百度指数爬虫代码说明

一、 环境配置

1、Python 需要安装 selenium 模块

2、需要安装浏览器驱动程序，这里我们使用的是谷歌浏览器，需要安装 chromedriver

二、 百度指数页面及爬取思路

图 1 为百度指数的页面，词条的指数以曲线图形式显示，横坐标为日期，纵坐标为指数，默认显示的是最近 30 天的数据。

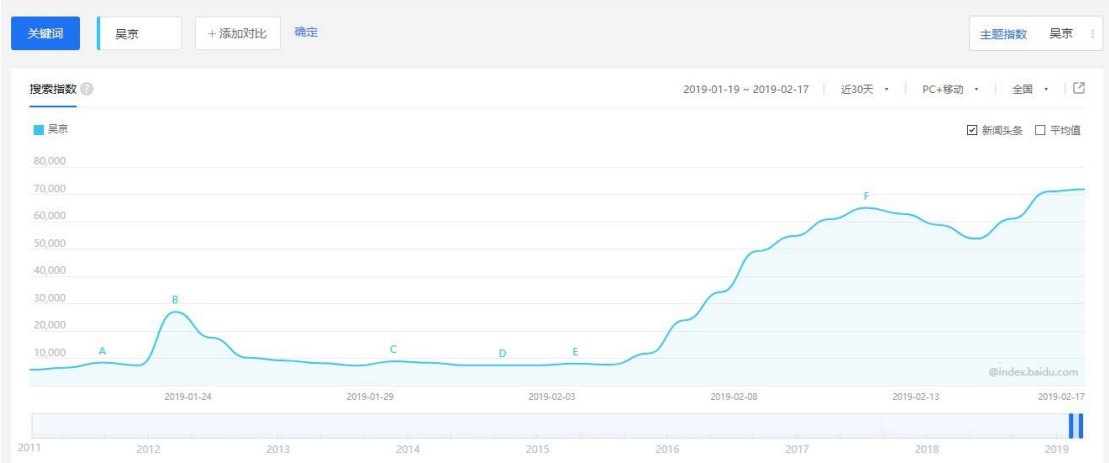

图 1 百度指数页面

页面中没有指数的具体数值，只有当将鼠标移动到曲线图中，才会根据鼠标悬停位置的横坐标显示具体的数值，如下图 2 所示。因此我们无法直接解析页面获得指数的数值，故考虑使用 selenium 模拟手动操作鼠标从指数图表左边滑到右边，获得每个位置的指数具体数据。

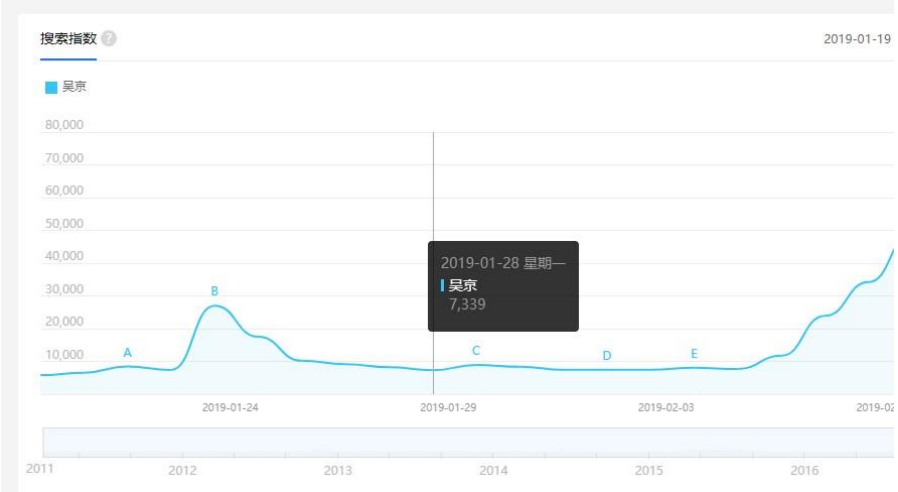

图 2 动态生成的指数显示在小黑框中

我们需要获得的是电影上映之前的各关键词的指数，因此需要调整日期。指数图表下方是一个滑动条，拖动蓝色滑块或直接在滑动条上点击，可以改变所显示的日期和范围

图 3 日期调整滑动条

三、代码说明

代码主要由 6 个函数构成

1、open_browser()

这个函数完成的功能是打开浏览器，进入百度主页并登录账号(百度指数必须登录账号才可以查询)，账号密码事先保存在文件中。由于登录百度账号需要邮箱或手机验证，验证码无法绕过，只能在程序打开浏览器后手动输入

2、check_date()

由于之前获取的电影信息文件中，有些没有日期或日期格式不正确，百度指数的爬取中日期是必不可少的数据，对于日期必须要进行检查，对日期不正确的词条将直接略过不查询指数。

3、calculate_offset()

函数的参数为电影上映日期，格式要求为 YYYY-MM-DD

计算在调整日期时，滑块应停留的位置。滑动条起始处代表的日期为 2011 年 1 月 1 日。

计算方法，1）计算代码运行的日期（datetime.date.today()）与 2011 年 1 月 1 日之间相差的天数 total_diff，2) 计算电影上映日期与代码运行日相差的天数 diff。（1- diff/total_diff）* 滑动条总长度，即为滑块应该停留的位置。

4、get_searched_list()

这是在程序运行中最先调用的函数，读取保存指数的文件，获得已经查询过的电影的列表，第一次运行时返回空列表。由于百度指数爬取较慢，程序运行总时间比较长，可能需要多次启动，在第二次及以后启动时防止重复数据。

5、get_info_list()

从保存电影信息的文件中返回要查询的电影的列表。返回值为一个二维 list，该二维 list 的每个元素是一个包含 [电影名,日期,关键词]的 list。

6、get_index()

获取指数的函数，该函数有两个参数，即 get_info_list()返回的关键词列表 info_list 和 get_searched_list()返回的已经查询的电影列表 searched_list,工作过程如下：

1） 调用 open_browser()打开浏览器，并登录账号。

2） 遍历 info_list，对于每一部电影，执行以下操作：

（1）检查电影名是否在 searched_list 中，调用 check_date()检查日期是否合法,不通过则进入下一个电影执行（1）。通过检查后，执行（2）。

（2）打开新窗口，使用 selenium 的 find_element_by_class 函数定位文字输出框，输入关键字，定位”开始搜索”按钮，点击搜索，等待页面加载。页面加载完成后，使用 selenium 的 find_element_by_name 查找代表指数图表的元素。如果不存在，则表示该词条未被收录，进入下一个电影，返回（1）。否则执行（3）

（3）定位指数图表后，可获得图表的坐标 x、y(该坐标是图表左上角的坐标)。调用 calculate_offset 函数()，计算日期调整滑块的偏移量 offset，由于日期滑动条与指数图表是对齐的且长度相等，所以 x+offset 即为滑块的横坐标；纵坐标为 y+（图表的宽度+空白部分宽度）。由此我们获得了该坐标，移动鼠标至该处并点击，我们便把日期调整到了需要的位置，进入（4）。

（4）创建一个空 dict 用来保存指数数值。模拟鼠标移动：将鼠标移动至图表左则，进入内层循环，循环次数为查询的总天数，每次滑动距离为图表长度/(总天数-1)，每次滑动后停留 1 秒，等待小黑框显示，此时指数已经生成，解析页面可以获得指数的数值，将指数添加到上述 dict 中(key 为天数 1-30，值为指数)。内层循环完毕，使用 csv.Dictwriter 将上述 dict 作为一行写入到 csv 文件中。返回(1),直到列表查询完毕。