暑期学校实验项目：高考志愿填报助手

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 小组名称 | 知识图谱构建B组 | | | | | | |
| 姓 名 | 刘漪琛 | 专业 | 计算机科学与技术（人工智能） | 班级 | 091182 | 学号 | 09118201 |
| 实验时间 | 2020.8.31-2020.9.23 | | 指导教师 | 孔祥龙 | | 成绩 |  |
| **一、实验背景和目的**  **实验背景：**高考志愿填报是人生中关键的一环，如何根据自己的成绩与兴趣选择合适的高校，让自己在大学生活中得到提高与充实是每一个人都需要考虑的问题，合理的选择无疑让自己接下来四年的本科生活有一个良好的开端。  **实验目的：**为了让学生在填报志愿时能够对自己成绩以及所填报高校的信息有更加充分的了解，并且在此基础上给出学生一定的合理建议，我们开发了这个高考志愿填报系统。 | | | | | | | |
| **二、小组任务和个人任务**  **小组任务：**知识图谱构建B组负责构建省份与高校，省份分数信息，高校分数信息的知识图谱，同时开发网页前端形象化地展示知识图谱的构建与相关分数信息，从而使填报志愿的学生更加深入地了解其目标省份的高校情况及相关分数信息。  **个人任务：**本人负责构建知识图谱的后端开发，建立适用于知识图谱的数据库，同时编写了部分从数据库中调用相关信息返回到前端的函数。 | | | | | | | |
| **三、个人任务需求分析**  构建知识图谱的后端开发旨在建立适用于知识图谱的数据库，存储省份--学校，省份--分数，一分一档三类的数据，为信息的查询分析、知识图谱的可视化做好基础。   1. 数据库设计   Class：省份-分数   |  |  |  |  | | --- | --- | --- | --- | | 字段名 | 类型 | 关系 | 备注 | | 年份 | int |  | 四位。例：2018 | | 文理科 | string | ‘art’ or ‘science’ or ‘no’ | | 一本线 | int |  | 例：420 | | 二本线 | int |  | 例：400 | | 专科线 | int |  | 例：300 |   Class：一分一档   |  |  |  |  | | --- | --- | --- | --- | | 字段名 | 类型 | 关系 | 备注 | | 省份名称 | string | 主键 |  | | 年份 | int |  | | 文理科 | string |  | | 分数 | int |  | | 排名 | int |  |  |   Class：省份-学校   |  |  |  |  | | --- | --- | --- | --- | | 字段名 | 类型 | 关系 | 备注 | | 省份名称 | string | 主键 |  | | 文理科 | string |  | | 省内可选学校 | Class 学校 |  |  |  1. 提供的查询需求   省份为中心实体  （1）用户可以通过省份来查询排名信息  （2）用户可以选择某一特定省份内的学校  （3）用户可以通过省份来查询相应的一本、二本线等分数信息  （4）通过知识图谱可视化的数据库可以直接展示数据库的相关信息     1. 为网页前端提供的接口   前端模型中设计直接从数据库中调用相关分数信息的函数并利用前端网页可视化展示。 | | | | | | | |
| **四、实验过程（需附上关键代码及相关说明）**  1. 数据库的建立使用的是Neo4j图数据库，首先在本地配置了Java SDK，采用的版本是jdk-11.0.8，然后安装了Neo4j社区版，采用的版本是neo4j-community-4.1.1，使用python中的py2neo库对数据库进行操作。  2. 连接到数据库之后开始创建相应的节点信息。  3. 创建省份节点： *# 创建34个省份节点* provinceNode = [] *# 存储创建好的34个省份节点，用于后续创建关系* **for** i **in** range(provinceinfo.shape[0]):  provinceNode.append(Node(**'省份'**, ID=int(provinceinfo.iloc[i][0]), Name=str(provinceinfo.iloc[i][1])))  graph.create(provinceNode[i])  创建之后Neo4j数据库中的一个省份节点中的信息为（以河南省为例）：  {  "identity": 147,  "labels": [  "省份"  ],  "properties": {  "ID": 1,  "Name": "河南"  }  }  Neo4j对数据库进行图的可视化之后为：  pic1  相应地创建了学校节点和分数信息节点。   1. 建立学校与省份节点之间的关系，分数信息与省份的关系，至此知识图谱数据库的建立已经完成。 2. 为了使前端网页能够获得数据库中的信息，编写了从前端向数据库发出请求获得相应节点信息的函数：   *#获取数据库中分数信息节点的信息* **def** get\_scoreline(provinceID):  scoreline = {  **'firstLine2017Art'**: -1,  **'firstLine2018Art'**: -1,  **'firstLine2019Art'**: -1,  **'firstLine2017Sci'**: -1,  **'firstLine2018Sci'**: -1,  **'firstLine2019Sci'**: -1,  **'secondLine2017Art'**: -1,  **'secondLine2018Art'**: -1,  **'secondLine2019Art'**: -1,  **'secondLine2017Sci'**: -1,  **'secondLine2018Sci'**: -1,  **'secondLine2019Sci'**: -1,  }  matcher = NodeMatcher(graph)  data = matcher.match(**"分数信息"**, provinceID=provinceID).all()  **for** item **in** data:  d = dict(item)  **if** d[**'category'**] == **'文科'**:  cate = [**'Art'**]  **elif** d[**'category'**] == **'理科'**:  cate = [**'Sci'**]  **elif** d[**'category'**] == **'综合'**:  cate = [**'Art'**, **'Sci'**]  *#获取分数线数据list的长度* **if** len(d[**'scoreLineValue'**]) > 2:  length = 2  **else**:  length = len(d[**'scoreLineValue'**])  *#添加分数线信息* **for** i **in** range(length):  **for** word **in** cate:  **if not** i:  scoreline[**'firstline'** + str(d[**'year'**]) + word] = d[**'scoreLineValue'**][i]  **else**:  scoreline[**'secondline'** + str(d[**'year'**]) + word] = d[**'scoreLineValue'**][i]  **return** scoreline   1. 相应的添加了从前端获取省份信息、学校信息、学校与省份之间关系、分数信息与省份之间关系的函数，从而便于网页前端的可视化工作。 2. 由负责前端的同学设计了网页的风格，排版，跳转，信息展示与可视化等等内容。 | | | | | | | |
| **五、实验结果与分析**  1. 后端数据库知识图谱的可视化展示结果：  （1）省份信息  pic1   1. 学校信息   pic2   1. 分数信息   pic4  （4）省份-学校关系  pic3  （5）省份-分数关系  pic5   1. 结果分析： 2. 借用Neo4j图数据库基本实现了省份信息、学校信息以及分数信息之间的知识图谱的建立。 3. 由于原始csv文件提供的信息不太完整，故建立的知识图谱缺失了部分节点以及关系的展示。 | | | | | | | |
| **六、实验总结与心得体会**  通过本次《软件实践》课程实验，认识到了团队协作开发软件的大致流程与运转方式，了解了使用django进行web开发的基本知识；通过在知识图谱B组担任后端数据库开发的一员，初步了解了知识图谱的原理及构建过程，接触到一些用于建立知识图谱的专业数据库，学会了使用Neo4j与他人合作建立小型的数据库并实现知识图谱的可视化，同时了解了一些前端开发的知识，并尝试将前端网页与后端数据库相联系，实现了信息的交互。  这次团队开发的经历让人认识到队员之间优势互补，协同合作的重要性，这也是在今后学习、科研、工作中需要继续锻炼的地方。 | | | | | | | |

2020年9月制