Python 程序设计课程实验报告

实验 1　设计一个程序抽取百度百科人物知识

姓　　名：李佩石

学　　号：22920212204129

院　　系：信息学院 人工智能系

专　　业：人工智能

年　　级：2021 级

指导教师：曹冬林

2023 年 3 月 24 日

一.实验任务与思路

任务:

网址：https://baike.baidu.com/#home

需要抽取下列信息：

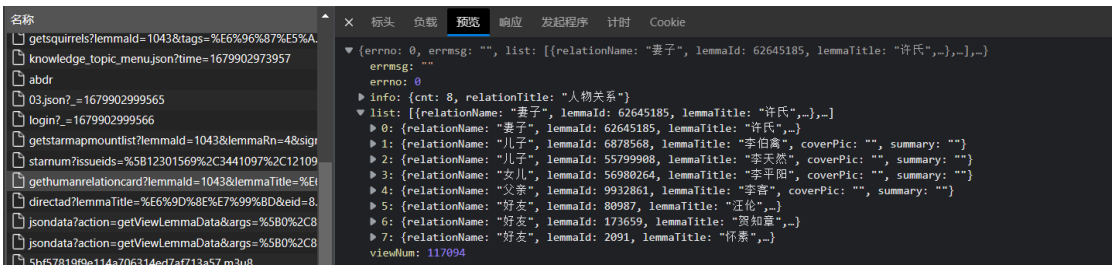
简介、基本信息、关联人物、人物履历

思路:

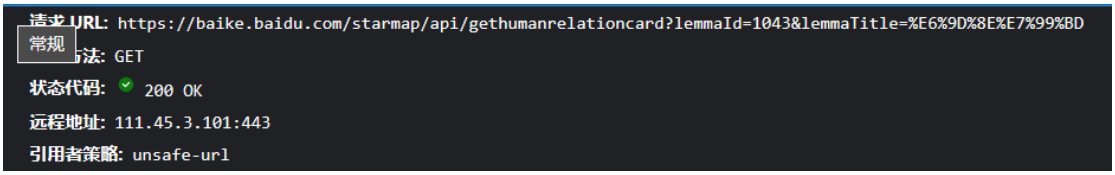
用一个函数,分成四部分,分别爬取简介、基本信息、关联人物、人物履历,并处理文本后,分段写入 txt 文件中

难点:人物关系为 js 渲染的动态网页元素,并不在网页 HTML 源码中,无法通过爬取匹配 HTML 源码获得

解决:应用 bf4 库,通过查看目标网站网页格式,审查元素,依次点击 XHR、JS 下的所有包，通过 Preview 来预览当前包的响应信息，寻找我们想要的结果,如:



，通过 Headers 找到了 Request URL，这才是我们真正应该请求的路径,如:



则构造

```
headers = {
    'User-Agent ': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36 Edg',
    "Referer": url
}

# 损API链接的查询实符串传给Params参数
params = {
    'lemmaId': a,
    'lemmaTitle': content
}
res = requests.get('https://baike.baidu.com/starmap/api/gethumanrelationcard', headers=headers, params=params)
```

Lemmid,lemma title 对应

```
lemmaId=1043&lemmaTitle=%E6%9D%8E%E7%99%BD
```

则需要获得 lemmaid ,查看网页源码

```
<link rel="alternate" hreflang="x-default" href="https://baike.baidu.com/item/%E6%9D%8E%E7%99%BD/1043" />
```

可由 beautifulsoup 解析获得完整 url,

```
url = 'https://baike.baidu.com/item/' + parse.quote(content)

r = requests.get(url)  # 获取目标网址所有信息
demo = r.text  # 定义所有信息的文本
soup = BeautifulSoup(demo, 'html.parser')  # BeautifulSoup中的方

a = soup.find('link', rel="alternate", hreflang="x-default")
a = a.get('href')
a = a.split('/')[-1]

headers = {
    'User-Agent ': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) A
    "Referer": url
}

# 损API链接的查询实符串传给Params参数
params = {
    'lemmaId': a,
    'lemmaTitle': content
}
```

初步输出可见

{'relationName': '妻子', 'lemmaId': 62645185, 'lemmaTitle': '许氏', 'coverPic': 'https://bkimg.cdn.bcebos.com/smart/08f790529822720e0cf3303b449d1d46f21fbf097de1-bkimg-process,
{'relationName': '儿子', 'lemmaId': 6878568, 'lemmaTitle': '李伯鸾', 'coverPic': '', 'summary': ''}
{'relationName': '儿子', 'lemmaId': 55799908, 'lemmaTitle': '李天然', 'coverPic': '', 'summary': ''}
{'relationName': '女儿', 'lemmaId': 56980264, 'lemmaTitle': '李平阳', 'coverPic': '', 'summary': ''}
{'relationName': '父亲', 'lemmaId': 9932861, 'lemmaTitle': '李客', 'coverPic': '', 'summary': ''}
{'relationName': '好友', 'lemmaId': 80987, 'lemmaTitle': '汪伦', 'coverPic': 'https://bkimg.cdn.bcebos.com/smart/622762d0f703918f00b98b645a3d269759eec430-bkimg-process,v_1,rw_
{'relationName': '好友', 'lemmaId': 173659, 'lemmaTitle': '贺知章', 'coverPic': 'https://bkimg.cdn.bcebos.com/smart/f9dcd100baa1cd117738f2d4b212c8fcc2ce2db3-bkimg-process,v_1,
{'relationName': '好友', 'lemmaId': 2091, 'lemmaTitle': '怀素', 'coverPic': 'https://bkimg.cdn.bcebos.com/smart/dcc451da81cb39db898ac7f9d6160924aa1830d2-bkimg-process,v_1,rw_1

再次数据处理后,

```
a=[]
for x in items['list']:
    s = x['relationName'] + ':' + x['lemmaTitle']
    print(s)
    a.append(s)
```

输出得

妻子:许氏
儿子:李伯鸾
儿子:李天然
女儿:李平阳
父亲:李客
好友:汪伦
好友:贺知章
好友:怀素

即解决了动态元素的爬取,重点即在 xhr 数据包的分析,标头的分析与关键部分的获得与对应构造

二.程序实现

1.代码

```
import re

import requests

import urllib
```

```
from urllib import request
from urllib import parse
from bs4 import BeautifulSoup
from lxml import etree


def queryBaidubaike(content):
    #爬取基本信息
    url = 'https://baike.baidu.com/item/' + urllib.parse.quote(content)
    # 重构请求头
    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:6.0) Gecko/20100101 Firefox/6.0'}

    req = requests.get(url)
    req.encoding = "utf-8"
    html = req.text
    soup = BeautifulSoup(req.text, features="html.parser")
    company_items = soup.find_all("div", class_="basic-info J-basic-info cmn-clearfix")
    a = []
    for i in company_items:
        x = i.text.strip()
```

```
        a.append(x)
    a = "".join(a)
    a = "".join(a.split('\xa0'))
    a = "\n".join(a.split('\n\n'))
    a = re.sub(u"\\[.*?]", "", a)
    a = "\n".join(a.split('\n\n'))
    a = "\n".join(a.split('\n\n'))

    fo = open(content + ".txt", "w", encoding='utf-8')
    fo.write('基本信息\n\n')
    fo.write(a)
    fo.close()


    #爬取关系
    url = 'https://baike.baidu.com/item/' + parse.quote(content)

    r = requests.get(url)  # 获取目标网址所有信息
    demo = r.text  # 定义所有信息的文本
    soup = BeautifulSoup(demo, 'html.parser')  # BeautifulSoup 中的方
法
```

```
        a = soup.find('link', rel="alternate", hreflang="x-default")
        a = a.get('href')
        a = a.split('/')[-1]

        headers = {
            'User-Agent ': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36 Edg',
            "Referer": url
        }

        # 损 API 链接的查询实符串传给 Params 参数
        params = {
            'lemmaId': a,
            'lemmaTitle': content
        }
        res = requests.get('https://baike.baidu.com/starmap/api/gethumanrelationcard', headers=headers, params=params)

        # 返回结果力 JSQN 格式,调用 json.()方法解析
        items = res.json()
```

```
a = []
for x in items['list']:
        s = x['relationName'] + ':' + x['lemmaTitle'] + '\n'
        a.append(s)
a = "".join(a)
fo = open(content+".txt", "a+", encoding='utf-8')
fo.write('\n\n 人物关系\n\n')
fo.write(a)
fo.close()



#爬取简介
# 请求地址
url = 'https://baike.baidu.com/item/' + urllib.parse.quote(content)
# 请求头部
headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) '
                                'AppleWebKit/537.36 (KHTML, like Gecko) '
                                'Chrome/67.0.3396.99 Safari/537.36'
}
# 利用请求地址和请求头部构造请求对象
req = urllib.request.Request(url=url, headers=headers, method='GET')
```

```
        # 发送请求，获得响应
        response = urllib.request.urlopen(req)
        # 读取响应，获得文本
        text = response.read().decode('utf-8')
        # 构造 _Element 对象
        html = etree.HTML(text)
        # 使用 xpath 匹配数据，得到匹配字符串列表
        sen_list = html.xpath('//div[contains(@class,"lemma-summary") or
contains(@class,"lemmaWgt-lemmaSummary")]//text()')

        # 过滤数据，去掉空白
        sen_list_after_filter = [item.strip('\n') for item in sen_list]
        # 将字符串列表连成字符串并返回

        b= ''.join(sen_list_after_filter)
        b=re.sub(u"\\[.*?]","",b)
        fo=open(content+".txt","a+", encoding='utf-8')
        fo.write('\n\n 人物简介\n\n')
        fo.write(b)
        fo.close()
```

```
        #爬取履历
        sen_list = html.xpath('//div[contains(@class,"para")]//text()')

        # 过滤数据，去掉空白
        sen_list_after_filter = [item.strip('\n') for item in sen_list]
        # 将字符串列表连成字符串并返回

        a =''.join(sen_list_after_filter)
        a = re.sub(u"\\[.*?]", "", a)

        a=a.replace(b,'',1)
        fo = open(content + ".txt", "a+", encoding='utf-8')
        fo.write('\n\n 人物履历:\n\n')
        fo.write(a)
        fo.close()


content=input('请输入:')
queryBaidubaike(content)
```

2.运行结果:

三.实验总结

本次实验,使我加深对已学过的 python 知识的理解与拓展应用,深刻感受到 python 的便捷与灵活,以及与自然语言的相近.

还提前学习了解了 python 关于文件的读写操作

此外,还了解并掌握了爬虫的基本流程,基本操作,以及实际应用了几种爬取方式,解决了动态元素的爬取与目标字段的匹配问题,并且学会写入文件前对文本进行尽可能的处理使之易于阅读,去除多余符号以及处理排版杂乱的问题