

本地LLM学习助手

一个集成RAG知识检索和OBS录屏功能的学习辅助工具。仅用于学习练习，**不支持考试期间实时答题**。

功能特性

RAG学习助手

- **文档上传:** 支持PDF/Word/Excel/文本文件
- **智能分块:** 自动文本分块和向量化存储
- **题型识别:** 自动识别单选/多选/判断/主观题
- **本地LLM:** 集成Ollama，无需依赖云端服务

OBS录屏集成

- **远程控制:** 通过API控制OBS录屏开始/停止/暂停
- **场景管理:** 切换OBS场景，指定窗口录屏
- **文件管理:** 录屏文件列表、下载、删除
- **状态监控:** 实时监控录屏状态

学习合规检查

- **考试检测:** 自动识别并拒绝考试相关请求
- **安全限制:** 防止在考试环境中使用

快速开始

1. 安装和初始化

```
# 克隆或下载项目
cd llm-study-assistant

# 运行初始化脚本
chmod +x setup.sh scripts/*.sh
./setup.sh
```

2. 启动Ollama服务

```
./scripts/start_ollama.sh
```

3. 下载语言模型

```
./scripts/download_model.sh
```

4. 启动服务

```
./scripts/start_server.sh
```

5. 访问系统

- **API服务:** <http://localhost:8000>
- **API文档:** <http://localhost:8000/docs>
- **系统状态:** <http://localhost:8000/system/status>

目录结构

```
llm-study-assistant/
├── app/                                # 主应用程序
│   ├── main.py                        # FastAPI入口
│   ├── schemas.py                    # 数据模型
│   ├── llm.py                        # LLM客户端
│   ├── rag.py                        # RAG流程
│   ├── store.py                      # 向量存储
│   ├── parsers.py                    # 文档解析
│   └── prompts.py                    # 提示词模板
├── obs_integration/                  # OBS集成
├── data/                             # 数据存储
│   ├── uploads/                     # 上传文件
│   ├── index/                       # 向量索引
│   └── recordings/                   # 录屏文件
├── scripts/                          # 脚本工具
├── config/                           # 配置文件
├── requirements.txt                  # Python依赖
├── .env                             # 环境配置
└── README.md                        # 说明文档
```

配置说明

环境变量 (.env)

```
# LLM配置
OPENAI_API_BASE=http://localhost:11434/v1
OPENAI_API_KEY=ollama
LLM_MODEL=llama3.2:3b-instruct-q4_0

# 向量模型
EMBEDDING_MODEL=BAAI/bge-small-zh-v1.5

# OBS配置
OBS_HOST=localhost
OBS_PORT=4455
OBS_PASSWORD=

# 服务配置
API_HOST=0.0.0.0
API_PORT=8000
```

OBS Studio配置

1. 安装OBS Studio
2. 安装obs-websocket插件（或使用内置版本）
3. 启用WebSocket服务器（工具 > WebSocket服务器设置）



API使用示例

上传文档

```
curl -X POST "http://localhost:8000/upload" \  
  -H "Content-Type: multipart/form-data" \  
  -F "files=@document.pdf"
```

提问

```
curl -X POST "http://localhost:8000/ask" \  
  -H "Content-Type: application/json" \  
  -d '{  
    "type": "auto",  
    "question": "什么是机器学习？",  
    "top_k": 5  
  }'
```

开始录屏

```
curl -X POST "http://localhost:8000/obs/recording" \  
  -H "Content-Type: application/json" \  
  -d '{"action": "start"}'
```

停止录屏

```
curl -X POST "http://localhost:8000/obs/recording" \  
  -H "Content-Type: application/json" \  
  -d '{"action": "stop"}'
```



系统监控

检查系统状态

```
curl http://localhost:8000/system/status
```

查看知识库统计

```
curl http://localhost:8000/knowledge/stats
```

查看录屏文件

```
curl http://localhost:8000/recordings
```



故障排除

常见问题

1. Ollama连接失败

- 检查Ollama是否运行: `curl http://localhost:11434/api/tags`
- 重新启动: `./scripts/start_ollama.sh`

2. 模型加载失败

- 检查模型是否存在: `ollama list`
- 重新下载: `./scripts/download_model.sh`

3. OBS连接失败

- 确保OBS Studio运行中
- 检查WebSocket设置（端口、密码）

4. 文档解析失败

- 检查文件格式是否支持
- 安装数缺失的Python库

日志查看

- Ollama日志: `logs/ollama.log`
- 服务日志: 控制台输出

Web界面开发

您可以基于RESTful API开发Web前端界面:

- 使用React/Vue.js等前端框架
- 集成文件上传组件
- 实现问答界面
- 录屏控制面板

重要声明

本系统仅用于学习和练习目的。严格禁止在以下场景中使用:

- **X** 考试期间实时答题
- **X** 任何形式的学术不端
- **X** 考试作弊行为

系统内置合规检查机制,会自动识别和拒绝相关请求。

相关链接

- [Ollama](#) - 本地LLM运行环境
- [OBS Studio](#) - 开源录屏软件
- [FastAPI](#) - Python Web框架



许可证

MIT License - 仅供学习使用

如有问题或建议，请通过[GitHub Issues](#)或相关渠道反馈。