

信息检索实验报告HW7

姜欣妮 1811482

一、实验要求

构建基于知识图谱的问答系统。

二、项目说明

1. 环境: pycharm+python3.8

2. web框架: web.py

3. 图数据库: neo4j

4. 项目目录:

- 数据:

其中**data文件夹**储存数据，其中**csv文件夹**存储原始数据csv和关系链接csv，**question文件夹**存储问题模板和问题分类的训练数据，**userdict.txt文件**存储一些实体名称（如人名）及其属性（属性为自定义的属性，用于词性标注），**neo4j.txt文件**存储构建图数据库的命令语句。

- web前端:

static文件夹是web前端的静态文件，**templates文件夹**是web前端的html文件

- 代码:

deal_csv.py文件用于处理csv文件，获取其关系链接。

get_userdict.py文件用来获取实体名称并赋予自定义属性。

question_classification.py文件用来训练问题模型，并根据模型对用户的问题匹配模板。

question_template.py文件用来定义不同模板的查询，并根据问题模板来做对应的处理，获取答案。

query.py文件是连接图数据库neo4j并进行查询的接口。

question文件是查询的主文件，功能为接收原始问题，对原始问题进行分词、词性标注等处理，对问题进行抽象，匹配模板，根据问题模板进行查询。

test文件是测试文件。

4. 运行方法: 安装requirements.txt里的包后，运行code.py，在浏览器中打开<http://localhost:8080/>。

三、具体实现

（一）获取数据

由于南开大学相关网页太多太杂，并且没有特定的规范格式，难以抽取关系链接，这里选取南开大学相关信息和计算机学院的相关信息。

将数据存入NK.csv文件和person_all.csv文件中。

csv文件，例：

pid	name	pt	department	research					
1	白刚	教授	物联网工程系	人工智能，机器学习，模式识别，计算机视觉					
2	程仁洪	教授	计算机科学与技术系	数据库技术及应用，软件性能，智能控制与检测系统					
3	程明明	教授	计算机科学与技术系	人工智能、计算机视觉、图像视频大数据分析、计算机图形学					
4	贾春福	教授	信息安全系	网络与信息安全；可信计算与软件安全；恶意代码分析；密码应用技术					
5	刘健	教授	计算机科学与技术系	大数据、人工智能、生物信息学、类脑计算、智能医学与合成生物学					
6	刘哲理	教授	信息安全系	数据安全；人工智能安全					
7	刘曙光	教授	计算机科学与技术系	云存储 搜索引擎 区块链系统 云计算/大数据平台 计算经济学					
8	李庆诚	教授	计算机科学与技术系	嵌入式操作系统与信息安全					
9	李涛	教授	物联网工程系	异构计算，机器学习，智能物联网					
10	邵秀丽	教授	计算机科学与技术系	人工智能、数据分析、智能系统、CSCW协同控制					
11	卫金茂	教授	计算机科学与技术系	机器学习，数据挖掘，生物信息学					
12	汪定	教授	信息安全系	公钥密码学，系统安全，人工智能安全					
13	王刚	教授	计算机科学与技术系	海量信息存储、并行与分布式计算					
14	徐敬东	教授	计算机科学与技术系	移动边缘计算、软件定义网络（SDN）、网络大数据分析、网络安全等					
15	辛运伟	教授	计算机科学与技术系	信息自动化 智能软件系统 数字医疗系统					
16	杨巨峰	教授	计算机科学与技术系	计算机视觉、机器学习、多媒体计算					
17	杨征路	教授	计算机科学与技术系	人工智能，数据挖掘，信息检索					
18	杨愚鲁	教授	计算机科学与技术系	并行与分布式处理，系统结构，可重构系统，网络计算					
19	袁晓洁	教授	信息安全系	数据库、知识工程、大数据技术、机器学习、Web检索与挖掘					
20	张健	教授	信息安全系	云安全，系统安全、恶意代码防治、人工智能安全、电子数据取证、网络犯罪打					
21	张建忠	教授	物联网工程系	网络大数据分析，网络安全，移动计算，软件定义网络，边缘智能					
22	蔡庆琼	副教授	公共计算机基础教学部	图算法设计与分析、图论与组合优化、图神经网络					

（二）获取关系链接

总结出四类实体：南开相关、教师、职称、所属部门。

将单独的实体及其属性抽出存入csv文件。

获取教师和职称、教师和所属部门之间的关系链接并存入csv文件中。

```
if __name__ == '__main__':
    # 教师
    person = pd.read_csv("data/csv/person_all.csv")
    # 职称
    pt = pd.read_csv("data/csv/professional_title.csv")
    # 所属部门
    department = pd.read_csv("data/csv/department.csv")
    # print(person)

    person_pt_pid = []
    person_pt_tid = []

    person_d_pid = []
    person_d_did = []

    # 获取关系链接
    for i in range(len(person)):
        # 职称
        if person['pt'][i]:
            tid = 0
            for j in range(len(pt)):
                if pt['name'][j] == person['pt'][i]:
                    tid = pt['tid'][j]
            person_pt_pid.append(person['pid'][i])
            person_pt_tid.append(tid)
        # 所属部门
        if person['department'][i]:
            did = 0
```

```

for k in range(len(department)):
    if department['name'][k] == person['department'][i]:
        did = department['did'][k]
        person_d_pid.append(person['pid'][i])
        person_d_did.append(did)

person_to_pt_dict = {'pid': person_pt_pid, 'tid': person_pt_tid}
person_to_d_dict = {'pid': person_d_pid, 'did': person_d_did}

p_to_t = pd.DataFrame(person_to_pt_dict)
p_to_d = pd.DataFrame(person_to_d_dict)

p_to_t.to_csv('data/csv/person_to_pt.csv', index=False)
p_to_d.to_csv('data/csv/person_to_department.csv', index=False)

```







csv文件，例：

pid	did
1	1
2	2
3	2
4	3
5	2
6	3

(三) 构建图数据库

使用neo4j图数据库，注意安装之前需要安装jdk。

在neo4j安装目录下的import文件夹中放入所有csv数据文件。

本地磁盘 (D:) > soft > neo4j-community-3.5.5 > import			搜索"import"
名称	修改日期	类型	
 department.csv	2021/1/3 0:43	XLS 工作表	
 NK.csv	2021/1/3 0:42	XLS 工作表	
 person.csv	2021/1/3 0:43	XLS 工作表	
 person_to_department.csv	2021/1/3 1:19	XLS 工作表	
 person_to_pt.csv	2021/1/3 1:19	XLS 工作表	
 professional_title.csv	2021/1/3 0:43	XLS 工作表	

在浏览器中输入网址<http://localhost:7474/>，进入neo4j的页面，并输入构建数据库的命令。

```
//导入节点 南开大学相关名词及信息 注意类型转换
LOAD CSV WITH HEADERS FROM "file:///NK.csv" AS line
MERGE (n:NK {nid:toInteger(line.nid), name:line.name, info:line.info, buildtime:line.buildtime, leader:line.leader,
        college:line.college, web:line.web, departments:line.departments})

//导入节点 教师信息
LOAD CSV WITH HEADERS FROM 'file:///person.csv' AS line
MERGE (p:Person { pid:toInteger(line.pid), name:line.name, research:line.research})

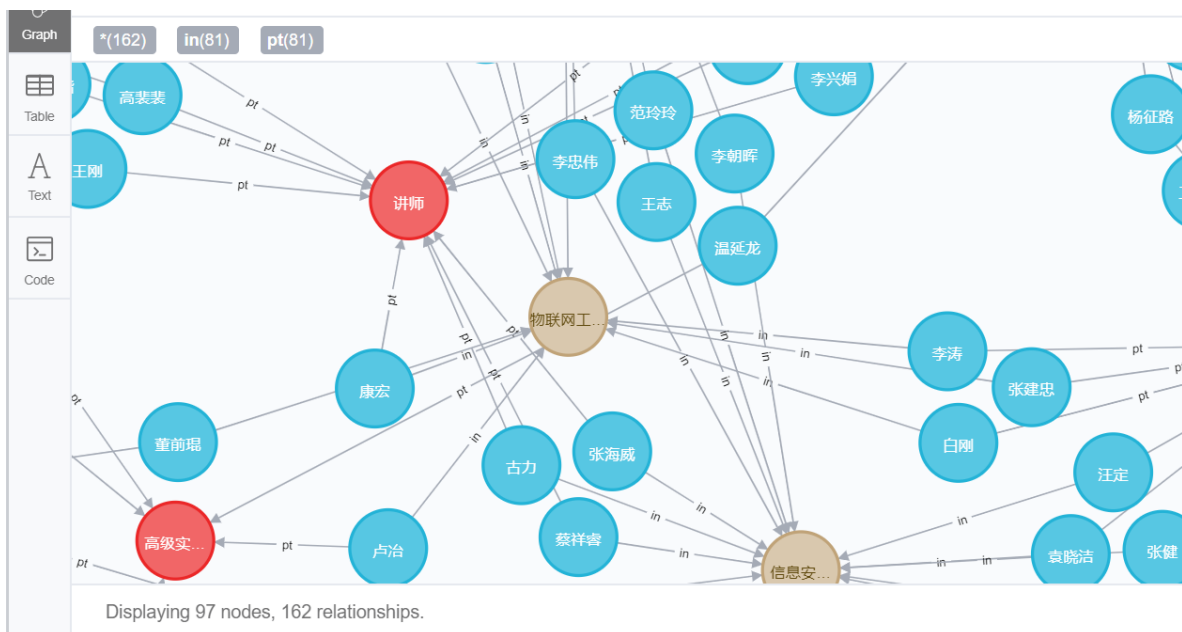
// 导入节点 职称
LOAD CSV WITH HEADERS FROM "file:///professional_title.csv" AS line
MERGE (t:Pt{tid:toInteger(line.tid), name:line.name})

// 导入节点 所属部门
LOAD CSV WITH HEADERS FROM "file:///department.csv" AS line
MERGE (d:Department{did:toInteger(line.did), name:line.name})

// 导入关系 教师有什么职称
LOAD CSV WITH HEADERS FROM "file:///person_to_pt.csv" AS line
match (from:Person{pid:toInteger(line.pid)}),(to:Pt{tid:toInteger(line.tid)})
merge (from)-[r:pt{pid:toInteger(line.pid),tid:toInteger(line.tid)}]->(to)

//导入关系 教师所属哪个部门
LOAD CSV WITH HEADERS FROM "file:///person_to_department.csv" AS line
match (from:Person{pid:toInteger(line.pid)}),(to:Department{did:toInteger(line.did)})
merge (from)-[r:in{pid:toInteger(line.pid),did:toInteger(line.did)}]->(to)
```

数据库构建完毕，可以看到图形化的实体间的关系。



(四) 获取实体名称并赋予自定义属性

从csv文件中获取实体名称，定义南开相关名词的属性为nk，教师名字的属性为pp。同时也可以保证在jieba分词的时候不会被分词。

```
if __name__ == '__main__':
    # 南开相关名词 nk
    nk = pd.read_csv("data/csv/NK.csv")
    # 教师 pp
    person = pd.read_csv("data/csv/person_all.csv")

    nk = nk['name']
    nk = [n.strip() + " 15 nk\n" for n in nk]
```

```
pp = person['name']
pp = [p.strip() + " 15 pp\n" for p in pp]

with open("data/userdict.txt", "w", encoding="utf-8") as fw:
    fw.writelines(nk)
    fw.writelines(pp)
```

txt文件，例：

```
宋春瑶 15 pp
宋起泉 15 pp
沈玮 15 pp
苏明 15 pp
```

（五）构建问题模板和问题分类的训练数据

选取了一些问题，抽取模板，并对每个模板构建训练数据

问题模板：

```
0:nk 简介
1:nk 成立时间
2:nk 领导
3:nk 学院
4:nk 网址
5:nk 职能部门
6:pp 职称
7:pp 所属部门
8:pp 研究方向
```

模板的训练数据，例：

```
pp老师的研究方向
pp的研究方向
pp的研究方向是什么
pp有什么研究方向
pp研究方向
什么是pp的研究方向
```

（六）训练问题模型，并根据模型对用户的问题匹配模板

使用sklearn里的贝叶斯分类器。首先组织训练数据，接着训练多分类贝叶斯分类器模型并返回，最后可以利用训练好的模型来对用户的问题进行分类，返回用户问题所属的类别编号，这个编号也就对应一个问题模板。

```
class QuestionClassify:
    def __init__(self):
        # 读取训练数据
```

```

self.train_x, self.train_y = self.read_train_data()
# 训练模型
self.model = self.train_model_NB()

# 获取训练数据
def read_train_data(self):
    train_x = []
    train_y = []
    file_list = get_file_list("./data/question/")
    # 遍历所有文件
    for one_file in file_list:
        # 获取文件名中的数字
        num = re.sub(r'\D', "", one_file)
        # 如果该文件名有数字，则读取该文件
        if str(num).strip() != "":
            # print(num)
            # 设置当前文件下的数据标签
            label_num = int(num)
            # 读取文件内容
            with(open(one_file, "r", encoding = "utf-8")) as fr:
                data_list = fr.readlines()
                for one_line in data_list:
                    word_list = list(jieba.cut(str(one_line).strip()))
                    # 将这一行加入结果集
                    train_x.append(" ".join(word_list))
                    train_y.append(label_num)
    return train_x, train_y

# 训练并测试模型-NB
def train_model_NB(self):
    x_train, y_train = self.train_x, self.train_y
    self.tv = TfidfVectorizer()

    train_data = self.tv.fit_transform(x_train).toarray()
    clf = MultinomialNB(alpha = 0.01)
    clf.fit(train_data, y_train)

    # 保存成Python支持的文件格式Pickle
    # 保存再读取无法获得self.tv，不行
    # with open('model.pickle', 'wb') as fw:
    #     pickle.dump(clf, fw)
    return clf

# 预测
def predict(self, question):
    question = [" ".join(list(jieba.cut(question)))]
    test_data = self.tv.transform(question).toarray()
    y_predict = self.model.predict(test_data)[0]
    # print("question type:", y_predict)
    return y_predict

```

(七) 定义不同模板的查询

定义不同模板的查询，并根据问题模板来做对应的处理，在知识图谱中查询答案。

具体代码很长，见question_template.py文件，这里贴一个主查询函数和一个模板查询为例。

```

def get_question_answer(self, question, template):
    # 如果问题模板的格式不正确则结束
    assert len(str(template).strip().split("\t")) == 2
    template_id, template_str = int(str(template).strip().split("\t")
[0]), str(template).strip().split("\t")[1]
    self.template_id = template_id
    self.template_str2list = str(template_str).split()

    # 预处理问题
    question_word, question_flag = [], []
    for one in question:
        word, flag = one.split("/")
        question_word.append(str(word).strip())
        question_flag.append(str(flag).strip())
    assert len(question_flag) == len(question_word)
    self.question_word = question_word
    self.question_flag = question_flag
    self.raw_question = question
    # 根据问题模板来做对应的处理，获取答案
    answer = self.q_template_dict[template_id]()
    return answer

# 6:pp 职称
def get_pp_professional_title(self):
    pp_name = self.get_name("pp")
    cq1 = f"match(p:Person)-[r:pt]->(t) where p.name='{pp_name}' return
t.name"
    print(cq1)
    answer = self.graph.run(cq1)[0]
    final_answer = pp_name + "老师的职称是" + str(answer)
    return final_answer

```

(八) 用系统界面展示结果

采用github上的开源模板。

四、结果截图

(一) 测试截图（有详细信息）

```

问题： 南开大学的成立时间
['南开大学/nk', '的/uj', '成立/v', '时间/n']
抽象问题为： nk的成立时间
使用模板编号： 1
问题模板： nk 成立时间
match (n:NK) where n.name='南开大学' return n.buildtime
回答： 南开大学的成立时间为1919年

```

问题： 杨征路老师的职称是什么

['杨征路/pp', '老师/n', '的/uj', '职称/n', '是/v', '什么/r']

抽象问题为： pp老师的职称是什么

使用模板编号： 6

问题模板： pp 职称

match(p:Person)-[r:pt]->(t) where p.name='杨征路' return t.name

回答： 杨征路老师的职称是教授

问题： 袁晓洁老师在什么部门

['袁晓洁/pp', '老师/n', '在/p', '什么/r', '部门/n']

抽象问题为： pp老师在什么部门

使用模板编号： 7

问题模板： pp 所属部门

match(p:Person)-[r:in]->(d) where p.name='袁晓洁' return d.name

回答： 袁晓洁老师的所属部门是信息安全系

(二) 系统界面截图

NKU Question Answering System Based on Knowledge Graph

4:27

计算机学院的成立时间

计算机学院的成立时间为2018年7月

计算机学院有哪些领导

计算机学院的现任领导有党委书记：
黄家友 党委副书记：仇林 院长：袁晓
洁 副院长：刘晓光、张志刚、刘哲
理、仇林

南开大学的学院有哪些

南开大学的学院有文学院、历史学
院、哲学院 外国语学院、法学院、周
恩来政府管理学院 马克思主义学院、
汉语言文化学院、经济学院 商学院、
旅游与服务学院、金融学院 数学科学
学院、物理科学学院、化学学院 生命
科学学院、环境科学与工程学院、医
学院 药学院、电子信息与光学工程学
院、材料科学与工程学院 计算机学
院、网络空间安全学院、人工智能学
院 软件学院、统计与数据科学学院

计算机学院的网址是什么



NKU Question Answering System Based on Knowledge Graph

院、材料科学与工程学院 计算机学院、网络空间安全学院、人工智能学院 软件学院、统计与数据科学学院

计算机学院的网址是什么

计算机学院的网址是
<https://cc.nankai.edu.cn/>

李涛老师的职称是什么

李涛老师的职称是教授

宋春瑶老师在什么部门

宋春瑶老师的所属部门是计算机科学与技术系

什么是杨愚鲁老师的研究方向

杨愚鲁老师的研究方向是并行与分布式处理，系统结构，可重构系统，网格计算

白刚老师的所属部门是什么

白刚老师的所属部门是物联网工程系

