

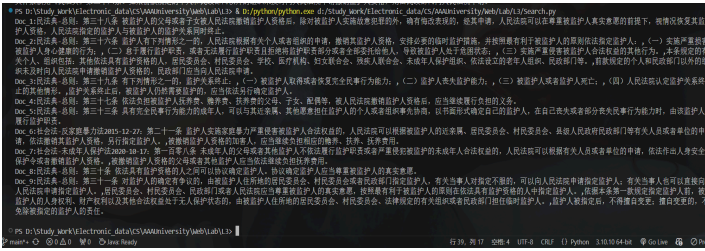
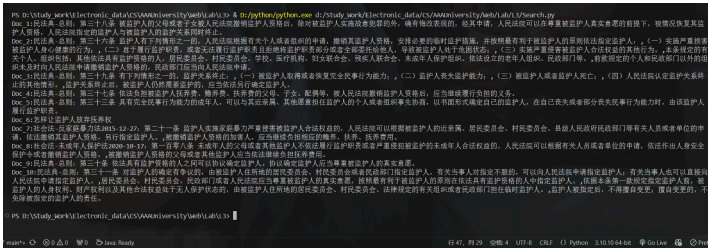
Task 2

基于任务一中入库后的数据，实现了一个简单的检索方法。

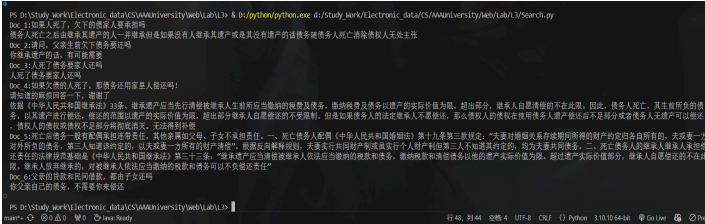
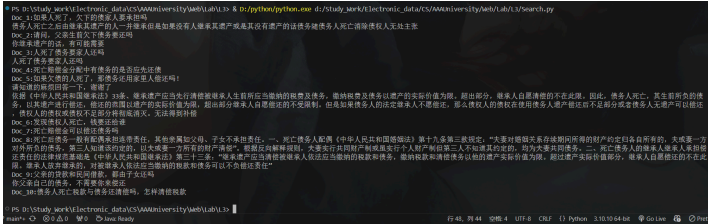
如果直接调用 m3e-base 模型进行相似性搜索，得到的效果并不太好，这里通过观察数据的基本格式，发现响应中可能包含 问题-回答、 回答、 问题 三种情况，基于此新增两条筛选规则试图筛选掉仅包含 问题 的无效信息：

- 总字数不少于 35
- 响应中包含换行符

两条规则满足其一即可。



左边是筛选前，右边是筛选后。这里可能还不太明显，让我们来看看另一个问题



可以看到成功过滤掉了几条无用提问

```

def get_docs(self, query):
    embedding_function = MyEmbeddingFunction()
    # 数据库
    persist_directory = "lab3_1\chroma_data_300"
    # 加载 Chroma 数据库
    db = Chroma(
        persist_directory=persist_directory,
        embedding_function=embedding_function,
    )

def fetch_docs(query, num_results):
    try:
        docs = db.similarity_search(query, num_results)
        res = ""
        j = 0
        if docs:
            for i, doc in enumerate(docs):
                # 筛选输出的结果
                if len(doc.page_content) >= 35 or ("\n" in doc.page_content):
                    doc_content = (
                        doc.page_content
                        if doc.page_content[:5] != "data:"
                        else doc.page_content[6:]
                    )
                    res = res + f"Doc_{j + 1}:" + doc_content + "\n"
                    j += 1
            return res, j
        except Exception as e:
            print(f"Error during similarity search: {e}")
            return "", 0

# 首先尝试获取前10个相似文档
res, count = fetch_docs(query, 10)

# 如果结果少于5个，再尝试获取前20个相似文档
if count < 5:
    res, count = fetch_docs(query, 20)

return res

```

之后可以直接调用该方法获取问题的相似检索数据

```
print(MyEmbeddingFunction().get_docs("没有赡养老人就无法继承财产吗？"))
```

Task 3&4

经过试验，初步确定了以下 Prompt

```
identity = "你是专业的法律知识问答助手，必须严谨地按要求回答用户的提问。\\n\\n用户会向你提供参考文档（以Doc_{i}}命名，包含问题与回答，请直接忽略仅包含问题的片段。），\\n\\n如果回答时你需要引用文档请直接叙述其中内容，\\n\\n不允许在回答中包含文档编号或提到用户提供了文档（这对用户不可见）。"

template = ""请结合Context中文档的上下文片段来回答问题，禁止根据常识或其它已知信息回答问题。\\n\\n如果你不知道答案，直接简明扼要地回答“未找到相关答案”。

Question: {question}
Context: {context}
Answer:
"""
```

在阅读通义千问的官方手册时，发现可以仅使用其提供的 dashscope 库完成我们该部分的任务，对于问题 谁可以申请撤销监护人的监护资格？ 的回答效果也还不错（ Qwen-plus ）

```
PS D:\Study_Work\Electronic_data\CS\AAUniversity\Web\Lab\L3> & D:/python/python.exe d:/Study_Work/Electronic_data/CS/AAUniversity/Web/Lab/L3/Ask.py
根据提供的上下文，申请撤销监护人资格的主体包括：

- 有关个人：其他依法具有监护资格的人、被监护人的近亲属等；
- 有关组织：居民委员会、村民委员会、学校、医疗机构、妇女联合会、残疾人联合会、未成年人保护组织、依法设立的老年人组织、民政部门等。

具体条款见《民法典》第三十六条。如果前款规定的个人和民政部门以外的组织未及时向人民法院申请撤销监护人资格，则由民政部门向人民法院申请。
PS D:\Study_Work\Electronic_data\CS\AAUniversity\Web\Lab\L3> 
```

而由于本次实验要求基于 Langchain 实现，于是又阅读了 Langchain 的手册，写了第二版代码（ Qwen-turbo ），并要求其回答 借款人去世，继承人是否应履行偿还义务？

```
PS D:\Study_Work\Electronic_data\CS\AAUniversity\Web\Lab\L3> & D:/python/python.exe d:/Study_Work/Electronic_data/CS/AAUniversity/Web/Lab/L3/Askv2.py
借款人去世后，继承人是否应履行偿还义务取决于他们是否继承了借款人的遗产。根据《中华人民共和国继承法》第三十三条的规定，继承遗产应当清偿被继承人生前所负债务，偿还的范围以遗产的实际价值为限。如果继承人继承了遗产，则需在遗产实际价值范围内偿还债务；如果继承人放弃继承，则无需对债务负责。

因此，如果继承人继承了借款人的遗产，他们应在遗产实际价值范围内履行偿还义务；如果未继承遗产或放弃继承，则不应履行偿还义务。
```

上面两个问题依照的检索数据可以参考上一小节（ Task 2 ），而对于问题 没有赡养老人就无法继承财产吗？，虽然检索得到的数据中并没有直接相关的答案，大模型也能正确地进行回答（第二版），并且没有透露出我们在中间层为用户提供了文档

```
● PS D:\Study_Work\Electronic_data\CS\AAUniversity\Web\Lab\L3> &
未找到相关答案。
○ PS D:\Study_Work\Electronic_data\CS\AAUniversity\Web\Lab\L3> 
```

```
● PS D:\Study_Work\Electronic_data\CS\AAUniversity\Web\Lab\L3> & D:/python/python.exe d:/Study_Work/Electronic_data/CS/AAUniversity/Web/Lab/L3/Search.py
Doc 1:外孙女没有赡养过老人能代位继承老人遗产么？
遗产继承人先于遗产人去世的，代位继承还存在吗
Doc 2:将财产都留给某个子女，其他子女可以不赡养老人吗
没有得到财产和子女仍然有赡养义务，并且是法定义务（不以得到财产为条件），必须履行。
Doc 3:老人遗产继承问题。
老人留下一间房子，约20平米，是祖先传下来的，没有房产证。老人生前说是他两的养子。现在老大占用不还租。请问我有继承权吗？
Doc 4:我和后老伴结婚，后老伴可以继承房产吗
如果办理了结婚证，就可以继承房产。
Doc 5:养子没经过法律程序，有遗产继承权吗？
养子没经过法律程序，有遗产继承权吗？
Doc 6:寡嫂有权继承公婆的遗产吗
在我国继承法中，儿媳不属于法定继承人的范围，继承法只对丧偶儿媳对公婆尽了主要赡养义务的情况，即丈夫去世后，儿媳继续对公婆尽了主要赡养义务，才能做为第一顺序继承人
Doc 7:爷爷的房产孙子可以继承吗？
爷爷奶奶几年前去世，未留下遗嘱。家里父亲及两个姑姑已同意将房子给我，到公证处公证父亲及两个姑姑放弃继承后可以将房子直接过户给我吗？
Doc 8:媳妇是否有赡养婆婆的义务
丧偶儿媳无赡养义务，但若尽了主要赡养义务的，应做为公婆的第一顺序继承人
Doc 9:孤寡老人的财产继承问题
我伯伯没有结婚也没子女，只有我爸一个兄弟，我抚养我伯伯30年，我伯伯去年去世了，我爸爸说我伯伯的房子是他的，要卖掉，我不想卖，我爸就把我起诉了，说他是法定继承人，我没资格管，我想问下，上了法庭，我是否有这房子的部分产权，大概有多少，如果我这房子...
Doc 10:赡养老人孙子，孙女，外孙三代人真的没义务吗
根据《婚姻法》规定：有负担能力的孙子，外孙子女，对于子女已经死亡的祖父母、外祖父母，有赡养义务。这种赡养是有条件的，即须孙子女、外孙子女有负担能力，且祖父母、外祖父母的子女已经死亡。

○ PS D:\Study_Work\Electronic_data\CS\AAUniversity\Web\Lab\L3> 
```

两个库的使用方法其实差别不大，区别在于构建消息与模型传参的接口。对于 `dashscope`，可以这样使用

```
from dashscope import Generation
# 构建消息
messages = [
    {"role": "system", "content": identity},
    {"role": "user", "content": template.format(question=question, context=knowledges)},
]

# 调用大模型
response = Generation.call(
    api_key=os.getenv("DASHSCOPE_API_KEY"),
    model="qwen-plus",
    messages=messages,
    result_format="message",
)

# 处理响应
if response.status_code == 200:
    answer = response.output.choices[0].message.content
    # 处理文件名，去除特殊字符
    file_name = re.sub(r'[\\"/*?:"<>|]', "", question) + ".txt"
    print(answer)
    answer = messages + "\n\n" + answer
    with open(file_name, "w", encoding="utf-8") as file:
        file.write(answer)
else:
    print(f"HTTP返回码: {response.status_code}")
    print(f"错误码: {response.code}")
    print(f"错误信息: {response.message}")
    print(
        "请参考文档: https://help.aliyun.com/zh/model-studio/developer-reference/error-code"
    )
```

而对于 `Langchain`，可以这样使用

```
from langchain_community.chat_models.tongyi import ChatTongyi
from langchain_core.messages import HumanMessage, SystemMessage

# 构建消息
messages = [
    SystemMessage(content=identity),
    HumanMessage(content=template.format(question=question, context=knowledges)),
]

chatLLM = ChatTongyi(streaming=True, dashscope_api_key=DASHSCOPE_API_KEY)
response = chatLLM.invoke(messages)

# 处理响应
answer = response.content
file_name = re.sub(r'[/\*?:"<>|]', "", question) + "_v2.txt"
with open(file_name, "w", encoding="utf-8") as file:
    for message in messages:
        if isinstance(message, SystemMessage):
            role = "System"
        elif isinstance(message, HumanMessage):
            role = "Human"
        else:
            role = "Unknown"
        file.write(f"{role}: {message.content}\n")
    file.write("\n\n")
    file.write(answer)
```