



南京大学

研究生毕业论文 (申请工学硕士学位)

论文题目 基于句子分析的众包测试报告质量度量方法

作者姓名 张欢

专业方向 软件工程领域

研究方向 软件工程

指导教师 陈振宇 教授

2022 年 5 月 23 日

学 号：MG1932017

论文答辩日期：2022 年 05 月 20 日

指 导 教 师： (签字)

Automated Quality Assessment for Crowdsourced Test Reports Based on Dependency Parsing

by

Huan Zhang

Supervised by

Professor Zhenyu Chen

A dissertation submitted to
the graduate school of Nanjing University
in partial fulfilment of the requirements for the degree of

MASTER

in

Software Engineering



Software Institute
Nanjing University

May 23, 2022

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目： 基于句子分析的众包测试报告质量度量方法

软件工程领域 专业 2019 级硕士生姓名： 张欢
指导教师（姓名、职称）： 陈振宇 教授

摘 要

在众包测试活动中，测试工人需要在规定的时间内以众包测试报告的形式提交发现待测软件的异常现象，众测平台中的审核人员将进行质量审核和缺陷验证。同时，众包测试人员大多为非专业测试人员，且需要在规定时间内完成任务，其提交的众包测试报告质量参差不齐。对于众测平台的审核人员，众包测试报告的质量越高，内容更加易于理解，可以帮助审核人员更好完成 bug 审核。反之，众包测试报告的质量越低，往往缺乏更多 bug 描述细节，耗费了审核人员的时间和精力，一定程度影响工作的效率。对于诚信的测试工人而言，他们提交的测试报告质量越高，其审核通过率越高相应获得报酬也越多。为了获得更多的报酬，他们也渴望对高质量众包测试报告进行了解和学习。因此，自动化评估众包测试报告质量对提升众包测试效益具有积极意义。

针对众包测试报告是专业领域为软件测试且具有稳定结构的短文本特点，本文提出一种基于句子分析的众包测试报告质量度量方法。本文分析归纳了众包测试报告的期望属性和可量化的质量指标，创新性地提出基于句子分析的分析性度量指标以更好的度量众包测试报告质量。本文使用规则归纳来实现质量指标。特别地，利用句子分析技术，如语法依存分析、语义依存分析对众包测试报告文本内容进行句子语法依存关系分析、句子语义依存分析和附加语义角色标注，协助完成分析性规则构建。提出基于句子分析的众包测试报告质量评估算法用于众包测试报告质量评估。

在 JayMe、SLife、途牛旅行、雅思听力和沪江英语 5 个 APP 的真实众包测试报告数据集上，进行了关于有效性和准确性的质量评估验证实验，证明了本文提出的基于句子分析的众包测试报告质量度量方法能有效度量众包测试报告中的质量问题，且质量评估准确率高达 88%。

关键词： 众包测试，测试报告质量，期望属性，质量指标，句子依存分析

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Automated Quality Assessment for Crowdsourced Test
Reports Based on Dependency Parsing
SPECIALIZATION: Software Engineering
POSTGRADUATE: Huan Zhang
MENTOR: Professor Zhenyu Chen

Abstract

In crowdsourced testing, testing workers from the open platform submit crowdsourced test reports for abnormal phenomena within a specified time frame. One of the most important characteristics is that crowdsourced testing is strictly limited in time, such as several days or one week. Thousands of test reports are sent to auditors in a short time and severely exceeds the available resources to inspect them. At the same time, since most of the crowdsourced testers are non-professional testers, the quality of crowdsourced test reports may vary greatly. For auditors in the platform, high-quality test reports provide overall information and contain content that can be easily understood, which can help them better inspect test reports. In contrast, Low-quality test reports often lack important details and consume the time and effort of them, thus seriously reducing their efficiency. For integrity testing workers, the higher the quality of the test reports they submit and the higher the audit pass rate, then the more rewards they get. In order to get more rewards, they need to understand and learn about high-quality crowdsourced test reports. Therefore, it is important to automate the assessment of crowdsourced test report quality to enhance the benefits of crowdsourced testing.

For the crowdsourced test reports in the professional field of software testing, it is characterized by short text description with stable structure. In this paper, we propose an automated quality assessment for crowdsourced test reports based on Sentence Analysis. We summarize the desirable properties and measurable quality indicators of crowdsourced test reports, and innovatively propose analytical indicators based on sentence analysis to better determine the quality of crowdsourced test reports. We use rule induction in machine learning methods to implement quality indicators. In particular,

sentence analysis techniques, such as dependency parsing and semantic dependency parsing, are used to perform part-of-speech tagging, dependency parsing tagging, additional semantic role labeling, and semantic dependency tagging on the text content of crowdsourced test reports to assist in the construction of analytical rules. Meanwhile, a quality evaluation algorithm based on sentence analysis is proposed for crowdsourced test report quality evaluation.

We conducted two quality evaluation and verification experiments on real crowdsourced test report data sets which is collected from JayMe, SLife, TuNiu, IELTS Listening and Hujiang apps. Our proposed method can effectively judge the quality problems in test reports and correctly predict the quality of test reports with accuracy of up to 88%.

keywords: Crowdsourced testing, Test report quality, Desirable properties, Quality indicators, Dependency Parsing

目 录

目 录	v
插图清单	vii
附表清单	ix
第一章 绪论	1
1.1 研究背景和意义	1
1.2 研究现状	2
1.2.1 众包测试研究现状	2
1.2.2 众包测试报告研究现状	4
1.3 本文主要贡献	6
1.4 本文组织结构	8
第二章 相关工作	9
2.1 测试报告质量	9
2.2 需求规格说明书质量评估	10
2.3 自然语言句子分析	12
2.4 本章小结	15
第三章 背景知识	17
3.1 众包测试报告实例	17
3.2 语法依存分析	20
3.3 语义依存分析	22
3.4 本章小结	23
第四章 测试报告质量度量技术	25
4.1 质量评估框架	25
4.2 期望属性	27
4.3 度量指标	30
4.3.1 形态学指标	31
4.3.2 相关性指标	32
4.3.3 分析性指标	33

4.3.4 度量指标与期望属性关系	35
4.4 规则构建	36
4.4.1 形态学规则	36
4.4.2 相关性规则	38
4.4.3 分析性规则	40
4.4.4 规则与期望属性关系	44
4.5 质量得分设计	45
4.6 本章小结	49
第五章 实验设计与分析	51
5.1 研究问题	51
5.2 实验设置	52
5.2.1 实验数据集	52
5.2.2 实验方法	54
5.2.3 实验环境	54
5.2.4 实验评估	55
5.3 实验结果分析	56
5.3.1 CTRQSDP 评估结果实例分析	56
5.3.2 实验一：CTRQSDP 误差评估	60
5.3.3 实验二：质量指标有效性评估	65
5.3.4 实验三：CTRQSDP 准确性评估	67
5.4 有效性分析	69
5.4.1 构建有效性	69
5.4.2 内部有效性	70
5.4.3 外部有效性	70
5.5 本章小结	70
第六章 总结和展望	71
6.1 总结	71
6.2 展望	72
致 谢	73
参考文献	75
简历与科研成果	83

插图清单

1-1 众包测试流程	3
2-1 需求规格说明书质量期望属性间的关联	10
2-2 自然语言处理技术发展	13
2-3 自然语言处理流程	14
3-1 语法依存树实例	21
3-2 联合语义角色标注的语义依存树实例	23
4-1 基于句子分析的众包测试报告质量评估框架	26
4-2 众包测试报告质量期望属性间的关联	27
4-3 测试报告句子语法分析实例	41
4-4 测试报告句子语义分析实例	42
4-5 众包测试报告	47
5-1 众包测试报告数据集 bug 类型比例图	53
5-2 CTRQSDP 评估实例-优秀	57
5-3 CTRQSDP 评估实例-良好	58
5-4 CTRQSDP 评估实例-一般	59
5-5 不同误差等级测试报告数量	61
5-6 CTRQSDP 评分与人工评分平均误差	62
5-7 CTRQSDP 和人工标注拟合分布图-偏低	63
5-8 CTRQSDP 和人工标注拟合分布图-偏高	63
5-9 CTRQSDP 和人工标注拟合分布图-贴合	64
5-10 违反规则的报告数量	66
5-11 违反三大类规则的报告数量	67
5-12 CTRQSDP 准确度	68
5-13 不同 bug 类型检测成功数量	69

附表清单

2-1	需求规格说明书期望属性	11
3-1	众包测试报告实例	18
3-2	语法依存关系标注描述	20
3-3	附加语义角色标记	22
4-1	众包测试报告期望属性	28
4-2	众包测试报告文本内容组成	29
4-3	期望属性度量指标详情	31
4-4	众包测试报告高频专业词汇表	34
4-5	度量指标与期望属性关系	35
4-6	规则总结	37
4-7	分析性规则判断举例	44
4-8	规则与期望属性的关系	45
4-9	规则得分详情	46
5-1	五个 APP 众包测试报告数据集	52
5-2	实验数据集	54
5-3	实验环境	55
5-4	APP 得分误差结果	62

第一章 绪论

1.1 研究背景和意义

移动互联网时代,移动应用软件推陈出新更新迭代速度加快,软件产品生命周期缩短,对于软件测试的需求也日趋强烈。移动应用软件在测试过程中期望能够快速获得大量的测试信息反馈,尽快的修复缺陷完成快速迭代以满足市场需求,特别是短时间内获取大量用户体验测试信息反馈,传统软件测试方法不能很好的胜任。如何较低成本的获取足够多的测试反馈,高效的完成测试任务引起了广泛讨论。

众包是互联网带来的一种分布式问题解决和生产组织模式 [1]。众包被广泛应用于软件工程的各个环节 [2],应用于软件测试构建了新的研究领域即众包测试。众包测试能够提供软件使用真实场景和用户真实体验的良好模拟。众包测试平台中大量的在线测试工人通过众包测试报告的形式,反馈他们测试软件时发现的缺陷或异常,同时他们也将反馈他们的使用体验感受。发开者能够利用众包测试获取大量真实的反馈信息,包括功能缺陷和用户体验等 [3, 4]。众包测试能够在较短周期内,以较低的成本获取大量的关于软件缺陷和用户体验等方面的信息反馈,获得了学术界和工业界广泛关注,学术界中研究人员密切关注众包测试的理论和方法,工业界中涌现了众多的众包测试商业平台,例如 uTest^①、Bugcrowd^②、Testin^③、百度众测^④、阿里众测^⑤等。

众包测试活动中,一个待测应用软件往往有着测试时间限制,比如几天或者一周。在这期间,开放平台的工人们参与测试,成千上万的测试报告被提交至众测平台。其数量严重超过了众测平台审核人员审核这些报告的可用资源。同时,因为众包测试者们多数都是非专业测试者,因此众包检查报告质量也可能会有较大的差别。一个完整的众包测试报告通常提供一些关键信息,如环境信息、复现步骤、bug 描述、截图。对于平台中数量庞大的待审众包测试报告,

^①<https://www.utest.com>

^②<https://www.bugcrowd.com/>

^③<http://www.testin.io>

^④<http://test.baidu.com>

^⑤<https://mqc.aliyun.com/crowdtest/index.htm>

许多研究集中在通过减少检测的测试报告数量来缩短审核成本 [5–11]。然而，这些研究忽略了众包测试报告质量的影响。

现有关于众包测试报告质量的研究，主要是通过特定词汇进行众包测试测试报告质量度量 [12, 13]。根据分词建立众包测试报告的词向量，利用分类模型将众包测试报告分为高质量测试报告和低质量测试报告。但是词汇维度的度量有着较大的局限性，存在文本语义度量不足和文本语法度量不到位的问题。因此，众包测试报告的质量问题仍值得本文关注。

工业界中，为了获得更多高质量的众包测试报告。众测平台通常会对于众包测试报告的规范性做出一定的引导，例如建议测试工人在复现步骤描述中进行分步骤描述测试步骤。规范性也是众包测试报告质量中重要组成部分。众包测试报告质量评估对于众测平台和测试工人是双赢的。

对于众测平台的审核人员，在有限的资源内资源内选择一份测试报告进行审核。众包测试报告的质量越高，内容更加完整清晰易于理解，可以帮助审核人员更好完成 bug 审核。反之，众包测试报告的质量越低，往往缺乏更多 bug 描述细节，耗费了审核人员的时间和精力，一定程度影响工作的效率。

对于诚信的测试工人而言，他们提交的测试报告质量越高，其审核通过率越高相应获得报酬也越多。为了获得更多的报酬，他们也渴望对高质量众包测试报告进行一定程度的了解和学习。众包测试报告质量评分能够辅助激励机制提升测试工人积极性，例如根据众包测试报告质量评分给与测试工人一定的奖励，激励其编写更高质量的测试报告。因此，自动化评估众包测试报告质量，以分数形式对众包测试报告的质量进行度量显得极为重要。

1.2 研究现状

1.2.1 众包测试研究现状

已有研究人员总结了关于众包在软件工程的应用，主要包括众包软件工程的应用模式和商业平台的相关研究 [14–21]。其中 Mao 等人 [21] 定义了众包软件工程为：“众包软件工程是指由大量潜在的、未定义的在线工人以公开召集的形式，承担外部软件工程任务的行为”。基于众包软件工程的定义，章等人 [22] 定义众包软件测试为：“所有支持软件测试的众包方法、技术、工具和平台都属于众包软件测试领域”。

众包软件测试主要参与者包括任务发起方 (requester)、众包工人 (worker)

和众包平台方 (platform)[2]。众包测试平台是为众包测试任务发起方和众包测试工人提供在线服务的第三方平台。众包测试任务发起者，通常是软件开发方，会在众包测试平台发布待测软件产品和相应的测试需求。众包测试工人会根据众测平台推荐选择自己擅长的任务去完成，需要在规定时间内以众包测试报告的形式反馈自己测试过程中发现的软件系统异常现象。一个完整的众包测试报告通常提供一些关键信息，如环境信息、复现步骤、bug 描述、截图。这些众包测试报告由众包测试平台质量审核人员进行测试报告审核，并整合描述正确且高质量的测试报告，以交付给任务发起者。对于众包测试工人，如果其提交的众包测试报告通过了审核，那么他将获得相应的报酬。众包测试流程图如图 1-1。

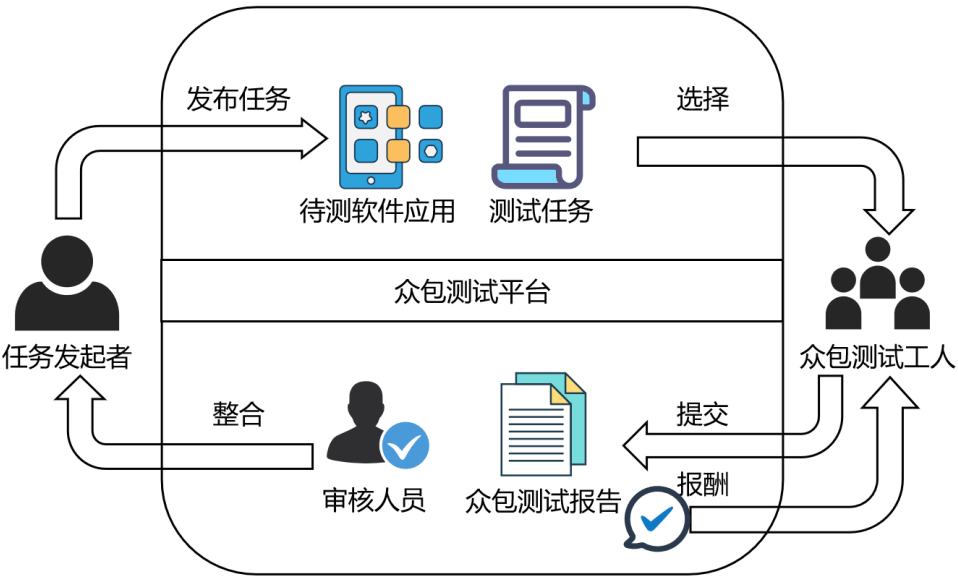


图 1-1: 众包测试流程

众包测试报告与传统测试相比，测试工人数量相对更多，但绝大多数是非专业的测试人员。许多众测平台会对平台测试工人开展众包测试基础知识基础培训，但是测试工人的专业素质还有待提升。众包平台机制有竞争模式和团队模式，但是工业界中大部分平台是竞争模式。在该模式下，众包测试工人彼此之间互为竞争对手，不沟通交流，所提交的测试报告通过审核后，将获得相应的奖励。一定程度上，他们提交的测试报告越多，获得的奖励报酬也越多。因此，众包测试人员想在规定时间内尽可能多的提交众包测试报告，往往忽略了测试报告质量问题。在较短周期内，审核人员需要审核数量庞大的众包测试报告，无疑面临较大困难。在审核阶段，许多研究集中在通过减少检测的众包测

试报告数量来缩短审核成本 [5–11]。部分研究通过预测众包测试报告质量，帮助审核人员在相似度高的测试报告集合中优先选择高质量测试报告进行审核 [7, 12, 13]。

众包测试报告质量评估对于众测平台和测试工人是双赢的。对于众测平台的审核人员，在有限的资源内资源内选择一份测试报告进行审核。高质量的测试报告提供了整体信息，所包含的内容很容易被理解，可以帮助审核人员更快更好完成 bug 审核。反之，低质量的测试报告往往缺乏重要细节，耗费了审核人员的时间和精力，一定程度影响工作的效率。对于诚信的测试工人而言，他们提交的测试报告质量越高，其审核通过率越高相应获得报酬也越多。为了获得更多的报酬，他们也渴望对高质量众包测试报告进行一定程度的了解和学习。

众包测试报告质量评分能够辅助激励机制提升测试工人积极性，例如根据众包测试报告质量评分给与测试工人一定的奖励，激励其编写更高质量的测试报告。因此，探索一种高效的众包测试报告质量得分评估方法具有重要意义。

1.2.2 众包测试报告研究现状

在众包测试活动中，众包测试工人会根据众测平台推荐选择自己擅长的任务去完成，需要在规定时间内以众包测试报告的形式反馈自己测试过程中发现的软件系统异常现象。众测平台的审核人员负责审核这些测试报告。

一个典型的众包测试报告通常由不同的字段组成，如环境、描述、截图信息，其中一些字段在不同的项目中可能略有不同。众包测试报告是内容描述简短的短文本，通常一份众包测试报告中只含有一个 bug 相关描述，此外还附带有屏幕截图信息和设备环境信息。众包测试通过群体智能的方式来改善传统测试中用户体验测试人力不足和自动化测试技术不足的问题，更佳实现人机协同，提高软件测试效率。如何人机协同高效自动化处理众包测试中产生的大量测试报告被广泛研究。针对众包测试报告的研究有众包测试报告相似度度量 [5, 23, 24]、众包测试报告分类 [11]、众包测试报告报告融合 [10]、众包测试报告质量评估 [12, 13, 25] 等。

目前研究者们已经提出了许多方法来检测众包测试报告的相似度。众包测试报告中不仅含有文本信息还含有截图信息，相似度判断包括以上两个部分。现有研究结合了文本和图片来进行众包测试报告的相似度检测。Feng 等人 [5] 创新性地提出了结合文本特征和图片特征的众包测试报告相似度度量方法。对

于文本信息和图片信息分别进行特征提取。在文本特征和图片特征的基础上建立各自的相似度矩阵，并按照公式将其合并为一个相似度矩阵，该矩阵即为这份众包测试报告的特征矩阵。利用测试报告的特征矩阵比较不同，计算众包测试报告相似度。

对数量庞大的众包测试报告进行审核和分类，已经成为了一种耗时但不可避免的任务。Li 等人 [11] 进行了基于图像特征的移动应用众包测试报告分类的实证研究。融合了文本特征和截图特征对众包测试报告进行分类，并研究了该特征融合方法在六种分类算法下的有效性，即 Naive Bayes (NB)、k-Nearest Neighbors (KNN)、Support Vector Machine (SVM)、Decision Tree (DT)、Random Forest (RF) 和 Convolutional Neural Network (CNN)。此外，还分析总结了众包测试报告常见类别，主要包括功能不完整、性能、页面布局、异常退出、安装失败、更新失败、用户体验、安全性、其他。

除了众包测试报告的相似度检测和分类，Hao 等人 [10] 提出了基于融合的众包测试报告处理方法。首先将众包测试报告按照文本特征和图片特征进行聚类，确保相似度高的众包测试报告在同一类簇中。其次是利用 PageRank 算法计算测试报告权重选择最高权重的众包测试报告作为该类簇的主报告。然后在这个类簇中，对于文本信息，以句子为最小单位，逐份将非主报告与主报告进行比较，找到差异点；对于图片信息则是比较图片相似度。最后对于每一个类簇，按照差异点和相似度，将非主报告的文本和图片融合进主报告，完成相似众包测试报告的融合。科技的不断发展应当是为人类服务的，不断的提高人工效率，实现人机协同是合理的也是必要的。自动化整合测试报告，能够更好的协助审核人员，提高其工作效率。相同地，众包测试报告质量的自动化评估也是一项有意义的研究，自动化的进行众包测试报告的质量评估，进而更好的帮助审核人员进行测试报告质量审核以及缺陷复现。

针对众包测试报告的整合和处理研究还有一个重要的研究方向即众包测试报告质量评估。陈等人 [12] 首次地提出了移动应用众包测试报告的自动质量评估方法。首先是在现有缺陷报告质量研究基础上总结归纳了众包测试报告的期望属性。受需求规格说明书质量评估度量指标启示，从分类学角度，构建了众包测试报告质量评估的期望属性相应的度量指标。其中的文本分析性指标主要是从词汇维度进行度量。其次利用分词建立众包测试报告的词向量，利用分类模型将众包测试报告分为高质量众包测试报告和低质量众包测试报告。但是从词汇角度出发有着较大的局限性，对于众包测试报告的文本信息的语义和语法

度量不足。本文提出的基于句子分析的众包测试报告质量度量方法能更好度量文本语义和语法。

在众包测试审核阶段实际应用中，往往是结合上述各部分研究内容，以此高效地处理数量庞大的众包测试报告。例如首先对众包测试报告进行相似性度量按照相似度分成不同集合，或者利用分类分成不同集合。然后再在同一个集合中优先选择高质量的测试报告进行审核判定，以此加快类似报告的审核速度。又或则是在一个集合中进行报告的融合加快审核进度。不论是众包测试报告的相似度检测、还是分类或者融合均是试图通过减少众包测试报告的数量来缩减审核成本，并没有涉及到众包测试报告的质量问题。众包测试报告的质量评估对众测平台和测试工人都具有重要作用。众测平台为了获取更多高质量的众包测试报告，需要设置高质量奖励来激励测试工人编写更加优秀的测试报告。对于测试工人，他们也愿意学习更多相关知识来获取更加丰厚的报酬。因此，自动化评估众包测试报告质量，以分数形式对众包测试报告的质量进行度量显得极为重要。

1.3 本文主要贡献

本文所设计和实现的基于句子分析的众包测试报告质量评估方法是用于众包测试平台审核阶段，针对众包测试报告是专业领域为软件测试且具有稳定结构的短文本特点，自动化的进行众包测试报告质量评估得分。这不仅能帮助审核者在有限的资源内预测选择哪一份测试报告进行检查，同时还能够辅助激励机制提升测试工人积极性。本文提出基于句子分析的众包测试报告质量评估方法：CTRQSDP（Automated Quality Assessment for Crowdsourced Test Reports Based on Dependency Parsing）。其具体内容如下：

首先本文在现有研究基础上从软件技术文档质量基本属性“精确性”和“原子性”出发，运用分类学方法分析众测平台审核人员和测试工人各自对众包测试报告质量期望，分析并归纳了众包测试报告的期望属性。众包测试报告期望属性的度量依赖于完备且具体量化的度量指标。不同于之前研究采用的词汇维度，即将众包测试报告文本内容进行分词，统计词频等，文本从句子维度出发，分析句子中各个成分的构成以及相关信息描述是否符合众包测试报告期望属性来评估众包测试报告质量。从词汇维度到句子维度的转变使得针对文本内容的度量指标也需要重新构建。为了更好的度量期望属性，本文创新性地提

出基于句子分析的众包测试报告质量度量新指标。

其次是将句子依存关系分析引入众包测试报告质量评估。利用自然语言处理中的句子依存关系分析对众包测试报告进行预处理。在众包测试报告质量评估中，对于测试报告文本内容部分采用的以句子为单位的自然语言分析。特别地，利用句子分析技术，如语法依存分析、语义依存分析对众包测试报告文本内容进行句子语法依存关系分析、句子语义依存分析和附加语义角色标注，更加有效的度量众包测试报告的文本语法和语义。

最后是在此基础上，提出基于句子分析的众包测试报告质量评估算法，用来预测测试报告的质量得分。本文使用机器学习方法中的规则归纳来实现度量指标。规则判断具有可解释性、可扩展性以及易维护性等特点。针对众包测试报告的特点，即众包测试报告是专业领域为软件测试且具有稳定文本结构的短文本，基于规则判断的众包测试报告度量有以下三个优势：首先是众包测试报告的专业领域强，其专业词表为软件测试领域，这使得规则制定空间范围小。其次众包测试报告文本结构稳定。一个完整的众包测试报告通常提供一些关键信息，如环境信息、复现步骤、bug 描述、截图。其中通常采用分点描述复现步骤和 bug 描述，且使用句子形式较为规范，规则判断可用性高。最后众包测试报告是短文本，使用规则判断速度快。

为了评估 CTRQSDP 的有效性和准确性，本文从慕测平台收集工业界真实的众包测试报告数据集，审核人员花了一周的时间来审核和评估这些测试报告。本文采用常用的准确度作为度量，并开展了三个验证实验，以评估 CTRQSDP 在众包测试报告质量评估中的有效性和准确性。实验结果表明 CTRQSDP 能有效判断测试报告中的质量问题，且质量评分准确率高达 88%。同时，结果还表明，本文提出的基于句子分析的分析性指标对 CTRQSDP 在众包测试报告质量评估中具有积极影响。

本文的主要贡献如下：

- (1) 本文从句子维度出发，创新性提出了基于句子分析的众包测试报告质量度量新指标，将句子分析引入众包测试报告质量评估。
- (2) 为了自动化度量众包测试报告的质量得分，本文提出了一个名为 CTRQSDP 的新框架，使用基于句子分析的规则判断来衡量众包测试报告的质量。
- (3) 本文在 5 个 APP 的真实众包测试报告数据集上进行关于有效性和准确性的质量评估验证实验，证明了 CTRQSDP 能有效且准确的预测众包测试报告的质量得分。

1.4 本文组织结构

本文的组织架构如下所示。

第一章 绪论。介绍了与众包测试报告质量度量相关的背景与研究意义，分析了众包测试报告领域的现有成果，特别是与质量相关的研究，说明了本文的重点工作和突出贡献。

第二章 相关工作。主要讨论了本文众包测试报告质量评估中涉及的相关工作，包括现有传统缺陷报告质量评判标准、需求规格说明质量评估中采用的分类学方法和自然语言句子依存分析方法理论以及现有常用工具。

第三章 背景知识。本章主要阐述了本文众包测试报告质量评估中涉及的相关技术，主要包括有缺陷报告质量标准、需求规格说明质量评估方法和自然语言句子语法分析方法。

第四章 测试报告质量度量技术。是本文的重点章节，介绍了众包测试报告质量评估框架 CTRQSDP 的细节，第一部分是众包测试报告的期望属性和期望属性具体的度量指标。第二部分是在度量指标基础上，利用自然语言处理的句子分析预处理测试报告完成规则库构建。最后部分是众包测试报告质量规则得分设计，用于众包测试报告的质量评估得分。

第五章 实验设计和分析。以第四章的技术描述为基础，介绍了本文的实验设计与分析部分。本章节通过设定实验目的，利用 5 个应用程序数据集关于有效性和准确性进行了质量评估验证实验，并对实验结果进行了深入分析。

第六章 总结与展望。首先总结了本文研究工作，主要包括众包测试报告质量和期望属性调研分析、度量指标的定义和使用、质量评估中规则库构建以及质量评估得分算法设计等。最后，本章进一步分析了基于句子分析的众包测试报告质量度量方法可后续改进的功能。

第二章 相关工作

2.1 测试报告质量

对于软件产品的软件测试中，主要的产出物是测试报告，即缺陷报告。测试报告是开发人员改进软件产品最重要的凭借，是整个软件开发过程重要的文档产出 [26, 27]。高质量的缺陷报告能够帮助开发人员快速有效的定位和理解缺陷；反之低质量的缺陷报告可能对开发人员产生误导，会浪费其时间和精力，极大的影响软件产品的维护过程。

关于缺陷报告的质量问题，研究者进行了一系列的实证研究 [28, 29]。调查研究发现用户提供的反馈信息质量普遍不高，不能很好的满足开发者的需要。开发者与用户之间关于一些词汇存在着极大的理解偏差，导致了沟通成本增加，用户提供的信息通常不能直接帮助开发者完成软件维护。研究者们通过分析和定义缺陷报告的期望属性来评估缺陷报告的质量 [29, 30]。Zimmermann 等人 [28] 研究分析了缺陷报告的期望属性，并开发了缺陷报告评估工具，用于自动化的评估缺陷报告质量。

软件产品的使用用户通常是相互独立的，他们会提交自己发现的软件缺陷。如果这些缺陷是出现频率较高的缺陷，那么他们提交的缺陷报告往往具有很高的相似性。Bettenburg 等人 [8] 开展了关于大量重复缺陷报告的实证研究。调查研究发现在大量重复的缺陷报告相互之间有可能存在部分互补的信息，这些互补的信息能够帮助开发者从不同的角度来理解同一个缺陷。此外他们还采用了可量化的方法表示大量重复的测试报告中可能存在的额外信息反馈。

基于大量重复缺陷报告中存在的额外信息反馈实证研究结果。Hao 等人 [10] 提出了基于融合的众包测试报告处理方法。首先将众包测试报告按照文本特征和图片特征进行聚类，确保相似度高的众包测试报告在同一类簇中。其次是利用 PageRank 算法计算测试报告权重选择最高权重的众包测试报告作为该类簇的主报告。然后在这个类簇中，对于文本信息，以句子为最小单位，逐份将非主报告与主报告进行比较，找到差异点；对于图片信息则是比较图片相似度。最后对于每一个类簇，按照差异点和相似度，将非主报告的文本和图片融

合进主报告，完成相似众包测试报告的融合。

与上述尝试将重复缺陷报告进行融合不同，Hooimeijer 等人 [31] 提出了一个信息摘要框架，通过抽取缺陷报告的公共表面特征来衡量缺陷报告的质量。缺陷报告的主体部分是文本描述且内容描述较长，Rastkar 等人 [9] 则采用了文本信息摘取技术，自动化进行缺陷报告信息摘要提取，用于辅助开发者进行缺陷报告审核。借用该工具，开发者可以读取该缺陷报告的摘要信息借此判断缺陷报告内容，而不需要完整阅读缺陷报告。

2.2 需求规格说明书质量评估

诚然，质量是一个模糊的概念，与不同的评价标准有关 [32]。不同的开发者通常对同一份文档质量有着不同看法。软件工程各个阶段都有着重要技术文档的生成。需求工程中，最为重要的专业技术是需求规格说明书，现有众多针对需求规格说明书质量的相关研究 [33–35]。此外，研究者们也尝试开发自动化工具来评估需求规格说明书的质量 [36]。

特别地，从软件技术文档质量基本属性“精确性”和“原子性”出发，运用分类学方法，Génova 等人 [32] 分析归纳了需求规格说明书的期望属性，包括有效性、可验证性、完整性、可修改性、一致性、可理解性、明确性、可追踪性、简洁性、精确性、原子性、确认和验证。详细内容如表 2-1。

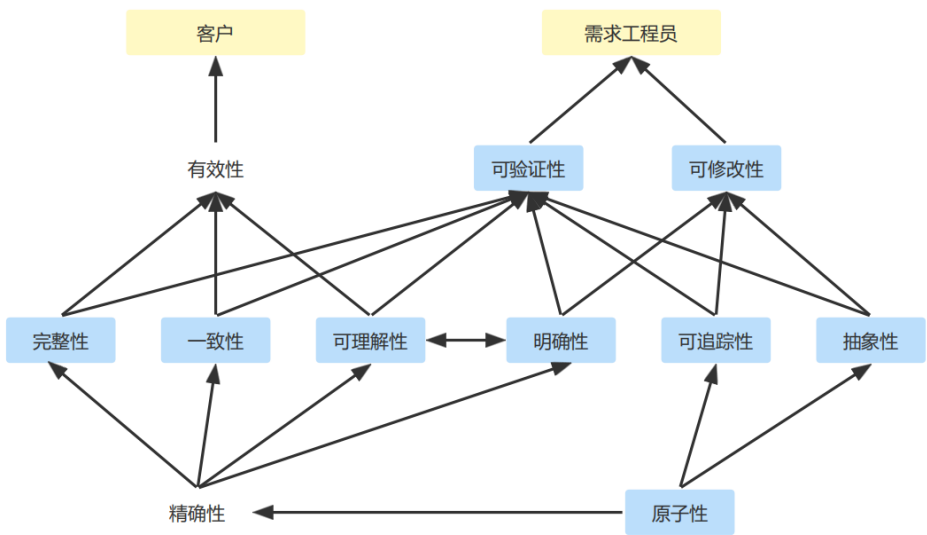


图 2-1: 需求规格说明书质量期望属性间的关联

其中，需求规格说明书质量评估期望属性之间关系如图 2-1。从客户

表 2-1: 需求规格说明书期望属性

期望属性	描述
有效性 (Validability)	客户必须能够确认需求有效地表达了满足其需求的系统。
可验证性 (Verifiability)	工程师必须能够检查生产的系统是否符合指定的系统。
完整性 (Completeness)	所有需求都包含在一组需求中。
可修改性 (Modifiability)	要求必须是可修改的，以方便系统修改以进行维护。
一致性 (Consistency)	需求之间不存在矛盾。
可理解性 (Understandability)	要求被正确理解，没有困难。
明确性 (Unambiguity)	每个要求只有一种解释。
可追踪性 (Traceability)	每个需求与设计、实现和测试工件之间存在明确的关系。
抽象性 (Abstraction)	需求说明了应用程序必须做什么，而没有说明它必须如何做，即在需求的规范中必须避免过多的关于实现的技术细节。
精确性 (Precision)	所有使用的术语都是具体且定义明确的。
原子性 (Atomicity)	每个需求都有明确的确定和标识，不与其他需求混合。

(Client) 的角度来看，需求规格说明书的最终质量是有效性 (Validability)，即客户必须能够确认需求有效地表达了满足其需求的系统。在第二个层次上，这个属性可以进一步展开为其他三个属性：完整性 (Completeness)、一致性 (Consistency) 和可理解性 (Understandability)。相反，从需求工程师 (Requirement Engineer) 的角度来看，本文可以说两个基本品质是：可验证性 (Verifiability)，即可以检查生产的系统是否符合指定的系统；和可修改性 (Modifiability)，因为如果很难修改需求，那么在生命周期的维护阶段，生成的系统也同样难以修改。可验证性取决于本文之前提到的相同的三个二阶属性（完整性、一致性和可理解性），以及其他三个属性：明确性 (Unambiguity)、可追溯性 (Traceability) 和简洁性 (Abstraction)；可修改性也密切依赖于最后三个属性。明确性和可理解性是相互关联的（它们甚至是同一个属性），因为如果一个需求是不明确的，那么它就不能被正确理解。特别地，抽象性与技术细节没有必然联系；非技术细节的存在并不影响抽象性。

因此，抽象性和明确性不是对立的观念：需求必须在规范的抽象级别（“什么”）具有单一解释，并且必须避免有关实现的技术细节（“如何”）；从抽象的角度来看，有几个可能的实现并不会使需求变得模棱两可。非技术细节的存在当然可以提高明确性并有助于可验证性和可修改性；相反，技术细节的存在（缺乏简洁性）使得需求过度依赖于实现，因此更难以验证和修改。

可以在以上期望属性分类法中区分出第三个更基本的层次，它将包含两个额外的、甚至更基本的属性：精确性（Precision）和原子性（Atomicity）。完整性、一致性、可理解性和明确性取决于精确性：更精确的语言有助于编写更完整、一致、可理解和明确的需求。另一方面，原子性影响精确性、抽象性（非原子性需求有过多细节的风险），尤其是可追溯性，从而影响生命周期维护阶段的可修改性。

受现有围绕错误报告和需求规格说明书的质量评估研究启发，本文将采用通过定义一系列可量化的指标，根据文本内容来衡量众包测试报告的质量。本文将从众包测试工人和众包测试审核人员两个角度出发分析众包测试报告的质量需求，同时从软件技术文档质量基本属性“精确性”和“原子性”出发，运用分类学方法分析众测平台方和测试工人各自对众包测试报告质量期望，分析和归纳众包测试报告的期望属性。

2.3 自然语言句子分析

自然语言处理（Natural Language Processing, NLP）研究人与计算机使用自然语言进行通信的技术。[37]。自然语言处理至今已有几十年历史了。

二十世纪 50 年代至二十世纪 70 年代初，大家更关心的是人类学习语言的认知研究上。这时被广泛使用的是基于规则推理的符号主义方法，同时人工智能（AI）也开始萌芽。二十世纪 80 年代，主要是基于统计的联结主义方法，此时符号主义和联结主义孰优孰劣也被人们广泛讨论，直到二十世纪 90 年代，随着机器学习在语音识别系统和翻译系统上取得关键性突破，基于统计的联结主义方法的走进大家视野。但是基于符号主义和联结主义的两种方法各有千秋，谁也没法完全取代另一方，充分利用和融合两者的优势也许是更好的选择。近些年，随着深度学习的发展，自然语言的研究再次成为一个热点。图 2-2 展示了近 70 年的主要发展，很多技术的发展都是互相促进的，比如机器学习的发展会影响到自然语言处理的发展。

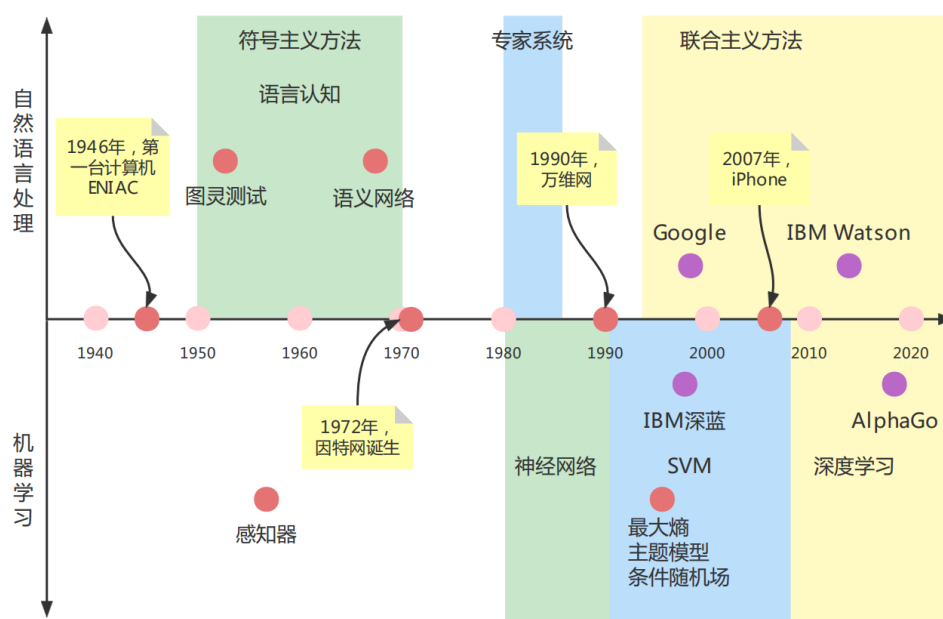


图 2-2: 自然语言处理技术发展

理想的自然语言处理流程如图2-3所示，这是一个层次结构。第一层是词汇分析，包括分词（Word Segmentation）、词性标注（Part-of-speech Tagging）等。第二层为句子语法分析。句法分析主要目的是揭示句子的内部语法成分和各个成分之间的关系，并依赖特定的语法来细化句法和语义的输出结构。第三个层面是句子语义分析。目的是理解句子表达的真实语义，语义分析通常都在语法分析的基础上完成。最后一层是应用，比如机器翻译（Machine Translation, MT）、问答（Question Answering, QA）、文本挖掘（Text Mining, TM）、和信息检索（Information Retrieval, IR）等各种任务。

句子分析（Parsing）是自然语言处理中一个重要的课题，其主要目的是揭示句子的内部结构以及结构间的关系 [38, 39]。从语言学的角度看，句子分析的目标是揭示词语是如何组成句子的，以及构成句子时的规则，即句子语法的表示。从NLP的角度来看，句子分析是对文本的基础处理，对许多任务都很有利，如机器翻译、问题回答、信息提取、情感分析和生成 [40, 41]。

句子分析中最为广泛研究的，主要是成分分析（Constituent Parsing）和依存分析（Dependency Parsing）。成分分析主要针对句子句法分析（Syntactic Analysis），而依存分析不仅可以进行句法还能进行语义分析（Semantic Analysis）。自句子分析研究问题诞生以来，研究者们对句子分子进行了广泛的研究。句法分析的主要目标是获得句子中的句法信息，比如主语、宾语和修饰语

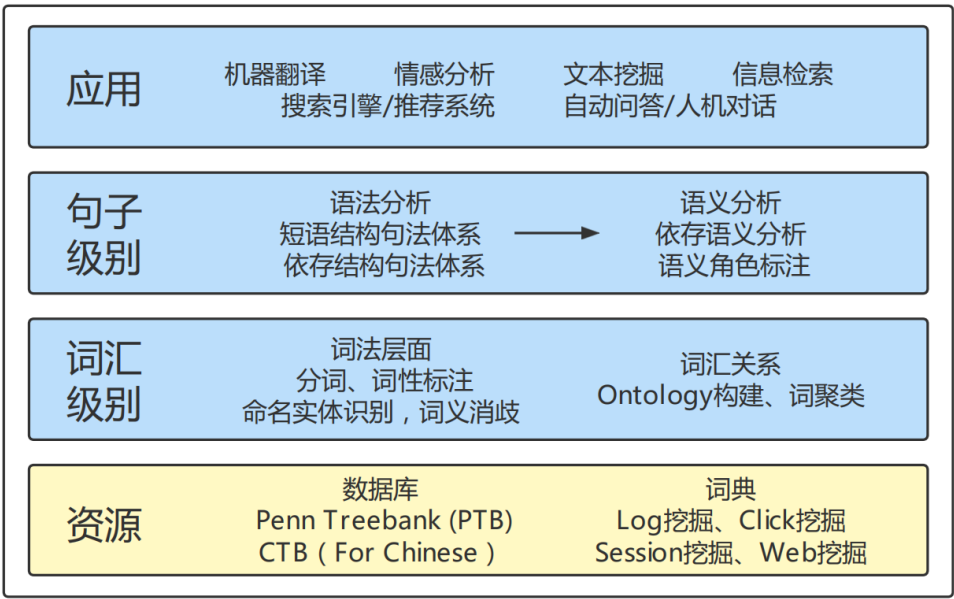


图 2-3: 自然语言处理流程

等。语义分析由于各种语义的复杂结构，如谓语、宾语等。语义分析与语法分析相比要困难得多。

句子分析通常依赖于特定的语法，这些语法被用来细化句法和语义的输出结构。有许多复杂的语法用于准确表达句子层面的句法和语义信息。其中最为广泛使用的语法是成分语法（Constituent Grammar）和依存语法（Dependency Grammar）。成分语法主要用于成分分析，采用层次化的短语结构树来组织句子层面的句法信息。依存语法广泛用于语法和语义分析，即依存分析中。在依存语法中，词由依赖性连接直接表示，标签表示其句法或者语义关系 [38]。由于依存关系结构的简洁性和容易标注性，依存分析相比成分分析更加受到关注，应用也更加广泛。

句子依存分析主要分为两类，语法依存分析（Syntactic Dependency Parsing, DP）和语义依存分析（Semantic Dependency Parsing, SDP）分别是关于句子语法和句子语义的依存分析，它们可以被视为依存分析中延伸出来的。语法依存分析主要针对句子层面中语法结构和语法规则关系分析。语义依存分析主要针对句子层面中语义的依赖关系分析 [42–44]。

依存分析中常用数据结构是依存关系树（Dependency Tree）主要反映句子中的句法和浅层语义信息。由于人们对深层语义的需求越来越大，随着对深度语义解析的需求不断增长，仅靠依赖树很难表达。因此对于依存关系图研究问题被广泛关注 [44–46]。依存关系图允许在一个句子中一个词有多个（包括零

个)连接核心词。特别地,依存关系图仍然由一组组双词汇的依赖关系构成,节点对应于表面词汇,边缘表示核心词与修饰词之间的语义关系。对于依存关系图有不同的形态化方式,如今,往往将依存分析和语义角色标注(Semantic Role Labeling)技术结合起来,生成一个联合了 SRL 的依存关系图[47, 48],用以更好的分析句子语法和语义特征。

针对句子的句法和语义分析已经研究了几十年。从传统的统计模型到目前占主导地位的神经网络方法。对于解析模型来说,带有预训练的语境化词汇表征的神经网络方法在几乎所有的数据集上都取得了最好的表现[49]。越来越多的人倾向于使用简单的编码器-解码器框架进行解析,这样就可以应用经过充分研究的训练策略。如今,针对句子分析任务表现逐渐被接受,这都要归功于神经网络模型以及语言资源的发展。

文本为了更好的度量众包测试报告中文本信息的语义和语法,本文将采用依存分析,并联合语义角色标注对众包测试报告进行句子分析的自然语言预处理。现有的针对句子分析的 NLP 工具已经非常成熟[50, 51]。特别地,高效的 NLP 处理工具如语言技术平台(LTP)^①和 Stanford Parser^②。本文研究中采用 LTP 工具,有关理论将在第三章背景知识中详细介绍。

2.4 本章小结

本章主要概述基于句子分析的众包测试报告质量度量方法中相关工作。第一是现有缺陷报告质量的相关研究。首先是缺陷报告实证调研,证明了缺陷报告质量的重要性。然后是利用缺陷报告重复性检测、缺陷报告融合和信息摘要提取等自动化技术协助人工完成缺陷报告质量审核。第二是关于需求规格说明书质量评估的相关研究。重点关注软件工程中重要报告文档质量是如何从期望属性进行评估,是如何从“精确性”和“原子性”出发,运用分类学方法进行分析 and 总结文档的期望属性。第三是关于自然语言处理句子分析的理论研究。重点关注句子分析中依存分析的研究方法和现有句子分析的成熟 NLP 工具。

^①<http://www.ltp-cloud.com/>

^②<https://nlp.stanford.edu/software/>

第三章 背景知识

3.1 众包测试报告实例

在众包测试活动中，众包测试工人会根据众测平台推荐选择自己擅长的任务去完成，需要在规定时间内以众包测试报告的形式反馈自己测试过程中发现的软件系统异常现象。众测平台的审核人员负责审核这些测试报告。这些测试报告是用自然语言写成的，同时还有一些基于预定义格式的屏幕截图 [52]。

一个典型的测试报告通常由不同的字段组成，如环境、描述、截图信息，其中一些字段在不同的项目中可能略有不同。表 3-1 中展示了四份真实众包测试报告，收集自慕测平台^①，该平台成立于 2012 年，是一个众包测试平台，同时也是全国大学生软件测试大赛的支撑平台。

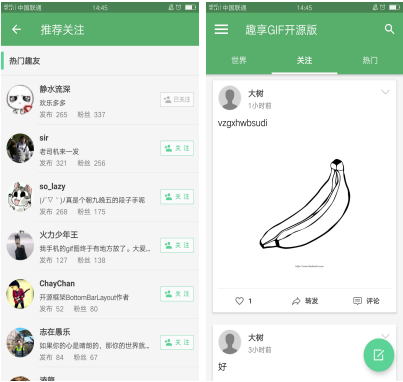
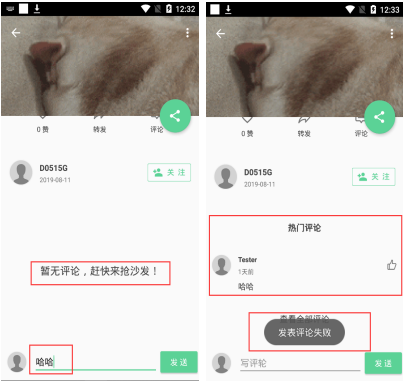
环境信息众包测试工人的设备信息，主要包括手机类型、操作系统、系统语言、屏幕分辨率。文本内容是主要是关于 bug 复现步骤和 bug 描述。bug 复现步骤描述地是实际的测试步骤，给审核人员提供复现 bug 支持。bug 描述是对错误的详细描述，偶尔也涉及到真实用户体验，帮助审核人员对 bug 正确性进行判断。现场截图有时会提供必要的一些图片信息，以捕捉错误发生时的状态，协助审核人员更好的理解文本描述。

由表 3-1 可知，众包测试报告 TR_1 无论是结构还是文本内容相对完整。高质量的众包测试报告应当如同 TR_1 所示。 TR_1 有着明确的环境信息和两张截图信息，并且在文本内容进行了分点描述。分点描述对于可再生性的理解上有着重要的其中 1-6 点一步步详尽的描述 bug 的复现步骤，在第 7 点中描述了出现的 bug。截图信息能够更加直观的给与审核人员相关提示，有截图的众包测试报告在审核时会更加的方便快捷。

TR_2 与 TR_1 相比缺少了截图信息，这总体上影响了报告的可理解性。虽然也能通过文本内容获知缺陷问题，但是没有截图那么直观的给与审核人员相关提示。这对于审核平台而言是不提倡的，众包测试平台希望众包测试工人们都能提供结构相对完整的测试报告，特别是关于缺陷的截图，这将大大提高平台

^①<http://www.moocetest.org/>

表 3-1: 众包测试报告实例

序号	环境信息	文本内容	图片信息
TR ₁	操作系统: Android 5.1 设备型号: OPPO R9 系统语言: 中文 屏幕尺寸: 5.5 英寸	1、打开个人界面 2、点击“推荐关注” 3、关注一个账号 4、回到主界面 5、点击“关注” 6、刷新界面 7、发现关注用户的信息没有显示	
TR ₂	操作系统: Android 8.0.1 设备型号: 华为 mate10 系统语言: 中文 屏幕尺寸: 5.9 英寸	1. 进入登录界面 2. 输入错误的手机号，如”123456” 3. 点击获取验证码 4. 直接提示验证码错误	
TR ₃	操作系统: Android 8.1 设备型号: 小米 8 系统语言: 中文 屏幕尺寸: 6.21 英寸	在 app 中点击进入任意趣图页面，点击评论功能，输入评论内容点击发送，评论栏中以成功显示自己发送的评论，而系统则提示发表评论失败！如图所示。	
TR ₄	操作系统: Android 6.1 设备型号: 华为荣耀 5A 系统语言: 中文 屏幕尺寸: 5.5 英寸	个人主页-发布信息-停止运行	

审核效率。

TR_3 与 TR_1 相比较, 主要是描述上没有严格遵守规定格式, 即序列化测试步骤。这不仅增加了工人漏掉步骤的风险, 还增加审核人员阅读负担。同截图信息会被平台提倡鼓励一样, 分点描述也是审核人员希望看到的, 平台往往会建议工人在描述字段采用序列化。文本内容分点描述是较为规范的一种形式, 这使得阅读起来条理更加清晰, 可理解性大大增强。此外, 审核人员也可以跟着分点步骤一步步的进行缺陷复现, 规范性的表达无形中加快了审核效率。

TR_4 是低质量众包测试报告的实例, 是一份不合格的测试报告, 其结构不完整, 没有截图信息, 而且文本描述是十分混乱, 模糊不清, 即没有步骤描述也没有缺陷描述, 这严重影响了审核人员对 bug 进行复现和确认。首先结构上来说, 缺少了截图信息。其次最为重要的文本内容描述部分寥寥数语, 既没有描述清除缺陷复现步骤, 也没有描述清除缺陷所在。这样低质量的众包测试报告对于测试工人来讲失去了一次获得测试奖励的机会, 对于审核人员而言则消耗了时间和精力。值得注意的是, 这份报告只从词汇维度进行判别是不够的, 例如词汇不能很好的理解“停止运行”的前后文关系。

从 4 份实例中可以看出一份高质量的众包测试报告首先结构上完整的, 其次文本内容描述是条理清晰描述 bug 复现步骤和相关缺陷描述的, 最后是其缺陷描述是正确的。当然其描述 bug 的正确性与否并不是文本研究的重心。众包测试报告的正确性即 bug 的真实性往往是众包测试平台人工审核进行判断的。本文主要关注的众包测试报告的结构完整性、复现步骤和缺陷描述质量如何。可见, 众包测试报告的质量对于审核而言有着重要意义。高质量的测试报告能够帮助审核人员快速定位 bug。同时关于高质量测试报告的相关规范定义对于测试人员也有重要意义。当测试工人发现了 bug 并编写一份高质量的测试报告能够更好的获取对应奖励。

总的来说, 众包测试报告质量评估是众包测试平台审核环节的重要组成部分。高质量的测试报告能够提供更完善的缺陷复现步骤帮助审核人员快速定位和识别缺陷。反之低质量的测试报告可能会误导审核人员影响审核进度。当面对多个揭示同一错误的测试报告时, 审核人员应当选择高质量的报告进行优先审核。对于众包测试工人而言, 他们也可以通过学习什么是高质量的测试报告, 进一步提升自己编写众包测试报告的能力。此外, 鉴于词汇维度的局限性, 本文从句子维度出发, 利用句子分析技术试图解决众包测试报告质量评估中文本语法和语义质量度量问题。

3.2 语法依存分析

本节和下一节将介绍本文使用的 LTP 工具的相关背景知识。语言技术平台（Language Technology Platform, LTP）是开源中文自然语言处理工具，提供词汇分析、句子语法分析和句子语义等服务。除特殊说明外，下面所介绍标签内容均为 LTP 平台使用的标注规范，即遵循 CTB5 中汉语数据集的标注规范。

语法依存分析（Syntactic Dependency Parsing, DP），主要针对句子层面中语法结构和语法规则关系分析，如识别句子中定状补等语法组成部分和相互之间的关系。表 3-2 中展示了语法依存关系标志符和详细描述。

表 3-2: 语法依存关系标注描述

标签	描述	标签	描述	标签	描述
SBV	主谓关系	ATT	定中关系	LAD	左附加关系
VOB	动宾关系	ADV	状中结构	RAD	右附加关系
IOB	间宾关系	CMP	动补结构	IS	独立结构
FOB	前置宾语	COO	并列关系	WP	标点
DBL	兼语	POB	介宾关系	HED	核心关系

依存分析中需要采用合理的数据结构，包括依存关系树和依存关系图等，用于反映句子中的句法和浅层语义信息。LTP 采用的数据结构是依存关系树。依存关系树中，节点是分词后的一个词汇，一条边表示依存关系，用于连接两个节点，分别是核心词和修饰词，其中树根是一个伪词（pseudo-word），即 ROOT，主要用于指向句子核心词以及使得句子结构更加清晰。

依存分析过程中，首先是将输入句子 $Sentences$ 进行分词和词性标注。分词序列的结果为： $Sentences = w_0w_1...w_n$ ，其中 w_i 表示句子的第 i 词。特别地， w_0 是伪词，即 ROOT。词性标注序列结果为： $Tag = t_0t_1...t_n$ ，其中 t_i 表示句子的第 i 词的词性。然后将 $Sentences = w_0w_1...w_n$ 和 $Tag = t_0t_1...t_n$ 作为依存分析的输入。一个依存树表示为 $T = \{(h, m, l) : 0 \leq h \leq n, 1 \leq m \leq n, l \in \mathcal{L}\}$ ，其中 (h, m, l) 表示一个从核心词（father） w_h 到修饰词（child） w_m 的依存弧， l 表示依存弧的关系类型， $\mathcal{L}$ 是依存弧关系类型的集合。依存弧的关系类型用来表示两个词之间的语法或语义关系 [42]，详细定义如下。

一个的依存图 $G = (V, A)$ 的节点为： $V = \{0, 1, ..., n\}$ ，其中节点 i 表示输入句子 $Sentences$ 的分词 w_i ， V 必须包含输入句子 $Sentences$ 中所有的分词；边为：

$A \subseteq V \times V \times \mathcal{L}$ 。那么输入句子 *Sentences* 所对应的依存树 (Dependency Tree) $T = (V, A)$ 还需要满足以下几个条件:

- (1) 树中任意子节点有且只有一个父节点: 每个分词 w_i 有且只能修饰一个核心词, 具体包括如下含义:
 - $\forall i \in V \setminus \{0\}, l \in \mathcal{L} : (i, 0, l) \notin A$ 。表明 w_0 没有核心词, 即句子中不存在任何词 w_i 指向 (修饰) 树根 **ROOT**, 符合树的定义。
 - 如果 $(i, j, l) \in A$, 那么 $\forall l' \in \mathcal{L} \setminus \{l\} : (i, j, l') \notin A$ 。表明任何一个词 w_i 有且只有一种依存关系修饰其核心词。
 - 如果 $(i, j, l) \in A$, 那么 $\forall r \in V \setminus \{i\}, l' \in \mathcal{L} : (r, j, l') \notin A$ 。表明任何一个词 w_i 只能修饰唯一的核心词, 即被修饰的核心词有且只有一个。
- (2) 树根可达任意子节点: 用 $i \rightarrow j$ 表示一条不考虑依存关系的依存弧 $(i, j, *)$ 。
 $i \rightarrow^* j$ 表示从节点 i 出发, 沿着依存弧, 可以到达节点 j 。即 $\forall i \in V \setminus \{0\}, 0 \rightarrow^* i$ 。表示节点 0 (树根 **ROOT**) 能够到达依存树中任何其他节点。
- (3) 树没有环: $\forall i \in V, j \in V$, 如果 $i \rightarrow^* j$, 那么 $j \nrightarrow^* i$ 。依存树不能存在环, 否则不符合树的定义。

一个的依存图 $G = (V, A)$ 同时满足条件 (1) 以及条件 (2, 3) 中的任意一条即为依存树。可以得知, 一个依存树包含 n 个有向边, 即 $|A| = n$ 。即如果依存图满足树中任意子节点有且只有一个父节点, 并且满足树根可达性, 那么肯定无环。同样如果依存图满足树中任意子节点有且只有一个父节点, 并且满足无环, 那么树根具有可达性。

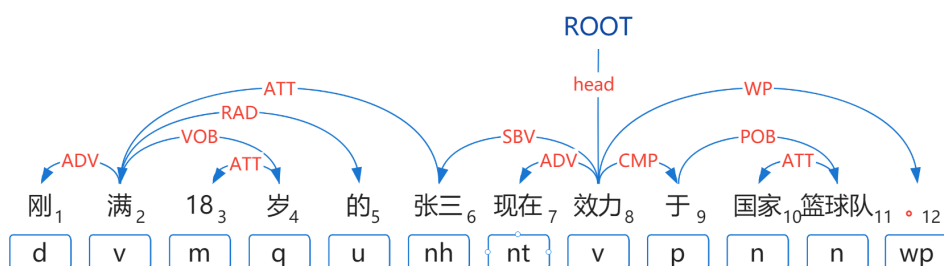


图 3-1: 语法依存树实例

图 3-2 给出了一个语法依存树的例子, “刚满 18 岁的张三现在效力于国家篮球队。”是输入句子。输入句子分词后总共有 13 个词, 编号从 0 到 12, 其中 w_0 是 **ROOT**。分词下面的蓝色方框表示每个词的词性, 例如 **n** 表示名词, 即“篮球队”词性是名词, **v** 表示动词, 即“效力”词性是动词。特别地, w_{12} 词

性为 **wp**，表示标点符号。句子分析的分词也包括了标点符号的分析和标注。更加详细的词性标注集可以参见 LTP 官网说明。

图中 w_0 是 **ROOT**，一个伪词，用于指向句子核心词以及使得句子结构符合树的定义。依存弧 $(4, 3, ATT)$ 表示“18”修饰“岁”，语法依存关系类型为定中关系（**ATT**），也可以表示为 $3 \xleftarrow{ATT} 4$ 。图中依存树高为 6，其中 **ROOT** 能够通过依存弧到达这个句子的任意一个词汇。例如 **ROOT** 通过依存弧到达最远子节点 w_3 的依次经过的依存弧为： $8 \xleftarrow{head} root$ 核心关系、 $6 \xleftarrow{SBV} 8$ 主谓关系、 $2 \xleftarrow{ATT} 6$ 定中关系、 $4 \xleftarrow{VOB} 2$ 动宾关系和 $3 \xleftarrow{ATT} 4$ 定中关系。语法依存关系详细说明见表 3-2

3.3 语义依存分析

语义角色标注 (Semantic Role Labeling, SRL) 一种浅层的语义分析技术, 标注句子中某些短语为给定谓词的论元 (语义角色), 如施事、受事、时间和地点等。SRL 中核心的语义角色有六种。记为: A0-A5。其中 A0 通常表示动作的施事, A1 通常表示动作的影响等, A2-A5 根据谓语动词不同会有不同的语义含义。除此此外, 详细附加语义角色列表如表 3-3。

表 3-3: 附加语义角色标记

Tag	描述	Tag	描述
ADV	default tag (默认标记)	MNR	manner (方式)
BNE	beneficiary (受益人)	PRP	purpose or reason (目的或原因)
CND	condition (条件)	TMP	temporal (时间)
DIR	direction (方向)	TPC	topic (主题)
DGR	degree (程度)	CRD	coordinated arguments (并列参数)
EXT	extent (扩展)	PRD	predicate (谓语动词)
FRQ	frequency (频率)	PSR	possessor (持有者)
LOC	locative (地点)	PSE	possessee (被持有)

语义依存分析 (Semantic Dependency Parsing, SDP), 分析句子中各词汇的语义关联, 并将语义关联以依存树的结构呈现, 目的是获取深层的语义信息。相较于语义角色标注, 语义依存对句子语义信息的刻画更加完整全面。LTP 中将语义依存关系分为三类: 分别是主要语义角色, 每一种语义角色对应存在一个

嵌套关系和反关系；事件关系，描述两个事件间的关系；语义依附标记，标记说话者语气等依附性信息。详细内容见 LTP 官网。

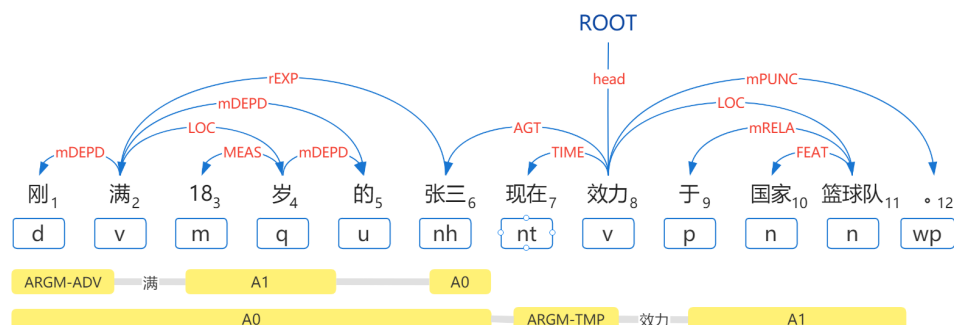


图 3-2: 联合语义角色标注的语义依存树实例

图 3-2 给出了一个联合语义角色标注的语义依存树例子，“刚满 18 岁的张三现在效力于国家篮球队。”仍是输入句子。词汇 w_i 是分词结果，每个词汇下方蓝色框表示内容是该词汇的词性。与语法依存树实例相同，ROOT 是一个伪词，在依存树中指向整个句子的核心词“效力”，即“效力”是核心谓语。依存弧 $(8, 7, TIME)$ 表示“效力”和“现在”的语义依存关系类型是主要语义角色中的时间角色（TIME）。依存弧 $(8, 12, mPUNC)$ 表示“效力”和“。”的语义依存关系类型是语义依附标记中的标点标记（mPUNC）。

词性标注下面是语义角色标注分析结果。SRL 能够浅层的分析语义，能够找到句子中的主要动作和该动作的论元（A0-A5）。A0 表示动作的实施，A1 表示动作的影响。一个句子中可能包含多个动作，图中例句有两个动作。第一个是张三满 18 岁，其中张三是动作的实施者，18 岁是动作影响，主要动作是满。第二个是“满 18 岁的张三”效力国家队，“满 18 岁的张三”是动作的实施，国家队是动作影响，主要动作是效力。

3.4 本章小结

本章介绍了文章相关的背景知识，众包测试报告实例重点介绍了众包测试报告的结构和文本特点。接下来介绍了本文提出的基于句子分析的众包测试报告质量度量方法中所采用的句子分析的基本原理，主要包括语法依存分析和语义依存分析。重点阐述了依存分析的数据结构表示之依存树，以及语法依存树实例和联合 SRL 的语义依存树实例。除了介绍其基本原理外，重点解释了 LTP 平台对于数据集的标注规范。

第四章 测试报告质量度量技术

本章将阐述基于句子分析的众包测试报告质量度量方法的具体研究内容，本文按照五个部分进行描述。首先是该方法的整体组成部分，即质量评估框架；其次是众包测试报告的期望属性的分析和总结；紧接着是期望属性的度量指标定义；再然后是质量评估规则判断的设计和构建；最后是规则判断详细得分设计。

4.1 质量评估框架

在本文中，我们试图通过基于句子分析的规则判断将测试报告进行质量评分来解决众包测试报告质量评估问题。我们提出了一个新的框架，Quality Assessment for Crowdsourced Test Reports Based on Dependency Parsing (CTRQSDP)，用于自动化评估众包测试报告质量得分。

首先，自然语言处理中的句子依存分析被用于预处理众包测试报告。句子分析中具体的标注集信息和依存关系见本文背景知识章节。然后，我们定义了一系列可量化的质量指标来衡量测试报告的期望属性。特别地，关于文本内容的度量指标依赖于句子分析预处理。最后，我们立足质量指标利用句子依存分析结果构建规则，用于自动化评估众包测试报告质量得分，并对于每一份众包测试报告输出相应的质量评估报告。质量评估报告包括具有期望属性的雷达图得分详情和违反规则的具体内容。

基于句子分析的众包测试报告质量度量框架如图4-1。主要描述一份众包测试报告怎么通过已经构建好的规则最后计算出具体得分。众包测试报告经过预处理再利用规则库进行判断，最后计算总得分。CTRQSDP的实现方式是规则判断，规则库中包含了形态学规则、相关性规则和分析性规则。形态学相关规则是判断测试报告的表面特征。形态学特征是判断测试报告的结构特征，主要面向的是众包测试报告中的环境信息还有截图信息两个组成部分。分析性规则主要面向的是众包测试报告中的文本信息部分。分析性规则需要使用预处理后的词性标注、语义角色标注以及句子依存分析结果。分析性规则利用这些标

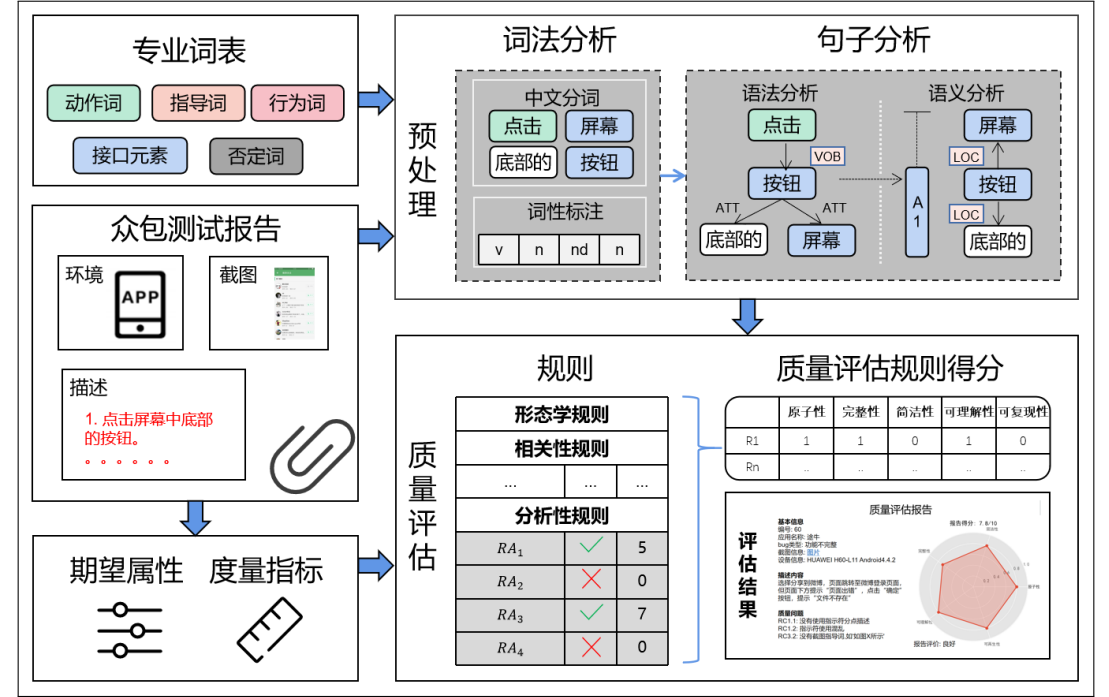


图 4-1: 基于句子分析的众包测试报告质量评估框架

注和关联关系，并结合本文统计汇总的众包测试报告专用词汇进行判断。

从组成结构来看，本文提出的质量度量方法主要包括三个部分。第一部分是总结和定义报告的期望属性以及对应期望属性可量化的质量度量指标；第二步部分是根据总结整理词表和句子分析自然语言处理技术进行基于句子分析的规则设计和构建；最后一步是众包测试报告质量详细规则得分设计，以及对每一份众包测试报告输出一份质量评估报告。质量评估报告包括具有期望属性的雷达图得分详情和违反规则的具体内容。

第一个部分是根据众包测试报告特性分析和定义测试报告的期望属性，在期望属性基础上确定对应的度量指标。3.3 期望属性详细描述了众包测试报告的期望属性，3.4 度量指标则详细描述了期望属性对应的具体度量指标，以及它们相互之间的关系。第二个部分是根据真实众包测试报告数据，整理众包测试报告专业词表，并基于句子分析完成规则库的构建。并总结了提出的规则和期望属性之间的关联关系。该部分的主要工作描述在 3.5 规则构建。最后一个部分在规则库上根据不同期望属性重要程度设计质量评估得分。我们还利用雷达图来描述一份众包测试报告在 5 个期望属性上的具体得分情况，便于我们更加直观的观察测试报告质量。对 CTRQSDP 的有效性和准确性的实验验证结果将在第四章实验部分详细说明。

4.2 期望属性

诚然，质量是一个模糊的概念，与不同的评价标准有关 [32]。不同的开发者通常对同一份文档质量有着不同看法。软件工程各个阶段都有着重要文档的生成。需求工程中，研究者们对需求规格说明书质量进行了深入研究 [33–35]。最常用的是定义一些列的指标来量化需求规格的期望属性 [32, 36, 53]。软件测试中，研究人员已经了一些列实证研究” 什么是一个好的 bug 报告”[8, 28, 29]，探讨了 bug 报告的期望属性，包括原子性、正确性、完整性、简洁性、可理解性和可再生性 [12, 13, 30]。

受 Génova 等人 [32] 研究启发，在现有研究基础上，从软件技术文档质量基本属性“精确性”和“原子性”出发，运用分类学方法分析众测平台方和测试工人各自对众包测试报告质量期望，总结了众包测试报告的期望属性如图 4-2。

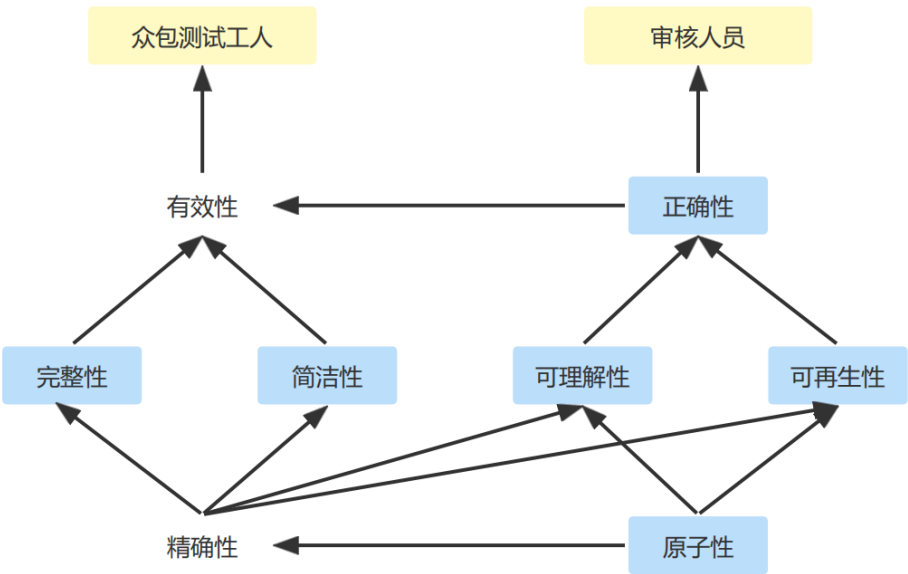


图 4-2: 众包测试报告质量期望属性间的关联

从众包测试工人的角度来看，众包测试报告的的最终质量是有效性（Validability），即众包测试工人编写的测试报告必须是有效的才能给工人带来实际的收益。其中有效性最为重要的体现是正确性。在第二个层次上，这个属性可以进一步展开为其他三个属性：完整性（Completeness）、简洁性（Consistency）和正确性（Correctness）。相反，从审核人员的角度来看，我们可以说两个基本品质是：可理解性（Understandability），即众包测试报告各

部分容易理解,清晰无歧义;可再生性(Reproducibility),即测试报告的步骤描述和错误描述详细明确,易于复现。在可理解性的基础之上,如果描述的缺陷是被审核人员验证是能够被复现的,那么意味着这份众包测试报告是正确的。正确的众包测试报告对于众包测试平台而言,能够完成发包方的任务以获得收益;对于测试工人而言能够获得对应的劳动报酬。

可以在以上期望属性分类法中区分出第三个更基本的层次,它将包含两个额外的、甚至更基本的属性:精确性(Precision)和原子性(Atomicity)。完整性、简洁性、可理解性和可再生性取决于精确性:更精确的语言有助于编写更完整、简洁、可理解和可复现的众包测试报告。另一方面,原子性影响精确性、可理解性和可再生性,尤其是可再生性,从而影响众包测试报告的正确性和有效性。

测试报告质量度量首先需要根据测试报告呈现出来的特点根据分类学定义规范其期望属性,其次需要在期望属性的基础上进行定义和描述,最后选择能够进行机器质量度量的期望属性以便进一步细化度量指标。本文通过实际调研,立足于众包测试报告属性,即众包测试报告以短文本为主,句子结构相对简单、文本结构相对稳定且完整以及文本内容为测试相关等特点,针对每一个测试报告总结如下期望属性:原子性、正确性、完整性、简洁性、可理解性和可再生性等。其详细含义如表 4-1:

表 4-1: 众包测试报告期望属性

期望属性	描述
原子性	测试报告包含且只包含了一个特定的缺陷信息和它的触发步骤。
正确性	测试报告描述了一个真实的缺陷。
完整性	测试报告必须具备每个部分,且每个部分都符合期望的格式。
简洁性	测试报告语言简洁不繁琐。
可理解性	测试报告各部分容易理解,清晰无歧义。
可再生性	测试报告的步骤描述和错误描述详细明确,容易复现。

原子性指众包测试报包含且只包含了一个特定的缺陷信息和它的触发步骤。众包任务具有原子性 [1]。在众包测试工业界中,众包测试工人发现并提供一个缺陷报告即可获得对应的奖励。面临竞争的测试工人们为了获取更多的奖励,当其发现一个 bug 时则会尽快编写报告提交至众包测试平台。逐渐地,众包测试平台以一个 bug 作为构建众包测试报告的最小单位。这使得众包测试报

告的原子性特指测试报包含且只包含了一个特定的缺陷信息和它的触发步骤。

正确性是测试报告描述了一个待测软件中客观存在的真实 bug。众包测试报告的最本质特征是提供真实的缺陷报告。关于正确性，需要众包测试平台的审核人员进行人工判断。因此本文的研究中将不对正确性进行质量评估建模。

完整性是指测试报告必须具备每个部分,且每个部分都符合期望的格式。众包测试报告有着自己独有的期望格式。如表 3-1 展示，众包测试报告通常含有环境信息、文本内容描述和截图信息等。其中文本内容描述部分由两个部分组成，一个部分是 bug 的复现步骤描述，另一个是缺陷描述。表 4-2 分别例举了众包测试报告 TR_1 和 TR_3 文本内容的具体组成。

表 4-2: 众包测试报告文本内容组成

测试报告文本内容	TR_1	TR_3
复现步骤	1、打开个人界面 2、点击“推荐关注” 3、关注一个账号 4、回到主界面 5、点击“关注” 6、刷新界面	在 app 中点击进入任意趣图页面，点击评论功能，输入评论内容点击发送，评论栏中以成功显示自己发送的评论，如图所示。
缺陷描述	7、发现关注用户的信息没有显示	而系统则提示发表评论失败！

简洁性指测试报告语言简洁不繁琐。作为软件工程技术文档的一类，众包测试报告必须要具备简洁性。尽可能简洁的语言描述详尽的问题，这是一份高质量的众包测试报告应该具有的特性。测试工人需要训练简洁性的描述方法，以尽可能的提高高质量的测试报告。这能极大减少审核人员审核阅读花费的精力，一定程度上降低测试工人和审核人员之间借助测试报告的沟通成本。加快审核效率的同时，也能快给予奖励，更好的激励测试工人，进而形成良性循环。

可理解性是测试报告各部分容易理解,清晰无歧义。众包测试报告是首先是一份文档，是众包测试活动中测试工人自己劳动成果的总结，以及众包平台发放奖励的凭证，是一份对于众包测试平台和测试工人都极为重要的文档，作为文档可理解性是基本的。如果一份众包测试报告不具备可理解性，那么意味着没有人可以理解它，不能被理解，这份文档也就没有存在价值。

可再生性是测试报告的步骤描述和错误描述详细明确, 容易复现。这也是众包测试报告的基本属性要求。即是众包测试平台的要求也是众包测试工人的要求。众包测试平台期望测试报告能够提供详细的步骤描述和错误描述, 以便审核人员能够尽复现 bug 并进行错误判断。众包测试工人也期望自己编写的测试报告能尽可能的审核通过, 以便获取对应的奖励。因此, 可再生性基本属性要求。

综上, 本文采用的用于众包测试报告质量评估的期望属性是: 原子性、完整性、简洁性、可理解性和可再生性。

4.3 度量指标

研究表明, 文本质量的期望属性依靠可量化的指标进行定量分析, 利用定量的质量指标直接或间接地反应定性的期望属性, 用以度量文本质量 [32]。现有的研究已经系统地提出了质量指标的总体分类法 [12, 13, 53], 然而, 其提出的分析指标是基于词汇的, 不能很好度量文本语法和语义。本文创新性地提出了基于句子分析的分析性指标。基于句子分析并从分类学角度根据不同维度将众包测试报告的度量指标分为三大类, 即形态学指标、相关性指标和分析性指标。其详细解释如下:

- 形态学指标。主要用来评估文本表面特征, 不涉及语法和语义信息, 具有通用性, 可以用于评估各类的文档, 不限于众包测试报告。例如文本长度、可读性和标点符号等。
- 相关性指标。评估众包测试报告的特有结构化特征以及内容格式规范性。例如环境信息完整性度量; 复现步骤和缺陷描述逐条记载, 条理清晰度量; 截图信息完整性度量。
- 分析性指标。通过相对复杂的自然语言分析对众包测试报告进行文本分析。本文基于依存语法分析方法进行众包测试报告文本分析。主要分析句子中是否准确且完整的描述了复现步骤的用户行为和缺陷问题。

下面将详细阐述每个指标的具体内容以及采用的度量方法, 此外还将详细描述度量指标和期望属性之间的关系, 即直接相关或间接相关。值得注意的是测试报告是由四个部分信息构成, 即环境信息、步骤复现描述、缺陷问题描述以及截图信息。其中环境信息是由设备信息专业词汇构成, 截图信息独立于自然语言为图片格式, 以上两者的相对容易评估, 主要涉及的质量指标分类是相

关性指标。步骤复现描述和缺陷描述是由自然语言描述，除了相关性指标中环境信息和截图信息指标以外，其余各项指标均用于评估众包测试报告中步骤复现描述和缺陷描述。本文的所有度量指标总结如表 4-3。

表 4-3: 期望属性度量指标详情

类别	度量指标	指标内容
形态学指标	文本长度	文本中字数长度
	可读性	阅读一段文本的困难程度
	标点符号	文本中标点符号使用情况
相关性指标	逐条记载	文本内容逐条记载方式
	环境	环境字段信息
	截图	截图信息
分析性指标	接口元素	接口元素描述详情
	用户行为	用户行为描述详情
	系统缺陷	系统缺陷描述详情

众包测试报告中的步骤复现描述和缺陷描述部分是由测试内容相关的短文本构成。其涉及的分类指标有通用的形态学指标和立足于依存语法分析的分析性指标。该部分内容对于开发者修复 bug 起到至关重要的作用。本文重点关注如何度量步骤复现描述和缺陷描述的质量。除了相关性指标中环境信息和截图信息指标以外，其余各项指标主要用于评估测试报告中步骤复现描述和缺陷描述。

4.3.1 形态学指标

形态学指标。主要用来评估文本表面特征，不涉及语法和语义信息，具有通用性，可以用于评估各类的文档，不限于众包测试报告。通用的文本质量度量指标主要有形态学指标如：文本长度，可读性和标点符号等 [32, 53]。这一维度旨在从整体上对于文本内容的质量进行一定程度的度量，是基本且必要的度量。

- 文本长度。众包测试报告的文本长度保持在合适数值范围较好 [13]。其直接地度量测试报告的原子性和简洁性，间接地反映其他的所有期望的属性。
- 可读性。文本可读性表示阅读一段文本的困难程度 [32, 53]。针对中文文

本，本文采用可阅读性度量采用的公式 [54]：

$$Y = 14.9596 + 39.07746X_1 + 1.011506X_2 + 2.48X_3 \quad (4-1)$$

其中 X_1 表示的难词（5600 个通用词语之外的词）的比例， X_2 和 X_3 分别表示句子的数量和汉字的平均笔画数。

- 标点符号。度量文本的规范程度和句子的可读性。主要评估方式有文本中是否中英文逗号或者句号混用，以及句子中是否标点符号使用过多影响了句子可读性。同时也是中文文本格式规范性度量的一种常见方法 [32]。

以上三个指标主要用于从文本整体内容出发，用于评估众包测试报告文本内容的表面特征，是通用的文本质量度量指标。其中标点符号度量指标的主要用于判别中文标点符号使用规范性，文本质量度量中规范性也是很重要的一环。

4.3.2 相关性指标

相关性指标。评估众包测试报告的特有结构化特征以及内容格式规范性。测试报告是一个由多个字段组成的结构性文件，每个字段都为开发人员提供指定的信息，以理解或重现错误 [7, 10, 11]。一个典型的测试报告通常由不同的字段组成，如环境、描述、截图信息，其中一些字段在不同的项目中可能略有不同。因此相关性指标主要包括环境信息完整性度量；复现步骤和缺陷描述逐条记载，条理清晰度量；截图信息完整性度量。

- 逐条记载。在众包测试报告中的复现步骤描述这部分通常会使用一些项目符号或者数字编号来组织内容。例如“+”、“*”、“•”或者数字编号等。
- 环境。在众包测试报告中，环境是一个重要且必要的结构化字段，用来表示移动设备的基本配置信息。通过阅读环境，开发者能清楚地意识到软件系统在不同移动设备上的可适配性。详细的环境信息如：操作系统、设备型号、系统语言和屏幕尺寸。
- 截图。一个完整的测试报告不仅仅包含描述性的自然语言信息，同时含一些图片信息。这些图片能有效地捕捉系统出现问题时的状态信息。图片信息也能更好帮助审核人员进行 bug 的复现和测试报告的审核。除了判断报告中是否含有截图信息，还可以判断是否引导截图信息的文字描述。详细准

确的描述可以帮助审核人员更好的定位 bug。

4.3.3 分析性指标

分析指标主要用于分析描述字段。描述是主要是关于 bug 复现步骤和缺陷描述。bug 复现步骤是由测试工人根据实际的测试步骤进行描述的，给审核人员提供复现 bug 支持。bug 描述是对错误的详细描述，偶尔也涉及到真是的用户体验，帮助审核人员对 bug 正确性进行判断。我们使用自然语言处理中的句子分析，对众包的测试报告进行文本分析。

分析性指标。通过相对复杂的自然语言分析对众包测试报告进行文本分析 [32]。本文基于句子依存关系分析方法进行众包测试报告文本分析。主要分析句子中是否准确且完整的描述了复现步骤的用户行为和缺陷问题。

- 接口元素。移动应用是由许多的界面接口元素组成的。众包测试中工人执行测试必须通过点击这些接口元素来完成相应功能的测试。在步骤复现描述中含有大量的界面接口元素词，如：“导航栏”、“搜索栏”、“侧边栏”和“按钮”等。
- 用户行为描述。众包测试报告中步骤复现通常每点中包含有用户行为，审核人员根据用户行为进行相应步骤的操作来实现 bug 复现以达到审核的目的。例如“点击“我的”按钮，进入新页面”的用户行为描述和“点击，进入新界面”的描述相比，前者描述更加完备拥有更高的可理解性。因此有必要对于用户行为描述的准确和完整性进行度量。
- 缺陷描述。众包测试报告中必须含有关于缺陷的描述。众包测试报告缺陷描述中，用户进行完整且详尽的缺陷描述可以更好的帮助审核人员进行 bug 的判断。众包测试报告质量评估必须要具备对于缺陷的检测度量。

分析性指标的实现需要对众包测试报告文本内容进行基于句子分析的自然语句预处理。其中最为基础的进行分词和词性标注。众包测试报告具有测试专业领域特性，因此其中涉及的词汇有着专业领域，即众包测试报告专业词汇。本文在现有研究基础上 [12, 13, 32]，在慕测平台中开展测试报告实际研究。我们广泛调研 8339 份众包测试报告，然后结合同义词林进行同义替换，总结和归纳移动应用众包测试报告专业词库。本文总结的移动应用众包测试报告专业词库的详细分类如下。

为了更加直观的理解具体的含义，我们例举上述 5 个分类词汇中的高频词

表 4-4: 众包测试报告高频专业词汇表

类别	词汇
指导词	例如、比如、注意、如、描述、尤其、如图所示
否定词	错误、错、故障、问题、症状、适配、显示、闪退、bug、崩溃、假死、卡死、冲突、死机、坏死、关闭、出错、重复、卡顿、延迟
动作词	打开、点击、选择、查看、滑动、进入、翻页、检查、退出、发送、分享、结束、注册、提示、连接、刷新、收到、下拉刷新、邀请、加载、搜索、点赞、收藏、加入、旋转、观看、安装、评论、撤销、反馈、删除、邀请、取消、定位、获取
行为词	否定含义的功能行为词：错误、错、故障、问题、闪退、bug、崩溃、假死、卡死、冲突、死机、坏死、关闭、出错、重复、卡顿、延迟
	肯定含义功能行为词：一致、反应、响应、提示、网络连接、网络、用户体验、登录、存在、可用、流畅
	页面描述行为词：悬浮、重叠、叠加、布局、遮挡、美观、显示、好看、错乱、完整、一大一小、紊乱、过大、过小、症状、适配、错别字
接口元素	导航栏、搜索栏、侧边栏、状态栏、标签栏、工具栏、通知栏、统计栏、收藏栏、按钮、标签、编辑框、进度条、刷新、滚动条、文本框、开关、菜单、键盘、图片、图形、消息框、表、窗口、对话框、视图、日期选择器、警告框、图标、登录、页面、屏幕、首页、App、主页、输入框、运动栏、栏目、二维码、照片

汇如表 4-4。

- (1) 指导词（Directive）。指导词是一个积极的指令性术语，可以帮助开发人员更好地理解测试报告。一般来说，指令性术语后面的词可能会添加一个具体的例子或从补充方面提供更多的额外信息来说明和加强错误描述。
- (2) 否定词（Negative）。测试工人通常会使用一些否定词来描述系统功能的缺失。

- (3) 动作词（Action）。在执行测试步骤时，通过点击移动应用程序的用户界面而触发的一系列动作。
- (4) 行为词（Behavior）。根据维基百科的定义，软件缺陷指的是软件中的错误、问题、误差、过失、故障或者症状等。本文将其分类为三类：
- 否定含义的功能行为词。通常包含了系统错误行为以及状态的描述，往往单个词汇就能指出系统缺陷。
 - 肯定或中性含义功能行为词。往往需要结合否定词对系统错误行为进行状态描述。
 - 页面描述行为词。页面布局中常用的错误描述词。
- (5) 接口元素（Interface）。软件系统是由许多的界面接口元素组成的。在步骤复现描述中含有大量的界面接口元素词。

4.3.4 度量指标与期望属性关系

基于分类学角度，众包测试报告的度量指标有三大类，即形态学指标、相关性指标和分析性指标。其详细的含义和内容如上述，本节重点描述这三大类度量指标与期望属性之间的关系。度量指标与期望属性之间的关系总结如表4-5。

表 4-5: 度量指标与期望属性关系

类别	度量指标	期望属性				
		原子性	简洁性	完整性	可理解性	可再生性
形态学指标	文本长度	●	×		●	
	可读性				×	
	标点符号		×		●	
相关性指标	逐条记载	●	●	●	×	×
	环境			×		×
	截图			×	×	×
分析性指标	接口元素		●		×	×
	用户行为	●		×	●	×
	系统缺陷	×	●	×	×	×

其中“×”表示指标和属性之间存在直接的关系,而“●”表示指标和属性之间存在间接的关系。

形态学指标是文本内容较为通用的表面特征，并不涉及众包测试报告的测

试专业领域的相语义度量。因此其与众包测试报告结构相关的完整性以及基本内容合格要求的可再生性没有必然联系，主要是与原子性、简洁性、可理解性相关联。相关性指标是根据众包测试报告特有的文本特征和结构提出的度量指标，与形态学指标相反，主要度量属性为完整性以及可再生性，当然，作为测试报告结构性的指标对于可理解性期望属性也有直接关联。分析性指标主要是针对众包测试报告内容文本的专业性进行评估，与原子性、完整性、可理解性和可再生性具有直接关系。

4.4 规则构建

基于众包测试报告的特点，文本采用了规则构建的方法来进行众包测试报告的质量评估。本文采用规则构建来实现众包测试报告的自动化质量评估，是因为规则判断具有可解释性、可扩展性以及易维护性等特点。其中可解释性是指基于规则判断的方法是可解释的；可扩展性指可以在不同领域制定不同的规则；易维护性指通过将规则解耦合，可方便维护规则库。规则判断可以用于众包测试报告质量度量。

众包测试报告是专业领域为软件测试且具有稳定文本结构的短文本。针对众包测试报告的特点，基于规则判断的众包测试报告度量有以下三个优势：首先是众包测试报告的专业领域强，其内容词表为软件测试领域，这使得规则制定空间范围小。其次众包测试报告文本结构稳定，通常采用分点描述复现步骤以及缺陷描述，且使用句子形式较为规范，规则判断可用性高。最后众包测试报告是短文本，使用规则判断速度快。综上本文提出如下众包测试报告质量度量规则。

从 3.3 节度量指标中可以得知，本文从分类学角度将提出的度量指标分为三大类，即形态学指标、相关性指标和分析性指标。本文下面也将在从这三大类角度进行相关规则的陈述。规则总结如表 4-6。

4.4.1 形态学规则

形态学指标主要用来评估众包测试报告文本内容的表面特征，主要采用的指标包括：文本长度、可读性和标点符号。根据形态学指标，本文提出形态学规则。

表 4-6: 规则总结

类别	判断规则	度量指标	规则内容
形态学规则	RM1	文本长度	文本长度在预设范围之间
	RM2	可读性	文本可读性在预设范围之间
	RM3.1	标点符号	标点符号使用符合规范
	RM3.2		平均句子长度在预设范围之间
相关性规则	RC1.1	逐条记载	存在指示符分点描述
	RC1.2		指示符使用符合规范
	RC2	环境	存在环境信息
	RC3.1	截图	存在截图
	RC3.2		存在截图指导词
分析性规则	RA1	接口元素	清晰无歧义的接口元素描述
	RA2	用户行为	存在交互行为描述
	RA3.1	系统缺陷	存在系统缺陷描述
	RA3.2		清晰无歧义的系统缺陷描述

规则 4.1 RM1(文本长度): 文本长度在预设范围之间

文本长度是最为直观的文本内容表面特征。由于众包测试报告是短文本，所以其文本长度不会太长，调研表明通常中文众包测试报告不超过 100 字。测试报告虽说是短文本，但是文本太短可能会造成错误内容描述不清楚，所以文本长度往往有着最低的字数限制。

规则 4.2 RM2(可读性): 文本可读性在预设范围之间

用于众包测试报告文本内容的表面特征评估的另一个主要度量指标是可读性，根据 3.3.1 节形态学指标中提出的公式计算本文的可读性。

规则 4.3 RM3.1(标点符号): 标点符号使用符合规范

众包测试报告文本内容的表面特征评估的最后一个度量指标是标点符号。其中一个规则是判断标点符号的使用规范的。本文研究的众包测试报告是主要是中文编写的，而中文编写的报告中往往存在标点符号中英混用的问题。众包测试报告的质量评估中也应该包含对于报告规范性的相关评估。

规则 4.4 RM3.2(标点符号): 平均句子长度在预设范围之间

标点符号的度量除了上面所描述的规范性以外，还有断句的重要作用，断句好坏直接影响句子可读性的。为了更好的度量断句的情况，需要对平均句子长度进行判断。

4.4.2 相关性规则

相关性指标主要是用来评估众包测试报告特有的结构化特征以及内容格式规范性。主要包括有逐条记载，环境和截图信息等。根据相关性指标，本文提出如下相关性规则。

规则 4.5 RC1.1(逐条记载): 存在指示符分点描述

首先是逐条记载，测试工人在提交的测试报告中会详细描述 bug 复现步骤。高质量的测试报告在步骤复现描述中采用的是分点描述，这使得阅读起来更加容易。同时，分点描述也是众包测试平台期望工人在编写报告时遵循的规范。因为分点描述不仅便于审核人员进行阅读，还能侧面引导测试工人更好的编写测试报告。测试工人进行分步骤后，有助于自己梳理步骤的实现前后关系，进而更好的编写更高质量的测试报告。因此逐条记载的判别至关重要。

具体的操作是判断众包测试报告中是否使用用一些项目符号或者数字编号来组织内容。例如“+”、“*”、“●”或者数字编号等。

例 4-1 例如众包测试报告实例 3-1 中的 TR_3 相比于 TR_1 在文本内容描述部分没有进行分点描述。

规则 4.6 RC1.2(逐条记载): 指示符使用符合规范

除了 RC1.1 规则关于其逐条记载的有无的判别，同时还应该具备针对规范性的使用规范性判别。对于规范性判断的 RC1.2，主要是判断同一个众包测试报告中用于步骤描述的相关符号是否具有统一性。

例 4-2 例如“+”、“*”、“●”混用，或者数字字母编号时不连续，有缺失的情形。

规则 4.7 RC2(环境): 环境信息完整

众包测试相较于传统的测试在手机用户使用环境以及适配性测试上有着巨大的优势。众包测试能够较低成本的获得多样性的环境信息，这依赖于数量众多众包测试工人。环境信息是众包测试报告中重要的组成部分，其中详细的环境信息包括操作系统、设备型号、系统语言和屏幕分辨等。这些环境信息的检测是众包测试报告质量评估中的重要的一环，本文将对环境信息的每一个字段进行一一核对。环境信息完整，即环境信息四个字段操作系统、设备型号、系统语言和屏幕尺寸均具有且正确。

例 4-3 例如众包测试报告实例 3-1 中 4 个测试报告的环境信息均是完整的。任何一个字段的缺失都会导致众包测试报告的环境信息不完整。

规则 4.8 RC3.1(截图): 存在截图

众包测试报告中还有一个显著的特征，就是往往具有图片信息。截图信息是测试工人在发现 bug 的时候设备屏幕截图，通常包含界面信息以及缺陷表现。截图信息一方面用来佐证测试工人发现的缺陷错误，另一方面帮助众测平台的审核人员进行缺陷的审核以及复现。现有众包测试平台都倾向鼓励众包测试工人提供截图信息，方便审核阶段的核实和复现。截图信息是众包测试报告质量度量中需要特别进行评估的相关性指标。本文不对图片信息的内容进行核对和校验，只是在测试报告中核对该报告是否具有截图信息。

例 4-4 例如众包测试报告实例 3-1 中， TR_1 和 TR_3 均含有截图信息，且都具有两张截图。 TR_2 和 TR_4 则没有包含截图。

规则 4.9 RC3.2(截图): 存在截图指导词

本文还将对众包测试报告中“如图所示”这类的引导词的测试报告进行判别，具有引导词的测试报告在文本编写上具有更高的文本质量。因为其很好的提供了引导词，帮助审核人员进行截图内容的更好甄别。因此，本文对截图文本描述中使用引导词的测试报告进行判别。引导词的相关定义以及具体内容见本章 3.3.3 节分析性指标。

例 4-5 例如众包测试报告实例 3-1 中， TR_3 相比较于 TR_1 ，含有截图指导词“如图所示”，更好的提示审核人员该份众包测试报告中含有截图信息。

4.4.3 分析性规则

分析性指标是指通过相对复杂的自然语言分析对众包测试报告进行文本分析。分析性指标度量的技术路线的第一步是将众包测试报告的文本内容提取处理，利用 LTP 工具进行预处理。预处理主要包括语法依存分析、语义依存分析和语义角色标注。以上预处理的相关细节见 2.3 自然语言句子分析，其中有详细的词性标记集、依存句法分析标注集和语义角色的详细说明。第二步是按照分析性规则进行判断。规则判断时需要结合本文总结的移动应用众包测试报告专有词汇表。

本文基于依存语法分析方法进行众包测试报告文本分析。主要分析句子中是否准确且完整的描述了复现步骤和缺陷问题。这部分指标以及接下来阐述的规则在整个众包测试报告质量评估中至关重要，如若没有特别的说明，下文所指的本文内容特指众包测试报告中除了环境信息之外的文本信息。特别地，分析性规则是基于句子分析的，即以句子为单位执行下面的规则。

高质量的众包测试报告中往往有着清晰的界面元素描述，特别是在复现步骤中。页面元素，即接口元素能够帮助审核人员更快地定位页面相关细节、能够更好的帮助审核人员复现缺陷找到问题所在以提高审核效率。

规则 4.10 RA1(接口元素): 清晰无歧义的接口元素描述

清晰无歧义的接口元素描述是指，首先该句子中存在“接口元素”词。“接口元素”词的定义以及具体内容见 2.4.3 分析性指标。其次在存在的基础上，接口元素词在句子中修饰的成分或者补充说明关系。修饰关系和补充关系越多表明该在接口元素的描述越加清晰无歧义。修饰关系是判断“接口元素”在句子中是否含有定状补等句子依存关系。对应表 3-2 中定中关系（ATT），状中结构（ADV）和动补结构（CMP）。补充关系是判断“接口元素”在句子中是否含有并列关系（COO）、左附加（LAD）或者右附加（RAD）。

为了更加形象的说明句子中的修饰关系和补充关系，我们先分析一个真实的例子。图 4-3 中为句子“听力时模式模式下显示全文而全文模式下不显示文字”的语法依存分析结果，以依存树这一数据结构来展示分析结果。该句来自真实的众包测试报告数据集中。可以观察得知其中包含修饰成分包括定中关系（ATT）、状中结构（ADV）和并列关系（COO）等。我们可以发现这些修饰关系或者补充关系起到了细节补充作用。合理的修饰关系和补充关系能够使得

句子的可理解性大大提高。当然这句来自真实数据集的句子并没有完全的符合句子语法规范，甚至是有语病的。不过我们主要是分析其结构用于举例，下面的规则中也会再次用到这个句子。

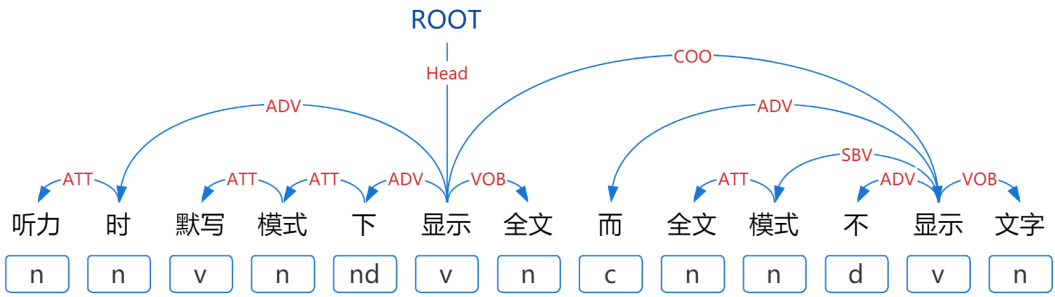


图 4-3: 测试报告句子语法分析实例

众包测试报告的复现步骤描述中，除了出现接口元素这样的名词以外，句子存在着“点击登录按钮”这样用户与系统的交互行为也能很好的帮助审核人员进行缺陷步骤复现和定位 bug。因此，对于复现步骤句子有必要进行交互行为的判断。这规则中需要进行句子依存关系分析，找到其中的依存关系，用于有交互行为的判断。通常这样的依存关系是动宾关系（VOB），在文本总结的词表中的表现则是往往由动作词加接口元素构成。基于句子依存关系分析在分词的基础上还能检测句子的完整性，能更好分析文本的语义和语法的规范性，这是分词不能做到的。

规则 4.11 RA2(用户行为): 存在交互行为描述

关于众包测试报告中是否含有交互行为的规则，是众包测试报告质量度量中复现步骤字段的基本要求。众包测试报告中对于句子中存在交互行为，本文将将其细分为三类。

一类是用户与界面元素（界面）的交互行为，如“滑动窗口”，这类交互行为语义描述为一个动作，动作往往由谓语动词以及其谓词的论元（语义角色）表现。语义角色的标注的相关细节见 2.3 自然语言句子分析。这类交互行为的判断是先找到句子中最核心的动作，而后利用语义角色 A1 关系判断交互动作的发生。如果句子中用户动作谓语 A1 关系含有接口元素词或者是界面名词，那么我们判定为用户与界面存在交互行为。接口元素词主要是移动应用中的一些页面组件名字，页面名词比如“首页”，“设置页”这类。

第二类是用户进行时间空间变化描述用以表述交互行为。具体实现利用语

义依存关系分析，判断是否有表示地点（LOC）、表示时间（TMP）等附加语义角色，借此判断是否有交互动作的发生。例如“听力时默写模式下显示全文而全文模式不显示文字”，句子利用 LOC+TMP 附加语义角色进行交互行为的描述。而“使用微信实现第三方登录时一直登不上去而且要等待很久”主要借助时间 TMP 附加语义角色进行交互行为的描述。

上述两类交互行为涉及的句子分析技术主要是关于句子的语义角色标注和语义依存分析。为了更好的理解上述两类中提到的 LOC、TMP 等标注的理解，我们将句子“听力时模式模式下显示全文而全文模式下不显示文字”进行语义角色标注和语义依存分析，并利用联合语义角色标的语义依存树数据结构来展示分析结果，如图4-4所示。“显示”与”模式”之间的语义关系是表示地点，位置（LOC）；“听力”与“显示”之间的语义关系是表示时间，时序（TIME）。我们利用时间与空间的变化来判断交互行为的发生。对于语义角色标注中，我们可以发现“ARGM+LOC”，“ARGM+TMP”这类的语义角色信息，意味着句子中存在着动作时间空间的变化，这同样可以用于交互行为的判断。

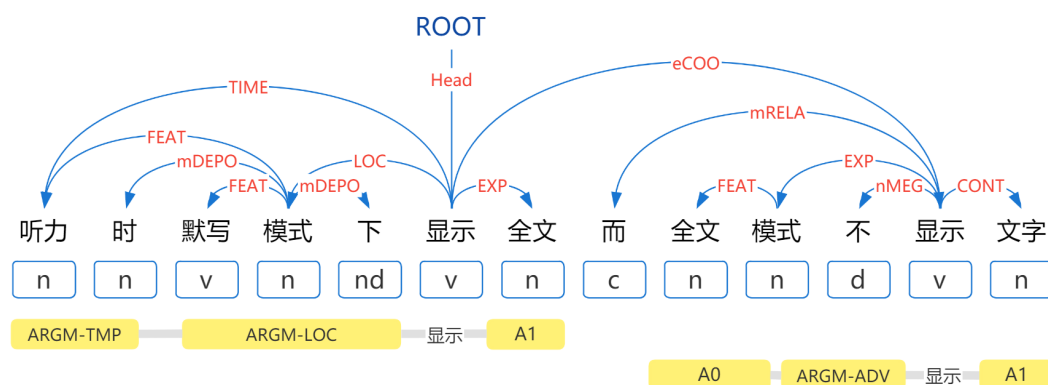


图 4-4: 测试报告句子语义分析实例

最后一类是用户直接与系统的交互，该动作在句子中没有谓词的论元，由一些特定的动词表示。如“登录”、“注册”这类的动词。

众包测试报告的本质特征是测试报告，既然是测试报告必然需要描述缺陷错误，这是最本质的要求。在进行众包测试报告质量评估时，针对文本内容的错误检测也是不可或缺的。最基本的是是否进行了缺陷描述，然后才是缺陷描述是否完整以及清晰等附加值。因此本文针对众包测试报告中的缺陷描述提出两个规则。

首先是存在性的判断，即该测试报告中是否含有了系统缺陷描述。其次是

缺陷描述中是否详尽。基于句子的分析，第一步先判断句子中是否存在相应的缺陷描述，这依赖于相关众包测试报告缺陷描述的词表。本文总结的众包测试报告领域相关词表见 3.3.3 分析性指标节。第二步有了系统缺陷描述之后，利用句子依存关系分析，错误的描述是否有修饰关系，或者一些附带的解释说明。

规则 4.12 RA3.1(系统缺陷): 存在系统缺陷描述

关于众包测试报告中是否含有系统缺陷描述的规则，是众包测试报告质量度量中本质要求。众包测试报告中对于缺陷的描述方式总体可以分为两类。

一类是“否定 + 动作”描述错误的存在，表示当前的动作操作没有完成。这里的动作可能是用户的动作，比如发送消息、用户体验；也可以是系统的动作，比如刷新，加载。具体的实现是利用语义角色标注找该句中主要核心动作，如果含有用户动作词或者系统动作词，则再利用语义依存关系分析判断该动作是否有否定关系（mNeg）进而判断“否定 + 动作”描述错误的存在性。例如图 4-4 中展示的“显示”与“不”之间的语义依存关系是否定关系（mNeg）。

另一类是错误词，比如“奔溃”、“宕机”、“闪退”这类名系统错误词用来描述错误。这类的判断相对容易很多，只要判断句子中是否含有系统错误词表中的词汇即可。还有部分页面布局缺陷词，如“重叠”、“过大”、“过小”这类属于页面布局缺陷的 bug 问题描述的判断是类似的。

规则 4.13 RA3.2(系统缺陷): 清晰无歧义的系统缺陷描述

在 RA3.1 的基础上，进一步判断众包测试报告中是否含有详细的、更加具体的缺陷描述。更高质量的描述能够提高审核人员审核效率，帮助其更好的定位 bug。检测到了错误描述，则根据句子依存关系分析这描述中含有的多少修饰关系及补充关系。修饰关系和补充关系越多表明该在接口元素的描述越加清晰无歧义。这点与接口描述清晰无歧义是相类似的。

分析性规则都是针对众包测试报告中除了环境信息以外的文本内容的，以句子为基本单位。其中接口元素 RA1 规则 and 用户行为 RA2 规则主要用于众包测试报告中复现步骤判断，系统缺陷主要用于测试报告中的关于缺陷的相关描述。

分析性规则是整体是针对众包测试报告中文本描述部分的。为了更直观的理解上述各个规则，将列举假设的众包测试报告文本内容如表 4-7，帮助理解

分析性规则。

表 4-7: 分析性规则判断举例

编号	文本内容	分析性规则判断结果
TC1	1. 进入 APP 首页， 2. 点击’ 按钮’，	RA1: 接口元素描述不清晰 RA3.1: 不存在系统缺陷描述 RA3.2: 系统缺陷描述不清晰
TC2	1. 进入 APP 首页， 2. 点击’ 按钮’， 3. 系统提示异常，	RA1: 接口元素描述不清晰 RA3.2: 系统缺陷描述不清晰
TC3	1. 进入 APP 首页， 2. 点击底部’ 我的按钮’， 3. 系统提示异常，	RA3.2: 系统缺陷描述不清晰
TC4	1. 进入 APP 首页， 2. 点击底部’ 我的按钮’， 3. 如图一所示, 系统提示异常， 4. 用户无法正常登录也无法退出该界面	

TC4 是高质量的众包测试报告文本内容，语义描述清晰句子语法正确，且具有详细描述 bug 的复现步骤和缺陷描述。TC1 与之相比较，接口元素“按钮”描述的不够详细，没有足够修饰，而且没有关于缺陷的描述，这是一个不合格的测试报告文本内容。TC2 与 TC1 相比，增加了错误描述，“系统提示异常”。但是并没有很详细的说明或者补充，这会造成歧义。TC3 与 T2 相比增加了“底部”对于接口元素的修饰，描述的更加详细了。这带给审核人员更多的提示，使得审核人员能够更好的复现 bug。与 TC4 相比，错误的描述还是不够完整。以上 4 个文本内容仅是简单的例子协助理解本文提出的分析性规则。

4.4.4 规则与期望属性关系

根据度量指标与期望属性的对应关系，现将各规则与度量指标以及期望属性之间的对应关系总结如表 4-8。

表 4-8: 规则与期望属性的关系

类别	判断规则	期望属性				
		原子性	简洁性	完整性	可理解性	可再生性
形态学规则	RM1	●	×		●	
	RM2				×	
	RM3.1				●	
	RM3.2		×		●	
相关性规则	RC1.1	●	●	●	×	×
	RC1.2				●	
	RC2			×		×
	RC3.1			×	×	×
	RC3.2			×	●	●
分析性规则	RA1		●		×	×
	RA2	●		×	●	×
	RA3.1	×		×	●	×
	RA3.2		●		×	×

其中“×”表示指标和属性之间存在直接的关系,而“●”表示指标和属性之间存在间接的关系。

本文提出的规则可以根据其适用的指标分为三大类，形态学规则、相关性规则和分析性规则。每一类规则立足于其度量指标，进行的可量化的、可判断的众包测试报告的质量评估，最终目的建立面向众包测试报告五个期望属性的评估。本文提出的基于句子分析的众包测试报告质量度量方法能够实现对众包测试报告在原子性、简洁性、完整性、可理解性和可再生性的评估。规则是立足于度量指标的，因此规则与期望属性的关联关系来源于度量指标与期望属性之间的关系，参见表 4-5。

4.5 质量得分设计

测试报告质量度量方法的第一步是在现有研究基础上，结合广泛的实际调研得出众包测试报告的期望属性。在期望属性的基础上，从分类学角度分别在形态学、相关性、分析性三个角度进行研究，进一步提出期望属性对应的度量指标。度量指标是为了更好的量化期望属性。度量指标可以看做是抽象层面

的，指导我们如何进行度量，但是具体的实现需要相应的技术手段。测试报告质量度量方法的第二步则是确立采用的规则归纳方式实现具体的度量指标。我们针对每个类别的度量指标提出对应规则，并总结这些规则和众包测试报告期望属性之间的关系。测试报告质量度量方法的最后是质量得分设计。在规则构建基础上确立众包测试报告的得分计算规则，并最后以雷达图的形式展示一份众包测试报告在五个期望属性上的最后得分。除了雷达图进行展示以外，测试报告质量度量技术还将进行众包测试报告评估结果以 PDF 格式文件进行输出，便于后续相关工作的真实性核对。

表 4-9: 规则得分详情

类别	规则	期望属性					得分
		原子性	简洁性	完整性	可理解性	可再生性	
形态学	RM1	1	1		1		3
	RM2				2		2
	RM3.1				1		1
	RM3.2		1		1		2
相关性	RC1.1	1	1	1	1	2	6
	RC1.2				1		1
	RC2			1		2	3
	RC3.1			2	1	2	5
	RC3.2			2	1	1	4
分析性	RA1		1		2	2	5
	RA2	1		2	1	2	6
	RA3.1	2		2	1	2	7
	RA3.2		1		2	2	5
得分		5	5	10	15	15	

我们按照规则与期望属性之间的直接关系和间接关系，以及规则的重要程度完成众包测试报告质量得分设计。其中分析性规则是针对众包测试报告复现步骤和缺陷描述的质量度量，即描述字段相比于其余两个字段更加重要，因此我们将其得分所占比比例设计的占比最多。在规则基础上，结合众包测试报告广泛调研，我们设计规则质量得分详情如表 5-3。

得分计算规则如下：对一份众包测试报告利用规则集进行规则判断。如果

当前测试报告质量满足当前规则，则获得该规则对应的得分。同时我们按照表 5-3 中所示的期望属性得分进行统计。因此我们可以获得这份众包测试报告在 5 个期望属性上的质量得分。可以从表中得知，5 个期望属性总得分为 50 分。其中各个期望属性占比见图 4-5。

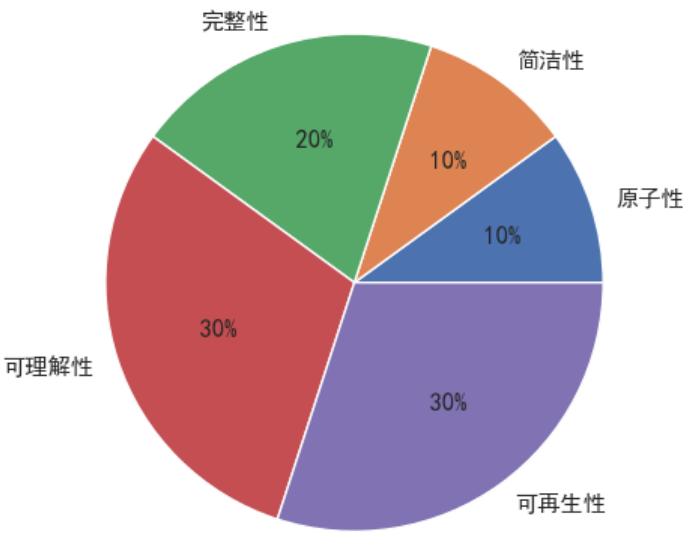


图 4-5: 众包测试报告

在众包测试活动中，对审核人员而言，测试报告最基本的要求是该报告中含有 bug 描述并且根据其描述 bug 能够被复现；对于测试工人获得奖励的最低要求同样如此，当然这是在工人是诚实的，所提交 bug 是正确的前提下。关于工人诚信的有关研究不在本文的研究中，我们认为工人诚信的提交了众包测试报告，其描述的 bug 是正确的。本文从 5 个期望属性出发度量众包测试报告的质量。总得来说，完整性、可再生性和可理解性的得分占有较大的比重，这是符合实际需求的。因此我们的得分计算规则是合理的，也是符合实际的。

我们设定众包测试报告质量得分从 0 分到 10 分，因此需要对 5 个期望属性的进行归一化处理。按照如下公式计算众包测试报告质量得分 $Score$ ，其中 $score_i$ 表示第 i 个期望属性得分，总计有 5 个期望属性。

$$Score = \sum_{i=1}^5 \frac{1}{5} \times score_i \tag{4-2}$$

按照以上得分计算规则，计算该报告的质量得分，并生成具有五个属性维度的雷达图的质量评估报告。基于句子分析的众包测试报告质量度量方法 CTRQSDP 的算法发如算法 4.1 所示。

算法 4.1 CTRQSDP

输入: TR : 测试报告集合

输出: TR' : 质量评估报告集合

```

1:  $T \leftarrow \emptyset$ ;
2: while  $t \leftarrow TR$  do
3:    $ws = WS(t)$ ;
4:    $ws = POS(ws, t)$ ;
5:    $t = DP(t, ws)$ ;
6:    $t = SDP(t, ws)$ ;
7:    $t = SRL(t, ws)$ ;
8:    $T \leftarrow T \cup \{t\}$ ;
9: end while
10:  $TR' \leftarrow \emptyset$ ;
11: while  $t \leftarrow T$  do
12:    $s \leftarrow 0$ ;
13:   while  $r \leftarrow RULE$  do
14:     if  $t \quad r$  then
15:        $s = s + Score(r)$ ;
16:     end if
17:   end while
18:    $tr \leftarrow t \cup s$ ;
19:    $TR' \leftarrow TR' \cup \{tr\}$ ;
20: end while
21:  $TR'$ 

```

CTRQSDP 旨在对众包测试报告进行质量度量，最后生成质量评估报告。在给定大小的众包测试报告集合上，CTRQSDP 首先对每一份测试报告进行预处理通过自然语言处理。预处理分为三个部分。首先是分词（WS）和词性标注（POS）；其次是语法依存分析（DP）；最后是语义角色分析（SDP）和语义角色标注（SRL）。接着用本文提出的规则集合 RULE 对每一份测试报告进行规则判断。如果符合了某一项规则，则获得相应规则的得分 $Score(r)$ 。最后统计每份测试报告在 5 个属性上的具体得分和总得分。生成每一份测试报告的质量评估报告。

4.6 本章小结

本章主要介绍了本文创新性提出基于句子分析的众包测试报告度量技术。首先是统领整个章节的质量评估框架，质量评估框架分为三大部分，即测试报告的期望属性和度量指标、质量评估规则构建和规则得分设计。

测试报告的期望属性和度量指标主要是阐述本文在现有软工文档质量评估相关研究基础上，提出关于众包测试报告的期望属性和度量指标。期望属性包括原子性、正确性、完整性、简洁性、可理解性和可再生性。期望属性往往是概念性的，每个期望属性的具体度量依赖于对应的度量指标。本文从分类学角度出发提出具体的度量指标，包括形态学指标、相关性指标和分析性指标。

在上述 3 大类质量指标中构建对应规则：形态学规则、相关性规则和分析性规则。特别地，利用句子分析技术，如依存语法分析、语义依存分析对众包测试报告文本内容进行词性标注、依存句法分析标注、附加语义角色标注以及语义依附标注，协助完成分析性规则构建。并分析规则与众包测试报告期望属性的关联关系。

在规则的基础上，我们进行众包测试报告质量评估得分设计。依据众包测试报告期望属性的不同重要程度，我们设计了每个规则具体得分和标准。采用雷达图从 5 个期望属性维度上展示众包测试报告的得分详情。我们提出了基于句子分析的众包测试报告质量评算法。CTRQSDP 的评估结果以及有效性和准确性将在第四章进一步验证和分析。

第五章 实验设计与分析

5.1 研究问题

在本节中，我们提出了三个研究问题以验证基于句子分析的众包测试报告质量度量方法（CTRQSDP）的性能。

问题一：CTRQSDP 与人工标注评分的差异程度如何？

对于众包测试报告质量度量来说，最重要的是评估结果和人工标注结果的差异程度，即准确性如何，因此我们首先提出了差异程度问题。问题一的提出是为了验证 CTRQSDP 对于众包测试报告质量评估能力如何。

为了回答问题一，我们在真实的众包测试报告数据集上，进行质量评估差异程度验证实验，将 CTRQSDP 的评估得分与人工标注得分进行对比，证明方法对众包测试报告质量评估的有效性。

问题二：可量化的质量指标能否有效度量众包测试报告的质量？

在 CTRQSDP 中，我们定义了一系列的指标，并按照分类学标准将其分为三类，包括形态学指标、相关性指标、分析性指标。从不同的角度度量众包测试报告的质量。不同的指标是否关注质量问题不同，指标是否能有效度量众包测试报告的质量。这些问题值得我们关注，因此我们提出了关于度量指标有效性的问题。

为了回答问题二，我们在真实的众包测试报告数据集上，进行度量指标有效性验证实验，分析 CTRQSDP 中各个质量指标对应规则发现的质量问题。证明本文提出的可量化指标能够有效度量众包测试报告的质量。

问题三：CTRQSDP 对不同类型的众包测试报告的准确性如何？

这里的不同类型主要是从两个角度来讲，第一个是不同 bug 类型的众包测试报告。不同类型的测试报告在错误描述上有些许的不同，我们的方法在不同 bug 类型的测试报告上质量度量的准确性如何。第二个是不同质量等级的众包测试。我们方法在不同质量等级测试报告上的准确性如何。从根本上说，这个问题旨在研究我们提出的方法有无偏向性，是否足够全面。这类问题研究对于方法的有效性分析是有帮助的。

为了回答问题三，我们在真实的众包测试报告数据集上，进行测试报告不同类型 CTRQSDP 评估准确性验证实验。通过分析 CTRQSDP 对于不同 bug 类型和不同质量等级的测试报告的评估准确度，证明本文提出的方法能够准确的评估不同类型众包测试报告的质量。

5.2 实验设置

5.2.1 实验数据集

本文实验数据集从慕测平台获取，我们共收集了 5 个移动应用 APP 的众包测试报告，APP 分别是 JayMe、SLife、途牛旅行、雅思听力和沪江英语。慕测平台将 bug 分为七类：功能不完整、不正常退出、页面布局缺陷、用户体验、性能、安全、其他 [11]。

表 5-1: 五个 APP 众包测试报告数据集

名称	版本	描述	数量
JayMe	1.0.5	粉丝服务 APP，包含最新资讯、互动交流、音乐交流、明星粉丝吧等功能。	1474 份
SLife	2.0.3	运动健身 APP，包含记录跑步步数、运动轨迹、消耗卡路里等功能。	1521 份
途牛旅游	9.0.0	旅游服务 APP，包含机票、酒店预订、加速抢票等功能。	3632 份
雅思听力	1.6.2	英语雅思听力练习 APP，包含雅思听力训练、全真模拟训练、学习榜排名等功能。	473 份
沪江英语	2.12.0	英语学习 APP，包含课程学习、评论交流等功能。	272 份

bug 类别的具体含义如下：功能不完整: 一些功能没有实现，如点击按钮没有反应；不正常退出：；页面布局缺陷：软件的布局页面的一些缺陷；用户体验：指的用户在使用产品过程中的一种纯粹的主观感受，但对于一个定义明确的用户群体，他们用户体验存在一定的共性；性能：这是软件的非功能需求，描述软件在响应性和稳定性方面的性能；安全：指的是可能导致未经授权或内部用户进入或破坏的安全风险；其他：难以划分为以上分类的。

表 5-1 中显示五个移动应用 APP 的众包测试报告数据集的详细内容。我们在这 5 个移动应用 APP 分别随机选取了 100 份报告，总计 500 份众包测试报告用于 CTRQSDP 有效性和准确性的验证实验。在随机抽取的 500 份众包测试报告中，5 个 APP 中 7 个 bug 类型占比如图 5-1。每个 APP 中 bug 类型占比不完全相同，总体我们可以发现功能不完整和用户体验两个类别数量最多。在 5 个数据集上，功能不完整和用户体验两类别相加所占比例至少 65%，最多 75% 以上。

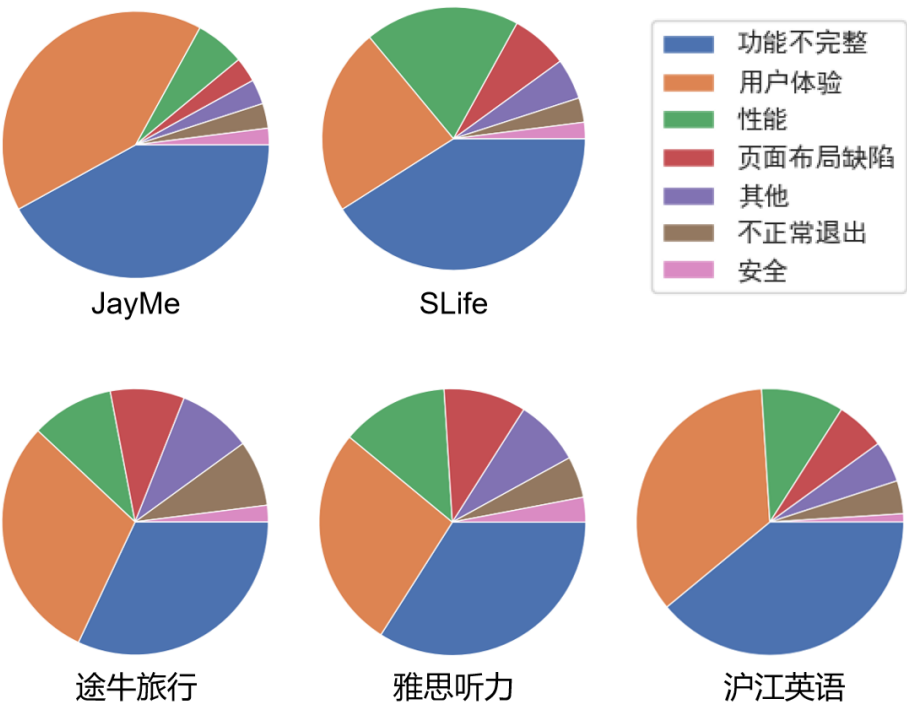


图 5-1: 众包测试报告数据集 bug 类型比例图

我们收集了 500 份众包测试报告，并邀请了 5 名平台审核人员来评估测试报告的质量。每份测试报告由两名审核人员根据预先定义的标准以 10 分的尺度进行打分。交叉验证尽可能的保证人工评分的客观性。我们以平均分作为这份众包测试报告的最终得分。如果两人评分相差差距超过了 2 分，我们将邀请第三个人对这份众包测试包进行重新评分。特别地，如果一份报告没有缺陷描述，那么这是一份不合格的众包测试报告，其得分为 0 分。并按照得分进行质量等级分类。8 分以上是优秀，6-8 分是良好，4-6 分是一般，4 分以下是差。在审核人员的帮助下，我们获得了关于 5 个众包数据集的质量评估得分。各个质量等级的测试报告数量如表 5-2。

表 5-2: 实验数据集

数据集	优秀	良好	一般	差
JayMe	6	21	61	12
SLife	17	54	22	7
途牛	3	23	49	25
雅思听力	9	38	38	15
沪江英语	12	49	29	10

5.2.2 实验方法

我们利用交叉验证的方式对实验数据集进行了质量评分并按照质量得分划分了质量等级，评估结果见表 5-2。接着利用 CTRQSDP 对 500 份众包测试报告进行质量评估，按照规则计算实验数据集中的每一份测试报告的质量得分。同时将 500 份报告分为 5 组，分别由人工按照众包测试报告的 5 个属性，即原子性、完整性、简洁性、可理解性以及可再生性上进行众包测试报告质量评估标注。我们设定一份测试报告的质量得分为 0 分 10 分，每个属性各占 2 分。特别地，没有缺陷相关描述表明这是一份不合格的众包测试报告，则其得分为 0 分。

此外，例如可理解性判断则是评估人员根据文本阅读经验判断报告描述清晰与否。可再生性评估则是评估人员根据经验判断文本中复现步骤是否描述清晰，是否具有进行复现的可能。为了验证 CTRQSDP 的有效性和准确性，我们进行了三个验证实验分别回答三个研究问题。第一个验证实验是关于 CTRQSDP 评分与人工评分的误差如何。第二个验证实验是关于本文提出的质量指标的有效性。最后一个验证实验是关于 CTRQSDP 对于不同质量等级类型和不同 bug 类型的众包测试报告的准确性如何。

5.2.3 实验环境

如表 5-3 所示，所有实验均在配备 1 个 NVIDIA GTX 1080Ti GPU，2 个 12 核处理器 “Intel(R) Core(TM) i7-6850K CPU @ 3.60GHz” 和 64GB 物理内存的 Ubuntu 16.04.5 LTS 服务器上进行。我们使用的编程语言是 python，版本为 3.6。实验环境还包含了第三方包，其中 ltp 是哈工大中文自然语言处理；Matplotlib 是数据图形化 2D 绘图库；Seaborn 是基于 Matplotlib 进行更高级 API

表 5-3: 实验环境

环境	名称	版本
CPU	12 核 Intel(R) Core(TM)	i76850K CPU@3.60GHz
GPU	NVIDIA	GTX 1080Ti
操作系统	Ubuntu	16.04.5LTS
软件	LTP	4.1.5
	Matplotlib	3.3.4
	Seaborn	0.11.2
	Numpy	1.19.5
	Reportlab	3.6.3
	Tqdm	4.62.3
	Termcolor	1.1.0
	Pandas	1.1.5

封装 2D 绘图库；Numpy 是一种开源的数值计算扩展, 支持大量的维度数组与矩阵运算；Reportlab 用于生成 pdf 版本质量度量报告；Tqdm 是一个快速, 可扩展的进度条；Termcolor 用于控制台标注文本颜色；Pandas 是基于 numpy 的解决数据分析任务工具。

5.2.4 实验评估

对于 CTRQSDP 质量评估实例有了了解后，我们接下来将验证其准确程度如何。我们将进行 3 个验证实验，对 CTRQSDP 的有效性和准确性进行验证和分析。在进入实验结果分析之前，我们先介绍下实验验证中使用的评估指标。我们使用的评估指标有三个，包括统计学中的平均绝对误差和标准差，以及准确性。

对于数据集中的每一份测试报告 $x_i(i \in \{1, 2, ..., n\})$, 我们计算预测得分 $\hat{y}_i$ 并利用下述指标进行评估。

平均绝对误差（mean absolute error, MAE），是标签值与预测值之差绝对值的和取平均，表示了预测值的平均误差幅度，而不需要考虑误差的方向，其公式如下所示：其中 x_i 为预测值， y_i 是标签值。主要用来评估 CTRQSDP 计算得分与人工标注得分的平均绝对误差。

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (5-1)$$

方差 (variance, VAR), 用于表现一组数据本身的离散程度。 $\bar{x}$ 为数据平均值。主要用于衡量误差的离散程度。方差越大表示数据在平均值附近波动越大; 方差越小数据越稳定

$$\text{VAR} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 \quad (5-2)$$

不同的开发者通常对同一份测试报告的质量有不同的看法。。基于人这种严格程度的差异性以及客观存在性, 众包测试报告质量评估分数差异在 2 分以内是允许的, 因此, CTRQSDP 的分数与人工标注的分数在 2 分的误差之内是准确评估的。在本文研究中, 我们采用准确性作为评价 CTRQSDP 性能的指标。假设 m 代表被 CTRQSDP 正确预测的测试报告的数量, n 是数据集中测试报告的数量, 计算公式如下。

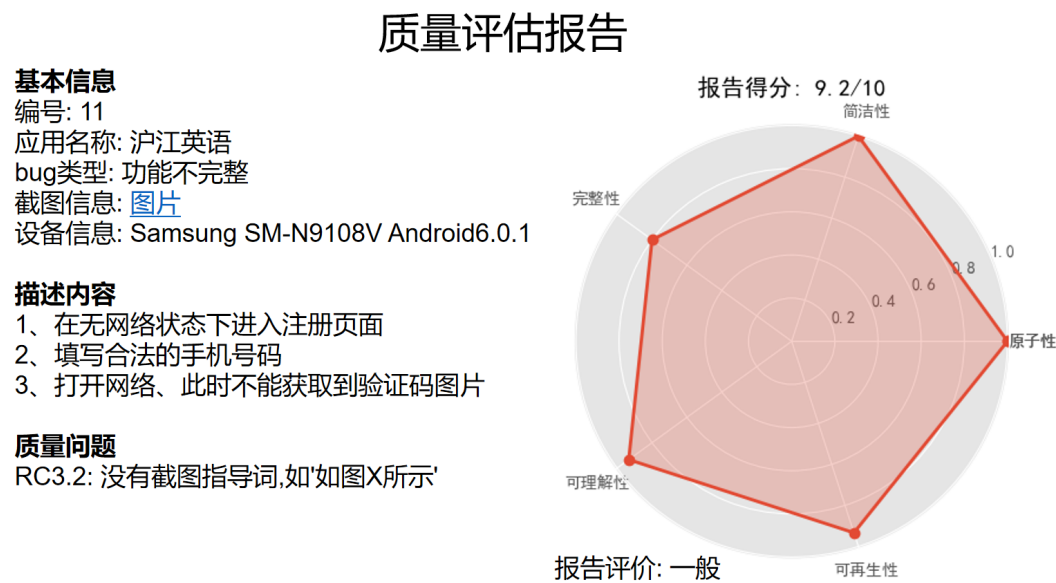
$$\text{Accuracy} = \frac{m}{n} \quad (5-3)$$

5.3 实验结果分析

5.3.1 CTRQSDP 评估结果实例分析

本文提出的方法最后会以雷达图的形式展示一份众包测试报告在五个期望属性上的最后得分。除了雷达图进行展示以外, 测试报告质量度量技术还将进行众包测试报告评估结果以 pdf 格式文件进行输出, 便于后续相关工作的真实性核对。我们分别例举了优秀、良好和一般的测试报告各两个进行规则实例分析。CTRQSDP 对于每一份测试报告最终会生成质量评估报告。CTRQSDP 质量评估报告实例包括优秀等级实例如图 5-2、良好等级实例如图 5-3、一般等级实例如图 5-4。

质量评估报告中包含的内容有报告编号、众包测试报告内容、存在的质量问题, 最后得分、质量等级和雷达图。其中众包测试报告内容包括环境信息、截图信息和文本内容描述。截图信息是图片链接。雷达图可以帮助我们更好获



(a) report11



(b) report45

图 5-2: CTRQSDP 评估实例-优秀

悉测试报告在 5 个期望属性上的评估情况。图 5-2中例举了两份质量优秀的众包测试报告。

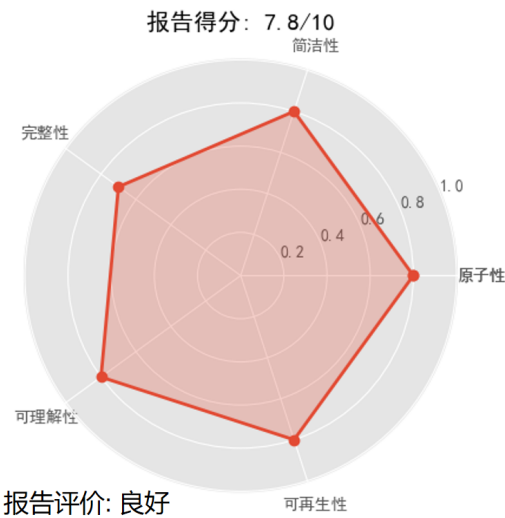
可以看出其结构是相对完整并且内容描述清晰, 只是在细节处存在一些不足。report11 测试报告中没有使用截图引导词, 不满足规则 RC3.2, 因此被扣了 0.8 分最后得分是 9.2 分。report45 测试报告则没有进行分点描述, 违反的规

质量评估报告

基本信息
编号: 60
应用名称: 途牛
bug类型: 功能不完整
截图信息: [图片](#)
设备信息: HUAWEI H60-L11 Android4.4.2

描述内容
选择分享到微博, 页面跳转至微博登录页面, 但页面下方提示“页面出错”, 点击“确定”按钮, 提示“文件不存在”

质量问题
RC1.1: 没有使用指示符分点描述
RC1.2: 指示符使用混乱
RC3.2: 没有截图指导词,如'如图X所示'



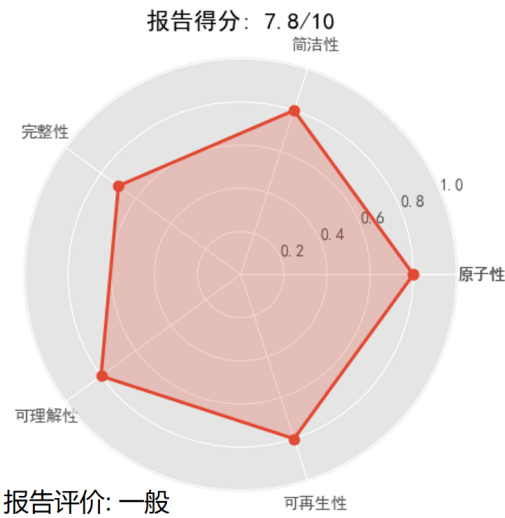
(a) report60

质量评估报告

基本信息
编号: 92
应用名称: 沪江英语
bug类型: 功能不完整
截图信息: [图片](#)
设备信息: Samsung Galaxy S6 Android5.1.1

描述内容
进入播放语音界面, 进行播放语音时, 语音进程条一直显示00, 没有语音的长度, 没有随着时间的变化而变化

质量问题
RC1.1: 没有使用指示符分点描述
RC1.2: 指示符使用混乱
RC3.2: 没有截图指导词,如'如图X所示'



(b) report92

图 5-3: CTRQSDP 评估实例-良好

则涉及两条 RC1.1 和 RC1.2, 被扣了 1.4 分总得分 8.6 分。两份优秀等级的测试报告的雷达图在 5 个期望属性维度是均衡的, 呈现五边形形状且五边形面积较大。这意味着这两份报告极大的满足了 5 个期望属性, 是具有高质量的, 事实也是如此。CTRQSDP 对于这些细节的判断是很严格的, 以上两份众包测试报告虽然细节上有着瑕疵, 但是从结构完整性和内容描述质量来讲都算是高质量

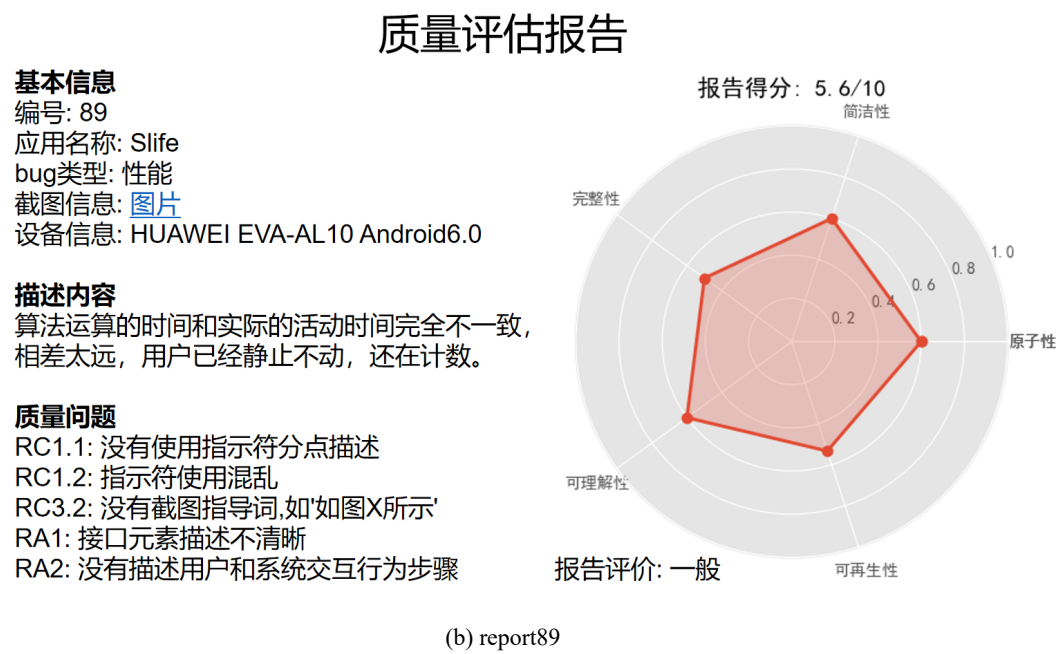
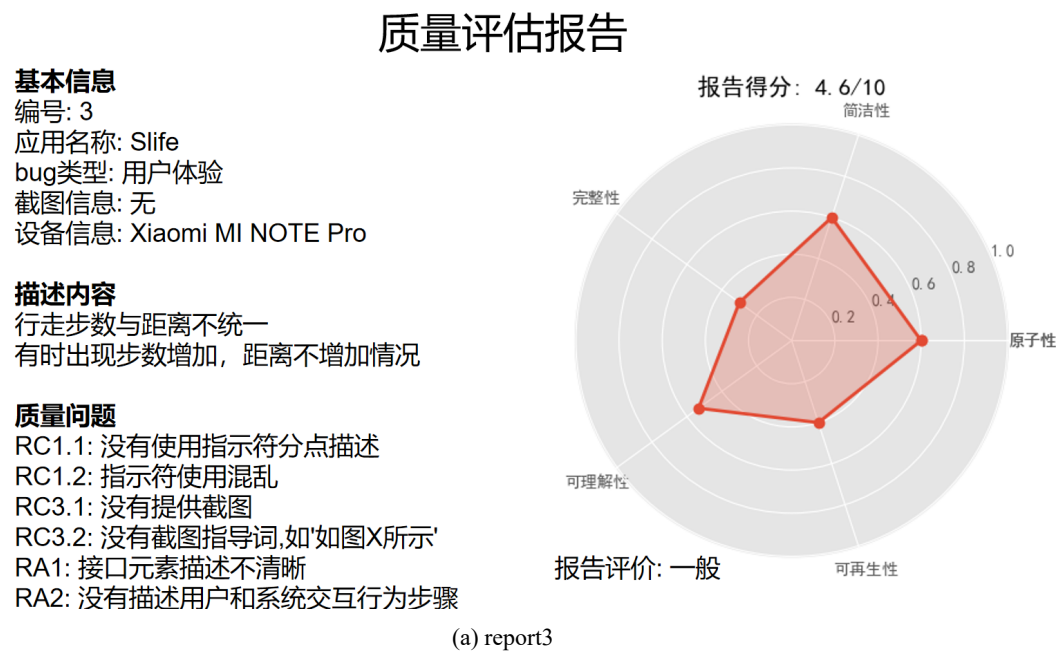


图 5-4: CTRQSDP 评估实例—一般

众包测试报告。这类报告缺陷描述清晰, 还能够很好地指示 bug 复现步骤, 能更好的引导审核人员完成 bug 复现和核对。

从图5-2中能看出, 优秀等级的众包测试报告所占比例 10% 左右, 极大部分的众包测试报告的等级为良好和一般。图5-3中例举了两份质量良好的众包测试报告。report60 违反了 RC3.2、RC1.1 和 RC1.2 三条规则, 被扣了 2.2 分。

与 report11 和 report45 相比较, report60 包含他们共同存在的质量问题。雷达图可以看出也是面积较大且均衡的五边形, 很好的满足了期望属性。report92 与 report60 的质量问题相同。相比于优秀的众包测试报告, 良好等级的存在的质量问题同样是一些描述细节, 只是存在问题的数量相对较多。对于众包测试平台的审核人员而言, 无论优秀等级还是良好等级均是较高质量的众包测试报告。

图 5-3 中例举了两份质量一般的众包测试报告。质量一般的众包测试报告除了细节问题外, 还存在文本内容描述不清晰的情形。除了上述细节问题 report3 还违反了 RC3.1、RA1 和 RA2。report3 文本内容只对 bug 进行了描述, 没有描述接口元素或者交互行为, 意味着该 bug 的复现步骤一定程度上是缺失的。这很不利于审核人员进行 bug 复现。此外还没有截图信息, 进一步加剧了复现理解难度。report89 也是相类似的情形, bug 复现步骤描述不清晰, 但是有截图信息, 一定程度上能够辅助审核人员理解其描述内容。从雷达图中也可以很好的观察到这一点, report89 比 report3 在完整性上得分更高。截图信息是众包测试报告重要结构, 直接影响众包测试报告的完整性。

此外, 我们统计发现, 很多测试工人尤其是新手在复现步骤描述上没有引起足够的重视, 很容易出现图中这样的例子, 没头没尾的描绘了缺陷问题而忽略了复现步骤的描写。因此对于测试工人开展测试报告基本质量相关培训是很有价值的, 我们的质量评估方法能够很好引导测试工人什么是好的众包测试报告, 具有重要的现实意义。

5.3.2 实验一: CTRQSDP 误差评估

实验一是为了验证 CTRQSDP 评分结果与人工标注评分之间的误差如何。CTRQSDP 在 5 个 APP 众包测试报告上进行质量评估, 将评估结果与人工标注评分进行比较。我们将采用平均绝对误差、标准差作为评估指标, 对 CTRQSDP 质量评估得分与人工标注得分进行误差分析。

在上一节对于 CTRQSDP 质量评估结果实例有了了解后, 我们接下来将验证 CTRQSDP 误差程度。为了更加具体的验证 CTRQSDP 质量评估性能, 我们将 CTRQSDP 评估得分与人工标注得分的误差绝对值进行分级, 即误差等级, 用于表示误差的大小, 从 A 到 D 递增。具体地说, 误差小于 1 为 A 等级; 误差 1 到 2 之间为 B 等级, 误差 2 到 3 为 C 等级; 误差 3 到 4 为 D 等级。CTRQSDP 与人工标注数据误差等级与数量关系图 5-6。横坐标表示误差的等级, 从 A 到

D 共计四个等级。纵坐标表示误差等级中众包测试报告的数量。

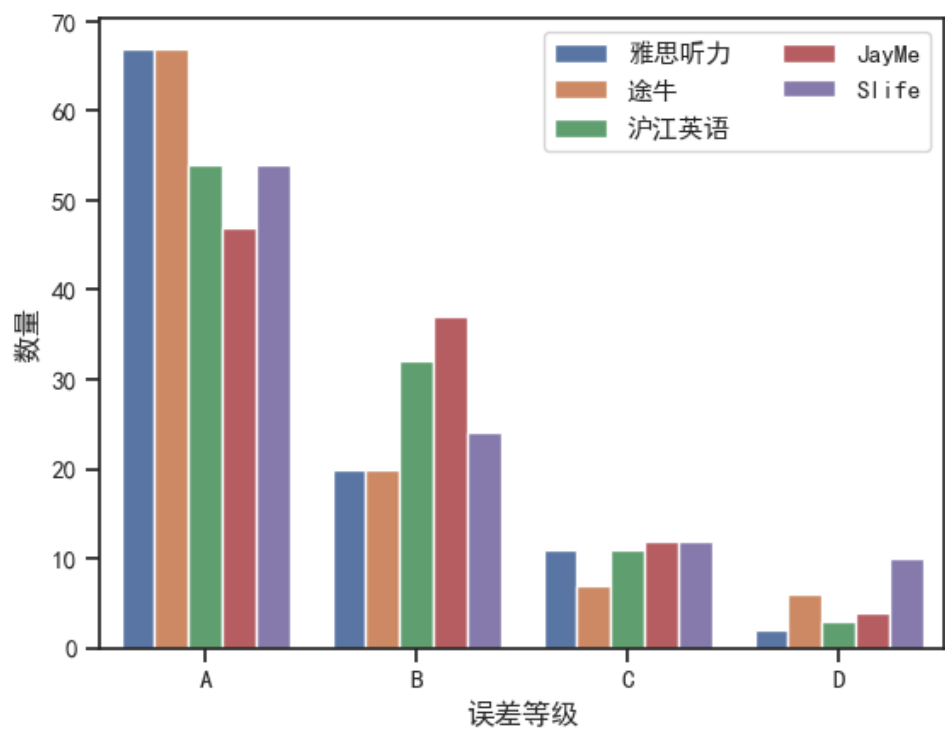


图 5-5: 不同误差等级测试报告数量

CTRQSDP 在雅思听力、途牛两个 APP 数据集上的质量评估要好于沪江英语、JayMe、Slife。尤其是在误差小于 1 的报告数量上，雅思听力、途牛比沪江英语、JayMe、Slife 多 10 份。对于 5 个 APP 来说误差等级为 A 或 B，即 CTRQSDP 评估误差小于 2 分的众包测试报告数量均在 78 到 87 之间。5 个 APP 数据上误差等级 A 和误差等级 B 相加所占比例平均值为 84.4%。特别地，雅思听力和途牛在误差等级 A，即相差 1 分的测试报告数量均有 87 份，占比高达 87%。5 个 APP 数据上误差等级 A 所占比例平均值为 57.8%。

我们的方法在不同数据集上的评估效果略有不同，总体来说，我们的方法能够准确的评估众包测试报告的质量。说明我们的方法在绝大部分情况下都是适用的。剩下的 15% 左右的测试报告，我们的方法则与人工评审有着较大的误差，3 到 4 分之间。这部分众包测试报告的质量等级多为 D 等级。这可能是我们的方法打分更为严谨，对于低质量的测试报告更倾向不给分或者给低分。

该实验中，我们采用统计学中的平均绝对误差和标准差作为评估指标。CTRQSDP 在五个 APP 众包测试报告数据集上分别进行了评估误差程度的实

验验证。CTRQSDP 得分与人工标注得分详细误差如表 5-4。表中分别统计了 CTRQSDP 在 5 个 APP 数据集上的误差情况。

表 5-4: APP 得分误差结果

	平均绝对误差 (MAE)	方差 (VAR)
雅思听力	0.908	0.770
途牛	0.978	1.204
沪江英语	1.197	0.754
JayMe	1.212	0.796
Slife	1.366	1.213
平均值	1.132	0.947

由表中可知 CTRQSDP 在雅思听力 APP 的平均绝对误差最小为 0.908。Slife APP 的平均绝对误差最大为 1.366。CTRQSDP 在 5 个 APP 数据集上的平均误差为 1.132。此外 CTRQSDP 在 5 个数据集上的方差在 0.770 到 1.213 之间，说明误差值偏离程度不大，误差比较均衡，整体波动变化不大。不同的人对于文本质量的严格程度不同，特别是对于行为描述清晰或者缺陷描述清晰的评判分数上有些许的差异，这是客观事实。同一份测试报告由不同的人进行质量评估也会有误差。基于这种差异性，我们认为 5 个 APP 上的平均误差为 1.132 是合理的，这证明了我们在众包测试报告质量评估上的准确性。

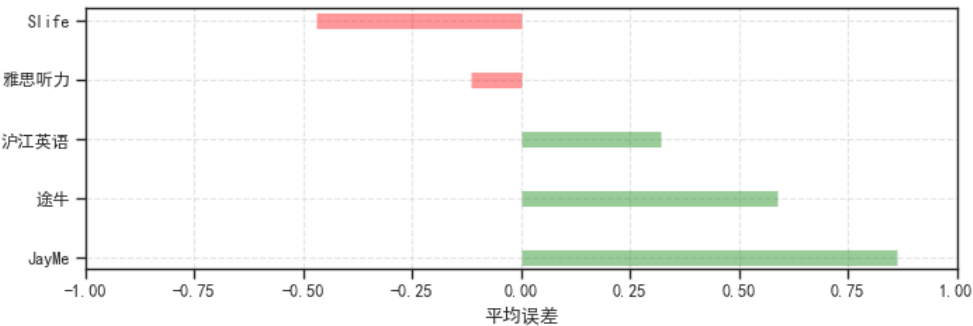


图 5-6: CTRQSDP 评分与人工评分平均误差

图 5-6 中展示了 CTRQSDP 和人工标注在 5 个 APP 数据集上平均误差。可以发现 SLife 和雅思听力数据集上，平均误差为负数，意味着 CTRQSDP 质量评分要比人工标注评分要低。为了更加直观的观察 CTRQSDP 质量评分与人工标注评分之间的分布情况，绘制 CTRQSDP 和人工标注拟合分布图如图 5-7。

可以观察得知，CTRQSDP 质量评分与人工标注评分整体上具有一致性。在 SLife 上，从报告分数来看，CTRQSDP 的质量评分曲线要稍低于人工标注曲线。这该数据集上平均误差为负数结论是一致的。在雅思听力上，CTRQSDP 的质量评分曲线要十分接近于人工标注曲线。在该数据集上，CTRQSDP 平均误差也最小。这证明着 CTRQSDP 质量评分在 SLife 和雅思听力数据集上是有效的。

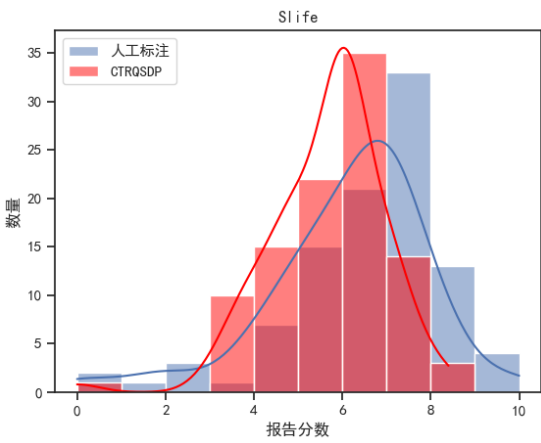


图 5-7: CTRQSDP 和人工标注拟合分布图—偏低

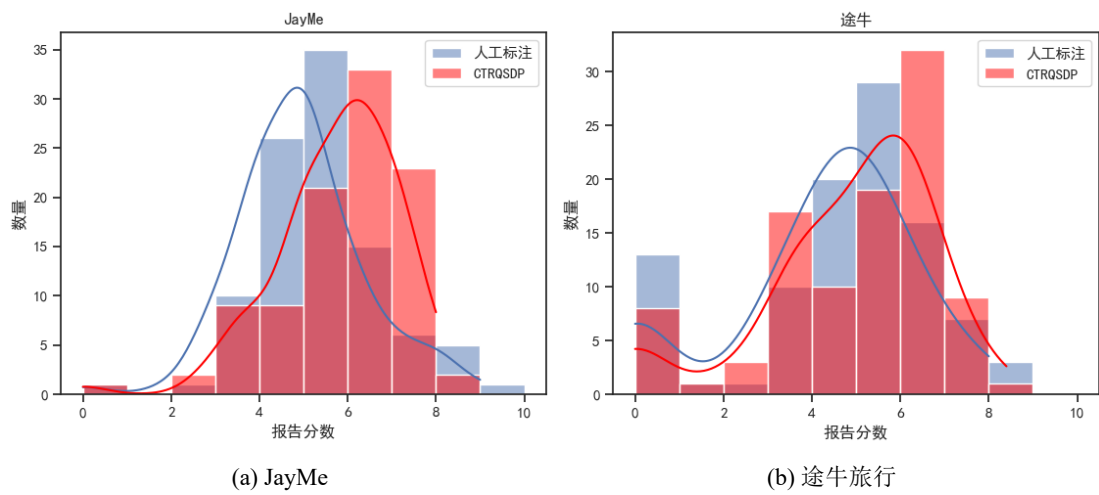


图 5-8: CTRQSDP 和人工标注拟合分布图—偏高

从图中，我们还可以得知图 5-6 可以发现沪江英语、途牛旅行和 JayMe 数据集上，平均误差为正数，意味着 CTRQSDP 质量评分要比人工标注评分要稍高。与上述方法相同，我们结合 CTRQSDP 和人工标注拟合分布图 5-8 进行实验结果的分析。可以观察得知，CTRQSDP 质量评分与人工标注评分整体上同

样具有一致性。在途牛旅行和 JayMe 数据集上，从报告分数来看，CTRQSDP 的质量评分曲线要稍高于人工标注曲线。这该数据集上平均误差为正数结论是一致的。其中在沪江英语数据集上，CTRQSDP 的质量评分曲线要十分接近于人工标注曲线。同时在该数据集上，CTRQSDP 平均误差与雅思听力数据集相类似比较小。这证明着 CTRQSDP 质量评分在沪江英语、途牛旅行和 JayMe 数据集上是有效的。

接下来我们详细分析 CTRQSDP 和人工标注拟合分布图。图 5-7、图 5-8 和图 5-9 中展示了测试报告由人工标注得分和我们方法计算得分在数量上的直方图，以及测试报告数量分布变化程度曲线。可以看出我们的评估得分在 0 分到 10 分间整体趋势与人工标注得分是具有一致性的。JayMe 和途牛数据中，我们的质量评估得分整体上比人工标注得分要高。Slife 中我们的质量评估得分整体上比人工标注得分要低。在雅思听力和沪江英语中评分与人工标注分布更接近。

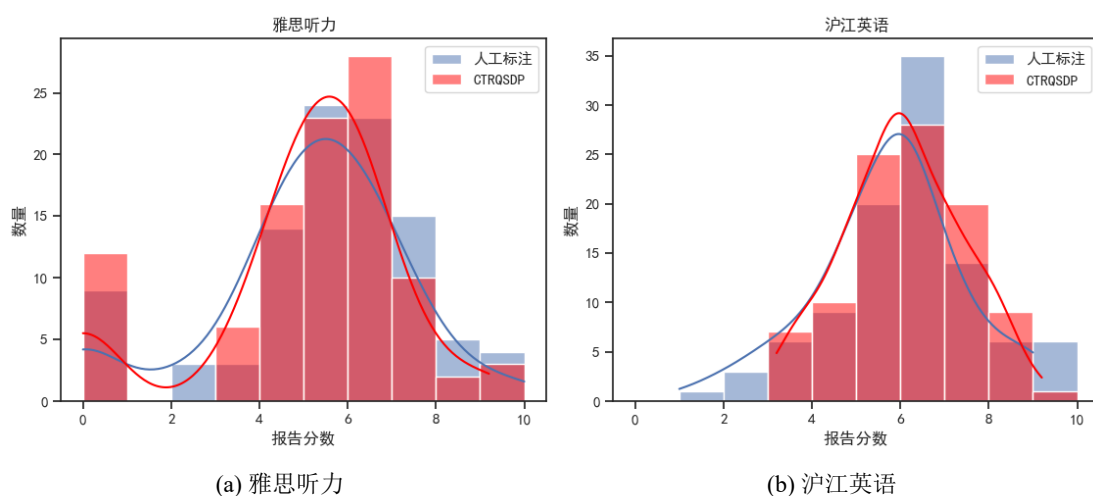


图 5-9: CTRQSDP 和人工标注拟合分布图-贴合

总体来讲，从测试报告得分分布上看，我们的方法打分倾向于给出高分，而人工标注往往低于算法打分，这可能因为我们的方法在满足判断规则时则会给予分数，而人工标注得分更为保守。这是因为相关人员除了考虑本身报告结构外，还会判断文本内容主旨一致性，这导致了人工标注得通常更为严格。综上，可以看出我们提出的 CTRQSDP 方法能够准确的评估众包测试报告质量。

5.3.3 实验二：质量指标有效性评估

实验二是为了验证本文提出的质量指标的有效性。我们在真实众包测试报告进行实验，用 CTRQSDP 检测其中可能含有质量问题，并与真实人工标注进行对比。我们将分析 CTRQSDP 中不同规则检测对于众包测试报告质量判断的能力，以及不同类别规则对于众包测试报告质量判断的能力。

针对众包测试报告是专业领域为软件测试且具有稳定文本结构的短文本的特点。采用规则判断众包测试报告质量有以下三个优势：首先是众包测试报告的专业领域强，其内容词表为软件测试领域，这使得规则制定空间范围小。本文提出的方法的具体实现是利用规则归纳。

利用规则检测的有效性能正确反映质量指标的有效性说明如下。其次众包测试报告文本结构稳定，文本内容通常包括复现步骤和缺陷描述，且句子形式较为规范，规则判断可用性高。最后众包测试报告是短文本，使用规则判断速度快。在第四章中详细的列举了每一个度量指标包含哪些规则。规则的可解释性表明如果违反这条规则同时违反了这个规则所表示的指标。因此，我们可以利用规则检测能力表示质量指标的检测能力，即通过证明规则检测能力的有效性来验证质量指标的有效性。

图5-10中展示了违反 CTRQSDP 规则的众包测试报告数量。RM1.1、RM3.1、RM3.1 检测到质量问题数量比较少。文本长度范围的规则和标点符号规则能够检测出质量问题。图中没有 RM2，是因为 RM2 没有检测出任何的质量问题。这表明针对的众包测试报告这类短文本，其文本内容相对短小精炼，文本表面可读性特征很难进行判断。更加说明了针对短文本分析性规则的必要性和重要性。

RC1.1、RC3.1、RC3.2 均能检测出质量问题，其中违反 RC1.1 和 R3.1 规则的测试报告数量在各个 APP 应用上有着较大的变化。可以从 APP 测试时间线观察发现：基于激励机制，测试工人为了提交测试报告获得更多的奖励，选择多提交截图信息使得众包测试更加完善；同时为了提升工作效率，选择减少分点描述节约时间成本。违反 RC1.2 规则的测试报告数量为 0，表明测试工人如果采用了分点描述，那么使用的分点标记是正确的没有混乱胡乱使用的情况。慕测平台会自动采用测试工人的设备信息，因此 RC2 规则没有检测出质量问题，所有的测试报告均含有环境信息。

分析性规则中的四个规则均能检测出质量问题。RA1 规则和 RA2 规则是针对复现步骤描述清晰与否的质量判断，可以看出绝大部分的测试报告有复现步

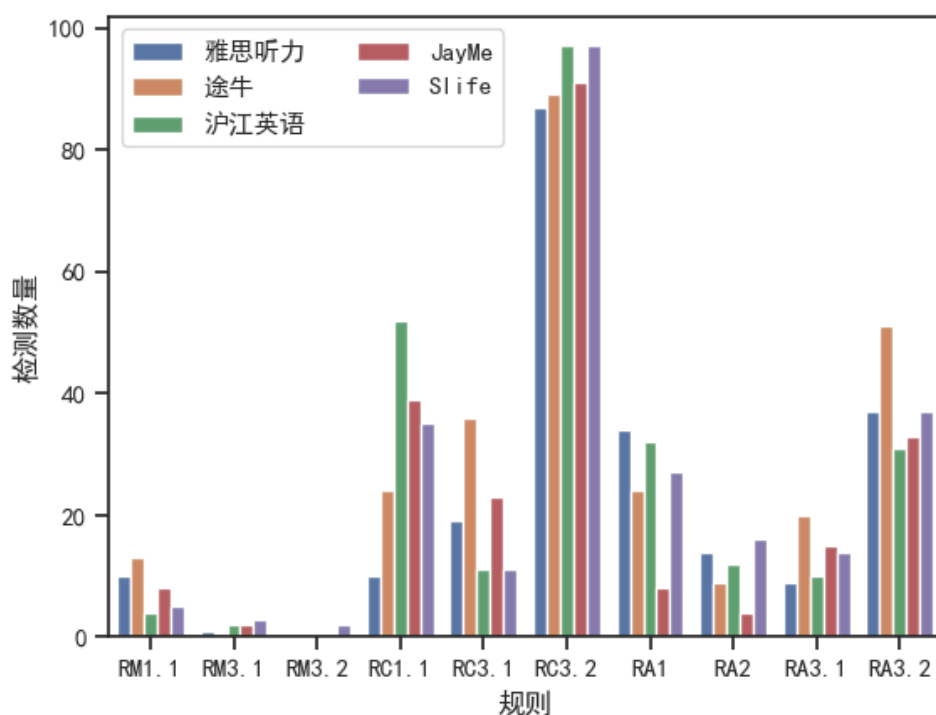


图 5-10: 违反规则的报告数量

骤的描述。将两个规则进行比较，我们可以发现测试工人更倾向于进行交互行为的描述，对于接口清晰描述上显得没有细致。RA3.1 规则和 R3.2 规则是关于缺陷描述判断，极大部分的众包测试报告有着缺陷描述。违反 RA3.2 规则的数量是 RA3.1 的两倍多，我们可以得出测试工人几乎都会进行缺陷描述，但是缺陷描述的质量还有待进一步提高。

从三个大类别规则来看以上结果。形态学规则、相关性规则、分析性规则对于众包测试质量度量结果如图 5-11。三类规则对应的分别是形态学指标、相关性指标和分析性指标。由图可知，违反形态学规则的测试报告占比约在 10% 左右。形态学规则主要从一些表面特征进行判断。不符合该类别的测试报告往往是不合格的，其文本内容描述为寥寥数个词汇，与 APP 测试也毫无关系，比如其文本内容为“默默默，测试”。

我们可以知道形态学规则作为文本内容表面特征判断对于众包测试报告质量度量能力是不足的，这说明了相关性规则分析性规则的必要性。违反相关性规则的测试报告较多，主要是大部分报告不满足截图信息引导描述进行的规则。可以得出大部分的测试工人编辑测试报告内容描述时还没有描述引导截图信息的习惯。雅思听力、途牛、沪江英语、Slife 数据集上有接近 60% 违反分析

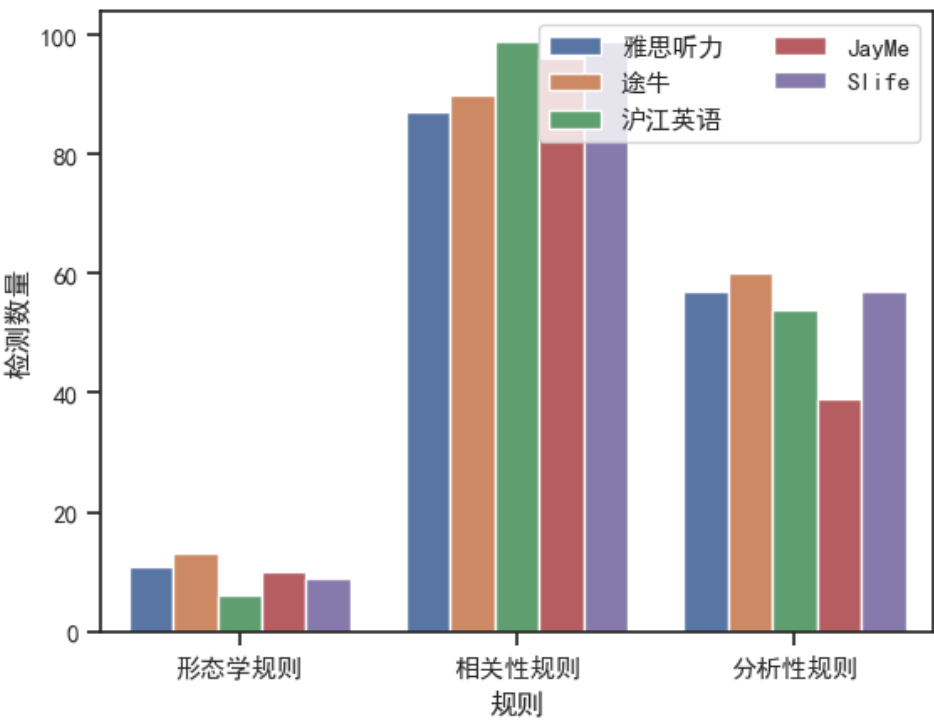


图 5-11: 违反三大类规则的报告数量

性规则，Slife 接近 40% 违反分析性规则。这证明 CTRQSDP 能够检测出在众包测试报告中的质量问题。综上，规则能够有效的检测出质量问题，由此可以证明我们提出的三大类质量指标是有效的。

5.3.4 实验三：CTRQSDP 准确性评估

实验三是为了验证 CTRQSDP 对于不同类型的众包测试报告的质量评估准确性如何。我们在真实众包测试报告数据集上进行实验，通过分析 CTRQSDP 在不同类型的众包测试报告上质量评估准确性，证明 CTRQSDP 能有效且准确的度量众包测试报告质量。

上述提到的不同类型的测试报告主要包括两类，一类是不同 bug 类型的众包测试报告。不同类型的测试报告在错误描述上有些许的不同，我们的方法在不同 bug 类型的测试报告上质量度量的准确性如何。另一类是不同质量等级的众包测试。我们方法在不同质量等级测试报告上的准确性如何。从根本上说，这个问题旨在研究我们提出的方法有无偏向性，是否足够全面。这类问题研究对于方法的有效性分析是有帮助的。同时将验证在不同类型测试报告上的质量评估准确度，因此也是对于 CTRQSDP 准确性验证的重要组成部分。

CTRQSDP 在不同质量等级上的评估准确度如图 5-12。横坐标是报告等级。纵坐标是准确度。对所有的数据集而言，质量等级为优秀的 CTRQSDP 准确率为 87.3%，良好等级的准确率为 93.5%，一般等级的准确率为 88.0%，差等级的准确率为 55.0%。在不同的 APP 数据集上有写不同，如雅思听力和 Slife 准确率最高的两个等级是良好和一般，而余下三个 APP 则是优秀和良好的准确率最高。特别地，途牛、JayMe 三个 APP 中针对优秀和良好等级的测试报告准确率为 100%。JayMe 中的优秀等级测试报告准确率为 100%。我们的方法在优秀和良好，一般三个等级上表现出色。然而在差等级上则有一些不足，我们的方法在 Slife 的差等级上准确率只有 28.5%，远低于其他水平，这可能是由于低质量报告，等级为差的报告特征不明显，相对而言更难判断。而高质量的报告由于其结构明显易于判断，因此我们的方法在优秀、良好和一般三个等级上的准确率高。

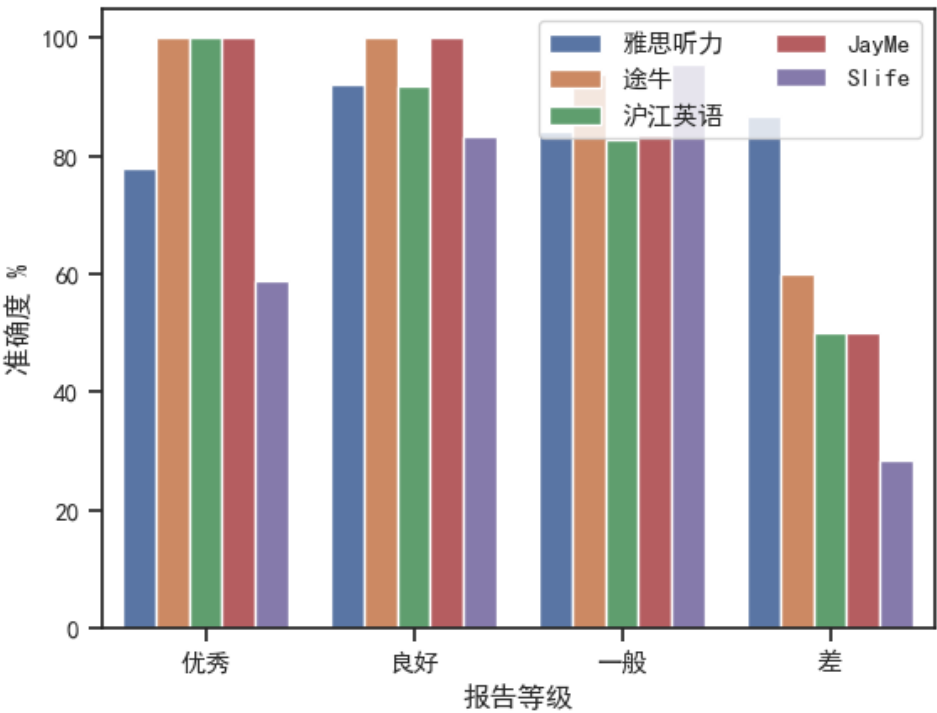


图 5-12: CTRQSDP 准确度

上述是针对 CTRQSDP 中不同规则检测对于众包测试报告质量判断的能力的分析，接下来我们将对 CTRQSDP 对不同 bug 类型的众包测试报告质量判断能力进行分析。众包测试报告按照 bug 类型的不同，kikbug 移动众包测试平台将其分为 7 类。主要包括：不正常退出、功能不完整、安全、性能、用户体

验、页面布局缺陷和其他。不同的 bug 类型，文本内容描述使用的句子格式有着差异。由 4.2.1 实验数据集介绍可知：功能不完整和用户体验占比平均 70% 以上。CTRQSDP 不同 bug 类型检测成功数量如图 5-13。由图可知，CTRQSDP 对于不同 bug 类型的众包测试报告质量问题检测成功率平均为 84%，其中针对功能不完整和用户体验两个类型的成功率分别高达 86.7%，86.2%。这证明了 CTRQSDP 能有效度量众包测试报告中的质量问题。

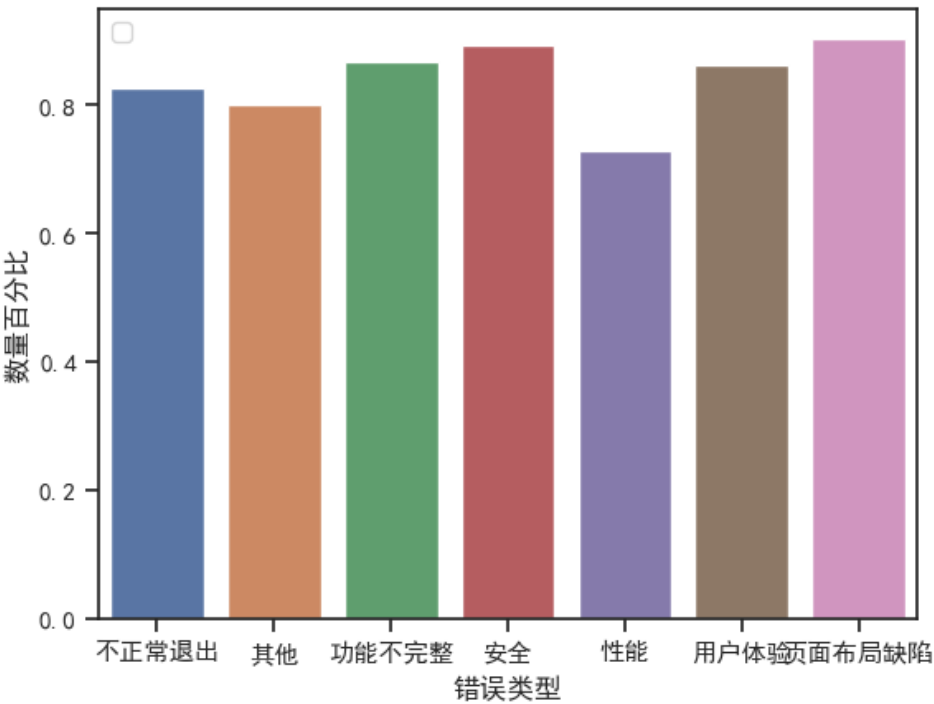


图 5-13: 不同 bug 类型检测成功数量

5.4 有效性分析

与任何学术研究一样，在本文中也存在一些因素会影响研究结果的有效性，包括数据集选择、自然语言选择和术语列表。

5.4.1 构建有效性

分析性指标中我们采用了一些用户定义术语表，这可能影响 CTRQSDP 的有效性。然而，许多研究已经总结了测试报告的术语表 [12, 13, 32]。在本文中，我们充分利用这些术语表并将其转化为中文，同时我们通过结合工业界

对测试报告的开展广泛调查，增加了一些相关术语，因此，其影响可以忽略不计。

5.4.2 内部有效性

在我们的实验中，所有的众包测试报告都是用中文编写的，CTRQSDP 是基于中文文本开发的。众所周知，中文与英语及其他拉丁语系有很大的不同，这可能对 CTRQSDP 在其他自然语言的推广产生威胁。然而，CTRQSDP 是通过句子分析从众包测试报告文本中提取相关语法和语义关系。句子分析仍然是一种自然语言处理技术，已经被证明可以被应用于其他自然语言。因此，这种威胁将被大大减少。

5.4.3 外部有效性

在我们的研究中，使用的所有众包测试报告数据集均是安卓应用程序众包测试。我们不能保证我们的结论适用于其他移动平台（如 iOS）。然而，考虑到我们的数据集来自于不同的领域，如音乐、健康、教育和旅游。我们相信我们的研究仍然给众包测试报告质量研究提供了一些有价值的信息。

5.5 本章小结

本章我们主要进行了三个验证实验，以分别评估 CTRQSDP 质量评估的有效性和准确性。实验数据来源于慕测平台。在 CTRQSDP 误差评估验证实验中，对 CTRQSDP 质量评估得分与人工标注评分进行误差分析，证明了 CTRQSDP 能够有效且准确的度量众包测试报告质量。在质量指标有效性验证实验中，通过对规则检测质量问题能力的分析，证明了所有规则对应的三大类质量指标能够有效实现对众包测试报告质量度量。在 CTRQSDP 准确性验证实验中，对于不同质量等级和不同 bug 类型的众包测试报告进行质量问题检测，并采用准确度实验评估指标，证明了 CTRQSDP 能够度量不同类型测试报告的质量，且具有较高的准确度。这进一步证明了本文提出的基于句子分析的众包测试报告质量度量方法是有效且准确的。

第六章 总结和展望

6.1 总结

众包被应用于软件测试领域以来，引起了学术界和工业界的广泛关注。在众包测试活动中，来自开放平台的测试工人会在规定的时间内以众包测试报告的形式提交发现的异常现象，平台中的审核人员将进行质量审核和缺陷验证。由于提交时间有着限制，比如几天或者一周。成千上万的测试报告被发送给审核人员，其数量严重超过了检查这些报告的可用资源。同时，众包测试人员大多为非专业测试人员，其提交的众包测试报告质量良莠不齐。

对于众测平台的审核人员，众包测试报告的质量越高，内容更加易于理解，可以帮助审核人员更好完成 bug 审核。反之，众包测试报告的质量越低，往往缺乏更多 bug 描述细节，耗费了审核人员的时间和精力，一定程度影响工作的效率。对于诚信的测试工人而言，他们提交的测试报告质量越高，其审核通过率越高相应获得报酬也越多。为了获得更多的报酬，他们也渴望对高质量众包测试报告进行了解和学习。因此，自动化评估众包测试报告质量对提升众包测试效益是具有重要意义的。

针对众包测试报告是专业领域为软件测试且具有稳定文本结构的短文本的特点。采用规则判断众包测试报告质量有以下三个优势：首先是众包测试报告的专业领域强，其内容词表为软件测试领域，这使得规则制定空间范围小。其次众包测试报告文本结构稳定，文本内容通常包括复现步骤和缺陷描述，且句子形式较为规范，规则判断可用性高。最后众包测试报告是短文本，使用规则判断速度快。

本文提出一种基于句子分析的众包测试报告质量度量方法。依照分类学方法分析总结众包测试报告 5 个期望属性，创新性地提出立足于句子维度的分析性度量指标，用于分析测试报告文本语义和文本语法。紧接着在度量指标基础上建立规则。特别地，利用句子分析技术，如依存语法分析、语义依存分析对众包测试报告文本内容进行词性标注、依存句法分析标注、附加语义角色标注以及语义依附标注，协助完成分析性规则构建。提出基于句子分析的众包测试

报告质量评估算法，用于众包测试报告质量评估。通过在 JayMe、SLife、途牛旅行、雅思听力和沪江英语 5 个 APP 的真实众包测试报告数据集上，进行了关于有效性和准确性的质量评估验证实验，证明了我们提出的众包测试报告句子分析质量评估方法是有效且准确。本文研究不仅仅协助审核人员实现自动化的测试报告质量评估，也为测试工人关于如何撰写一个好的测试报告提供了参考意见。

6.2 展望

由于时间和个人水平所限，本篇文章还有很多不足的地方需要进一步的改进，可作为未来拓展研究工作的方向与思路：

- (1) 本文的研究内容是提出一种基于句子分析众包测试报告的质量评估方法，还并没有形成可用的工具。因此，我们接下来的工作中还需要设计切实可行的工具来实现本文方法，并将这个工具部署在真实的场景中，协助审核人员进行测试报告审核，提高审核效率。
- (2) 本文提出的方法是基于众包测试工人是诚信且认真的这一假设。我们认为测试工人不会恶意提供含有错误描述，但是具有语言攻击，附带有情绪的众包测试报告。在众包测试活动中，大部分测试工人是诚实而认真的，但是这并不全面。如何应对这种情形，仍需要进一步研究和探索。
- (3) 本文提出方法的截图信息规则判断是判断众包测试报告中是否含有截图，没有对截图内容相关性进行判断。不诚信的测试工人有可能放入一些内容无关的截图。对于这类截图内容相关性研究，需要我们进行思考和分析。
- (4) 关于众包测试报告度质量度量结果以及评判标准如何反馈给测试工人，设计建立合理的激励机制引导测试工人提交高质量测试报告，这仍是一个具有挑战性的工作。

致 谢

研究生生活即将结束，回望过去的三年时光，感慨自己遇到贵人无数，在这里，我向所有关心、爱护和帮助过我的人们表示最诚挚的感谢。

首先我要感谢我的指导老师陈振宇老师。感谢陈老师对我的认可且接收我进入了 iSE 实验室，陈振宇老师渊博的专业知识、严谨的科研态度和永不懈怠的工作激情给了我研究的动力和对未来的憧憬。对待学术细致踏实的作风和对待事情细心认真的态度，是我学术上和生活中的榜样。再次感谢陈老师的指导和帮助。

感谢实验室的师兄、师姐、师弟、师妹和同学们，感谢李玉莹师姐在毕设和毕业论文上对我的帮助，陪我讨论毕设的实现方案，指导我修改毕业论文。在遇到困难时你们总会伸出援手，感谢你们对我的帮助，你们是我生活中的朋友，学术上的榜样。

感谢南京大学，虽然研究生阶段大部分时间身处疫情笼罩之下，但是正因为有南京大学正确的政策方针，这才守护了全校师生的安全。在此也特别感谢南京大学软件学院，是学院全体教职工的尽心尽责，让我在课堂上学到了专业的知识，体会到了南大老师的责任和担当。

感谢我的三个室友，三年充实快乐的宿舍时光是我学术生活中最大的调剂。遇到困难时本文一同互帮互助，遇到欢乐时本文一起共同分享。庆幸自己能拥有这份简单、珍贵的情谊。

最后且最重要的是感谢我的家人们和我的男朋友高新宇，无论我经历了什么，做出什么决定他们都一直在我身边支持我。在我最困难，最无助的时候，一直陪伴我，安慰我。他们就是我人生中的一盏盏明灯，让我有了在一次次的挫折后重新站起来，努力向前走的决心。我的每一点成绩都少不了家人的支持，我永远爱他们。

参考文献

- [1] HOWE J. The rise of crowdsourcing[J]. Wired Magazine, 2006, 14(6): 1–4.
- [2] WU W, TSAI W, LI W. An evaluation framework for software crowdsourcing[J/OL]. Frontiers Comput. Sci., 2013, 7(5): 694–709.
<https://doi.org/10.1007/s11704-013-2320-2>.
- [3] DOLSTRA E, VliegENDHART R, POUWELSE J A. Crowdsourcing GUI Tests[C/OL] // Sixth IEEE International Conference on Software Testing, Verification and Validation, ICST 2013, Luxembourg, Luxembourg, March 18-22, 2013. [S.l.]: IEEE Computer Society, 2013: 332–341.
<https://doi.org/10.1109/ICST.2013.44>.
- [4] KOMAROV S, REINECKE K, GAJOS K Z. Crowdsourcing performance evaluations of user interfaces[C/OL] // MACKAY W E, BREWSTER S A, BØDKER S. 2013 ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '13, Paris, France, April 27 - May 2, 2013. [S.l.]: ACM, 2013: 207–216.
<https://doi.org/10.1145/2470654.2470684>.
- [5] FENG Y, CHEN Z, JONES J A, et al. Test report prioritization to assist crowdsourced testing[C/OL] // NITTO E D, HARMAN M, HEYMANS P. Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, ESEC/FSE 2015, Bergamo, Italy, August 30 - September 4, 2015. [S.l.]: ACM, 2015: 225–236.
<https://doi.org/10.1145/2786805.2786862>.
- [6] WANG J, WANG S, CUI Q, et al. Local-based active classification of test report to assist crowdsourced testing[C/OL] // LO D, APEL S, KHURSHID S. Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering, ASE 2016, Singapore, September 3-7, 2016. [S.l.]: ACM, 2016:

- 190–201.
<https://doi.org/10.1145/2970276.2970300>.
- [7] FENG Y, JONES J A, CHEN Z, et al. Multi-objective test report prioritization using image understanding[C/OL] // LO D, APEL S, KHURSHID S. Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering, ASE 2016, Singapore, September 3-7, 2016. [S.l.] : ACM, 2016 : 202–213.
<https://doi.org/10.1145/2970276.2970367>.
- [8] BETTENBURG N, PREMRAJ R, ZIMMERMANN T, et al. Duplicate bug reports considered harmful ... really?[C/OL] // 24th IEEE International Conference on Software Maintenance (ICSM 2008), September 28 - October 4, 2008, Beijing, China. [S.l.] : IEEE Computer Society, 2008 : 337–345.
<https://doi.org/10.1109/ICSM.2008.4658082>.
- [9] RASTKAR S, MURPHY G C, MURRAY G. Automatic Summarization of Bug Reports[J/OL]. IEEE Trans. Software Eng., 2014, 40(4) : 366–380.
<https://doi.org/10.1109/TSE.2013.2297712>.
- [10] HAO R, FENG Y, JONES J A, et al. CTRAS: crowdsourced test report aggregation and summarization[C/OL] // ATLEE J M, BULTAN T, WHITTLE J. Proceedings of the 41st International Conference on Software Engineering, ICSE 2019, Montreal, QC, Canada, May 25-31, 2019. [S.l.] : IEEE / ACM, 2019 : 900–910.
<https://doi.org/10.1109/ICSE.2019.00096>.
- [11] LI Y, FENG Y, HAO R, et al. Classifying crowdsourced mobile test reports with image features: An empirical study[J/OL]. J. Syst. Softw., 2022, 184 : 111121.
<https://doi.org/10.1016/j.jss.2021.111121>.
- [12] CHEN X, JIANG H, LI X, et al. Automated quality assessment for crowdsourced test reports of mobile applications[C/OL] // OLIVETO R, PENTA M D, SHEPHERD D C. 25th International Conference on Software Analysis, Evolution and Reengineering, SANER 2018, Campobasso, Italy, March 20-23, 2018. [S.l.] : IEEE Computer Society, 2018 : 368–379.
<https://doi.org/10.1109/SANER.2018.8330224>.

- [13] CHEN X, JIANG H, LI X, et al. A systemic framework for crowdsourced test report quality assessment[J/OL]. *Empir. Softw. Eng.*, 2020, 25(2): 1382–1418. <https://doi.org/10.1007/s10664-019-09793-8>.
- [14] 冯剑红, 李国良, 冯建华. 众包技术研究综述 [J]. *计算机学报*, 2015, 38(9): 1713–1726.
- [15] ZHAO Y, ZHU Q. Evaluation on crowdsourcing research: Current status and future direction[J/OL]. *Inf. Syst. Frontiers*, 2014, 16(3): 417–434. <https://doi.org/10.1007/s10796-012-9350-4>.
- [16] YUEN M, KING I, LEUNG K. A Survey of Crowdsourcing Systems[C/OL] // *PASSAT/SocialCom 2011, Privacy, Security, Risk and Trust (PASSAT)*, 2011 IEEE Third International Conference on and 2011 IEEE Third International Conference on Social Computing (SocialCom), Boston, MA, USA, 9-11 Oct., 2011. [S.l.]: IEEE Computer Society, 2011: 766–773. <https://doi.org/10.1109/PASSAT/SocialCom.2011.203>.
- [17] KITTUR A, NICKERSON J V, BERNSTEIN M S, et al. The future of crowd work[C/OL] // *BRUCKMAN A S, COUNTS S, LAMPE C, et al. Computer Supported Cooperative Work, CSCW 2013*, San Antonio, TX, USA, February 23-27, 2013. [S.l.]: ACM, 2013: 1301–1318. <https://doi.org/10.1145/2441776.2441923>.
- [18] DOAN A, RAMAKRISHNAN R, HALEVY A Y. Crowdsourcing systems on the World-Wide Web[J/OL]. *Commun. ACM*, 2011, 54(4): 86–96. <https://doi.org/10.1145/1924421.1924442>.
- [19] CHITILAPPILLY A I, CHEN L, AMER-YAHIA S. A Survey of General-Purpose Crowdsourcing Techniques[J/OL]. *IEEE Trans. Knowl. Data Eng.*, 2016, 28(9): 2246–2266. <https://doi.org/10.1109/TKDE.2016.2555805>.
- [20] TONG Y, YUAN Y, CHENG Y, et al. Survey on spatiotemporal crowdsourced data management techniques[J]. *J. Softw.*, 2017, 28(1): 35–58.

- [21] MAO K, CAPRA L, HARMAN M, et al. A survey of the use of crowdsourcing in software engineering[J/OL]. *J. Syst. Softw.*, 2017, 126 : 57 – 84.
<https://doi.org/10.1016/j.jss.2016.09.015>.
- [22] 章晓芳, 冯洋, 刘嶝, et al. 众包软件测试技术研究进展 [J]. *软件学报*, 2018, 29(1).
- [23] WANG X, ZHANG L, XIE T, et al. An approach to detecting duplicate bug reports using natural language and execution information[C/OL] // SCHÄFER W, DWYER M B, GRUHN V. 30th International Conference on Software Engineering (ICSE 2008), Leipzig, Germany, May 10-18, 2008. [S.l.] : ACM, 2008 : 461 – 470.
<https://doi.org/10.1145/1368088.1368151>.
- [24] SUN C, LO D, WANG X, et al. A discriminative model approach for accurate duplicate bug report retrieval[C/OL] // KRAMER J, BISHOP J, DEVANBU P T, et al. Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 1, ICSE 2010, Cape Town, South Africa, 1-8 May 2010. [S.l.] : ACM, 2010 : 45 – 54.
<https://doi.org/10.1145/1806799.1806811>.
- [25] 陈信. 众包测试报告的挖掘与评估 [D]. [S.l.] : 大连: 大连理工大学, 2018.
- [26] NAZAR N, JIANG H, GAO G, et al. Source code fragment summarization with small-scale crowdsourcing based features[J/OL]. *Frontiers of Computer Science*, 2016, 10(3) : 504 – 517.
<https://doi.org/10.1007/s11704-015-4409-2>.
- [27] JIANG H, ZHANG J, MA H, et al. Mining authorship characteristics in bug repositories[J/OL]. *Science China Information Sciences*, 2017, 60(1) : 12107.
<https://doi.org/10.1007/s11432-014-0372-y>.
- [28] ZIMMERMANN T, PREMRAJ R, BETTENBURG N, et al. What Makes a Good Bug Report?[J/OL]. *IEEE Trans. Software Eng.*, 2010, 36(5) : 618 – 643.
<https://doi.org/10.1109/TSE.2010.63>.

- [29] BETTENBURG N, JUST S, SCHRÖTER A, et al. What makes a good bug report?[C/OL] // HARROLD M J, MURPHY G C. Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering, 2008, Atlanta, Georgia, USA, November 9-14, 2008. [S.l.]: ACM, 2008: 308 – 318.
<https://doi.org/10.1145/1453101.1453146>.
- [30] JOORABCHI M E, MIRZAAGHAEI M, MESBAH A. Works for me! characterizing non-reproducible bug reports[C/OL] // DEVANBU P T, KIM S, PINZGER M. 11th Working Conference on Mining Software Repositories, MSR 2014, Proceedings, May 31 - June 1, 2014, Hyderabad, India. [S.l.]: ACM, 2014: 62 – 71.
<https://doi.org/10.1145/2597073.2597098>.
- [31] HOOIMEIJER P, WEIMER W. Modeling bug report quality[C/OL] // STIREWALT R E K, EGYED A, FISCHER B. 22nd IEEE/ACM International Conference on Automated Software Engineering (ASE 2007), November 5-9, 2007, Atlanta, Georgia, USA. [S.l.]: ACM, 2007: 34 – 43.
<https://doi.org/10.1145/1321631.1321639>.
- [32] GÉNOVA G, FUENTES J M, MORILLO J L, et al. A framework to measure and improve the quality of textual requirements[J/OL]. *Requir. Eng.*, 2013, 18(1): 25 – 41.
<https://doi.org/10.1007/s00766-011-0134-z>.
- [33] HECK P, ZAIDMAN A. A systematic literature review on quality criteria for agile requirements specifications[J/OL]. *Softw. Qual. J.*, 2018, 26(1): 127 – 160.
<https://doi.org/10.1007/s11219-016-9336-4>.
- [34] PARRA E, DIMOU C, MORILLO J L, et al. A methodology for the classification of quality of requirements using machine learning techniques[J/OL]. *Inf. Softw. Technol.*, 2015, 67: 180 – 195.
<https://doi.org/10.1016/j.infsof.2015.07.006>.
- [35] THAKURTA R. A framework for prioritization of quality requirements for inclusion in a software project[J/OL]. *Softw. Qual. J.*, 2013, 21(4): 573 – 597.
<https://doi.org/10.1007/s11219-012-9188-5>.

- [36] ROSENBERG L, HAMMER T. A methodology for writing high quality requirement specifications and for evaluating existing ones[J]. NASA Goddard Space Flight Center, Software Assurance Technology Center, 1999: 1–8.
- [37] WEIKUM G. Foundations of Statistical Natural Language Processing - Book Review[J/OL]. SIGMOD Rec., 2002, 31(3): 37–38.
<https://doi.org/10.1145/601858.601867>.
- [38] KÜBLER S, MCDONALD R T, NIVRE J. Synthesis Lectures on Human Language Technologies: Dependency Parsing[M/OL]. [S.l.]: Morgan & Claypool Publishers, 2009.
<https://doi.org/10.2200/S00169ED1V01Y200901HLT002>.
- [39] ABBAS M, FREIHAT A A. 3rd International Conference on Natural Language and Speech Processing, ICNLSP 2019, Trento, Italy, September 12-13, 2019[C]. [S.l.]: Association for Computational Linguistics, 2019.
- [40] YAMADA K, KNIGHT K. A Syntax-based Statistical Translation Model[C/OL] // Association for Computational Linguistic, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference, July 9-11, 2001, Toulouse, France. [S.l.]: Morgan Kaufmann Publishers, 2001: 523–530.
<https://aclanthology.org/P01-1067/>.
- [41] CHAN Y S, ROTH D. Exploiting Syntactico-Semantic Structures for Relation Extraction[C/OL] // LIN D, MATSUMOTO Y, MIHALCEA R. The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA. [S.l.]: The Association for Computer Linguistics, 2011: 551–560.
<https://aclanthology.org/P11-1056/>.
- [42] 李正华. 汉语依存句法分析关键技术研究 [D]. [S.l.]: 哈尔滨: 哈尔滨工业大学, 2013.
- [43] SUN W, DU Y, KOU X, et al. Grammatical Relations in Chinese: GB-Ground Extraction and Data-Driven Parsing[C/OL] // Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27,

- 2014, Baltimore, MD, USA, Volume 1: Long Papers. [S.l.] : The Association for Computer Linguistics, 2014 : 436–456.
<https://doi.org/10.3115/v1/p14-1042>.
- [44] CHE W, SHAO Y, LIU T, et al. SemEval-2016 Task 9: Chinese Semantic Dependency Parsing[C/OL] //BETHARD S, CER D M, CARPUAT M, et al. Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016. [S.l.] : The Association for Computer Linguistics, 2016 : 1074–1080.
<https://doi.org/10.18653/v1/s16-1167>.
- [45] OEPEN S, KUHLMANN M, MIYAO Y, et al. SemEval 2015 Task 18: Broad-Coverage Semantic Dependency Parsing[C/OL] //CER D M, JURGENS D, NAKOV P, et al. Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015. [S.l.] : The Association for Computer Linguistics, 2015 : 915–926.
<https://doi.org/10.18653/v1/s15-2153>.
- [46] OEPEN S, KUHLMANN M, MIYAO Y, et al. SemEval 2014 Task 8: Broad-Coverage Semantic Dependency Parsing[C/OL] //NAKOV P, ZESCH T. Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014. [S.l.] : The Association for Computer Linguistics, 2014 : 63–72.
<https://doi.org/10.3115/v1/s14-2008>.
- [47] HAJIC J, CIARAMITA M, JOHANSSON R, et al. The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages[C/OL] //HAJIC J. Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2009, Boulder, Colorado, USA, June 4, 2009. [S.l.] : ACL, 2009 : 1–18.
<https://aclanthology.org/W09-1201/>.
- [48] SURDEANU M, JOHANSSON R, MEYERS A, et al. The CoNLL 2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies[C/OL] // CLARK A, TOUTANOVA K. Proceedings of the Twelfth Conference on Computational

- Natural Language Learning, CoNLL 2008, Manchester, UK, August 16-17, 2008. [S.l.]: ACL, 2008: 159–177.
<https://aclanthology.org/W08-2121/>.
- [49] ZHANG M. A Survey of Syntactic-Semantic Parsing Based on Constituent and Dependency Structures[J/OL]. CoRR, 2020, abs/2006.11056.
<https://arxiv.org/abs/2006.11056>.
- [50] de MARNEFFE M, MACCARTNEY B, MANNING C D. Generating Typed Dependency Parses from Phrase Structure Parses[C/OL] // CALZOLARI N, CHOUKRI K, GANGEMI A, et al. Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC 2006, Genoa, Italy, May 22-28, 2006. [S.l.]: European Language Resources Association (ELRA), 2006: 449–454.
<http://www.lrec-conf.org/proceedings/lrec2006/summaries/440.html>.
- [51] ZEMAN D, HAJIC J, POPEL M, et al. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies[C/OL] // ZEMAN D, HAJIC J. Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Brussels, Belgium, October 31 - November 1, 2018. [S.l.]: Association for Computational Linguistics, 2018: 1–21.
<https://doi.org/10.18653/v1/k18-2001>.
- [52] ZHANG T, CHEN J, LUO X, et al. Bug Reports for Desktop Software and Mobile Apps in GitHub: What's the Difference?[J/OL]. IEEE Softw., 2019, 36(1): 63–71.
<https://doi.org/10.1109/MS.2017.377142400>.
- [53] CARLSON N, LAPLANTE P A. The NASA automated requirements measurement tool: a reconstruction[J/OL]. Innov. Syst. Softw. Eng., 2014, 10(2): 77–91.
<https://doi.org/10.1007/s11334-013-0225-8>.
- [54] YANG S-J. A readability formula for Chinese language[M]. [S.l.]: The University of Wisconsin-Madison, 1971.

简历与科研成果

基本信息

张欢，女，汉族，1996 年 5 月出生，四川省泸州市人。

教育背景

2019 年 9 月 — 2022 年 6 月	南京大学软件学院	硕士
2015 年 9 月 — 2019 年 6 月	南京邮电大学计算机软件学院	本科

攻读工学硕士学位期间参与的科研课题

- 国家重点研发计划：信息产品及科技服务集成化众测服务平台研发与应用 (2018YFB1403400)（课题年限 2019 年 7 月 — 2022 年 6 月），负责众测服务的共性理论和服务模式研究。