

# 重庆大学计算机学院

# 实习实训报告书

学生学号： 20214796

学生班号： 计科03班

学生姓名： 黄天昊

学生专业：计算机科学与技术

实习单位：四川华迪信息技术有限公司

指导教师： 任平

一、专业实习总结报告

|  |
| --- |
| **一、实习项目名称：**基于语音识别和聊天机器人的智能音箱实现  **二、实习时间、地点：**2024年6月24日-2024年7月14日 重庆大学DS3401  **三、实习目的和任务：**随着人工智能技术的不断发展，智能音箱作为一种新型的智能家居设备，逐渐成为人们日常生活中不可或缺的一部分。为了提高智能音箱的智能化程度和用户体验，本项目旨在基于语音识别和聊天机器人技术，开发一款智能音箱实现项目。本项目标是实现基于语音识别的智能音箱功能，包括语音识别、语音合成等；实现基于聊天机器人的智能音箱功能，包括自然语言处理、情感分析、智能推荐等；提高智能音箱的智能化程度和用户体验，满足用户多样化的需求；为其他类似场景提供一种基于语音识别和聊天机器人的智能音箱解决方案。  **四、实习内容和要求：**   1. **需求分析**   智能音箱现在在很多家庭中都有一席之地，是智能家居的重要组成部分，为用户能提供极大的便利和智能化体验。在本次项目中，我们计划开发一款基于语音识别和聊天机器人的智能音箱实现。本项目的核心目标是通过语音交互、自然语言处理和个性化推荐等技术，实现智能音箱的多功能应用，并提供流畅、个性化的用户体验。  首先，语音识别模块应该能实现从用户语音输入到文本输出的准确转换，并且支持多种语言和口音。用户能语音输入向智能音箱发出指令，智能音箱也对指令做出准确且及时的反馈，为用户提供更简单高效的交互方式。在不同的环境噪声当中，也应该有一定的稳定性，能够稳定运行和准确识别。不仅如此，也应当支持实时语音识别和批量语音处理，以满足不同应用场景的需求。  其次在聊天机器人模块，智能音箱应当具备强大的自然语言处理能力，能够实现自然对话的理解与生成，支持语义分析、意图识别和多轮对话管理。并且为了进一步提升智能音箱的智能程度，在与用户聊天的过程中，应该能够识别用户隐藏在语句之中的情绪并相应调整对话策略，提升用户的体验。智能音箱应该具有图像分析功能，使得用户可以上传图像获取相关信息。不仅如此，保存历史记录是了解用户方法的有效措施和提高用户体验和交互效率的重要手段。此举既可以使得用户访问和管理先前的交互记录，也可以方便根据用户历史交互偏好，推荐个性化内容和服务，不断提升用户的体验。  再者，智能音箱的音色和风格应该符合每一位用户的需求。默认的语音无法做到这一点，我们应该提供符合大众需求的音色模型，也要同时兼顾少部分人的需求，而提供更多的音色模型。根据不同的情景，选择不同的音色播放回复，使得智能音箱更加生动，用户体验更加具体。例如，在儿童模式下，使用温柔、友好的音色模型与儿童进行交互；在办公模式下，则使用简洁、清晰的音色模型提供答复。  不仅如此，我们还添加了提示词语音唤醒功能。在通过唤醒词唤醒智能音箱后，我们则能与音箱进行实时的语音交互，此举可以提高智能音箱的便捷性。我们对唤醒词有一个预设值，如果用户不喜欢，也可以根据自己的喜好进行更改。 我们的项目也应该有直观的用户界面设计和具有用户友好性。在web端应该设计直观、响应式的用户界面，包括语音输入和文本聊天界面。界面操作简单明了，支持主题定制和个性化设置。  对于系统的性能需求，应该做到在接收语音输入后应快速响应，实现实时性和流畅的交互体验。且在运行时的资源消耗需控制在合理范围内，避免过度占用用户设备资源；对于系统的安全需求，用户数据在传输和存储时需采取加密措施，确保隐私和数据安全，还应该提供合适的身份验证和访问控制机制，防止未授权访问和数据泄露；对于系统的可维护性需求，系统代码需模块化、清晰易懂，便于后续维护和扩展，提供完整的开发文档、用户手册和日志记录，记录系统设计和变更，方便团队协作和问题追踪。  综上所述，我们的智能音箱项目将为用户提供一个功能丰富、智能化和个性化的家居设备，通过这些功能模块，全面提升用户的生活质量和使用体验。   1. **参与的功能设计**   **2.1 语音输入和识别**  **设计目标：**语音输入和识别模块旨在实现以下设计目标，实现快速、准确的语音转文本功能，支持实时语音输入和即时识别；在各种环境中（包括嘈杂环境）都能保持稳定的识别性能，不受语音质量和背景噪声的影响；支持多种语言的语音输入与识别，满足各种用户的需求；能够根据用户的语音输入习惯和发音特点进行自适应优化，提高识别准确率；实现即时的语音转文本转换，支持实时对话和交互，以提供流畅的用户体验。  **实现技术：**我们基于Python语言和PyTorch深度学习框架，利用其灵活性和效率，实现语音识别模型的训练和部署。在项目中，我们使用了ASRT技术，基于深度学习来实现中文语音识别。ASRT利用了TensorFlow和Keras框架，结合深度卷积神经网络DCNN和长短时记忆神经网络LSTM来处理和解析语音信号，并基于N-gram模型和更复杂的统计方法，用于理解和评估识别后的文本。通过引入注意力机制，我们提高了语音识别的准确率，并采用连接性时序分类CTC算法进行序列建模，从而实现了高效、准确的语音转文字功能。  **2.2 快捷插件和模式**  **设计目标**：由于项目可能会面对许多不同的用户，而这些用户对项目功能有着不同的需求，所以我们可以预先给模型设置的不同提示词，分为不同的模式，为用户提供多样化的应用场景和便捷的使用体验，用户可以按需选择不同的模式。这些模式包括儿童模式、语言翻译、新闻更新、数学计算、百科问答等，能够快速响应用户的特定需求，提高系统的实用性和灵活性。  **实现技术：**在项目中，我们通过对模型设置不同的基础提示词，构建了多个快捷插件和模式。每个模式都有专门的提示词设置，以指导模型在特定情境下生成适合的回复。用户可以根据需求选择相应的模式，系统会自动切换到相应的提示词和处理逻辑。例如，儿童模式使用简单友好的语言，数学计算模式提供详细的计算步骤和解释，语言翻译模式则负责将文本翻译成指定的语言。所有这些模式的实现都依赖于对LLAMA模型的不同提示词设置，并通过Flask框架和HTTP API实现用户交互和模式切换。  **2.3 历史记录存储重用**  **设计目标：**我们还将设计历史记录存储的功能，这不仅可以帮助用户访问和管理先前的交互内容，还能使得模型能够自学习不断了解用户的需求和语句中的情绪感情，能很大程度提高用户体验。通过保存和检索历史记录，用户可以在需要时快速访问过去的信息和操作记录，也可以方便根据用户历史交互偏好，推荐个性化内容和服务，不断提升用户的体验。  **实现技术：**在项目中，我们采用关系数据库MySQL实现历史记录的存储，并通过python中的flask作为前端页面框架传递消息内容，用户每次交互后，用户可以选择将交互内容保存。系统则会将本次的所以交互内容、和用于区分的时间戳和用户标识存储到MySQL数据库中，并确保每条记录都关联到特定用户，保证数据的安全，以及下次可以准确读取交互内容。在下次读取时，可以通过发送get请求，从数据库中获取历史记录，系统按时间顺序返回相关记录，用户可以查看并重用这些记录。根据用户需要，可以选择某条记录重新加载到当前会话中，从而方便地进行后续操作。  **2.4 智能提示词唤醒**  **设计目标：**智能提示词唤醒功能旨提供一种便捷的方式来激活智能音箱并为用户提高服务。该功能应该提供高精度的唤醒词检测，确保在各种噪声环境下都能可靠触发系统，减少误唤醒的情况；设计低功耗的唤醒模块，能够在待机状态下长时间运行；实现快速响应，从唤醒到系统准备接受语音输入的时间应尽可能短，提升用户体验。为了符合所有用户的需求，我们还可以提供了更改提示词的功能，使其更加个性化。  **实现技术：**在项目中，我们通过实时监听周围环境的音频流，在获取后进行检测和匹配，如果用户的语音输入与唤醒词匹配，则开启实时语音交流互动的功能。为了实现这些功能，我们采用基于声学模型的唤醒词检测算法，通过实时音频流进行检测和匹配，确保高准确率的唤醒词触发。引入噪声抑制技术和环境适应性算法，提高在复杂环境下的唤醒词检测能力。   1. **项目架构**   **3.1 前端架构**  **技术栈：**Flask、HTML、CSS、JavaScript、jQuery  前端主要负责用户交互界面的实现，包括用户注册、登录、语音输入、文本输入、图像上传、历史记录查看和管理等功能。前端采用Flask框架，通过Jinja2模板引擎渲染HTML页面，并使用CSS进行样式设计。JavaScript和jQuery用于实现前端动态交互和AJAX请求，以便与后端进行数据通信。  **主要组件：**   1. **用户界面：**提供用户注册、登录、语音输入、文本输入和图像上传等功能的界面。 2. **插件和模式选择：**用户可以通过前端界面选择不同的插件和模式，如儿童模式、语言翻译、新闻更新、数学计算和百科问答等。 3. **历史记录管理：**用户可以查看、重用和管理过去的对话记录。 4. **实时语音唤醒：**提供提示词语音唤醒的设置和使用界面。   **3.2 后端架构**  **技术栈：**Flask、Python、MySQL、SocketIO、PyTorch  后端主要负责业务逻辑的实现，包括用户管理、语音识别、自然语言处理、情感分析、图像分析和数据存储等功能。后端使用Flask框架搭建Web服务，MySQL数据库用于存储用户信息、历史记录和系统配置。SocketIO用于实现实时通信，PyTorch用于深度学习模型的加载和推理。  **主要组件：**   1. **用户管理模块：**负责用户注册、登录、修改个人设置和注销登录等功能。 2. **语音识别模块：**使用ASRT技术进行语音转文字识别。 3. **自然语言处理模块：**调用预训练的LLAMA3:8b模型进行文本问答和情感分析。 4. **图像分析模块**：使用LLAVA模型进行图像内容分析。 5. **情感分析模块：**基于BERT模型的情感分析，实现对用户文本情感状态的识别和反馈。 6. **推荐系统模块：**分析用户历史记录，提供个性化的搜索推荐。 7. **提示词语音唤醒模块：**实时监听用户的语音输入，检测特定的提示词以唤醒设备。   **3.3 数据库架构**  **技术栈：**MySQL  数据库用于存储用户信息、历史记录、系统配置和其他相关数据。通过设计合理的数据表结构，确保数据的存储和访问高效、安全。  **主要表结构：**   1. **用户表：**存储用户的基本信息，如用户名、密码、邮箱等。 2. **历史记录表：**存储用户的对话历史记录，包括时间戳、对话内容和用户标识。 3. **系统配置表：**存储系统的各种配置项，如提示词、音色配置等。   **3.4 通信架构**  **技术栈：**HTTP、WebSocket  前后端通过HTTP协议进行常规数据通信，例如用户注册、登录、数据请求等。对于需要实时交互的功能，例如实时语音唤醒和流式文本问答，采用WebSocket进行通信，以确保数据的实时性和交互的流畅性。   1. **功能测试**   **4.1 语音输入和识别**  **测试目标：用户**语音输入是否能够转换为文本且是否准确。  **测试方法：**  在不同环境噪声条件下，提供语音输入，测试语音识别是否准确。再给出多种语音输入，测试算法识别的准确率。最后给出多种用户不同语气语调的输入，测试对多样化语音输入的适应能力。  **测试步骤：**   1. 打开项目的web端，输入一段语音，检查输出文本内容。 2. 在不同的环境噪声条件下，提供语音输入，检查输出文本内容。 3. 输入完中文后，输入一段英文语音，检查输出文本内容。 4. 切换测试人员，输入不同的语气语调的语音，检查输出内容。   **4.2 快捷插件和模式**  **测试目标：**测试不同插件和模式下功能的实用性。  **测试方法：**  测试儿童模式、语言翻译、新闻更新、数学计算、百科问答等插件。进入儿童模式下，输入问答内容，检查回复的结果以及输出的语音是否符合儿童需求。切换模式，测试是否符合相应功能需求。测试插件切换的便捷性和响应速度。  **测试步骤：**   1. 选择不同的插件和模式。 2. 输入相应的问题或命令，观察系统反应。   (3) 记录每个模式的功能表现和用户体验。  **4.3 历史记录存储重用功能**  **测试目标：**测试历史记录是否能够准确存储和重用。  **测试方法：**  测试系统对用户交互记录的存储和检索功能。与模型进行正常的交流对话，在对话完后保存历史记录。再点击进入历史记录，检查是否准确和完整。再测试是否能够重用，确保记录的连贯和一致性。  **测试步骤：**   1. 进行多次交互，生成历史记录。 2. 检索并查看历史记录，检查其完整性。   (3) 重用历史记录，观察系统反应。  **4.4 智能提示词唤醒**  **测试目标：**测试是否能够准确检测唤醒词且是否能够唤醒设备。  **测试方法：**  测试不同环境噪声条件下提示词的检测效果。检查提示词唤醒的响应速度和准确性。测试提示词的更改功能，确保用户自定义提示词的有效性。  **测试步骤：**   1. 设置不同的提示词。 2. 在多种环境下进行提示词唤醒测试。 3. 检查系统的唤醒反应和准确性。 4. **项目部署**   我们的项目部署采用将整个项目打包为可执行文件（exe）的方式，通过使用pyinstaller工具将Python代码及其依赖库打包成一个独立的可执行文件。首先，我们确保项目中的所有依赖库都在requirements.txt文件中列出，然后编写一个启动脚本来启动Flask应用。接着，使用pyinstaller命令将项目打包为一个单独的exe文件，并包含所有必要的模板和静态文件。最终生成的exe文件可以在任何Windows系统上直接运行，无需安装Python环境。通过这种方式，用户可以方便地使用我们的智能音箱系统，而不必进行复杂的环境配置。   1. **实习工作成果：**   首先进入登录界面，输入账号密码后进入主界面    登录成功后跳转至主界面    左上角是快捷插件和模式，可以根据不同需求选择    左边中间是情感分析，能够根据用户说话分析情感    当感受到开心后，情感分析也会随之改变    左边下面是聊天记录，可以保存，删除，重命名    主体部分则是聊天框，可以进行对话，播放语音。可以文本输入或者语音输入，也可以传入图片进行分析，下面还有智能推荐，根据聊天历史记录可以推荐内容    点击头像之后，可以进入自定义选项，修改头像，调整音色模型    还可以进入深夜模式    还可以选择进入监听模式，通过唤醒口令唤醒，进行实时对话  **六、总结及心得体会：**  我们小组选择了基于语音识别和聊天机器人的智能音箱实现作为本次的项目。这个项目旨在利用先进的人工智能技术，为用户提供高效、自然的语音交互体验，并结合情感分析与智能推荐功能，以实现更个性化、智能化的服务。在本次的20天实训中，我们一步一步，从无到有，从需求分析、功能设计到实现和测试，完成了这个项目，看着自己做出的项目慢慢完善，内心的喜悦溢于言表。通过本次的实训经历，我不仅提升了人工智能方向上的技术，也在项目管理、团队协作和问题解决方面积累了宝贵的经验。  这次实训中，我在技术方面掌握到了多项关键技术。我们采用python语言结合pytorch框架，实现了transformer模型等先进算法。这些技术不仅为系统的性能提升奠定了基础，也增强了我们对人工智能领域的理解和应用能力。不仅如此，我还学会了运用ASRT语音识别算法，来解决语音转文字方面的复杂问题；还学会了使用LLAMA和LLAVA的集成模型，来实现高效的基础问答和分析图像的功能；还学会了使用GPT-SoVITs语音合成技术，来展现多种音色和语气的生成语音。在整个项目的设计和开发过程中，我们始终以用户体验为中心。通过设计直观简洁的Web端界面，并结合流畅的语音交互，我们努力确保用户能够轻松地使用和享受我们的产品。不仅学到了各个技术模块的实现原理和应用场景，还通过实际操作进一步巩固了这些知识。  除了技术方面的收获，我还在这次实训中深刻体会到了小组分工协作的重要性。在整个项目中，团队合作和良好的项目管理起到了至关重要的作用。我们采用了CMMI3与RUP标准，确保了项目的质量和效率。每个成员在各自的领域发挥了专业能力，并通过持续的沟通和协作解决了遇到的各种问题。这种团队精神不仅提升了项目的成果，也加强了我们之间的信任和合作关系。在项目开始之初，我们就开展了小组会议，对需求分析和功能设计进行了详细的分析，为后续整个项目的稳步推进奠定了基础。并且，我们讨论了各自的分工合作，每个人都负责不同的功能模块，防止有重复的工作耽误效率，并且在完成了任务之后，也会对其他小组成员进行力所能及的帮助，确保整个项目进度不会推迟。不仅如此，我们每周都会进行小组会议，讨论项目的进度，和后续工作安排，确保问题能够及时发现并且解决，此举保证了我们项目能够如期交付。通过本次经历，我能更好的处理团队之间的沟通和协作，如何能使得项目更有灵活性和计划性。  当然在项目开发过程中，我们也并不是一帆风顺的，我们也遇到了许多挑战和困难。例如，文本转语音的效率和语音连贯性问题以及语音识别和实时监听在不同噪声环境中的转换准确率问题。这些问题毫无疑问是我们通向成功的绊脚石，阻碍了我们的效率。对于此类问题，我们会召开小组会议，讨论问题的多种解决方案，并选取一种最合理的方案，最后分配至个人进行解决，通过这样我们逐一克服了这些难题。  随着项目的收尾和web端的正式发布，我们对未来充满了期待。我们将继续关注使用情况和反馈，进一步优化系统的性能和功能。参与这个项目不仅让我获得了丰富的技术经验，更深刻地体会到了团队合作和项目管理的重要性。在未来的工作中，我将继续秉持学习和创新的精神，不断挑战自我，为推动技术进步和提升用户体验贡献自己的力量。  通过这次项目，我不仅收获了技术上的成就，还结识了优秀的组员们，共同追求着技术创新和服务卓越的目标。我深信，这段宝贵的经历将成为我学习生涯中的重要财富，激励我在未来的工作中不断进步和成长。    学生本人签名：  2024年7月12日 |

指导老师评语：

1. 学生实践期间的思想品德表现，身心健康与安全生产方面的意识
2. 学生沟通能力、协作能力等综合能力的锻炼情况
3. 学生实践过程中的工作内容、执行情况、主要成果评价

实习成绩：

指导教师签字：

（实习单位盖章）

年 月 日