项目文档 (作业三)

路线三 11 组

组长:

宋定杰 学号: 171250628 手机: 18851132226

组员:

姓名: 李辰辉 学号: 171250645 姓名: 梁斌 学号: 171830506 姓名: 陈维烨 学号: 171250599

组员分工

任务 1:

陈维烨:数据爬取、清洗、整理;扩充数据,获取更多数据源;与其他组别进行数据交换

梁斌: 数据库搭建; 后端数据持久层搭建; 数据集成;

任务 2、3:

宋定杰: 服务端搭建; 知识问答系统搭建 李辰辉: neo4j 数据库搭建; 图数据持续集成

项目 git 地址

https://github.com/bbsngg/Data-Integration/

项目地址

图数据展示: http://ismzl.com:3000/

基于图数据的股票知识问答系统: http://ismzl.com:3000/ga.html

数据库地址

Neo4j: http://ismzl.com:7474/browser/

MySQL: http://ismzl.com:3306

数据获取与数据整理

数据源

作业三数据源分为两部分来源:一部分是其他组的数据,另一部分是来自 http://pycs.greedyai.com/的新增数据。由于更细节的基金和债券等信息需要在 tushare 上获取,而获取相应数据需要大量付费积分,因此无法将 tushare 作为新的数据源来获取数据。

获取方式

第一部分中包括另外两组的数据,直接通过联系其他组组长获取。

第二部分数据仍然通过 scrapy 进行爬取,爬取重点变为董事会相关信息,同时由于标签更加不具有特殊性,在定位数据位置时添加了对 xpath 表达式的使用。

具体而言是获取到董事会、监事会和高管面板的 tbody 标签, 然后获取其下的所有 tr 元素。观察到每一行 tr 元素至少包含一位管理层成员信息, 至多两位, 用于存储信息的 td 标签以三个作为一组, 因此直接通过索引值对特定的内容进行获取:

对于索引值 x: 若 mod(x)=0,则获取字段为人名;若 mod(x)=1,则获取字段为职位;若 mod(x)=2,则获取字段为持有股数。

最后为了使得新的数据能够与原有数据产生关联, 在每一行数据开头都加入了 stock_no 信息。

相较于参考 demo 的董事会信息数据,该部分还增量爬取了监事会和高管层的成员数据,也就是在公司高层成员数据方面更加完善,同时增加了数据源有提供的持股信息。对于回调函数的调用与传参处理与作业二相同。

集成流程

- 1. 将多个数据的数据源导出,为避免各方数据源格式不唯一的情况,我们从数据源爬取下来的数据以及从其他小组获取的数据统一抽取为 csv 格式。
- 2. 将所有的 csv 文件进行初步过滤, 如清除无效字段后导入数据库。
- 3. 按照我们数据库内定义的数据字段与类型对数据进行二次过滤,将增量获取的数据分为以下几类:
 - a) 重复数据:对此类数据进行筛除,抛弃无用字段或只将部分数据添加到已有数据库中,如对于已存在数据库中的实体,扩充数据只有部分字段可供使用,其余则被抛弃(不从 CSV 导入)。
 - b) 命名冲突数据: 对于某些实体的相同属性在不同表 (CSV 文件) 中可能存在不同的命名, 需要对其进行分析, 查找相同的属性, 再做数据合并。
 - c) 包含新实体类的数据: 对于有新的实体类的数据, 我们新建了表保存, 以保证数据 库表和实体的一致性。
- 4. 遇到困难:信息去重——按照有相同信息的字段进行去重,如股票名有出入但是根据股票代码这一主键可以去重。

关键代码

获取三个面板页面

```
boss_table = response.css("div#ml_001 tbody") #董事会成员 list_table = response.css("div#ml_002 tbody") #监事会成员 high_table = response.css("div#ml_003 tbody") #高管成员 tr_list1 = boss_table.css("tr") tr_list2 = list_table.css("tr") tr_list3 = high_table.css("tr")
```

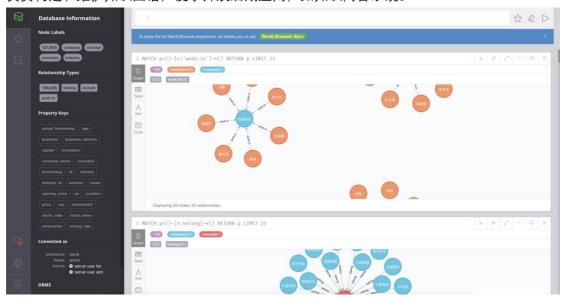
获取字段信息

```
person_1 = td_list[0].css("a::text").get()
position_1 = td_list[1].xpath("./text()").extract()[0]
stocknum_1 = td_list[2].css("span.to_tip::text").get()
if stocknum_1 is None or len(stocknum_1)<=2:
    stocknum_1 = 'NULL'
with open(r'G:\scpy\MyDtScpy\MyDtScpy\MyDtScpy\data.txt', 'a') as bossFile:
    row = stock_no + '\t' + meeting + '\t' + person_1 + '\t' + position_1 + '\t' + stocknum_1
    bossFile.write(row)
    bossFile.write('\n')
    bossFile.close()
    print("yield one: " + row)</pre>
```

数据管理

Neo4j

负责构建和维护知识图谱,便于开展后期应用,如知识问答系统。



MySQL

负责维护数据之间的关系,便于后端对数据进行计算挖掘与前端展示。 股票:

概念:

董事:

产业: