

实践实验4：软件集成与改进

1. 实验目的

1. 掌握复杂应用程序的开发方法
2. 掌握故障诊断的方法
3. 掌握将UI设计、软件模型结合的能力

2. 实验任务

1. 结合LLM、本地数据（招聘数据集）完成基于自然语言交互和可视化图像的就业咨询的完整APP
2. 包含左侧栏（内容自定义，可以是key、参数、约束条件等等）
3. 包含聊天界面（内容包括图文等，根据能力完成，至少必须完成文字交互）
4. 至少包含3个问题（含图文）的交互展示，可以更多（加分）
5. 加分项：部署在服务器上，使用手机打开APP操作
6. 加分项：如果能力可及，可以采用语音输入的方式获取用户提问。
7. 本次没有固定的实验报告格式，请同学们内容格式自拟（.md），包括界面介绍、实现的功能截图。

3. 注意事项

3.1. 尽量使用数据驱动，而非逻辑驱动

随着项目进展，软件结构越来越复杂。学生在能力可及的范围内，可以考虑使用数据驱动。

许多的回调操作基础流程是差不多的，虽然使用if-elif-elif....可以将这些操作区分开，但是更好的方式是将逻辑差不多的操作整合成一个统一的处理函数，通过输入列表来实现调用。如果缺乏编程经验，还是需要先完成功能，再考虑优化。

3.2. 学会管理复杂度

在本次实验中，使用streamlit来启动程序，这个方式的问题在于无法使用IDE的跟踪工具去跟踪故障。对于复杂的应用开发，要学会管理复杂度。

例如编写一个test.py程序，对某个问题或函数进行单独测试。例如下面的代码test.py中，test_asso这些开关打开（设置为True）时就可以针对某个特定函数做单元测试。当测试成功后，再挪到app.py中调用。

```
cacu_aprior = False
test_asso = False
test_func = False
chat_test = False
print(dataset.info())

if cacu_aprior:
    apriorifile = "apriori.bin"
    skill_list = getSkillList(dataset, 'skill_list')
    big, biga = calcu_apriori(skill_list, 0.01, savefile=apriorifile)
    s = getAsso(biga, 'linux', 0.1)
    print(s)
```

```

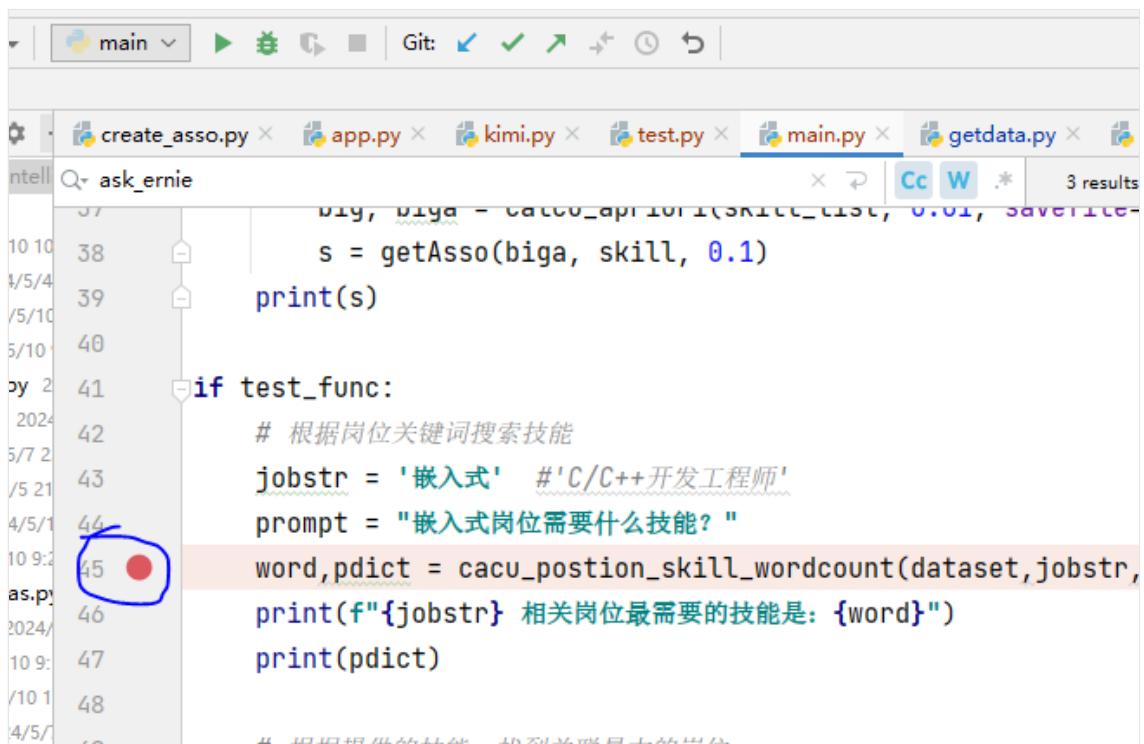
if test_func:
    # 根据岗位关键词搜索技能
    jobstr = '嵌入式' # 'C/C++开发工程师'
    prompt = "嵌入式岗位需要什么技能？"
    word,pdict = cacu_postion_skill_wordcount(dataset,jobstr,prompt,count = 20)
    print(f"{jobstr} 相关岗位最需要的技能是: {word}")
    print(pdict)

```

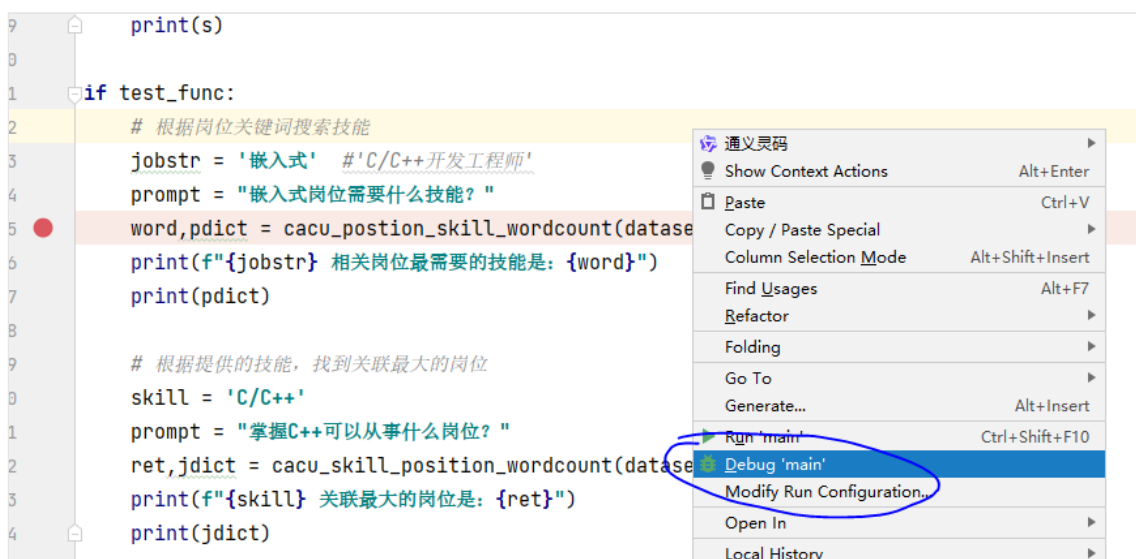
3.3. 单步跟踪

以pycharm为例:

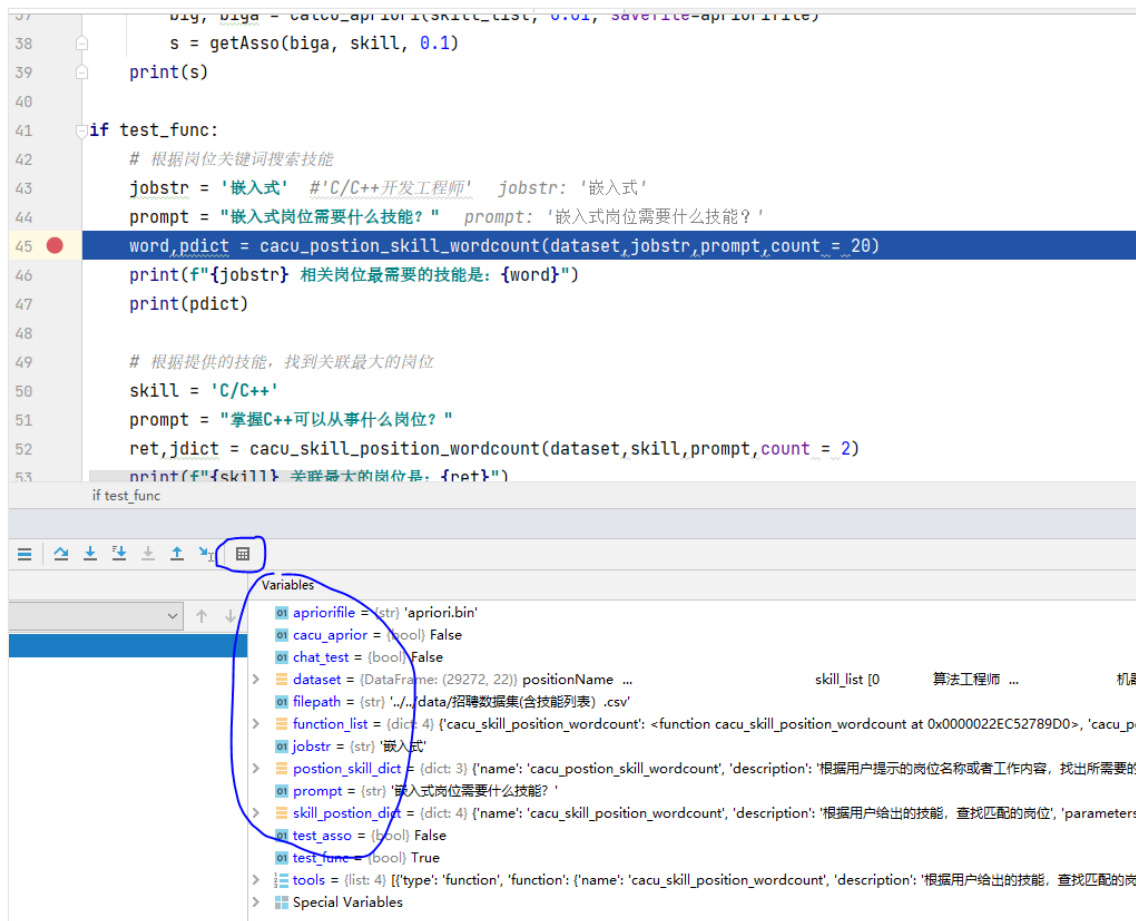
1. 点击要设置断点的代码左侧，出现一个红点，如果要取消，点击红点就可以了:



2. 点击右键选择debug....



3. 开始跟踪，程序运行到断点处会停止。此时可以观察下方的窗口，可以看到各个变量的值。也可以点击评估器（蓝色圈），输入表达式查看结果。



4. 如果选择进入函数跟踪, 使用F7, 如果不进入函数, 按F8, 如果要去下一个断点, 按F9

3.4. 必须掌握辅助编程插件

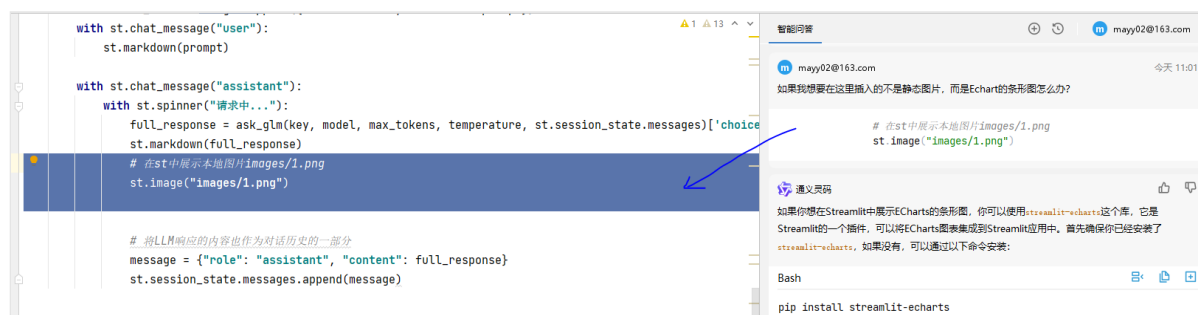
强烈建议使用辅助编程插件而不是网页版的大模型对话。

推荐的插件: 通义千问、copilot等, 除了生成代码, 还可以就上下文情况做提问。可以帮助快速找到bug。

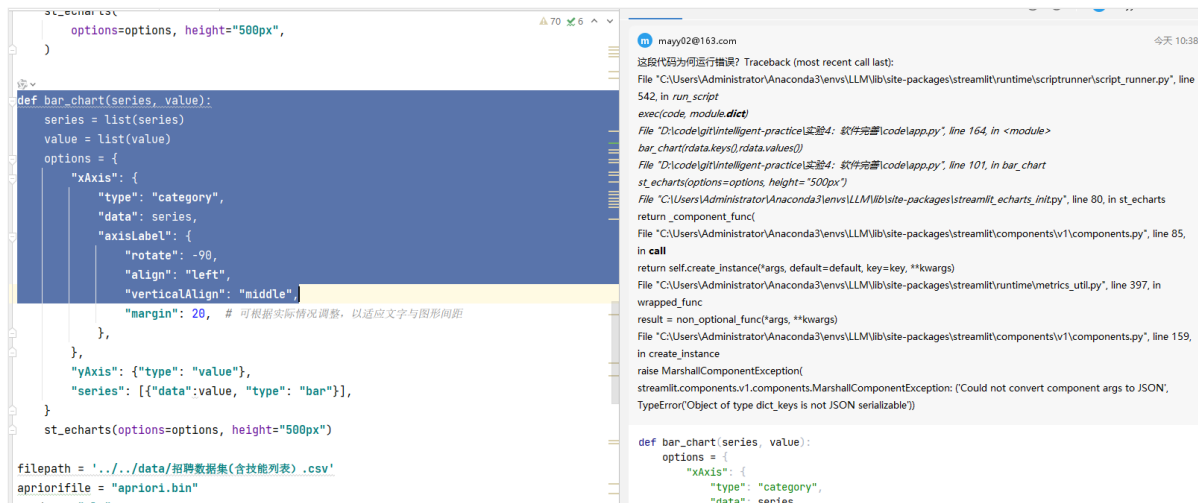
只需要给出批注, 自动补充:



可以针对某个上下文位置提问:



可以针对运行错误提问:



4. 参考技术

4.1. 构建聊天界面

参考代码chat_glm.py（先运行这个例子测试一下，注意运行时请输入glm的apikey）

1. 左侧栏

```
# 这个部分是左边栏目的内容，之前实验1做过，demo中的内容可以替换为之前的内容
with st.sidebar:
    略
```

2. 记录对话历史信息：

`st.session_state` 是一种记录在st会话中的变量的方法。在st中的变量都是以字典形式出现的。以变量名：值的形式给出。例如下面这段话的意思是，在session中添加一个叫做message的变量。

```
# 初始化的对话。如果会话中没有message这个变量，就添加一个。这里添加的就是模拟LLM机器人的口气开启的第一声招呼。
if "messages" not in st.session_state.keys():
    st.session_state.messages = [{"role": "assistant", "content": "你好我是ChatGLM，有什么可以帮助你？"}]
```

随着会话的进展，消息每一次都被添加到 `st.session_state.messages` 中，因此这里，当界面上的插件有变动的时候，都会将历史对话在界面上重写一次。也就是说，时刻刷新历史对话信息到界面上。这里 `st.chat_message` 根据角色呈现聊天机器人的头像。st.write是将聊天文本写到界面上。

在 `with st.chat_message(message["role"]):` 下面段落的内容都会绘制在聊天对话中。

```
for message in st.session_state.messages:
    with st.chat_message(message["role"]):
        st.write(message["content"])
```

这段代码效果如下：



你好我是ChatGLM，有什么可以帮助你？



请问什么是AI agent?请简短回答

这里还提供了一个清空历史记录的功能。随着时间的增加，对话记录会越来越多。如果你选择每次都考虑历史对话信息，发往LLM的对话token消耗速度会越来越快。（例如用户在前文透露了一些个人信息，那么后面的对话就可以根据前面的历史信息给出用户更加偏爱的回答。这种就是带着历史记录的对话。如果不希望带历史记录，就比如之前实验做的那样，那就不需要考虑历史信息放入chat）

这里将清空历史记录放在左侧栏中（st.sidebar）

```
def clear_chat_history():
    st.session_state.messages = [{"role": "assistant", "content": "你好我是
    ChatGLM，有什么可以帮助你？"}]

st.sidebar.button('清空聊天记录', on_click=clear_chat_history)
```

通过API与大模型的对话



API Token已经配置

输入Token:

选择模型

glm-3-turbo ▼

max_tokens

512

0 2000

temperature

0.80

0.00 2.00

清空聊天记录

2. 根据用户输入的对话，调用LLM获取返回信息。将返回信息添加到message中。其中 `st.chat_input()` 对应下面的输入框：

Your message

>

当用户输入了key之后，才可以显示对话信息。

```
if len(key) > 1:
```

```
# 界面上绘制输入框，并且将用户输入的内容添加到对话列表中
```

```
if prompt := st.chat_input():
```

```
# 将用户输入的文字（提示词）添加到message中。
```

```

st.session_state.messages.append({"role": "user", "content": prompt})
with st.chat_message("user"):
    # 将用户输入的提示词以markdown格式打印出来。
    st.markdown(prompt)

with st.chat_message("assistant"):
    with st.spinner("请求中..."):
        #调用大模型，将历史对话（st.session_state.messages）送给大模型，返回值放入full_response中。
        full_response = ask_glm(key, model, max_tokens, temperature,
st.session_state.messages)['choices'][0]['message']['content']
        # 将返回值打印出来。注意，这里还在st.chat_message("assistant")的应答范围内。

        st.markdown(full_response)

        # 将LLM响应的内容也作为对话历史的一部分
        message = {"role": "assistant", "content": full_response}
        st.session_state.messages.append(message)

```

4.2. 在对话中调用LLM

在当前的应用中，通过获得用户的提问，调用LLM转为对本地函数的调用，函数调用的结果再调用LLM转为自然语言回复给大模型。

具体流程如下：

1. 用户输入信息，例如“请问北京薪资最高的岗位是什么？”（提示词）
2. 将提示词信息连同定义了回调函数的tools字段的信息发送给LLM，由LLM判断应该调用哪个函数。例如应该调用的函数是get_max_salary_job()，假设返回的是“算法工程师”。
3. 重新构造提示词（这里只是举例，具体的自定义）：用户的问题是“请问北京薪资最高的岗位是什么？”，根据2023年拉勾网数据查询结果是“算法工程师”，请将查询结果转为对用户的回答。
4. LLM返回润色后的自然语言，例如：根据2023年拉勾网的查询结果，北京薪资最高的岗位是算法工程师。
5. 将LLM返回的自然语言作为full_response返回（上面代码中ask_glm的返回，注意这里只是例子，大家可以根据自己的代码情况设计。）

4.3. 在对话中绘制Echart图像

为了更有说服力，需要将本地查询结果呈现出来。为了更好地呈现查询结果，可以在执行本地查询的时候生成图片。

有两种实现方式：

1. 在回调函数中生成图片文件，在应答的时候插入图片。（这个方法简单，如何编写自己查询）
 2. 在回调函数中增加数据返回，在界面上调用Echart绘图（其实也可以直接在回调函数中直接渲染）
- 效果如下：



掌握C++可以从事什么岗位?



■ C/C++开发工程师 ■ 算法工程师



掌握C++可以从事C/C++开发工程师和算法工程师等岗位。作为C/C++开发工程师，您将负责使用C++编写高效、稳定的代码，为软件产品提供坚实的基础。作为算法工程师，您将运用C++实现复杂的算法，为人工智能、机器学习等领域提供技术支持。为了更好地从事这些岗位，建议您深入学习C++语言特性和标准库，同时掌握常用的数据结构和算法，不断提升自己的编程能力和解决问题的能力。{'C/C++开发工程师': 2485, '算法工程师': 1896}

Your message





我C++学得不错，请推荐相关的技术学习



c linux 编程 arm 通信 spi 单片机 i2c uart 操作系统
算法 设备 freertos



查询结果显示，C++语言与多个技术领域紧密相关，尤其是Linux操作系统、SPI和I2C通信协议、编程、ARM架构单片机开发等。这些岗位通常涉及硬件设备编程，操作系统开发，以及嵌入式系统中的通信和算法实现。为了更好地适应这些岗位，建议深入学习C++，掌握Linux系统编程，熟悉SPI、I2C、UART等通信协议，并了解FreeRTOS等嵌入式操作系统。同时，加强算法和数据结构的学习，以提升解决问题的能力。

通过API与大模型的对话

API Token已经配置

输入Token:

选择模型

glm

选择可视化图像类型

条形图

清空聊天记录

我是就业小助理，有什么可以帮助你？

嵌入式岗位需要什么技能？



技能	频率
C	285
编程	280
C++	270
Linux	230
通信	210
arm	210
单片机	210
spi	190
i2c	145
uart	140
mcu	115
电子	105
计算机	105
设备	105
通讯	105

嵌入式岗位要求具备以下核心技能：首先是熟练掌握C和C++编程语言，这是基础中的基础。其次，对Linux系统的深入了解和操作能力也非常重要。此外，对于通信协议如SPI、I2C、UART的掌握，以及对ARM架构和单片机（MCU）的了解同样关键。电子和计算机知识也是必备的。在学习上，建议重点提升这些技能，并通过实际项目来加深理解与应用。

例如将Echart的生成结果放入对话中。

4.4. 表格呈现

对于一些问题的结果，也可以考虑使用表格做呈现：（使用 `st.dataframe(rdata)` ,其中rdata是过滤后的dataframe，注意控制表格长度）

有没有java相关的招聘岗位给我推荐一些？



	positionName	companyFullName	city	salary	workYear	education
1	算法工程师	北京字节跳动网络技术有限公司	北京	20k-40k	3-5年	本科
4	算法工程师	联通数字科技有限公司	北京	20k-30k	3-5年	本科
5	算法工程师	西安亚购信息科技有限公司	西安	20k-35k	3-5年	本科
6	算法工程师	创新奇智（北京）科技有限公司	上海	15k-25k	1-3年	本科
8	算法工程师	杭州星迈科技有限公司	杭州	18k-36k	1-3年	硕士
9	算法工程师	创新奇智（北京）科技有限公司	上海	18k-30k	1-3年	本科
11	算法工程师	达而观信息科技（上海）有限公司	上海	16k-25k	不限	本科
26	C/C++开发工程师	联通数字科技有限公司	北京	15k-27k	3-5年	本科
28	C/C++开发工程师	深圳软牛科技集团股份有限公司	深圳	12k-20k	1-3年	本科
35	Java工程师	五八同城信息技术有限公司	北京	25k-45k	不限	本科

Your message >

5. 注意事项

本次实验无需像例子中那样每次聊天都携带历史对话。学生可以选择携带历史记录，也可以不带。考虑到token的消耗量较大，调试多了会涉及费用问题，本次实验推荐使用后者（不带历史对话），且这种方式实现与之前的实验一致，相对简单。

开发过程中一定要注意将前期的功能测试完整，才能继续做后面的实验，不然到了集成阶段会陷入越来越混乱的场面。