

Exp2 实验报告

袁雨 PB20151804

一、实验目的

从以下两个实验任意选择一项完成

1. 豆瓣网站<https://movie.douban.com>的电影详细信息爬取
2. POJ网站<http://poj.org/problemset>的题目详细信息爬取

二、实验要求

Part1

给定网站: <https://movie.douban.com>, 需要设计一个网站遍历策略, 爬取每部电影的相关信息, 记录于 json 文件中。部分信息标于红框中:

← → ↺ movie.douban.com/subject/34801038/?tag=热门&from=gaia

豆瓣 读书 电影 音乐 同城 小组 阅读 FM 时间 豆品

豆瓣电影 搜索电影、电视剧、综艺、影人

影讯&购票 选电影 电视剧 排行榜 分类 影评 2021年度榜单 2021书影音报告

2021 豆瓣年度电影榜单

黑客帝国：矩阵重启 The Matrix Resurrections (2021)



导演: 拉娜·沃卓斯基
编剧: 拉娜·沃卓斯基 / 大卫·米切尔 / 亚历山大·赫蒙 / 莉莉·沃卓斯基
主演: 基努·里维斯 / 凯瑞-安·莫斯 / 叶海亚·阿卜杜勒-迈丁 / 乔纳森·格罗夫 / 杰西卡·亨维克 / 更多...
类型: 动作 / 科幻
官方网站: thechoiceisyours.whatisthematrix.com
制片国家/地区: 美国
语言: 英语
上映日期: 2022-01-14(中国大陆) / 2021-12-22(美国)
片长: 148分钟 / 147分钟(中国大陆)
又名: 22世纪杀人网络: 复活次元(港) / 骇客任务: 复活(台) / 黑客帝国4: 矩阵重生 / 骇客帝国4 / 骇客任务4 / 黑客帝国: 复兴
IMDb: [tt10838180](https://www.imdb.com/title/tt10838180)

豆瓣评分
5.7 ★★★★★
85536人评价

5星 3.9%
4星 15.0%
3星 48.4%
2星 27.1%
1星 5.7%

好于 26% 科幻片
好于 21% 动作片

样例数据:

```
{
    "片名": "黑客帝国:矩阵重启 The Matrix Resurrections",
    "导演": "拉娜·沃卓斯基",
    "编剧": ["拉娜·沃卓斯基", "大卫·米切尔", "亚历山大·赫蒙", "莉莉·沃卓斯基"],
    "主演": ["基努·里维斯", "凯瑞-安·莫斯", "叶海亚·阿卜杜勒-迈丁", "乔纳森·格罗夫", "杰西卡·亨维克"],
    "类型": ["动作", "科幻"],
    "官方网站": "thechoiceisyours.whatisthematrix.com",
    "制片国家/地区": "美国",
    "语言": "英语",
    "上映日期": ["2022-01-14(中国大陆)", "2021-12-22(美国)"],
    "片长": ["148分钟", "147分钟(中国大陆)"],
    "评分": 5.7
}
```

Part2

在Part1爬取文本信息的基础上，爬取每部电影对应的图片（红框所示），保存在文件夹中。

注意事项

1. 每位同学爬取至少100部电影的信息，电影种类不限
2. 保存到json文件的python代码，供参考（sample 即为你解析得到的一个网页的数据字典）

```
import json

for url in urls:
    sample = get_obj(url)

    file = open('result.json', 'a', encoding='utf8')
    file.write(json.dumps(sample, ensure_ascii=False))
    file.write('\n')
    file.close()
```

3.图片文件命名规则

以对应的电影名称命名：电影名称_计数.jpg/jpeg/png

如黑客帝国：矩阵重启 The Matrix Resurrections_3.jpg/jpeg/png

图片单独存放在一个文件夹里。


名称

-  阿甘正传 Forrest Gump_1.jpg
-  霸王别姬_2.jpg
-  黑客帝国：矩阵...rrections_3.jpg
-  美丽人生 La vita è bella_4.jpg
-  千与千寻 千と千尋の神隠し_7.jpg
-  泰坦尼克号 Titanic_5.jpg
-  辛德勒的名单 S...dler's List_8.jpg
-  这个杀手不太冷 Léon_6.jpg

三、实验思路

1. 准备工作

通过浏览器查看分析目标网页。

 <https://movie.douban.com/top250>

 <https://movie.douban.com/top250?start=25>

.....

 <https://movie.douban.com/top250?start=225>

发现页面包括250条电影数据，分10页，每页25条。每页的URL不同之处：最后的数值=（页数-1）*25。

```
baseurl = "https://movie.douban.com/top250?start="
for i in range(0, 4): # 调用获取页面信息的函数 4次
    url = baseurl + str(i*25)
    html = askURL(url) # 保存获取到的网页源码
```

借助Chrome开发者工具来分析网页，在Elements下找到需要的数据位置。
则可通过先爬取每部影片的详情链接，再爬取详情链接里的具体信息。

2. 获取数据

定义一个获取页面的函数askURL,传入一个url参数表示网址。对每一个页面，调用askURL函数来获取页面内容。

urllib.Request生成请求；urllib.urlopen发送请求获取响应，read获取页面内容。

在访问页面时有时会出错，为保证程序的正常运行，加入异常捕获try...except...语句。

```
def askURL(url):
    # 用户代理，表示告诉豆瓣服务器，我们是什么类型的机器、浏览器（本质上是告诉浏览器，我们可以接收什么水平的文件内容）
    # 模拟浏览器头部信息，向豆瓣服务器发送消息
    head = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74 Safari/537.36'}
    request = urllib.request.Request(url, headers=head)
    html = ""
    try:
        response = urllib.request.urlopen(request)
        html = response.read().decode("utf-8")
```

```

    # print(html)
except urllib.error.URLError as e:
    if hasattr(e, "code"):
        print('1')
        print(e.code)
    if hasattr(e, "reason"):
        print('2')
        print(e.reason)
return html

```

3. 解析内容

对爬取的html文件进行解析。借助Chrome开发者工具来分析网页后，选择BeautifulSoup定位特定的标签位置，使用正则表达式找到具体的内容。将寻找规则定义为全局变量。详见源代码中的getData函数。这里列举一些解析过程中遇到的问题：

1. 某些属性所含的项不止一个，则以列表形式存储。以编剧为例：

```

findWriter = re.compile(r'<a href="/celebrity/(\d*)/">(.*?)</a>')
writer = re.findall(findWriter, content)
writers = []
for i in range(len(writer)):
    writers.append(writer[i][1])
    data.append(writers)

```

2. 有的属性某些影片含有，某些影片不含有。则用if语句进行判断，不含有的留空。以官方网站为例：

```

findWeb = re.compile(r'<a href="(.*?)" rel="nofollow" target="_blank">(.*?)</a>')
web = re.findall(findWeb, content)
if len(web) != 0:
    data.append(web[0][0])
else:
    data.append(" ")

```

3. 有些数据中含有"/"， "< br/>"等，则用re.sub()函数替换，然后去掉前后的空格。以片长为例：

```

findRuntime = re.compile(r'<span class="pl">片长:</span> <span content="(.*?)" property="v:runtime">(.*?)</span>(.*?)')
runtime = re.findall(findRuntime, content)
runtimes = []
runtimes.append(runtime[0][1])
if runtime[0][2] != '<br/>':
    otherRuntime = runtime[0][2]

```

```

        otherRuntime = re.sub('<br/>', " ", otherRuntime) # 去掉<br/>
        otherRuntime = re.sub('/', " ", otherRuntime) # 替换/
    runtimes.append(otherRuntime.strip()) #去掉前后的空格
data.append(runtimes)

```

4. 保存数据

将每个网页的数据以字典方式存储，利用json.dumps()函数转化为json格式。

```

def saveData(datalist):
    for i in range(len(datalist)):
        sample = {}
        sample["片名"] = datalist[i][0]
        sample["导演"] = datalist[i][1]
        sample["编剧"] = datalist[i][2]
        sample["主演"] = datalist[i][3]
        sample["类型"] = datalist[i][4]
        sample["官方网站"] = datalist[i][5]
        sample["制片国家/地区"] = datalist[i][6]
        sample["语言"] = datalist[i][7]
        sample["上映日期"] = datalist[i][8]
        sample["片长"] = datalist[i][9]
        sample["评分"] = datalist[i][10]
        # for key, value in sample.items(): # 测试
        #     print("%s:%s" % (key, value))
        file = open('result.json', 'a', encoding='utf8')
        file.write(json.dumps(sample, ensure_ascii=False))
        file.write('\n')
        file.close()

```

5.保存图片

爬取并解析图片的路径，直接保存图片。注意windows系统中不允许文件名中包含某些字符，对其进行过滤。



指环王3 王者无敌
The Lord of the
Rings The Return
of the
King_74.jpg

文件名不能包含下列任何字符:
\\/:*?"<>|

```

def saveImg(datalist):

```

```

saveDir = "./movie_poster/" # 保存电影海报的路径，方便修改
for i in range(len(datalist)):
    img_src = datalist[i][1]
    path_list = re.sub(r'[:](:)(\.)', "", datalist[i][0]) # 过滤特殊字符
    a = '_' + str(i+1) + '.jpg'
    print(saveDir + path_list + a)
    objPath = saveDir + path_list + a
    isExists = os.path.exists(saveDir)
    if not isExists: # 如果不存在
        os.mkdir(saveDir) # 创建文件夹
    img = urllib.request.urlopen(img_src)
    with open(objPath, "ab") as f:
        f.write(img.read())

```

6. 反爬虫方法

1. 设置headers，模拟浏览器头部信息，向豆瓣服务器发送消息。

```

head = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74 Safari/537.36'}
request = urllib.request.Request(url, headers=head)

```

2. 设置time.sleep()

```

t = random.random() # 随机大于0 且小于1 之间的小数
time.sleep(t)

```

3. 更换ip地址。