

基于大模型的自然语言转SQL查询工具 - 需求分析文档

一、项目背景与意义

随着数据库技术的广泛应用，越来越多的用户需要与数据库进行交互。然而，传统的SQL查询需要用户具备专业的SQL知识，这对于非技术用户来说是一个较大的障碍。基于大语言模型的自然语言转SQL查询工具应运而生，它可以将用户的自然语言描述自动转换为可执行的SQL查询，降低了用户使用数据库的门槛，提高了数据查询的效率。

本项目旨在开发一个基于大语言模型的自然语言转SQL查询工具，支持用户通过自然语言进行数据库查询，并具备自动错误修正和E-R图可视化等功能，为用户提供便捷、高效的数据查询体验。

二、功能需求

2.1 核心功能

2.1.1 自然语言转SQL

- 功能描述：用户输入自然语言查询，系统自动生成相应的SQL语句
- 输入：自然语言文本
- 输出：SQL查询语句
- 技术要求：
 - 支持中文自然语言输入
 - 生成的SQL语句语法正确，能够在SQLite数据库中执行
 - 支持复杂查询，包括多表关联查询、条件查询、排序、分组等

2.1.2 SQL执行与结果展示

- 功能描述：执行生成的SQL语句，并将查询结果以直观的方式展示给用户
- 输入：SQL查询语句
- 输出：查询结果表格
- 技术要求：
 - 支持SQLite数据库
 - 结果展示支持分页、排序等操作
 - 结果展示清晰、美观

2.1.3 自动错误修正

- 功能描述：当生成的SQL语句执行失败时，系统自动将错误信息反馈给大模型，进行自我修正
- 输入：错误的SQL语句和错误信息
- 输出：修正后的SQL语句

- **技术要求：**

- 能够识别常见的SQL错误，如语法错误、表名/列名错误、关联条件错误等
- 支持多次修正尝试
- 修正过程透明，用户可查看修正历史

2.2 加分功能

2.2.1 E-R图可视化

- **功能描述：**生成并展示数据库的实体关系图，帮助用户理解数据库结构
- **输入：**数据库结构信息
- **输出：**E-R图
- **技术要求：**
 - 清晰展示数据库中的表、列和表之间的关系
 - 支持交互式操作，如缩放、拖动等
 - 支持导出为图片或PDF格式

2.2.2 数据库结构查看

- **功能描述：**展示数据库的表结构、列信息、外键关系等
- **输入：**表名
- **输出：**表结构信息
- **技术要求：**
 - 展示表的列名、数据类型、约束等信息
 - 展示表之间的外键关系
 - 支持查看表的示例数据

三、非功能需求

3.1 性能要求

- **响应时间：**自然语言转SQL的响应时间不超过5秒
- **并发处理：**支持10个并发用户
- **查询执行：**简单查询的执行时间不超过1秒

3.2 可用性要求

- **易用性：**界面简洁、直观，用户无需培训即可使用
- **可靠性：**系统稳定运行，无崩溃现象
- **容错性：**能够处理用户的错误输入，给出友好的提示

3.3 可扩展性要求

- **模块化设计**: 系统采用模块化设计，便于后续功能扩展
- **支持多种大模型**: 支持集成不同的大模型，如OpenAI、Anthropic、Ollama等
- **支持多种数据库**: 支持扩展到其他数据库类型，如MySQL、PostgreSQL等

3.4 安全性要求

- **数据安全**: 保护用户输入的敏感信息
- **SQL注入防护**: 防止恶意SQL注入攻击
- **权限控制**: 支持基本的权限控制，限制用户对数据库的操作权限

四、系统架构需求

4.1 技术栈

- **前端框架**: Streamlit
- **后端语言**: Python
- **大模型**: 支持OpenAI、Anthropic、Ollama等
- **数据库**: SQLite
- **可视化库**: Graphviz、Matplotlib

4.2 架构设计

- **模块化设计**: 系统分为前端层、服务层、数据层
- **松耦合**: 各模块之间通过清晰的接口进行通信，便于独立开发和测试
- **可配置**: 支持通过配置文件调整系统参数，如大模型类型、API密钥等

五、数据需求

5.1 数据库设计

- **模拟场景**: 图书管理系统
- **表结构**:
 - 出版社表 (publishers)
 - 作者表 (authors)
 - 图书分类表 (categories)
 - 图书表 (books)
 - 图书-作者关系表 (book_authors)
 - 图书-分类关系表 (book_categories)
 - 用户表 (users)
 - 借阅记录表 (borrow_records)

5.2 示例数据

- 出版社数据：至少5条示例数据
- 作者数据：至少5条示例数据
- 分类数据：至少5条示例数据
- 图书数据：至少5条示例数据
- 用户数据：至少3条示例数据
- 借阅记录数据：至少4条示例数据

六、用户界面需求

6.1 页面布局

- 左侧边栏：功能导航、系统信息
- 主内容区：功能模块展示
- 顶部导航：系统标题、用户信息

6.2 功能模块布局

- 自然语言查询：输入框、执行按钮、结果展示区
- E-R图可视化：图表展示区、操作按钮
- 数据库结构：表列表、表详情、示例数据

6.3 交互设计

- 实时反馈：操作过程中给予用户实时反馈
- 清晰的错误提示：当操作失败时，给出清晰、友好的错误提示
- 响应式设计：适应不同屏幕尺寸

七、测试需求

7.1 功能测试

- 单元测试：对核心模块进行单元测试，覆盖率不低于80%
- 集成测试：测试各模块之间的集成情况
- 端到端测试：测试完整的业务流程

7.2 性能测试

- 响应时间测试：测试系统的响应时间
- 并发测试：测试系统的并发处理能力

7.3 兼容性测试

- 浏览器兼容性：测试在不同浏览器中的表现
- 操作系统兼容性：测试在Windows和Mac系统中的表现

八、项目验收标准

8.1 功能验收

- 自然语言转SQL功能正常工作，支持复杂查询
- SQL执行和结果展示功能正常
- 自动错误修正功能能够处理常见的SQL错误
- E-R图可视化功能正常，能够清晰展示数据库关系
- 数据库结构查看功能正常

8.2 性能验收

- 自然语言转SQL的响应时间不超过5秒
- 简单查询的执行时间不超过1秒
- 支持10个并发用户

8.3 可用性验收

- 界面简洁、直观，用户易于使用
- 系统稳定运行，无崩溃现象
- 错误提示清晰、友好

8.4 文档验收

- 需求分析文档完整、清晰
- 概要设计文档详细、合理
- 项目报告包含完整的开发过程和测试结果
- 代码注释完整，符合编码规范

九、风险评估

9.1 技术风险

- 大模型生成SQL的准确性：大模型可能生成不准确或错误的SQL语句
- 系统性能问题：大模型调用可能导致系统响应时间过长
- 数据库兼容性问题：不同数据库的SQL语法可能存在差异

9.2 解决方案

- 多轮修正机制：当SQL执行失败时，系统自动进行多次修正尝试

- **性能优化**: 对大模型调用进行缓存，减少重复调用
- **模块化设计**: 支持不同数据库类型的扩展

十、总结

本需求分析文档详细描述了基于大模型的自然语言转SQL查询工具的功能需求、非功能需求、系统架构需求、数据需求、用户界面需求、测试需求和项目验收标准。该文档为项目的后续设计和实现提供了明确的指导，确保项目能够按照预期目标完成，为用户提供高质量的数据查询体验。