这部分将介绍爬取2019年9月1日~2019年10月10日所有上过热搜榜的话题的高评论微博下面对应评论的过程。

0. 实现爬取评论的思路

首先先将这些日期下所有上过热搜榜的话题找到。考虑到每个热搜话题（包含买的热搜）下面一定会有一条相对的高赞高评论微博（即一般网友都是在这条微博下面讨论最为热烈），因此对于每个话题，我们可以找到其高评论微博的评论区。然后我们将评论区页面的html爬取下来，经过正则表达式的清洗与整理，我们就可以得到我们需要的评论以及其对应的用户、评论时间等。在爬取下来这些数据以后，我们可以对该日每个用户出现的频次进行统计，并标记在每一条评论的后面作为参考。为实现以上的步骤，我们可以基于Python的selenium库与requests库，并配合Chrome Driver来控制Chrome浏览器，以模拟用户浏览的效果，从而可以降低被微博服务器发现爬虫的风险；然后再利用re库与csv库进行数据清洗与整理。

1. 找到所有上过热搜榜的话题

为了寻找热搜话题，我们使用“热搜神器“，如下图。



图1 “热搜神器”界面示意图

（由于此步骤是在2019年10月完成的，而本文所作日期为2020年5月，因此可能其页面及HTML代码都与之前有很大的区别）

通过选取一个日期并将HTML代码下载下来，我们可以得到包含那一天所有热搜话题的未清洗数据，如下图。



图2 热搜话题未清洗数据示意图

将40天的HTML代码均按“日期.txt“保存好即可。

2. 通过热搜榜定位所有热搜的评论区

任意选取步骤1中的一条热搜来观察。

|  |
| --- |
| <div class="card">  <div class="card-main clear">  <h3 class="card-keyword">  <span class="card-num">1</span>胡冰卿恋情</h3>  <p class="search-count right">搜索量 4304458</p></div>  <div class="card-other clear">  <p class="left">当日累计在榜时间：7小时30分</p>  <p class="right">当日最高排名：  <strong>1</strong></p>  </div>  </div> |

（这里为了便于讲解从而进行了格式化，实际如图2那样保存下来的是没有换行的。）

可以发现，<span class="card-num">、</span>和</h3>中间夹着热度序号与热搜名称，而后面可以按此法发现搜索量、在榜时间、当日最高排名等信息。据此我们可以写出一个正则表达式来进行数据的筛选。正则表达式如下。

|  |
| --- |
| r'<span class="card-num">(.\*?)</span>(.\*?)</h3>.\*?搜索量 (.\*?)</p>.\*?当日累计在榜时间：(.\*?)</p>.\*?当日最高排名.\*?<strong>(.\*?)</strong>' |

对上例进行清洗后的结果为

|  |
| --- |
| [1, 胡冰卿恋情, 4304458, 7小时30分, 1] |

将所有热搜话题按此法炮制并以.csv格式保存，结果如下。此过程步骤代码见附件“zhengliresou.py“。

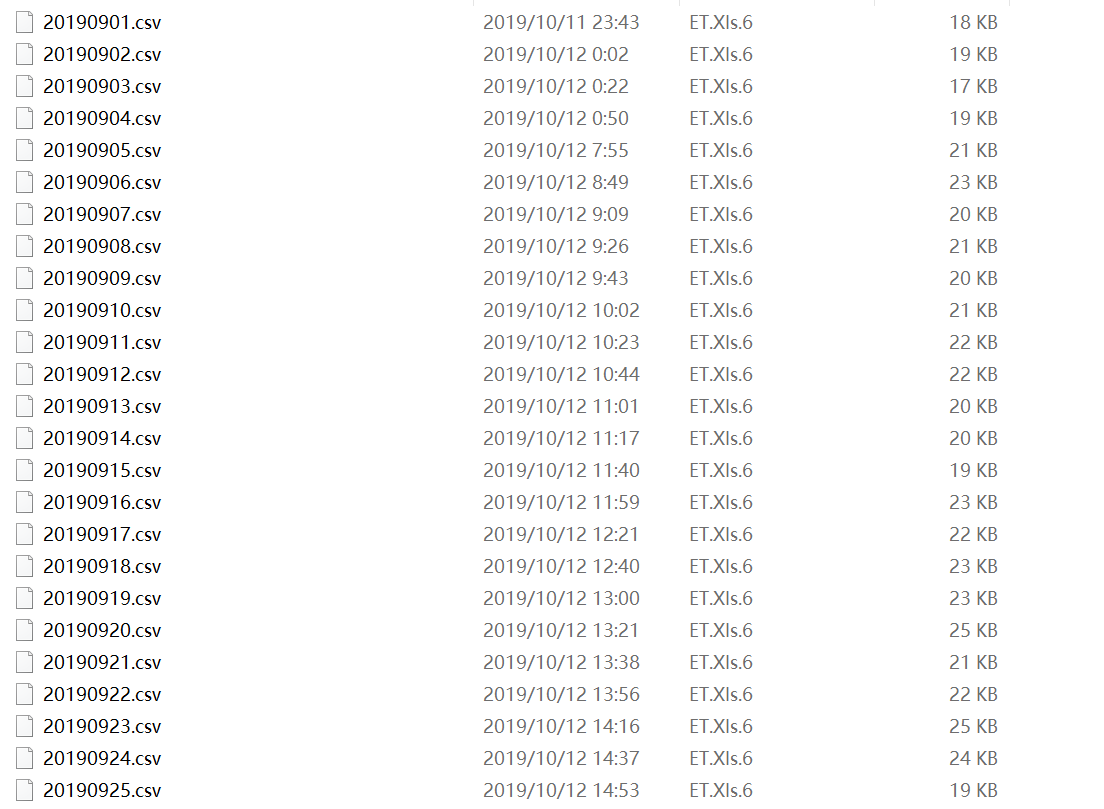


图3 所有日期热搜话题整合列示意图

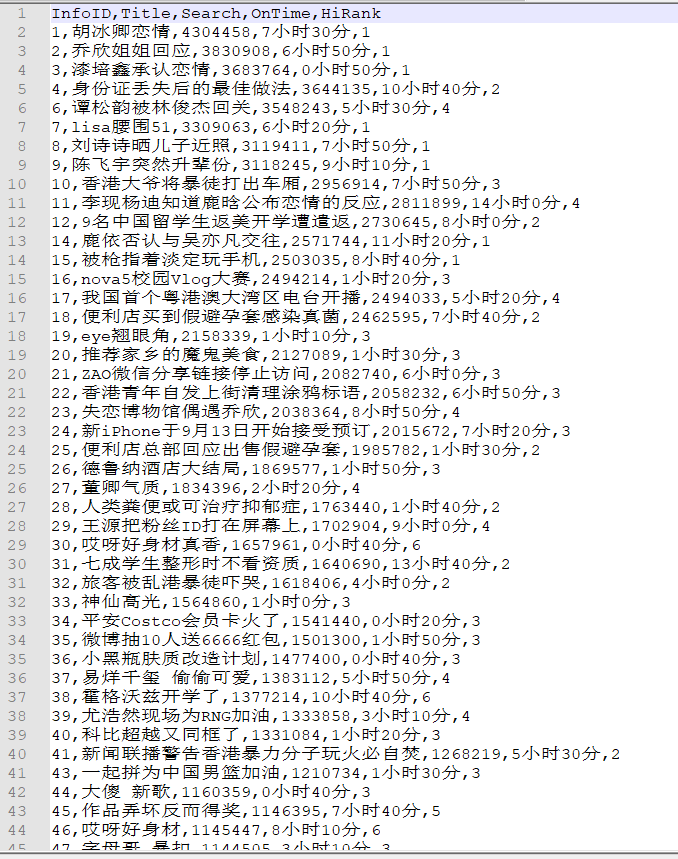


图4 单个日期热搜话题示意图

接下来，用以下命令打开一个Chrome（需要安装对应版本的Chrome Driver）来作为模拟用户。

|  |
| --- |
| chrome.exe --remote-debugging-port=9222 --user-data-dir="C:\selenum\AutomationProfile" |

首先用一个真实的微博帐号登录<https://weibo.cn/>，将cookies存入该Chrome的缓存。接着模拟用户搜索话题、点进第一个最高评微博、点开“热门评论“（用评论的热度进行排序），并将该url存入csv中。结果如下。此过程步骤代码见附件”sousuo.py“。



图5 将整合url后的单个日期热搜话题示意图

3. 通过url进入微博并将所有评论爬取。

获得了所有话题对应的高评热评第一页url以后，即可开始遍历这些链接，并爬取所有评论。

先随意打开一个链接，页面如下。

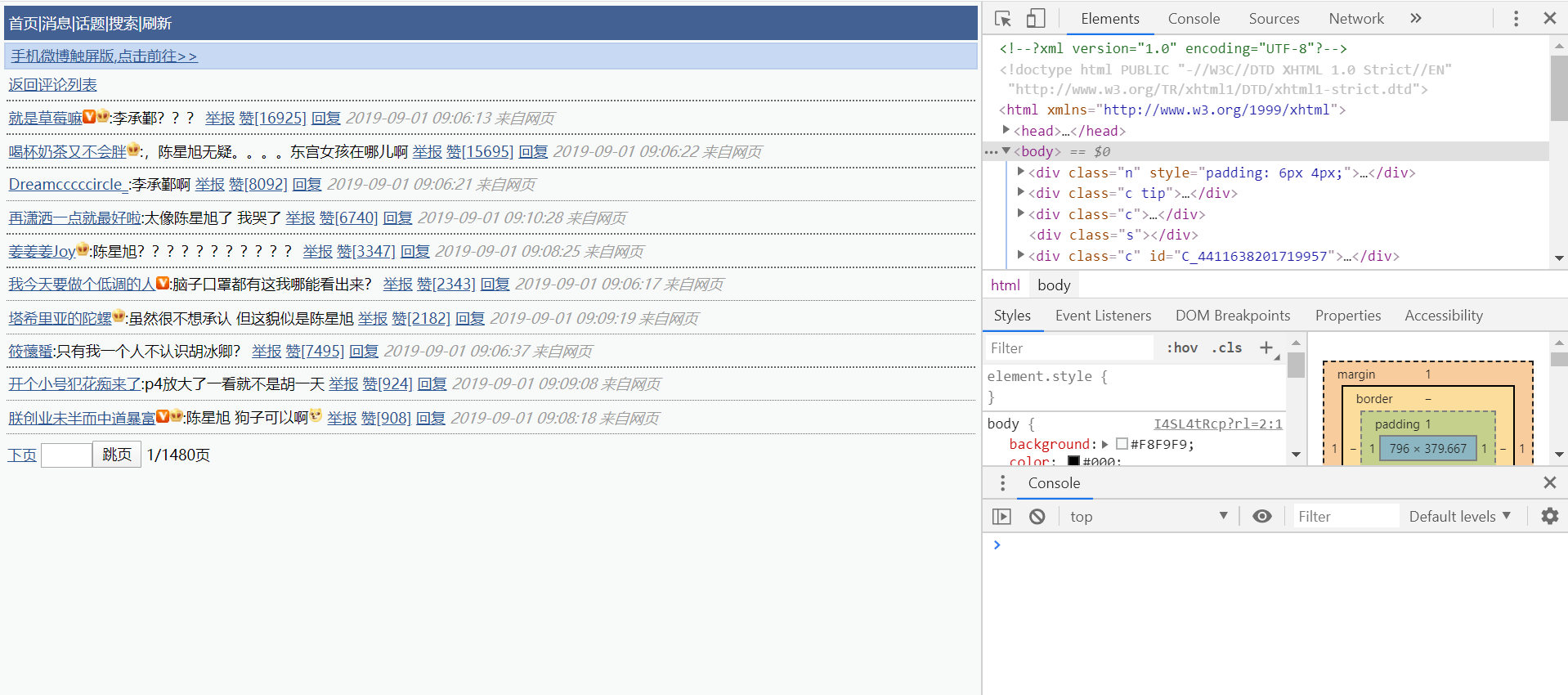


图6 微博及其html代码示意图

选取任意一条评论查看其HTML代码。

|  |
| --- |
| <div class="c" id="C\_4411638201719957">  <a href="/u/5963621128">就是草莓嘛</a>  <img src="https://h5.sinaimg.cn/upload/2016/05/26/319/5338.gif" alt="V">  <img src="https://h5.sinaimg.cn/upload/2016/05/26/319/donate\_btn\_s.png" alt="M">:  <span class="ctt">李承鄞？？？</span>&nbsp;  <a href="/spam/?cid=4411638201719957&amp;fuid=5963621128&amp;type=2&amp;rl=3">举报</a>&nbsp;  <span class="cc">  <a href="/attitude/I4SLi7drf/update?object\_type=comment&amp;uid=1619163742&amp;rl=3&amp;st=6f9df7">赞[16925]</a></span>&nbsp;  <span class="cc">  <a href="/comments/reply/I4SL4tRcp/4411638201719957?rl=3&amp;st=6f9df7">回复</a></span>&nbsp;  <span class="ct">2019-09-01 09:06:13&nbsp;来自网页</span></div> |

用与步骤2相同的分析方法我们可以用正则表达式筛选出评论内容、评论用户、点赞数、评论时间等信息。

而完成此步骤的前提是我们需要获得页面的HTML代码。由于页面众多，且每一个页面还有“下一页“链接到更多评论页面，因此手动下载是不现实的。我们可以利用request库获得链接的HTML代码，同时由于url中是用page=来表示页码的，因此也可以很方便地爬取一个话题下的所有评论。

同时，会有以下情况需要在编写代码时进行相应的应对。首先，若被微博识别出是爬虫，可能就不会返回任何数据。我们可以在代码中的关键步骤前设置time.sleep()（即停止运行）来看起来“更像真实用户操作“，同时用selenium不定期重新登录获取新的cookies。其次，用url来控制页码可能会出现已经没有更多评论但爬虫还在继续寻找的情况。观察到对于没有数据的页面，正则表达式清洗其HTML时会返回空列表。因此，我们可以设置一个页码上限（如10000），并进行判断，若连续超过3页没有数据或到达页码上限，则判定为该话题以爬取完毕。

据此，我们可以爬出以上话题的所有评论。爬取代码截取和程序运行页面如下图。

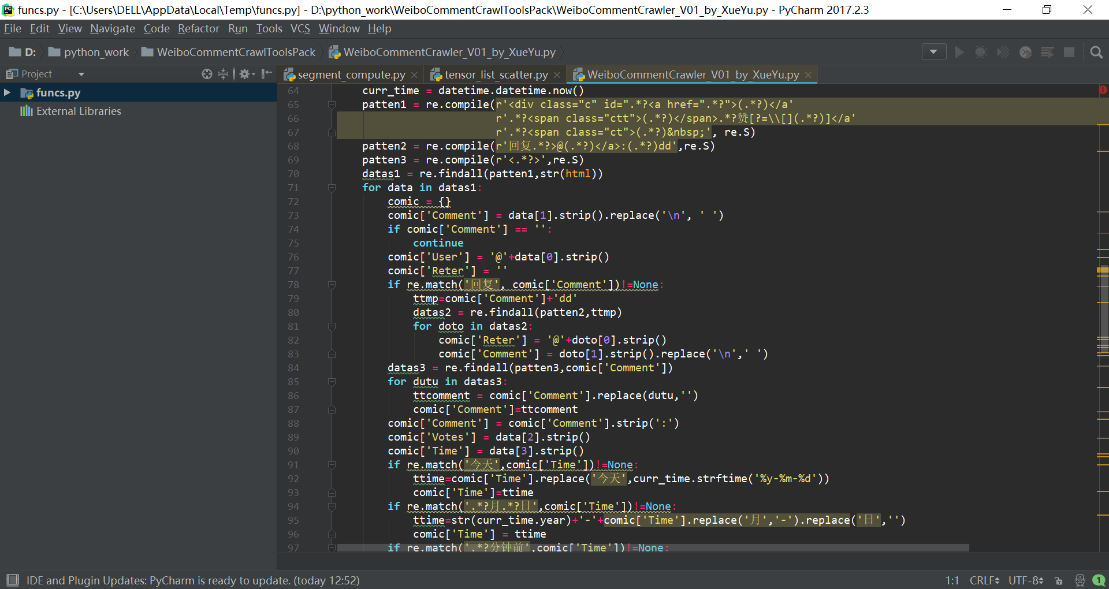


图7 爬虫部分代码截取示意图



图8 爬虫运行界面示意图

爬取结果如下图。

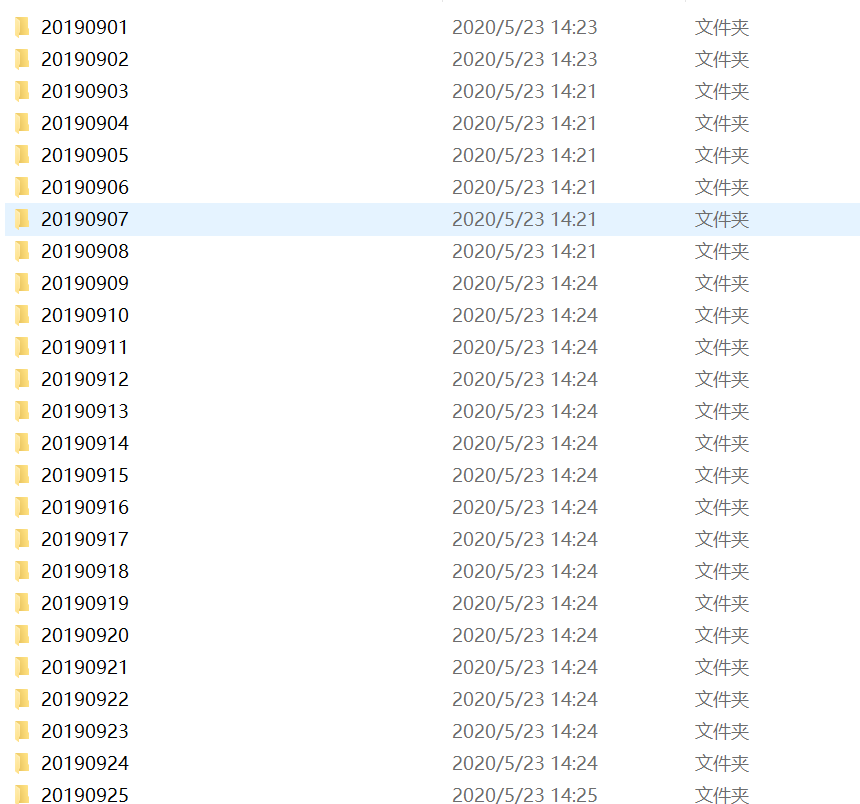


图9 爬虫爬取后所有文件夹示意图

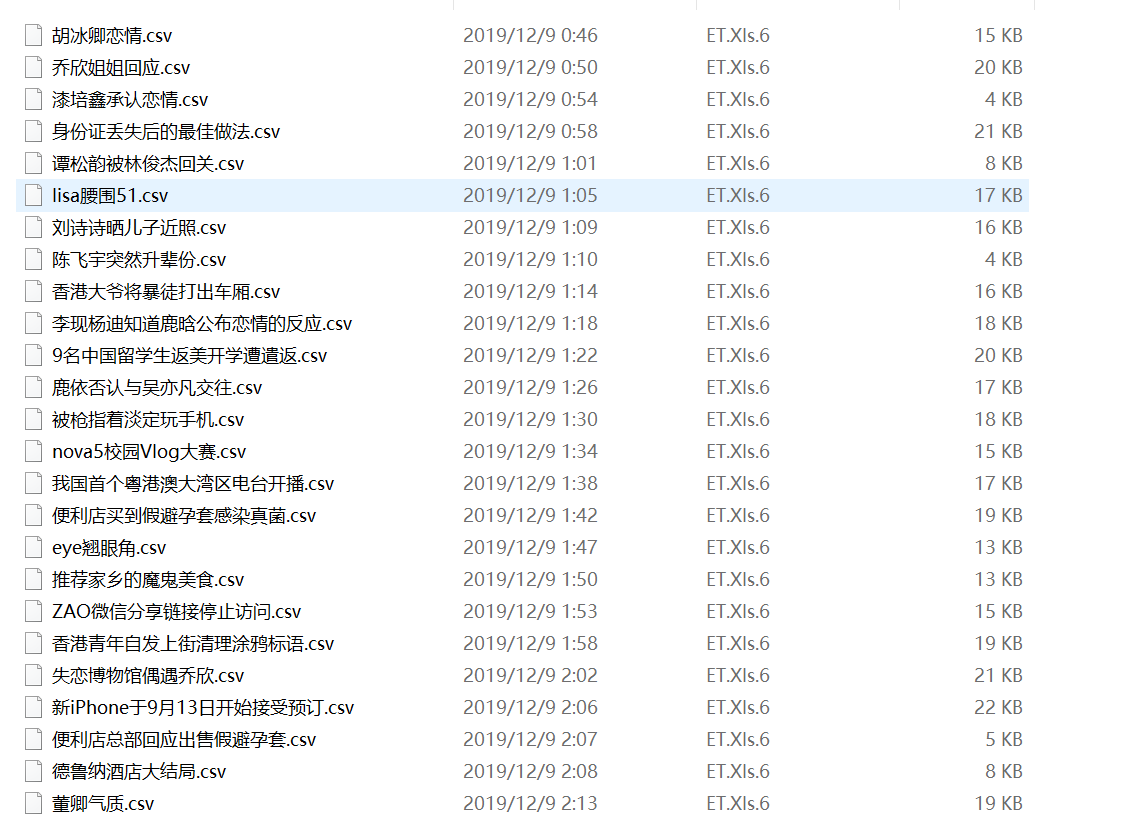


图10 爬虫爬取后单日所有话题示意图



图11 单个话题热搜爬取结果示意图

以上步骤所用到的步骤已经封装至应用“WeiboCommentCrawler\_V01\_by\_XueYu.exe“。在“dateconfig.txt“中填入日期（如”dates\\20190901“）即可爬取对应日期的评论。源代码见附件“WeiboCommentCrawler\_V01\_by\_XueYu.py“。

由于设置了多处time.sleep()，因此爬取速度相对较慢。我们租用了一台阿里云轻量应用服务器以挂起应用该从而能在组员不使用电脑时实现持续的爬取。

4. 统计用户在一天中出现的频次

获得了所有评论后，我们可以统计每个用户在当天出现过的次数，并在每条评论后面做标注。遍历一遍所有文件即可实现。结果如下图。此过程步骤代码见附件“tongji.py“。



图12 统计用户出现次数后的单个话题热搜示意图