东方财富股吧爬虫

> @李瑶函

遇到的几个问题

主要是频繁请求数据以后网站可能会拒绝返回数据，或者返回数据的速度很慢，或者由于程序设计原因使得爬虫运行比较慢

可能的解决方法

1. 对于New connection error，主要是因为请求的链接过多，并且没有及时关闭（python的request库对于网页的TCP请求默认是维持链接状态的）所以需要设定获取每一条数据以后将这个链接关闭。使用以下方法

   ```
   s=requests.session()
   s.keep_alive=False#对于每一个TCP链接，不要keep alive
   data=s.get(url,headers={'Connection':'close'})#并且在http请求包头中设定关闭链接
   ```

2. 设置一定的时间间隔（如0.5-1s）
3. 另外，考虑到对request获取的数据的解析需要消耗一些时间，可以使用多线程爬虫，并行获取数据并解析。这里使用队列来实现各线程的协调，先将每一页要爬的帖子加入队列，然后各个线程从该队列中获取任务进行爬取，然后再把爬到的东西加入结果队列。此外，各个由于各个线程共享使用上面两个全局队列对象，可能会出现死锁问题，这里简单使用timeout解决（不过可能会损失很少量的几条评论，更好的方法是使用锁机制来解决） 代码见后面
4. 同一个IP重复请求数据可能会被服务器识别并拒绝向该IP返回数据。因此这里找了一些网上的免费代理IP，每次请求时从中随机选取一个 （可能是因为免费的问题，这些代理不是很稳定，可能需要收费的代理IP才比较有效。）

代码

```
import requests
requests.adapters.DEFAULT_RETRIES = 5 #增加重连次数
from bs4 import BeautifulSoup
import re
import json
import sys
from datetime import datetime
import time
import numpy as np
import pandas as pd
import random
import threading
#threading.TIMEOUT_MAX=5
from queue import Queue
stkcd=np.load('stkcd.npy').tolist()
#stkcd=stkcd[:1]

start_page=1#int(sys.argv[1])
end_page=40#int(sys.argv[2])

proxy_list=[{"http":'http://60.217.64.237:31923'},
{'http':'http://115.223.120.254:8010'},
```

```
{'http':'http://117.88.177.174:3000'},{"https":'https://117.88.176.63:3000'},
{'https':'https://49.65.160.69:18118'},
{'http':'http://117.88.253.225:8118'},{'http':'http://110.73.8.171:8123'},
{'http':'http://61.135.155.82:443'},
{'https':'https://117.88.176.186:3000'},{'http':'http://222.95.144.183:3000'},
{'http':'http://121.237.148.16:3000'}]

def get_detail(detail_list,dct_queue,i):
    s=requests.session()#
    s.keep_alive=False#防止维持的链接过多
    while(not detail_list.empty()):
        try:
            #print('thread {} working'.format(i))

            time.sleep(0.3)
            item=detail_list.get()

            reads=item.find_all('span',class_='l1 a1')[0].get_text()
            href='http://guba.eastmoney.com'+item.find_all('span',class_='l3
a3')[0].a['href']

            detail=BeautifulSoup(s.get(href,headers=
{'Connection':'close'}).content, "lxml")#在请求报头里面写明关闭链接

            title=detail.find_all('div', class_='stockcodec .xeditor')
[0].get_text().strip()
            comment_num=detail.find_all('span', class_='comment_num')
[0].get_text().strip([' （','）'])
            #print(title)

            data=json.loads(detail.find_all(class_='data')[0]['data-json'])
            user_id=data['user_id']
            if user_id=='':
                user_id=-1
            else:
                user_id=int(user_id)

            star=data['user_influ_level']
            time_stamp=detail.find_all(class_='zwfbtime')[0].get_text()
            time_stamp=re.sub(r'[\u4e00-\u9fa5a-zA-Z]','',time_stamp)
            #record['id']=hash(user_name+title[:5])
            record={}
            record['time']=time_stamp
            record['read_count']=reads
            record['user_id']=user_id
            record['star']=star
            record['content']=title
            record['comment_count']=comment_num
            dct_queue.put(record)

        except Exception as err:
            #pass
            print(err)

k=1
step=20
for i in list(range(0,len(stkcd),step)):
    df=pd.DataFrame()
```

```
    m=1
    for this_id in stkcd[i:i+step]:
        base_url='http://guba.eastmoney.com/list,'+this_id+'_{}.html'
        print('num of stocks: ',m)
        m+=1
        print(this_id)
        for page in list(range(start_page,end_page)):
            try:
                time.sleep(1)
                print('cur page: ',page)
                #proxy=random.choice(proxy_list)

                s=requests.session()
                s.keep_alive=False

                r=s.get(base_url.format(page),headers=
{'Connection':'close'})#,proxies=proxy)
                article_list = BeautifulSoup(r.content,
'lxml').find_all(class_='articleh normal_post')
                item_count=0
                #print('item_count: ')
                detail_url_queue = Queue(maxsize=200)
                for item in article_list:
                    detail_url_queue.put(item)
                dct_queue=Queue(maxsize=200)

                #设置线程数
                num_of_threads=3
                ths=[]#初始化线程
                for i in range(num_of_threads):
                    th=threading.Thread(target=get_detail,args=
(detail_url_queue,dct_queue,i))
                    ths.append(th)
                #start_time = time.time()
                for i in range(num_of_threads):
                    #ths[i].setDaemon(True)
                    ths[i].start()
                for i in range(num_of_threads):
                    ths[i].join(5)#设置timeout
                while(not dct_queue.empty()):
                    r=dct_queue.get()
                    df=df.append(pd.DataFrame(r,index=[this_id]))
            except Exception as err:
                #pass
                print(err)
    df.to_csv('./comment'+str(k)+'.csv' )
    k+=1
```