

《自动化测试 2022》大作业 & 综述要求

— 自动修复、随机测试、测试排序方向

助教：张犬俊 quanjun.zhang@smail.nju.edu.cn

有任何不明确或者进一步探讨，欢迎邮件联系。

1. 工具选题一：基于迁移学习的人工智能框架缺陷技术探索

近年来，随着人工智能技术在自动驾驶、图像识别等领域不断产生突破性成果，研究人员逐步加强对智能软件安全的重视。为了更好地实现人工智能任务、推动人工智能的发展，近年来涌现出越来越多的用于构建人工智能模型的框架和平台，例如 TensorFlow、PyTorch 和 Keras 等。开发者可以使用人工智能框架提供的编程接口，轻松地设计、训练以及测试深度学习模型。然而人工智能框架作为整个人工智能模型构建的基础架构显得至关重要，**框架中的缺陷可以影响人工智能模型预测结果的准确性，甚至会产生严重的后果**。因此如何快速的修复人工智能框架中发现的缺陷显得至关重要。**尽管已经存在一些较为成熟的基于深度学习的程序缺陷修复技术，但是人工智能框架由于其特殊性使得通用的修复技术无法直接应用**，首先（1）人工智能框架**数据量较少**，而通用缺陷修复技术往往需要大量语料训练成熟的修复模型；同时（2）人工智能框架往往**涉及多个语言**，然而现有修复技术往往针对单个编程语言。

本项目拟设计一种基于迁移学习的人工智能框架缺陷修复技术，通过迁移学习技术学习通用缺陷语料中的修复模型，然后在框架数据集中进行微调，同时使用提示学习技术使得修复模型能够识别不同语言之间的修复模式。

(1) 通用缺陷数据集和框架缺陷数据集的提取

本项目已提供相应的通用缺陷数据集和框架缺陷数据集，请自行下载

<https://box.nju.edu.cn/d/31372e01ac624d798700/>

(2) 通用修复模型的设计与生成

使用 transformer 或者预训练模型（如 CodeT5）**设计一种通用缺陷修复模型**。

(3) 框架修复模型的微调

在步骤（2）训练好的修复模型基础上，进一步**使用框架缺陷数据集微调**，并使用**提示学习技术**指定相应编程语言，以此设计一种**框架特定的修复模型**。

(4) 修复效果的评估

使用上述步骤完成的框架修复模型对框架缺陷数据进行测试，**输出完全修复正确的缺陷比例**。

参考文献：

- [1] CIRCLE: Continual Repair across Programming Languages.

<https://arxiv.org/abs/2205.10956>

通用缺陷修复技术文献

- [2] CodeT5: Identifier-aware Unified Pre-trained Encoder-Decoder Models for Code Understanding and Generation

<https://arxiv.org/abs/2109.00859>

预训练模型技术参考文献

- [3] 2022-TSE-Neural Transfer Learning for Repairing Security Vulnerabilities in C Code

<https://arxiv.org/pdf/2104.08308.pdf>

迁移学习技术参考文献

- [4] Zero-Shot Program Representation Learning

<https://arxiv.org/abs/2204.08360>

提示学习技术参考文献

2. 工具选题二：面向数值程序的自适应随机测试复现技术探索

自适应随机测试作为一种有效地测试用例生成方式，被广泛地用于许多领域，如模糊测试以及 AI 测试等。在众多功能测试方法中，基础的测试方法是随机测试。它从输入域中随机挑选测试用例，直到达到某个条件，比如发现软件系统中的错误。然而，由于随机测试很少利用甚至没有利用额外的信息，其测试效果是有限的。因此，自适应随机测试被提出，在保证测试用例随机性的同时，使测试用例能够均匀地分散在整个输入域中。自适应随机测试作为一种能够提高随机测试有效性的手段，近些年来受到学术界和工业界的广泛关注和研究，但是较少工作对自适应随机测试进行系统性评估。其中最经典的算法是固定候选集大小的自适应随机测试算法（Fixed-Size-Candidate-Set ART, FSCS-ART）。FSCS-ART 算法利用了两个测试用例集合的优势，即已执行测试用例集 E 和候选测试用例集 C 。每一轮从候选测试用例集 C 中选取的一个到已执行测试用例集 E 距离最远的候选测试用例将作为下一个测试用例执行，并将其加入到已执行测试用例集 E 中。

本项目针对数值程序领域，旨在复现现有的经典的自适应随机测试算法，并对其进行有效性比较。

初步项目框架地址 <https://box.nju.edu.cn/d/9288e9a87a5c4c3c961e/>。

(1) 选择待复现的经典的自适应随机测试算法

根据现有文献选择不少于 15 种基于数值程序的自适应随机测试算法，已初步提供相应算法，也可自行添加，可根据包含的算法数量进行加分。

(2) 自适应随机测试算法框架搭建

阅读各个算法原始论文进行算法实现，各个算法实现使用统一数据结构，并要求使用规范注释说明各个函数。使用统一的接口搭建高耦合的测试框架，初步框架已提供，可借鉴或自行搭建。根据搭建框架的代码和注释质量评分。

(3) 自适应随机测试算法有效性对比评估

计算各个算法检测到数值程序中缺陷所需的用例个数以及时间开销，并对其进行比较。已初步提供部分待测数值程序，也可以自己添加（可加分）。

参考文献

- [1] A Survey on Adaptive Random Testing

<https://arxiv.org/pdf/2007.03885.pdf>

自适应随机测试综述

- [2] Adaptive Random Testing: The ART of test case diversity

<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.150.4003>

自适应随机测试介绍

- [3] Adaptive Random Testing: An Illusion of Effectiveness

<https://www.simula.no/sites/default/files/publications/Simula.approve.48.pdf>

自适应随机测试算法比较实证

3. 综述选题：基于机器学习的测试用例排序综述

回归测试对现在软件系统的质量保证至关重要，但是在实践中往往代价高昂。测试用例优先级排序技术旨在通过对回归测试用例集进行重新排序，从而尽可能早地发现缺陷。该技术可以有效地减少回归测试的开销，因此近些年来被研究人员广泛的研究。一个典型场景式对 Java 程序中的 Junit 测试用例进行排序，首先获得各个测试用例的代码覆盖率，之后依据代码覆盖率的带下对测试用例进行排序，将更加重要的测试用例放在序列前面优先执行。本项目需针对测试用例排序与机器学习交叉领域进行整理，即考虑（1）使用机器学习技术进行传统的测试用例排序、以及（2）测试用例排序技术在机器学习任务中的应用。要求在这两个小方向整理相应工作，并完成一篇格式正确、内容丰富、条理清晰的中文综述。从排序技术、机器学习技术、排序准则、评估准则、被测集以及应用场景等方面介绍。

参考文献：

- [1] 回归测试中的测试用例优先排序技术述评

<http://www.jos.org.cn/jos/article/abstract/4420>

测试用例排序技术中文综述，可参考格式和框架，严禁直接拷贝

- [2] A Survey on Regression Test-Case Prioritization

<https://yilinglou.github.io/papers/RTPSur.pdf>

测试用例排序最新综述

- [3] A Taxonomy of Information Attributes for Test Case Prioritisation: Applicability, Machine Learning

<https://arxiv.org/abs/2201.06044>

测试用例排序与机器学习结合综述

- [4] Reinforcement Learning for Test Case Prioritization

<https://arxiv.org/abs/2011.01834>

强化学习在测试排序的应用

- [5] Optimizing Test Prioritization via Test Distribution Analysis

<https://yilinglou.github.io/papers/FSE18.pdf>

机器学习算法在经典排序算法中的应用