爬虫：知乎用户信息

PB17061266 孙书情

1.完成情况：

1. 只完成了第一部分的一些，未实现第二部分。
2. 可以搜索出1000个用户的`userName,userLink,headline,sideItem,following_count,follower_count`但不能搜索出具体的following和follower的url，原因在于知乎的反爬虫。

2.代码思路：

1. 代码思路还是很简单的，根据一个种子用户，获取种子用户的follow者，然后再获取种子用户关注者的follow者，进而爬取1000个用户，但事实上，以我的代码思路如果要发现所有follower以及following的url，就不止是只爬取1000个数据了，应该是指数级的用户，因此在该lab中我只实现了调取关注着和被关注者的数量。
2. 主要应用了mongo数据库。
3. code部分共3个文件，其中有两个都是用于数据处理的，主要代码在scraw.py中:
    1. 我首先设置了访问知乎用户网页获取信息的http格式：

        `https://www.zhihu.com/api/v4/members/zhouyuan/followees?include=data%5B*%5D.answer_count%2Carticles_count%2Cgender%2Cfollowing_count%2Cfollower_count%2Cis_followed%2Cis_following%2Cbadge%5B%3F(type%3Dbest_answerer)%5D.topics&offset=0&limit=20`

        可以看到我设置了对用户名，用户token等信息的询问，得到对应用户的有关信息，json格式，该json中的data部分是对于该用户每个follower的信息描述，也是我们进行迭代的关键。

```
def get_page(url):
    header = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) '
                      'Chrome/69.0.3497.12 Safari/537.36 '
    }
    response = requests.get(url, headers=header)
    return response.json()
```

    2. 对html中的json文件进行分析：数据存入mongo数据库zhihu_user_network中

```
def parse(html):
```

```
    print(html)
    if('data' not in html):
        return
    items = html['data']
    i=0
    for item in items:
        i=i+1
        name = item['name']
        url_token = item['url_token']
        headline = item['headline']
        badge = item['badge']
        if(len(badge)>0):
            sideItem = badge[0]['description']
        else:
            sideItem = 0
        follower_count = item['follower_count']
        answer_count = item['answer_count']
        gender = item['gender']
        url = 'https://www.zhihu.com/api/v4/members/' +
str(url_token) + '/followees?include=data%5B*%5D.answer_count' \

    '%2Carticles_count%2Cgender%2C2Cfollowing_count%2Cfollower_count'
\

    '%2Cis_followed%2Cis_following%2Cbadge%5B%3F(' \

    'type%3Dbest_answerer)%5D.topics&offset=0' \

    '&limit=20 '
        info = {
            'name': name,
            'url_token':url_token,
            'headline':headline,
            'sideItem':sideItem,
            'follower_count': follower_count,
            'gender': gender,
            'answer_count': answer_count
        }
        print(i)
        print('name', name)
        print('url_token',url_token)
        print('headline', headline)
        print('sideItem',sideItem)
        print('follower_count:', follower_count)
        print('gender', gender)
        print('answer_count:', answer_count)
```

```
42
43
44          url_list.append(url)
45          print('-' * 20)
46          # 存入数据库
47          save_to_mongo(info)
```

3. 下面代码是mongo存入数据的小函数：

```
1  def save_to_mongo(info):
2      # 保存到MongoDB中
3      try:
4          if db[MONGO_COLLECTION].insert(info):
5              print('存储到 MongoDB 成功')
6      except Exception:
7          print('存储到 MongoDB 失败')
```

4. 主函数：设置了一个全局变量url列表 存储未访问的用户url，我们将每次调用parse（html)解析函数得到的follower的token信息加上预设值的http格式构成新的url存入列表中，以待被访问

```
1  if __name__ == '__main__':
2      html = get_first_page()
3      parse(html)
4      for url in url_list:
5          try:
6              html_next = get_page(url)
7              parse(html_next)
8              time.sleep(2)
9          except OSError:
10             pass
11         continue
```

5. 数据处理：database2csv.py是将数据库中的文件存入csv文件中，csv3json.py是将csv文件转成特定格式的json提交文件，在此不赘述其中的函数，可以直接看源码。

3.实验截图：

1. Ubuntu16.04 终端运行scrawl.py程序时输出的提示信息：

```
gender 0
answer_count: 176
--------------------
存储到 MongoDB 成功
17
name 夜猫学姐HR
url_token he-lu-lu-10-68
headline 求职就业叮当猫，深夜写稿小姐姐。
sideItem 0
follower_count: 18195
gender 0
answer_count: 104
--------------------
存储到 MongoDB 成功
18
name 知乎求职
url_token zhao-pin-xiao-zhu-shou-59
headline 聚集一线互联网公司优质职位，分享知乎优秀回答者的职场干货
sideItem 已认证的官方帐号
follower_count: 5973
gender -1
answer_count: 12
--------------------
存储到 MongoDB 成功
19
name 王宇HR
url_token zhihu-wyu
headline 欢迎邀请回答职场&amp;amp;管理问题、付费咨询及各类商务合作
sideItem 中国人民大学 人力资源管理专业硕士在读
follower_count: 28762
gender 1
answer_count: 121
--------------------
存储到 MongoDB 成功
20
name 陈大可
url_token chen-qi-69-43-69
headline 育儿快乐成长，职场全力以赴，10年德系汽车营销经验
```

2. 最终的json文件形式：

```
[{"userName": "故事档案局", "userLink": "https://www.zhihu.com/people/
gu-shi-dang-an-ju-71", "headline": "", "SideItem": "", "followers": "23343"},
{"userName": "北京互联网法院", "userLink": "https://www.zhihu.com/people/
bei-jing-hu-lian-wang-fa-yuan", "headline": "", "SideItem": "", "followers": "38270"},
```

4.实验总结：

由于临近期末，我只完成了该实验的一小部分，对于如何反爬虫并没有进行学习。根据我的思路可以通过构造链表的形式将每个用户对应的follower和following的url获取存储到数据库，但是我猜测如此频繁的访问知乎页面的api很可能会造成知乎对我的ip地址访问进行限制，仍旧绕不出要进行反爬虫的学习。