

基于微信平台和大语言模型的信息自动提取与存储系统设计与实现

蒋靖雯, 侯卓轩, 毕晓琳

(广东东软学院, 广东 佛山 528225)

摘要 本文设计了一款基于微信平台和大语言模型的会谈邀约信息自动提取与存储系统。该系统无感知地从微信信息智能提取会谈邀约关键信息, 同时将信息进行云同步。研究通过 UML 进行系统需求分析和设计, 系统包括展现层、数据处理层和数据层, 主要功能包括创建与会企业和人员信息、监听聊天记录并存储会议数据, 供用户随时查询留档。该系统提升了会谈信息记录效率, 从而提高办公效率, 符合信息时代的智能化办公的发展需求。

关键词 大语言模型; 会谈邀约信息; 自动存储系统

中图分类号: TP3

文献标志码: A

文章编号: 2097-3365(2024)09-0013-03

1 研究背景

随着社会信息化的不断深化, 微信作为我国主要沟通平台在日常工作中扮演着日益重要的角色。尽管各类组织频繁组织会谈, 然而相关信息记录却未能得到及时有效的整理和存储。截至 2024 年 1 月统计, 微信用户数量已接近 14 亿, 成为国内主流社交工具之一。然而, 会谈邀约信息的记录和管理却存在滞后和不及时的问題^[1], 给工作人员带来繁琐耗时的挑战, 从而直接影响会谈的组织效率和执行力。

在此背景下, 大语言模型逐渐成熟, 国家相关政策也对其应用提供了支持^[2]。国家发改委等部门于 2022 年初明确鼓励企业信息化、管理信息化, 并积极推进大语言模型的应用落地。这一政策的目标是通过新技术的应用提高企业的管理效率和竞争力, 为各行各业创造更多的发展机遇。

因此, 本文旨在设计一款会谈邀约信息自动提取与存储系统, 充分利用微信平台的便捷性和大语言模型的强大处理能力^[3], 提高组织内会谈邀约信息的记录效率, 减少信息记录的时间成本, 为企业和组织提供智能高效的信息管理解决方案, 以适应信息时代的发展潮流。这样的系统有望为提升会谈效率和管理水平提供实质性的支持, 让企业在竞争激烈的环境中脱颖而出。

2 系统设计与功能概述

2.1 系统设计思路

本系统设计的核心原则是无感知地实现会谈邀约信息的自动记录与管理, 并与微信平台紧密结合。这

种设计旨在为用户提供一种零干扰、高效智能的会谈邀约信息管理方式, 使用户能够专注于交流工作内容, 同时摆脱繁琐的信息记录过程。

2.2 核心设计特征

1. 无需额外应用软件: 系统与微信平台集成, 无需用户安装其他应用软件, 为用户提供便捷的会谈邀约信息记录服务。

2. 无感知信息提取: 系统通过智能技术自动获取用户的微信聊天记录, 实时分析是否包含会谈邀约信息, 并通过大语言模型提取并存储邀约的关键信息。

3. 智能云同步: 会议邀约信息安全存储至云服务器, 实现智能云同步。用户可随时查看和管理信息, 确保信息备份和安全性。

2.3 系统功能优势

1. 微信平台集成: 系统使用 WechatPyApi 与微信平台无缝结合。

2. 智能信息提取与处理: 通过大语言模型自动提取会谈信息, 过程透明。

3. 云端信息存储与同步: 信息被安全地存储于云端, 可随时访问和管理, 确保数据安全。

4. 时序数据处理与会议管理: 动态处理微信端的会谈信息, 实时更新, 自动创建和管理会议信息。

5. 数据层管理与优化: 采用高效的事务管理和数据库技术, 保证数据完整性和一致性。

6. 高效数据传输与处理技术: 确保系统响应速度和处理效率。

7. 稳定的运行环境: 部署在腾讯云服务器上, 使

用 SQLITE3 数据库，保障系统稳定性和可靠性。

3 系统分析与设计

3.1 系统功能分析

在系统功能分析时，将运用统一建模语言(Unified Modeling Language, UML)对本平台系统的功能需求进行建模分析。对系统功能分析时的时序图，如图1所示。时序图展示了用户使用平台时后端系统的一系列基础活动。时序图有三个不同模块，分别是微信端、大语言模型和数据库后端。系统应实现的功能如下。

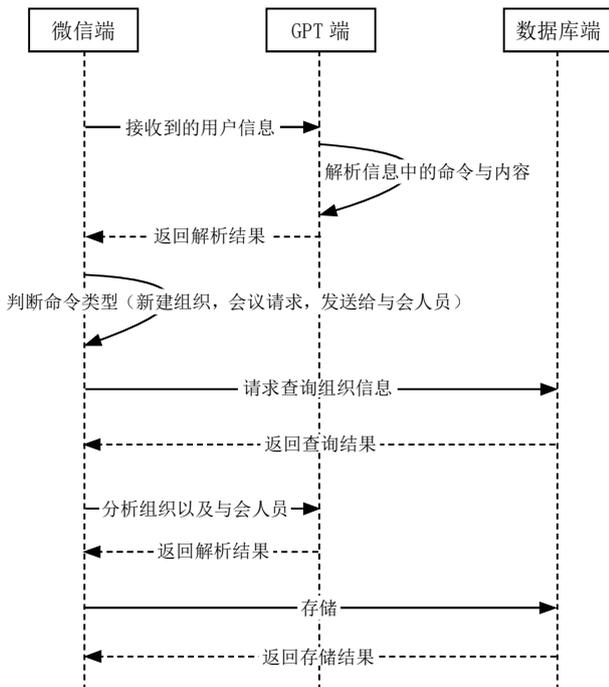


图1 系统时序图

1. 信息采集与处理。(1) 消息接收与初步分析：系统通过微信端持续监听用户与与会者的实时聊天，当接收到一条包含会议相关信息的信息（如关键词、任务指示等）时，微信端将消息以 JSON 格式发送至大语言模型进行深度分析。为了避免因常见问候语引起的误触发以及时效性和准确性等问题，大语言模型会从最近的三条消息开始逐一进行判断。(2) 会议信息启动与确认：当大语言模型识别到“会议启动”相关指示时，会将“开始记录”信号返回微信端，指示从最近的三条消息开始记录。每一条新消息都会再次传回大语言模型进行详细解析，直到大语言模型确认会议已正式开始。此时，大语言模型会识别并存储邀请者和被邀请者信息，将他们加入数据库的与会人员列表。(3) 会议信息存储：一旦会议信息完成记录，大语言模型会确认并返回“结束记录”信号给微信端，

通知微信端存储会议详情，包括参与人员、会议时间等关键信息。数据库后端会将这些数据持久化，以便后续查询。

2. 功能交互与查询。查询功能：用户通过微信端发送查询请求，大语言模型接收并解析该请求，随后向数据库端发送查询组织信息的指令。数据库端响应请求并返回查询结果，大语言模型对结果进行解析后传递给微信端，微信端将结果展示给用户。

3.2 系统架构

系统架构分为展现层、数据处理层和数据层。展现层是微信聊天框，是用户与系统进行交互的主要界面。数据处理层位于服务器后端，负责处理和分析从微信聊天框（展现层）发送过来的数据。此外，数据处理层还包括大语言模型的调用和处理。数据后端提供了创建组织、创建会议信息和获取信息等功能。为了提高数据处理的效率，本系统在数据层管理和优化数据的存储与读取来确保数据的完整性和一致性^[4]。最后，整个系统运行在腾讯云服务器上，使用 SQLITE3 作为数据库。

3.3 数据库设计

SQLITE3 数据库为本文设计的信息自动提取与存储系统选用的数据库，进入系统数据库的数据主要被划分为日志型数据与内容型数据。日志型数据主要用于记录会议内容以及用户信息；内容型数据则被用于微信端研究的数据。由于本系统在使用过程中产生和存储的数据量较大，因此会存在大量重复数据。对于重复数据，SQLITE3 支持内置的唯一约束，并且优化了数据结构，避免不必要的冗余，将利用这些特性来保证数据的准确性。

Node.js 作为后端开发框架，其异步 I/O 处理能力有助于高效地执行数据库操作和服务器逻辑管理^[5]。前端交互方面，python 被应用于微信端和 GPT 端的接口开发，实现了前后端之间的无缝协作，实现对成员、会议、组织相关数据的灵活管理。

3.4 系统和用户的数据流图

根据功能需求，本文绘制出系统和用户的数据流图(Data Flow Diagram)，如图2所示。数据流图展示了系统中不同模块之间的数据流动和处理过程。为实现对微信会议消息的高效处理和管理划分了四个关键模块，确保整个系统协同运作。首要模块是微信消息监控模块，负责实时监听微信会议消息，捕捉关键信息如参会人员的公司和姓名。然后是信息提取与处理模块，识别并提取得到的原始数据，高效转化为结构

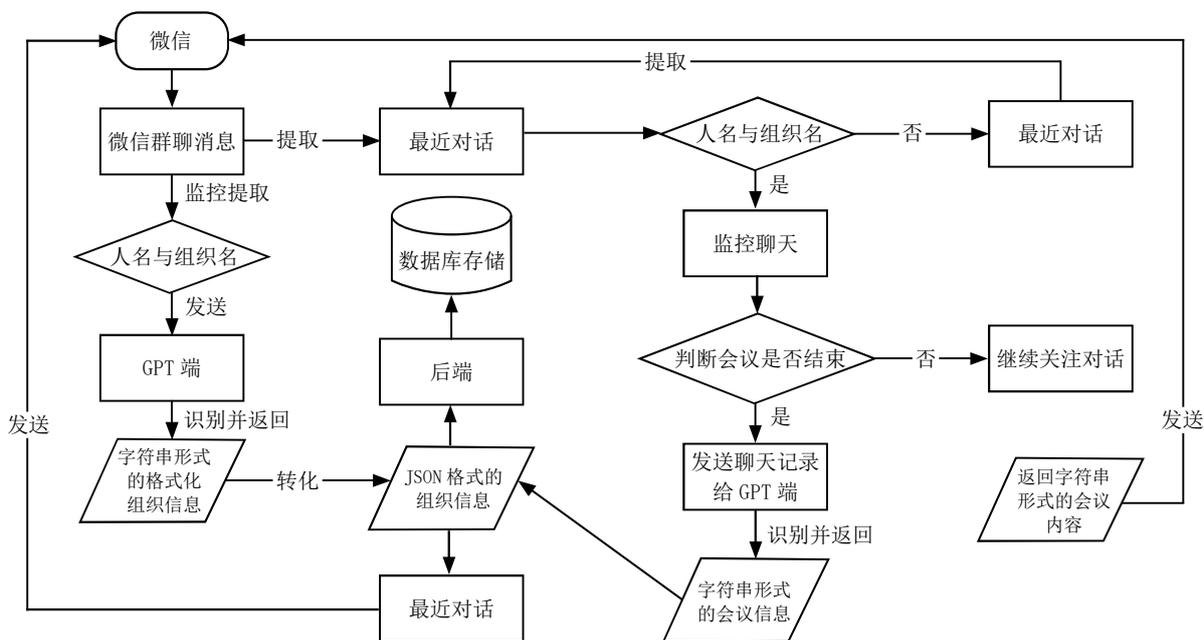


图 2 系统设计数据流图

化数据，提交到智能处理引擎模块进行下一步处理。系统核心智能处理引擎采用 GPT 模型，接收并处理信息提取模块传递的数据，生成具有特定格式的输出（JSON 格式），以便与数据库进行交互。最后是数据库管理模块，接收来自 GPT 端的输出，解析 JSON 格式的信息，并执行相应的数据库操作，如创建新组织、存储会议信息等。

4 系统功能实现

4.1 创建与会企业对象和人员功能

为了准确识别需要监听的微信聊天用户，首先需要在数据库中建立相关档案。系统设计为直接通过微信进行数据库操作，避免使用其他平台进行数据输入。用户向文件传输助手发送创建企业和人员的请求，大语言模型将识别并分析该请求。

系统利用 WechatPyApi 接收用户请求，并将信息传递至大语言模型进行解析。如果识别到创建企业或人员的指令，系统将在数据库中新增相应记录。记录新增成功后，系统会通过文件传输助手反馈新增企业结果。

4.2 监听与会人员的聊天记录并存储会议数据

当用户发送聊天信息时，系统将信息传递给大语言模型进行分析。大语言模型会识别信息中的关键信息。如果检测到会谈邀约信息，大语言模型将提取相关数据，并发送至数据库端进行存储。存储成功后，

系统会将关键信息组织成自然语言，通过文件传输助手反馈给用户。

5 结束语

本文专注于微信平台的会谈邀约信息的自动提取与存储管理，通过融合大语言模型与 WechatPyApi，设计了一种无感知信息提取的会谈管理系统。系统能自动解析聊天记录，实现实时信息存储于云端，无需额外软件，显著提升工作效率。其创新地结合了大模型与微信 API，简化了会谈管理。虽然基础功能已就绪，但仍有优化空间，如跨平台兼容和用户隐私保护。未来计划聚焦性能优化和用户体验，目标是智能办公提供更高效率便捷的解决方案，助力企业在信息时代保持竞争优势。

参考文献：

- [1] 郑杨. 基于微信平台的档案信息服务发展探究[J]. 理论观察, 2022(12):109-112.
- [2] 聂传清、汪文正. 推进生活服务业数字化转型升级(锐财经)[N]. 人民日报海外版, 2023-12-23(03版).
- [3] 张丹. 大语言模型与档案资源开发: 前景、挑战与应对[J]. 山西档案, 2023(05):108-111.
- [4] 黄葵, 邱宇, 孔藤桥. 云计算如何助力提升企业信息管理效率[J]. 办公自动化, 2023, 28(23):8-10.
- [5] 刘红卫. Node.js 对多种数据库协同访问实践: 以图书馆短信服务系统为例[J]. 天津科技, 2019, 46(07):77-81.