# **AI 校务问答机器人质量改进方案**

## **一、项目现存问题**

### **（一）功能实现问题**

1. ****问题覆盖度不足****：当前系统仅能回答 10 类基础校务问题，无法满足学生多样化需求。例如，学生反馈无法查询 “第二课堂学分认定标准”“留学生奖学金申请流程” 等个性化问题。
2. ****多轮对话逻辑混乱****：在处理复杂问题时，如 “如何申请转专业”，机器人无法根据用户的追问（如 “需要哪些材料”）关联上下文，导致回答零散、不连贯。
3. ****缺乏个性化服务****：无法根据用户身份（如本科生、研究生、教职工）提供差异化回答，例如研究生的学术活动审批流程与本科生不同，但机器人未做区分。

### **（二）数据处理问题**

1. ****数据来源单一****：仅依赖人工编写的少量问答对，未整合学校官网动态信息（如教务处通知、图书馆新书上架）、各学院特色课程信息等。
2. ****数据时效性差****：课程表、考试安排等数据更新不及时，导致学生查询到过期信息。例如，某课程考试时间调整后，机器人仍提供旧时间。

### **（三）交互体验问题**

1. ****界面设计简陋****：采用控制台文本交互，缺乏直观的视觉引导。学生反馈 “输入问题后等待时间不明确，不知道系统是否在处理”。
2. ****回答形式单一****：所有回答均为纯文字，对于复杂问题（如图书馆楼层分布），无法提供可视化图表或地图导航。

### **（四）技术实现问题**

1. ****自然语言理解薄弱****：基础关键词匹配算法无法识别同义表述，如 “选课” 与 “课程选修” 被视为不同问题。
2. ****系统响应缓慢****：处理多用户请求时，平均响应时间超过 5 秒，影响用户体验。

## **二、用户需求分析**

通过问卷调查（回收 120 份有效问卷）和焦点小组访谈，整理核心需求如下：

1. ****高频问题类型****：课程查询（35%）、考试安排（28%）、校园卡服务（15%）、图书馆资源（12%）、活动报名（10%）。
2. ****功能期望****：
   1. 82% 的用户希望支持多轮对话，如 “先查询教室位置，再询问该教室明天是否可用”。
   2. 65% 的用户期望个性化服务，如根据学号自动关联个人课表、成绩等信息。
   3. 58% 的用户需要即时通知功能，如考试时间变更、校园卡挂失成功提醒。
3. ****交互偏好****：
   1. 70% 的用户倾向于使用微信小程序或网页端，而非命令行工具。
   2. 60% 的用户希望回答中包含链接、图片等富媒体元素。

## **三、质量改进方案**

### **（一）功能完善**

1. ****扩展问题覆盖范围****
   1. ****实施方法****：按用户需求优先级，新增 “第二课堂”“留学生服务”“学术活动” 等 15 类问题库。
   2. ****示例****：针对 “转专业流程” 问题，设计子问题引导：“请问您是本科生还是研究生？”→“需要准备成绩单、个人申请书等材料”。
2. ****优化多轮对话逻辑****
   1. ****技术方案****：引入有限状态机（FSM），使用 Python 的transitions库实现对话状态管理。
3. ****个性化服务实现****
   1. ****数据关联****：设计用户认证模块，允许绑定学号后查询个人课表、成绩等信息。
   2. ****差异化回答****：在知识库中为不同用户角色（本科生 / 研究生）标注专属回答。

### **（二）数据优化**

1. ****多源数据整合****
   1. ****官网爬虫****：使用requests和BeautifulSoup每周自动爬取教务处、图书馆公告，更新知识库。
   2. ****API 对接****：尝试对接学校现有系统（如校园卡系统）获取实时数据。
2. ****数据时效性保障****
   1. ****定期更新机制****：设置定时任务，每天凌晨同步课程表、考试安排等动态数据。
   2. ****人工审核流程****：重要通知（如考试变更）需人工确认后才推送给用户。

### **（三）交互体验提升**

1. ****界面升级****
   1. ****Web 端开发****：使用 Bootstrap 5 构建响应式网页界面，包含加载动画、问题分类标签。
   2. ****微信小程序****：同步开发轻量级小程序，支持消息推送功能。
2. ****富媒体回答****
   1. ****图片生成****：对于图书馆布局等问题，生成楼层示意图（使用 Python 的matplotlib库）。
   2. ****链接跳转****：在回答中嵌入官网链接，如 “点击此处查看最新奖学金政策”。

### **（四）技术改进**

1. ****自然语言处理优化****
   1. ****关键词扩展****：建立同义词库（如 “选课”→“课程选修”“课程选择”）。
   2. ****相似度计算****：使用 TF-IDF 向量空间模型，提高语义匹配准确率。
2. ****性能优化****
   1. ****缓存机制****：对高频问题（如 “食堂营业时间”）设置 1 小时缓存。
   2. ****异步处理****：使用 Flask 的async模块处理并发请求，将响应时间缩短至 2 秒内。

## **四、实施计划（4 周）**

| **阶段** | **任务** | **负责人** | **时间线** |
| --- | --- | --- | --- |
| 第一周 | 需求分析报告定稿、问题库扩展至 30 类 | 全体成员 | 第 1-3 天 |
|  | 完成多轮对话模块设计与编码 | 开发组 | 第 4-7 天 |
| 第二周 | 实现数据爬虫与定时更新机制 | 开发组 | 第 8-11 天 |
|  | 完成 Web 界面原型开发 | UI 组 | 第 12-14 天 |
| 第三周 | 优化 NLP 算法、实现个性化服务 | 算法组 | 第 15-18 天 |
|  | 开发微信小程序基础功能 | 开发组 | 第 19-21 天 |
| 第四周 | 系统集成测试、用户反馈收集 | 测试组 | 第 22-25 天 |
|  | 修复 BUG、最终版本交付 | 全体成员 | 第 26-28 天 |

## **五、预期效果**

1. ****功能指标****：问题覆盖率提升至 85%，多轮对话成功率≥70%。
2. ****性能指标****：平均响应时间≤2 秒，并发支持 100 用户同时在线。
3. ****用户体验****：界面满意度评分≥80 分，问题解决率提升至 65%。