|  |  |
| --- | --- |
|  | **第十七届全国大学生软件创新大赛** |
| **文档编号：SWC2024-Hi安卓** |



**按键精灵**

**AutoDroid**

**项目开发文档**

**Version: 1.0.5**



**Hi安卓**

**2024-03-19**

**All Rights Reserved**

目录

[1 项目概述 1](#_Toc161739511)

[1.1 项目背景 1](#_Toc161739512)

[1.2 项目定位 2](#_Toc161739513)

[1.2.1 应用场景 2](#_Toc161739514)

[1.2.2 目标人群 2](#_Toc161739515)

[1.3 项目方案 2](#_Toc161739516)

[1.4 项目目标 3](#_Toc161739517)

[1.5 项目价值 3](#_Toc161739518)

[2 开发计划 4](#_Toc161739519)

[2.1 最终呈现形式 4](#_Toc161739520)

[2.2 主要功能描述 4](#_Toc161739521)

[2.3 运行环境 5](#_Toc161739522)

[2.4 验收标准 5](#_Toc161739523)

[2.5 关键问题 6](#_Toc161739524)

[2.6 进度安排 7](#_Toc161739525)

[2.7 开发预算 7](#_Toc161739526)

[3 可行性分析 8](#_Toc161739527)

[3.1 技术可行性分析 8](#_Toc161739528)

[3.2 资源可行性分析 8](#_Toc161739529)

[3.3 市场可行性分析 9](#_Toc161739530)

[4 需求分析 10](#_Toc161739531)

[4.1 数据需求 10](#_Toc161739532)

[4.1.1 静态数据 10](#_Toc161739533)

[4.1.2 动态数据 10](#_Toc161739534)

[4.1.3 数据词典 10](#_Toc161739535)

[4.1.4 数据采集 10](#_Toc161739536)

[4.2 功能需求 11](#_Toc161739537)

[4.2.1 功能模块 11](#_Toc161739538)

[4.3 性能需求 11](#_Toc161739539)

[4.3.1 时间特性 11](#_Toc161739540)

[4.3.2 适应性 11](#_Toc161739541)

[4.4 界面需求 11](#_Toc161739542)

[4.5 接口需求 12](#_Toc161739543)

[4.5.1 硬件接口 12](#_Toc161739544)

[4.5.2 软件接口 12](#_Toc161739545)

[4.6 其他需求 12](#_Toc161739546)

[5 概要设计 12](#_Toc161739547)

[5.1 处理流程 12](#_Toc161739548)

[5.2 总体结构设计 13](#_Toc161739549)

[5.3 功能设计 13](#_Toc161739550)

[5.4 数据流转设计 13](#_Toc161739551)

[5.5 用户界面设计 13](#_Toc161739552)

[5.6 数据结构设计 13](#_Toc161739553)

[5.7 接口设计 14](#_Toc161739554)

[5.7.1 外部接口 14](#_Toc161739555)

[5.7.2 内部接口 14](#_Toc161739556)

[5.8 错误/异常处理设计 14](#_Toc161739557)

[5.8.1 错误/异常输出信息 14](#_Toc161739558)

[5.8.2 错误/异常处理对策 14](#_Toc161739559)

[5.9 系统配置策略 14](#_Toc161739560)

[5.10 系统部署方案 14](#_Toc161739561)

[5.11 跨端应用架构设计 15](#_Toc161739562)

[5.12 其他相关技术与方案 15](#_Toc161739563)

[6 数据库设计 15](#_Toc161739564)

[7 手机端侧部署设计 16](#_Toc161739565)

[7.1 手机环境需求 16](#_Toc161739566)

[8 详细设计 17](#_Toc161739567)

[8.1 随机探索模块 17](#_Toc161739568)

[8.1.1 功能描述 17](#_Toc161739569)

[8.1.2 性能描述 17](#_Toc161739570)

[8.1.3 输入 17](#_Toc161739571)

[8.1.4 输出 17](#_Toc161739572)

[8.1.5 程序逻辑 18](#_Toc161739573)

[8.1.6 限制条件 18](#_Toc161739574)

[8.2 记忆生成模块 18](#_Toc161739575)

[8.2.1 功能描述 18](#_Toc161739576)

[8.2.2 性能描述 18](#_Toc161739577)

[8.2.3 输入 19](#_Toc161739578)

[8.2.4 输出 19](#_Toc161739579)

[8.2.5 程序逻辑 19](#_Toc161739580)

[8.2.6 限制条件 19](#_Toc161739581)

[8.3 隐私过滤模块 19](#_Toc161739582)

[8.3.1 功能描述 19](#_Toc161739583)

[8.3.2 性能描述 20](#_Toc161739584)

[8.3.3 输入 20](#_Toc161739585)

[8.3.4 输出 20](#_Toc161739586)

[8.3.5 程序逻辑 20](#_Toc161739587)

[8.3.6 限制条件 20](#_Toc161739588)

[8.4 安卓模块 21](#_Toc161739589)

[8.4.1 功能描述 21](#_Toc161739590)

[8.4.2 性能描述 21](#_Toc161739591)

[8.4.3 输入 21](#_Toc161739592)

[8.4.4 输出 21](#_Toc161739593)

[8.4.5 程序逻辑 22](#_Toc161739594)

[8.4.6 限制条件 22](#_Toc161739595)

文档修订历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **修订原因** | **版本号** | **作者** | **修订日期** | **备注** |
| **1** | **初稿** | **0.5.0** | **Hi安卓队** | **2024.01.07** |  |
| **2** | **完善了需求分析** | **0.6.0** | **Hi安卓队** | **2024.01.15** |  |
| **3** | **初步进行了**  **概要设计** | **0.7.0** | **Hi安卓队** | **2024.02.15** |  |
| **4** | **初步进行了**  **数据库设计** | **0.8.0** | **Hi安卓队** | **2024.02.27** |  |
| **5** | **完善了详细设计** | **0.9.0** | **Hi安卓队** | **2024.03.02** |  |
| **6** | **完善了概要设计** | **1.0.0** | **Hi安卓队** | **2024.03.10** |  |
| **7** | **完善了技术可行性** | **1.0.5** | **Hi安卓队** | **2024.03.18** |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 项目概述

## 项目背景

随着移动计算，物联网技术的高速发展，智能手机已经成为了人们日常生活中的必需品。根据2023年08月28日发布的中国互联网络发展状况统计报告，截至2023年6月，国内市场上监测到的活跃APP数量达260万款，覆盖网民日常学习、工作、生活等方方面面。通过智能手机上搭载的海量的应用资源，我们可以随时随地得在手机端上完成一系列日常活动，诸如点餐、叫车、订票、社交，以及健康监测、运动计划等等。

与此同时，随着用户需求的不断提升，当下海量的APP资源也存在一些缺陷。第一，随着用户需要使用的APP数量增多，用户的学习成本也随之增长。第二，在有些APP上执行任务的操作较为繁琐。第三，APP的使用没有全面考虑到残障人士的使用场景，例如盲人无法看到手机界面，因此无法准确得进行操作。

为解决上述问题，当下诸多手机助手类的商业产品(包括Siri、华为小艺、Google Assistant、Cortana等等) 都引入了对移动任务进行自动化的功能。但是这些手机助手均采用开发类的方法，整个助手所支持的功能需要开发者逐一实现，不能处理用户提出的任何移动任务。以 Google Assistant 为例，如果需要新增一个功能，开发者首先需要确定应用中哪个组件会被触发，然后编写相应的内容并注册到助手中；在执行任务时，Google Assistant 使用 NLU(自然语言理解) 模型，将用户的命令和预设的内容进行映射，提取相应的参数，最后调用整个功能。显然，这种方式在功能的可扩展性上表现很差，每一个新的功能都需要开发者来完成，既提高了开发难度，也增加了用户的学习成本。

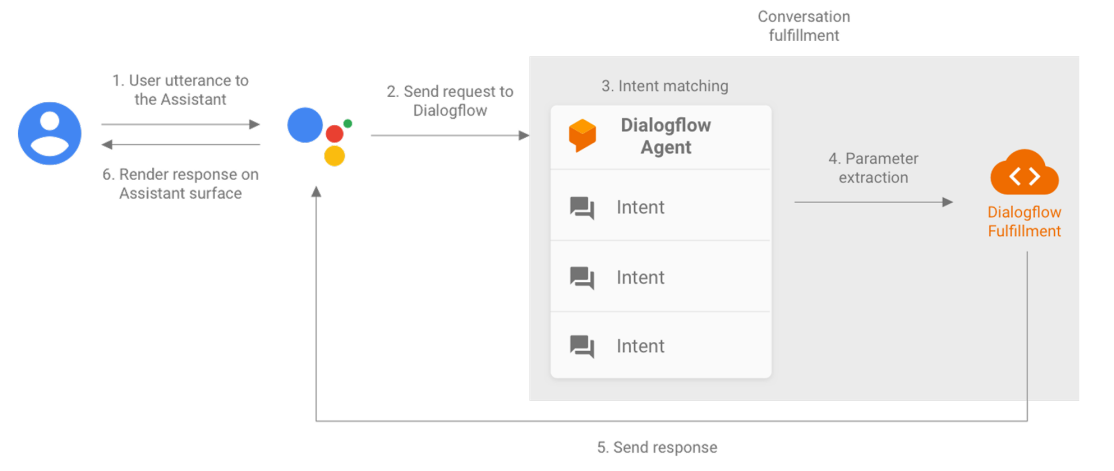


图 1 Google Assistant工作流程图

一直以来，实现通用类人智能都是人类不懈追求的目标，随着人工智能AIGC （（AI Generated Content，人工智能生成内容））技术的发展，大语言模型对文本表现出了强大的理解、推理能力，基于大语言模型开发的智能代理（Agent），在各种下游任务上取得了非常好的性能，方便了人们的生活。

因此，我们希望基于大语言模型的海量的先验知识，突破当前市面上开发类手机助手只能处理部分APP上的部分移动任务的限制，开发一款能够处理用户提出的任何移动任务的新一代手机助手，使得用户能够在手机上通过语音或文字指令，即可让个人助手跳转到相应的应用程序完成任务。这样的智能助手将会提高用户的生活效率，减少操作步骤，带来更便捷的服务体验。与此同时，这样的智能助手也可以为残障人士使用手机带来更多的便利。

## 项目定位

### 应用场景

A场景：某一个APP上完成某个任务的过程较为繁琐，使用自动化的方法可以方便得完成任务。

B场景：某一个APP的使用对用户来说不熟悉，不知道如何操作来完成某个任务，这时可以使用自动化的方法可以方便得完成任务。

C场景：残障人士难以控制手机界面（如盲人），这时可以通过语音输入指令来自动化得完成任务。

### 目标人群

智能手机的使用者，不方便进行手机上的点击、输入操作的残障人士。

## 项目方案

针对复杂用户界面难以被模型理解的问题，该项目通过将GUI元素转换成模型可以理解的简化HTML格式，并利用大型语言模型的强大文本处理能力来实现对用户界面的精确解析和操作。这样，模型就能像理解常规文本一样理解和操作GUI，提高任务执行的准确性。

对于缺少应用程序特定上下文和知识的挑战，该项目通过收集和整合应用程序的界面转换记忆和任务执行模拟，形成丰富的应用相关知识库。这些知识被嵌入到模型的提示词中，使模型在执行任务时能够引用这些具体的应用知识，提升完成复杂任务的能力。

面对计算资源消耗大的问题，该项目采取了优化模型请求策略，包括合并功能相似的界面元素和预先执行可能的动作以减少不必要的模型交互，减少了对模型的请求次数和请求的复杂度，从而降低了运行成本和提高了效率。

为了应对动作选择不准确和任务完成率不高的难题，项目实施了基于相似任务的快速跳转和界面合并技术，允许模型通过比较当前任务与已知任务的相似性来快速做出决策，同时通过整合多个界面状态为单一请求来提高任务完成的连贯性和准确性。这些策略使得模型能够更加高效和准确地完成一系列复杂任务。

## 项目目标

设计并实现一个集成大语言模型（云端、本地），安卓APP，APP Memory，大语言模型微调生态的移动任务自动化系统。

## 项目价值

（1）顺应AI大语言模型的发展所向

3月5日，政府工作报告指出，制定支持数字经济高质量发展政策，积极推进数字产业化、产业数字化，促进数字技术和实体经济深度融合；深化大数据、人工智能等研发应用，开展“人工智能+”行动，打造具有国际竞争力的数字产业集群。大语言模型技术作为人工智能领域的一项重要创新在今年引起了广泛关注。据我们所知，这是第一项通过结合 LLM 和特定应用程序知识来增强移动任务自动化的工作。不仅扩展了LLM的应用场景，还验证了其在实际操作中的有效性和可行性。这种实践推动了AI技术从理论研究和简单任务处理，向更复杂的实际应用环境转变，为LLM在更广泛领域的应用开辟了新途径。

1. 顺应未来无障碍技术发展的趋势

AutoDroid通过自动化操作移动应用，能够为视觉障碍者等残障人士提供更为便捷的手机使用体验。它不仅限于提供传统的屏幕阅读器功能，而是通过理解和执行具体的应用内任务，进一步扩展了无障碍技术在智能手机领域的应用范围。这种技术的进步，使得残障人士能更加自主地参与到数字生活中，提高了他们的生活质量和社会参与度。

1. 提高用户效率、具有普适性与灵活性

AutoDroid通过自动化完成繁琐的移动应用内任务，大幅减少用户在手机上的操作时间，提高日常工作和生活的效率。不依赖特定API或开发者适配，能够在任何APP上执行任务的能力，展示了AutoDroid的高度普适性和灵活性，这意味着其有潜力服务于更广泛的用户和应用场景。

# 开发计划

## 最终呈现形式

项目最终成果为基于LLM的安卓端移动任务自动化APP与项目开发、设计、测试等文档。

## 主要功能描述

1.指令输入：可以通过文本或语音的形式输入用户指令。

2.APP指定：用户可以指定完成任务的APP。

3.虚拟点击、输入：该系统可以解析LLM的输出，并基于Android提供的无障碍功能，在相应组件上执行虚拟的点击或输入功能。

4.隐私保护：该系统会在解析用户界面的时候，检测用户界面中的隐私信息，避免将用户的隐私信息传到云端。

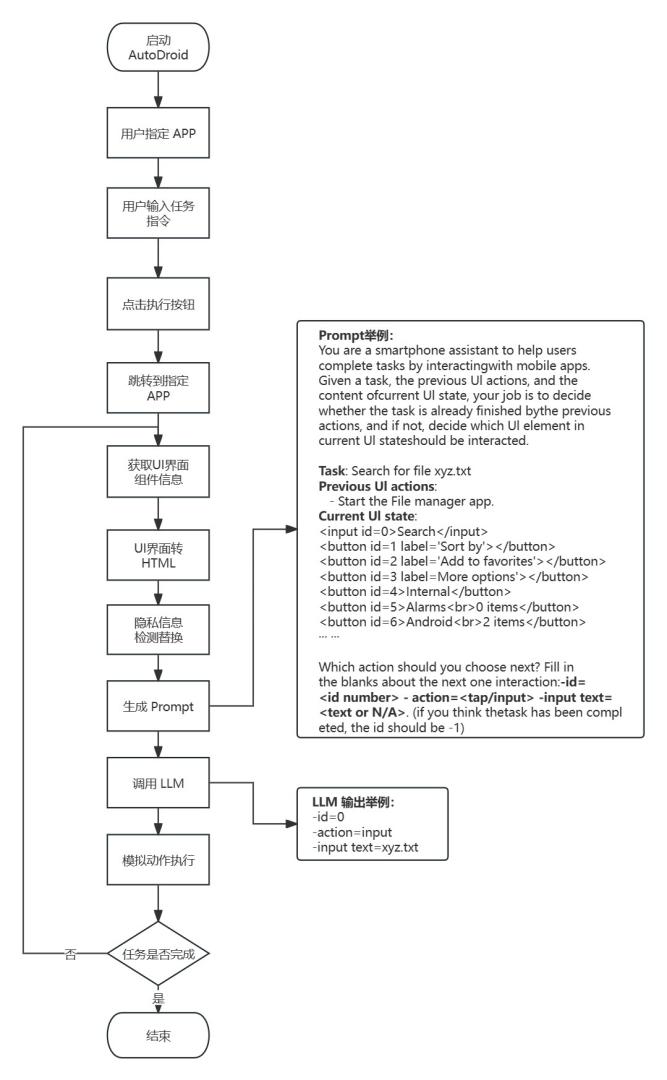


图 2 系统工程流程图

用户启动AutoDroid，指定APP，输入任务指令，然后点击执行按钮。系统便会自动跳转到该指定的APP当中，然后会使用无障碍服务来获取UI界面的组件信息，并且将其转化为HTML的文本形式，然后将检测替换其中的用户隐私信息，生成提示词（prompt）。然后使用该提示词去调用本地的或者云端的大语言模型。如果该任务没有完成，大语言模型会返回为完成该任务所需执行的下一步动作，然后AutoDroid会调用无障碍服务来执行该模拟动作，执行完成之后再次执行UI转文字并调用大语言模型的操作；如果该任务已经完成，那么大语言模型就会返回id = -1，表示没有组件被选择，该任务结束。

## 运行环境

**硬件环境：**

1. 手机：运行内存8GB，存储256GB。
2. PC机：处理器Intel i5-11，内存16GB，磁盘500GB。
3. GPU 服务器：8卡80GB A100 GPU。

**软件环境：**

1. PC 操作系统：Windows10
2. Android 操作系统：Android 9，Android 10，Android 11.
3. 开发工具：Android Studio，用于应用的编译、构建和调试。
4. 测试辅助工具：ADB (Android Debug Bridge) 用于设备管理和调试，Android Emulator 用于虚拟设备测试。

## 验收标准

* 能够提取用户手机上的APP信息，并在前端展示。
* 能够成功跳转到执行任务的APP。
* 能够解析当前GUI界面，并且将其转换为自然语言的描述。
* 能够生成提示词并且询问LLM获取在当前界面完成任务所需的动作。
* 能够根据LLM的输出在手机上执行对应动作。
* 《项目开发文档》、《技术研究报告》、《项目测试文档》、《作品创新性分析报告》材料完整。
* 软件可正常运行。
* 实现项目软件需求说明书要求的各项功能需求。
* 软件界面友好，易于交互。
* 软件功能新颖，有较强创新。
* 能够以较高的正确率完成用户任务。

## 关键问题

1. **设备支持**

安卓运行设备为OnePlusACE2Pro，搭载8个3.2GHz的ARM内核，以及AdrenoTM 740 GPU。本地的Vicuna-7B模型使用MLC-LLM的方法部署到手机上，并事先使用8个A100GPU微调4小时。

1. **技术要点**

·基于大语言模型的用户指令解析与理解

手机个人助手需要有一定的灵活性，能够满足用户不书面化的要求。因此我们引入了大语言模型。大型语言模型是近年来出现在自然语言处理领域的一种新兴模型。这些模型通常包含数十亿到数千亿的参数，并在大型文本语料库上进行训练，其在生成任务上展现出了很好的效果。我们利用GPT-4生成模拟任务存储在本地，执行任务时个人助手可以先查询本地记忆，从而加快任务执行。考虑到访问云端大模型会有网络延迟，我们在本地运用MLC技术部署了Vicuna-7B，并对其进行了微调，使其更好的适配任务的执行。Vicuna-7B是一个文本大模型，是LLaMA的变种。

·基于LoRA的大语言模型微调技术

微调是深度学习中迁移学习的一种方法。大模型的权重可以在新数据上进行训练，使其能够更好的完成特定的任务。我们构建了一个自己的数据集并用其对Vicuna-7B进行微调，使其获得更好的表现，拥有与GPT-3.5相接近的效果，但是可以部署在本地，摆脱网络延迟的影响，同时能够避免将用户界面中的隐私信息传到云端，保护了用户的隐私安全。

·基于Android Accessibility的模拟点击与输入技术

Android Accessibility（安卓无障碍服务）旨在帮助用户改善设备的可访问性，使得那些有视觉、听觉、运动或认知障碍的用户也能更容易地使用设备。无障碍服务可以在后台监控用户界面的事件，进行模拟点击、输入等功能。

·基于Microsoft Presidio的隐私数据检测与替换技术

通过模式识别、实体检测、模式验证、基于上下文提高检测信度等步骤，快速识别并替换文本中的敏感数据，如：信用卡号码、姓名、位置、身份证号码、电话号码等。将用户隐私信息用非专用词取代（例如“＜姓名＞”→“张三”），在收到LLM的响应后，再将特殊单词映射回原始单词。从而保护用户的隐私信息。

·基于MLC的LLM手机端部署技术：

MLC是一种高性能的通用部署解决方案，该项目的使命是让每个人都能使用 ML 编译技术在每个人的设备上原生开发、优化和部署 AI 模型。在MLC中已经内置了量化技术，可以在尽量少的降低大模型正确率的情况下大幅减少大模型参数，部署本地大模型量化是非常关键的一步，因为手机的内存消耗是非常重要的。MLC还会对模型进行部署优化，使其的代码和内存管理等方面更加适合手机这个载体。

**数据集**：MoTiF数据集。MoTiF是一个大规模的移动应用任务数据集，包含超过4.7k个任务（不包括没有有效演示的任务）。该数据集还提供了注解者在完成这些任务时与之交互的 GUI 屏幕的截图和基于树结构的表示。后续会用来来微调我们的LLM。

## 进度安排

表 1 进度安排表

|  |  |  |  |
| --- | --- | --- | --- |
| 任务编码 | 任务名称 | 开始时间 | 结束时间 |
| 1.1 | 项目资料查找 | 2023-12-13 | 2023-12-17 |
| 1.2 | 初步需求分析 | 2023-12-13 | 2023-12-15 |
| 1.3 | 成员技能学习 | 2023-12-16 | 2023-12-31 |
| 2.1 | 开发文档初稿撰写 | 2023-12-31 | 2024-01-02 |
| 2.1 | 技术文档初稿撰写 | 2023-12-31 | 2024-01-02 |
| 2.2 | 动作获取功能开发 | 2023-12-13 | 2023-01-02 |
| 2.3 | 动作执行功能开发 | 2023-12-13 | 2023-01-02 |
| 2.4 | 文档撰写（初赛定稿） | 2024-01-03 | 2024-01-04 |
| 2.5 | 各模块功能升级 | 2024-02-29 | 2024-03-01 |
| 2.6 | 开发文档、技术文档三稿（复赛定稿） | 2024-03-01 | 2024-03-14 |

## 开发预算

开发预算如表2所示：

表 2 开发预算表

|  |  |
| --- | --- |
| 项目 | 价格 |
| ChatGPT API | 100元 |
|  |  |
|  |  |

# 可行性分析

## 技术可行性分析

1. 大模型本地部署：

为了使手机个人助手运行速度加快，我们需要运用MLC将大模型部署在本地，以去除网络延迟的影响。MLC是一种成熟的高性能的通用部署解决方案，它可以通过对大模型的代码和内存管理等方面进行优化，使其更加匹配手机硬件的运行，加快模型的推理速度。并且它还会对模型进行量化处理，在尽量不降低正确率的情况下大幅减少模型的参数量，从而减少大模型占据的内存，有利于在手机这种内存较少的边缘设备上运行。

1. 大模型微调：

大语言模型微调已经在科研领域取得了广泛的应用，已经训练好的大模型可以通过对其参数进行微调的方式来让大模型在特定任务上的正确率得到提升。因此我们自己构建了一个任务数据集，并用其对Vicuna-7B进行了微调，使其在手机自动化任务中得到更好的表现。

## 资源可行性分析

1. 人力资源：

目前团队中有四名成员，分别负责App前端开发、APP后端开发、LLM微调和手机端部署的研究、UI转文字算法和APP Memory生成的研究，能够保证项目全方面的顺利进行。

两位开发APP的同学曾有过多个安卓端APP的开发经验。

负责LLM微调和手机端部署的研究同学曾多次微调Vicuna 7B模型，取得了良好的效果，并且有将LLM部署到手机端的成功实践。

负责UI转文字算法和APP Memory生成的研究的同学有将安卓界面信息进行文本化的相关经验，并且已有生成APP Memory并使用的成功实践。

指导老师主要研究方向为多模态大模型，能够指导项目的进行。

1. 设备资源：

已购一台OPPO品牌测试手机一部、其他品牌测试手机4部；租用8卡80G A100的云GPU服务器一台；实验室共享GPU服务一台。

1. 大语言模型资源：

云端大模型：目前团队可以调用ChatGPT 3.5的API，区域赛决赛开始会得到大赛官方提供的OPPO AndesGPT的API。

本地大模型：目前团队可以使用开源大语言模型：Vicuna、LLaMa。

## 市场可行性分析

在现代社会，智能手机已经成为了人们日常生活中不可或缺的一部分，从社交娱乐到工作学习，智能手机都是我们的得力助手。因此，针对智能手机的各类应用开发也变得尤为重要。尤其是随着人工智能AIGC技术的飞速发展，如何将这些技术应用到移动端，使日常活动更加智能化、自动化，成为了当前的一大热点。

然而当下诸多手机助手类的商业产品所支持的移动任务自动化功能大多使用开发类方法来实现，也即所有的功能需要开发者逐一实现，不能处理用户提出的任何移动任务。因次，市场上缺乏具有较强扩展性和功能性的移动任务自动化助手。

因此，从市场角度来看，本项目开发的AutoDroid系统具有巨大的市场潜力，具体来讲有如下五个方面。

一、用户定制服务：个性化需求满足。

随着智能手机在人们日常生活中的普及，用户对于手机操作的需求也日益多样化。传统的手机助手类应用往往只能提供固定的功能，无法满足用户的个性化需求。而AutoDroid的核心优势在于其对用户自定义任务的强大处理能力。这意味着，用户可以根据自己的需求，通过自然语言描述来定义任务，从而实现各种个性化的操作。例如，用户可以描述“每天晚上10点，帮我清理手机内存”，AutoDroid就可以根据这个描述自动执行相应的操作。这样的服务对于那些经常需要处理复杂手机操作的用户来说，具有极高的吸引力。

二、扩展性强：无限的可能性。

由于AutoDroid项目是基于大语言模型构建的，其扩展性非常强。开发者可以根据市场需求，不断优化和增加新的功能。此外，由于用户也可以自定义任务，这意味着AutoDroid的服务不仅仅局限于现有的功能，而是可以根据用户需求进行不断的创新和变化。

三、企业级应用：提高工作效率。

除了个人用户外，企业也是AutoDroid的重要目标客户。在企业中，员工经常需要处理各种手机任务，如发送邮件、查询信息等。通过AutoDroid，企业可以大大提高员工的工作效率，减少不必要的时间浪费。此外，企业还可以根据自身需求，定制特定的任务，如设置自动回复短信、批量发送邮件等。

四、长期价值：数据驱动的决策。

AutoDroid项目不仅可以通过服务获取收入，其长期价值还体现在数据驱动的决策上。通过收集和分析用户的任务数据，AutoDroid可以深入了解用户的行为习惯和需求，从而为未来的产品开发和市场策略提供重要的数据支持。

综上所述，AutoDroid项目在商业上具有巨大的潜力和价值。无论是满足用户的个性化需求、提供扩展性服务、开发企业级应用、创造额外收入还是数据驱动的决策，在市场上都具有广阔的应用前景，为其商业成功奠定了坚实的基础。

# 需求分析

## 数据需求

### 静态数据

AutoDroid项目需要以下静态数据：

1. APP Memory信息：为了构建和维护APP Memory表，需要收集各个应用程序的用户界面元素及其对应的功能描述、路径信息等静态数据。
2. 无障碍服务规则集：开发无障碍服务功能时，需要一套预定义的UI组件识别规则库，以便准确地将屏幕上的动态组件转化为HTML文本描述。
3. 隐私敏感词汇列表：基于Microsoft Presidio算法构建的隐私信息词典，包含姓名、电话号码、邮箱地址等多种类型的数据标识符。

### 动态数据

1. 实时UI变化记录：系统在运行过程中需动态监测手机APP界面的变化情况，包括页面跳转、按钮点击和输入框内容变化等实时事件数据。
2. 用户指令日志：记录用户的语音或文字指令，用于后续模型训练与优化。
3. 任务执行反馈数据：跟踪并记录AutoDroid在不同APP上完成任务过程中的交互操作、时间消耗以及结果状态，以评估性能和改进系统。

### 数据词典

1. LLM训练数据集：AutoDroid依赖于大语言模型进行推理，因此需要相应的训练数据集来提升模型对移动任务的理解能力，涵盖各种情境下的任务描述和解决方案样本。
2. 自然语言理解(NLU)词汇表：为提高NLU模块处理用户指令的准确性，需建立一个词汇表，包含可能出现在用户指令中的关键词汇及对应的语义解释。

### 数据采集

人工精细标注：开发了一款专门用于安卓端的手动任务标注APP，借助该工具，我们已手动收集并精心标注了1000个高质量的移动任务实例。这些实例不仅包括任务名称，还详尽描述了完成任务所需的步骤路径、具体操作动作，并且每一项任务均附带了对应的界面截图以及整个任务执行过程的录屏文件，确保了数据的直观性和准确性。

## 功能需求

### 功能模块

1. 指令输入与解析模块：接收用户文本或语音指令，并进行有效的自然语言理解与解析。
2. 智能决策模块：基于LLM技术，根据当前用户界面信息生成下一步动作指令。
3. 无障碍服务模块：实现对APP界面上的虚拟点击、输入等操作。
4. 隐私保护模块：对用户界面中涉及隐私的信息进行检测和替换，确保信息安全传输。
5. APP Memory管理模块：利用APP Memory技术提供快速跳转功能，减少任务执行步骤。

## 性能需求

### 时间特性

1. 响应时间：系统应当能够在接收到用户指令后，在合理的时间内（例如秒级）完成任务分析和执行操作。
2. 任务执行效率：对于复杂的任务流程，AutoDroid应具备高效的自动化执行能力，尽量减少不必要的延迟和等待。

### 适应性

1. 系统必须能够兼容市面上主流的Android操作系统版本，对各类APP有良好的兼容性和适配性。
2. 能够适应不同的网络环境，在离线状态下仍可通过本地部署的大语言模型提供基础服务。

## 界面需求

AutoDroid的前端展示层需具有简洁易用的设计，提供直观的任务输入界面、执行进度展示、历史任务查询等功能，并针对残障人士优化触控及语音交互体验。

## 接口需求

### 硬件接口

1. 与智能手机的无障碍服务API对接，实现实时监听和控制APP界面的功能。
2. 支持麦克风和扬声器设备，确保语音指令的录入和反馈顺畅。

### 软件接口

1. 开放API供其他应用调用，允许外部程序向AutoDroid发送任务请求。
2. 实现与云端大语言模型（如ChatGPT）和本地部署模型（如Vicuna）之间的数据交换接口。

## 其他需求

1. 安全需求：保证用户数据安全，防止未经授权的数据泄露。
2. 可扩展性需求：预留足够的接口和支持，便于未来功能升级和服务扩展。
3. 用户定制化需求：允许用户自定义快捷指令集，方便常用任务一键执行。

# 概要设计

## 处理流程

AutoDroid系统处理流程如下：

1. 用户启动应用并指定目标APP及任务指令。
2. 系统通过无障碍服务监控指定APP的UI变化，实时获取界面组件信息。
3. 将UI信息转化为HTML文本，并利用Microsoft Presidio算法检测和替换敏感隐私数据。
4. 生成包含操作提示词（Prompt）的请求，将此Prompt传送给本地或云端的大语言模型（LLM）进行解析决策。
5. LLM根据Prompt返回下一步动作指令，如模拟点击、输入等。
6. AutoDroid调用无障碍服务执行模拟动作。
7. 动作完成后再次检查当前UI状态，循环上述过程直至任务完成或判断任务已完成。

## 总体结构设计

AutoDroid采用四层架构：显示层、服务层、LLM层和数据层。显示层提供用户交互界面；服务层利用无障碍服务与手机APP交互；LLM层负责理解任务指令并决策行动方案；数据层记录APP Memory信息以加速任务执行。

## 功能设计

1. 智能决策：基于大语言模型对用户指令和当前界面状态的理解来制定下一步操作策略。
2. 隐私保护：使用隐私过滤器自动识别并替换敏感信息，确保在传输至LLM时不泄露隐私数据。
3. 任务自动化：模拟人类操作，实现对APP内部功能的无感自动化执行。
4. 记忆跳转：利用APP Memory信息快速定位到相关界面，提高任务执行效率。

## 数据流转设计

数据从显示层经用户指令传递至服务层，服务层抓取UI界面信息并转化为可被LLM处理的格式。隐私过滤后，数据流向LLM层，接收LLM的决策反馈后，数据流回服务层触发模拟操作。操作结果再次经过UI转文字环节，形成新的数据输入给LLM，直到任务完成或得到终止指令。

## 用户界面设计

用户界面应简洁直观，包括APP选择区域、用户指令输入框、开始执行按钮以及任务进度展示区。同时，为保证用户体验，还需考虑不同用户群体的易用性需求，特别是针对残障人士提供无障碍设计支持。

## 数据结构设计

1. UI组件信息结构：存储每个UI组件的ID、标签、类型、位置信息等属性。
2. 隐私信息映射表：用于存储替换前后的隐私词汇及其对应关系。
3. APP Memory表：记录各个APP内关键组件的位置、关联任务和快捷路径。

## 接口设计

### 外部接口

1. 提供与其他第三方应用对接的API接口，允许外部应用向AutoDroid发送任务请求。
2. 实现与云端LLM（例如ChatGPT、OPPO AndesGPT）的API连接，确保远程请求与响应的高效稳定。

### 内部接口

1. 定义无障碍服务与核心逻辑模块之间的通信接口，保证UI事件的准确捕捉与命令的及时执行。
2. 设计LLM层与服务层之间的数据交换协议，确保LLM输出指令能够准确无误地传达给服务层。

## 错误/异常处理设计

### 错误/异常输出信息

系统应明确记录错误发生的时间、地点、原因、影响范围，并以友好的方式呈现给用户和开发者，便于排查问题。

### 错误/异常处理对策

1. 针对网络波动导致的LLM访问延迟或失败，设计重试机制和备用方案，如切换至本地部署的LLM。
2. 当无法识别或执行某项UI操作时，采取默认安全策略（如停止操作），并向用户报告异常情况及可能的解决建议。

## 系统配置策略

系统应具备灵活的配置选项，让用户可以根据自身设备性能、网络环境及隐私需求调整LLM的使用策略（如是否优先使用本地模型）、数据缓存策略等。

## 系统部署方案

1. 在内存有限的移动端设备上，通过MLC技术量化并优化大模型，将其部署于本地，减少资源占用。
2. 对于云端资源丰富的场景，可以提供云端计算模式，借助服务器端强大的LLM进行更复杂的任务处理。

## 跨端应用架构设计

考虑到多平台兼容性，需要设计一套跨端解决方案，确保AutoDroid能够在不同Android版本和硬件环境下稳定运行，并评估未来向iOS或其他移动操作系统扩展的可能性。

## 其他相关技术与方案

1. 使用LoRA微调技术优化本地部署的Vicuna-7B模型，使其更好地适应移动端的任务执行要求。
2. 引入先进的编译技术和内存管理策略，确保AI模型在移动设备上的高效运行和低能耗表现。

# 数据库设计

AutoDroid系统涉及到多种数据存储需求，以下是针对这些需求的数据库设计概要：

1. 用户任务数据库：

* 表结构: 用户任务表（UserTasks）
* 字段：包括用户ID、任务指令文本、执行APP名称、任务创建时间、任务完成状态、任务执行过程记录等。用于存储用户提交的任务指令及相关的任务执行历史。

1. UI转换与隐私信息数据库：

* 表结构: UI组件描述表（UIComponents）和隐私信息替换记录表（PrivacyReplacements）
* 字段：UI组件描述表包含组件ID、应用ID、组件类型、标签、HTML文本描述、所在页面路径等；隐私信息替换记录表记录原始敏感数据、替换后的占位符以及还原时对应的原数据。

1. APP Memory数据库：

* 表结构: APPMemory表
* 字段：包括APP ID、组件ID、到达该组件的路径（点击序列）、模拟任务描述、特征向量等。此表用于存储各个APP的关键操作链路和相关元数据，以便快速检索和执行任务。

1. 大语言模型响应数据库：

* 表结构: LLMResponses表
* 字段：请求ID、用户输入Prompt、LLM返回的动作序列、执行结果、错误代码（若发生异常）、调用时间戳等。用于保存LLM对每个用户请求所生成的响应及其执行情况。

1. 用户偏好设置数据库：

* 表结构: UserPreferences表
* 字段：用户ID、是否启用本地LLM、默认使用的APP、语音识别偏好、隐私保护等级设置等。用于个性化定制用户的使用环境和服务质量。

在实际开发中，以上各表将根据业务需求进行合理的设计，并确保数据之间的关联性，通过索引、外键约束等方式提高查询效率，同时配合合理的数据备份和同步策略，保证系统的稳定性和数据安全性。此外，考虑到移动端资源限制，还需要设计相应的缓存策略，减少对数据库的频繁访问以降低功耗并提升用户体验。

# 手机端侧部署设计

## 手机环境需求

系统版本要求：应用最低支持Android 12（API级别31）的设备，且已针对Android 13（API级别33）进行过测试和优化。

功能实现要求：允许应用开启无障碍服务，后台运行权限。

# 详细设计

## 随机探索模块

### 功能描述

AutoDroid系统在离线阶段会对APP进行探索，生成APP的UI转换图，并对UI组件进行剪枝操作，对于不同界面中出现的重复的组件，仅保留离初始界面最近的一个。然后会见之后的所有组件通过query LLM的方式获取其功能，我们称之为模拟任务。基于此，我们生成了APP Memory表，每一个表项中存储：一个组件、到达该组件的路径（从初始界面到达该组件所在界面所需点击的组件）、该组件模拟任务。APP Memory会在之后用于对Prompt的增强，以及Memory Shrortcut快速跳转的操作。

### 性能描述

1. 运行时间：对于每个应用程序，生成UI Transition Graph (UTG)大约需要0.5-1小时，然后根据llm对其进行分析以合成模拟任务，大约需要5-10分钟。

2. 稳定性：本模块可以在APP端正常工作，并且不会出现崩溃和错误，具有良好的可靠性和一致性。同时，我们会持续进行测试和维护，确保模块的稳定性和可靠性。

3. 资源占用：因为要生成模拟任务并存储在APP Memory表中，所以对需要一定的内存占用。

4. 兼容性：本模块能够在不同的平台、操作系统和设备上正常工作。

### 输入

手机应用的apk

### 输出

一个装有模拟任务的APP Memory表。

### 程序逻辑



图 3 随机探索模块流程图

### 限制条件

目前只能探索不需要登录就可以正常使用的APP。

## 记忆生成模块

### 功能描述

所有的APP Memory中的模拟任务会通过一个特征提取模型Instructor-XL映射成固定维度的特征向量。在用户输入指令以及执行任务的APP后，系统首先会通过Instructor-XL将用户输入指令做映射，然后和该APP Memory中的模拟任务计算相似度。挑选相似度最高的模拟任务对应的组件，在任务执行开始的时候，先行跳转到该组件对应的界面并点击该组件。这样的设计减少了执行指令需要query LLM的次数，减少了开销，也减少了完成指令的所花费的时间。

### 性能描述

1. 运行时间：对于每个应用程序，将模拟任务映射到高维向量平均需要10s。

2. 稳定性：本模块可以在APP端正常工作，并且不会出现崩溃和错误，具有良好的可靠性和一致性。同时，我们会持续进行测试和维护，确保模块的稳定性和可靠性。

3. 资源占用：因为要生成高维向量并进行存储，所以需要一定的存储占用。

4. 兼容性：本模块能够在不同的平台、操作系统和设备上正常工作。

### 输入

APP Memory表

### 输出

一组带有任务信息的高维向量

### 程序逻辑



图 4 记忆生成模块流程图

### 限制条件

高维向量带有的信息不全，此模块还可以进行优化，使其更好的浓缩探索信息。

## 隐私过滤模块

### 功能描述

基于Microsoft Presidio算法，在线处理过程中，AutoDroid运行一个人身份信息（PII）扫描仪，检测UI信息中的用户隐私信息，包括姓名、电话号码、电子邮件地址等。将用户隐私信息用非专用词取代（例如“＜姓名＞”→“张三”），然后将Prompt发送到云。在收到LLM的响应后，AutoDroid在解析操作之前将特殊单词映射回原始单词。

### 性能描述

1. 运行时间：运行较为迅速，占用时间不到1s。

2. 稳定性：本模块可以在APP端正常工作，并且不会出现崩溃和错误，具有良好的可靠性和一致性。同时，我们会持续进行测试和维护，确保模块的稳定性和可靠性。

3. 资源占用：因为需要在线处理，所以会有一定的网络带宽占用。

4. 兼容性：本模块能够在不同的平台、操作系统和设备上正常工作。

### 输入

包含个人隐私的数据

### 输出

去除个人隐私的数据

### 程序逻辑

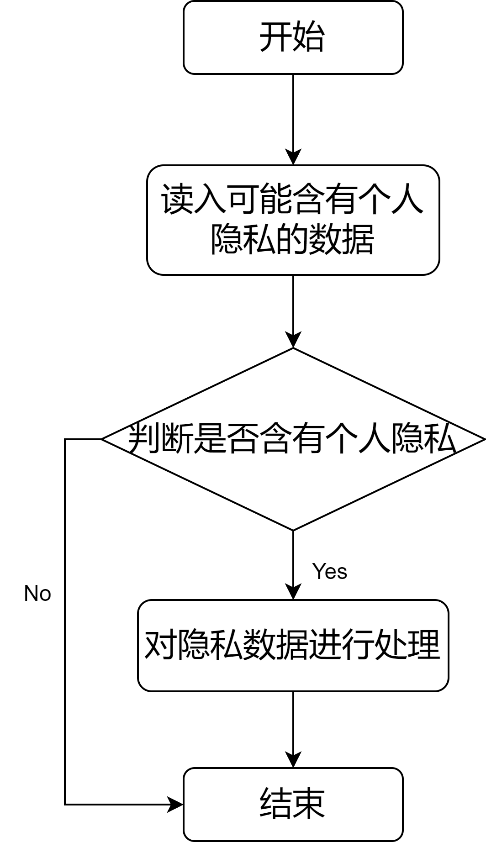


图 5 隐私过滤模块流程图

### 限制条件

隐私信息处理方式较为简单，少数情况会去除失败。

## 安卓模块

### 功能描述

1. 扫描设备应用列表：应用后台读取手机上安装的应用信息。
2. 获取当前页面UI标签信息：利用无障碍服务对当前页面UI信息进行广度优先遍历，并将标签信息按照一定格式发送至模型。
3. 无障碍模拟点击：服务根据模型返回的操作要求实现无障碍自动点击，实现任务的自动化实现。

### 性能描述

响应速度：单次UI获取及模型处理信息返回操作及执行时间在1~2s左右，综合执行一次任务时间在5~10s。

资源占用：无障碍服务已经本地大模型运行占用资源很少，能够在后台顺利运行。

稳定性和兼容性：运行时无闪退情况发生，运行情况良好。

### 输入

用户想要使用的APP以及执行的任务描述。

### 输出

APP逐次的无障碍自动点击结果。

### 程序逻辑

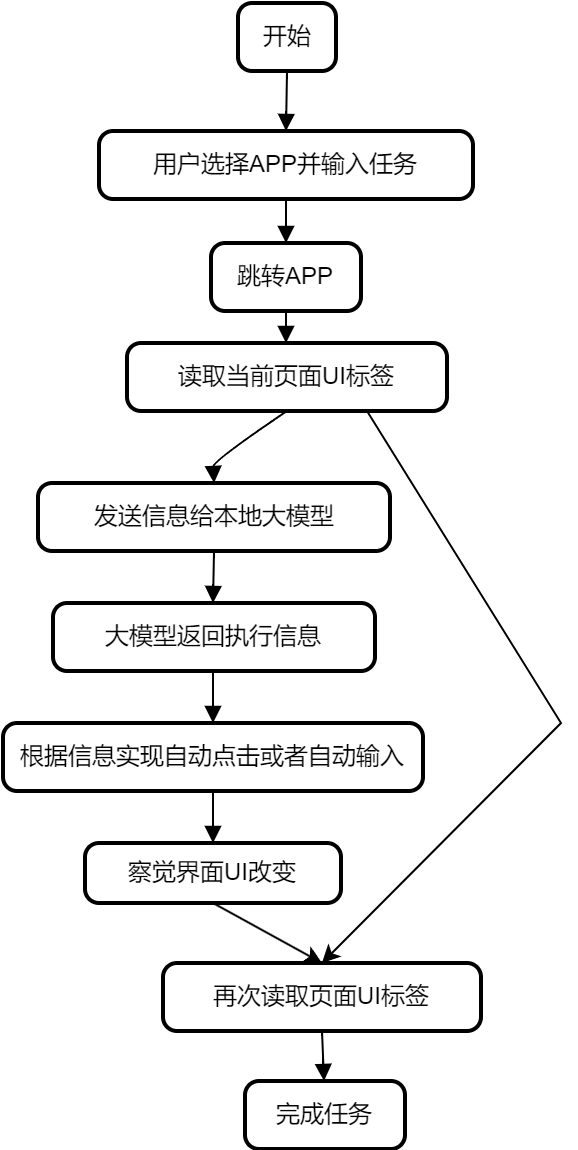


图 6 安卓模块流程图

### 限制条件

网络连接限制：由于本地部署的模型，不需要API进行网络通信，但是操作的APP可能需要联网。

系统设置限制：需要支持并且开启无障碍服务。