionalities....

...GUI test reuse approaches generate new tests for a target app by migrating tests designed for a source app, an application that shares similar functionalities with the target app....

*Keywords:*  
TEST, GUI, MATCHING, Semantic matching, APP, GUI tests, GUI events, COMPONENT, test reuse approaches, target app  
  
*Digest:*

...Semantic Matching of GUI Events for Test Reuse: AreWe There Yet?...

...Test reuse approaches automatically migrate human-designed GUI tests from a source app to a target app that shares similar functionalities....

...Semantic matching of GUI events plays a crucial role in these approaches....

...We report several key findings that indicate how to improve semantic matching of test reuse approaches, propose SemFinder a novel semantic matching algorithm that outperforms existing solutions, and identify several interesting research directions....

...A GUI test consists of (i) a sequence of events that interact with the GUI, and (ii) one or more assertion oracles that predicate on the GUI state....

...GUI test reuse approaches generate new tests for a target app by migrating tests designed for a source app, an application that shares similar functionalities with the target app....

...When test migration succeeds, GUI test reuse approaches (i) generate semantically meaningful GUI tests that properly exercise the functionalities of the target app, and (ii) adapt semantically relevant oracle assertions to the target app [15, 45], thus addressing the main limitations of GUI test generators....

...As such, automatic approaches try to migrate GUI tests across apps by mapping semantically similar GUI events....

...These two approaches successfully migrate nontrivial test cases, showcasing the potential of test reuse....

...ATM and CraftDroid combine semantic matching of GUI events with test generation....

...Semantic matching of GUI events identifies ISSTA '21, July 11-17, 2021, Virtual, Denmark Leonardo Mariani, Ali Mohebbi, Mauro Pezzè, and Valerio Terragni !es 1 = click es 4 = fill es e3s = fill 2 = fill es 5 = fill os 1 = exist(SM) es 6 = fill et1= click ot1= exist(SM) et3= fill et2= click et4= fill et5= click et6= fill et7= fill et8= click et9= click (A) Source test case ts for Rainbow app (B) Target test case tt for Yelp app Figure 1: Test reuse example, the target test cases (B) is obtained by migrating the source test case (A) semantically similar events across source and target apps, by applying word embedding techniques [61] to the textual descriptors of events in the GUI widgets....

...Test generation exploits the similarities identified with semantic matching to migrate GUI tests from the source to the target app....

...The overall effectiveness of test reuse strongly depends on the effectiveness of semantic matching of GUI events....

...Indeed, the semantic matching is what drives the matching of the events between the source and the target test....

...In summary, this paper : develops the first framework to automatically evaluate the semantic matching of GUI events; : identifies and extracts the core components of the semantic matching exploited in current test reuse approaches; : evaluates 253 configurations of the semantic matching, and reveals important insights; : proposes a new semantic matching algorithm (SemFinder) and a corpus of documents that outperform existing ones; : makes our framework implementation and all data publicly available, for future research in this area [53]....

An Evolutionary Approach to

Adapt Tests Across Mobile Apps

..Both approaches generate new tests for a recipient app, by adapting the tests designed for some donor app that shares semantically similar functionalities with the recipient app....

...ADAPTDROID explores the huge space of GUI tests with a fitness function that rewards the tests that are most similar to the donor test....

...Given two Android apps AD (donor), AR (recipient), and a "donor" test tD for AD, ADAPTDROID generates a "recipient" test tR that tests AR as tD tests AD....

*Keywords:*  
APP, WIDGET, ADAPT, EVENT, ADAPTDROID, DONOR, APPLICATION, SIMILAR, GUI applications, tests across similar, GUI tests across similar  
  
*Digest:*

...An Evolutionary Approach to Adapt Tests Across Mobile Apps Leonardo Mariani, Mauro Pezz`eyz, Valerio Terragniy and Daniele Zuddasy University of Milano Bicocca, Milan, Italy yUniversit`a della Svizzera italiana, Lugano, Switzerland zSchaffhausen Institute of Technology, Schaffhausen, Switzerland leonardo.mariani@unimib.it -fmauro.pezze, valerio.terragni, daniele.zuddasg@usi.ch Abstract-Automatic generators of GUI tests often fail to generate semantically relevant test cases, and thus miss important test scenarios....

...To address this issue, test adaptation techniques can be used to automatically generate semantically meaningful GUI tests from test cases of applications with similar functionalities....

...In this paper, we present ADAPTDROID, a technique that approaches the test adaptation problem as a search-problem, and uses evolutionary testing to adapt GUI tests (including oracles) across similar Android apps....

...GUI testing is a popular way to verify the behavior of GUI applications, which amounts to design and execute GUI test cases....

...A GUI test case (GUI test in short) consists of (i) a sequence of events that interact with the GUI, and (ii) assertion oracles that predicate on the GUI state....

...Because manually designing GUI tests is expensive, many automatic GUI test generators have been proposed....

...Both approaches generate new tests for a recipient app, by adapting the tests designed for some donor app that shares semantically similar functionalities with the recipient app....

...Test adaptation, when successful, addresses the limitations of existing GUI test generators: (i) it yields to semantically meaningful GUI tests that characterize canonical usages of the app under test....

...CRAFTDROID and APPTESTMIGRATOR explore a GUI model of the recipient app to find a sequence of events that maximize the semantic similarity with the events of the donor test....

...Both techniques greedily explore a single test adaptation scenario, missing the many alternative adapted tests that could be generated starting from a same donor test....

...ADAPTDROID explores the huge space of GUI tests with a fitness function that rewards the tests that are most similar to the donor test....

...This paper presents ADAPTDROID to adapt GUI tests (including oracles) across interactive applications that implement similar functionalities....

...Given two Android apps AD (donor), AR (recipient), and a "donor" test tD for AD, ADAPTDROID generates a "recipient" test tR that tests AR as tD tests AD....

...Automatically adapting GUI tests across apps presents three main challenges: 1) Huge space of GUI tests The space of the possible GUI tests grows exponentially with both the length of the donor test and the number of widgets in the recipient app [1]....

...Adapting tests requires an effective search strategy that recognizes the relevant GUI events in the recipient app....

GUI Widget Detection and Intent Generation via

Image Understanding

*Keywords:*  
SOFTWARE, IMAGE, WIDGET, Aerospace control software, TECHNOLOGY, ELEMENT, METHOD, GUI Widget, GUI element, Testing Technology  
  
*Digest:*

...ABSTRACT Aerospace control software is the most important part of aerospace software....

...The traditional testing method has been challenging to meet its development requirements, and software automation testing has gradually become the main tool for testing aerospace control software....

...For the automation testing of aerospace control software, the core problem is to locate the GUI widgets on the software screenshots and identify their intent, which directly affects the accuracy of the test....

...Because of this, we use the widget recognition technology based on image matching and use the image understanding and analysis technology to extract the widget image in the screenshots....

...INDEX TERMS GUI widget detection, GUI widget intent generation, aerospace control software....

...I. INTRODUCTION As a core part of aerospace software, aerospace control software has a unique working environment and strict requirements on reliability and functionality....

...With the increasing requirements for developing control software in aerospace missions, traditional testing methods can no longer meet the new development requirements....

...System-level functional testing is a critical task in the development of aerospace models and their control software....

...Automated testing technology can provide an excellent convenience for aerospace control software system-level functional testing....

...We adopt GUI element detection technology based on image matching to meet software characteristics under different operating systems and development tools, such as various types and signicant differences in interface styles....

...The intention to generate the widget image is to summarize the image into text and describe the target information in the widget image and the relationship between the targets through a paragraph of text to assist the computer in understanding and facilitating automated testing.We use a convolution neural network (CNN) to extract image features and then use the encoder module to encode this extracted information feature into a tensor....

...In summary, we declare the following contributionsV We propose a new method to improve the automated testing level of general software and aerospace control software through deep image understanding....

...The aerospace control software system is characterized by many parallel tasks, complex software logic and interfaces, and harsh external and electromagnetic environments for software operation....

...Automated testing technology can provide an excellent convenience for aerospace control software testing....

...B. GUI ELEMENT DETECTION The traditional automated testing technology mainly obtains the test source GUI object through the API function of the automated testing framework....

...GUI elements can be roughly divided into text elements and nontext elements....

**Feature Matching-based Approaches to Improve the**

**Robustness of Android Visual GUI Testing**

..CCS Concepts: : Software and its engineering→Software defect analysis; Software verification and validation; Software testing and debugging; : Computing methodologies → Matching; Additional KeyWords and Phrases: Mobile computing, software testing, visual GUI testing, feature matching ACM Reference format: Luca Ardito, Andrea Bottino, Riccardo Coppola, Fabrizio Lamberti, Francesco Manigrasso, Lia Morra, and Marco Torchiano....

...Many of these tools identify the GUI widgets through layout properties of the GUI structure that serve as textual locators and are hence called Layout-based testing tools....

...Our experimental results show how our approach can increase the robustness of visual locator matching, improve over state-of-the-art VGT tools, and achieve higher portability across devices....

*Keywords:*  
APP, DEVICE, TESTING, FEATURE, TEST, TOOL, visual locators, Visual GUI Testing, GUI Testing, VGT tools  
  
*Digest:*

...To soften these issues, we evaluate two feature matching-based approaches for widget detection in VGT scripts, which use, respectively, the complete full-screen snapshot of the application (Fullscreen) and the cropped images of its widgets (Cropped) as visual locators to match on emulated devices....

...Our analysis includes validating the portability of different feature-based visual locators over various apps and devices and evaluating their robustness in terms of cross-device portability and correctly executed interactions....

...CCS Concepts: : Software and its engineering→Software defect analysis; Software verification and validation; Software testing and debugging; : Computing methodologies → Matching; Additional KeyWords and Phrases: Mobile computing, software testing, visual GUI testing, feature matching ACM Reference format: Luca Ardito, Andrea Bottino, Riccardo Coppola, Fabrizio Lamberti, Francesco Manigrasso, Lia Morra, and Marco Torchiano....

...The vast majority of Android apps are Graphical User Interface (GUI) intensive and collect most of the user's interaction through GUI widgets, or Views....

...Many of these tools identify the GUI widgets through layout properties of the GUI structure that serve as textual locators and are hence called Layout-based testing tools....

...These tools, however, are unable to test the actual graphical appearance of the Application Under Test (AUT) when shown to its final user....

...To address this limitation, researchers proposed to tackle app testing with the Visual GUI Testing (VGT) paradigm [2]....

...With this paradigm, the verification of behavior's correctness involves a visual comparison between the app's current and expected visual appearance....

...These features make VGT techniques optimal for testing those applications that need to provide high portability across different platforms....

...Therefore, since every app must be compatible with many different devices (with varying screen sizes, pixel densities, and rendering specifications), marginal variations in the graphical rendering can invalidate the recognition of visual locators and oracles....

...Moreover, graphical changes in the same app's consecutive releases may break the tests, requiring additional maintenance in existing test suites to adapt locators and oracles....

...Our previous studies have shown that state-of-the-art VGT testing tools are complicated to port to different devices unless leveraging hybrid techniques that regenerate VGT tests for each device from an original layout-based test suite [22]....

...Our experimental results show how our approach can increase the robustness of visual locator matching, improve over state-of-the-art VGT tools, and achieve higher portability across devices....

...First, we validate the portability of different locator matching strategies and feature descriptors on different devices over a large set of Android apps by comparing recall, precision, and execution time....

...Second, we perform an exploratory study on a real test suite, and we compare our feature matching approaches with state-of-the-art VGT tools in terms of resilience to device change....

FrUITeR: A Framework for Evaluating UI Test Reuse

..FrUITeR: A Framework for Evaluating UI Test Reuse Yixue Zhao yixue.zhao@usc.edu University of Southern California USA Justin Chen justin.chen@columbia.edu Columbia University USA Adriana Sejfia sejfia@usc.edu University of Southern California USA Marcelo Schmitt Laser marcelo.laser@gmail.com University of Southern California USA Jie Zhang jie.zhang@ucl.ac.uk University College London UK Federica Sarro f.sarro@ucl.ac.uk University College London UK Mark Harman mark.harman@ucl.ac.uk University College London UK Nenad Medvidovic neno@usc.edu University of Southern California USA ABSTRACT UI testing is tedious and time-consuming due to the manual effort required....

...Recent research has explored opportunities for reusing existing UI tests from an app to automatically generate new tests for other apps....

...We introduce FrUITeR, a framework that automatically evaluates UI test reuse in a reproducible way....

*Keywords:*  
TEST, TECHNIQUE, EVALUATE, APP, FRUITER, Test Reuse, EVENT, USA, existing technique's, test-reuse techniques  
  
*Digest:*

...FrUITeR: A Framework for Evaluating UI Test Reuse Yixue Zhao yixue.zhao@usc.edu University of Southern California USA Justin Chen justin.chen@columbia.edu Columbia University USA Adriana Sejfia sejfia@usc.edu University of Southern California USA Marcelo Schmitt Laser marcelo.laser@gmail.com University of Southern California USA Jie Zhang jie.zhang@ucl.ac.uk University College London UK Federica Sarro f.sarro@ucl.ac.uk University College London UK Mark Harman mark.harman@ucl.ac.uk University College London UK Nenad Medvidovic neno@usc.edu University of Southern California USA ABSTRACT UI testing is tedious and time-consuming due to the manual effort required....

...Recent research has explored opportunities for reusing existing UI tests from an app to automatically generate new tests for other apps....

...We introduce FrUITeR, a framework that automatically evaluates UI test reuse in a reproducible way....

...We apply FrUITeR to existing test-reuse techniques on a uniform benchmark we established, resulting in 11,917 test reuse cases from 20 apps.We report several key findings aimed at improving UI test reuse that are missed by existing work....

...CCS CONCEPTS - Software and its engineering; KEYWORDS Software Testing, Test Reuse, Mobile Application, Open Science ACM Reference Format: Yixue Zhao, Justin Chen, Adriana Sejfia, Marcelo Schmitt Laser, Jie Zhang, Federica Sarro, Mark Harman, and Nenad Medvidovic....

...To reduce the manual effort of writing usage-based tests, recent research has explored reusing existing tests in a source app to generate new tests automatically for a target app [20, 21, 23, 25, 28]....

...This suggests that it is possible to reuse existing UI tests across appsÐin effect generating the tests automaticallyÐby mapping similar GUI elements....

...Four recent techniques have targeted usage-based test reuse across Android apps [20, 21, 23, 25].1 While these techniques have shown promise, we have identified five important limitations that hinder their comparability, reproducibility, and reusability....

...It is possible that events are transferred correctly, but the generated test is łwrongž....

...This can be, e.g., because a generated test is missing events and thus not executable....

...Moreover, the metrics used in existing work are not standardized even when evaluating same aspects of different techniques, making it difficult to compare the techniques....

...This imposes a practical limit on the number of tests that can be evaluated....

...In this paper, we focus on Android apps due to the availability of a larger number of existing techniques to evaluate, although in principle our work is not limited to Android....

...For instance, GTM [20] and ATM [21] require access to apps' code, and cannot be directly compared with techniques evaluated on close-sourced apps....

...FrUITeR consists of three key elements: a set of new evaluation metrics that consolidate the metrics used by existing techniques and expand them to measure important aspects that are currently missed; two baseline UI test-reuse techniques that establish the lower-and upper-bounds for the evaluation metrics; and an automated workflow that modularizes UI test-reuse functionality and significantly reduces the manual effort....

...With FrUITeR, one can automatically evaluate test-reuse techniques on apps/tests of interest against the same baseline, thus opening the possibility of large-scale studies....

...Our benchmark currently contains 20 subject apps with 239 test cases, involving 1,082 GUI events....

Mining Android API Usage to Generate Unit Test Cases for

Pinpointing Compatibility Issues

...Experimental results on thousands of real-world Android apps show that JUnitTestGen is effective in generating valid unit test cases for Android APIs....

...There is hence a need to automatically generate compatibility unit tests for Android APIs....

...These test cases subsequently allow our approach to pinpoint various types of compatibility issues, outperforming a state-of-the-art generic test generation tool named EvoSuite, which can only generate test cases to reveal a small subset of compatibility issues....

*Keywords:*  
Compatibility Issues, APIS, Unit Test, API, FRAMEWORK, GENERATE, Android APIs, Android apps, Android framework, Android API  
  
*Digest:*

...Such a poor test case coverage rate has led to many compatibility issues that can cause apps to crash at runtime on specific Android devices, resulting in poor user experiences for both apps and the Android ecosystem....

...Unfortunately, these approaches have only focused on detecting signature-induced compatibility issues (i.e., a certain API does not exist in certain Android versions), leaving other equally important types of compatibility issues unresolved....

...In this work, we propose a novel prototype tool, JUnitTestGen, to fill this gap by mining existing Android API usage to generate unit test cases....

...After locating Android API usage in given real-world Android apps, JUnitTestGen performs inter-procedural backward data-flow analysis to generate a minimal executable code snippet (i.e., test case)....

...Experimental results on thousands of real-world Android apps show that JUnitTestGen is effective in generating valid unit test cases for Android APIs....

...Because of the aforementioned advantages, it is recommended to always write unit tests when developing software, and the unit tests should cover as many units as possible....

...The Android framework provides thousands of public APIs that are heavily leveraged by app developers to facilitate their development of Android apps....

...Ideally, each such public API should be provided with a set of unit tests to ensure that the API is correctly implemented and the continuous evolution of the framework will not change its semantics....

...[21] demonstrate that various Android APIs suffer from compatibility issues as the evolution of the Android framework will regularly remove APIs from or add APIs into the framework....

...The authors further propose a prototype tool called PIVOT to automatically learn device-specific compatibility issues from existing Android apps....

...This could be achieved by providing unit tests for every API introduced to the framework and regressively testing the APIs against Android devices with different manufacturers and different framework versions....

...However, it is time-consuming to manually write and maintain unit tests for each Android API (which probably explains why there is only a small set of APIs covered by unit tests at the moment)....

...There is hence a need to automatically generate compatibility unit tests for Android APIs....

...In this work, we present a prototype tool, JUnitTestGen, that attempts to automatically generate test cases for Android APIs based on their practical usage in real-world apps....

...Experimental results on thousands of Android apps show that JUnitTestGen is effective in generating test cases for Android APIs....

...These test cases subsequently allow our approach to pinpoint various types of compatibility issues, outperforming a state-of-the-art generic test generation tool named EvoSuite, which can only generate test cases to reveal a small subset of compatibility issues....

Test Migration Between Mobile Apps

with Similar Functionality

...We believe that the cost of testing mobile apps can be reduced by considering similarities between apps and migrating test cases across similar apps....

...Based on our intuition, we defined APPTESTMIGRATOR (App Test Migrator), a technique for migrating test cases (i.e., test inputs and oracles) between apps that share part of their functionality....

...APPTESTMIGRATOR takes as input a source app, a test case for the source app (source test), and a target app, and produces as output the source test migrated to the target app (target test)....

*Keywords:*  
APP, TEST, APPTESTMIGRATOR, EVENT, test cases, ORACLE, TARGET, SIMILARITY, Test Migration, migrate oracles  
  
*Digest:*

...To reduce this effort, and thus the overall cost of mobile app testing, we propose APPTESTMIGRATOR, a technique for migrating test cases between apps in the same category (e.g., banking apps)....

...APPTESTMIGRATOR leverages these commonalities between user interfaces to migrate existing tests written for an app to another similar app....

...Specifically, given (1) a test case for an app (source app) and (2) a second app (target app), APPTESTMIGRATOR attempts to automatically transform the sequence of events and oracles in the test for the source app to events and oracles for the target app....

...Manually developing test cases for an app tends to be extremely expensive, as it involves human effort to define test cases and check test results....

...We believe that the cost of testing mobile apps can be reduced by considering similarities between apps and migrating test cases across similar apps....

...Based on our intuition, we defined APPTESTMIGRATOR (App Test Migrator), a technique for migrating test cases (i.e., test inputs and oracles) between apps that share part of their functionality....

...APPTESTMIGRATOR takes as input a source app, a test case for the source app (source test), and a target app, and produces as output the source test migrated to the target app (target test)....

...To address this issue, APPTESTMIGRATOR migrates each event independently from other events while crawling a target app....

...Third, APPTESTMIGRATOR improves the way in which GUITESTMIGRATOR computes the similarity between app entities and between events across apps....

...Fourth, unlike GUITESTMIGRATOR, APPTESTMIGRATOR can migrate oracles in addition to test inputs....

...Migrating oracles for app tests involves understanding how developers write such oracles in practice and defining a technique that can map them across apps accordingly....

...This is particularly challenging, as there are typically multiple ways to find a counterpart for an oracle in a target app, but it can make the approach considerably more useful in practice: tests that are successfully migrated together with their oracles can in fact also detect failures that do not result in crashes....

...We used both the test cases provided with the apps, when available, and additional test cases written by CS graduate students not involved in this research....

Test Reuse Based on Adaptive Semantic Matching across Android Mobile Applications

...To overcome this challenge, we propose TRASM (Test Reuse based on Adaptive Semantic Matching), a test reuse approach based on adaptive strategies to find a better event matching across android mobile applications....

...TRASM first performs GUI events deduplication on the initial test set obtained from test generation, and then employs an adaptive strategy to find better event matching, which enables reusing the existing test....

...B. Test Reuse Test reuse, as an alternative method to automatically generate GUI test, makes full use of existing resources to provide convenience for verifying the application's behavior....

...Based on semantic matching of events, TRASM considers the test (source test) of the existing application (source app), and the new application (target app) as inputs and outputs target test....

*Keywords:*  
TEST, APPLICATION, Test Reuse, GUI, EXIST, WIDGET, MATCHING, GENERATE, Semantic Matching, event matching  
  
*Digest:*

...Test reuse based on semantic similarities between applications of the same category has been utilized to reduce the manual effort of Graphical User Interface (GUI) testing....

...However, most of the existing studies fail to solve the semantic problem of event matching, which leads to the failure of test reuse....

...To overcome this challenge, we propose TRASM (Test Reuse based on Adaptive Semantic Matching), a test reuse approach based on adaptive strategies to find a better event matching across android mobile applications....

...TRASM first performs GUI events deduplication on the initial test set obtained from test generation, and then employs an adaptive strategy to find better event matching, which enables reusing the existing test....

...Keywords-adaptive semantic matching; android mobile applications; GUI event; test reuse; oracle generation I. INTRODUCTION Graphical User Interface (GUI) testing is commonly employed to verify and develop the behaviors of applications by designing and executing test cases of GUI applications [1]....

...Combining the selected characteristics, they design the semantic similarity calculation method between widgets to generate meaningful tests....

...They attempt to select widgets with high similarity in a similar application for matching each event of the existing test....

...This simple example explains that we need to solve the event redundancy issue in the process of test reuse caused by some particular functionality in the existing test....

...The test (b) is obtained by reusing the existing test (a)....

...TRASM can get more semantic matches in the generated test....

...B. Test Reuse Test reuse, as an alternative method to automatically generate GUI test, makes full use of existing resources to provide convenience for verifying the application's behavior....

...Considering that the success of test reuse heavily depends on the semantic matching of test events, there is still space for improvement by adopting appropriate strategies to increase the quality of reused tests from the perspective of application functionality....

...Trying to optimize the generated test sequence to ensure the quality of reused tests should be an optional strategy.III....

...Based on semantic matching of events, TRASM considers the test (source test) of the existing application (source app), and the new application (target app) as inputs and outputs target test....

...TRASM employs two significant phases to implement test reuse: preliminary preparation and source test reuse....

...For the former, the existing data is processed through test augmentation and model extraction to facilitate the implementation of source test reuse....

...A. Test augmentation The main task of test augmentation is to extract semantic information of widgets during the execution of collected source tests....

...The semantically represented widgets, together with actions, compose augmented tests, which are used to match widgets in the GUI of the target app....

Test Transfer Across Mobile Apps

Through Semantic Mapping

*Keywords:*  
APP, TRANSFER, CRAFTDROID, INPUT, REGISTRATION, SHOP, TECHNIQUE, GUI, ORACLE, test input, test oracles  
  
*Digest:*

...To address these limitations, we propose CRAFTDROID, a framework that leverages information retrieval, along with static and dynamic analysis techniques, to extract the human knowledge from an existing test suite for one app and transfer the test cases and oracles to be used for testing other apps with the similar functionalities....

...This paper makes the following contributions: : A novel technique for transferring both test inputs and oracles across mobile apps through semantic mapping of actionable GUI widgets....

...Despite these challenges, CRAFTDROID is able to transfer a test case that verifies the registration process in Rainbow Shops to Yelp by semantically mapping their GUI widgets....

*eywords:*  
APP, TRANSFER, CRAFTDROID, INPUT, REGISTRATION, SHOP, TECHNIQUE, GUI, ORACLE, test input, test oracles  
  
*Digest:*

...Despite the numerous GUI-based test input generation techniques proposed in the literature, these techniques are still limited by (1) lack of context-aware text inputs; (2) failing to generate expressive tests; and (3) absence of test oracles....

...To address these limitations, we propose CRAFTDROID, a framework that leverages information retrieval, along with static and dynamic analysis techniques, to extract the human knowledge from an existing test suite for one app and transfer the test cases and oracles to be used for testing other apps with the similar functionalities....

...Index Terms-Test transfer, test migration, GUI testing, natural language processing, semantic similarity I. INTRODUCTION GUI testing is the primary way of examining the functionality and usability of mobile apps....

...Most of the stateof-the-art techniques use random input values or rely on the manual configurations for text inputs....

...Without such meaningful inputs, exploration of the App Under Test (AUT) may get stuck at the very beginning and GUI states deep in the testing flow may never been exercised....

...Without automated test oracles, such tests cannot thoroughly verify correct behavior of the AUT....

...To address these limitations, we propose CRAFTDROID, a framework to reuse an existing test suite for one app to test other similar apps....

...Like their works, our proposed technique transfers available test cases corresponding to a specific feature or use-case scenario of one app to other apps with similar functionality....

...However, unlike their work, CRAFTDROID is also able to transfer the test oracles, if they exist....

...To enable context-awareness for text inputs, CRAFTDROID relies on information retrieval techniques to extract the human knowledge from an existing test suite and reuse it for other apps....

...As CRAFTDROID not only transfers test inputs, but also oracles (assertions), it is able to thoroughly verify correct behavior of the AUT....

...For example, shopping apps should implement user registration and authentication to provide personalized services....

..."Place Order" button on another shopping app....

...This paper makes the following contributions: : A novel technique for transferring both test inputs and oracles across mobile apps through semantic mapping of actionable GUI widgets....

...While the overall registration process in these two apps follows the same steps-clicking on a button to start registration, filling the registration form, and submitting information-a direct copy of the test steps from Rainbow Shops to Yelp is not possible due to the following reasons: (1) The mapping of test steps between the two apps is not one-to-one....

...Despite these challenges, CRAFTDROID is able to transfer a test case that verifies the registration process in Rainbow Shops to Yelp by semantically mapping their GUI widgets....

UI Test Migration Across Mobile Platforms

..Specifically, these approaches leverage existing usage-based tests from a source Android app to automatically generate equivalent tests for a target Android app....

...A largely unexplored variant of this problem is transferring tests written for an app implemented on one platform (e.g., Android), to the same app implemented for another platform (e.g., iOS)....

...Source platform and target platform are the platforms on which the source and target apps run, respectively....

*Keywords:*  
TEST, APP, TRANSFER, PLATFORM, EVENT, IO, SOURCE, MAPIT, APPROACH, Android and iOS  
  
*Digest:*

...Several approaches have tried to address this problem in mobile apps: by exploiting the similarities of different apps within the same domain on a single platform, they have shown that it is possible to transfer tests that exercise similar functionality between the apps....

...A related recent technique enables transfer of UI tests uni-directionally, from an open-source iOS app to the same app implemented for Android....

...MAPIT runs existing tests on a "source" app and builds a partial model of the app corresponding to each test....

...Our evaluation on a diverse set of widely used, closed-source sibling Android and iOS apps shows that MAPIT is feasible, accurate, and useful in transferring UI tests across platforms....

...Specifically, these approaches leverage existing usage-based tests from a source Android app to automatically generate equivalent tests for a target Android app....

...A largely unexplored variant of this problem is transferring tests written for an app implemented on one platform (e.g., Android), to the same app implemented for another platform (e.g., iOS)....

...Cross-platform transfer has unique challenges as compared to test transfer within Android alone....

...Second, iOS is a closed-source platform, which has led to fewer and more limited tools for analyzing iOS apps compared to Android....

...First, it assumes the availability of both the Android and iOS apps' source code, which, as mentioned, is especially unlikely for iOS apps....

...Second, TestMig only covers transferring tests in one direction (iOS to Android)....

...To address these limitations, we have developed MAPIT, a novel approach for bi-directional transfer of usage-based tests across different mobile platforms, with no source code required on either platform....

...MAPIT is also the first approach capable of transferring oracle events and system events across mobile platforms....

...First, MAPIT dynamically extracts a GUI model of the app on the source platform while executing the source test....

...Second, based on this GUI model, the source test is migrated to the target platform by mapping the GUI widgets from the source app to the most similar widgets in the sibling app....

...The source app is the app with existing tests that are to be transferred to the target app....

...Source platform and target platform are the platforms on which the source and target apps run, respectively....

**Rico: A Mobile App Dataset**

**for Building Data-Driven Design Applications**

...This paper presents Rico, the largest repository of mobile app designs to date, created to support five classes of data-driven applications: design search, UI layout generation, UI code generation, user interaction modeling, and user perception prediction....

...Rico also exposes an additional view of each app's design data: while ERICA provides a collection of individual user interaction traces for an app, Rico additionally provides a list of the unique UIs discovered by aggregating over user interaction traces and merging UIs based on a similarity measure....

...DATA-DRIVEN DESIGN APPLICATIONS Rico was built to support a variety of data-driven applications for mobile app design....

*Keywords:*  
RICO, DATASET, UIS, user interaction, SEARCH, Mobile App, app designs, design search, App Dataset, design datasets  
  
*Digest:*

...This paper presents Rico, the largest repository of mobile app designs to date, created to support five classes of data-driven applications: design search, UI layout generation, UI code generation, user interaction modeling, and user perception prediction....

...To create Rico, we built a system that combines crowdsourcing and automation to scalably mine design and interaction data from Android apps at runtime....

...To demonstrate the kinds of applications that Rico enables, we present results from training an autoencoder for UI layout similarity, which supports queryby-example search over UIs....

...ACM Classification Keywords D.2.2 Software Engineering: Design Tools and Techniques Author Keywords Mobile app design; design mining; design search; app datasets INTRODUCTION Data-driven models of design can scaffold the creation of mobile apps....

...To build these models, researchers require design datasets which expose the details of mobile app designs at scale....

...These properties can be combined in different ways to support five classes of datadriven applications: design search, UI layout generation, UI code generation, user interaction modeling, and user perception prediction....

...Design Mobile Layout UI User Contents Year # Apps # UIs Mining View Hierarchies Screenshots User Interactions Rico 2017 9.7K 72.2K Dynamic ERICA 2016 2.4K 18.6K Dynamic Shirazi et al....

...When crowd workers use an app, the system records a user interaction trace that captures the UIs visited and the interactions performed on them....

...For each app, Rico presents a collection of individual user interaction traces, as well as a collection of unique UIs determined by a novel content-agnostic similarity heuristic....

...Additionally, since the Rico dataset is large enough to support deep-learning applications, each UI is annotated with a low-dimensional vector produced by training an autoencoder for UI layout similarity, which can be used to cluster and retrieve similar UIs from different apps....

...Mobile app designs comprise several different components, including user interaction flows (e.g., search, login), UI layouts, visual styles, and motion details....

...Similarly, combining user interaction details with view hierarchies and screenshots can help identify the user flows that apps are designed to support....

...Like the ERICA dataset, Rico is created by mining design and interaction data from apps at runtime....

...Rico also exposes an additional view of each app's design data: while ERICA provides a collection of individual user interaction traces for an app, Rico additionally provides a list of the unique UIs discovered by aggregating over user interaction traces and merging UIs based on a similarity measure....

...DATA-DRIVEN DESIGN APPLICATIONS Rico was built to support a variety of data-driven applications for mobile app design....

...To support keyword search over mobile app designs, Rico exposes app-level metadata from the Google Play Store and element-level metadata contained within the Android view hierarchies....

**Learning Design Semantics for Mobile Apps**

...Similarly, we analyze text buttons and icons to determine functional semantics: UX concepts that describe how elements and screens are used....

...Text Buttons Based on the text button examples we identified while categorizing UI components, we developed a heuristic for extracting text buttons from Android view hierarchies: any element whose class or ancestors property contains the string "button." Using this criteria, we extracted 130,761 buttons from the Rico dataset, and found 20,386 unique button text strings....

...To mine UX concepts from these buttons, we filtered out button text comprising a single character or appearing in fewer than five apps, and clustered text strings that have substrings in common such as "retry" and "try again." We determined a UX concept (e.g., create, log in) for each cluster based on the most frequently occurring words in the set....

*Keywords:*  
BUTTON, COMPONENT, ICON, CONCEPT, APP, DESIGN, text button, SCREEN, icon classes, elements on the screen  
  
*Digest:*

...Through an iterative open coding of 73k UI elements and 720 screens, we contribute a lexical database of 25 types of UI components, 197 text button concepts, and 135 icon classes shared across apps....

...Given a UI's screenshot and view hierarchy, we automatically identify 25 UI component categories, 197 text button concepts, and 99 icon classes....

...These annotations identify both the structural roles (e.g., image content, bottom navigation) and the functional ones (e.g., login button, share icon) that elements play in the UI's design....

...To develop this approach, we first generated a lexical database of UI components and UX concepts (i.e., text buttons and icons) that are shared across apps through an iterative open coding of 73k UI elements and 720 screens....

...Then, we leveraged this database to learn code-based patterns to detect different components, and trained a convolutional neural network (CNN) to distinguish between icon classes....

...To bootstrap this lexical database, we referenced popular design libraries and app-prototyping tools to create a vocabulary of UI components and UX concepts....

...Through this process, we identified 25 types of UI components, 197 text button concepts, and 135 icon classes....

...The resultant database also exposes icon and text-button synonym sets related to each UX concept, creating links between the visual and textual elements used in digital design....

...In this paper, we categorize UI components to provide semantics that describe the structural roles (e.g., image content, bottom navigation) elements play on a screen....

...Similarly, we analyze text buttons and icons to determine functional semantics: UX concepts that describe how elements and screens are used....

...Three researchers from our team independently coded the screen's elements, noting any component types that were not part of the initial vocabulary....

...UX Concepts To construct a set of functional semantic concepts, we further analyzed TEXT BUTTON and ICON elements, since they indicate possible interactions on a screen (e.g., login button, search icon) [3]....

...We mine UX concepts directly from the text buttons by extracting and clustering the text contained within them....

...Text Buttons Based on the text button examples we identified while categorizing UI components, we developed a heuristic for extracting text buttons from Android view hierarchies: any element whose class or ancestors property contains the string "button." Using this criteria, we extracted 130,761 buttons from the Rico dataset, and found 20,386 unique button text strings....

...To mine UX concepts from these buttons, we filtered out button text comprising a single character or appearing in fewer than five apps, and clustered text strings that have substrings in common such as "retry" and "try again." We determined a UX concept (e.g., create, log in) for each cluster based on the most frequently occurring words in the set....

**Unblind Your Apps: Predicting Natural-Language Labels for**

**Mobile GUI Components by Deep Learning**

...To overcome these challenges, we develop a deeplearning based model, called LabelDroid, to automatically predict the labels of image-based buttons by learning from large-scale commercial apps in Google Play....

...KEYWORDS Accessibility, neural networks, user interface, image-based buttons, content description ACM Reference Format: Jieshan Chen, Chunyang Chen, Zhenchang Xing, Xiwei Xu, Liming Zhu, Guoqiang Li, and JinshuiWang....

...Table 2 shows that, there are 15,595 imagebased buttons from 6,175 apps as the training set, 1,759 buttons from 714 apps as validation set and 1,879 buttons from 705 apps as testing set....

*Keywords:*  
APP, LABEL, TRAINING, METRIC, DESCRIPTION, USER, MODEL, UNIVERSITY, BUTTON, Evaluation Metric  
  
*Digest:*

...Unblind Your Apps: Predicting Natural-Language Labels for Mobile GUI Components by Deep Learning Jieshan Chen Jieshan.Chen@anu.edu.au Australian National University Australia Chunyang Chen∗ Chunyang.Chen@monash.edu Monash University Australia Zhenchang Xing: Zhenchang.Xing@anu.edu.au Australian National University Australia Xiwei Xu Xiwei.Xu@data61.csiro.au Data61, CSIRO Australia Liming Zhu:: Liming.Zhu@data61.csiro.au Australian National University Australia Guoqiang Li∗ Li.G@sjtu.edu.cn Shanghai Jiao Tong University China Jinshui Wang∗ ymkscom@gmail.com Fujian University of Technology China ABSTRACT According to the World Health Organization(WHO), it is estimated that approximately 1.3 billion people live with some forms of vision impairment globally, of whom 36 million are blind....

...Users with vision impairment can adopt the screen reader embedded in the mobile operating systems to read the content of each screen within the app, and use gestures to interact with the phone....

...However, the prerequisite of using screen readers is that developers have to add natural-language labels to the image-based components when they are developing the app....

...Unfortunately, more than 77% apps have issues of missing labels, according to our analysis of 10,408 Android apps....

...And even if developers want to add the labels to UI components, they may not come up with concise and clear description as most of them are of no visual issues....

...To overcome these challenges, we develop a deeplearning based model, called LabelDroid, to automatically predict the labels of image-based buttons by learning from large-scale commercial apps in Google Play....

...:Also with University of New South Wales....

...KEYWORDS Accessibility, neural networks, user interface, image-based buttons, content description ACM Reference Format: Jieshan Chen, Chunyang Chen, Zhenchang Xing, Xiwei Xu, Liming Zhu, Guoqiang Li, and JinshuiWang....

...Compared with the normal users, they may be more eager to use the mobile apps to enrich their lives, as they need those apps to 322 2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE) 323 324 325 326 327 Unblind Your Apps: Predicting Natural-Language Labels for Mobile GUI Components by Deep Learning ICSE '20, May 23-29, 2020, Seoul, Republic of Korea Table 2: Details of our accessibility dataset....

...To enable the mini-batch training, we need to add a < > token to pad the word sequence of labels into a fixed length....

...For each app category, we randomly select 80% apps for training, 10% for validation and the rest 10% for testing....

...Table 2 shows that, there are 15,595 imagebased buttons from 6,175 apps as the training set, 1,759 buttons from 714 apps as validation set and 1,879 buttons from 705 apps as testing set....

...We use Adam optimizer [51] with 1 = 0.9, 2 = 0.98 and = 10−9 and change the learning rate according to the formula \_ = −0.5 × ( \_ −0.5, \_ × \_ −1.5) to train the model, where \_ is the current iteration number of training batch and the first \_ training step is used to accelerate training process by increasing the learning rate at the early stage of training....

...The first metric we use is exact match rate, i.e., the percentage of testing pairs whose predicted content description exactly matches the ground truth....

...BLEU is an automatic evaluation metric widely used in machine translation studies....

...METEOR [26] (Metric for Evaluation of Translation with Explicit ORdering) is another metric used for machine translation evaluation....

AppFlow: Using Machine Learning to Synthesize Robust,

Reusable UI Tests

...It enables developers to write a library of modular tests for the main functionality of an app category (e.g., an "add to cart" test for shopping apps)....

...Thus, a test script for an app often cannot locate the right screens and widgets for another app....

...It enables developers - e.g., those in the "shopping app" community or a testing services company - to write a library of modular UI tests for the main functionality of a given category of apps....

*Keywords:*  
TEST, APP, TESTING, APPFLOW, SCREEN, DEVELOPER, app category, SHOP, screens and widgets, test scripts  
  
*Digest:*

...It leverages machine learning to automatically recognize common screens and widgets, relieving developers from writing ad hoc, fragile logic to use them in tests....

...It enables developers to write a library of modular tests for the main functionality of an app category (e.g., an "add to cart" test for shopping apps)....

...It can then quickly test a new app in the same category by synthesizing full tests from the modular ones in the library....

...Optionally, developers can customize AppFlow by adding app-specific tests for completeness....

...We evaluated AppFlow on 60 popular apps in the shopping and the news category, two case studies on the BBC news app and the JackThreads shopping app, and a user-study of 15 subjects on the Wish shopping app....

...CCS CONCEPTS : Software and its engineering → Software testing and debugging; Empirical software validation; Software evolution; KEYWORDS mobile testing; test reuse; test synthesis; UI testing; machine learning; UI recognition ACM Reference Format: Gang Hu, Linjie Zhu, and Junfeng Yang....

...Thus, a test script for an app often cannot locate the right screens and widgets for another app....

...For instance, the sign-in flow of an app may contain just the sign-in screen, but another app may show a welcome screen first....

...These subtle differences prevent directly reusing test scripts on different apps....

...It enables developers - e.g., those in the "shopping app" community or a testing services company - to write a library of modular UI tests for the main functionality of a given category of apps....

...Then, when developers want to test a new app in the same category, they can quickly synthesize full tests from the modular ones in the library with a few lines of customization, greatly boosting productivity....

...By focusing on the main functionality of an app category, AppFlow provides "smoke tests" or build verification testing for each source code change, requiring little or no manual work....

...Optionally, developers can customize AppFlow to add app-specific tests or override defaults to perform complete regression testing....

...The training dataset can come from a developer community for an app category, and AppFlow provides several utilities to simplify this mostly one-time data collection....

...This machine learning approach enables the AppFlow tests to refer to canonical screens and widgets instead of app-specific ones, enjoying a variety of benefits....

...Third, canonical screens and widgets abstract app-specific variations, making it easy to share tests across apps....

IconIntent: Automatic Identification of Sensitive UI Widgets based on Icon

Classification for Android Apps

...However, many UI widgets leverage icons of specific shapes (object icons) and icons embedded with text (text icons) to express their intentions, posing challenges for existing detection techniques that analyze only textual data to identify sensitive UI widgets....

...ICONINTENT automatically associates UI widgets and icons via static analysis on app's UI layout files and code, and then adapts computer vision techniques to classify the associated icons into eight categories of sensitive data....

...Many apps today include UI widgets such as buttons and text boxes, which expect users' consensus to use their sensitive data (e.g., pressing a button), or users' input of sensitive data (e.g., filling financial information in a text box)....

*Keywords:*  
ICON, WIDGET, APP, ICONINTENT, TEXT, TECHNIQUE, Sensitive UI Widgets, sensitive data, PERMISSION, IDENTIFY  
  
*Digest:*

...Thus, to identify suspicious sensitive data usage such as UI-permission mismatch, it is crucial to understand the intentions of UI widgets....

...However, many UI widgets leverage icons of specific shapes (object icons) and icons embedded with text (text icons) to express their intentions, posing challenges for existing detection techniques that analyze only textual data to identify sensitive UI widgets....

...In this work, we propose a novel app analysis framework, ICONINTENT, that synergistically combines program analysis and icon classification to identify sensitive UI widgets in Android apps....

...ICONINTENT automatically associates UI widgets and icons via static analysis on app's UI layout files and code, and then adapts computer vision techniques to classify the associated icons into eight categories of sensitive data....

...But this line of works are limited because they cannot address sensitive user inputs, where apps express their intentions to use or collect users' sensitive data....

...Many apps today include UI widgets such as buttons and text boxes, which expect users' consensus to use their sensitive data (e.g., pressing a button), or users' input of sensitive data (e.g., filling financial information in a text box)....

...For example, given an app that (a) (b) (c) Figure 1: UIs containing icons that indicate the uses of sensitive data in mobile apps requests a permission (e.g., microphone), an inspection of the app's UIs can determine that the permission is suspicious if this permission cannot be justified by the text and /or icons on any UI widget....

...Second, some icons are embedded with text, referred to as text icons....

...This inspires us to develop object icon classification techniques to detect similar icons based on the sensitive icons collected from interactive widgets....

...This inspires us to develop icon mutation techniques to amplify and normalize this contrast, making icons easier to be recognized by the icon classification techniques....

...The icon-widget association module provides a UI layout analysis technique to identify the associations between icons and UI widgets defined in the UI layout files....

...The icon mutation analysis module extracts icons from an app, and produces mutated icons for each of the extracted icon....

...Additionally, this module adapts OCR techniques to extract text from the icons, and then classifies the icons using the edit-distance based similarity between the extracted text and the keywords in each category....

...Also, we evaluate the effectiveness in reducing the inspection effort of suspicious permissions: if an identified intention of a UI widget matches a requested permission, then the permission is considered not suspicious....

...Example UI with a Sensitive Icon....

1. Start
2. Get the first definition of the elements
3. If the algorithm is ATM: 4. Get label𝑠1 and label𝑠2
   1. For each element in the target set: 6. Get label𝑡1 and label𝑡2
      1. Calculate the similarity score between label𝑠 and label𝑡 using the "sum" method
      2. Add the similarity score to the scores list
   2. Set the score for the current element in descScores to the maximum score in the scores list
   3. Go to the next element in the source set
4. If the algorithm is CraftDroid:
5. For each element in the target set: 13. Calculate the similarity score between each attribute in the source and target elements using the "avg" method
   1. Add the similarity score to the scores list
6. Set the score for the current element in descScores to the average score in the scores list
7. Go to the next element in the source set
8. If the algorithm is SemFinder: 18. For each element in the target set: 19. Collect the text attributes from the source and target elements 20. Calculate the similarity score between the collected text using the "avg" method 21. Set the score for the current element in descScores to the calculated similarity score 22. Go to the next element in the source set
9. Merge the three descScores arrays
10. Sort the elements in the target set by their similarity score in descScores
11. Return the sorted elements
12. End