

面向安卓应用 GUI 的视障用户 可访问性测试方法

何振涛, 徐一放, 张梦玺, 刘华斌

(吉林大学 计算机科学与技术学院, 长春 130012)

摘要: 提出一种自动识别安卓应用中可读文本缺失问题组件的方法, 以提高应用的可访问性。首先, 利用 UI Automator 提取应用的图形用户页面, 并对无关组件进行裁剪, 同时补全组件属性, 以生成相应的视图树; 其次, 设计 3 种启发式规则, 以识别视图树中存在可读文本缺失的组件, 通过对 6 个热门应用的评估实验, 该方法以平均 97% 的准确率成功识别了问题组件; 最后, 生成的测试报告通过在源码与截图中标记问题组件, 帮助应用开发者清晰定位并修复缺失的可读文本。该研究成果不仅能有效改善视障用户的使用体验, 使他们更顺畅地与应用进行交互, 还为开发者提供了一种实用的工具, 促进安卓应用的整体可访问性提升。通过这样的方式, 开发者可以更好地理解和解决可访问性问题, 为所有用户创建一个更友好的数字环境。

关键词: 可访问性; 可读文本; 替代文本; 问题组件识别

中图分类号: TP311.5 **文献标志码:** A **文章编号:** 1671-5489(2025)01-0099-08

Accessibility Testing Method for Visually Impaired Users of Android Application GUI

HE Zhentao, XU Yifang, ZHANG Mengxi, LIU Huaxiao

(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

Abstract: We proposed a method for automatically identifying components with missing readable text in Android applications to improve the accessibility of these applications. Firstly, We used UI Automator to extract the graphical user interface of the application, prune irrelevant components and complete component attributes to generate the corresponding view tree. Secondly, we designed three heuristic rules to identify components with missing readable text in the view tree. Through evaluation experiments on six popular applications, the proposed method successfully identified problematic components with an average accuracy of 97%. Finally, a generated test report helped application developers clearly locate and rectify the missing readable text by marking the problematic components in both the source code and screenshots. The research achievement not only effectively improves the user experience for visually impaired users, enabling them to interact with the applications more smoothly, but also provides developers with a practical tool to promote the overall accessibility of

收稿日期: 2023-11-29.

第一作者简介: 何振涛(1999—), 男, 汉族, 硕士, 从事软件工程的研究, E-mail: zhentaohe.work@foxmail.com. **通信作者简介:** 刘华斌(1986—), 男, 汉族, 博士, 副教授, 博士生导师, 从事智能化软件工程、数据挖掘和人机交互的研究, E-mail: liuhuaxiao@jlu.edu.cn.

基金项目: 吉林省自然科学基金(批准号: 20230101070JC).

Android applications. Through this approach, developers can better understand and address accessibility issues, creating a more user-friendly digital environment for all users.

Keywords: accessibility; readable text; alternative text; problematic component recognition

近年来,随着智能手机的快速发展与普及,移动应用正越来越全方位地融入人们的生活,人们愈发依赖其享受学习、娱乐、购物和社交等服务^[1].移动应用的普及不仅为普通用户带来了便利,也给视障用户带来了更多认识世界的机会.视障指视觉功能受损,除盲人外,视障用户还包括许多需要借助放大镜等辅助器具才能看清对象的弱视人群^[2].而提高应用的可访问性,正是要提高应用的无障碍化程度,使视障用户也有平等的使用应用、享受科技带来便利的机会^[3-4].

借助屏幕阅读器获取信息是视障用户,尤其是全盲用户使用应用(App)最重要的方法.屏幕阅读器能为视障群体提供语音辅助服务,帮助他们通过语音的方式获取手机屏幕上的内容,以及通过手势进行交互.对于占有移动设备操作系统份额最大的安卓系统,谷歌公司推出的 TalkBack 应用是视障用户使用最多的屏幕阅读器. TalkBack 等屏幕阅读器主要通过 GUI(graphical user interface)中组件的文字、描述文本(contentDescription)等可读文本信息作为输入为视障用户提供语音反馈.但开发人员常会因为功能需求和经济效益等原因而忽视视障用户群体的特殊需求,因此安卓应用中的 GUI 组件通常存在为实现 TalkBack 功能提供基础信息的可读文本缺失问题.可读文本缺失问题是影响应用可访问性的最常见问题之一^[5],它的存在会影响屏幕阅读器的语音播报功能和交互功能,从而影响视障用户对应用功能的使用.

因此,本文基于移动应用的无障碍化改造,提出一种针对 GUI 内组件可读文本缺失的自动化检测方法,为应用开发者提高应用的可访问性提供帮助.首先,本文对安卓应用的 GUI 进行预处理,对代表 GUI 的视图树(View Tree)进行裁剪和补全;其次,通过设计启发式规则,识别可读文本缺失问题;最后,对识别到的文本缺失问题以及与之相应的 GUI 组件,为其生成一份检测报告,以供开发者对问题进行修复.本文进行的评估实验也验证了该方法的有效性,能检测出应用中存在的可读文本缺失问题,为提高应用的可访问性提供帮助.

1 本文方法设计

本文方法以 UI Automator 提取到的 GUI 截图和 View Tree 作为输入,以测试报告作为输出. View Tree 表示视图和视图组在布局中的树形结构.首先,对 View Tree 进行裁剪和补全,剪去 View Tree 中与 GUI 无关的干扰节点,并为每个缺失 ID 的节点生成一个唯一标识;其次,对 View Tree 中的节点,通过制定 3 个统一的启发式规则,使用组件的特征与相互之间的关系层层过滤,识别出 View Tree 中存在可读文本缺失问题的组件;最后,对被检 GUI 生成检测报告,该报告涵盖标记问题的布局结构文件、组件列表以及对应的 GUI 截图,以便开发者定位问题组件,提升应用可访问性.本文方法流程如图 1 所示.

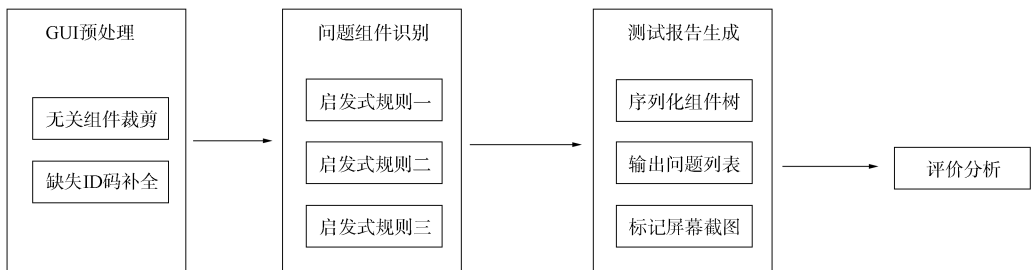


图 1 本文方法流程

Fig. 1 Flow chart of proposed method

1.1 GUI 预处理

安卓应用的 GUI 由一系列容器组件(ViewGroup)和非容器组件(View)构成的 View Tree 表示.

获取 View Tree 主要有两种方法: 一种是获取与 GUI 相对应的静态 XML (extensible markup language) 布局文件; 另一种是使用 AccessibilityService 接口获取 GUI 运行时的 View Tree. UI Automator 是一个安卓系统测试框架, 它基于 AccessibilityService 获取 GUI 信息, 可以使开发者编写自动化测试脚本. 由于 GUI 运行时所能捕获到的信息常比静态页面更丰富, 且能更完整地获取视图信息^[6], 因此本文采用安卓系统提供的基于 AccessibilityService 的界面测试框架 UI Automator 获取应用运行时的 View Tree. AccessibilityService 是安卓系统提供的一种服务, 可使应用程序访问用户界面元素的属性和状态.

但通过 UI Automator 获取到的 View Tree 仍存在一些会对后续测试造成影响的问题. 一是 View Tree 中存在一些与应用无关的组件, 这些组件通常由操作系统产生, 因此本文使用无关组件的类型特征、位置特征等信息将其从 View Tree 中剔除, 以避免其对后续测试产生影响; 二是 View Tree 中的组件常因开发人员的不规范设计而缺少唯一标识符 (ID) 导致组件难以被有效甄别, 从而阻碍测试方法识别组件时的完备性, 且该情况会使开发者难以定位存在文本标签问题的组件, 进而影响他们理解本文测试方法输出的测试结果. 为解决该问题, 本文对 View Tree 中组件的 ID 属性进行补全, 补全算法如下.

算法 1 组件 ID 生成算法.

输入: View Tree;

输出: View Tree;

- 1) let cmp_list be a list of all components of the component tree
- 2) let id_set be a set
- 3) for each component in cmp_list do
- 4) if component does not have an id then
- 5) let idx=1
- 6) let id=String(component.class)+String(idx)
- 7) while id in id_set do
- 8) idx=idx + 1
- 9) id=String(component.class)+String(idx)
- 10) id_set.add(id)
- 11) component.id=id
- 12) return component tree.

算法 1 以经过裁剪后的 View Tree 作为输入, 输出每个组件的 ID 属性得到补全的 View Tree, 通过递增组件 ID 的尾号以及集合去重, 为组件生成唯一的 ID, 并在此过程中组合组件的类型信息, 使阅读本文测试结果的开发者能利用组件的类型信息更快速地定位问题组件.

1.2 可读文本缺失问题识别

可读文本是 TalkBack 读屏时的信息来源, 其中组件的 text 属性值与 contentDescription 属性值是最重要的两类可读文本. 对视力正常用户所能看到的文字, 其一般是开发者在组件中设置的 text 属性值, TalkBack 可以通过获取组件的 text 属性值顺利地把文字内容通过语音反馈给视障用户. 而对于 ImageView, ImageButton, Checkbox, Toggle, Slider 等类型的组件, 其一般不会存在 text 属性值, 而是通过其包含图片的形状和色彩等信息表达自己的含义. 为使视障用户能获取这些组件的含义, 需要开发者为这些组件设置能正确反映组件功能和内容等信息的 contentDescription 属性值. 如图 2 中标号为 1 的两个图标, 若开发者分别为其设置 contentDescription 为“游戏”、“消息”, 则视障用户可以获知这两个组件分别为游戏按钮和消息按钮. 但若未为组件设置 contentDescription 属性值, 则由于这些组件缺乏可读文本, TalkBack 只能告知视障用户这两个按钮为未加标签的按钮, 导致视障用户只能通过尝试点击的方式了解这两个图标按钮的功能.

根据安卓应用可访问性问题的官方指南 (<https://developer.android.com/codelabs/starting->

android-accessibility#5)及对 TalkBack 读屏机制,本文设定如下规则识别存在可读文本缺失问题的元素.

规则 1(文本缺失组件判定规则) 令 N 表示 GUI 的 View Tree 中组件的集合, S 表示 View Tree 中存在可读文本缺失问题组件的集合. 对任意的 $s \in S$, 满足以下条件:

- 1) $s \in N$, 表示 S 是 N 的一个子集.
- 2) s 的组件类型是以下类型或以下类型子类中的一种:

- ① ImageView, 表示 s 用于展示图片;
- ② ImageButton, 表示 s 是图形按钮;
- ③ CheckBox, 表示 s 是复选框;
- ④ Toggle, 表示 s 是切换开关;
- ⑤ Slider, 表示 s 是进度条.

- 3) s 的 contentDescription 属性值为空或 @null, 表示组件缺乏可读文本以解释其含义.

对常见的需要添加 contentDescription 作为可读文本的组件, 利用规则 1 能有效识别其是否缺失 contentDescription. 但有时组件缺失 contentDescription, 其却并不缺乏可读文本, 这时规则 1 便无法适用. 如图 2 编号为 2 的导航栏中, 导航栏中的每个导航项都由一个图标及对图标相应的解释文字组成. 虽然图标缺乏可读文本, 但 TalkBack 能把图标及其解释文字作为一个整体读出, 视障用户能通过其解释文字知晓其用途, 因此这些图标并不需要额外添加 contentDescription, 它们也就不能被简单归类为存在可读文本缺失问题的组件. 安卓应用的 GUI 中存在大量这种图片加文字的搭配以及使用容器中的 contentDescription 充当子组件可读文本的情况, 因此仅利用规则 1 判定组件是否存在文本缺失问题将导致许多误标问题. 为解决该问题, 经过对大量常见应用页面的研究, 本文设定如下补充规则.

规则 2(组件解释文本判定规则) 令 S 表示 View Tree 中存在解释文本组件的集合. 对任意的 $s \in S$, 令 $T(s)$ 表示 s 的解释文本的集合. 对任意的 $t \in T(s)$ 满足以下条件之一:

- 1) t 是 text 属性值非空的 TextView 类型的组件, 且 t 满足下列条件之一:

- ① s 是图片, 且 t 满足下列条件: s 和 t 重叠部分的面积超过 t 面积的 0.5 倍, 表示文字位于图片中.
- ② t 满足下列条件之一:

(i) $t.top > s.top \ \&\& \ t.bottom < s.bottom \ \&\& \ t$ 和 s 的距离小于 $0.5 \times s.width$, 表示 t 是 s 左右侧的解释文本, 其中 top, bottom 和 width 分别表示组件的顶点、底部和宽度;

(ii) $t.left > s.left \ \&\& \ t.right < s.right \ \&\& \ t$ 和 s 的距离小于 $0.5 \times s.height$, 表示 t 是 s 上下侧的解释文本, 其中 left, right 和 height 分别表示组件的左、右边缘和高度.

- 2) t 是 contentDescription 属性非空的 ViewGroup 类型的组件, 且 t 满足下列条件: s 是 t 的子节点, 表示用户能通过 t 的可读文本理解 s 的含义.

使用规则 2 能找到那些存在解释文本解释其用途的组件, 从而使其不被错误地标记为缺失可读文本的组件. 但规则 2 仍不完善, 本文发现现实应用的 GUI 中存在一些满足规则 2 的组件和文本, 但文本却并非是该组件的解释文本. 如图 2 中编号为 3 的方框中框选的文字与图标按钮组件, 文本并非为解释图标按钮的作用, 但却会被规则 2 误判. 经过观察, 发现这些被误判的组件在 View Tree 中的路径存在明显差异. 因此, 为完善规则 2, 本文补充了如下规则.

规则 3(组件相关性判定规则) 令 N 表示 GUI 的 View Tree 中组件的集合. 对任意的 $n \in N$, 令 $R(n)$ 表示与组件 n 相关的组件集合. 对任意的 $r \in R(n)$ 满足以下条件:

- 1) $r \in N$, 表示 $R(n)$ 是 N 的一个子集.



图 2 可读文本缺失问题示例

Fig. 2 Example of readable text missing issue

为代表^[7-9].

为测试本文方法的有效性,分别从视频、社交、音乐、浏览器、学习教育和购物 6 类应用中选取用户常用的一个应用,分别为 Bilibili、微信、网易云音乐、百度、有道翻译和京东.对这 6 个应用每个选取 4~6 个具有代表性的常用页面,共得到 27 个测试页面.通过手工标注的方式统计每个应用中目标组件(ImageView,ImageButton,CheckBox,Toggle,Slider),即可能存在可读文本缺失问题的组件的数量,以及真实存在可读文本缺失问题的组件数量.统计结果如图 5 所示.

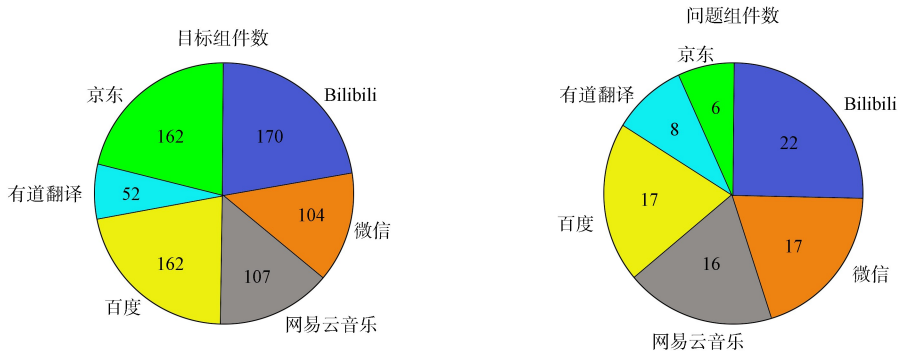


图 5 数据集组件分布情况

Fig. 5 Distribution of components in dataset

2.2 实验方法

由于问题组件只会在目标组件中产生,而对于页面中的目标组件,本文使用规则 1、规则 2 和规则 3,将其按是否存在可读文本缺失问题分为问题组件和非问题组件两类.因此,本文的 3 种规则对问题组件检测的有效性可视为是 3 种规则对目标组件进行二分类的有效性.为对本文方法的有效性进行评估,使用混淆矩阵对 3 种规则进行分析,分别计算精准率、召回率以及 F_1 值 3 个指标.表 1 为一个二分类问题的混淆矩阵,其中:真阳性(TP)表示样本被预测为正,其真实情况也为正;假阳性(FP)表示样本被预测为正,其真实情况却为负;假阴性(FN)表示样本被预测为负,其真实情况也为负;真阴性(TN)表示样本被预测为负,其真实情况却为正.

表 1 二分类混淆矩阵

Table 1 Confusion matrix of binary classification

真实情况	预测情况	
	正例	反例
正例	真阳性	假阴性
反例	假阳性	真阴性

精确率(Precision)、召回率(Recall)和 F_1 值(F_1 -score)的计算公式分别如下:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

2.3 实验结果和分析

表 2~表 4 分别列出了仅用规则 1、使用规则 1 和规则 2 以及使用全部 3 种规则对目标组件进行分类的实验结果.由表 2~表 4 可见,独立使用规则 1 对 GUI 中的问题组件进行检测,其精确率极低,但召回率达 100%.这是因为规则 1 将所有缺乏 contentDescription 属性的目标组件都归类为问题组件,而其中许多组件虽然未设置 contentDescription 属性,但却具有相关的解释文本.规则 1 单独应用产生的基础 F_1 值显然并不适用.引入规则 2 后,考虑到解释文本的存在,准确率显著提高,但由于基于规则的方法可能存在误判,导致漏过部分问题组件,从而降低了召回率.规则 2 的引入将平均 F_1 值

从原来的 0.21 提升至 0.93, 产生了明显效果. 当采用本文设计的全部 3 种规则时, 准确率与召回率进一步提升, F_1 平均值达 0.96, 证明了本文设计的规则显著有效, 能识别 GUI 中存在可读文本缺失问题的组件.

表 2 使用规则 1 的分类实验评估结果

Table 2 Evaluation results of classification experiment using rule 1

应用	精确率	召回率	F_1 值
Bilibili	0.13	1.00	0.23
微信	0.17	1.00	0.29
网易云音乐	0.16	1.00	0.27
百度	0.10	1.00	0.18
有道翻译	0.18	1.00	0.30
京东	0.03	1.00	0.05

表 3 使用规则 1 和规则 2 分类实验评估结果

Table 3 Evaluation results of classification experiment using rule 1 and rule 2

应用	精确率	召回率	F_1 值
Bilibili	1.0	0.86	0.92
微信	1.0	1.00	1.00
网易云音乐	1.0	0.93	0.96
百度	1.0	0.94	0.96
有道翻译	1.0	1.00	1.00
京东	0.6	0.50	0.54

表 4 使用 3 种规则的分类实验评估结果

Table 4 Evaluation results of classification experiment using three rules

应用	精确率	召回率	F_1 值
Bilibili	1.00	0.86	0.92
微信	1.00	1.00	1.00
网易云音乐	1.00	1.00	1.00
百度	1.00	1.00	1.00
有道翻译	1.00	1.00	1.00
京东	0.75	1.00	0.85

综上所述, 针对安卓应用 GUI 的视障用户可访问性, 本文提出了一种针对应用中存在可读文本缺失问题的自动化测试方法. 通过本文的检测报告, 应用开发者可以清晰定位到存在可读文本缺失问题的组件, 并针对问题组件进行可读文本的注入, 以提升应用的可访问性, 提升应用对视障用户的使用体验. 实验结果表明, 本文方法具有极高的准确率和召回率, 能有效检测出存在可读文本缺失问题的组件.

参 考 文 献

- [1] ZHANG M X, LIU H X, CHEN C Y, et al. AccessFixer: Enhancing GUI Accessibility for Low Vision Users with R-GCN Model [J]. IEEE Transactions on Software Engineering, 2024, 50(2): 173-189.
- [2] XU T L, WANG B S, LIU H, et al. Prevalence and Causes of Vision Loss in China from 1990 to 2019: Findings from the Global Burden of Disease Study 2019 [J]. The Lancet Public Health, 2020, 5(12): e682-e691.
- [3] 工业和信息化部, 中国残疾人联合会. 中国残联关于推进信息无障碍的指导意见 [EB/OL]. (2020-09-11) [2023-05-16]. <http://www.scio.gov.cn/xwfbh/xwfbh/wqfbh/42311/44021/xgzc44027/Document/1690214/1690214.htm>. (MINISTRY OF INDUSTRY AND INFORMATION TECHNOLOGY, CHINA DISABLED PERSONS' FEDERATION. Guiding Opinions of the China Disabled Persons' Federation on Promoting Information Accessibility [EB/OL]. (2020-09-11) [2023-05-16]. <http://www.scio.gov.cn/xwfbh/xwfbh/wqfbh/42311/44021/xgzc44027/Document/1690214/1690214.htm>.)

- [4] 中华人民共和国国家发展和改革委员会. 关于印发《“十四五”公共服务规划》的通知 [EB/OL]. (2021-01-10) [2023-05-18]. https://www.ndrc.gov.cn/xxgk/zcfb/ghwb/202201/t20220110_1311622.html?code=&.state=123. (NATIONAL DEVELOPMENT AND REFORM COMMISSION OF THE PEOPLE'S REPUBLIC OF CHINA. Notice on Issuing the “14th Five Year Plan for Public Services” [EB/OL]. (2021-01-10)[2023-05-18]. https://www.ndrc.gov.cn/xxgk/zcfb/ghwb/202201/t20220110_1311622.html?code=&.state=123.)
- [5] ALSHAYBAN A, AHMED I, MALEK S. Accessibility Issues in Android Apps: State of Affairs, Sentiments, and Ways Forward [C]//International Conference on Software Engineering. Piscataway, NJ: IEEE, 2020: 1323-1334.
- [6] LIU Z, CHEN C Y, WANG J J, et al. Nighthawk: Fully Automated Localizing UI Display Issues via Visual Understanding [J]. IEEE Transactions on Software Engineering, 2022, 49(1): 403-418.
- [7] ALOTAIBI A S, CHIOU P T, HALFOND W G J. Automated Detection of Talkback Interactive Accessibility Failures in Android Applications [C]//IEEE Conference on Software Testing, Verification and Validation (ICST). Piscataway, NJ: IEEE, 2022: 232-243.
- [8] ALOTAIBI A S, CHIOU P T, TAWSIF F M, et al. ScaleFix: An Automated Repair of UI Scaling Accessibility Issues in Android Applications [C]//IEEE International Conference on Software Maintenance and Evolution (ICSME). Piscataway, NJ: IEEE, 2023: 147-159.
- [9] LIU Z, CHEN C Y, WANG J J, et al. Unblind Text Inputs: Predicting Hint-Text of Text Input in Mobile Apps via LLM [C]//Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI). New York: ACM, 2024: 1-12.

(责任编辑:韩 啸)