

原创案例：Qwen Agentic RAG 智能问答系统

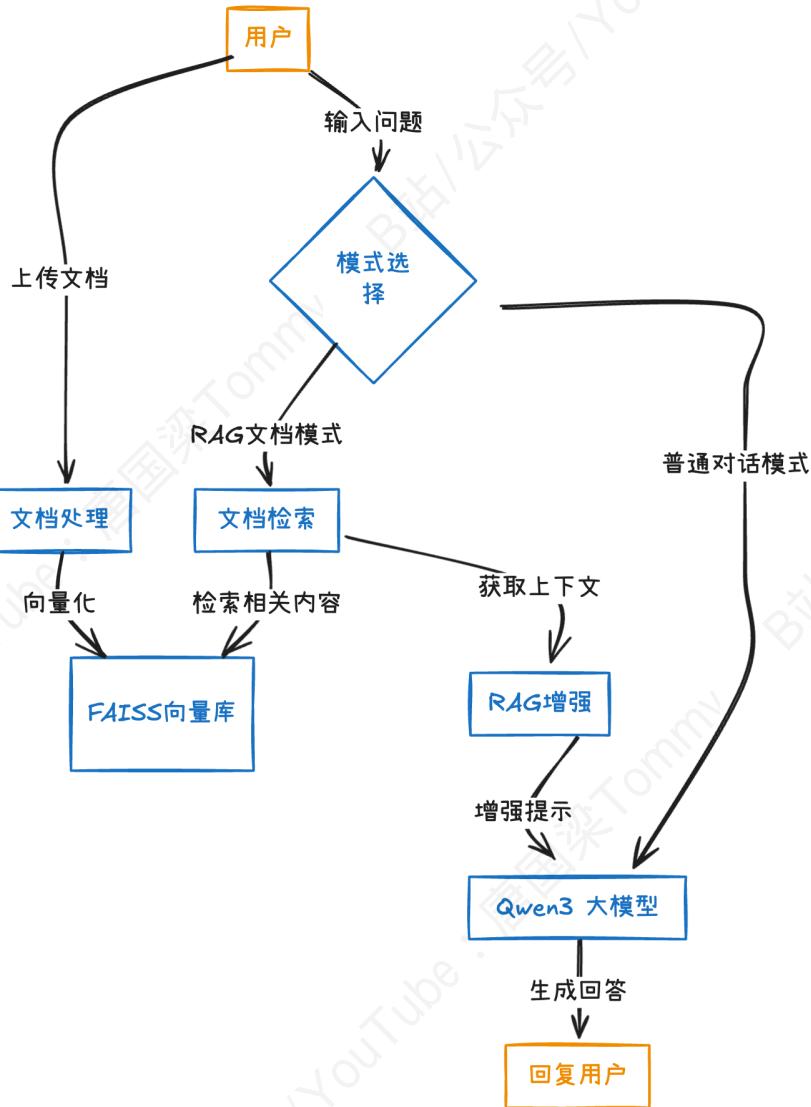
基于通义千问模型（Qwen3）的本地RAG（检索增强生成）智能问答系统，支持文档问答和天气查询功能。

一、系统特点

- 强大的问答能力：基于通义千问大模型，提供高质量的对话能力
- 本地RAG检索增强：上传文档后可针对文档内容进行智能问答
- 实时天气查询：集成高德地图API，支持查询全国城市天气
- 多模式对话：支持RAG文档问答模式和普通对话模式
- 灵活配置：可选择不同的模型版本和嵌入模型
- 友好的用户界面：基于Streamlit构建的简洁易用界面

二、技术架构

- 大模型引擎：基于Qwen3系列模型，支持本地部署
- 框架基础：基于agno框架构建，提供强大的代理能力
- 向量数据库：使用FAISS构建高效的向量检索系统
- 嵌入模型：支持多种嵌入模型，默认使用BGE-M3
- Web框架：基于Streamlit构建用户界面
- 工具能力：集成天气查询高德天气等工具功能，易于扩展
- 文档处理：支持多种格式文档的智能处理和分块



三、快速开始

1. 环境要求

- Python 3.12
- Ollama (用于本地部署大模型)
- NVIDIA GPU 24GB (推荐, 但非必需)
- FAISS

2. 虚拟环境配置

2.1 使用 uv 工具 (推荐)

[uv](#) 是一个快速的 Python 包管理器和虚拟环境工具，比传统的 pip 更高效。

1. 安装 uv:

```
# 使用官方安装脚本
curl -LsSf https://astral.sh/uv/install.sh | sh

# 或者通过 pip 安装
pip install uv
```

2. 创建虚拟环境:

```
# 创建虚拟环境
uv venv .venv

# 激活虚拟环境 (Linux/Mac)
source .venv/bin/activate

# 激活虚拟环境 (Windows)
.venv\Scripts\activate
```

3. 使用 uv 安装依赖:

```
# 从 requirements.txt 安装所有依赖
uv pip install -r requirements.txt
```

2.2 使用Ollama安装所需模型

[ollama安装指南](#)

```
# 安装Qwen3模型
ollama pull qwen3:8b
# 安装嵌入模型
ollama pull bge-m3:latest
```

2.3 启动应用

```
streamlit run app.py --server.port 6006
```

在浏览器中访问 <http://localhost:6006>

四、项目结构

```
qwen_agent_rag/
├── app.py                  # 主应用入口
├── chat_history.json       # 聊天历史记录
├── config/                 # 配置文件目录
│   └── settings.py         # 系统配置和常量
├── models/                 # 模型相关代码
│   └── agent.py            # RAG智能体实现
└── services/               # 核心服务
    └── vector_store.py     # 向量存储服务
```

```
|   └── weather_tools.py      # 天气查询工具
|   └── utils/
|       ├── chat_history.py    # 聊天历史管理
|       ├── decorators.py     # 装饰器工具
|       ├── document_processor.py # 文档处理器
|       └── ui_components.py   # UI组件
└── faiss_index/             # FAISS索引存储目录
```

五、主要功能

1. 文档问答 (RAG模式)

- 上传文档（支持PDF、TXT、DOCX等格式）
- 系统自动处理文档并构建向量索引
- 询问与文档相关的问题
- 系统检索相关内容并生成准确回答

2. 普通对话模式

- 切换至普通对话模式
- 直接与模型进行自由对话
- 享受大模型的通用能力

3. 天气查询功能

无论在哪种模式下，都可以查询全国各地的天气情况：

北京今天天气怎么样?
上海明天会下雨吗?

六、系统配置

在侧边栏中可以调整以下配置：

- 模型选择**: 可选择不同大小的Qwen3模型
- 嵌入模型**: 可选择不同的文本嵌入模型
- 相似度阈值**: 调整文档检索的相似度要求
- RAG模式开关**: 切换RAG文档问答模式和普通对话模式

七、使用提示

- 上传文档后，系统会自动处理并构建索引，请耐心等待
- 更改嵌入模型后，可能需要重新处理文档以更新索引
- 对于查询效果不佳的情况，可以尝试调整相似度阈值
- 天气查询功能需要网络连接以访问高德地图API

八、开发者参考

1. 主要组件

- **App类**: 主应用类, 管理整体流程和UI渲染
- **RAGAgent**: 封装大模型交互和工具调用
- **VectorStoreService**: 管理文档向量存储和检索
- **DocumentProcessor**: 处理和分块各种格式的文档
- **WeatherTools**: 提供天气查询功能
- **ChatHistoryManager**: 管理对话历史
- **UIComponents**: 提供UI渲染组件

2. 如何扩展

1. **添加新工具**: 参考 `weather_tools.py`, 实现新工具后在 `agent.py` 中注册
2. **支持新文档格式**: 在 `document_processor.py` 中添加新的文档加载器
3. **自定义嵌入模型**: 在 `settings.py` 中添加新的嵌入模型, 并确保Ollama中可用
4. **优化检索策略**: 可在 `vector_store.py` 中修改检索逻辑