爬了 48048 条评论，解读 9.3 分的「毒液」是否值得一看？

Python开发者 昨天

(给Python开发者加星标，提升Python技能)

> 转自：CSDN-Ryan

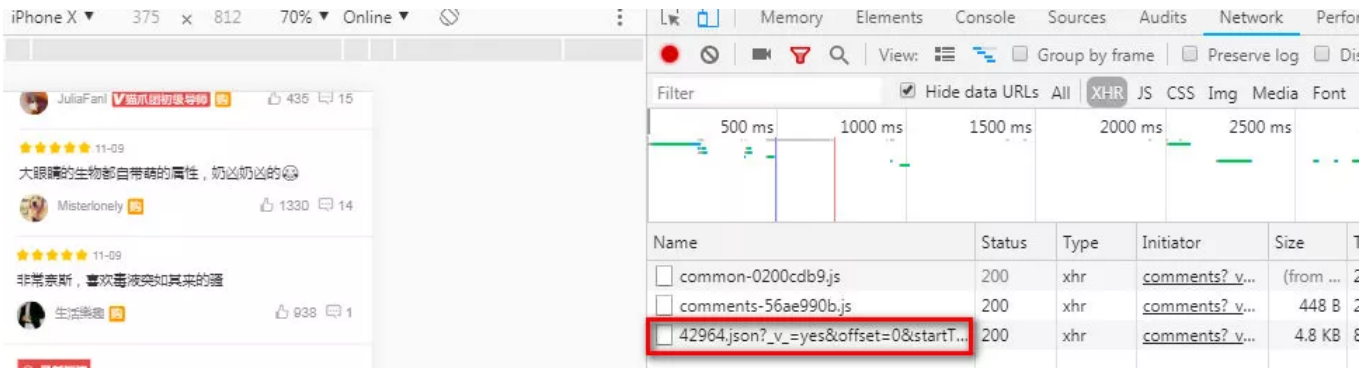
11月，由汤姆·哈迪主演的“毒液：致命守护者”在国内上映，依托漫威的光环以及演员们精湛的演技，这部动作科幻片在猫眼评分得到豆瓣7.4的评分，口碑和票房都高于大多数同期上映的其他影片。

所以周日的时候跟基友去电影院去看了这场正邪共生的电影，100多人的影院座无虚席，不过看完之后对比其他漫威作品，我倒也没觉得有多大的惊喜，觉得猫眼上的9.3评分的感受不符。

头部的几条评论显然有些夸大，那大众对“毒液”感受是怎么呢？于是笔者动手开始分析起来。

获取数据

首先要获取数据，准备爬取猫眼上的电影评论作为本次分析样本，PC官网上只显示了电影的10条热门短评，显然不够，于是准备从M端抓包找到评论接口。



> 接口链接：
> http://m.maoyan.com/mmdb/comments/movie/42964.json?v=yes&offset=15&startTime=2018-11-20%2019%3A17%3A16。

接口中对我们本次抓取主要有用的参数是offset偏移量以及日期，这两个条件限制了抓取的条数。分析接口结果：

```
{
 - cmts: [
    - {
        approve: 0,
        approved: false,
      - assistAwardInfo: {
            avatar: "",
            celebrityId: 0,
            celebrityName: "",
            rank: 0,
            title: ""
        },
        avatarurl: "https://img.meituan.net/avatar/6e308a5af0e240012ba2c444f8556c514359.jpg",
        cityName: "洛阳",
        content: "很多东西都没说明白，为什么毒液可以自由选择宿主呢？实验的时候就不行。不过还可以，期待与复仇者的碰面
        filmView: false,
        gender: 1,
        id: 1045405777,
        isMajor: false,
        juryLevel: 0,
        majorType: 0,
        movieId: 42964,
        nick: "很磐的阿石",
        nickName: "很磐的阿石",
        oppose: 0,
        pro: false,
        reply: 0,
        score: 4.5,
        spoiler: 0,
        startTime: "2018-11-20 19:24:06",
        supportComment: true,
        supportLike: true,
        sureViewed: 1,
```


这里有用户评论的相关数据，我们选取了地理位置（用户为授权无法获取）、评论内容、用户名、评分以及评论时间的数据，通过python的requests模块开始爬取。导入本次爬取需要的包，开始抓取数据。

```
`def get_data(url):
    headers = {
        'User-Agent': 'Mozilla/5.0 (iPhone; CPU iPhone OS 11_0 like Mac OS X) AppleW
    html = requests.get(url, headers=headers)
    if html.status_code ==200:
        return html.content
    else:
        return none`
```

其次是解析Json数据，每个接口有15条评论数据，10条热门评论数据，我们将评论数据中用户名、城市名、评论内容、评分、评论时间依次解析出来，并返回。

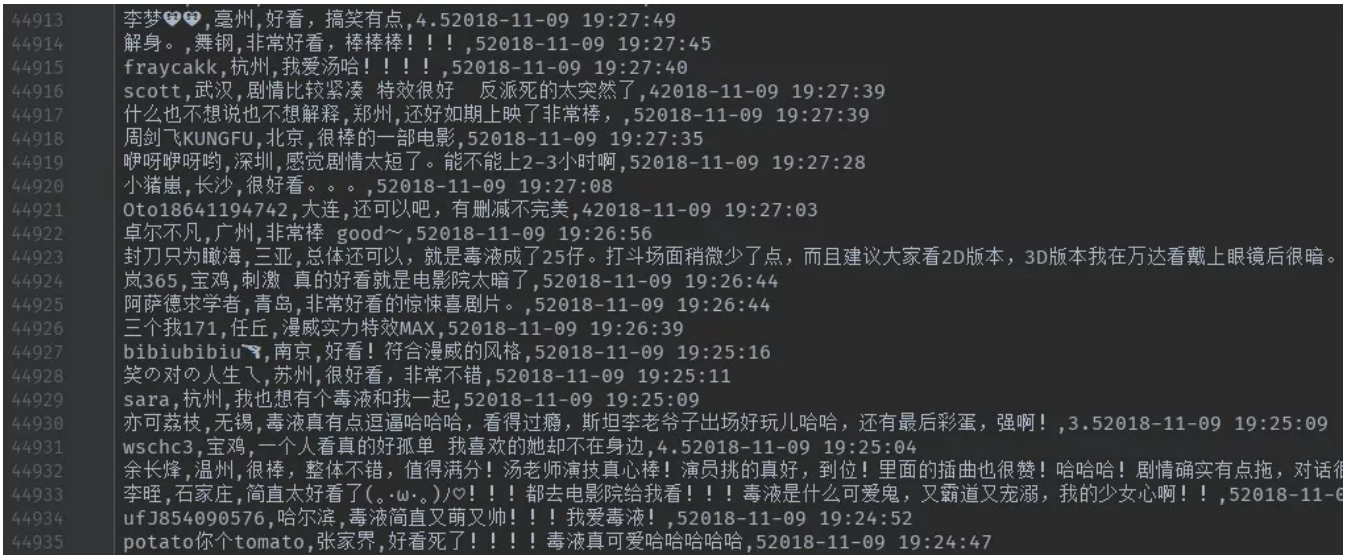
```
`def parse_data(html):
    json_data = json.loads(html)['cmts']
    comments = []
    try:
        for item in json_data:
            comment = {
                'nickName': item['nickName'],
                'cityName': item['cityName'] if 'cityName' in item else '',
```

```
                'content': item['content'].strip().replace('\n', ''),
                'score': item['score'],
                'startTime': item['startTime']
            }
            comments.append(comment)
        return comments
    except Exception as e:
        print(e)`
```

接着我们将获取到的数据保存到本地。此过程中，对接口url中时间的处理借鉴了其他博主的爬虫思路，将每次爬取的15条数据取最后一条的评论时间，减去一秒（防止重复），从该时间向前获取直到影片上映时间，获取所有数据。

```
`def save():
    start_time = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    end_time = '2018-11-09 00:00:00'
    while start_time > end_time:
        url = 'http://m.maoyan.com/mmdb/comments/movie/42964.json?_v_=yes&offset=15&star
            ' ', '%20')
        html = None
        try:
            html = get_data(url)
        except Exception as e:
            time.sleep(0.5)
            html = get_data(url)
        else:
            time.sleep(0.1)
        comments =parse_data(html)
        start_time = comments[14]['startTime']
        print(start_time)
        start_time = datetime.strptime(start_time, '%Y-%m-%d %H:%M:%S') + timedelta(secc
        start_time = datetime.strftime(start_time, '%Y-%m-%d %H:%M:%S')
        for item in comments:
            print(item)
            with open('files/comments.txt', 'a', encoding='utf-8')as f:
                f.write(item['nickName']+','+item['cityName'] +','+item['content']+','+s
if __name__ == '__main__':
    url = 'http://m.maoyan.com/mmdb/comments/movie/42964.json?_v_=yes&offset=15&startTim
    html = get_data(url)
    reusults = parse_data(html)
    save()`
```

最终抓取了48048条评论相关数据作为此次分析样本。



数据可视化

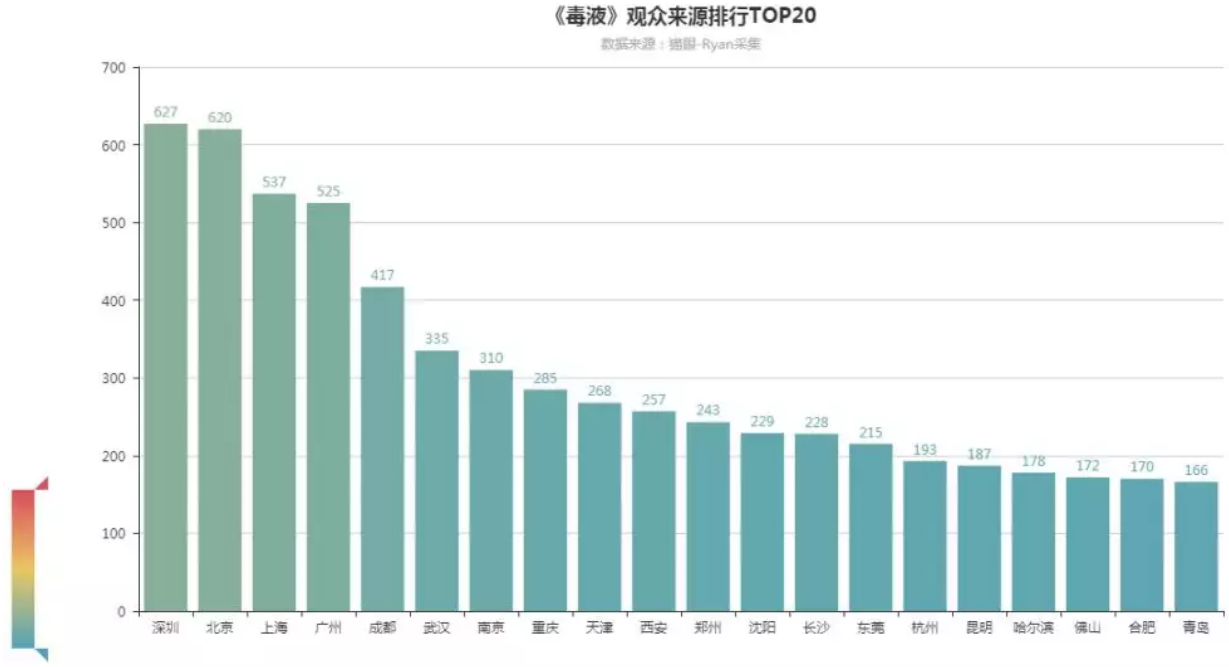
数据可视化采用了pyecharts，按照地理位置制作了毒液观众群的分布图。部分代码如下：

```
`geo = Geo('《毒液》观众位置分布', '数据来源：猫眼-Ryan采集', **style.init_style)
    attr, value = geo.cast(data)
    geo.add('', attr, value, visual_range=[0, 1000],
            visual_text_color='#fff', symbol_size=15,
            is_visualmap=True, is_piecewise=False, visual_split_number=10)
    geo.render('观众位置分布-地理坐标图.html')

    data_top20 = Counter(cities).most_common(20)
    bar = Bar('《毒液》观众来源排行TOP20', '数据来源：猫眼-Ryan采集', title_pos='center', w
    attr, value = bar.cast(data_top20)
    bar.add('', attr, value, is_visualmap=True, visual_range=[0, 3500], visual_text_colc
            is_label_show=True)
    bar.render('观众来源排行-柱状图.html')`
```

从可视化结果来看，“毒液”观影人群以东部城市为主，观影的top5城市为深圳、北京、上海、广州、成都。

观众地理位置分布图



观众来源排行TOP20

用户评论，词云图

只看观众分布无法判断大家对电影的喜好，所以我把通过jieba把评论分词，最后通过wordcloud制作词云，作为大众对该电影的综合评价。

```
comments = []
with open('files/comments.txt', 'r', encoding='utf-8')as f:
    rows = f.readlines()
    try:
        for row in rows:
```

```
                comment = row.split(',')[2]
                if comment != '':
                    comments.append(comment)
                # print(city)
            except Exception as e:
                print(e)
        comment_after_split = jieba.cut(str(comments), cut_all=False)
        words = ' '.join(comment_after_split)
        #多虑没用的停止词
        stopwords = STOPWORDS.copy()
        stopwords.add('电影')
        stopwords.add('一部')
        stopwords.add('一个')
        stopwords.add('没有')
        stopwords.add('什么')
        stopwords.add('有点')
        stopwords.add('感觉')
        stopwords.add('毒液')
        stopwords.add('就是')
        stopwords.add('觉得')
        bg_image = plt.imread('venmo1.jpg')
        wc = WordCloud(width=1024, height=768, background_color='white', mask=bg_image, font
                       stopwords=stopwords, max_font_size=400, random_state=50)
        wc.generate_from_text(words)
        plt.imshow(wc)
        plt.axis('off')
        plt.show()
`
```

从最终的词云结果上来看，大多数观众还是对“毒液”很满意的。

推荐阅读 （点击标题可跳转阅读）

手把手教你写网络爬虫（2）：迷你爬虫架构

Python 爬虫实践：《战狼2》豆瓣影评分析

Python 爬虫抓取纯静态网站及其资源

觉得本文对你有帮助？请分享给更多人

关注「Python开发者」加星标，提升Python技能