**开源软件基础大作业**

**CSDN社区编程语言信息抓取与统计**

**The Subject of Undergraduate Graduation Project (Thesis) of DUT**

学 生 姓 名： 蒋伯扬

学 号： 201892142

学 生 姓 名： 王子承

学 号： 201892048

学 生 姓 名：

学 号：

学 生 姓 名：

学 号：

完 成 日 期： 2021年1月9日

大连理工大学

目 录

[1. 研究背景及主要内容 1](#_Toc61122813)

[1.1. 研究背景 1](#_Toc61122814)

[1.2. 研究内容 1](#_Toc61122815)

[2. 开发环境 1](#_Toc61122816)

[3. 主要代码设计 1](#_Toc61122817)

[3.1. 搜索结果网页数据爬取与存储 1](#_Toc61122818)

[3.1.1. 访问搜索链接 1](#_Toc61122819)

[3.1.2. 动态网页的无渲染抓取 2](#_Toc61122820)

[3.1.3. 关键信息提取 2](#_Toc61122821)

[3.2. 数据分析与统计 3](#_Toc61122822)

[3.2.1. 热词分析 3](#_Toc61122823)

[3.2.2. 分级及可视化 4](#_Toc61122824)

[3.2.3. 统计排行 4](#_Toc61122825)

[4. 成果展示 5](#_Toc61122826)

[4.1. 热词分析 5](#_Toc61122827)

[4.2. 分级及可视化 6](#_Toc61122828)

[4.3. 统计排行 7](#_Toc61122829)

[5. 成果总结 8](#_Toc61122830)

[6. 成员贡献及感想 8](#_Toc61122831)

# 研究背景及主要内容

## 研究背景

中国专业IT社区CSDN (Chinese Software Developer Network) 创立于1999年，致力于为中国软件开发者提供知识传播、在线学习、职业发展等全生命周期服务。CSDN APP是开发者专属移动APP，提供最新技术资讯、开发知识，助力开发者学习和成长，让技术交流更简单。CSDN Blog是致力于为开发者打造专业技术写作、分享与交流平台，为技术人提供全面的资讯与知识交流互动。CSDN技术论坛：全力为IT开发者打造分享技术心得、讨论技术话题的信息平台。CSDN资源下载中心：为IT专业人士提供丰富全面、专业的技术资料分享、交流与下载服务。

在CSDN平台上，有丰富的软件相关知识，从入门级别的教学帖子，到大佬级别引起热议的内容，CSDN上有内容广泛的代码知识，同时也有诸多开源的代码及项目工程。

## 研究内容

针对在CSDN上的搜索结果，进行存储和分析。其中存储采用列表格式，对每条搜索结果的标题、时间、作者、阅读量、点赞量、评论量以及标签进行存储，存储到txt文本文档中。对结果的分析包括关键词/热词词频排序、阅读量、点赞量和评论量的分级可视化统计以及排行榜，并且将结果同样存储到txt中。

# 开发环境

开发环境选取了python语言，安装了matplotlib等库，访问浏览器为chrome。

# 主要代码设计

## 搜索结果网页数据爬取与存储

### 访问搜索链接

由于我们的项目是爬取csdn上关于各个语言的帖子内容等，因此需要访问搜索链接。经分析，搜索链接里包含了搜索的内容和页码，因而我选择将这两部分用%language%与%page%进行填充，在实际访问时使用replace进行具体内容的替换。因而只需要预先在列表里输入好要获得的语言数据，再选定要爬取的页码，通过两层for循环即可。

overall\_url = "https://so.csdn.net/so/search/all?q=%language%&t=all&p=%page%&s=hot&tm=0&lv=-1&ft=0&l=&u="

language\_list = ["python", "c++", "C", "java", "objective-c",

                 "php", "basic", "ruby"]

代码块3.1 替换搜索链接

在分析得到的bs4文本内容时，通过审查元素得知，需要提取的内容都含有"data-v-08755ee8"的标签，再结合诸如标题、作者等标签即可得到想要的内容。而在实际爬取的时候会发现有些帖子经常报错，经分析得知当没有回复、点赞数，或者帖子为下载内容，不存在浏览量时，对应的标签都为空，应当进行一些特殊处理。

### 动态网页的无渲染抓取

由于CSDN网页的特殊性，使用request发送请求经过分析后发现并未得到所需要的结果。通过对页面审查元素，得知网页采用了大量的javascript，属于动态页面，通过简单的抓取并不能执行代码块。因此通过安装python中selenium包并使用了其中的webdriver组件。从而在chrome上实现后台自动跳过渲染网页并爬取信息，在页面加载之后将网页信息全部抓取，以便后处理

chrome\_driver = 'chromedriver.exe'

    driver = webdriver.Chrome(

        executable\_path=chrome\_driver, chrome\_options=option)

*# driver.implicitly\_wait(50)*

    driver.get(url)  *# 请求页面，会打开一个浏览器窗口*

    time.sleep(1)  *# 加这句等待时间是防止页面未加载完成导致数据丢失，可酌情修改   【实际经常出现来不及加载完的情况】*

    html\_text = driver.page\_source

代码块3.2 动态网页抓取

通过使用的是chrome浏览器的内核。得到页面数据后再通过beautifulsoup即可提取标签。而由于这种方式渲染需要额外的加载时间，而且页面的某些元素加载可能耗费更久，因而添加了等待时间以保证页面能渲染完毕。当等待时间较短或网络情况较差时，很可能出现丢失信息的情况。

### 关键信息提取

由于CSDN只展示搜索结果的前20页，而每页有30条信息，因此对于每个搜索关键词，只能有至多600条搜索结果。我们测试用的关键词分别为"python", "c++", "C", "java", "objective-c","php", "basic", "ruby"，目的是针对这些语言分析出论坛上对于各个语言的热度和语言之间的信息比较。对网页的源数据进行分析，可以找到时间、标题、作者、阅读数、点赞数、评论数、标签等信息的标签，从而对这些数据进行分析和提取。

nrb\_time = list\_item[i].find(

            "span", {"data-v-08755ee8": "", "class": "nrb-time"}).text  *# 时间*

代码块3.3 时间信息提取

而有些数据，例如浏览量和点赞量，有些帖子并没有这些信息，所以要针对这些信息进行空值处理，防止存储环节出现问题，以下是以浏览量为例子的空值处理：

        nrb\_view = ""

*if* len(list\_item[i].find\_all("span", {"data-v-08755ee8": "", "class": "nrb-view"})) > 0:

            nrb\_view = list\_item[i].find(

                "span", {"data-v-08755ee8": "", "class": "nrb-view"}).text  *# 浏览量*

*else*:

            nrb\_view = "0"

代码块3.4 空值信息处理

对于数量不定的标签，需要采取另一种灵活的处理方式，针对不同长度进行变值存储。

nrb\_tags\_ = list\_item[i].find(

            "span", {"data-v-08755ee8": "", "class": "nrb-tags"})

        nrb\_tags\_list = nrb\_tags\_.find\_all("span", {"data-v-08755ee8": ""})

        nrb\_tags\_num = int(len(nrb\_tags\_list)/2)  *# 标签数量*

        nrb\_tags = []  *# 标签列表*

*for* j *in* range(nrb\_tags\_num):

            nrb\_tags.append(nrb\_tags\_list[j\*2].text)

代码块3.5 标签信息处理

对所有信息爬取之后，由于浏览量在超过10000之后会显示“1万+”的字样，无法进行排序或以int类型存储，因此还需要进行进一步处理

*if* nrb\_view[-3] == "万":

            nrb\_view = nrb\_view[1:-3]

            nrb\_view = int(float(nrb\_view)\*10000)

*else*:

            nrb\_view = int(nrb\_view[1:-1])

代码块3.6 字符串信息处理

最终将爬取并处理好之后的数据整合至列表中，再用.write()函数以列表的格式写入到文件中即可。

## 数据分析与统计

### 热词分析

对于每个语言都有其相关的讨论热点，分析标题中的词频即可分析出在此语言领域的讨论热度。对于英语可以直接对单词的频率进行分析，但是中文词组具有更复杂的联系性，因此选取正则表达式对中文标题进一步处理。

words = re.split('\W|\_', line)

*for* word *in* filter(lambda word: word.lower() *not* *in* '', words):

                d[word] = d.get(word, 0)+1

代码块3.7 正则处理中文词组词频

对每个中文词组进行计数，对于每个搜索的语言，分别将关键词存储到以语言命名的文件夹下。

### 分级及可视化

对于每个语言，对浏览量、点赞量和评论量进行分级，以浏览量为例，将浏览量分为四个级别，一级为浏览量大于10000，二级为小于10000并且大于1000，三级为小于1000并大于100，四级为小于100。统计每种语言的帖子在每个级别中的数量，进行统计。

temp = [0, 0, 0, 0]

    path = language\_list[i]+".txt"

    result.write(language\_list[i]+" statistics:")

*#print(language\_list[i]+" statistics:"+"\n")*

*with* open(path, encoding='utf-8')*as* f:

*for* line *in* f:

            temp\_list = eval(line)

            view = int(temp\_list[3])

*if* view > 10000:

                temp[0] += 1

*elif* view > 1000:

                temp[1] += 1

*elif* view > 100:

                temp[2] += 1

*else*:

                temp[3] += 1

代码块3.8 正则处理中文词组词频

统计各语言数据之后，使用matplotlib库，对数据进行可视化，并且进行充分的标注和表明，使统计结果直观易见。

plt.xlabel("Views level")

plt.ylabel("number of article")

plt.title("Diffrent languages' articles view level")

plt.text(1, 300, "Level1:view<100\nLevel2:100<view<1000\nLevel3:1000<view<10000\nLevel4:view>10000")

plt.show()

代码块3.9 利用matplotlib库进行绘图

### 统计排行

对每种语言，分别统计其内部浏览量、点赞量、评论量的排行，并且对于所有的搜索结果进行总榜的排序

    view\_rank = sorted(list, key=lambda list: list[3], reverse=True)

    like\_rank = sorted(list, key=lambda list: list[4], reverse=True)

    comment\_rank = sorted(list, key=lambda list: list[5], reverse=True)

代码块3.10 排行排序

# 成果展示

## 热词分析

对于每种语言出现频率最高的前30个词组进行展示，下面以python的热词结果为例：

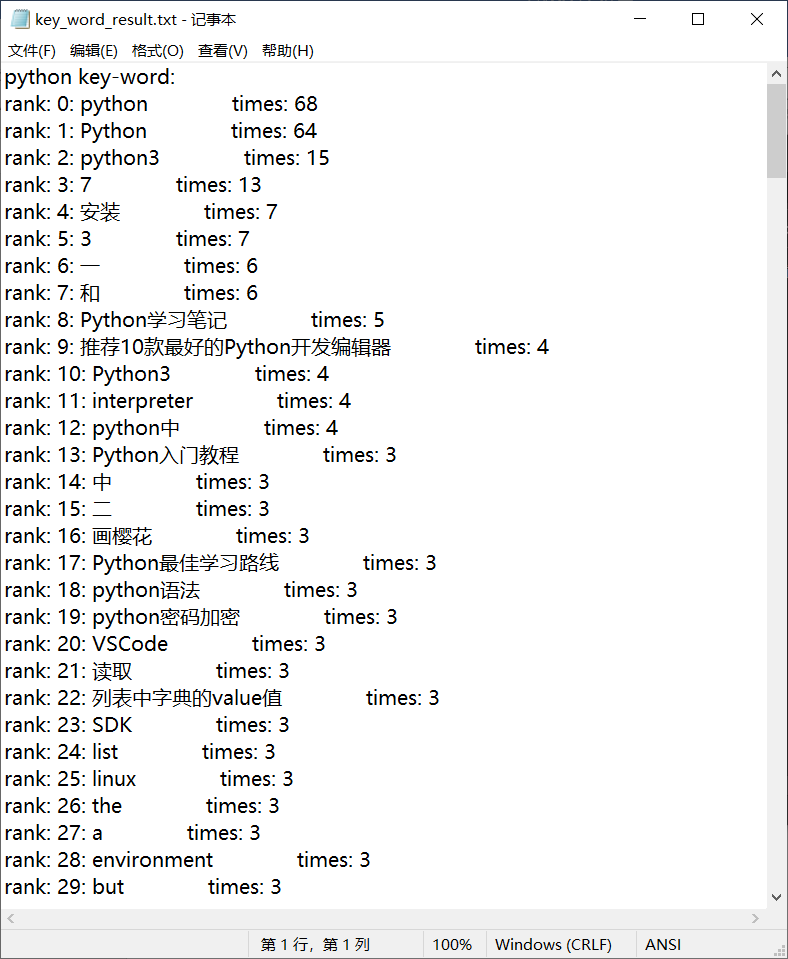


图4.1 python热词统计展示

## 分级及可视化

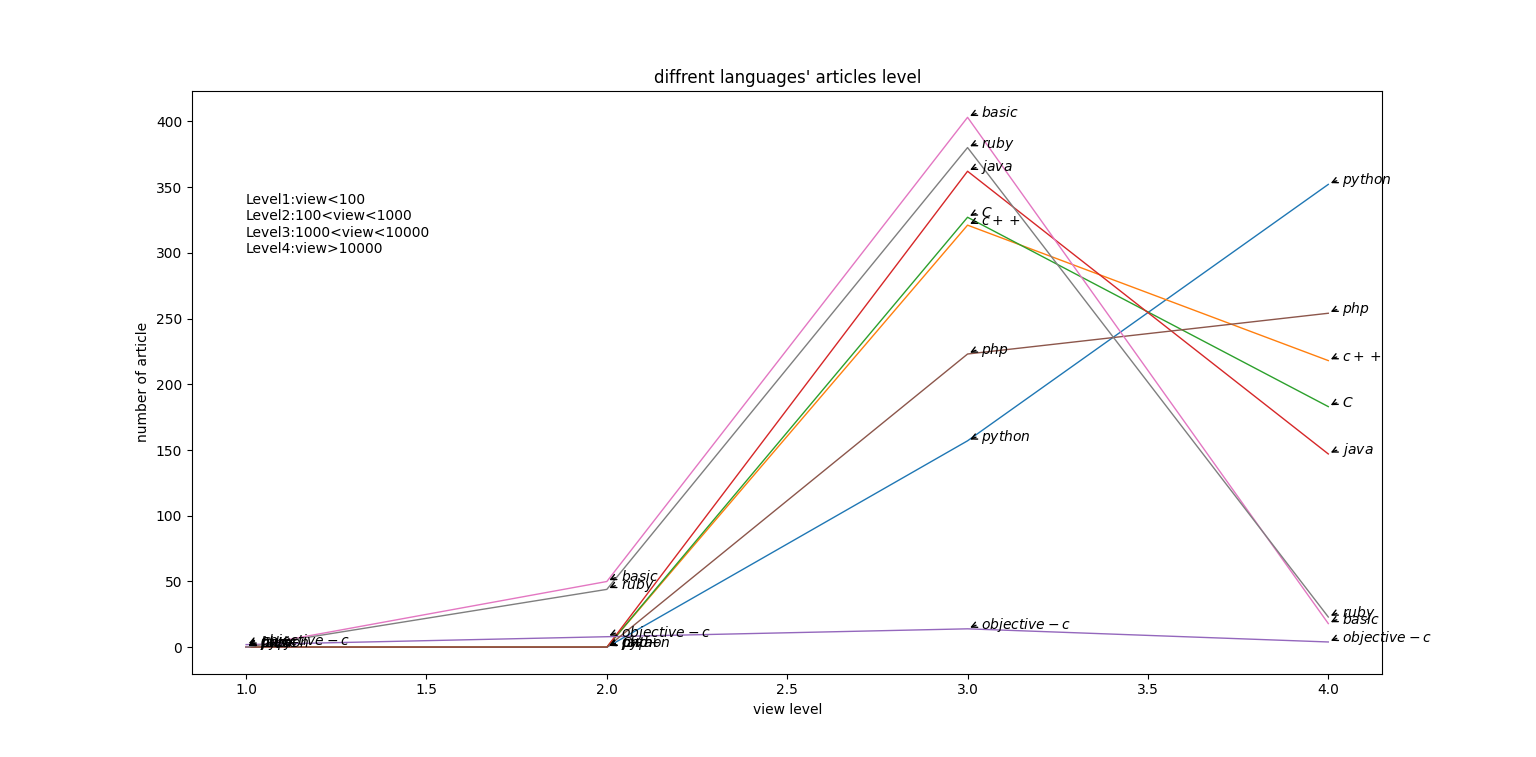
对于每种语言，将浏览量、点赞量和评论量进行分级和可视化展示，结果如下：（请放大查看图片）。

图4.2 浏览量分级可视化

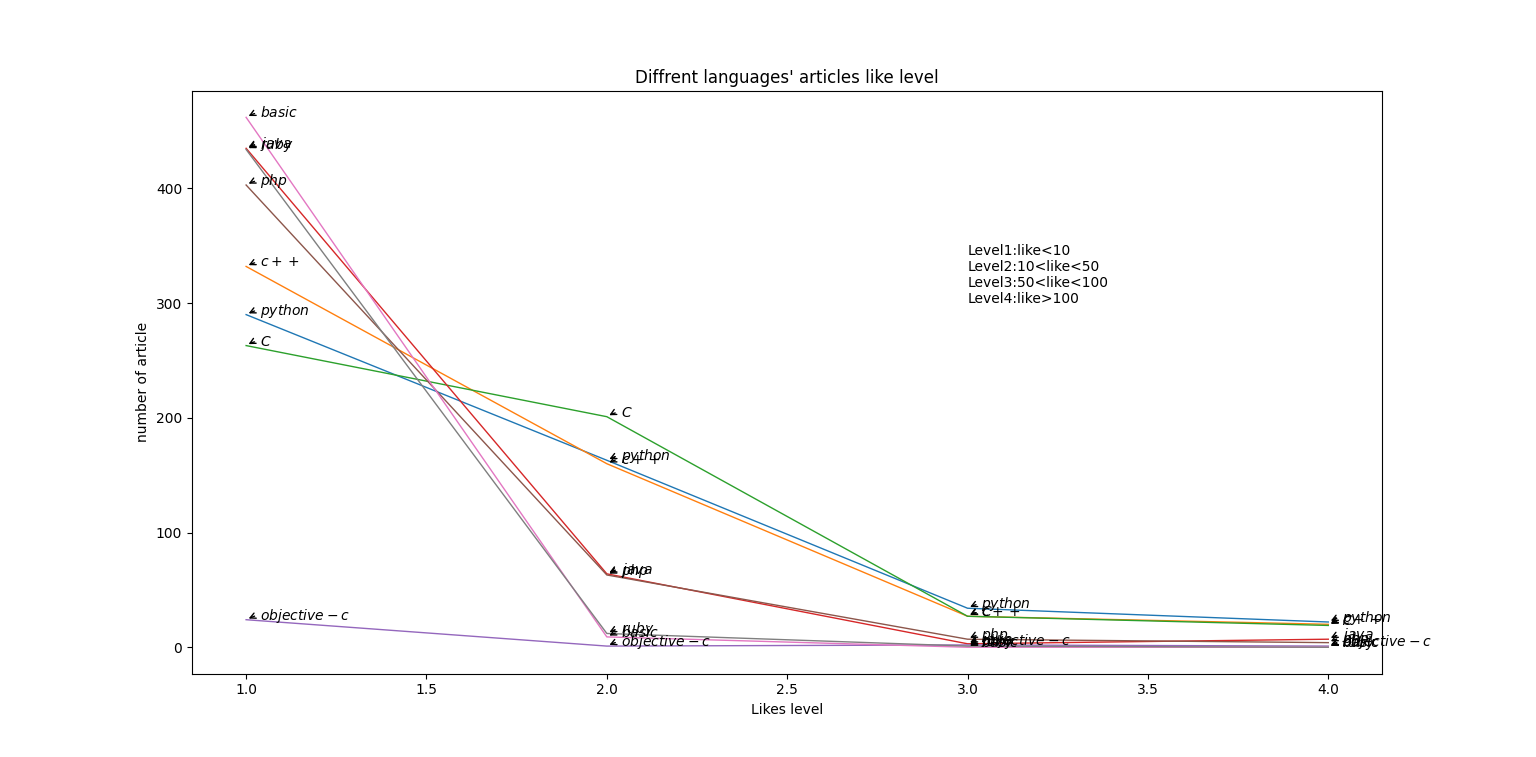


图4.3 点赞量分级可视化

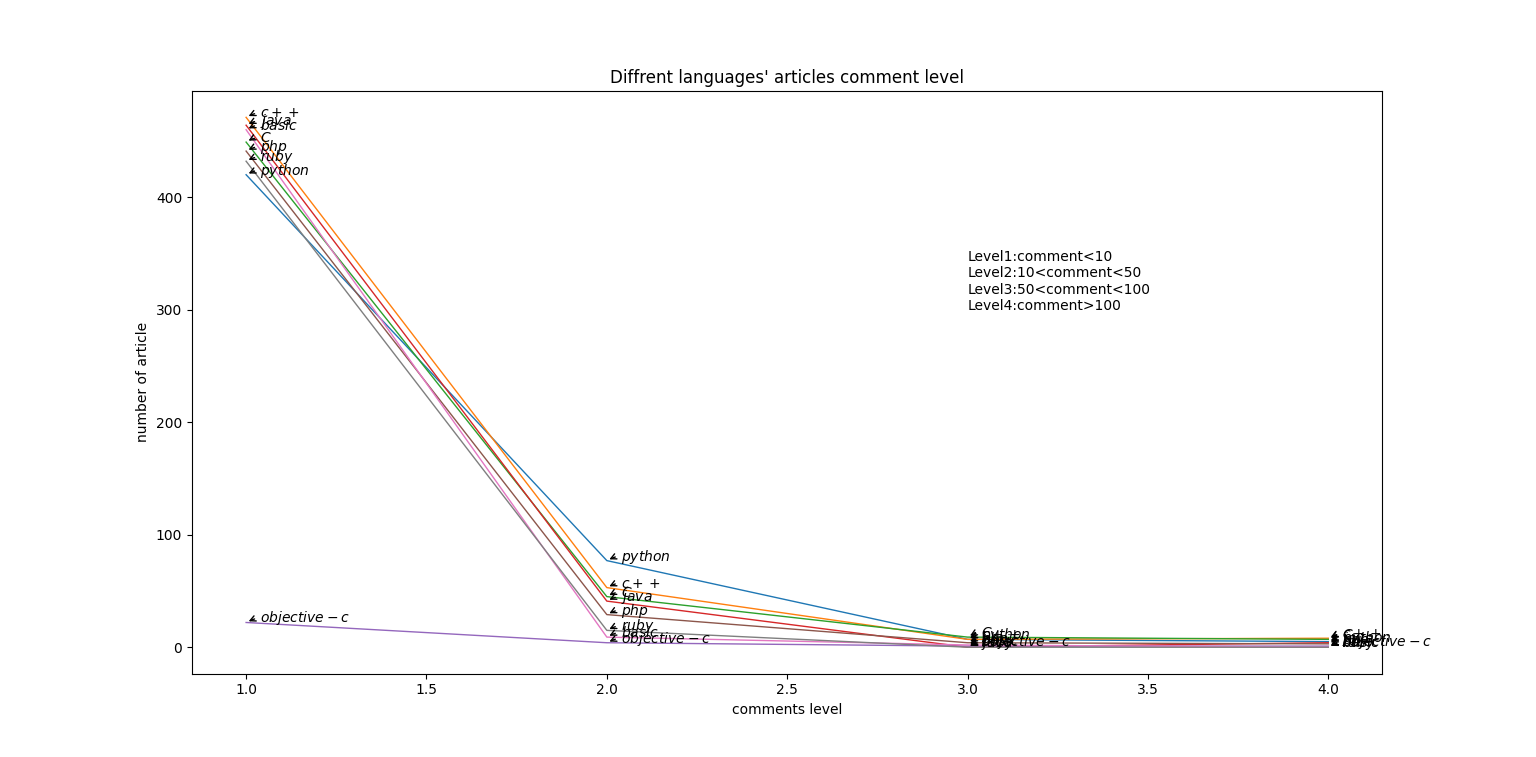


图4.4 评论量分级可视化

## 统计排行

对每种语言，分别统计其内部浏览量、点赞量、评论量的排行，并且对于所有的搜索结果进行总榜的排序。部分结果如下

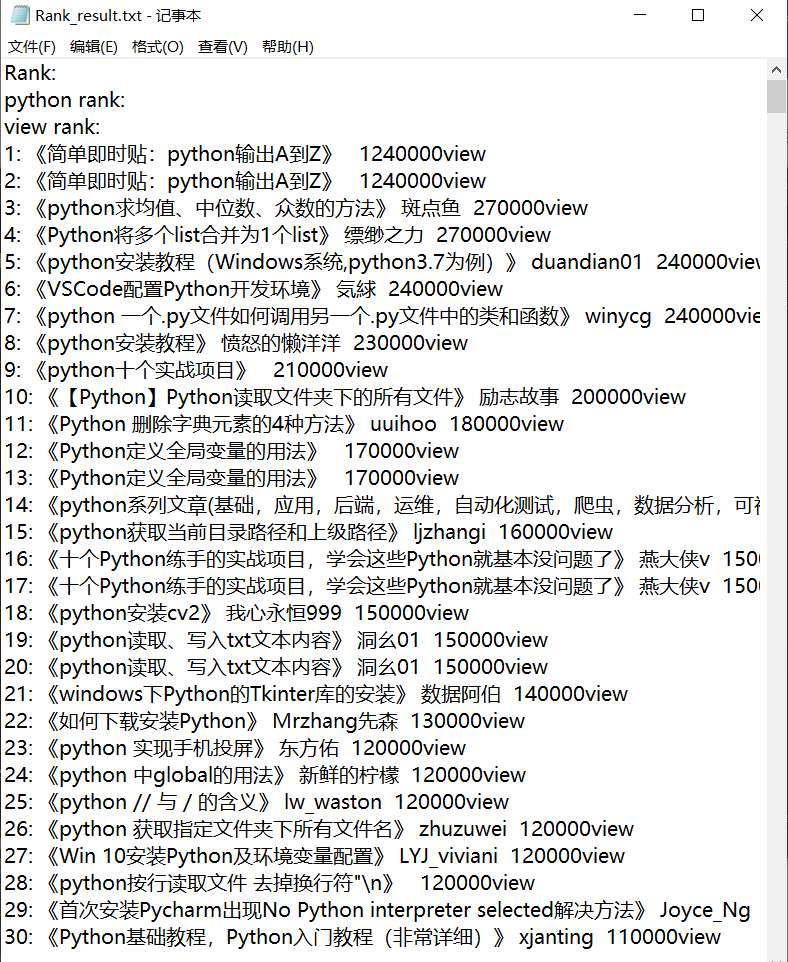


图4.5 python语言搜索结果浏览量排行

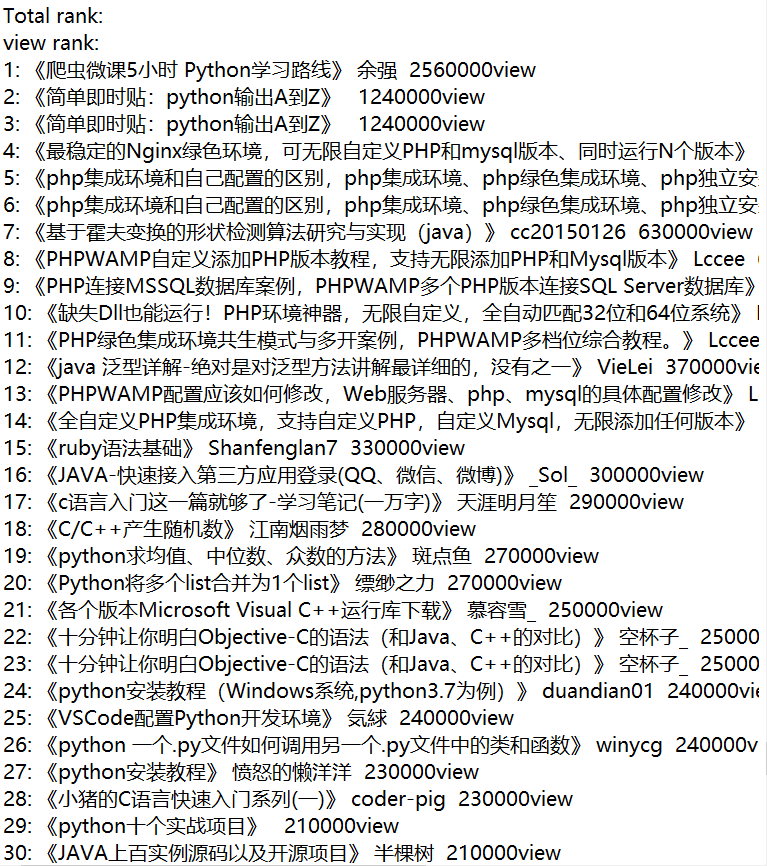


图4.6 各语言搜索结果浏览量总排行

# 成果总结

本程序能够针对CSDN上的搜索结果进行数据爬取和数据处理。可以看出本程序对于动态页面的处理十分到位。另外对于数据的处理非常直观和全面。但是同时也有一些不足，比如可以针对数据进行更为深入的分析，以及对更多页面统计数据，这样可以提高结果的信服度和丰富的。如果后续能够有更多改进将会使得程序更加完整。

# 成员贡献及感想

（1） 201892048 王子承（组长）

**工作占比：50.00%**

主要工作内容：爬取页面和数据提取

感想：在这次项目的开发中，我们所面对的网站与课上所学的有所不同，服务器不是直接发送完整的页面，而是发送包含了js脚本的动态页面，需要在本地进行运行和页面渲染，因而我使用了selenium库，并且通过调用浏览器内核的方式来进行本地渲染，虽然比直接爬取要慢得多，但是这种方法能应对绝大多数的情况。从中我不仅仅是学到了一个库，而是学到了一种思想，遇到问题要尝试从另一个角度去解决。此外，在编程中也遇到了诸多细节方面的问题，包括浏览器内核会经常弹出无关紧要的调试信息，影响结果的分析等，通过查阅多方的资料和论坛都是能解决的。在平时的开发中，学会去查阅资料，学习别人的经验也是很重要的环节。

（2） 201892142 蒋伯扬（组员）

**工作占比：50.00%**

主要工作内容：数据分析处理

感想：在数据的处理过程中，可以引用多方面库，尤其是在热词分析的过程中，能应用到神奇的正则表达式对中文词组进行处理是学习到的一个新知识。另外在对数据进行可视化处理的过程中，学习到了matplotlib库等其他库，这些库能够很简便的对数据可视化，除此之外还有许多强大的功能，也学到不少。在编写代码的过程中，查阅相关资料、处理编码中遇到的bug和相关问题，让自己收益匪浅。尤其是在CSDN查阅问题的过程中，有很多IT大佬的帖子整理得非常全面，能够在短时间内提高自己的代码能力，很多人也愿意公开自己的代码，让其他人学习。整个过程让我领悟到，开源代码的确是促进整个行业发展的重要推力。