Test Report Prioritization to Assist Crowdsourced Testing

Yang Feng, Zhenyu Chen, Chunrong Fang, Baowen Xu
State Key Laboratory for Novel Software Technology, Nanjing University

James A. Jones
Department of Informatics, UC Irvine

This work is partially supported by the National Basic Research Program of China (973 Program 2014CB340702), the National Natural Science Foundation of China (61170067, 61373013), and the National Science Foundation under awards CAREER CCF-1350837 and CCF-1116943.
Problem Domain

- Crowdsourced testing is increasingly popular.
  - Testing Companies (such as crowdsourcedtesting.com, pay4bugs.com, and itestin.com)
  - Software Corporations (such as Baidu, Alibaba, Tencent)
Problem Domain

- Users perform testing tasks and write natural-language *test reports* to describe behavior.
Problem

- Crowdsourced test reports are numerous
- Reported failures must be inspected manually
## Example Test Report

<table>
<thead>
<tr>
<th>Environment</th>
<th>Input</th>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Operating System:** Windows 7-64-SP1  
**OS Version No:** MS Windows 6.1.7601  
**System Language:** Chinese  
**Screen Resolution:** 1920x1080 | Select menu -> options in the browser, set “ad block” closed in the Security page, open the link “http://www.qidian.com/Default.aspx”, and floating ads or ads around the edge of the web page are found; select menu -> options in the browser, set “ad block” enhanced in the security page, open the link to check; switch the browser mode, refresh the page to check again. | ![Image](image1.png) ![Image](image2.png) | When the blocking mode is switched, the number of blocked ads is not consistent with the previous one. |
Pilot Case Study

• In 2013, Chen and Liu conducted pilot study [ICSE-NIER 2014]
• Three software projects in collaboration with Baidu

<table>
<thead>
<tr>
<th>Project</th>
<th>Information</th>
<th># Test Reports</th>
<th># Failed Test Reports</th>
<th># Faults</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Baidu-Input, Verson: 3.7 <a href="http://shurufa.baidu.com/">http://shurufa.baidu.com/</a></td>
<td>274</td>
<td>186</td>
<td>27</td>
</tr>
<tr>
<td>P2</td>
<td>Baidu-Browser M18 <a href="http://liulanqi.baidu.com/">http://liulanqi.baidu.com/</a></td>
<td>231</td>
<td>47</td>
<td>22</td>
</tr>
<tr>
<td>P3</td>
<td>Baidu-Player <a href="http://player.baidu.com/">http://player.baidu.com/</a></td>
<td>252</td>
<td>62</td>
<td>18</td>
</tr>
</tbody>
</table>
Observations from Conversations with Baidu Developers

- The number of test reports was quickly too great to efficiently inspect
- Great number of false positives
- Many duplicates (for both true and false positives)
- Crowdsourced tests often performed easier/faster tasks
- Natural-language words were often sufficiently similar for duplicated reports
- Faults tended to "cluster" in functionality

Motivation for Automation

Motivation for Seeking Diversity

Motivation for Using Natural-Language Techniques

Motivation for Seeking Riskier Reports
Test Report Prioritization

1. Test Reports
2. Word Segmentation
3. Synonym Replacement
4. Synonym Dictionary
5. Build Keyword Dictionary
6. Build Keyword Vector
7. Prioritize Test Reports

Crowdsourced Workers → Test Reports → Word Segmentation → Synonym Replacement → Synonym Dictionary → Build Keyword Dictionary → Build Keyword Vector → Prioritize Test Reports → Developer
**Word Segmentation**

- ICTCLAS for word segmentation, which is a widely used Chinese NLP platform.
- Test report Input and Description segmented into words.
- Words tagged for Part-Of-Speech (POS).
- Bi-gram model was introduced to count the classes of words.
Synonym Replacement

- Used Chinese natural-language library to standardize synonyms and normalize words to help account for slight variations in the ways that different people described the same behavior.
Set of words in the test reports form the keyword dictionary (i.e., the corpus of words across test reports).

Only nouns and verbs are used.

Words are weighted by their frequency.
Build Keyword Vector

- Vector of boolean values
- One vector per test report
- Indexed by word in the keyword dictionary
Prioritize Test Reports

- Two primary prioritization strategies:
  - "Div": Seeking diversity among test reports
  - "Risk": Seeking riskier test reports
- Hybrid prioritization strategy:
  - "DivRisk": Balancing these two primary strategies
<table>
<thead>
<tr>
<th>No.</th>
<th>Input</th>
<th>Description</th>
<th>Ground Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>Log in renren.com in the compatibility mode, click “Home” or send friend “Send Gift”, then click back button, and click forward button.</td>
<td>There are some inconsistence between the page after clicking forward button and the page before clicking back button.</td>
<td>Not a bug</td>
</tr>
<tr>
<td>R2</td>
<td>Enter the compatibility mode, log in renren.com, click the link of some friends, then click “Home” button.</td>
<td>Click back button twice, and then go back to friend’s page</td>
<td>Bug 1</td>
</tr>
<tr>
<td>R3</td>
<td>Open IE, click tools-options-security, set Ad Blocking level to “strong”, enter <a href="http://soudu.org/">http://soudu.org/</a> in the address bar.</td>
<td>Floating ads and corner ads haven’t been blocked successfully.</td>
<td>Bug 2</td>
</tr>
<tr>
<td>R4</td>
<td>In compatibility mode in the homepage of baidu.com,input “group on” and search, then search “icecream”, search “red bull”, click back button twice, and then click forward.</td>
<td>Inconsistence between backing and forwarding operations.</td>
<td>Bug 1</td>
</tr>
<tr>
<td>R5</td>
<td>Open IE, in the maximum window, wait till program is loaded, then switch program, quick and continuously switch screen to maximum or full screen.</td>
<td>Sometimes the task bar is lost, especially when open other IE or system is busy. Move the mouse to some task and wait till the task thumbnail appears, the system restores.</td>
<td>Not a bug</td>
</tr>
<tr>
<td>R6</td>
<td>In IE click menu-options, in the security page set ad blocking to “closed”, open the link “<a href="http://www.narutom.com%E2%80%9D">http://www.narutom.com”</a>, ads are detected when loading the page. Then click menu-options, in the security page set ad blocking level to “strong”, open the link “<a href="http://www.narutom.com/%E2%80%9D">http://www.narutom.com/”</a> and check.</td>
<td>Failed to block ads.</td>
<td>Not a bug</td>
</tr>
<tr>
<td>R7</td>
<td>In IE click menu-options, in the security page set ad blocking to “closed”, open the link “<a href="http://www.gidian.com/Default.aspx%E2%80%9D">http://www.gidian.com/Default.aspx”</a>, many floating ads and side ads appear. Click menu-options, in the security page set ad blocking level to “strong”, open the link again and check.</td>
<td>When the mode of ad blocking is change, the number of blocked ads shows inconsistence.</td>
<td>Not a bug</td>
</tr>
<tr>
<td>No.</td>
<td>Input</td>
<td>Description</td>
<td>Input</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>R1</td>
<td>[Compatibility/v, modern, underf, log/v, renemen, /n, comen, /n, click/nw, &quot;/&quot;, personal, homepage, /n, ort, /w, givep, friend, /n, send, /w, pick, /w, /w, /n, cjohn, /n, /n, click/w, back/n, button/n, /w, click/w, forward/v, button/n, /w]</td>
<td>[Compatibility/v, modern, underf, log/v, renemen, /n, comen, /n, click/nw, &quot;/&quot;, personal, homepage, /n, ort, /w, givep, friend, /n, send, /w, pick, /w, /w, /n, cjohn, /n, /n, click/w, back/n, button/n, /w, click/w, forward/v, button/n, /w]</td>
<td></td>
</tr>
<tr>
<td>No.</td>
<td>Input</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>----------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td>[compatibility/v, mode/n, log/v, click/nw, personal/n, homepage/n, friend/n, gift/n, backword/v, button/n, forwad/v, page/n, content/n]</td>
<td>[兼容/v, 模式/n, 登录/v, 点击/nw, 个人/n, 主页/n, 好友/n, 礼物/n, 后退/v, 按钮/n, 前进/v, 页面/n, 内容/n]</td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>[enter/v, compatibility/v, mode/n, log/v, click/nw, friend/n, link/nw, personal/n, homepage/n, button/n, load/nw, back/v, page/n]</td>
<td>[进入/v, 兼容/v, 模式/n, 登录/v, 点击/nw, 好友/n, 链接/nw, 个人/n, 主页/n, 按钮/n, 加载/nw, 后退/v, 页面/n]</td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td>[open/v, IE/n, tool/n, options/n, security/an, ads/n, block/v, select/v, address/n, input/v, page/n, floating/v, corner/nw, not/v]</td>
<td>[打开/v, 浏览器/n, 工具/n, 选项/n, 安全/an, 广告/n, 拦截/v, 选择/v, 地址/n, 输入/v, 页面/n, 悬浮/v, 右下角/nw, 没有/v]</td>
<td></td>
</tr>
<tr>
<td>R4</td>
<td>[compatibility/v, mode/n, input/v, groupon/v, click/nw, search/v, button/n, icecream/n, red-bull/nw, back/v, forward/v, back-forward/nw, result/v]</td>
<td>[兼容/v, 模式/n, 输入/v, 团购/v, 点击/nw, 搜索/v, 按钮/n, 冰淇淋/n, 红牛/nw, 后退/v, 前进/v, 前进后退/nw, 结果/v]</td>
<td></td>
</tr>
<tr>
<td>R5</td>
<td>[open/v, IE/n, max/v, condition/n, wait/v, program/n, load/nw, finish/v, do/v, switch/v, full-screen/nw, appear/v, system/n, task/n, missed/v, possibility/n, mouse/n, thumbnail/n, return/v]</td>
<td>[打开/v, 浏览器/n, 最大化/v, 状况/n, 等待/v, 程序/n, 加载/nw, 完成/v, 进行/v, 切换/v, 全屏/nw, 出现/v, 系统/n, 任务/n, 消失/v, 可能性/n, 鼠标/n, 略图/n, 恢复/v]</td>
<td></td>
</tr>
<tr>
<td>R6</td>
<td>[IE/n, click/nw, menu/n, options/n, security/an, page/n, ads/n, block/v, close/v, open/v, link/nw, page/n, load/nw, find/v, strong/n, check/v, fail/v]</td>
<td>[浏览器/n, 点击/nw, 菜单/n, 选项/n, 安全/an, 页面/n, 广告/n, 拦截/v, 关闭/v, 打开/v, 链接/nw, 网页/n, 加载/nw, 发现/v, 强力/v, 查看/v, 失败/v]</td>
<td></td>
</tr>
<tr>
<td>R7</td>
<td>[IE/n, click/nw, menu/n, options/n, security/an, page/n, ads/n, block/v, closed/v, open/v, link/nw, appear/v, floating/v, strong/n, check/v, switch/v, mode/n, refresh/v, button/n, select/v, change/v, number/n]</td>
<td>[浏览器/n, 点击/nw, 菜单/n, 选项/n, 安全/an, 页面/n, 广告/n, 拦截/v, 关闭/v, 打开/v, 链接/nw, 显示/v, 漂浮/v, 强力/v, 查看/v, 切换/v, 模式/n, 刷新/v, 按钮/n, 选择/v, 改变/v, 数目/n]</td>
<td></td>
</tr>
<tr>
<td>Word</td>
<td>Freq</td>
<td>Word</td>
<td>Freq</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
<td>----------</td>
<td>------</td>
</tr>
<tr>
<td>do</td>
<td>1</td>
<td>button</td>
<td>4</td>
</tr>
<tr>
<td>missed</td>
<td>1</td>
<td>input</td>
<td>2</td>
</tr>
<tr>
<td>gift</td>
<td>1</td>
<td>IE</td>
<td>4</td>
</tr>
<tr>
<td>floating</td>
<td>1</td>
<td>options</td>
<td>3</td>
</tr>
<tr>
<td>back</td>
<td>3</td>
<td>mode</td>
<td>4</td>
</tr>
<tr>
<td>ice-cream</td>
<td>1</td>
<td>open</td>
<td>4</td>
</tr>
<tr>
<td>finish</td>
<td>1</td>
<td>ads</td>
<td>3</td>
</tr>
<tr>
<td>full-screen</td>
<td>1</td>
<td>tool</td>
<td>1</td>
</tr>
<tr>
<td>condition</td>
<td>1</td>
<td>page</td>
<td>1</td>
</tr>
<tr>
<td>wait</td>
<td>1</td>
<td>link</td>
<td>3</td>
</tr>
<tr>
<td>forward</td>
<td>2</td>
<td>red-bull</td>
<td>1</td>
</tr>
</tbody>
</table>

| 进行       | 1    | 按钮     | 4    | 强力     | 2    | 主页     | 2    | 改变     | 1    | 程序     | 1    |
| 消失       | 1    | 输入     | 2    | 个人     | 2    | 切换     | 2    | 出现     | 1    | 地址     | 1    |
| 礼物       | 1    | 浏览器   | 4    | 好友     | 2    | 略图     | 1    | 内容     | 1    | 没有     | 1    |
| 漂移       | 1    | 选项     | 3    | 登录     | 2    | 前进后退 | 1    | 数目     | 1    | 查看     | 2    |
| 后退       | 3    | 模式     | 4    | 搜索     | 1    | 拦截     | 3    | 右下角   | 1    | 可能性   | 1    |
| 冰淇淋     | 1    | 打开     | 4    | 团购     | 1    | 失败     | 1    | 关闭     | 2    | 点击     | 5    |
| 完成       | 1    | 广告     | 3    | 加载     | 3    | 菜单     | 2    | 进入     | 1    | 鼠标     | 1    |
| 全屏       | 1    | 工具     | 1    | 恢复     | 1    | 安全     | 3    | 最大化   | 1    | 显示     | 1    |
| 状况       | 1    | 网页     | 1    | 刷新     | 1    | 系统     | 1    | 任务     | 1    | 选择     | 2    |
| 等待       | 1    | 链接     | 3    | 悬浮     | 1    | 页面     | 5    | 结果     | 1    | 发现     | 1    |
| 前进       | 2    | 红牛     | 1    | 兼容     | 3    |
All three prioritization strategies (Div, Risk, and DivRisk) begin with choosing a test report that contains the most keywords (i.e., most information).
Div Prioritization Strategy

• Based on the principle of diversity of test selection

• Greedy algorithm

• Prefer to select the next test report with the maximal distance to already chosen and sorted test-report set

• Fully automatic
Risk Prioritization Strategy

• Seeks to reveal faults as early as possible

• Dynamic prioritization strategy, based on thus-far inspection results

• Risk values of each keyword are updated as reports are inspected

• Remaining test reports are sorted based on updated risk values

• Semi-automatic
**DivRisk** Prioritization Strategy

- Hybrid technique to balance the goals of the primary strategies

- Dynamic prioritization strategy, based on thus-far inspection results to assess risk and keyword distance to assess diversity

- Risk values of each keyword are updated (by $\delta$) as reports are inspected

- Remaining test reports are sorted based on updated risk values and keyword distance

- First a candidate set (of size $n_c$) of test reports with the greatest distance from the already prioritized set (i.e., diversity), then the report with the greatest risk values (sum of keyword risk values) is selected next

- Semi-automatic
Experiment

- **Subject:** Three Baidu projects from pilot study [Chen, Liu 2014]
- **Ground truth (report inspection):** Baidu developers
- **Independent variable:** Prioritization Strategy
  - Div
  - Risk
  - DivRisk
  - Random (baseline)
  - Best-case
  - Worst-case
- **Dependent variable:** APFD
- **Parameters:** δ=0.2, n_c=8, repeat=50
Results: P1
Results: P2
Results: P2
Results: P3

![Box plots showing performance comparison]

- **Best** line at 0.95
- **APFD** on the y-axis
- **Random**, **Risk**, **Div**, and **DivRisk** categories on the x-axis

The box plots indicate the distribution of performance metrics across different categories, with **DivRisk** showing the best performance.
Results: P3
Observations

• Prioritization strategies performed best on P1, which was the Android mobile application and was dominated by compatibility issues

• Prioritizing is, in general, better than random (i.e., un-prioritized)

• There may be room for improvement to approach the best-case order
Related Work

Test-Suite Prioritization
• Prioritizing risky tests (e.g., Dickenson01, Yan10)
• Diversity prioritization (e.g., Ledru09, Yoo09, Fang14)

Failure-Report Classification
• Clustering bug reports (e.g., Runeson07, Wang08)
• Clustering execution information (e.g., Bowring04, Lo09, Dhaliwal11)

Crowdsourced Testing
• E.g., Liu12, Nebeling12, Pastore13, Dolstra13
Future Work

• Images and screenshots may be used in future test-report prioritization to help distinguish behaviors

• Multi-objective heuristic search could be applied

• Benchmarks of crowdsourced data to be gathered to assist future replication and technique improvement (current archive is at http://www.qingkaishi.com/?Android_Resources_Repository)
Conclusions

- Test-report prioritization technique, based on natural-language descriptions
- Three prioritization strategies, inspired by test-suite prioritization and failure classification
- Results showed promise for our three crowdsourced projects