《人工智能程序设计实践》实验大作业报告

大作业题目	基于寻医问药网站的知识图谱构建和可视化			类型	数据爬取和处理
班　号			学　号		
所在院系	计算学部	学　期	22 年秋季学期	任课教师	关毅
实验类型	综合设计型				

实验目的：

- 掌握程序设计的基本算法和简单数据结构基础，能够综合运用基本控制语句、算法和数据结构，以及自顶向下、逐步求精的模块化设计方法，能够设计具有一定规模的系统级python语言程序，提高系统编程能力；
- 掌握常见的XPath方法从网页HTML代码中检索、爬取信息
- 针对复杂的数据类型以及较为庞大的数据，能够使用恰当的算法和数据结构，完成计算、统计分类、检索、匹配等相关的软件系统的构造、测试与实现；
- 掌握常用的程序调试和测试方法。

实验要求：

- 掌握网络爬虫的爬取数据方法；
- 掌握三元组数据格式的存储方法，以及txt、csv、json文件类型的读写操作
- 掌握图数据库（Neo4J）的数据导入以及可视化展示；

实验内容：

- 利用网络爬虫，按照疾病部位爬取寻医问药网站中的疾病相关介绍，以及相关应用知识。
- 将爬取结果存储为三元组格式
- 用图数据库（Neo4J）存储并展示知识图谱

实验环境：

操作系统： Windows11
集成开发环境： Pycharm
外部库： json、requests、re、lxml、pandas、csv

输入输出设计：

- **get_json.py**

  无输入，输出为记录网站上具有索引结构的疾病信息json文件

- **test_关系.py**

  输入get_json输出的json文件，输出关系三元组txt文件

- **test_属性.py**

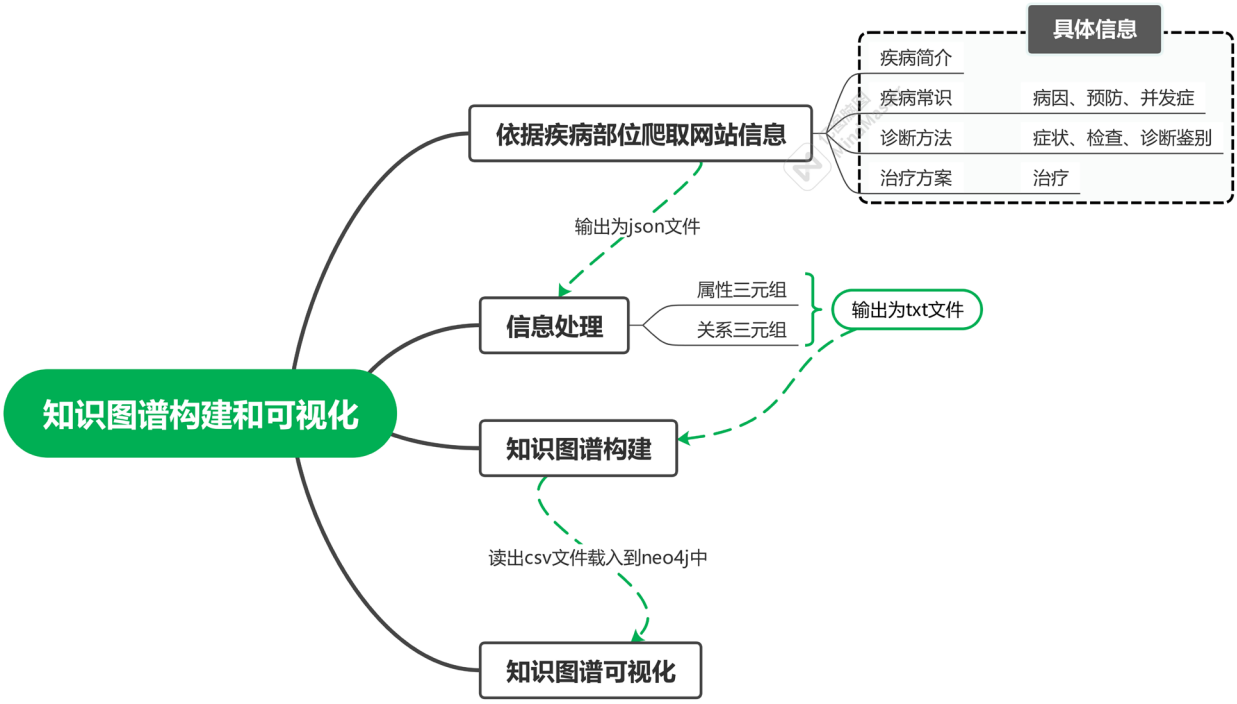
  输入get_json输出的json文件，输出属性三元组txt文件

- **get_node_relation.py**

  输入关系三元组txt文件，输出为已搭建好的知识图谱的结点信息的csv文件

系统设计与实现:

1. 系统功能模块划分

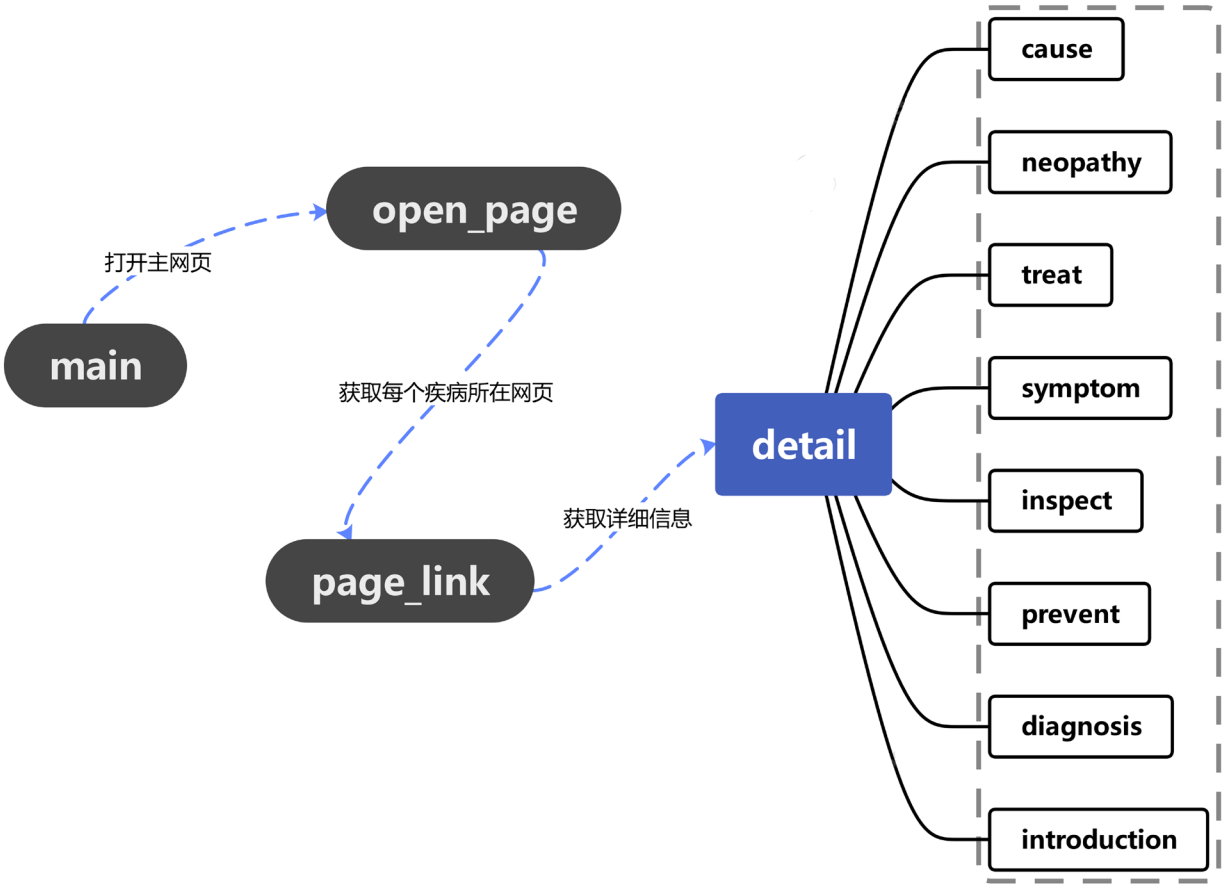


2. 函数功能和外部接口设计

◆ get_json.py ——爬取数据并记录为json文件

序号	函数名	函数功能	函数参数	函数返回值
1	open_page(url)	打开网页	网址url	无
2	introduction(url)	爬取疾病简介	网址url	简介(list)introduction
3	symptom(url)	爬取症状页面	网址url	常见症状(str)sum
4	inspect(url)	爬取检查页面	网址url	检查方式(str)sum
5	diagnosis(url)	爬取诊断鉴别页面	网址url	诊断手段(str)sum
6	treat(url)	爬取治疗页面	网址url	治疗手段(str)sum
7	cause(url)	爬取病因页面	网址url	病因(str)sum
8	prevent(url)	爬取预防页面	网址url	预防方法(str)sum
9	neopathy(url)	爬取并发症页面	网址url	并发症介绍(str)sum
10	extract(res,ser,modi)	正则化提取信息	目标字段res,ser,modi	提取字段列表(list)
11	page_link(res,menu)	提取，拼接网址	提取字段res，目录menu	网址url
12	detail(link)	封装提取所有信息	疾病的特征数字	信息列表(list)content

各个函数之间的调用关系如下所示：



- **test_关系.py ——读取数据为关系三元组，记录为txt文件**

序号	函数名	函数功能	函数参数	函数返回值
1	my_split(data)	以空格分割文本	字符串data	分割后的列表list
2	name(data)	以冒号分割文本，同时提取信息	字符串data	分割后的第一个字符串

- **test_属性.py ——读取数据为属性三元组，记录为txt文件**
- **get_node_relation.py ——读取txt文件，记录为三元组形式的csv文件**

3. 数据结构

列表：利用列表存储网址、疾病关键词等信息。

字典：按照层级关系以及对应关系将相关疾病的关键词键和对应的值储存在多层嵌套字典中。

三元组：依据关系和属性建立知识图谱所需要的三元组数据类型。

4. 算法

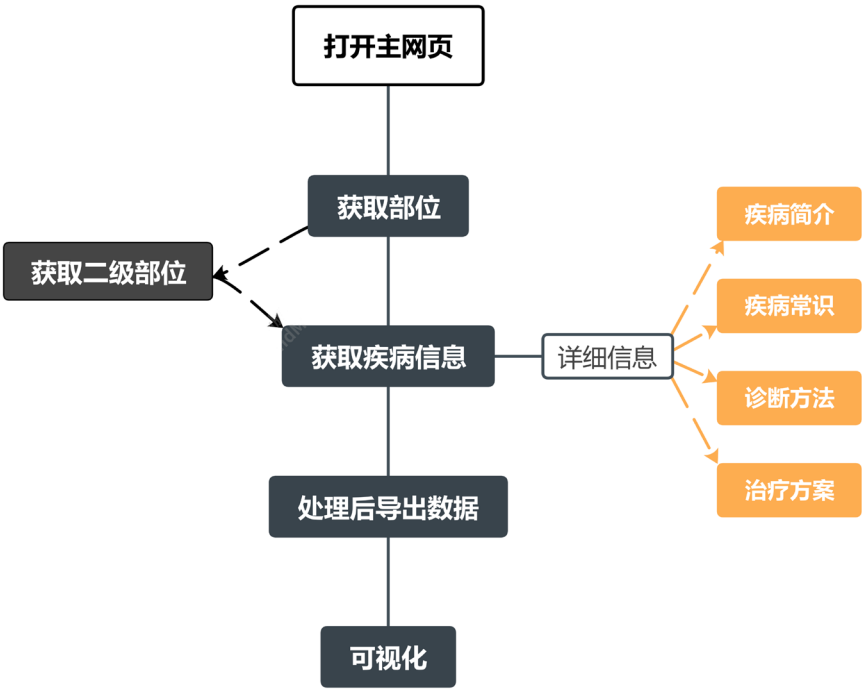
网络爬虫，使用 XPath 方法从网页 HTML 代码中检索、爬取信息。

循环遍历，多重循环遍历，读取、处理、统计爬取得到信息。

文件操作，读写 json、txt、csv 文件，搭建知识图谱。

可视化操作，利用 neo4j 导入文件达到可视化的效果。

5. 程序流程图



实验过程中遇到的问题及解决方法与思路：

问题 1： 爬取网站信息时出现乱码

原因：编码方式不符合网站所使用的编码方式

解决方法：将默认的 utf-8 编码方式更改为网站使用的 gb2312。

问题 2： 爬取信息时出现遗漏、信息不全的情况。

原因：网站 HTML 代码目标目录下子标签不同。

解决方法：使用代码 content=ht.xpath('//div[@class="……"]').xpath('string(.)').strip()，得到某个标签下所有文本，可以忽略 url。

问题 3： 进行字典键、值匹配对应读取时出现不对应的情况

原因：某些关键词对应值为空。

解决方法：增加是否为空的判断。

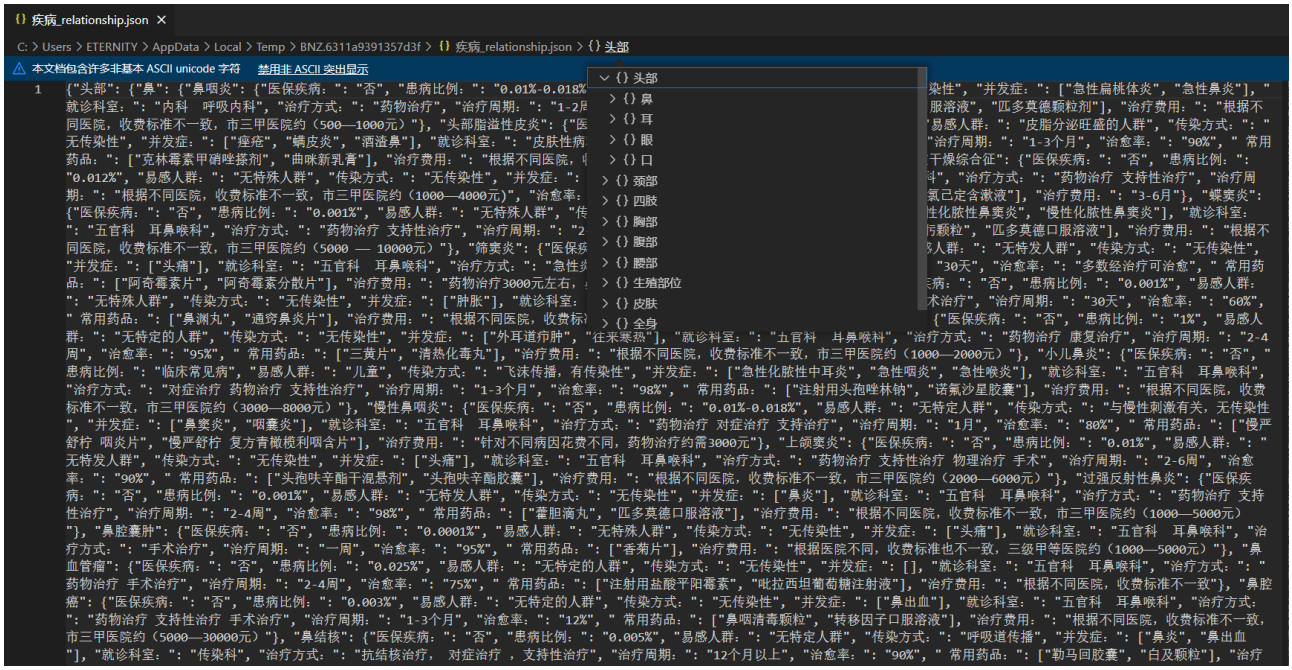
问题 4： 安装 neo4j 时配置不成功

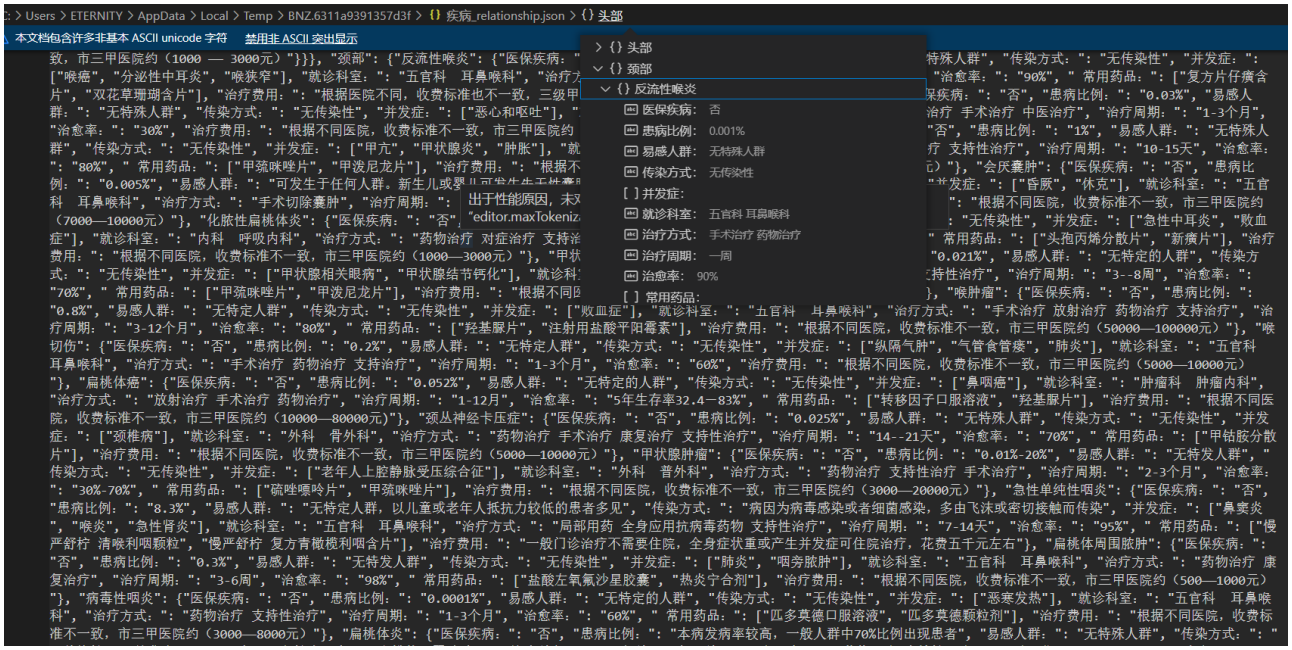
原因：JDK 与 neo4j 版本不适配

解决方法：下载 **JDK-15.0.2** 版本，重新配置环境变量。

测试结果：

Json 文件：


疾病_relationship.json
C: > Users > ETERNITY > AppData > Local > Temp > BNZ.6311a9391357d3f > {} 疾病_relationship.json > {} 头部
本文档包含许多非基本 ASCII unicode 字符
禁用非 ASCII 突出显示
{} 头部
{} 鼻
{} 耳
{} 眼
{} 口
{} 颈部
{} 四肢
{} 胸部
{} 腹部
{} 腰部
{} 生殖部位
{} 皮肤
{} 全身


Users > ETERNITY > AppData > Local > Temp > BNZ.6311a9391357d3f > {} 疾病_relationship.json > {} 头部
本文档包含许多非基本 ASCII unicode 字符
禁用非 ASCII 突出显示
{} 头部
{} 颈部
{} 反流性喉炎
医保疾病: 否
患病比例: 0.001%
易感人群: 无特殊人群
传染方式: 无传染性
[ ] 并发症:
就诊科室: 五官科 耳鼻喉科
治疗方式: 手术治疗 药物治疗
治疗周期: 一周
治愈率: 90%
[ ] 常用药品:

Txt 文件：


属性 - 记事本
文件 编辑 查看
鼻咽炎@@@鼻咽炎简介@@@鼻咽部黏膜、黏膜下和淋巴组织的炎症即为鼻咽炎，分为急性鼻咽炎和慢性鼻咽炎。鼻咽炎病人鼻咽干燥不适，有黏稠分泌物不易咳出，故咳嗽频繁，常伴有恶心，严重者有声嘶、咽痛、头痛、头晕、乏力、消化不良、低热等局部或全身症状。鼻咽部检查见黏膜慢性充血、增生、肥厚，覆以分泌物或干痂。
鼻咽炎@@@温馨提示@@@积极彻底治疗急性鼻炎，可起到预防本病的发生。
鼻咽炎@@@常见症状@@@1、局部临床表现鼻咽干燥不适，有黏稠样分泌物不易咳出，故病人咳嗽频繁常伴有恶心。严重者有声嘶、咽痛、头痛、头晕、乏力、消化不良、低热等全身或局部症状。鼻咽部检查见黏膜慢性充血，增生肥厚，覆以分泌物或干痂。2、全身临床症状鼻咽炎是鼻咽腔受到病毒、细菌的侵扰后引发的炎症。通过检查可发现鼻咽部黏膜水肿、肥大、增生，并伴有分泌物。患者常感到鼻塞、打喷嚏、流清(脓)鼻涕、鼻咽部发痒、干咳痛有异物感、呼吸困难、头昏头痛、乏力、嗅觉减退、记忆力下降等全身性临床症状。
鼻咽炎@@@检查@@@一、局部检查：鼻咽部黏膜呈急性弥漫性充血、肿胀。咽后壁淋巴滤泡隆起，有黏脓自鼻咽部流下。下颌下淋巴结肿大，压痛。慢性患者可见咽侧索红肿，或增生肥厚。二、实验室检查：1、外周血象：急性期病毒性感染时白细胞数正常或偏低。淋巴细胞比例升高。细菌性感染时，白细胞数比例增多和核左移现象。中性粒细胞比例增多和核左移现象。、2、病原学检查：一般情况下不做。必要时可用免疫荧光法、酶联免疫吸附检测法、血清学诊断法或病毒分离和鉴定方法确定病毒的类型。细菌培养和药物敏感试验有助于细菌感染的诊断和治疗。3、间接鼻咽镜或纤维鼻咽镜下易窥见鼻咽部的病变情况，可见鼻咽部干燥充血或红肿，有分泌物。4、胸部X线检查，未见异常。
鼻咽炎@@@诊断鉴别@@@诊断根据病史、症状及体征，本病诊断不难。鉴别诊断1.流行性感冒(1)流行病学：流感为广泛性流行疾病，以冬春季节为主，急性鼻咽炎为全年发生，以秋、冬、春季节多见。(2)临床表现：流感的全身中毒症状明显，常为高热，39～40℃，持续3～5天，头痛、全身疼痛常见且严重，疲乏虚弱早期出现，可持续2～3周，可以伴有鼻塞、喷嚏、咽痛，胸部不适及咳嗽常见，且程度较重，可并发支气管炎、肺炎，甚至可危及生命。气管炎和软腭黏膜颗粒性变化是流感的典型卡他症状。急性鼻咽炎主要表现为鼻塞、喷嚏、咽痛、全身症状较轻，头痛、全身疼痛及疲乏虚弱少且轻微，发热少，部分可有轻到中度胸部不适及咳嗽，并发症少见。2.鼻腔疾病(1)过敏性鼻炎：有过敏史，呈季节性(花粉症)或常年打喷嚏，鼻溢，鼻充血伴瘙痒感。症状特征和鼻分泌物内嗜酸性粒细胞增加有助于本病诊断。(2)血管舒缩性鼻炎：无过敏史，以鼻黏膜间歇性血管充盈、打喷嚏和流清涕为特点，干燥空气能使症状加重。根据病史以及无脓涕和痂皮等可与病毒性或细菌性感染相鉴别。(3)萎缩性鼻炎：鼻腔异常通畅，黏膜固有层变薄且血管减少，嗅觉减退并有痂皮形成及臭味，容易鉴别。(4)鼻中隔偏曲、鼻息肉 鼻镜检查即可明确诊断。3.某些急性传染病如麻疹、脑炎、流行性脑脊髓膜炎、脊髓灰质炎、伤寒、斑疹伤寒和HIV感染前驱期的上呼吸道炎鉴别 根据症状病史 动态观察和相关实验室检查 鉴别不难。
行1，列1
100%
Windows (CRLF)
UTF-8

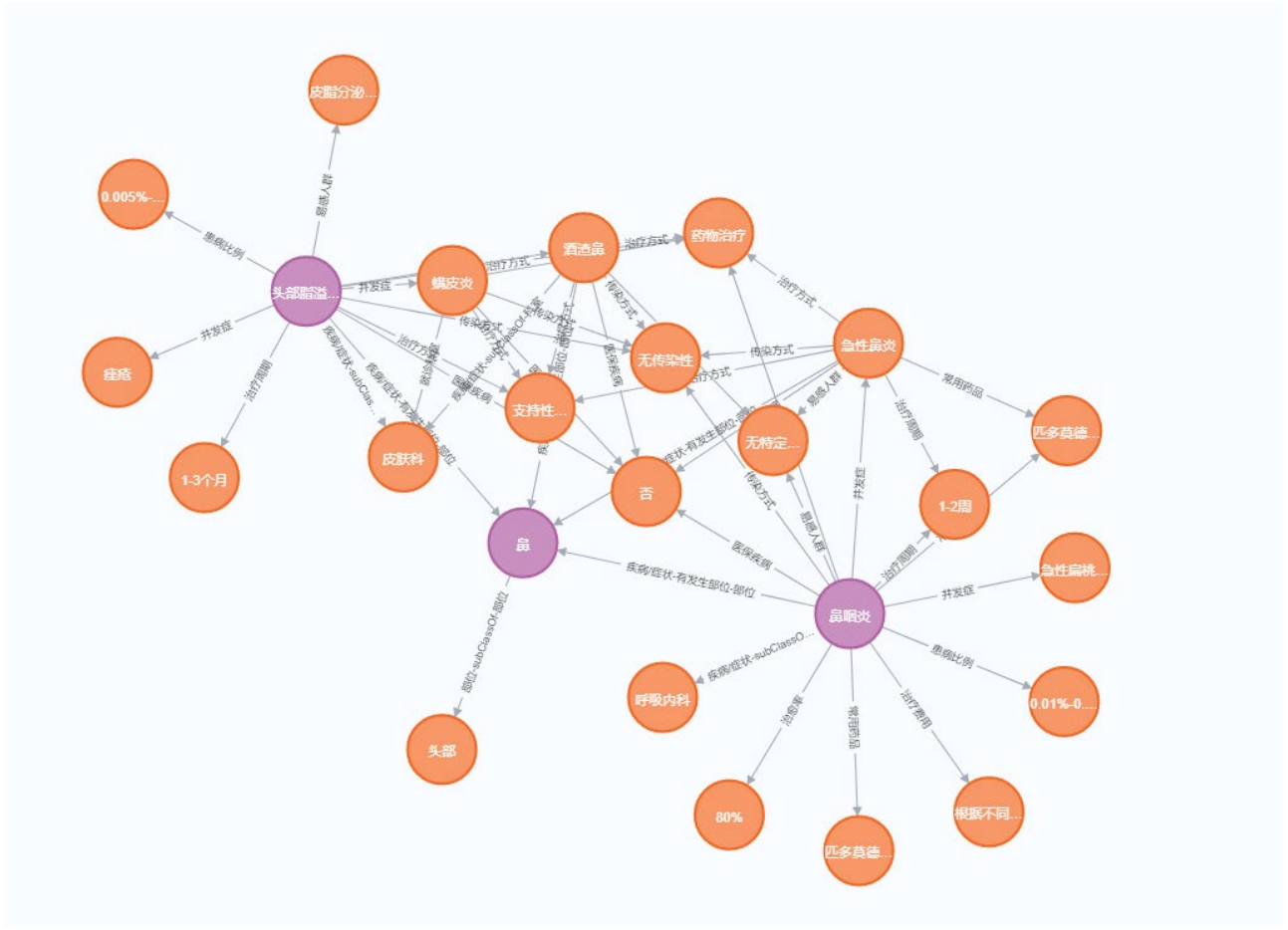

关系 - 记事本
文件 编辑 查看
鼻@@@部位-subClassOf-部位@@@头部
鼻咽炎@@@疾病/症状-有发生部位-部位@@@鼻
鼻咽炎@@@医保疾病@@@否
鼻咽炎@@@患病比例@@@0.01%-0.018%
鼻咽炎@@@易感人群@@@无特定人群
鼻咽炎@@@传染方式@@@无传染性
鼻咽炎@@@并发症@@@急性扁桃体炎
鼻咽炎@@@并发症@@@急性鼻炎
鼻咽炎@@@疾病/症状-subClassOf-科室@@@呼吸内科
鼻咽炎@@@治疗方式@@@药物治疗
鼻咽炎@@@治疗周期@@@1-2周
鼻咽炎@@@治愈率@@@80%
鼻咽炎@@@ 常用药品@@@匹多莫德口服溶液
鼻咽炎@@@ 常用药品@@@匹多莫德颗粒剂
鼻咽炎@@@治疗费用@@@根据不同医院，收费标准不一致，市三甲医院约（500—1000元)
头部脂溢性皮炎@@@疾病/症状-有发生部位-部位@@@鼻
头部脂溢性皮炎@@@医保疾病@@@否
头部脂溢性皮炎@@@患病比例@@@0.005%-0.008%
头部脂溢性皮炎@@@易感人群@@@皮脂分泌旺盛的人群
头部脂溢性皮炎@@@传染方式@@@无传染性
头部脂溢性皮炎@@@并发症@@@痤疮
头部脂溢性皮炎@@@并发症@@@蜡皮炎
行1，列1
100%
Windows (CRLF)
UTF-8

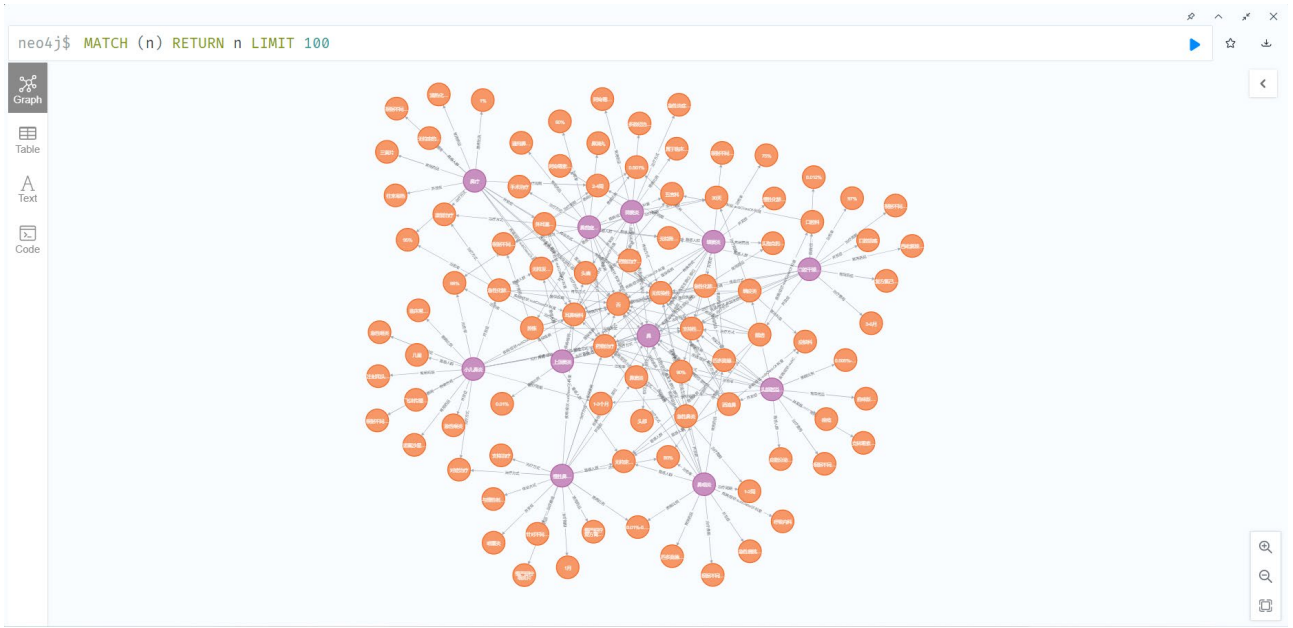
Csv 文件：

```
entity:ID,name,:LABEL
e0,鼻,疾病
e1,头部,tail-type
e2,鼻咽炎,疾病
e3,否,tail-type
e4,0.01%-0.018%,tail-type
e5,无特定人群,tail-type
e6,无传染性,tail-type
e7,急性扁桃体炎,tail-type
e8,急性鼻炎,tail-type
e9,呼吸内科,tail-type
e10,药物治疗,tail-type
e11,1-2周,tail-type
e12,80%,tail-type
e13,匹多莫德口服溶液,tail-type
e14,匹多莫德颗粒剂,tail-type
e15,根据不同医院，收费标准不一致，市三甲医院约（500——1000元）,tail-type
e16,头部脂溢性皮炎,疾病
e17,0.005%-0.008%,tail-type
e18,皮脂分泌旺盛的人群,tail-type
e19,痤疮,tail-type
e20,螨皮炎,tail-type
e21,酒渣鼻,tail-type
e22,皮肤科,tail-type
e23,支持性治疗,tail-type
e24,1-3个月,tail-type
e25,90%,tail-type
e26,克林霉素甲硝唑搽剂,tail-type
```

```
:START_ID,:END_ID,:TYPE
e0,e1,部位-subClassOf-部位
e2,e0,疾病/症状-有发生部位-部位
e2,e3,医保疾病
e2,e4,患病比例
e2,e5,易感人群
e2,e6,传染方式
e2,e7,并发症
e2,e8,并发症
e2,e9,疾病/症状-subClassOf-科室
e2,e10,治疗方式
e2,e11,治疗周期
e2,e12,治愈率
e2,e13,常用药品
e2,e14,常用药品
e2,e15,治疗费用
e16,e0,疾病/症状-有发生部位-部位
e16,e3,医保疾病
e16,e17,患病比例
e16,e18,易感人群
e16,e6,传染方式
e16,e19,并发症
e16,e20,并发症
e16,e21,并发症
e16,e22,疾病/症状-subClassOf-科室
e16,e10,治疗方式
e16,e23,治疗方式
e16,e24,治疗周期
```

Neo4j 可视化：


皮脂分泌...
0.005%-...
患病比例
易感人群
头部脂溢...
并发症
螨皮炎
治疗方式
酒渣鼻
药物治疗
传染方式
痤疮
并发症
治疗周期
1-3个月
皮肤科
支持性...
无传染性
传染方式
急性鼻炎
治疗方式
常用药品
匹多莫德...
无特定...
易感人群
1-2周
治疗周期
舌
传染方式
鼻
医保疾病
急性扁桃...
并发症
鼻咽炎
患病比例
0.01%-0...
呼吸内科
治愈率
常用药品
治疗费用
80%
匹多莫德...
根据不同...
头部


neo4j$ MATCH (n) RETURN n LIMIT 100
Graph
Table
Text
Code

程序的全部源代码：见附录

分析总结、收获和体会:

数据统计：（使用图或表的方法分析关系三元组、属性三元组数据，包括总数、分布情况等）

- 结点共计 34897 个，统计信息包括传染方式、是否医保疾病、就诊科室、常用药品、并发症、患病比例、易感人群、治愈率、治疗周期、治疗方式、治疗费用。
- 数据之间索引类型包括疾病与其所属科室、所属部位的关系，以及疾病属性之间的相互连接。可依据某个特定属性提取子图分析。

优点：

- 功能全面，几乎涵盖上千种疾病的所有信息，覆盖知识面较广
- 封装好，代码的可读性较高，各个变量以及函数见名知意，注释较为详细。
- 结果利用 neo4j 实现可视化，更加清晰明了。
- 使用文件操作及时保存写出数据，提高程序调式的运行速度以及效率。
- 函数变量少，减少了参数的传递，使程序较为简洁。

创新之处：

- 使用网络爬虫，在读取时就加以处理，减少了程序的空间及时间复杂度。
- 利用 neo4j 实现知识图谱的可视化，在索引以及分类上更为直观清晰。

不足之处：

- 未区分部位与疾病，一定程度上分类不明确，且部分函数重复度较高，程序比较冗杂

需要改进的地方：

- 增加分类以及查询算法，后期还可基于人工智能生成自助问答系统等交互性更强的软件

收获与学习体会：

通过两周的程序设计实践，对人工智能基本分类、聚类算法有基本了解与实践，学习了网络爬虫以及知识图谱构建，对专业知识以及数据结构算法有了基本了解，同时代码能力也有所锤炼

自我评价：	是	否
程序运行是否无 bug？	√	
是否在撰写报告之前观看了 spoc 里的代码规范视频？	√	
程序代码是否符合代码规范(对齐与缩进，有必要的注释)？	√	
是否按模块化要求进行了程序设计，系统功能是否完善？	√	
是否是独立完成？		√

自我评语：

熟悉了 python 语言的使用，以及了解的代码的基本格式及基本操作，尝试去锤炼程序的健壮性和可读性。知识面层面，对人工智能常见的分类聚类思想有了了解。

报告完成日期： 2022/9/5

附件：

1.get_json.py

```
import json
import requests
import re
from lxml import html

def open_page(url):
    headers={'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.102 Safari/537.36 Edg/104.0.1293.63'}
    res=requests.get(url,headers=headers)
    res.encoding='gb2312'
    res=res.text.replace('\ufffd','')
    return res
#--------------------------------------------------------------------------------------------------------

def introduction(url):
    data=open_page(url)
    ht = html.fromstring(data)
    introduction= {}

    introduction_key = ht.xpath('//div[@class="jib-articl fr f14 "]/div[{p}]/strong/text()'.format(p=1))[0]
    introduction_value = ht.xpath('//div[@class="jib-articl fr f14 "]/div[{p}]/p/text()'.format(p=1))[0]
    introduction[introduction_key] = introduction_value.strip()

    for i in range(2, 5):
        if i == 2:
            basic_know_att_value = []

            basic_know_key = ht.xpath('//div[@class="jib-articl fr f14 "]/div[{p}]/strong/text()'.format(p=i))[0]
            introduction[basic_know_key] = {}

            basic_know_att_key = ht.xpath('//div[@class="jib-articl fr f14 "]/div[{p}]/p/span[1]/text()'.format(p=i))

            basic_know_att_value_init = ht.xpath('//div[@class="jib-articl fr f14 "]/div[{p}]/p/span[2]/text()'.format(p=i))
```

```
            for value in basic_know_att_value_init:
                value = value.strip()
                basic_know_att_value.append(value)
            if '9144' in url:
                basic_know_att_value.insert(3, ' ')
            ill_all = ht.xpath('//div[@class="jib-articl fr f14 "]/div[{p}]/p/span[2]/a/text()'.format(p=i))
            for i in range(len(ill_all) - 1):
                basic_know_att_value.pop(4)
            basic_know_att_value.append(ill_all)


            for i in range(len(basic_know_att_key)):
                introduction[basic_know_key][basic_know_att_key[i]] = basic_know_att_value[i]

        elif i == 3:
            com_sen_att_value = []

            com_sen_key = ht.xpath('//div[@class="jib-articl fr f14 "]/div[{p}]/strong/text()'.format(p=i))[0]
            introduction[com_sen_key] = {}

            com_sen_att_key = ht.xpath('//div[@class="jib-articl fr f14 "]/div[{p}]/p/span[1]/text()'.format(p=i))

            com_sen_att_value_init = ht.xpath('//div[@class="jib-articl fr f14 "]/div[{p}]/p/span[2]/text()'.format(p=i))
            for value in com_sen_att_value_init:
                value = value.strip()
                com_sen_att_value.append(value)
            medicine = ht.xpath('//div[@class="jib-articl fr f14 "]/div[{p}]/p/span[2]/a/text()'.format(p=i))
            if '9992' in url or '9865' in url or '9987' in url:
                com_sen_att_value.insert(1, ' ')
            for i in range(len(medicine) + 1):
                com_sen_att_value.pop(4)
            com_sen_att_value.insert(4, medicine)

            for i in range(len(com_sen_att_key)):
                introduction[com_sen_key][com_sen_att_key[i]] = com_sen_att_value[i]

        else:
```

```
            tip_key_t = ht.xpath('//div[@class="jib-articl fr f14 "]/div[{p}]/strong/text()'.format(p=i))
            if len(tip_key_t) != 0:
                tip_key = ht.xpath('//div[@class="jib-articl fr f14 "]/div[{p}]/strong/text()'.format(p=i))[0]

                tip_value_init = ht.xpath('//div[@class="jib-articl fr f14 "]/div[{p}]/p/text()'.format(p=i))[0]
                tip_value = tip_value_init.strip()

                introduction[tip_key] = tip_value

    return introduction

def symptom(url):
    data=open_page(url)
    ht = html.fromstring(data)

    members = ht.xpath('//div[@class="jib-articl fr f14 jib-lh-articl"]/p')

    sum = ""
    for member in members:
        member = member.xpath('string(.)').strip()
        member = str(member)
        sum += member


    return "常见症状",sum

def inspect(url):
    data=open_page(url)
    ht = html.fromstring(data)

    members = ht.xpath('//div[@class="jib-articl fr f14 jib-lh-articl"]/p')

    sum = ""
    for member in members:
        member = member.xpath('string(.)').strip()
        member = str(member)
        sum += member


    return "检查",sum

```

```
112.def diagnosis(url):
113.    data=open_page(url)
114.    ht = html.fromstring(data)
115.
116.    members = ht.xpath('//div[@class="jib-articl fr f14 jib-lh-articl"]/p')
117.
118.    sum = ""
119.    for member in members:
120.        member = member.xpath('string(.)').strip()
121.        member = str(member)
122.        sum += member
123.
124.    return "诊断鉴别",sum
125.
126.def treat(url):
127.    data=open_page(url)
128.    ht = html.fromstring(data)
129.
130.    members = ht.xpath('//div[@class="jib-lh-articl"]/p')
131.
132.    sum = ""
133.    for member in members:
134.        member = member.xpath('string(.)').strip()
135.        member = str(member)
136.        sum += member
137.
138.
139.    return "治疗",sum
140.
141.def cause(url):
142.
143.    data=open_page(url)
144.    ht = html.fromstring(data)
145.    members = ht.xpath('//div[@class=" jib-articl fr f14 jib-lh-articl"]')
146.
147.    sum = ""
148.    for member in members:
149.        member = member.xpath('string(.)')
150.        member = str(member.strip())
151.        sum += member
152.
153.    return "病因",member
154.
155.
```

```
def prevent(url):
    data=open_page(url)
    ht = html.fromstring(data)
    members = ht.xpath('//div[@class="jib-articl fr f14 jib-lh-articl"]/p')

    sum = ""
    for member in members:
        member = member.xpath('string(.)').strip()
        member = str(member)
        sum += member
    return "预防",sum

def neopathy(url):
    data=open_page(url)
    ht = html.fromstring(data)
    members = ht.xpath('//div[@class="jib-articl fr f14 jib-lh-articl"]/p')

    sum = ""
    for member in members:
        member = member.xpath('string(.)').strip()
        member = str(member)
        sum += member
    return "并发症",sum

#-------------------------------------------------------------------------------------------------------

def extract(res,ser,modi):
    temp=re.findall(ser,res,re.S)
    return re.findall(modi,str(temp),re.S)

def page_link(res,menu):
    ser_='<a  href="(.*?)">'+menu+'</a>'
    return 'https://jib.xywy.com'+re.findall(ser_,res)[0]

#-------------------------------------------------------------------------------------------------------

def detail(link):
    chai='/il_sii_(.*?).htm'
    feature=(re.findall(chai,link))[0]
    print(feature)
    content=introduction('https://jib.xywy.com/il_sii/gaishu/'+feature+'.htm')
```

```
    title1,value1=symptom('https://jib.xywy.com/il_sii/symptom/'+feature+'.htm')
    content[title1]=value1
    title2,value2=inspect('https://jib.xywy.com/il_sii/inspect/'+feature+'.htm')
    content[title2]=value2
    title3,value3=diagnosis('https://jib.xywy.com/il_sii/diagnosis/'+feature+'.htm')
    content[title3]=value3
    title4,value4=treat('https://jib.xywy.com/il_sii/treat/'+feature+'.htm')
    content[title4]=value4
    title5,value5=cause('https://jib.xywy.com/il_sii/cause/'+feature+'.htm')
    content[title5]=value5
    title6,value6=prevent('https://jib.xywy.com/il_sii/prevent/'+feature+'.htm')
    content[title6]=value6
    title7,value7=neopathy('https://jib.xywy.com/il_sii/neopathy/'+feature+'.htm')
    content[title7]=value7
    return content

# def correct(res,item):
    # ser='<a title="(.*)" href="/il_sii_3865.htm">穿透性角膜移...</a>'
    # #<a title="穿透性角膜移植术所致青光眼" href="/il_sii_3865.htm">穿透性角膜移...</a>
    # r=re.findall(ser,res)
    # if len(r)==0:
    #     return item
    # return r[0]


info={}
url='https://jib.xywy.com/html/toubu.html'
code=open_page(url)
ht = html.fromstring(code)
name = ht.xpath('//li[@class="pr"]/a/text()')
for i in name:
    info[i]={}
    ser_item='>'+i+'</a>\r\n                        <ul class="jbk-sed-menu jb-body-nav bor pa none f12">(.*?)</ul>'
    modi_item='<li><a title="(.*?)" href="(.*?)">'
    items=extract(code,ser_item,modi_item)
    if len(items)==0:
        link=page_link(code,i)
```

```
234.         item_code=open_page(link)
235.         item_ht = html.fromstring(item_code)
236.         items= item_ht.xpath('//ul[@class="ks-ill-list clearfix mt10"]/li/a/
   @title')
237.         href=item_ht.xpath('//ul[@class="ks-ill-list clearfix mt10"]/li/a/@h
   ref')
238.         ix=0
239.         for s in items:
240.             print(ix)
241.             content=detail(href[ix])
242.             info[i][s]=content
243.             ix+=1
244.     else:
245.         for j in items:
246.             info[i][j[0]]={}
247.             link2='https://jib.xywy.com'+j[1]
248.             item2_code=open_page(link2)
249.             item2_ht = html.fromstring(item2_code)
250.             items2=item2_ht.xpath('//ul[@class="ks-ill-list clearfix mt10"]/
   li/a/@title')
251.             href2=item2_ht.xpath('//ul[@class="ks-ill-list clearfix mt10"]/l
   i/a/@href')
252.             idx=0
253.             for s in items2:
254.                 print(idx)
255.                 content=detail(href2[idx])
256.                 info[i][j[0]][s]=content
257.                 idx+=1
258.output_path = "疾病_edit4.json"
259.print(info)
260.with open(output_path, "w",encoding='utf-8') as f:
261.     json.dump(info, f, ensure_ascii=False)
```

2.test_关系.py

```
1. import json
2.
3.
4.
5. def my_split(data):
6.     return data.split(" ")
7.
8.
9. def name(data):
10.     return data.split(": ")[0]
```

```
11.
12.if __name__=='__main__':
13.    output_path = "疾病_edit3.json"
14.    with open(output_path, "r",encoding='utf-8') as f:
15.        data=json.load(f)
16.    with open("test.txt","w",encoding='utf-8') as f2:
17.        menu=['头部','四肢','生殖部位']
18.        mm=[]
19.        for i_1 in data:
20.            if i_1 in menu:
21.                for i_11 in data[i_1]:
22.                    wri=i_11+'@@@部位-subClassOf-部位@@@'+i_1
23.                    f2.write(wri+'\n')
24.                    for i_12 in data[i_1][i_11]:
25.                        w=i_12+'@@@疾病/症状-有发生部位-部位@@@'+i_11
26.                        f2.write(w+'\n')
27.                        for item in data[i_1][i_11][i_12].keys():
28.                            if item=='治疗费用: ':
29.                                w=i_12+'@@@治疗费用
   @@@'+data[i_1][i_11][i_12][item]
30.                                f2.write(w+'\n')
31.                                continue
32.                            n=name(item)
33.                            if False==isinstance(data[i_1][i_11][i_12][item],
   list):
34.                                re=my_split(data[i_1][i_11][i_12][item])
35.                                if item=='就诊科室: ':
36.                                    if len(re)==1:
37.                                        wde=i_12+'@@@疾病/症状-subClassOf-科室
   @@@'+re[0]
38.                                        f2.write(wde+'\n')
39.                                    else:
40.                                        wde=i_12+'@@@疾病/症状-subClassOf-科室
   @@@'+re[2]
41.                                        f2.write(wde+'\n')
42.                                        me=re[2]+'@@@科室-subClassOf-科室
   @@@'+re[0]
43.                                        if me not in mm:
44.                                            mm.append(me)
45.                                    continue
46.                                for de in re:
47.                                    wde=i_12+'@@@'+n+'@@@'+de
48.                                    f2.write(wde+'\n')
49.                            else:
```

```
                                    for de in data[i_1][i_11][i_12][item]:
                                        wde=i_12+'@@@'+n+'@@@'+de
                                        f2.write(wde+'\n')
            else:
                for i_2 in data[i_1]:
                    wri=i_2+'@@@疾病/症状-有发生部位-部位@@@'+i_1
                    f2.write(wri+'\n')
                    for item in data[i_1][i_2].keys():
                        if item=='治疗费用: ':
                            w=i_2+'@@@治疗费用@@@'+data[i_1][i_2][item]
                            f2.write(w+'\n')
                            continue
                        n=name(item)
                        if False==isinstance(data[i_1][i_2][item],list):
                            re=my_split(data[i_1][i_2][item])
                            if item=='就诊科室: ':
                                if len(re)==1:
                                    wde=i_2+'@@@疾病/症状-subClassOf-科室
@@@'+re[0]
                                    f2.write(wde+'\n')

                                else:
                                    wde=i_2+'@@@疾病/症状-subClassOf-科室
@@@'+re[2]
                                    f2.write(wde+'\n')
                                    me=re[2]+'@@@科室-subClassOf-科室
@@@'+re[0]
                                    if me not in mm:
                                        mm.append(me)
                                continue
                            for de in re:
                                wde=i_2+'@@@'+n+'@@@'+de
                                f2.write(wde+'\n')
                        else:
                            for de in data[i_1][i_2][item]:
                                wde=i_2+'@@@'+n+'@@@'+de
                                f2.write(wde+'\n')
        for it in mm:
            f2.write(it+'\n')
```

3. get_node_relation.py

```
#读取 txt 文件中的三元组
#将节点去重复，写入 csv 文件
#将关系写入 csv 文件
```

```
4.
5. import pandas as pd
6.
7. import csv
8. # 读取三元组文件
9.
10.entity_list = []
11.tri_list = []
12.
13.entity_dict = {}
14.ent_num = 0
15.with open("关系.txt", "r", encoding="utf-8") as f:
16.    for line in f:
17.        line = line.strip().split('@@@')
18.        head = line[0].strip()
19.        rel = line[1].strip()
20.        tail = line[2].strip()
21.        if head not in entity_dict:
22.            entity_dict[head] = 'e' + str(ent_num)
23.            entity_list.append(['e' + str(ent_num), head, "疾病"])
24.            ent_num += 1
25.        if tail not in entity_dict:
26.            entity_dict[tail] = 'e' + str(ent_num)
27.            entity_list.append(['e' + str(ent_num), tail, "tail-type"])
28.            ent_num += 1
29.        #添加三元组，使用实体 id 和关系 id
30.        if [entity_dict[head], entity_dict[tail], rel] not in tri_list:
31.            tri_list.append([entity_dict[head], entity_dict[tail], rel])
32.
33.#将去重复后的实体写入 csv 文件
34.
35.entity_header = ["entity:ID", "name", ":LABEL"]
36.with open ('entity.csv','w',encoding='utf-8',newline='') as fp:
37.    # 写
38.    writer =csv.writer(fp)
39.    # 设置第一行标题头
40.    writer.writerow(entity_header)
41.    # 将数据写入
42.    writer.writerows(entity_list)
43.
44.#将去重复后的三元组写入 csv
45.tri_header =  [':START_ID', ':END_ID', ':TYPE']
46.with open ('triplets.csv','w',encoding='utf-8',newline='') as fp:
47.    # 写
```

```
48.     writer =csv.writer(fp)
49.     # 设置第一行标题头
50.     writer.writerow(tri_header)
51.     # 将数据写入
52.     writer.writerows(tri_list)
53.#h_r_t_name = [":START_ID", "role", ":END_ID"]
54.h_r_t = pd.read_csv("triplets.csv", decimal="\t", names=tri_header)
55.print(h_r_t.info())
```