PNSQC

OCTOBER 10-12 2022

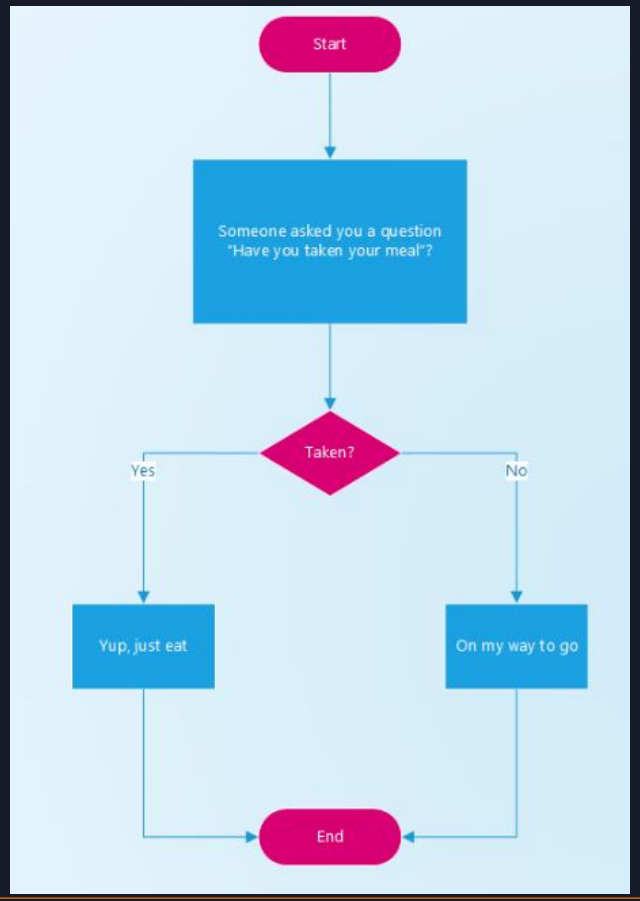
WILLAM LOO

Test Case Prioritization Using a Deep Learning Hybrid Approach

Author Self Introduction

- Intel Employee who takes master course in University Malaysia Pahang (UMP)
- ISQTB Certified Tester in year 2012.
- Validation experience after my graduation are nearly 9 years
- Local Penangite who loves to travel abroad across multiple countries
- Loves to get in touch environment includes sea and jungle
- The motto I believe “The God we trust, the rest we validate”

Warm Up



WILLAM LOO

Test Case Prioritization Using a Deep Learning Hybrid Approach

40TH ANNUAL PACIFIC NW SOFTWARE QUALITY CONFERENCE
OCT. 10–12, 2022

Introduction

- Exhaustive testing is impossible for complex use case
- Used up a lot of duration to validate all combinations
- Objective:
  - Acquire suitable deep learning algorithms
  - Evaluate performance of developed algorithm in software validation process
- Project Scope:
  - In Scope: Will develop a deep learning algorithm combined with Greedy approach
  - Out Scope: Will not develop automation framework and not focus on standalone project

Literature Review Overview

- Test Case Prioritization
- Test Coverage
- Deep Learning in Software Test Validation

Test Case Prioritization

- Yanshan mentioned that behavior pattern of the test will influence the prioritization of test case in the group of the test cases in the pool.
- Aizaz had discussed that previous researchers had coded one test case prioritization using automatic history-based approach
- In Reinforcement Learning for Test Case Prioritization paper, we are clearly understanding that test prioritization depends on number of cycles, feature records, and the optimal ranking

Test Coverage

- Maximizing the requirement coverage [4]
- Applying Pairwise testing or t-way testing [5]

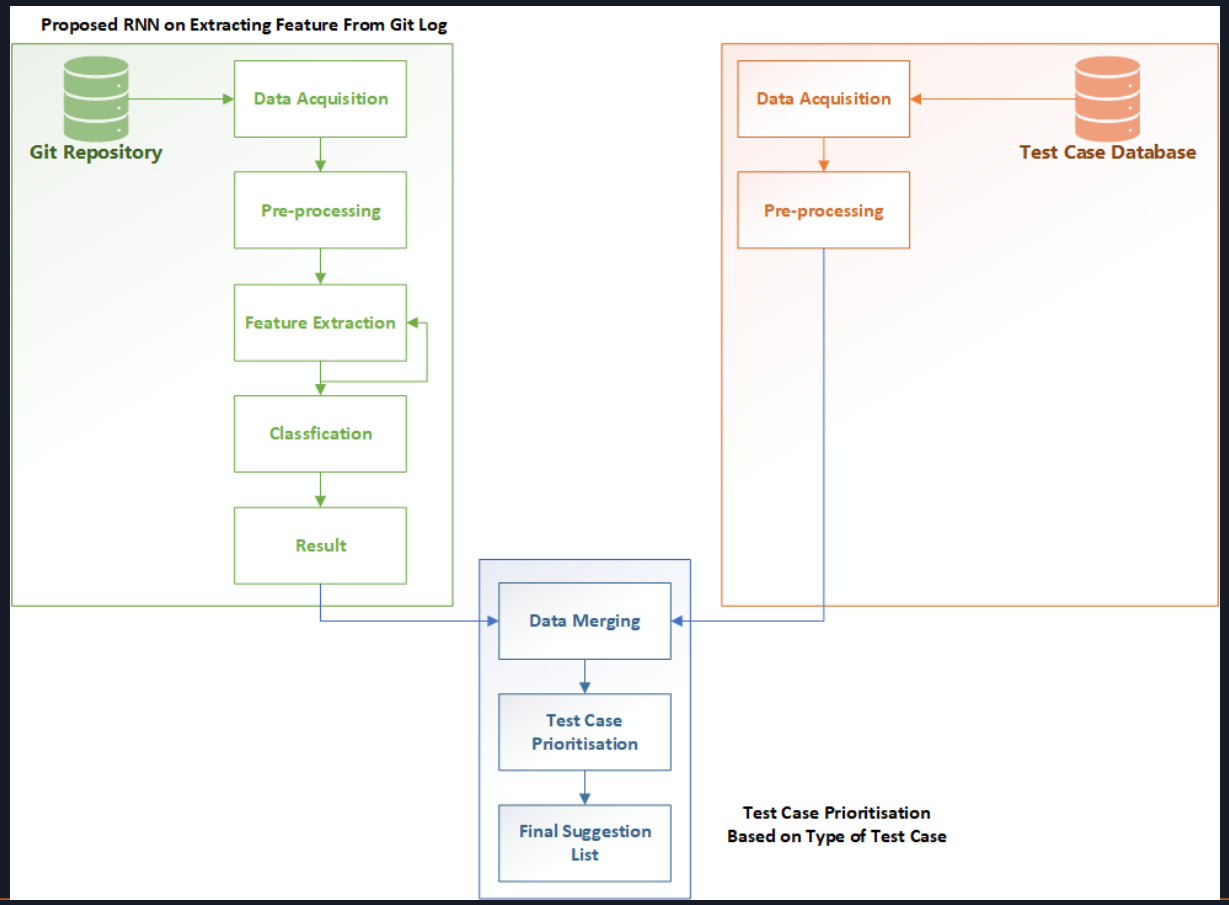
Deep Learning in Software Test Validation

- Eran proposed the QUADRANT approach.
- From Kai's findings, they are using deep neural network (DNN) as the tool performing the test case prioritization

Proposed Methodology Overview

- Data Acquisition Phase
- Data Pre-processing Phase
- Feature Extraction and Classification Phase
- Data Merging Phase
- Test Case Prioritization Phase

Proposed Methodology Diagram Overview



WILLAM LOO

Test Case Prioritization Using a Deep Learning Hybrid Approach

40TH ANNUAL PACIFIC NW SOFTWARE QUALITY CONFERENCE
OCT. 10–12, 2022

Git Log Text Preprocessing

- Data Acquisition Phase
  - Clone a branch from a repository. Inside the git folder, we had collected the log and exported it into the text file
- Data Pre-processing Phase
  - Perform the meta data removal, like commit hash, author name and email, header tag and branch name as well as the date of the commit.

Git Commit Categorization

- Feature Extraction and Classification Phase
  - RNN formula formulated as follows where h_t represent the current state at period t, fw represents the function with the parameter w, h_{t-1} shows the previous state of the RNN, and x_t represents the input vector at t timestamp.

$$h_t = fw(h_{t-1}, x_t)$$

Test Case Population

- Data Merging Phase
  - In this phase, the outcome of the predicted git commit category will be the keyword to populate the associated test cases from the database or list.
  - Equation below shows the notation on the method that we used to filter out the test cases pool.

$$FTC_i = TCP_i \cap C_i$$

  - In this equation, FTC represents filtered test cases and TCP represents test case pool, C represents category label and i represents iteration in the for loop

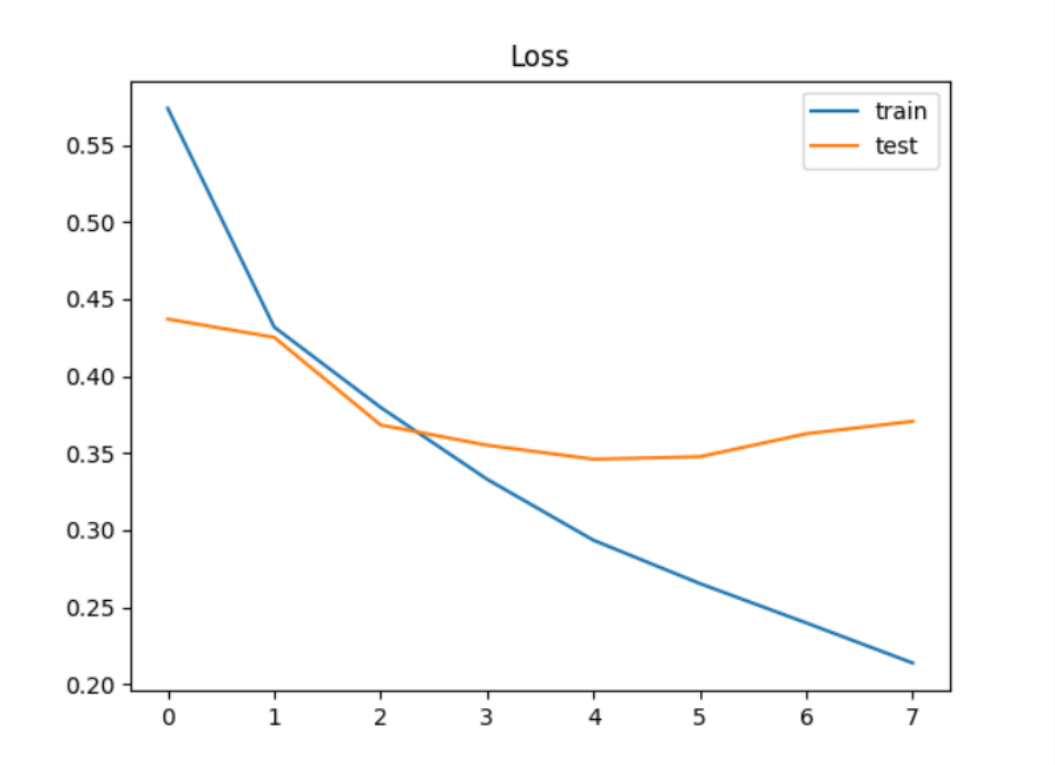
Test Case Prioritization

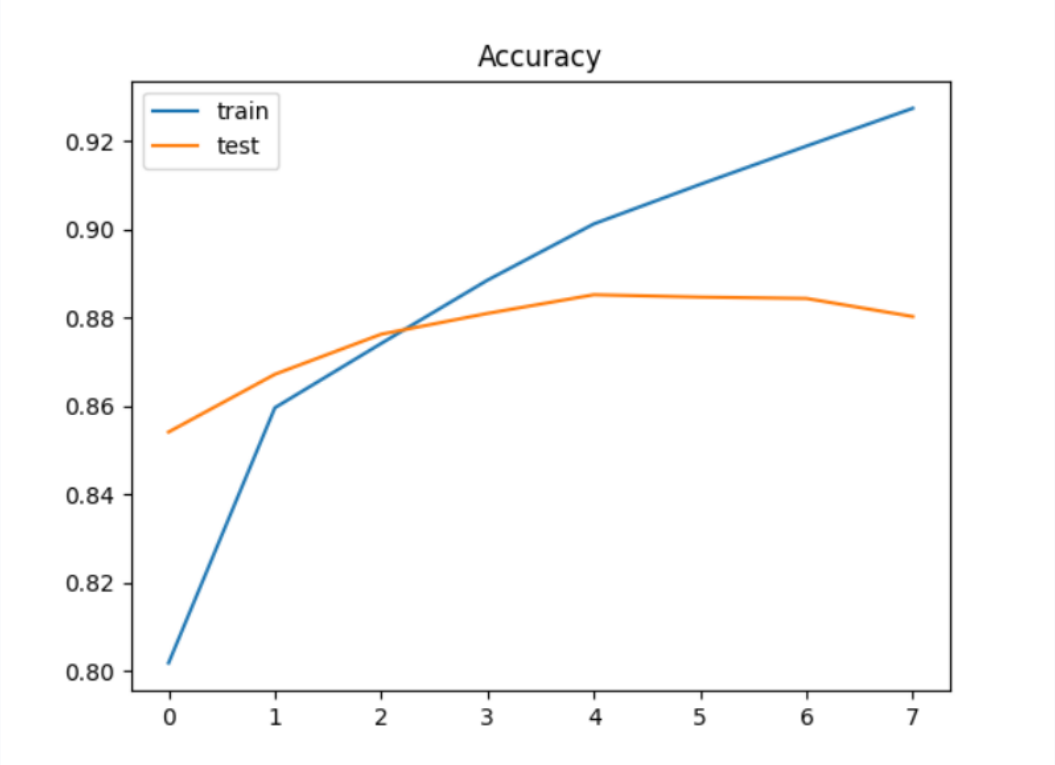
- Test Case Prioritization Phase
  - Will implement the test case reduction method, which is Greedy algorithm to remove the redundancy of the test cases and prioritize the test cases by maximize the coverage of the requirement.

$$TC_i = \sum_{i=0}^{n} \left(\sum_{j=0}^{p} REQ_{(i,j)} \geq 1 \right) \leq \frac{1}{2} n$$

  - From this equation, TC shows that the test cases, given that $TC = \{TC_0, TC_1, \dots, TC_{n-2}, TC_{n-1}\}$, REQ represent the requirement row, given that $REQ = \{REQ_0, REQ_1, \dots, REQ_{p-2}, REQ_{p-1}\}$, i and j are these coordinates of the test cases versus to the requirements and n and p are the total row and column numbers in the test requirement matrix.

Result – Text Analysis Part





Result – Test Case Prioritization Part

Requirement Test Matrix (RTM) Number	RTM #1	RTM #2
Original Test Case Number	30	51
Reduced Test Case Number	12	9
Total reduction of test cases in percentage (%)	60.000	82.353
Time Taken (seconds)	0.00081	0.00087

Discussion

- For the text analysis part, LSTM had been used.
  - Loss and accuracy is in indirectly proportional relationship.
  - From the result comparison perspective, loss and accuracy highly affected on the dropout and the recurrent dropout value in the LSTM.
    - Falls in between 0 to 1
    - Purpose: Prevent overfitting situation from happening.
    - Current research using 0.2 for both variables.
- For test case prioritization part, advanced greedy approach had been implemented.
  - The least occurrence of the requirement will prioritize, and redundant test cases of the overlapped requirements will consolidate by test case which covers the greatest number of requirements
  - JPCT performed better than the SPCT [6] and compared the result with the JPCT and the proposed enhanced Greedy algorithm proposed
    - JPCT can cover 73% of the coverage while proposed enhanced version of Greedy algorithm covers 100% of the requirements listed with the minimum test cases needed.

Conclusion

- Research contribution
  - Enhanced Greedy algorithm
    - By improving the algorithm which helps to maximize the test case coverage to 100% and reducing the duplicate test cases up to 80%
- Limitation
  - LSTM has will drag the performance of the CPU and GPU and it will affect the epoch cycle.
  - Greedy algorithm unable to perform the mix-and-match test cases prioritization while maximize the requirement coverage.
  - Company's private and confidential git log and test requirement for the internal project is prohibited to use in this research. Hence, open-source database, named customer complaints had been used in this research.
- Future Works
  - LSTM able to train with multiple epochs with the better hardware and software requirements.
  - Advanced Greedy algorithm able to incorporate with the pairwise testing which will help in removing the test case redundancy happened either from Greedy algorithm or pairwise testing.

Appendices

- Example of Test Cases in the form of list

```
t0=1:1:0:0:0:0:0:0:0:0:0:0:0:0:0:1
t1=0:0:0:0:0:0:0:0:0:0:1:0:0:0:0:0
t2=0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0
t3=0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0
t4=0:0:0:0:0:0:1:0:0:0:0:0:0:0:0:0
t5=0:0:0:0:0:0:0:0:0:0:1:0:0:0:0:1:0
t6=0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0
t7=0:0:0:0:0:0:1:0:0:0:0:0:0:0:0:0
t8=0:0:0:0:0:0:0:0:0:0:1:0:0:0:0:0
t9=0:0:0:1:0:0:0:0:0:0:0:0:0:0:0:0
t10=0:0:0:0:0:1:1:0:0:0:0:0:1:0:0:0
t11=0:0:1:0:0:0:0:0:0:0:0:0:0:0:0:0
t12=0:0:0:1:0:0:0:1:0:0:1:0:0:0:0:0
t13=0:1:0:0:0:0:0:0:0:0:0:0:0:0:1:0
t14=0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0
t15=0:1:0:0:0:0:0:0:0:0:0:0:0:0:0:0
t16=1:0:0:0:0:0:0:0:1:1:0:0:0:0:0:0
t17=0:0:0:0:0:0:0:1:0:0:0:0:1:0:1:0
t18=0:0:0:0:1:0:0:0:0:0:0:0:0:0:0:0
t19=0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0
```

Appendices

- Final test cases after prioritization had completed

```
[redacted]:~$ reset; python3 rtm_greedy.py --rtm rtm_ori.txt
{'t5': [0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]}
[3 2 3 2 1 5 2 3 2 2 2 2 4 3 4 7]
==========Fine Tune Test Case ================

[[{'t5': [0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], 't6': [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], 't15': [1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
 0, 0, 0, 0], 't20': [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0], 't9': [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0], 't13': [0, 1, 1, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0], 't19': [0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], 't0': [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0], 't16': [0, 0, 0, 0,
 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0], 't18': [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0], 't3': [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0], 't27':
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1]}]]
Original number of test cases: 30
Number of reduced test cases: 12
Total reduction of test cases: 60.0%
Time Taken : 0.0007989406585693359
```

PNSQC

OCTOBER 10-12 2022

Q&A Session

PNSQC

OCTOBER 10-12 2022

THANK YOU