移动应用众包测试

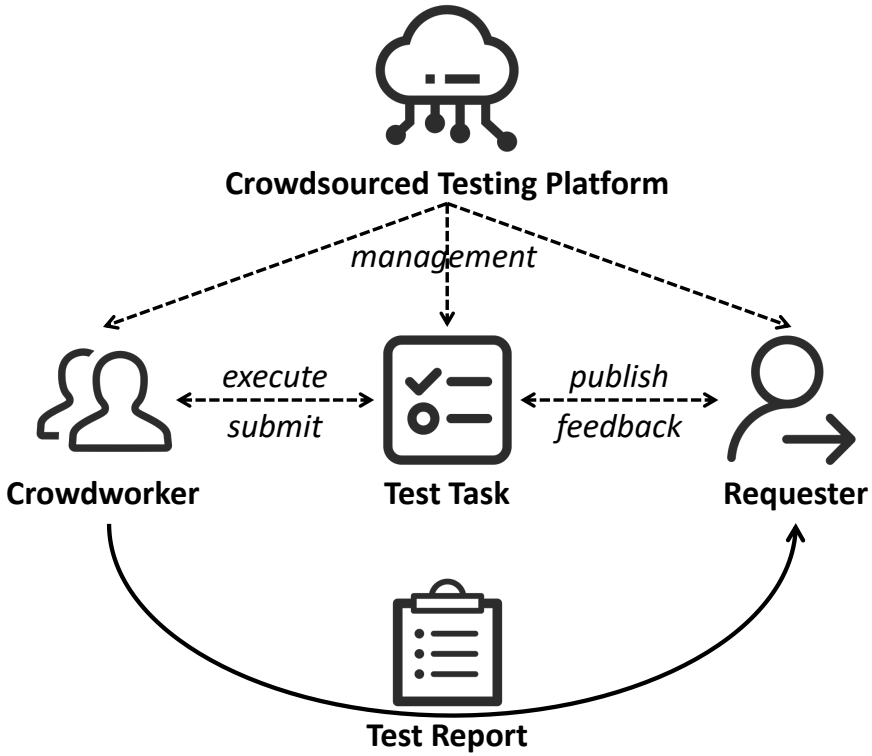
南京大学 软件学院 iSE实验室

背　景

- 众包：利用群体力量来完成传统方法中成本高昂或耗时的大规模任务，是Howe Jeff于2006年在美国《连线》杂志上首次提出的一种商业模式。

- 移动应用众包测试：利用群体力量完成移动应用测试任务
  - 移动应用碎片化问题：品牌、型号、系统、传感器……
  - uTest, Testin, Baidu Crowd Test, Alibaba Crowd Test, TestIO, MoocTest

众包测试流程



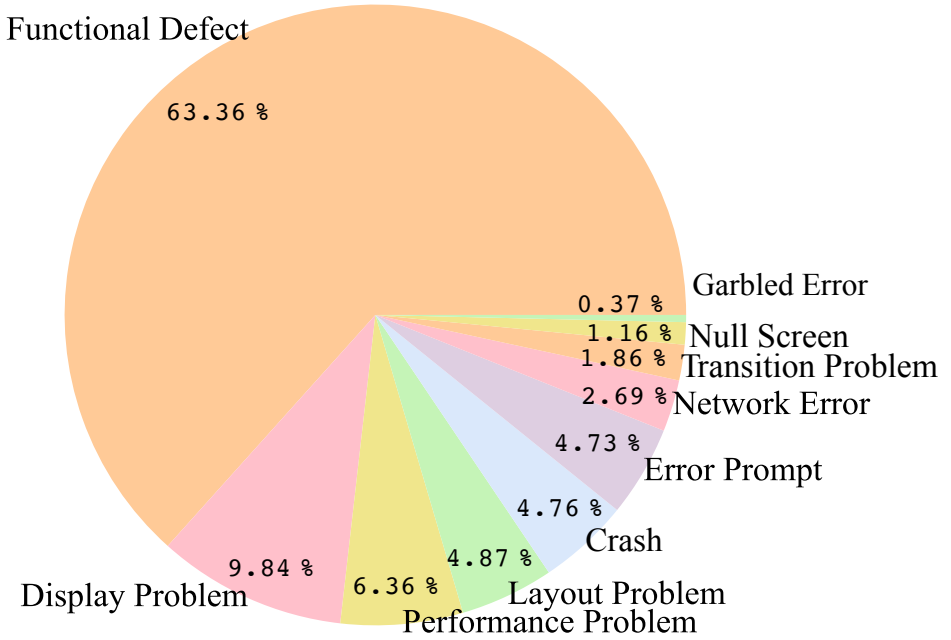
- **申请上传**：用户将自己的应用程序上传到众测平台，并指定相应的测试任务和酬劳信息。
- **任务选择和环境设置**：众测人员自由选择他们想要完成的任务。选择后测试人员从平台上下载应用程序进行测试。
- **提交报告**：众测人员根据选择的待测应用，对测试到的缺陷提交缺陷报告。
- **生成最终测试报告**：平台收集补充信息，生成最终的缺陷报告，包括：一般信息、设备信息、操作路径等。
- **报告验证**：客户将验证所有最终的缺陷报告，并决定如何酬劳每个提交报告的众包测试人员。

众包测试

- 众包测试的优势
  - 更充分的测试时间
  - 更广泛的测试方法
  - 更多样的测试环境
  - 更全面的测试方法
  - ……

众包测试

- 众包测试中的缺陷类型
  - 功能缺陷
  - 显示问题
  - 性能问题
  - 布局问题
  - 应用崩溃
  - 错误提示
  - 空白屏



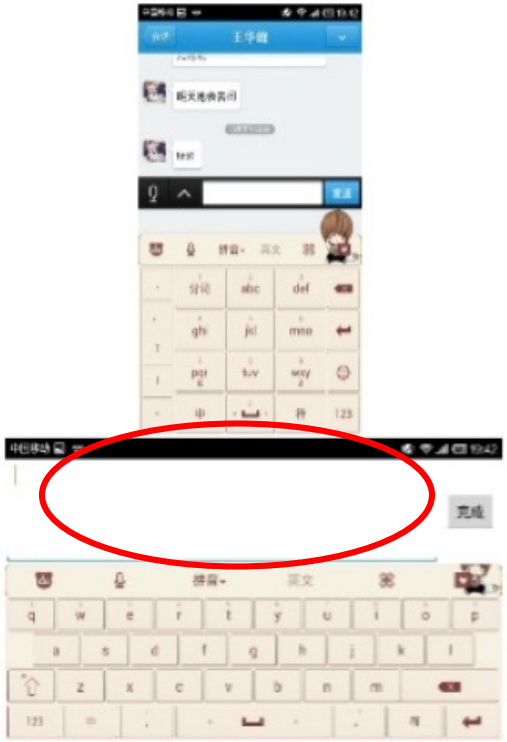
众包测试

- 众包测试面临的挑战
  - 任务分配
  - 任务奖励
  - 众测过程引导
  - 测试报告质量控制
  - ……

协作式众包测试

- 完成测试任务过程中进行信息共享与任务分配，用户在本系统中既承担测试任务也承担审核任务，充分利用用户协作，完成目标任务。
- **信息共享**：用户在提交报告时进行实时相似报告推荐，避免重复报告提交。
- **任务分配**：审核页面推荐待审核的报告列表，测试页面推荐待测页面。
- **协作方式**：点赞点踩操作：利用用户的交叉审核，验证报告有效性。
- **一键Fork**：Fork他人结果后进行修改，利用多人协作提升报告质量。

测试报告

E: Environment	I: Input	O: Output	D: Description
前置条件：手机开启允许横屏模式 测试机型:小米2A、魅族M2； 测试环境： 手机QQ2012	1. 进入设置页面 2. 关闭浮动显示主题特效 3. 开启横屏时启用全屏模式 4. 进入超级皮肤页面 5. 选择有凸出图片的主题 6. 返回后进入横屏模式		使用默认主题列表中第一列第一个主题，关闭浮动显示主题特效和打开横屏全屏模式，横屏时布局出错，主题图片会被遮挡。
Prerequisite: Turn on the landscape mode Mobile Type: MIUI 2A, SUMSUNG MX2 Testing Environment: Mobile QQ2012	1. Go to settings page 2. Close the floating effect and turn on the topic effects 3. Turn on the landscape mode and enable the full-screen mode 4. Go to the super skin setting page 5. Select any theme with a floating picture 6. Exit setting page and enter the landscape mode		Use the first column the first topic of default topic list, close floating display effect and turn on landscape screen and full-screen mode. A landscape screen layout error occur, the theme pictures will be blocked.

众测报告质量优化

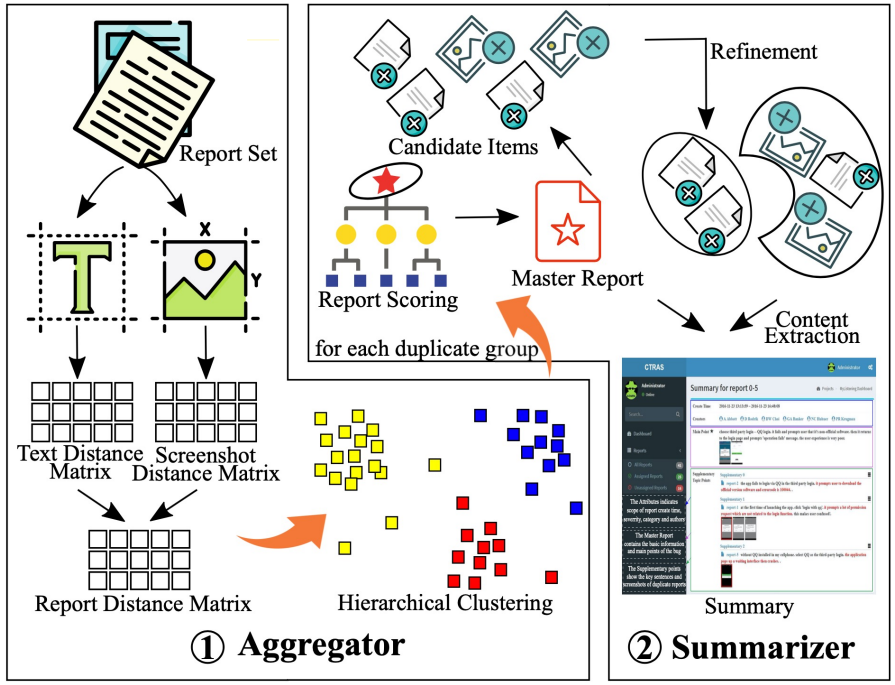
- **众测报告信息**
  - 缺陷截图
  - 文本描述
    - 缺陷行为描述、复现步骤、期望行为
  - 测试环境信息

众测报告聚合

- 大量的测试报告——较多的重复数据
  - 懒惰和缺乏经验的用户。
  - 测试报告搜索功能较差。
  - 偶然的重复——由于网络等原因不小心提交了多次。

  - 给开发者增加了许多工作量。
  - 不同的信息可以使开发者对Bug有一个更全面的了解。

众测报告聚合



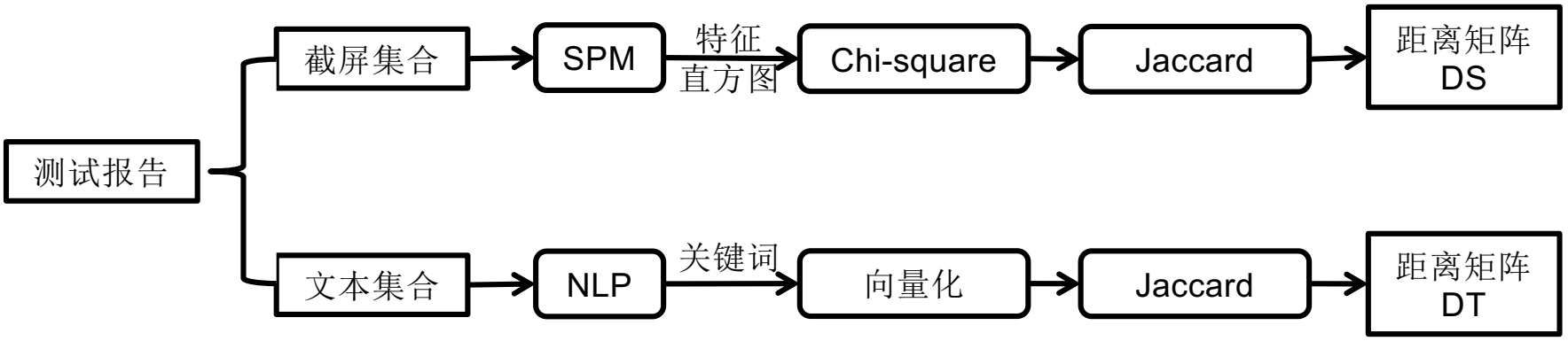
- Aggregator：对所有的测试报告做聚类，将相同的或相似的测试报告聚为同类。
- Summarizer：对每一类测试报告做整合，将其中的相关信息以可视化的方式最大化的呈现给开发者。

众测报告聚合

具体实现——Aggregator

1. 截屏集合的距离矩阵DS计算

2. 文本集合的距离矩阵DT计算



众测报告聚合

具体实现——Aggregator

3. 距离矩阵的合并——Balanced Distance的计算

- 若文本完全相同，则认为平衡距离为0
- 若截屏完全相同，则平衡距离为加权的文本距离
- 文本和截屏均不相同时，使用二者的调和平均数

$$BD(r_i, r_j) = \begin{cases} 0, & \text{if } DT(r_i, r_j) = 0 \\ \alpha \times DT(r_i, r_j), & \text{if } DS(r_i, r_j) = 0 \\ (1+\beta^2)\frac{DS(r_i, r_j) \times DT(r_i, r_j)}{\beta^2 DS(r_i, r_j) + DT(r_i, r_j)}, & \text{otherwise} \end{cases}$$

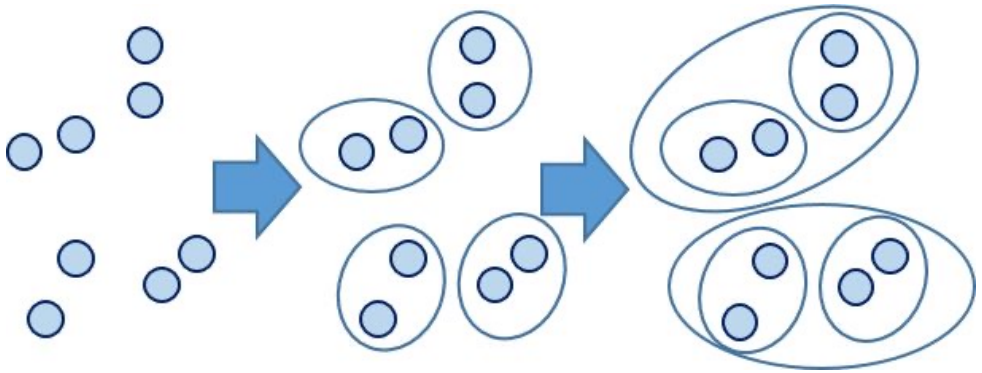
具体实现——Aggregator

4. 合成层次聚类（Agglomerative Hirarchical Clustering）

1) 将所有样本点各自归为一类，两两类别计算距离

2) 将距离最短的两个类合成一个类，计算新类与旧类的距离

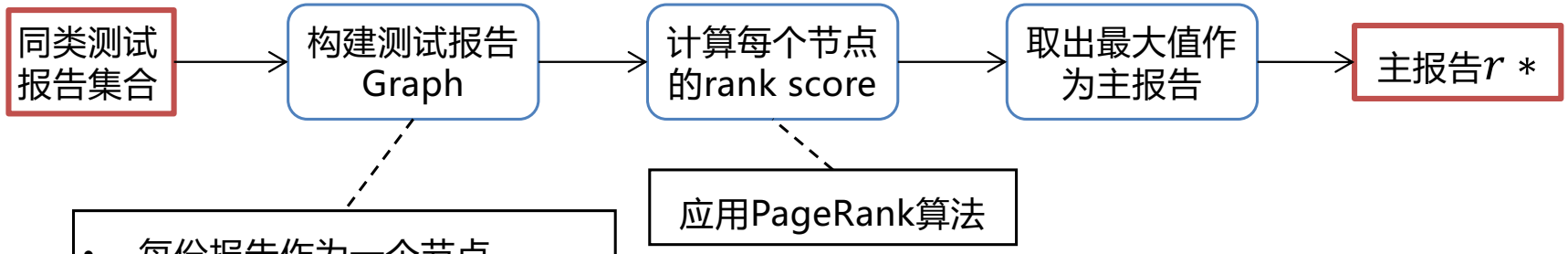
3) 重复操作2)直至所有样本点都归于一类

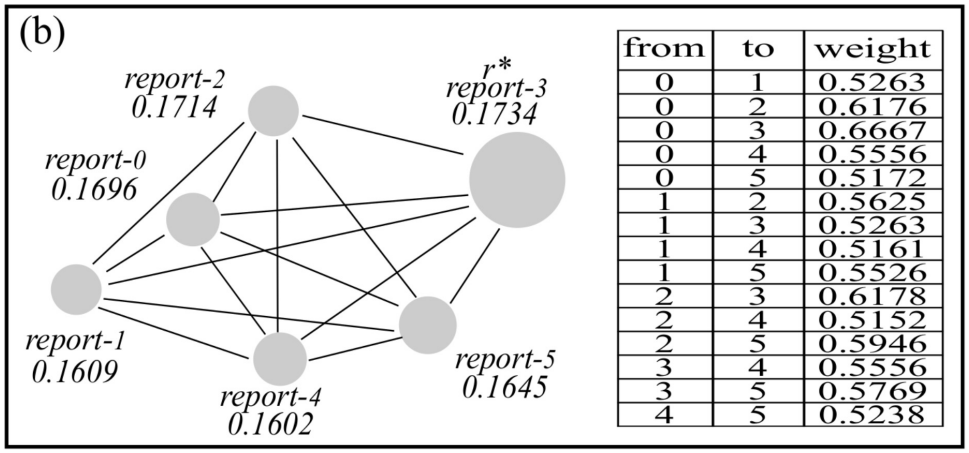


众测报告聚合

具体实现——Summarizer

1. 识别主报告$r*$





from	to	weight
0	1	0.5263
0	2	0.6176
0	3	0.6667
0	4	0.5556
0	5	0.5172
1	2	0.5625
1	3	0.5263
1	4	0.5161
1	5	0.5526
2	3	0.6178
2	4	0.5152
2	5	0.5946
3	4	0.5556
3	5	0.5769
4	5	0.5238

众测报告排序

具体实现——Summarizer

2. 生成补充信息

1） 补充信息识别

- 以非主报告的截屏/文本与主报告的截屏/文本的距离确定补充信息

2) 补充信息提纯

- 补充信息聚类：根据距离对补充信息进行聚类
- 同源信息聚类：再次根据信息来源对补充信息进行聚类
- 计算每个补充信息类的权重

3. 生成总结

- 将主报告与补充信息结合形成总结

众测报告排序

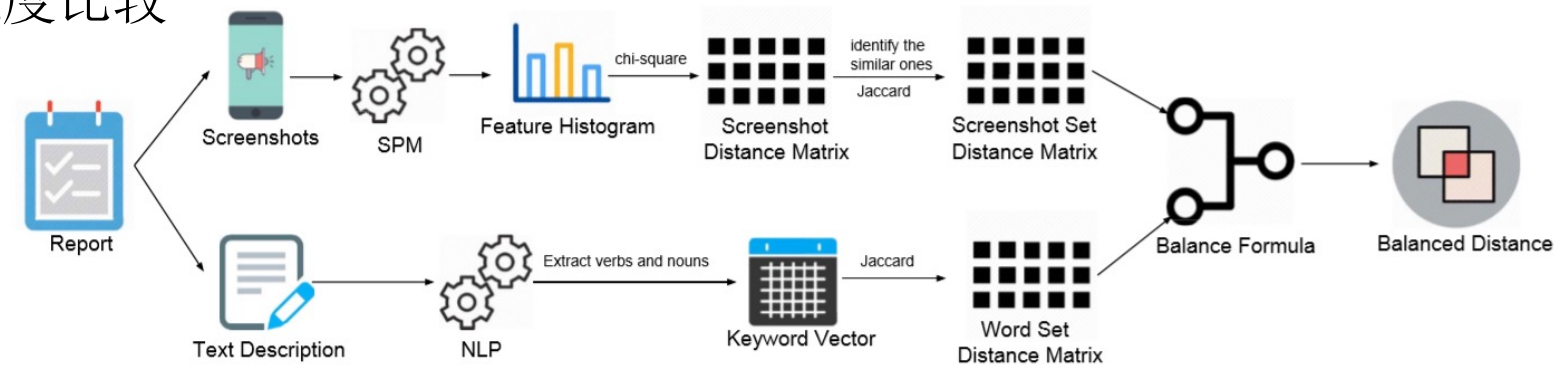
- 众包测试的优势
  - 可以向开发者提供多种不同软硬件平台支持的真实用户数据和操作信息
- 移动应用测试众包报告
  - 少量文本+多张Bug截图
- 众包测试质量管控问题
  - 测试报告数量庞大、人工审查过于耗时
- 为开发者提供一种定义和排序测试报告的技术非常重要

众测报告排序

- 当前研究主要关注测试报告的文本描述或者程序代码的执行树
- 不适用于移动应用测试报告：
  - 移动应用测试报告：少量的文字描述 + 缺陷截图
- 针对移动应用应该同时将缺陷截图和文本描述纳入考虑以排序的众包测试报告处理方法
- 报告排序原则：The earlier a bug is detected, the cheaper it is to remedy.

众测报告排序

- 使用文本+图片结合分析的方法
- 通过自然语言处理算法计算文本的相似度
- 通过SPM算法（Spatial Pyramid Matching）识别缺陷截图相似度
- 通过使用一种多目标的优化算法结合文本与截图相似度完成针对测试报告的相似度比较



众测报告排序

- 处理文本主要涉及两个操作：
  - 1）关键词的提取
  - 2）文本距离的计算
- 首先我们进行文本的切割来处理中文描述
- 根据上下文标记每个词的词性
- 过滤无用词汇：这些词会对文本距离计算产生消极影响
- 处理过程不仅局限于中文文本分析，可使用其他NLP算法识别多语言文本

文本处理

- 形式化定义
  - 测试报告集合$R(r) = \{(S_i, T_i) | i = 0, \dots, n\}$
  - S表示可能包含所描述Bug的屏幕截图
  - T表示描述缺陷行为的文本

- 使用Jaccard距离计算测试报告集$R(r)$中文本描述T_i间距离
  - $DT(r_i, r_j) = 1 - \frac{|K_i \cap K_j|}{|K_i \cup K_j|}$
  - K_i表示测试报告T_i的关键词集合
  - $DT(r_i, r_j)$表示报告r_i和r_j的文本距离

众测报告排序

- 处理截图主要涉及三个操作：
  - 建立特征直方图
  - 计算截图之间的距离
  - 计算截图集合之间的距离

(a) Playing-1 (b) Playing-2 (c) Lyrics-1 (d) Lyrics-2

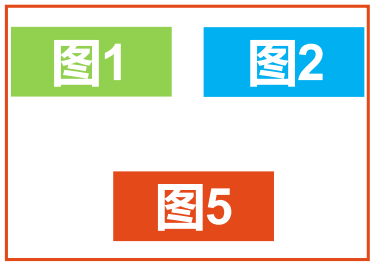
- 将图片转化为特征向量便于计算
- 我们需要计算距离，由于移动应用的界面跟app强相关，且有各种各样的分辨率和背景，使用简单的RGB分析不能满足需求
- 使用SPM进行计算

截图处理

- 截图的距离 VS. 截图集合的距离
- 为了解决移动设备多样性，尤其是分辨率多样性，设定阈值 γ 来评估不同设备同一页面截图
- 针对每个测试报告中的截图集合，通过 γ 来筛选出表述内容相同的图片，并将相同图片的特征直方图以求平均的形式形成一张新的直方图

截图处理

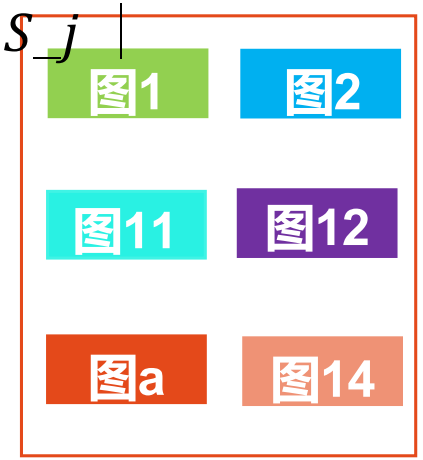
- 在完成测试报告内部的筛选工作之后，再通过阈值 Y 在测试报告的截图集合之间进行筛选
- 对于测试报告ri和rj，及其对应的筛选过后的截图集合Si和Sj，计算距离的公式为：
- $DS(r\_i, r\_j) = 1 - |S\_i \cap S\_j| / |S\_i \cup S\_j|$



测试报告i的截图集合S_i

测试报告j的截图集合S_j

$S_i \cap S_j$



$S_i \cup S_j$

众测报告排序

- 当文本距离为0的时候，我们认为两份报告的距离也为零；
- 当图像报告的距离为0的时候，我们将文本距离乘以一个系数作为最终的报告距离，在实践中，经过验证，我们将系数设置为0.75时能达到最佳效果。
- 当文本距离和图像距离均不等于零时，我们计算文本距离和图像距离的调和平均数作为最终的报告距离。β用来设置图片和文本的权重。在实践中，我们使用β为1，即图片和文本权重相同。

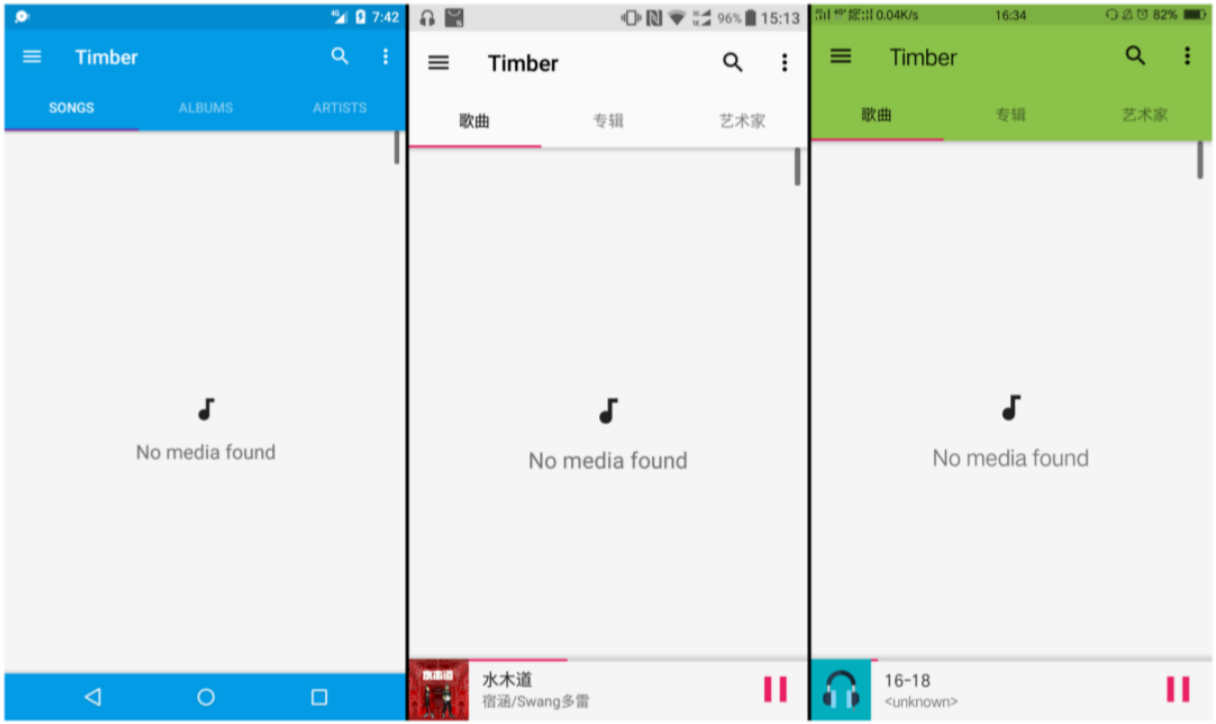
$$BD(r_i, r_j) = \begin{cases} 0, & \text{if } DT(r_i, r_j) = 0 \\ \alpha \times DT(r_i, r_j), & \text{if } DS(r_i, r_j) = 0 \\ (1+\beta^2)\frac{DS(r_i, r_j) \times DT(r_i, r_j)}{\beta^2 DS(r_i, r_j) + DT(r_i, r_j)}, & \text{otherwise} \end{cases}$$

深度众测报告排序

- 众包可以非常好的解决移动应用碎片化的问题
- 移动应用测试报告：文本描述+缺陷截图
- 众包问题：审查效率问题
  - 大量的报告被提交上来，近乎82%的报告为重复的
  - 文本问题：同样的问题可以通过不同的描述方式编写
  - 截图问题：图片整体UI相似，涉及到Bug的部分非常小

深度众测报告排序

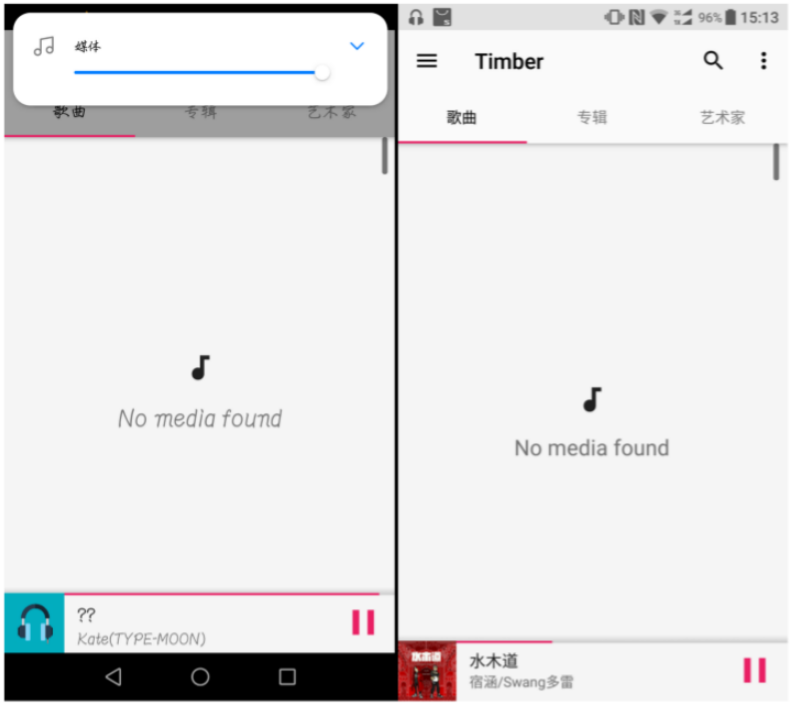
- 不同的应用主题
  - 缺陷：未找到本地文件资源



Report #1135 Report #1128 Report #1163

深度众测报告排序

- 同页面的不同Bug
  - 缺陷：#1127-音量自动调整至最大；#1128-未找到本地文件资源



Report #1127 Report #1128

深度众测报告排序

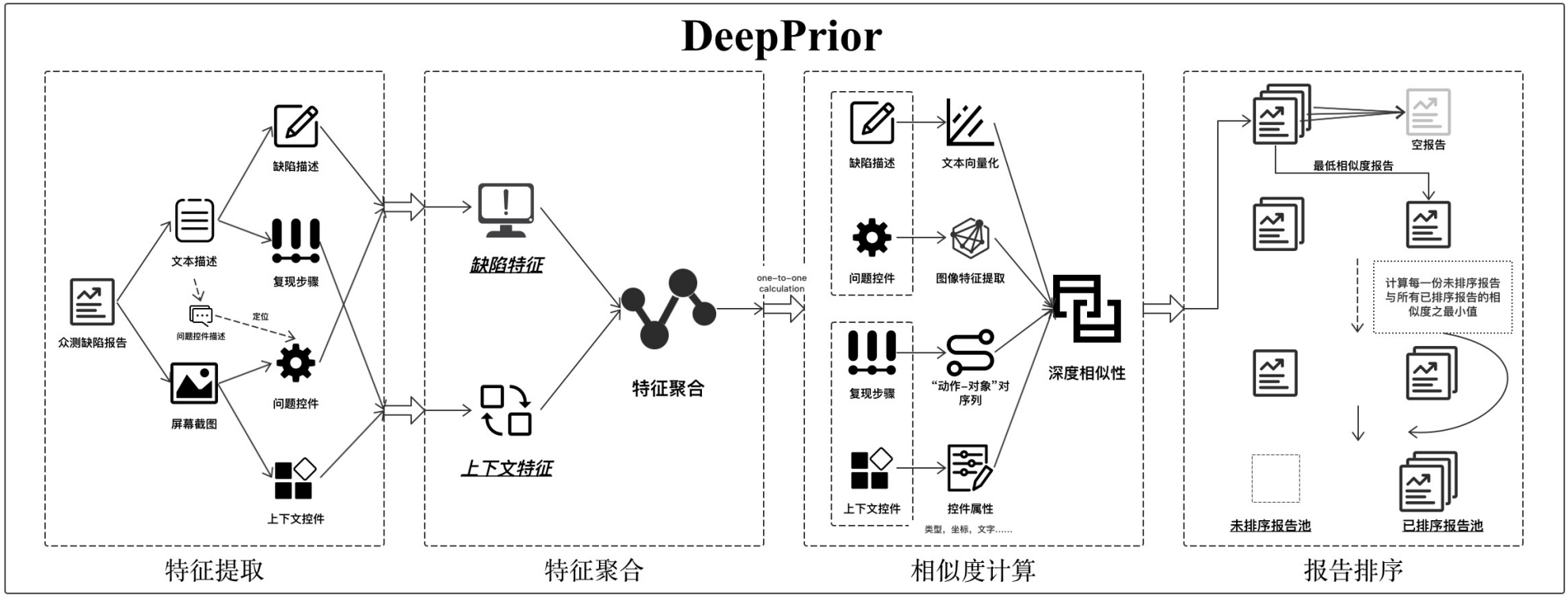
- 不同页面的相同Bug
  - 缺陷：提示发表失败的评论显示在评论别表中

Report #25　　Report #146

深度众测报告排序

- 当前研究存在较大局限性
  - 1）主要着眼于文本的分析理解
  - 2）仅引入简单的截图分析以辅助文本
  - 3）简单地将文本与图片拼接起来进行分析
- 文本：预定义词表——提取关键字 + 标准化
- 截图：将图片按整体处理 + 提取多个特征向量

深度众测报告排序


DeepPrior
缺陷描述
文本描述
复现步骤
定位
问题控件描述
众测缺陷报告
问题控件
屏幕截图
上下文控件
缺陷特征
上下文特征
特征聚合
one-to-one calculation
缺陷描述
文本向量化
问题控件
图像特征提取
复现步骤
“动作-对象”对序列
上下文控件
控件属性
类型，坐标，文字……
深度相似性
空报告
最低相似度报告
计算每一份未排序报告与所有已排序报告的相似度之最小值
未排序报告池
已排序报告池
特征提取
特征聚合
相似度计算
报告排序

深度众测报告排序

- 通过计算机视觉(CV)技术提取控件
  - 问题控件(W_p)
  - 上下文控件(W_c)
- 文本通过NLP技术分解为两部分
  - 复现步骤(R)
  - 缺陷描述(P)
- 建立DeepFeature
  - Bug Feature (BFT): $W_p + P$
  - Context Feature (CFT): $W_c + R$
- DeepPrior使用DeepFeature计算DeepSimilarity
- 开始排序

Report #25 Report #146

深度众测报告排序

图片特征提取：

- 使用CV和DL技术提取所有控件并分析
  - 问题控件：定位产生Bug的控件元素
  - 上下文控件：除问题控件之外的其他控件

文本特征提取：

- 使用DL算法处理文本
  - TextCNN、jieba库、停用词库、Word2vec
- 分析Bug描述
- 分析复现步骤

深度众测报告排序

- 将所有特征聚合为两种特征类型：
  - Bug Feature (BFT)：
    与缺陷直接相关，包括问题控件和缺陷描述
  - Context Feature (CFT)：
    为缺陷构建上下文，包括上下文控件和复现步骤

深度众测报告排序

- $DeepSimilarity = \gamma * Sim_{BFT} + (1 - \gamma) * Sim_{CFT}$
- $Sim_{BFT} = \alpha * Sim_{W_P} + (1 - \alpha) * Sim_P$
- $Sim_{CFT} = \beta * Sim_{W_C} + (1 - \beta) * Sim_R$

- Sim_{W_P}：SIFT+FLANN库
- Sim_P：Word2Vec+Euclidean Metric
- Sim_{W_C}：CNN+Euclidean Metric
- Sim_R：DWT算法

深度众测报告排序

Algorithm 1 Crowdsourced Test Report Prioritization

Input: Crowdsourced Test Report Set $R_{initial}$
Output: Prioritized Crowdsourced Test Report Set P

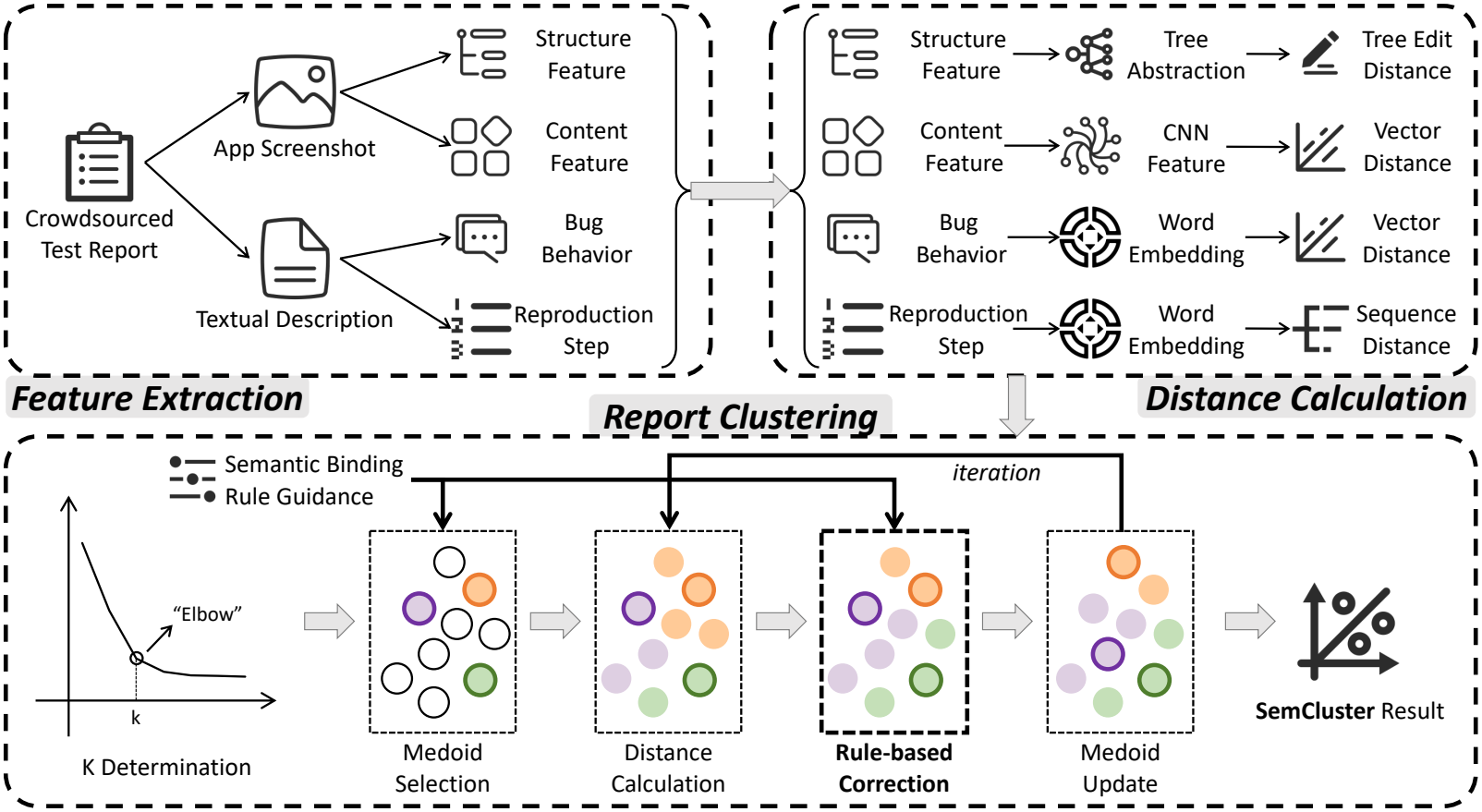
initiate unprioritized report pool $U \leftarrow R_{initial}$
initiate prioritized report pool $P = \emptyset$
initiate target report r_t
initiate NULL Report r_{null}
P.append(r_{null})
while $|U| \neq 0$ **do**
    initiate $similarity = 2$
    for each $r \in U$ **do**
        for each $r_p \in P$ **do**
            initiate $similarity_r = 2$
            $Sim_{BFT} = \alpha * Sim_{W_P} + (1 - \alpha) * Sim_P$
            $Sim_{CFT} = \beta * Sim_{W_C} + (1 - \beta) * Sim_R$
            calSemSim()$= \gamma * Sim_{BFT} + (1-\gamma) * Sim_{CFT}$
            $similarity_r$ += calSemSim(r, r_p)
            if calSemSim(r, r_p) $< similarity_r$ **then**
                $similarity_r$ = calSemSim(r, r_p)
            end if
        end for
        if $similarity_r < similarity$ **then**
            $r_t = r$
            $similarity = similarity_r$
        end if
    end for
    P.append(r_t)
    U.remove(r_t)
end while
return P

- 两个报告池：未排序报告池、已排序报告池
- 将所有报告初始化到未排序报告池中
- 向已排序报告池中添加Null Report
- Null Report：一种预定义的测试报告
- 从未排序报告池中，选出一份和已排序报告池中最不相似的报告加入到已排序报告池中，循环调用

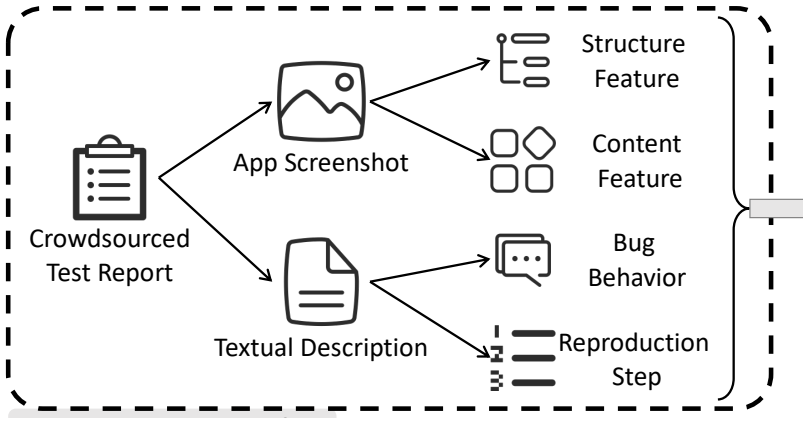
众测报告半监督聚类

- 报告数量繁多、所描述缺陷重复
- 重复报告带来的信息增益
- 传统聚类方式的缺陷与不足
  - 未利用报告语义信息
  - 忽略图像文本关联

众测报告半监督聚类

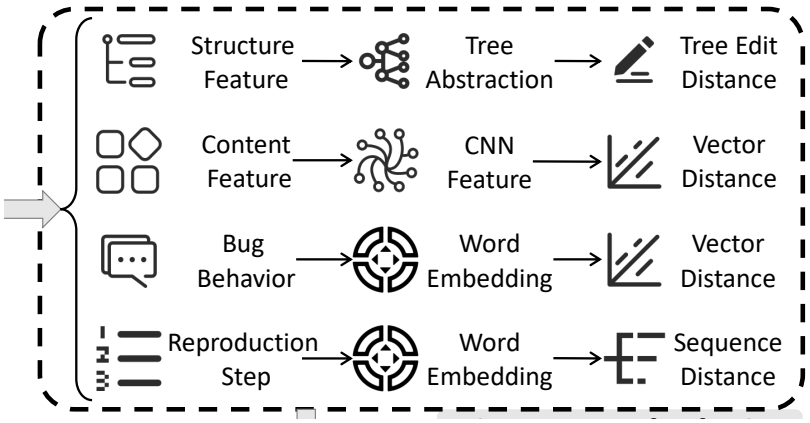

Crowdsourced Test Report
App Screenshot
Textual Description
Structure Feature
Content Feature
Bug Behavior
Reproduction Step
Feature Extraction
Structure Feature
Tree Abstraction
Tree Edit Distance
Content Feature
CNN Feature
Vector Distance
Bug Behavior
Word Embedding
Vector Distance
Reproduction Step
Word Embedding
Sequence Distance
Distance Calculation
Report Clustering
Semantic Binding
Rule Guidance
iteration
"Elbow"
k
K Determination
Medoid Selection
Distance Calculation
Rule-based Correction
Medoid Update
SemCluster Result

众测报告半监督聚类

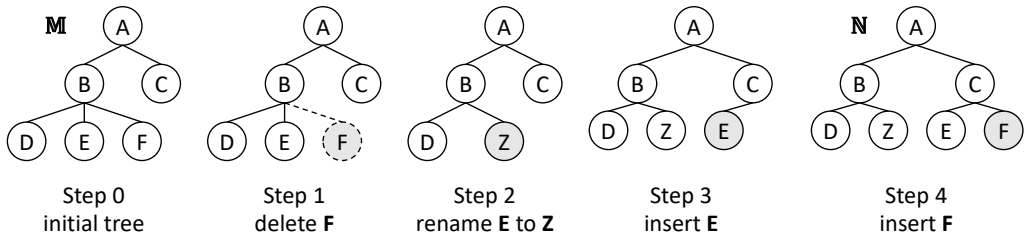


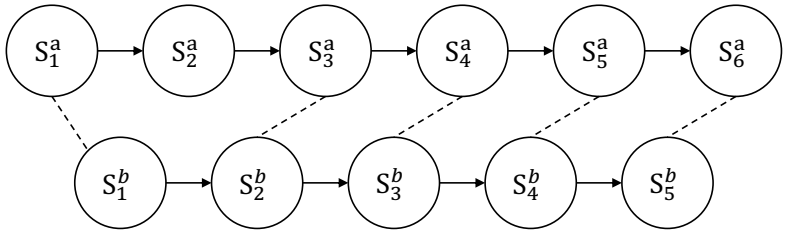
- 特征提取
  - 应用截图
    - 结构特征
    - 内容特征
  - 文本描述
    - 缺陷行为
    - 复现步骤

众测报告半监督聚类

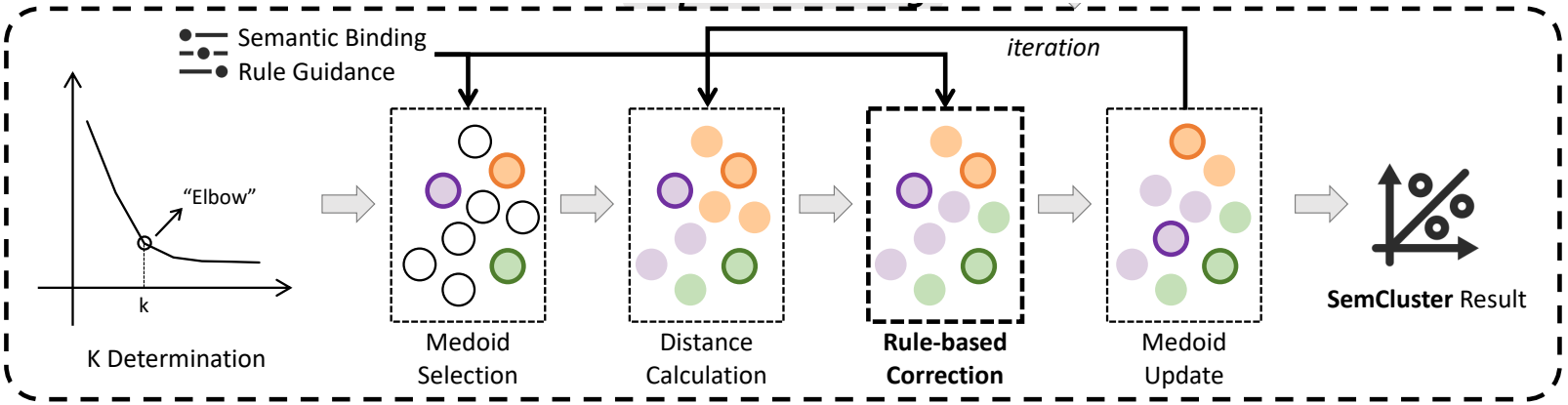


- 特征距离计算
  - 结构特征：树编辑距离
  - 内容特征：向量特征距离
  - 缺陷行为：向量特征距离
  - 复现步骤：序列距离



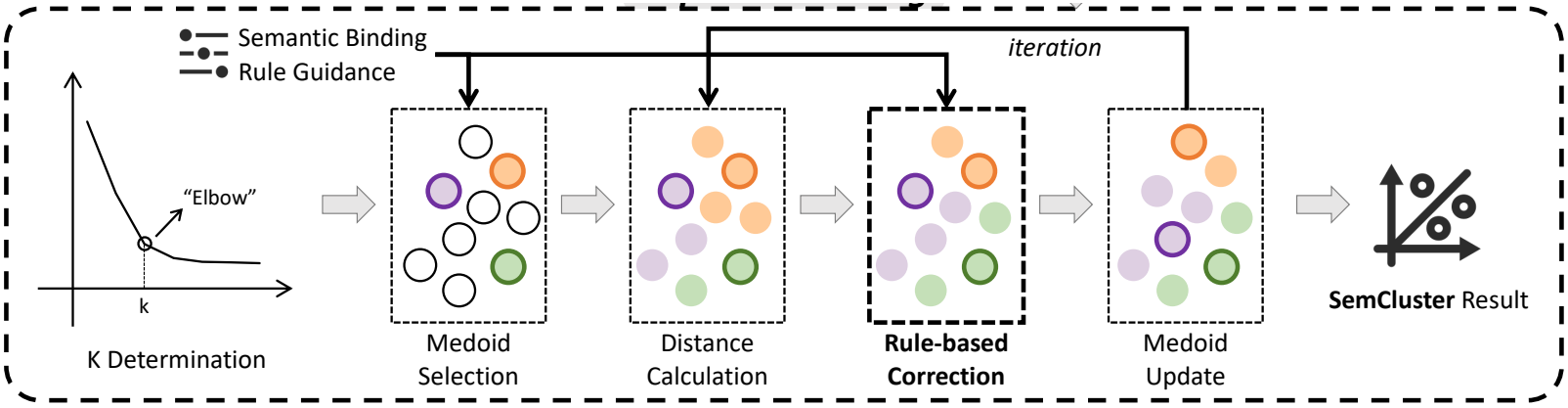


众测报告半监督聚类



- 图文语义绑定规则
  - Must-Link：$Dis(SF) < \alpha$ && $Dis(BB) < \theta$
  - Cannot-Link：$Dis(RS) > \omega$ && $Dis(CF) > \beta$

众测报告半监督聚类

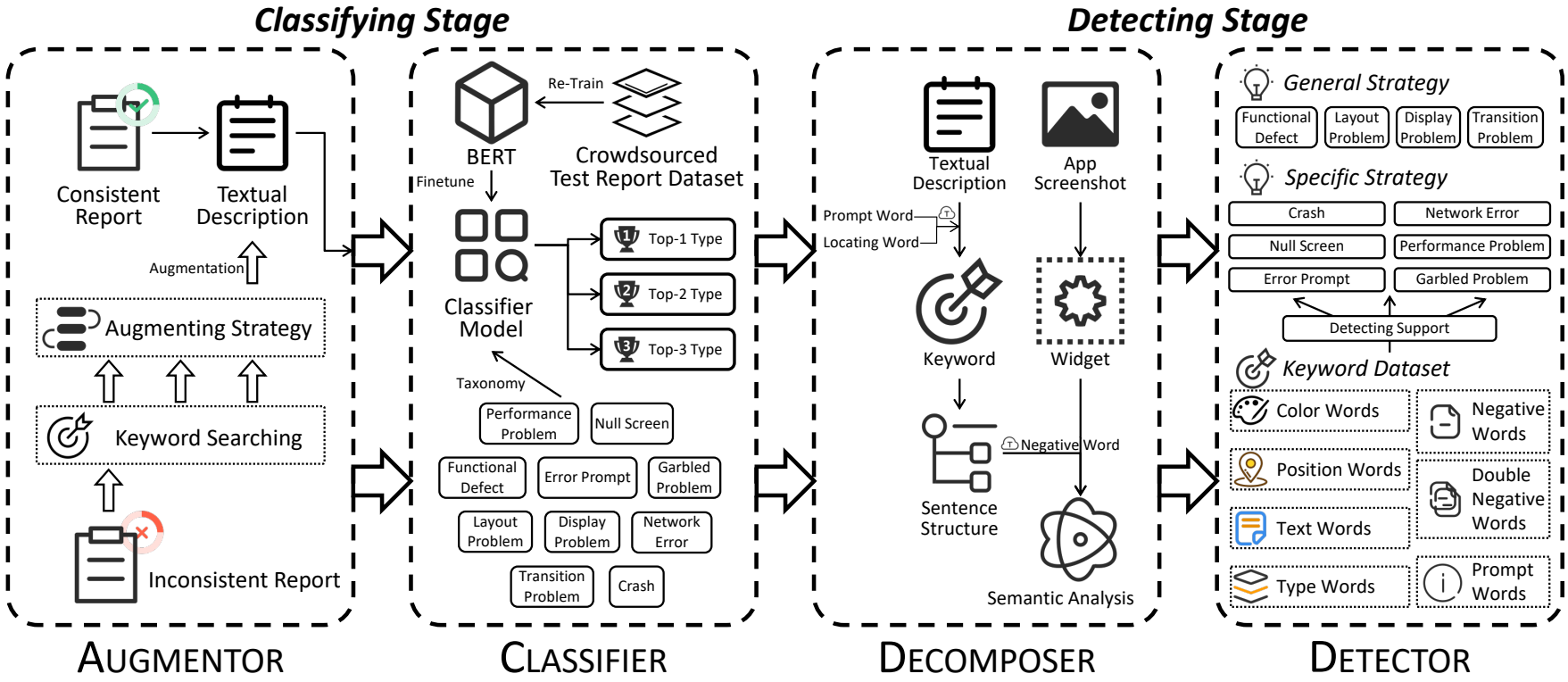


- 距离从小到大，将index与各类簇进行适配
  - Must-Link 且 不Cannot-Link：直接放入Must-Link距离最小类
  - 不Must-Link 且 Cannot-Link ：直接放入无Cannot-Link距离最小类
  - 不Must-Link 且 不Cannot-Link：按照无监督结果放入
  - Must-Link 且 Cannot-Link ：按照无监督结果放入

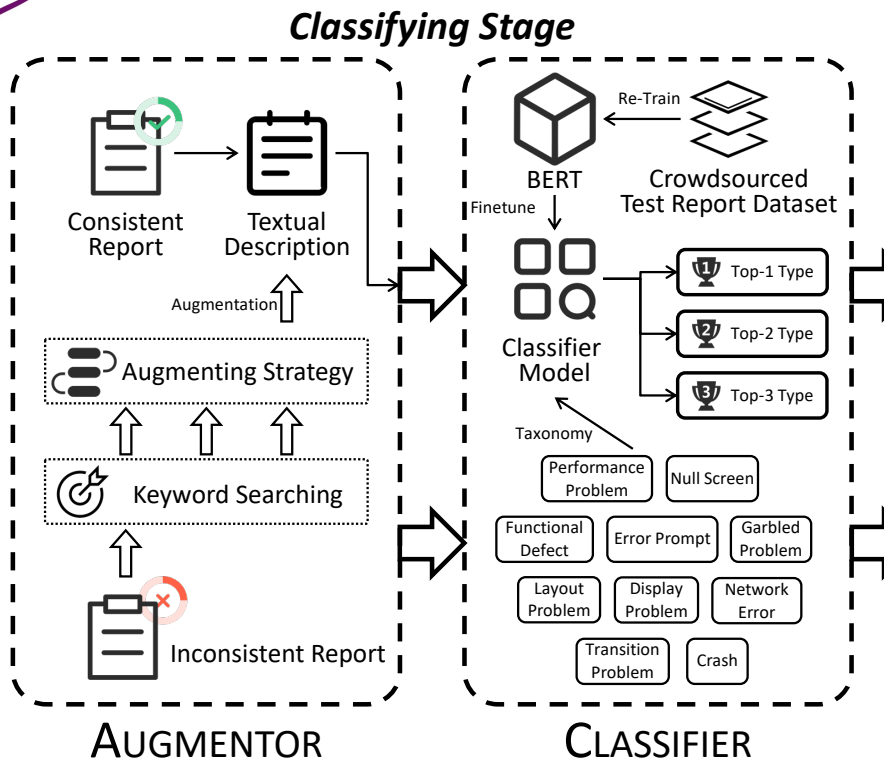
众测报告一致性检测

- 众测工人能力与专业性参差不齐
- 众测报告质量参差不齐
- 众测报告质量的底线——截图与文本的一致性
- 一阶段方法（多模态深度学习模型）的不足
- 二阶段方法——现分类后检测

众测报告一致性检测

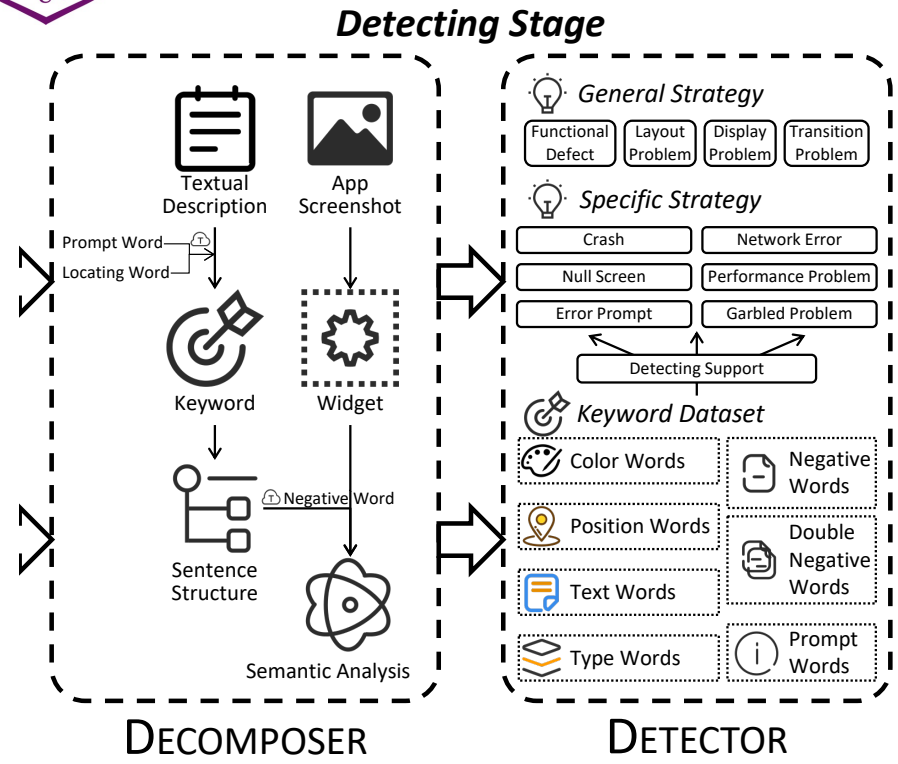

Classifying Stage
Detecting Stage
Consistent Report
Textual Description
Augmentation
Augmenting Strategy
Keyword Searching
Inconsistent Report
Re-Train
BERT
Crowdsourced Test Report Dataset
Finetune
Classifier Model
Top-1 Type
Top-2 Type
Top-3 Type
Taxonomy
Performance Problem
Null Screen
Functional Defect
Error Prompt
Garbled Problem
Layout Problem
Display Problem
Network Error
Transition Problem
Crash
Textual Description
App Screenshot
Prompt Word
Locating Word
Keyword
Widget
Negative Word
Sentence Structure
Semantic Analysis
General Strategy
Functional Defect
Layout Problem
Display Problem
Transition Problem
Specific Strategy
Crash
Network Error
Null Screen
Performance Problem
Error Prompt
Garbled Problem
Detecting Support
Keyword Dataset
Color Words
Negative Words
Position Words
Double Negative Words
Text Words
Type Words
Prompt Words
AUGMENTOR
CLASSIFIER
DECOMPOSER
DETECTOR

众测报告一致性检测



- 基于深度学习模型的文本分类
- 众测报告包含缺陷的不确定性
  - Top-3缺陷类型
- 不平衡数据数量
  - 基于关键词替换的文本扩增

众测报告一致性检测



- 图像与文本的分解
  - 颜色、位置、文本、类型
- 检测策略
  - 通用策略
  - 特定策略

zychen@nju.edu.cn
fangchunrong@nju.edu.cn

Thank you!