

问题分类测评（法律,sql, 正常）

笔记本： 我的第一个笔记本

创建时间： 2024/11/22 10:51

更新时间： 2024/11/23 0:43

作者： 耿介

URL： about:blank

一.数据准备

首先准备大量的sql查询的问题，包括了中标数据，招标数据和异常等等，覆盖所有的表格提出的问题

然后准备法律的问题，包括一对一的，二对一的，三对一的，日常的非日常的问题

还有普通的日常问题，询问天气问候等

一共12000多条数据

二. 数据整合成json文件

把所有数据整理成merged.json

具体数据处理步骤不做赘述

```

...{
....."Question": "评审专家被选定且接受邀请后, 无正当理由未按时参与评审影响工作的, 会有什么后果?",
....."Answer": "law_query"
...},
...{
....."Question": "专利代理机构和专利代理师在执业时需要遵守哪些规定?",
....."Answer": "law_query"
...},
...{
....."Question": "合伙企业形式的专利代理机构申请办理执业许可证, 在名称方面需要符合什么要求?",
....."Answer": "law_query"
...},
...{
....."Question": "上海市公共资源拍卖中心在司法委托拍卖中主要负责哪些事项?",
....."Answer": "law_query"
...},
...{
....."Question": "上海市建设工程设计招标根据项目特点和实际需要可以采用哪些招标方式?",
....."Answer": "law_query"
...},
...{
....."Question": "专利代理机构和专利代理师在铁路工程项目招标投标管理中应遵守哪些规定?",
....."Answer": "law_query"
...},
...{
....."Question": "会员及其从业人员和客户需要遵守哪些规定?",
....."Answer": "law_query"
...},
...{
....."Question": "国家知识产权局和省、自治区、直辖市人民政府管理专利工作的部门依据什么对专利代理行业组织进行监督和管理?",
....."Answer": "law_query"
...}

```

三. 大模型分类答案生成并插入

调用问题分类中的route_question函数, 把函数返回的键值对添加到merged.json中作为预测值, 后面会把预测值与真实值进行对比计算准确率
 注意路径中必须包含router_selection.py
 实现代码如下

```

import json
from router_selection2 import Router # 假设你的脚本文件名是 your_router_script.py

# 初始化路由器
router = Router()

# 加载 merged.json
with open("merged.json", "r", encoding="utf-8") as file:
    data = json.load(file)

# 遍历问题并调用 Router
for entry in data:
    question = entry["Question"]
    state = {"question": question} # 构造适合 Router 的输入

    try:
        # 调用路由方法
        datasource = router.route_question(state)
        entry["PredictedDatasource"] = datasource # 添加新的键值对
    except Exception as e:
        # 如果出现任何错误, 打印错误信息并跳过当前问题
        print(f"Error processing question '{question}': {e}. Skipping...")
        entry["PredictedDatasource"] = 'notsure'
        continue

# 将更新后的数据保存回文件
with open("merged.json", "w", encoding="utf-8") as file:
    json.dump(data, file, ensure_ascii=False, indent=4)

```

```
print("Finished processing all questions!")
```

等待大模型生成12000多个问答对，得到下列结果：

三. 结果测评和bad_case整理

把预测值和真实值对比，计算准确率，然后把预测错误的case整理到新的json文件中，便于优化分析

代码如下

```
import json

# 加载更新后的 merged.json
with open("merged.json", "r", encoding="utf-8") as file:
    data = json.load(file)

# 检查 data 是否为列表
if isinstance(data, list):
    # 初始化统计变量
    total = 0
    correct = 0
    errors = []

    # 遍历问答对，检查正确性
    for entry in data:
        # 确保 entry 是字典，并包含所需的键
        if isinstance(entry, dict) and "PredictedDatasource" in entry and "Answer" in entry:
            total += 1
            if entry["PredictedDatasource"] == entry["Answer"]:
                correct += 1
            else:
                # 保存错误预测的问答对
                errors.append({
                    "Question": entry.get("Question", "N/A"),
                    "Answer": entry["Answer"],
                    "PredictedDatasource": entry["PredictedDatasource"]
                })
        else:
            print(f"警告：条目 {entry} 格式无效，已跳过。")

    # 计算准确率
    accuracy = correct / total if total > 0 else 0
```

```

# 保存错误预测到新文件
with open("errors.json", "w", encoding="utf-8") as file:
    json.dump(errors, file, ensure_ascii=False, indent=4)

# 输出结果
print(f"总数: {total}")
print(f"正确预测: {correct}")
print(f"准确率: {accuracy:.2%}")
print(f"错误预测保存到 errors.json 文件中!")
else:
    print("错误: JSON 文件内容不是列表, 请检查文件格式。")

```

结果为

```

总数: 12017
正确预测: 8946
准确率: 74.44%
错误预测保存到 errors.json 文件中!

```

这个准确率是远远不够的

四.提示词优化

修改了提示词, 做出了概括性的范围覆盖

```

1. **vectorstore**:
    - 包含与招标项目相关的问题, 尤其是提及项目名称。
    - 关键词包括: 项目、采购、招标、公告、中标、变更、异常中标结果、变更和异常、
      招标公告、项目分类, 网址查询, 时间查询,
      价格查询, 项目编号等。
    - 示例问题:
      - "铁路中标结果一共有多少条?"
      - "请提供最新的招标公告。"
      - "有哪些项目发生了变更?"
      - "xxxx能否提供相关网址"
      - "xxxx公司的网址是什么"
      - "xxxx的招标时间"
      - "项目编号为JTHY20240718002的项目有结果了吗"
    - 对于涉及招标项目细节的问题,
      满足上述条件的, 选择 'vectorstore'。

2. **law_query**:
    - 包含与法律法规法条相关的内容。
    - 适用于涉及行为合规性判断、具体法律规定的提问。
    - 或者包括这些关键词, 关键词包括, 法律、法条、条例、规定、合规、程序、权限、
      义务、责任, 专利代理
    - 含有这种符号《》
    - 示例问题:
      - "中华人民共和国招标投标法实施条例具体涉及哪些信息?"
      - "投标过程中有哪些法律要求?"
      - "这种行为是否符合相关法规?"
      - "专利代理师对其签名办理的专利代理业务负责?"
    满足上述条件的, 对于涉及法律法规的问题, 选择 'law_query'。

3. **answer_directly**:

```

- 用于所有其他问题，尤其是问候、闲聊等。
- 示例问题：
 - "你是谁？"
 - "今天天气怎么样？"
 - "能告诉我一个笑话吗？"

包括上述以及对于无法归类到以上两个数据源的问题，选择 'answer_directly'。

重新进行测评

总数：12114

正确预测：12055

准确率：99.51%

错误预测保存到 errors.json 文件中！

准确率达到预期