

研究助理Agent - 本地浏览器搜索版

本项目是一个多代理研究助手，它使用OpenAI的Agents SDK和本地浏览器进行网络搜索，收集信息并生成研究报告。本项目为使用Ollama等本地模型设计，不需要OpenAI API密钥。

1. 功能特点

- 使用本地浏览器（通过Playwright）进行网络搜索，不依赖API
- 与本地模型（如Ollama）完全兼容
- 多代理协作架构（规划、搜索、撰写）
- 自动规划搜索策略
- 并行执行多个搜索任务
- 使用BeautifulSoup提取和分析网页内容
- 生成结构化研究报告
- 健壮的错误处理和重试机制

2. 安装要求

1. 首先，确保你已安装uv。如果没有安装，可以通过以下命令安装

```
curl -LsSf https://astral.sh/uv/install.sh | sh
```

2. 创建虚拟环境（本项目已经创建，直接跳过）

```
uv venv
```

3. 激活虚拟环境

```
# 在macOS/Linux上
```

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

```
# 在Windows上
```

```
.venv\Scripts\activate
```

4. 使用uv安装依赖项

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

5. 安装Playwright

```
playwright install
```

6. 确保有一个运行中的Ollama服务，并且已加载了所需模型

```
ollama pull qwq:latest
```

3. 使用方法

运行主程序：

```
cd openai-agents-python/
```

```
python -m examples.research_bot_ollama.main
```

程序会提示你输入研究主题，然后自动执行完整研究流程：

- 规划搜索策略
- 打开本地浏览器执行搜索

3. 收集和整理信息
4. 生成研究报告

4. 核心架构与运行逻辑

4.1 三种智能代理

项目使用三种不同的代理协同工作:

1. 规划代理 (**planner_agent**)
 - 分析用户的研究主题
 - 生成搜索查询词和相应的理由
 - 输出类型为 `WebSearchPlan`
2. 搜索代理 (**search_agent**)
 - 使用本地浏览器执行网络搜索
 - 分析并摘取搜索结果
 - 为每个查询生成简洁摘要
3. 撰写代理 (**writer_agent**)
 - 整合所有搜索结果
 - 生成结构化研究报告
 - 输出类型为 `ReportData`, 包含摘要、详细报告和后续问题

4.2 运行流程

1. 初始化阶段
 - 用户输入研究主题
 - 设置与本地Ollama模型的连接
2. 规划阶段
 - 规划代理分析研究主题
 - 生成多个搜索查询词及理由
 - 为提高稳定性, 最多选取前5条搜索
3. 搜索阶段
 - 并行执行多个搜索任务
 - 每个搜索使用本地浏览器:
 - 启动Playwright浏览器
 - 访问搜索引擎
 - 输入搜索词并获取结果
 - 解析和提取搜索结果
 - 设置超时机制避免单个搜索任务卡住
4. 报告生成阶段
 - 撰写代理接收所有搜索结果
 - 整合信息生成结构化报告
 - 包含三个部分: 简短摘要、详细报告和后续问题
 - 检测报告是否完整, 必要时重试

- 将报告保存为markdown和txt两种格式

5. 技术实现

本项目使用以下技术:

- **OpenAI Agents SDK**: 用于创建和管理智能代理
- **Playwright**: 用于自动化控制浏览器
- **BeautifulSoup**: 用于HTML内容解析和提取
- **AsyncComputer**: 实现浏览器控制接口
- **Function Tools**: 实现自定义函数工具用于搜索
- **Pydantic**: 数据类型定义和验证
- **Rich**: 提供丰富的终端输出和进度显示
- **Asyncio**: 实现异步操作和并行处理

6. 文件结构

- `main.py`: 主程序入口
- `manager.py`: 研究管理器, 协调整个研究流程
- `agents/`: 包含各种代理实现
 - `planner_agent.py`: 规划搜索策略
 - `search_agent.py`: 执行网络搜索
 - `writer_agent.py`: 生成研究报告
 - `browser_computer.py`: 本地浏览器实现
 - `browser_search.py`: 浏览器搜索函数工具
- `output_report/`: 保存生成的报告
- `printer.py`: 用于显示进度和状态的终端输出工具

AI进阶学习

 **B站课堂**: <https://space.bilibili.com/3546748815411393/pugv>

 **官方网站**: <https://www.tgltommy.com/> (国内访问需科学上网)

如果你正在关注大模型 Agent、强化学习后训练、RLHF、DPO、GRPO、RLVR 等前沿方向, 欢迎关注我最新上线的精品课程:
[《大模型 Agent 强化学习实战: 从后训练、奖励设计到工业级对齐系统》](#)

这门课程围绕当前大模型 Agent 强化学习的核心技术路线展开, 不只是介绍零散算法和论文概念, 而是希望帮助学习者建立一套完整的技术地图: 从基础原理、论文脉络、关键算法, 到开源项目源码解析与工业级系统实践, 逐步理解 Agent RL 为什么重要、如何演进, 以及在真实应用中如何落地。

课程配套专属飞书知识库《Agent RL 学习宝典》, 包含学习路线、论文资料、项目说明、源码阅读辅助资料和后续持续更新内容, 方便长期查阅和跟进前沿技术。

Agent RL 变化很快, 但越是变化快, 越需要一套系统的学习框架。如果你希望系统进入 Agent RL 方向, 而不是停留在碎片化了解阶段, 这门课会是一个很好的起点。

课程官网: <https://www.tgltommy.com/p/agent-rl> (国内访问需科学上网)

B站课堂: <https://www.bilibili.com/cheese/play/ss842375604>

更多课程详情, 扫描以下二维码咨询课程助理:

新课首发

大模型 **Agent**
强化学习实战:

从后训练、奖励设计到工业级对齐系统

唐国梁Tommy

TGL AI
Tommy

官方微信课程助理



B站/公众号/YouTube: 唐国梁Tommy

**关注微信公众号
获取更多学习资源**



官方网站: TGLTommy.com(需科学上网)