《软件实践》课程实验报告

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 A 组						
姓　　名	张骥	专业	人工智能	班级	091182	学号	09118242
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：

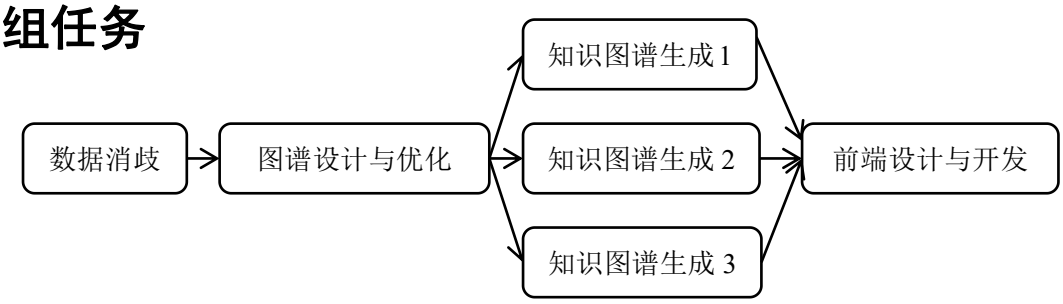
高考是人生的重要关口，如何选择自己最适合的学校是历年各个考生都关注的重要问题。如何选择专业，如何观察历年录取分数线变化，如何了解各个学校在不同省份的录取分数差异，是每年高考结束以后学生与家长共同的关注焦点。

实验目的：

通过构建知识图谱，训练人工智能算法实现出一个推荐算法，满足学生与家长对高考志愿填报的咨询需求。

二、小组任务和个人任务

小组任务



任务 1：数据消歧

本项目需要用到的数据源，是第一组清洗的包含学校，专业，省份，分数，年份的 csv 文件。由于专业名称等信息存在相同专业不同名称等现象，需要先进行消歧工作。

任务 2：知识图谱设计与优化

利用已有的数据构建一个小型的报考知识图谱(知识库)，通过调用该图　谱可以实现如下功能：

1. 已知自己某分数能上什么学校
2. 某个特定的专业哪个学校分数最高
3. 已知自己的分数判断自己能学什么样的专业
4. 查询某学校的特定专业
5. 我只想学 XX 专业，能去什么学校？

任务 3：知识图谱生成

此任务包括：

1. 对所有实体生成可以导入Neo4j 的 csv 文件
2. 对所有关系生成可以导入Neo4j 的 csv 文件

3.将以上文件导入 neo4j，生成知识图谱
4.再根据图谱，改进不足

任务 4：基于构建好的知识图谱，构建显示网页

此部分属于前端操作，主要考虑图谱可视化效果

个人任务

任务 1：创建可以导入Neo4j 的部分 csv 文件

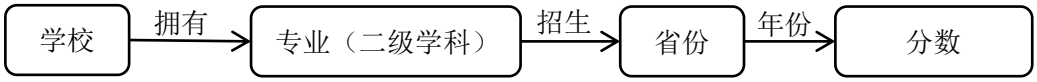
任务 2：将可以导入的 csv 文件导入 neo4j,初步形成知识图谱

三、个人任务需求分析

任务 1：创建可以导入Neo4j 的部分 csv 文件

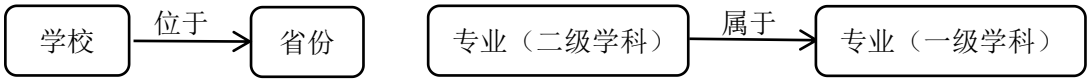
其他同学已完成知识图谱设计，并且对原始数据进行消歧以及命名主键等工作，在本任务中，我的任务是从预处理后的数据，提取出实体、关系，创建出可以导入的 csv 文件。

在知识图谱中，每个节点就是实体，节点之间的联系就是关系。由于我们组是从学校入手，所以我们设计了以下图谱：



其含义为：某学校拥有多个专业，该专业在不同省份招生，不同省某年该专业不同学校的分数线。我的具体工作是，创建分数实体，包含文理科、学校、专业等属性；创建省份-分数关系，关系属性包含年份。

此外，还可以添加以下关系：



这两个关系已存在，只需导入即可。

任务 2：将可以导入的 csv 文件导入 neo4j,初步形成知识图谱

在结束任务 1 后，我们已经得到可导入的 csv 文件。之后利用命令行将文件按 entity 和 relation 批量导入，之后可以在网页浏览结果。

四、实验过程（需附上关键代码及相关说明）

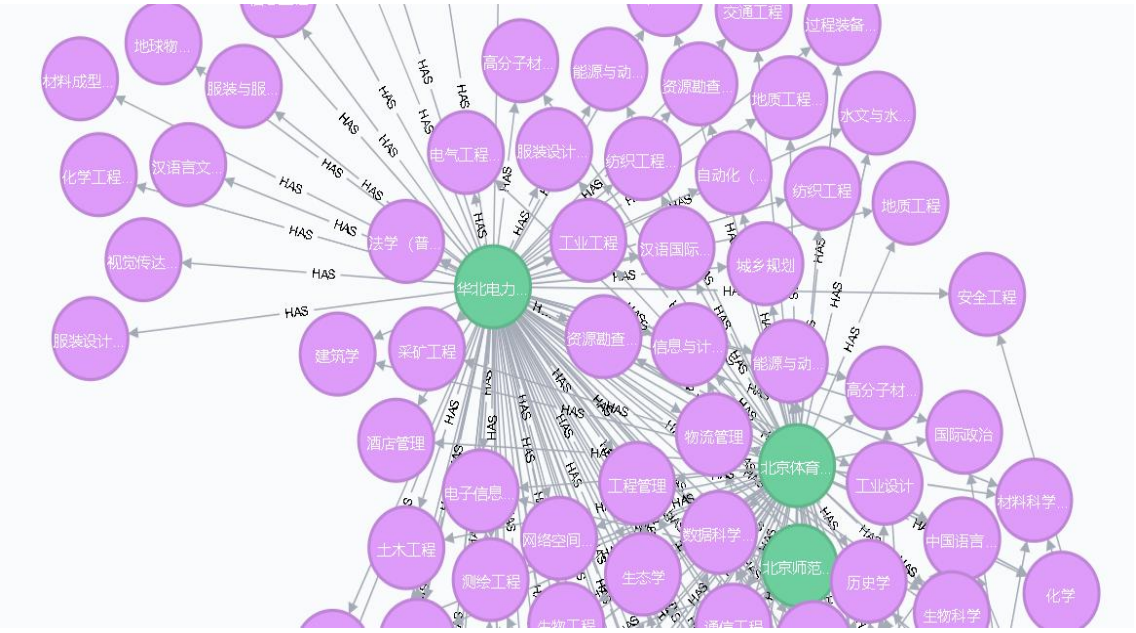
```
####################创建分数实体和省份-分数关系###################
import numpy as np
import pandas as pd

major_file = pd.read_csv(r'major.csv',encoding = 'gbk') #读取专业文件
province_file = pd.read_csv(r"entity\province.csv") #读取省份文件
province_file.rename(columns={'Province:ID':'Province'}, inplace = True) #将列名修改为 neo4j 格式
df = major_file.merge(province_file,how = 'left') #将两张表连接
df.sort_values(by=['Major:ID'])

df['Province'].drop_duplicates() #将'Province'列去重后输出，发现有脏数据
#数据清洗
df['Province'] = df['Province'].replace('宁  夏', 'p31')
df['Province'] = df['Province'].replace('全国联招', 'p36')
df['Province'] = df['Province'].replace('华侨', 'p35')
df['Province'].drop_duplicates()

college_file = pd.read_csv(r'college.csv') #读取学校文件，目的是在分数实体标注学校
college_file.drop(['985:int','211:int','Top:int',':LABEL'],axis = 1,inplace =True)
df1 = college_file.merge(has_file,how = 'left')
df2 = df.merge(df1,how = 'inner') #将分数与学校关联
df2['Major:ID'] = ['M{}'.format(x) for x in df2.index] #赋予唯一标识符
score = df2[['Major:ID','category','score','Contributor','Name','Major']]
score[':LABEL']='Score'
score.to_csv(r"D:\score.csv",index=None) #保存分数实体

province_score = df2[['Province','Major:ID','Year']]
province_score.rename(columns={'Province':':START_ID','Major:ID':':END_ID','Year':':TYPE'}, inplace = True)
province_score.to_csv(r"D:\province_score.csv",index=None) #保存省份-分数关系
```

```
####################修改#########################
import pandas as pd


 #赋予唯一标识符
df = pd.read_csv('D:\province_score.csv')
df.drop(['END_ID'],axis = 1,inplace =True)
df[':END_ID'] = ['M_{}'.format(x) for x in df.index]
df.to_csv(r"D:\province_score.csv",index=None)


 #将列名修改为 neo4j 格式
df = pd.read_csv(r'D:\province.csv')
df.rename(columns={'Province':'Province:ID'}, inplace = True)
df.to_csv(r"D:\province_import.csv",index=None)


 #将列名修改为 neo4j 格式
df = pd.read_csv('D:\college.csv',encoding = "gbk")
df.drop(['Unnamed: 3','Unnamed: 4','Unnamed: 5'],axis = 1,inplace =True)
df[':LABEL'] = 'College'
df.to_csv(r"D:\college.csv",index=None)


#将标签修改为内容
df = pd.read_csv('D:\score.csv')
df.drop(['END_ID'],axis = 1,inplace =True)
df['category'] = df['category'].replace('s1', '文科')
df['category'] = df['category'].replace('s2', '理科')
df['category'] = df['category'].replace('s3', 'all')
df.to_csv(r"D:\score.csv",index=None)
```

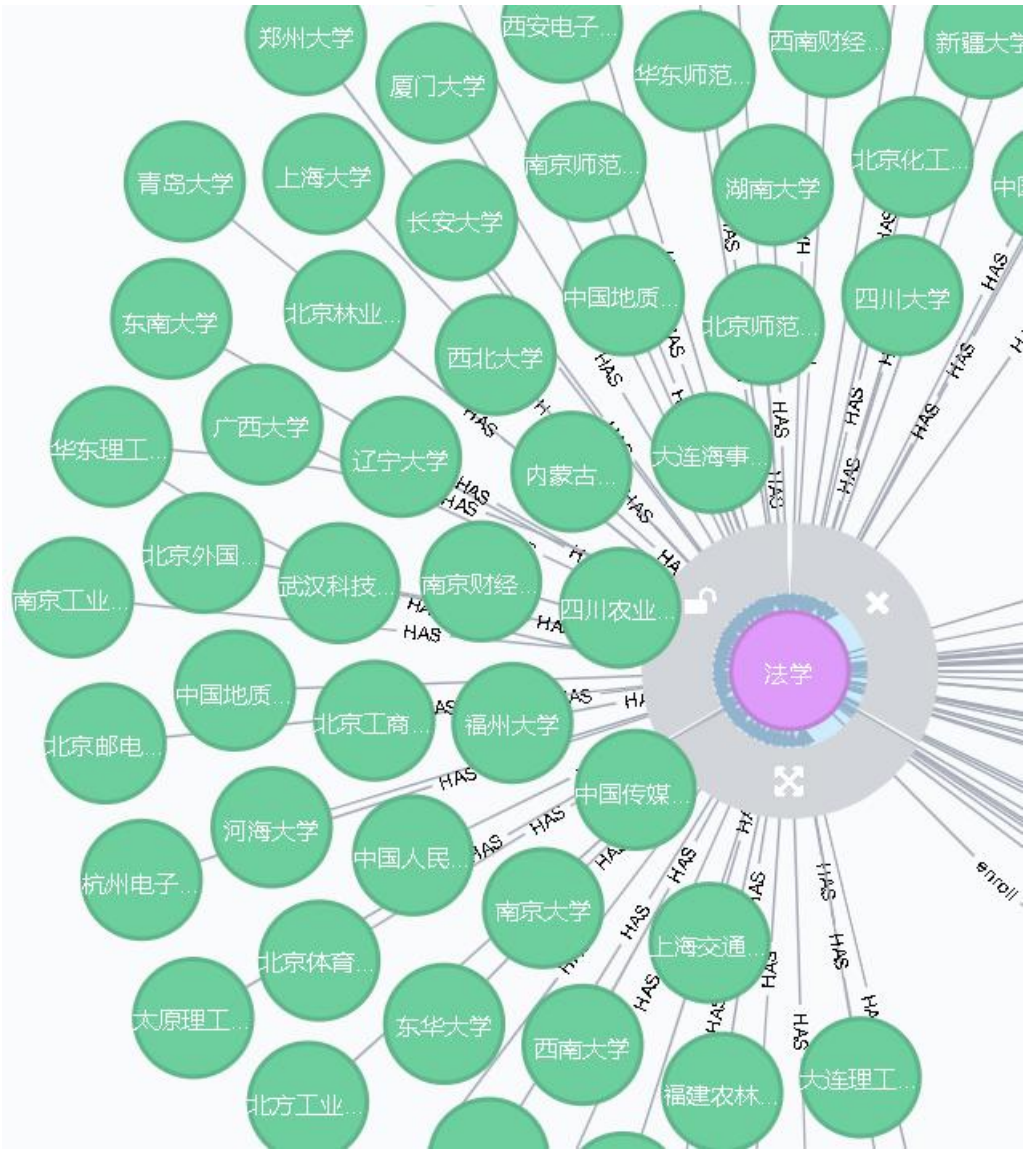
五、实验结果与分析

初始可以打开学校视图，如图，显示不同的学校

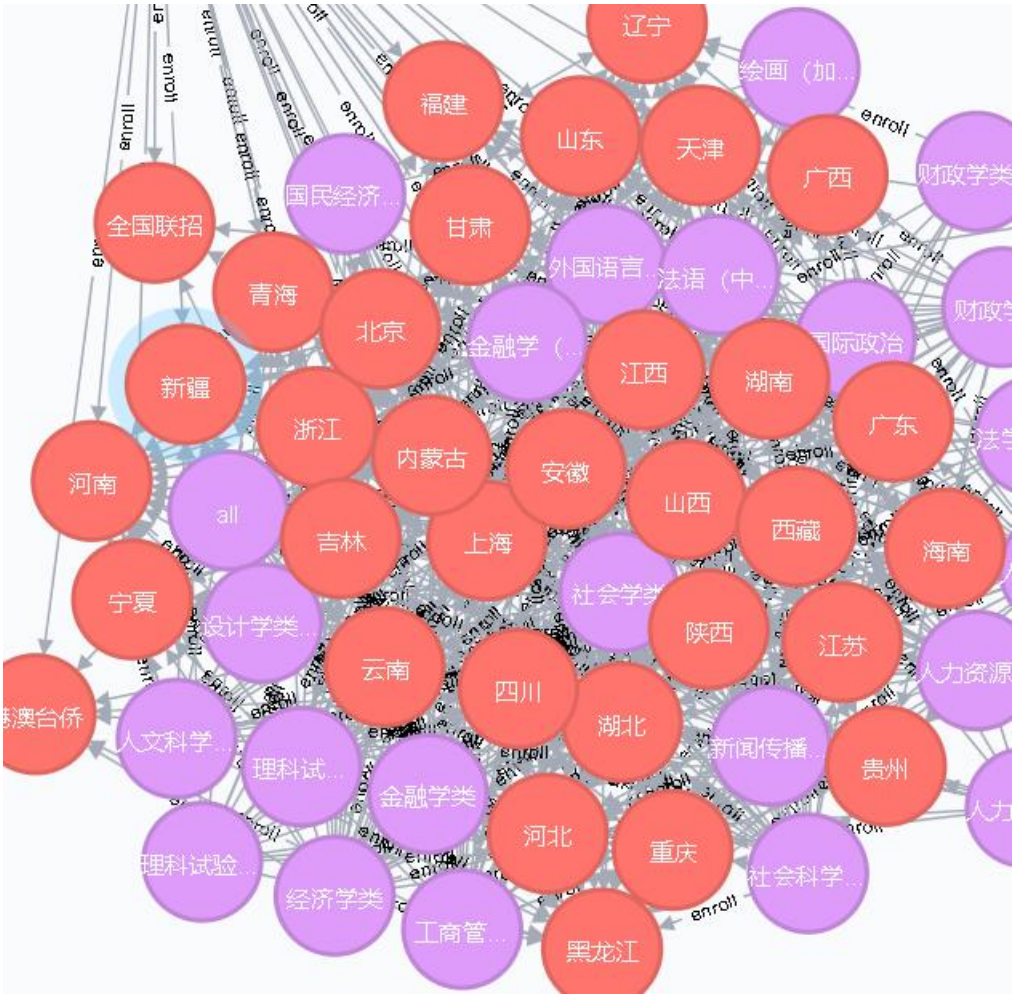
点击一个学校，可以看到它拥有的专业（HAS）



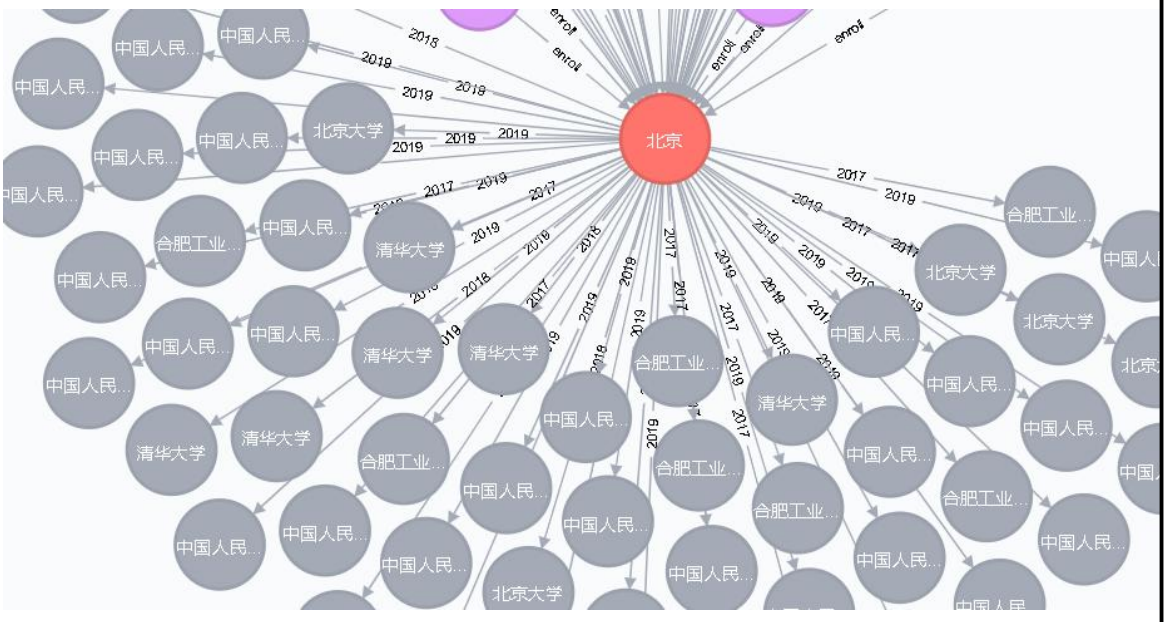
点击某一专业，如法学，可以看到拥有该专业的学校



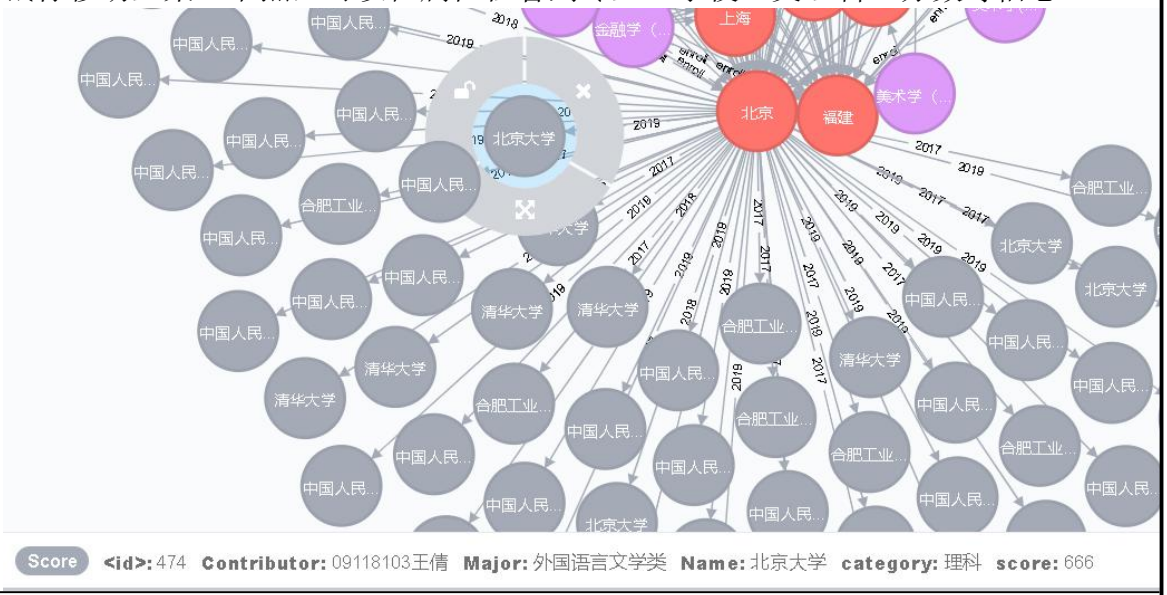
也可以看到它在那些省份招生



点击北京省，可以看到北京省该专业某一年对应分数线



鼠标移动至某一节点，可以在属性栏看到专业、学校、文理科、分数等信息



分析：

由于我们有四类节点（学校、专业、省份、分数），所以可以有四个视图。初始视图都是一个一个独立的节点，在选择一个节点后，会出现许多与之关联的节点。一次选择，显示还是比较清楚，但是在两次以上的选择后，会显示大量节点与关系，不仅出现卡顿，而且会有大量数据聚集导致可视化效果很差。一个现有的方法是在选择节点后切换视图，清空之前的显示。

此外，我们所希望的是点击学校后，显示该学校专业，点击专业，显示该学校该专业的招生省份，即对节点的选择存在向前的包含关系。经过观察，我们发现，在点击一个节点后，再点击一个节点，所有与之相连的节点都会显示，即我们选择学校的专业，显示的还有其他拥有该专业的学校，而该专业连接的省份，是所有学校该专业招生的省份，省份与分数的关系连接与此类似。这与我们的预期相反。

六、实验总结与心得体会

这次实验，我的两个任务分别是处理数据和导入 neo4j。在数据处理这部分，我主要利用 pandas 库来实现。相比于传统的 python 代码处理，无论是读写还是清洗数据，pandas 都很便捷。第一个任务的完成，让我对数据处理有了更多的认识。neo4j 是我对知识图谱的第一次接触。这次实验，也让我初步了解知识图谱技术，以及对 neo4j 的使用。

此外，从本次项目的角度，一方面，一个巨大的任务分摊下来，每个人都可以去做自己擅长的地方，可以说扬长避短，提高了效率。还有 Git 仓库的使用也是本次实验的收获。另一方面，任务的明确以及组员的协调是一个难题，比如在我们组出现了工作的冗余。总之，这次课程的经验将为我以后的项目经历提供很大的帮助。

2020 年 9 月制