

脚本说明汇总

笔记本：	我的第一个笔记本		
创建时间：	2024/11/19 10:47	更新时间：	2024/11/19 11:59
作者：	153klix022		
URL：	file:///D:/科大讯飞实习/上海市交通系统交易问答框架v3.0/项目详细说明/脚本说明汇...		

用面向对象编程的方式，把单个功能整合到一个类 或者是一个文件中，
在主程序中将所有功能串联在一起，但同时所有功能又是独立的，不会相互影响，保证了扩展性和独立性

一. 主程序：

App_gradio_combined.py 这个是主程序，用两个函数概括了聊天问答和商品检索的功能，分别是answer_question和search_image，然后启动gradio端去呈现两个函数的交互

```
from Figure_search.app_figure_search import search_image
from chatbot.app_gradio import answer_question
```

界面如下



商品图片检索功能

上传商品图片

将图像拖放到此处
- 或 -
点击上传

相似图片

商品型号

二. 项目结构:

/ 上海市交通系统交易问答框架 /	
Name	Last Modified
名词向量存储	2 hours ago
评测sql脚本数据集	a day ago
微调合并脚本	13 days ago
项目详细说明	13 days ago
chatbot	35 minutes ago
Figure_search	2 days ago
flagged	15 days ago
SQL_database	21 hours ago
utils	an hour ago
可视化结构.ipynb	8 days ago
app_gradio_combined.py	23 minutes ago
README.md	25 minutes ago
requirements.txt	14 days ago

项目结构如上图所示

三. 文件内容说明

(一) 名词向量存储

/ 上海市交通系统交易问答框架 / 名词向量存储 /		
Name		Last Modified
chroma_non2.db		2 hours ago
公司名称.txt		a day ago
检索.py		15 hours ago
向量存储_公司名称.py		2 hours ago
向量存储_项目名称.py		2 hours ago
项目名称.txt		a day ago
中标总表.json		a day ago
local_file.json		a day ago
search_Milvus.py		2 hours ago
vectorstore_company_name_Milvus....		2 hours ago
vectorstore_project_name_Milvus.py		15 hours ago

详细内容见名词向量存储的文档

(二) 评测sql脚本数据集

用于评测sql问答效果，是独立的文件夹，与脚本无关，当前内容不完整，后续进行完善

运行如下脚本即可进行评测

```
python 上海市交通系统交易问答框架/评测sql脚本数据集/测评sql_questions_100条微调.py
```

(三) 微调合并脚本

把微调部分的.gguf和本体的.gguf进行合并

(四) 项目详细说明

存放项目文档

(五) chatbot

存放聊天问答部分的脚本，路径如下

上海市交通系统交易问答框架/chatbot/app_gradio.py

下面的example_list.py描述了聊天界面需要的示例问题

上海市交通系统交易问答框架/chatbot/example_list.py

(六) Figure_search

存放图片检索部分的脚本，具体文件如下

/ 上海市交通系统交易问答框架 / Figure_search /	
Name	Last Modified
flagged	2 days ago
app_figure_search.py	2 days ago
best.pth	2 days ago
database_insert.py	2 days ago
pca.pkl	2 days ago
resnet_pytorch_feature_extract.py	2 days ago

app_figure_search.py是实现图片检索方法的脚本，可以直接单独运行

best.pth文件是Resnet18训练后的权重文件，用于图向量的生成

database_insert.py是用于插入数据至数据库的脚本，已经完成，无需再运行

pca.pkl是pca降维算法的实现文件，因为图向量维度过大无法存入数据库，因此需要降维

resnet_pytorch_feature_extract.py 用于特征提取，提取出需要的embedding.pkl和pca.pkl

embedding.pkl在上传完数据后已经删除，因为文件过大

(七) SQL_database

用于存放本地sqlite数据库

(八) utils

这个文件夹作为工具包存放了所有实现分支功能的脚本

/ 上海市交通系统交易问答框架 / utils /	
Name	Last Modified
label_predict	15 days ago
chroma.sqlite3	a month ago
enhancement_functions.py	a day ago
function_tools.py	3 hours ago
law_retriever.py	2 hours ago
law_selection.py	11 days ago
noun_retriever_Mil.py	2 hours ago
noun_retriever.py	16 hours ago
retriever_SQL.py	21 hours ago
router_selection.py	3 hours ago
tools.py	a month ago

enhancement_functions.py 这个函数用于判断用户查询的sql相关问题是否包含具体或者模糊名称，然后进入不同的节点运行

你是一位专家，负责判断用户的问题是否包含具体的公司名称或项目名称，或者是否包含模糊指代（例如“不记得名字”、“可能是某个项目”、“公司名称记不清”等）。

****规则**:**

1. 如果问题中包含具体的公司名称（例如：“上海市XXXX公司”、“北京ABC有限公司”）或具体的项目名称（例如：“XXXXXX招标公告”、“项目Y进展报告”），返回“enhancement”。
2. 如果问题中包含模糊的公司名称或模糊的项目名称，例如：
 - “记不太清名字的公司”
 - “某个项目”
 - “可能是北京的一家公司”也需要返回“enhancement”。
3. 如果问题完全不包含具体的公司名称、具体的项目名称或任何模糊指代，则返回“no_enhancement”。

function_tools.py

存放了所有langgraph节点的回调函数

law_retriever.py

从milvus向量库的法律数据进行召回并处理

law_selection.py

选择法律问题的类型并进入节点，未完成的函数

noun_retriever_Mil.py

从Milvus数据库的名词向量库中召回名词，并且进行提示词处理

noun_retriever.py

从chroma数据库的名词向量库中召回名词，并且进行提示词处理

retriever_SQL.py

从sqlite数据库的名词向量库中召回名词，并且进行提示词和chain处理

router_selection.py

从初始问题进行三个支路的分支，分别是'vectorstore'、'answer_directly' 或 'law_query'

```
self.router_instructions = """
    你是一个专家，负责根据用户的问题将其引导到以下三个数据源之一：
    'vectorstore'、'answer_directly' 或 'law_query'。
```

tools.py

声明图数据结构类型

```
class GraphState(TypedDict):
    """
    Graph state is a dictionary that contains information we want to propagate
    to, and modify in, each graph node.
    """

    question: str # User question

    # Binary decision to run retriever
    # User question
    generation: str # LLM generation
    web_search: str # Binary decision to run web search
    max_retries: int # Max number of retries for answer generation
    answers: int # Number of answers generated
    loop_step: Annotated[int, operator.add]
    documents: List[str] # List of retrieved documents
    #retriever_type: str
```