上海大学 2018～2019 学年 秋 季学期研究生课程论文

课程名称： 企业级应用开发技术 课程编号： 3ZS081003

论文题目: 微博热搜数据获取

作者姓名: 李琦 学 号: 18721802 成 绩:

论文评语:

任课教师签名:

批阅日期:

摘要

社交媒体作为 web2.0 时代的标志，提供了以用户为中心的各种交流模式和途径。用户在社交媒体上发表和传播消息，关注自己感兴趣的人物。社交媒体中一般拥有数以亿计的人物节点，他们之间通过关注和粉丝关系连成了巨大的社会网络，消息通过这张巨大的社会网络传播。大部分社交媒体提供 API 以便获取社交媒体数据进行相关研究。

本项目主要着眼于人们当前社会上热点事件的看法以及其中的网民数据分析。使用新浪微博提供的开发者 API 通过对微博热门事件评论的数据获取，并针对得到的数据进行分析与可视化。

关键词：社交媒体 微博 数据授权

Abstract

As a symbol of the web2.0 era, social media provides a variety of user-centric communication modes and approaches. Users post and spread messages on social media, focusing on the people they are interested in. Social media generally has hundreds of millions of person nodes, and they have become a huge social network through attention and fan relationships. The news spread through this huge social network. Most social media provide APIs to get social media data for related research.

This project focuses on the views of people's current hot events and the analysis of Internet users. Use the developer API provided by Sina Weibo to obtain data from Weibo's popular event comments, and analyze and visualize the obtained data.

Keywords: social media, microblogging, data authorization

目录

摘要..... 1
Abstract..... 2
第一章 绪论 4
1.1 课题研究背景 4
1.2 网络爬虫的相关技术概述 5
第二章 微博爬取 6
2.1 获取步骤总览 6
2.2 前期准备..... 7
2.2.1 站点分析..... 7
2.2.2 微博登陆..... 7
2.2.3：环境搭建..... 8
2.3 利用 APP 开发者授权进行数据获取..... 9
总结..... 10
参考文献..... 10
附录代码..... 11
Spiders.py..... 11
Cookies.py..... 14
Items.py..... 15
middleware.py..... 16

第一章 绪论

1.1 课题研究背景

伴随整个社会技术的发展，互联网跨入到 web2.0 时代。在 web2.0 时代中，整个互联网平台不停扩展着人们之间的社会关系以及相互之间的交互，从而促使多种社交媒体平台的出现，产生了多种新颖的交互模式和途径。社交媒体是一种在线交互媒体，该媒体最显著特点为具有强大的信息传播能力与影响力，该媒体为用户提供各种即时交流方式，如今已经拥有了大量的用户。近些年来，社交媒体迅速发展，从早期的博客、维基百科、论坛发展到时下流行的社交网站、微博，其正在成为 web2.0 时代的代表性媒体。在新的交互模式的需求下，大量社交媒体涌现出来，在国外，以 FaceBook，Twitter 为代表，在国内，以新浪微博，腾讯微博，QQ 空间，百度贴吧以及人人网为代表。这些社交媒体平台向人们提供社会网络服务，使得用户能够方便快捷地通过互联网自由分享自己的个人信息，获取和传播其他用户的信息。如此一来，整个社交人群的交互信息和背景信息等形成一张社会网络。 国内社交媒体中，近年来以四大微博的发展最为迅速，用户群规模增长最快。有三大微博平台跻身社会化媒体分享榜前十，微博平台已经成为社会化媒体中最受欢迎的平台，其中尤以新浪微博最为火热，截止去年年底，新浪微博注册用户突破 5.03 亿，同比去年年初时增长 73%。去年年底，日均活跃用户数在 4620 万，同比去年年初时增长 82%，日博文量超过 1 亿条，其中有 75%的活跃用户通过移动终端登录微博，服务横跨两岸三地和新马地区。

在各大微博平台飞速发展的同时，微博平台也为开发者和研究者提供了良好的数据获取方式。Twitter、新浪微博、腾讯微博等微博平台都提供了 open API。如今各大微博平台如此风靡，在海量用户数据，关系数据及内容数据的环境下，各大微博平台通过 open API 的方式使得大量用户可以在其平台上开发出各式各样的应用，提高平台的服务质量，同时也为社交网络研究者提供了以网站服务方式对外的数据接口，这其中就包括大量数据下载的 API，为针对微博平台的相关研究工作提供了优良的数据通道。随着各大微博平台提供越来越完善的 open API 服务，出现了大量的微博应用，成为了微博平台中一个相当重要的组成部分，新浪微博为此专门创办了中国微博创新基金为基于其开发平台的开发者提供与创业相关的必要辅导，帮助开发者加速实现创业梦想。

1.2 网络爬虫的相关技术概述

随着网络的发展，无论是传统网络爬虫还是本文研究的微博数据爬虫都面临着飞速增长的信息。对于微博数据而已，由于其近年来的飞速发展，传统通过替换高性能机器已提高爬虫性能的方式变得越来越不合理。网络爬虫现在大量采用分布式的架构，其中以 Hadoop 分布式系统的发展在近些年深受瞩目。如何设计和应用 Hadoop 分布式系统平台是网络爬虫的一项重要技术。尤其是微博数据的飞速增长以及数据结构经常扩展的背景下，利用分布式技术来应对微博数据的变化。 传统网络爬虫中一般通过 http 请求获取相应的页面，而微博爬虫的获取数据形式呈现多样化，主要包括传统的网页获取方式以及微博平台自身为开发者提供的 API。其中随着移动端的飞速发展，为了满足移动用户较慢的网速，微博平台也提供一种更为简化的页面。如何在灵活应用多种途径最快捷迅速的获取到微博数据也是网络爬虫的重要技术之一。

新浪微博每天大约有 2 亿条博文数据产生，现在有接近 5 亿的微博注册用户，人物之间的网络数据已达到亿级别并且每天都在不停变化微博数据虽然是结构化数据，但是其结构在动态扩展，采用非分布式的结构化数据存储显得不合适，根据微博数据的特点选择合适的数据库以及设计针对海量数据的存储技术是不可或缺的。

随着社交网站的日益发展，网民对热点事件的关注度也随之上升。网民的发声散布于众多网络媒体社交网站中，而我们难以从其中某一条提取出重大舆情风向和网民对此的整体看法。基于此问题，本项目主要着眼于人们当前社会上热点事件的看法以及其中的网民数据分析。而众多社交网站中，微博的用户数量稳居首位，据统计截止 2018 年 6 月，中国网民规模达到 8.02 亿人，而其中微博用户数量达到 3.37 亿人，所以我们决定对微博数据进行爬取分析。

第二章 微博爬取

2.1 获取步骤总览

通过新浪微博开发平台获取数据的主要步骤分为两步：第一步创建站内应用，获得应用的 App Key 以及 App Secret，具体申请流程如图 2-1 新浪微博的站内应用创建示意图所示，用户作为新浪微博开发者创建一个未审核的站内应用，并且填入该站内应用的相关信息，提交站内应用并通过审核，最后获得应用对应的 App Key 和 App Secre；第二步为微博用户通过 OAuth2.0 方式授权该应用，返回 tokens，开发者通过这个 token 获得新浪微博数据，如图 2-2 所示，用户登录之后访问第一步创建的应用并请求该应用的授权，应用返回授权信息，用户获取授权信息并将其发送给第三方认证结构用于获得 tokens，之后将 tokens 填入调用 API 的参数中，获取相关数据，每个 tokens 的有效期为 48 小时。

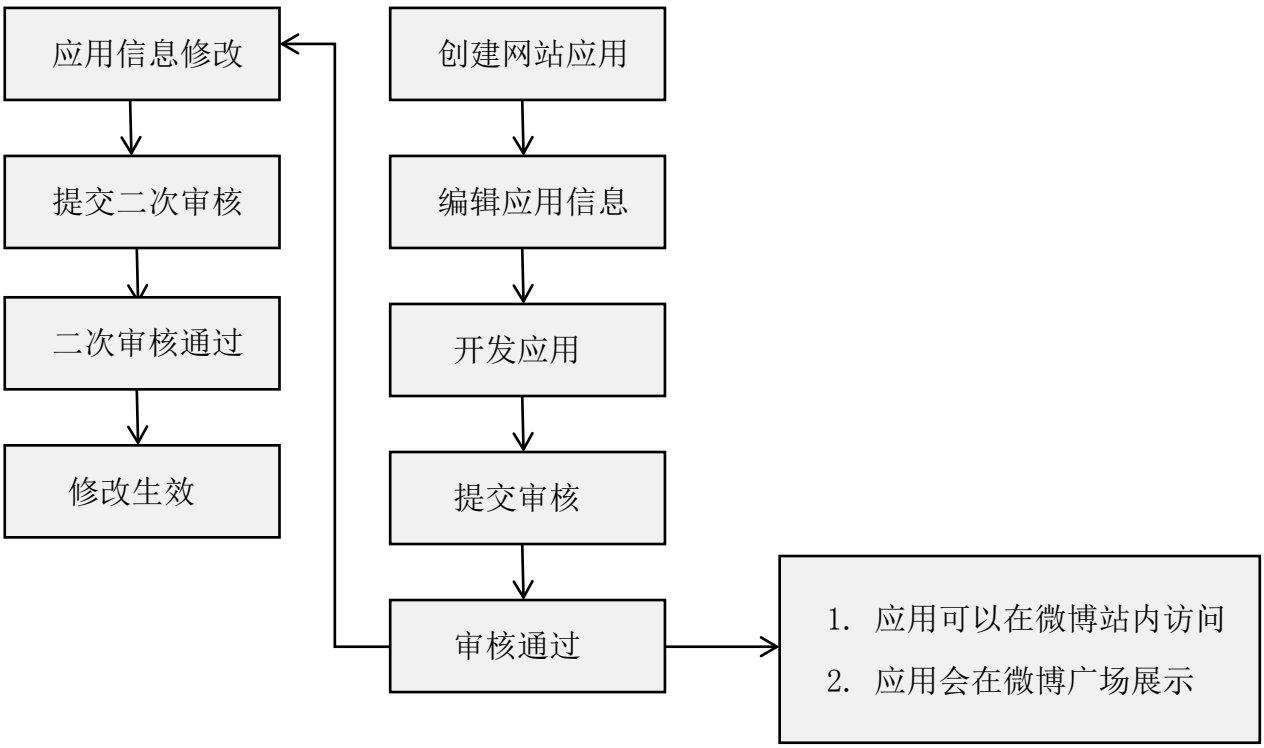

图 2-1 微博站内应用创建示例图

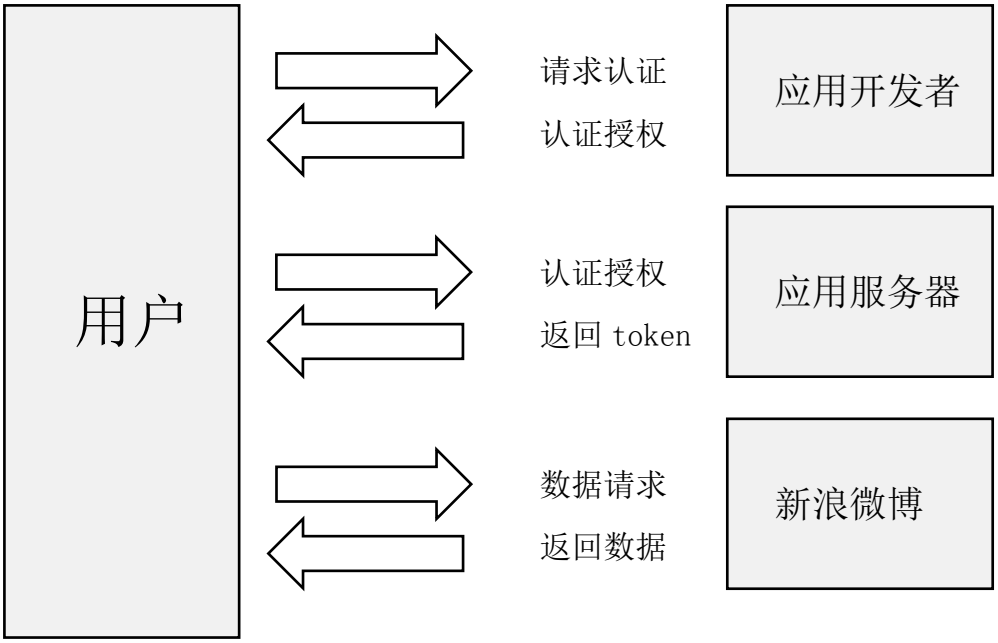

图 2-2 新浪微博的 OAuth2.0 授权方式示意图

2.2 前期准备

2.2.1 站点分析

微博目前一共有三种类型的网站分别是：https://weibo.cn，https://m.weibo.com，https://weibo.com。三个网站的复杂程度依次提高，而其中第二个网站并未提供微博高级搜索功能，并且由于微博的反扒机制，我们选择了第一个网站。

2.2.2 微博登陆

要抓取到微博的数据，首先就是要登陆微博，否则就会重定向到登陆界面。而微博检测用户是不是登陆了微博，就是检查这次 Request 请求携带的 cookie。

而针对大量用户我们不能逐个使用复制 cookie 的方法,必须通过自动化登陆来实现。自动化登陆，就是通过代码来驱动浏览器，进行微博的登陆操作，具体通过自动化工具 selenium 来实现，它支持 Chrome，Firefox 和 PhantomJS 等多种浏览器。好处就是，不用再去分析登陆时候 js 加密，解密的过程，直接而且简单，坏处就是效率比较慢，但是我们只是用它来完成登陆并获取 cookie 的操作。

获取 cookie 以后，可以保存到数据库中。以后每次 request 请求，随机从数据库中选一个 cookie 加上，就免登录了。

2.2.3：环境搭建

1）数据库： mongodb

2）开发环境： Python2.7

3）一个新浪开发者账号：这里我们通过申请新浪合作 APP，来注册一个仅供学习的开发者账号。

4）需要的库：requests 和 csv(这些都可以在 Pycharm 中下载)

5）申请好开发者账号之后，我们打开自己的应用，获取应用的 APPkey 和 APPsecret。

图 2-3

6）设置回调页面

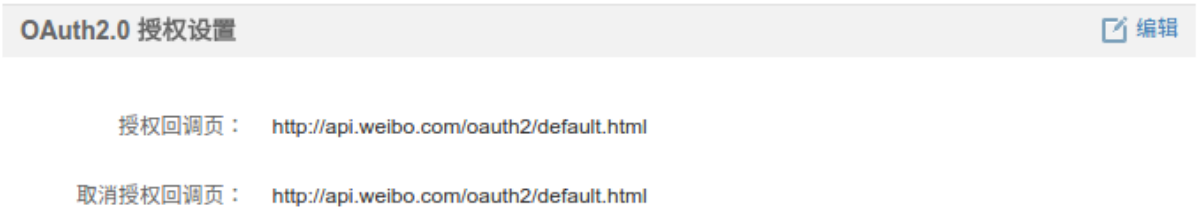

图 1-4

通过新浪微博的 API 接入网站，由于用户无需在网站上注册，就可以直接使用他/她在新浪微博的帐号和口令登录网站，这就需要确保网站在无需知道，也不能知道用户口令

的情况下确认用户已经登录成功。由于用户的口令存储在新浪微博，因此，认证用户的过程只能由新浪微博完成，所以我们只能通过 OAuth(也就是我们自己的应用的回调网址)来确认用户登陆，OAuth 是一个标准的第三方登录协议，借助 OAuth，网站就可以安全地接入来自新浪微博登录成功的用户。

2.3 利用 APP 开发者授权进行数据获取

1）用户在网站上点击“使用新浪微博登录”，网站将用户重定向到新浪微博的 OAuth 认证页，重定向链接中包含 client_id 参数作为网站 ID，redirect_uri 参数告诉新浪微博当用户登录成功后，将浏览器重定向到网站。

2）用户在新浪微博的认证页输入帐号和口令。

3）新浪微博认证成功后，将浏览器重定向到您的网站，并附上 code 参数。

4）网站通过 code 参数向新浪微博请求用户的 access token。

5）网站拿到用户的 access token 后，用户登录完成。

OAuth 的 access token 是提供认证服务的网站（例如新浪微博）生成的令牌，代表一个用户认证信息。在随后的 API 调用中，传入该 access token 就代表这个登录用户，这样，通过 OAuth 协议，网站将验证用户的步骤交给新浪微博完成，并由新浪微博告知用户是否登录成功。

至此，我们的项目就成功完成了对于微博数据权限的获取。接下来就可以通过对微博 API 进行请求获取微博数据，以及进行数据提取。

本项目主要选取的微博内容是微博评论内容，具体有以下几个成员：

created_at（评论时间）

comment_id（评论者 ID）

text（评论内容）

followers（粉丝人数）

follow（关注人数）

province（所在省份代码）

总结

通过本次项目的实践，我们分析了几个当前的热门话题，如重庆公交车坠江事件以及金庸先生去世。从数据分析中我们不难看出，网民对于热点事件的关注程度非常之高，并且对于热点事件能够有足够的正面情感输出。这也间接反映了我国网民素质的普遍提高和网络环境的极大改善。

参考文献

[1] http://open.weibo.com/wiki/API

附录代码

Spiders.py

```
    # encoding=utf-8

from scrapy.spiders import CrawlSpider
from weiboCAR.items import CommentItem, AttitudeItem, RepostItem
from scrapy.http import Request
from bs4 import BeautifulSoup
from weiboCAR import settings

class WeiboLiQi (CrawlSpider):
    host = "http://weibo.cn"
    name = "weiboLiQi"
    allowed_domains = ["weibo.cn"]
    start_urls = settings.WEIBO_IDS
    weiboIDs = set(start_urls)

    def start_requests(self):

        method = getattr(self, 'method', None)

        for weiboID in self.start_urls:
            comment_url = "https://weibo.cn/comment/%s?page=1" %weiboID
            attitude_url = "http://weibo.cn/attitude/%s?page=1" %weiboID
            repost_url = "http://weibo.cn/repost/%s?page=1" %weiboID
        if method is not None:
            if method == "attitude":
                yield Request(url=attitude_url, callback=self.parseA, meta={"weiboID":weiboID})
            elif method == "comment":
                yield Request(url=comment_url, callback=self.parseC, meta={"weiboID":weiboID})
            elif method == "repost":
                yield Request(url=repost_url, callback=self.parseR, meta={"weiboID":weiboID})
            else:
                yield Request(url=comment_url, callback=self.parseC, meta={"weiboID":weiboID})
                yield Request(url=repost_url, callback=self.parseR, meta={"weiboID":weiboID})
                yield Request(url=attitude_url, callback=self.parseA, meta={"weiboID":weiboID})
        else:
            print "请输入参数 method，可能的取值为 comment(只抓评论)，repost(只抓转发)，
attitude(只抓点赞)，all(三种都抓)"
```

```
    def parseC(self,response):
        """ 提取评论信息 """
        html = response.text
        soup = BeautifulSoup(html,"html.parser",from_encoding="utf8")
        comments = soup.find_all("div",{"class":"c"})
        for c in comments:
            try:
                print 'wwwwwwwwwwwwwwwwwwwww'
                item = CommentItem()
                item["weiboID"] = response.meta["weiboID"]
                item["userId"] = str(c.get("id"))
                item["userName"] = c.find("a").text
                item["userUrl"] = c.find("a").get("href")
                print item["userId"]
                item["commentLike"] = c.find("span",{"class":"cc"}).find("a").text
                item["commentText"] = c.find("span",{"class":"ctt"}).text
                item["commentTime"] = c.find("span",{"class":"ct"}).text.strip()
                yield item
            except:
                print 'liqidebug'
                pass

            next_url = None
            try:
                next_url = soup.find("div",{"id":"pagelist"}).find("form").find("a",text=r'下页').get("href")
            except:
                pass

        if next_url:
            yield Request(url=self.host+next_url,callback=self.parseC,meta={"weiboID":response.meta["weiboID"]})
        else:
            pass

    def parseA(self,response):
            """ 提取点赞信息 """
            html = response.text
            soup = BeautifulSoup(html,"html.parser",from_encoding="utf8")
            comments = soup.find_all("div",{"class":"c"})

            for c in comments:
                try:
```

```
                    item = AttitudeItem()
                    item["weiboID"] = response.meta["weiboID"]
                    item["userName"] = c.find("a").text
                    item["userUrl"] = c.find("a").get("href")
                    item["attitudeTime"] = c.find("span",{"class":"ct"}).text.strip()
                    yield item
                except:
                    pass

                next_url = None
                try:
                    next_url = soup.find("div",{"id":"pagelist"}).find("form").find("a",text=r'下页
').get("href")
                except:
                    pass

            if next_url:
                yield Request(url=self.host+next_url,
callback=self.parseA,meta={"weiboID":response.meta["weiboID"]})
            else:
                pass

    def parseR(self,response):
            """ 提取转发信息 """
            html = response.text
            soup = BeautifulSoup(html,"html.parser",from_encoding="utf8")
            comments = soup.find_all("div",{"class":"c"})
            print len(comments)
            for c in comments:
                try:
                    item = RepostItem()
                    item["weiboID"] = response.meta["weiboID"]
                    item["userName"] = c.find("a").text
                    item["userUrl"] = c.find("a").get("href")
                    texts = c.find_all(text=True)
                    texts = [t.strip() for t in texts if t.strip() != ""]
                    item["repostText"] = "".join(texts[1:-2])
                    item["repostTime"] = c.find("span",{"class":"ct"}).text.strip()
                    item["repostLike"] = c.find("span",{"class":"cc"}).find("a").text
                    yield item
                except:
                    pass

                next_url = None
```

```
                try:
                    next_url = soup.find("div",{"id":"pagelist"}).find("form").find("a",text=r'下页
').get("href")
                except:
                    pass

            if next_url:
                yield Request(url=self.host+next_url,
callback=self.parseR,meta={"weiboID":response.meta["weiboID"]})
            else:
                pass
```

Cookies.py

```
    # encoding=utf-8
import json
import base64
import requests
myWeiBo = [
    {'no': '18817843537', 'psw': 'bailey19931015'},
    {'no': '18817843537', 'psw': 'bailey19931015'},
]
def getCookies(weibo):
    """ 获取 Cookies """
    cookies =[ '_T_WM=b1bea5c50051455228eb37efdf5e2d29;
SUB=_2A252vOl1DeRhGeBI71AR9yvKyzmIHXVSXvc9rDV6PUJbkdBeLU_ykW1NRnp-IGCh861bR7
4VWgE6WhF-ejmo7lfv; SUHB=0rrrEA_CYD09f5;
SCF=At8AZMsqTEZq-LsMo-2HDe9038MNQovJ_SM1K1imtRjoUZ9tj_ivU4srX8ADDaeCMDIjALv-C
Dpb0TbepnQNLgs.; WEIBOCN_FROM=1110006030; MLOGIN=1;
M_WEIBOCN_PARAMS=uicode%3D20000174%26fid%3D102803']
    loginURL = r'https://login.sina.com.cn/sso/login.php?client=ssologin.js(v1.4.15)'
    for elem in weibo:
        account = elem['no']
        password = elem['psw']
        username = base64.b64encode(account.encode('utf-8')).decode('utf-8')
        postData = {
            "entry": "sso",
            "gateway": "1",
```

```
            "from": "null",
            "savestate": "30",
            "useticket": "0",
            "pagerefer": "",
            "vsnf": "1",
            "su": username,
            "service": "sso",
            "sp": password,
            "sr": "1440*900",
            "encoding": "UTF-8",
            "cdult": "3",
            "domain": "sina.com.cn",
            "prelt": "0",
            "returntype": "TEXT",
        }
        session = requests.Session()
        r = session.post(loginURL, data=postData)
        jsonStr = r.content.decode('gbk')
        info = json.loads(jsonStr)
        if info["retcode"] == "0":
            print "Get Cookie Success!( Account:%s )" % account
            cookie = session.cookies.get_dict()
            cookies.append(cookie)
        else:
            print "Failed!( Reason:%s )" % info['reason']
    return cookies


cookies = getCookies(myWeiBo)
print "Get Cookies Finish!( Num:%d)" % len(cookies)
```

Items.py

```
    # -*- coding: utf-8 -*-

# Define here the models for your scraped items
#
# See documentation in:
# http://doc.scrapy.org/en/latest/topics/items.html

from scrapy import Item, Field
```

```
class CommentItem(Item):
    """ 微博评论 """
    weiboID = Field()
    userId = Field() #评论者 ID
    userName = Field() #评论者
    userUrl = Field()  #评论者首页
    commentLike = Field() #评论被点赞数
    commentText = Field() #评论内容
    commentTime = Field() #发布时间

class AttitudeItem(Item):
    """ 微博点赞 """
    weiboID = Field()
    userName = Field() #点赞者
    userUrl = Field()  #评论者首页
    attitudeTime = Field() #发布时间

class RepostItem(Item):
    """ 微博转发 """
    weiboID = Field()
    userName = Field() #转发者
    userUrl = Field()  #评论者首页
    repostText = Field() #转发内容
    repostLike = Field() #点赞数
    repostTime = Field() #发布时间
```

middleware.py

```
# encoding=utf-8
import random
from cookies import cookies
from user_agents import agents
class UserAgentMiddleware(object):
    """ 换 User-Agent """
    def process_request(self, request, spider):
        agent = random.choice(agents)
        request.headers["User-Agent"] = agent
class CookiesMiddleware(object):
```

```
""" 换 Cookie """
def process_request(self, request, spider):
    cookie = random.choice(cookies)
    request.cookies = cookie
```

pipeline.py

```
# -*- coding: utf-8 -*-
from weiboCAR.items import CommentItem, AttitudeItem, RepostItem
class saveTxtPipeline(object):
    def process_item(self, item, spider):
        f = item["weiboID"]
        if isinstance(item, CommentItem):
            line = ""
            for v in item.values():
                line = line + v.encode("utf8") + "\t"
            with open(f+"_comment.txt","a") as outfile:
              outfile.write(line+"\n")
        elif isinstance(item, AttitudeItem):
            line = ""
            for v in item.values():
                line = line + v.encode("utf8") + "\t"
            with open(f+"_attitude.txt","a") as outfile:
              outfile.write(line+"\n")
        elif isinstance(item, RepostItem):
            line = ""
            for v in item.values():
                line = line + v.encode("utf8") + "\t"
            with open(f+"_repost.txt","a") as outfile:
              outfile.write(line+"\n")
        else:
           pass
```

```
return item
```