

银河麒麟智能问答助手 v2.5

银河麒麟智能问答助手 v2.5

基于DeepSeek-R1 API和麒麟SDK2.5的智能问答系统，专为银河麒麟操作系统设计。

🌟 功能特性

- 智能问答:** 基于RAG(检索增强生成)技术，结合本地文档知识库和AI大模型
- 文档处理:** 支持PDF、Markdown、文本等多种格式文档
- 系统集成:** 深度集成麒麟SDK2.5，获取系统信息和硬件状态
- 图形界面:** 友好的GUI界面，支持文档管理和实时问答
- 多架构支持:** 支持x86_64和ARM64架构

📋 系统要求

推荐环境

- 操作系统:** 银河麒麟 V10 SP1/SP2/SP3 或更高版本
- Python版本:** Python 3.7+ (推荐Python 3.8+)
- 内存:** 4GB以上 (推荐8GB)
- 存储:** 2GB可用空间
- 网络:** 需要互联网连接访问DeepSeek API

必需依赖

- 麒麟SDK2.5开发库
- DeepSeek API访问权限
- Python第三方库 (见requirements.txt)

🚀 快速安装指南

方法一：一键安装（推荐）

1. 下载项目

```
# 使用git克隆（推荐）
git clone https://github.com/your-username/kylin-qa-assistant.git
cd kylin-qa-assistant

# 或者下载ZIP包并解压
wget https://github.com/your-username/kylin-qa-assistant/archive/main.zip
```






```
unzip main.zip
cd kylin-qa-assistant-main
```

2. 运行安装脚本

```
# 给安装脚本执行权限
chmod +x install.sh

# 运行安装脚本（需要sudo权限）
sudo ./install.sh
```

安装脚本将自动完成以下操作：

-  检查系统兼容性
-  安装Python依赖包
-  配置系统服务
-  创建桌面快捷方式
-  设置命令行快捷方式

3. 配置API密钥

```
# 编辑配置文件
sudo nano /opt/kylin-qa-assistant/config.py

# 修改以下行，填入您的DeepSeek API密钥
DEEPSEEK_API_KEY = "your-actual-api-key-here"
```

4. 启动程序

```
# 方法1：使用桌面快捷方式
# 双击桌面上的“银河麒麟智能问答助手”图标

# 方法2：使用命令行
kylin-qa

# 方法3：启动系统服务
sudo systemctl start kylin-qa
```

方法二：手动安装

1. 环境准备

```
# 更新系统包
sudo apt update
sudo apt upgrade -y

# 安装Python和pip（如果未安装）
```

```
sudo apt install python3 python3-pip python3-venv -y
```

```
# 安装系统依赖
```

```
sudo apt install python3-tk python3-dev build-essential -y
```

```
# 安装麒麟SDK开发包（如果可用）
```

```
sudo apt install libkysdk-dev -y
```

2. 下载项目

```
git clone https://github.com/your-username/kylin-qa-assistant.git
```

```
cd kylin-qa-assistant
```

3. 创建虚拟环境（推荐）

```
# 创建虚拟环境
```

```
python3 -m venv venv
```

```
# 激活虚拟环境
```

```
source venv/bin/activate
```

4. 安装Python依赖

```
# 升级pip
```

```
pip install --upgrade pip
```

```
# 安装项目依赖
```

```
pip install -r requirements.txt
```

5. 配置程序

```
# 复制配置文件
```

```
cp config.py config.py.bak
```

```
# 编辑配置文件
```

```
nano config.py
```

```
# 修改API密钥
```

```
# DEEPSEEK_API_KEY = "your-actual-api-key-here"
```

6. 创建必要目录

```
mkdir -p data docs logs assets
```

7. 运行程序

```
# 启动图形界面
```

```
python3 main.py
```

```
# 或使用启动脚本
./run.sh
```

获取DeepSeek API密钥

1. 访问 [DeepSeek官网](#)
2. 注册账号并登录
3. 进入API管理页面
4. 创建新的API密钥
5. 复制密钥并填入配置文件

注意: 请妥善保管您的API密钥，不要泄露给他人。

使用指南

首次使用

1. 启动程序

```
# 命令行启动
kylin-qa

# 或双击桌面图标
```

2. 添加文档

- 点击"添加文档"按钮
- 选择您要添加的文档文件
- 支持格式：PDF、Markdown、TXT、Word等
- 等待文档处理完成

3. 开始问答

- 在问题输入框中输入您的问题
- 可选择是否包含系统信息
- 点击"提问"或按回车键
- 查看AI生成的回答

常用功能

文档管理

- **添加文档:** 支持批量添加多个文档
- **查看状态:** 实时显示知识库中的文档数量
- **清空知识库:** 一键清空所有文档（谨慎操作）

智能问答

- **普通问答:** 基于文档内容回答问题
- **系统问答:** 结合当前系统信息回答
- **相关文档:** 显示回答所参考的文档来源

系统信息

- **查看系统状态:** 点击"系统信息"按钮
- **硬件信息:** CPU、内存、磁盘等信息
- **网络状态:** 网络配置和连接状态
- **进程信息:** 系统负载和进程统计

高级配置

配置文件说明

主要配置项位于 `config.py` 文件中：

```
# API配置
DEEPSEEK_API_KEY = "your-api-key" # 必须配置
DEEPSEEK_API_ENDPOINT = "https://api.deepseek.com/v1/chat/completions"

# 文档路径
DOCUMENT_PATH = "./docs" # 文档存储路径
VECTOR_DB_PATH = "./data/vector_db" # 向量数据库路径

# RAG配置
RAG_CONFIG = {
    "top_k": 5, # 检索文档数量
    "similarity_threshold": 0.7, # 相似度阈值
    "max_context_length": 2000, # 最大上下文长度
    "temperature": 0.7, # AI回答的随机性
    "max_tokens": 1000 # 最大回答长度
}
```

性能优化

```
# 性能配置
PERFORMANCE_CONFIG = {
    "max_concurrent_processes": 4, # 最大并发进程数
    "cache_size": 100, # 缓存大小
    "batch_size": 32 # 批处理大小
}
```

系统服务管理

```
# 查看服务状态
sudo systemctl status kylin-qa

# 启动服务
sudo systemctl start kylin-qa

# 停止服务
sudo systemctl stop kylin-qa

# 重启服务
sudo systemctl restart kylin-qa

# 开机自启
sudo systemctl enable kylin-qa

# 禁用自启
sudo systemctl disable kylin-qa

# 查看服务日志
journalctl -u kylin-qa -f
```

项目结构

```
kylin-qa-assistant/
├── main.py           # 主程序入口
├── config.py         # 配置文件
├── requirements.txt  # Python依赖
├── README.md        # 项目说明
├── run.sh            # Linux启动脚本
├── install.sh        # 安装脚本
├── uninstall.sh      # 卸载脚本
├── kylin-qa.service  # 系统服务配置
├── src/              # 源代码目录
│   ├── ai_models.py # AI模型接口
│   ├── vector_store.py # 向量存储
│   ├── rag_engine.py # RAG引擎
│   ├── gui.py        # 图形界面
│   ├── document_processor.py # 文档处理
│   └── system_info_helper.py # 系统信息
├── data/             # 数据目录
│   └── vector_db     # 向量数据库
├── docs/             # 文档目录
└── logs/             # 日志目录
```

支持的文档格式

格式	扩展名	说明
PDF文档	.pdf	支持文本提取和OCR
Markdown	.md	完整的Markdown语法支持
文本文档	.txt	纯文本文件
reStructuredText	.rst	技术文档格式
Word文档	.doc, .docx	Microsoft Word文档
HTML文档	.html, .htm	网页文档

故障排除

常见问题及解决方案

1. 程序无法启动

问题: 运行程序时出现错误

解决方案:

```
# 检查Python版本
python3 --version

# 检查依赖是否安装完整
pip3 list | grep -E "numpy|scikit-learn|requests"

# 重新安装依赖
pip3 install -r requirements.txt --force-reinstall

# 查看详细错误信息
python3 main.py --debug
```

2. API调用失败

问题: DeepSeek API调用失败

解决方案:

```
# 检查网络连接
ping api.deepseek.com
```

```
# 验证API密钥
curl -H "Authorization: Bearer your-api-key" \
  https://api.deepseek.com/v1/models
```

```
# 检查API额度
# 登录DeepSeek控制台查看使用情况
```

3. 麒麟SDK库加载失败

问题: 系统信息功能不可用

解决方案:

```
# 检查SDK库是否存在
ls -la /usr/lib/*/libkysysinfo.so

# 安装麒麟SDK开发包
sudo apt install libkysdk-dev

# 检查库文件权限
sudo chmod 755 /usr/lib/*/libkysysinfo.so
```

4. 文档处理失败

问题: 无法处理某些文档格式

解决方案:

```
# 安装额外的文档处理库
pip3 install pdfplumber python-docx

# 检查文件权限
ls -la your-document.pdf

# 尝试转换文档格式
# PDF -> TXT: pdftotext document.pdf
```

5. GUI界面问题

问题: 图形界面无法显示

解决方案:

```
# 检查显示环境
echo $DISPLAY

# 安装tkinter
```



```
sudo apt install python3-tk
```

```
# 使用X11转发（SSH连接时）
```

```
ssh -X username@hostname
```

日志查看

```
# 查看应用日志
```

```
tail -f logs/app.log
```

```
# 查看系统服务日志
```

```
journalctl -u kylin-qa -f
```

```
# 查看详细调试信息
```

```
python3 main.py --debug
```

性能优化建议

1. 内存优化

- 适当调整 `PERFORMANCE_CONFIG` 中的参数
- 定期清理向量数据库
- 限制同时处理的文档数量

2. 网络优化

- 使用国内API镜像（如果可用）
- 调整API请求超时时间
- 启用请求缓存

3. 存储优化

- 定期清理日志文件
- 压缩大型文档
- 使用SSD存储提升性能

更新升级

检查更新

```
# 检查远程更新
```

```
git fetch origin
```

```
git status
```

```
# 查看更新内容
```

```
git log HEAD..origin/main --oneline
```

升级程序

```
# 备份配置
cp config.py config.py.backup

# 拉取最新代码
git pull origin main

# 更新依赖
pip3 install -r requirements.txt --upgrade

# 重启服务
sudo systemctl restart kylin-qa
```

版本回退

```
# 查看版本历史
git log --oneline

# 回退到指定版本
git checkout <commit-hash>

# 恢复配置
cp config.py.backup config.py
```

卸载程序

```
# 运行卸载脚本
sudo ./uninstall.sh

# 手动清理（如果需要）
sudo rm -rf /opt/kylin-qa-assistant
sudo rm -f /etc/systemd/system/kylin-qa.service
sudo rm -f /usr/local/bin/kylin-qa
rm -f ~/Desktop/kylin-qa.desktop
```

许可证

本项目采用 MIT 许可证 - 查看 [LICENSE](#) 文件了解详情。

贡献

欢迎提交Issue和Pull Request来改进项目！

贡献指南

1. Fork 项目
2. 创建特性分支 (`git checkout -b feature/AmazingFeature`)

3. 提交更改 (`git commit -m 'Add some AmazingFeature'`)
4. 推送到分支 (`git push origin feature/AmazingFeature`)
5. 开启 Pull Request

开发环境搭建

```
# 克隆开发版本
git clone https://github.com/your-username/kylin-qa-assistant.git
cd kylin-qa-assistant

# 创建开发环境
python3 -m venv dev-env
source dev-env/bin/activate

# 安装开发依赖
pip install -r requirements.txt
pip install pytest black flake8 # 开发工具

# 运行测试
pytest tests/

# 代码格式化
black src/

# 代码检查
flake8 src/
```

支持与反馈

如果您在使用过程中遇到问题，请：

1. **查看文档**: 首先查看本README和相关文档
2. **搜索问题**: 在 [Issues](#) 页面搜索类似问题
3. **提交Issue**: 如果没有找到解决方案，请提交新的Issue
4. **联系开发者**: 通过邮件或其他方式联系项目维护者

Issue提交模板

提交Issue时，请包含以下信息：

****环境信息****

- 操作系统：银河麒麟 V10 SP3
- Python版本：3.8.10
- 程序版本：v2.5.0

问题描述

简要描述遇到的问题

复现步骤

1. 执行了什么操作
2. 期望的结果
3. 实际的结果

错误信息

粘贴完整的错误信息或日志

截图

如果有GUI相关问题，请提供截图

更新日志

v2.5.0 (当前版本)

- ✨ 集成DeepSeek-R1 API
- ✨ 支持麒麟SDK2.5
- ✨ 优化RAG检索算法
- ✨ 改进GUI界面体验
- ✨ 增强系统兼容性
- ✨ 添加一键安装脚本
- ✨ 完善文档和使用指南
- 🐛 修复文档处理bug
- 🐛 优化内存使用
- 🐛 改进错误处理

未来计划

- 🚀 支持更多AI模型
- 🚀 添加语音交互功能
- 🚀 支持多语言界面
- 🚀 增加插件系统
- 🚀 优化性能和稳定性

银河麒麟智能问答助手 - 让AI助力您的麒麟系统使用体验！

🌟 如果这个项目对您有帮助，请给我们一个Star！