News Crawling Based on Python Crawler

Jihou Wang *, Qintao Li, Jiaqian Zhou, Yulian Wen, Yan Zhang, Peiguang Lin

School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan 250014, China

*w17853131657@163.com

Abstract. News is an important form to reflect current politics, which attracts people's attention. The emergence of crawlers provides a convenient way for people to obtain useful information from mass news. Through Python language, this paper takes the relevant news of business logistics hub as an example to carry out the experiment of network news crawling. This paper introduces the workflow, design and implementation of the crawler in detail. The experiment proves that the designed crawler can get news quickly and can provide people with the information they need.

Keywords: News, Logistics, Python, Crawler

1. Preface

News, as an important form of reflecting social events, is of great significance for the people to understand the development of the social state. With the development of the Internet era, the carrier of news has been transferred from newspapers and television to the network, and the content of news media has become rich and diversified. How to find useful news from large-scale and diverse news has become the focus of attention. The emergence of web crawlers can help people obtain useful news from a large amount of data[1-3].

This article implements a web crawler through Python language. This article selects five business service logistics hubs in Linyi, Shanghai, Ganzhou, Yiwu, and Shenzhen, and press releases from 17 years to the present as the crawl content. Select CCTV, CNR News and China News Network , crawl the content of the news articles in the website.

2. Technical introduction

2.1. Crawler design
The crawler designed in this paper is based on Python language[4], running in Windows system, and using the function of Google browser to view the source code of webpage to modify the crawler. The implementation of the crawler needs to import a variety of function packages, including requests package, lxml package, csv package and pandas package. Among them, requests package is mainly used to send a request to the webpage address to obtain the source code of the address page; lxml package is mainly used to parse the webpage source code to obtain the data to be crawled; the csv package is mainly used to write data to the CSV file and save the data; pandas package is mainly used to import the data in the CSV file.

2.2. Crawler workflow
The workflow of the crawler in this article is shown in Fig.1. Firstly, the crawler obtains the source code of the webpage according to the webpage link. Then the source code of the webpage is parsed, and the webpage link to the press release is stored in the specified file. Then read the webpage links in the file one by one to obtain the source code of the webpage where the press release is located. Finally, the source code of the webpage is parsed, and the obtained news content is stored in a file to complete the crawling.

![Fig.1 Crawler workflow](image_url)
3. Crawler implementation
This section takes CCTV network as an example. Introduce the crawling process.

3.1. Get news search page source code
Use get (URL, headers) in the requests package to send a request to the web page pointed to by the URL to obtain the source code of the web page, and name the obtained content as response. Convert the response information into json format by calling response.text function. In order to simulate browser access, add specified headers, this article chooses to simulate Google Chrome. The code is shown in Fig.2.

```python
def get_page(url):
    headers = {
        'USER-AGENT': 'Mozilla/5.0 (Windows NT 10.0 X86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.182 Safari/537.36'
    }
    response = requests.get(url=url, headers=headers)
    html = response.text
    return html, response.status_code
```

**Fig.2** Get (URL, headers) code implementation

Since the news search page only displays 10 pieces of information per page, it is not possible to crawl all the links in a url. Therefore, this article simulates the page turning of the website through a for loop. The code is shown in Fig.3.

```python
for i in range(1, 11, 1):
    url = 'https://search.cctv.com/search.php?page=' + str(i)
    html = get_page(url)
```

**Fig.3** For loop simulation website page turning

3.2. Analysis of web source code
Using the etree.HTML(html.text) function to parse the web source code to get the element object. After that, use xpath() to filter the element object to get the required data. Because the source code on each website is different, the parameters of xpath() need to be adjusted according to the source code of a web page.

What we need to crawl is the title, the link and the release time. Where the title is under the a tag whose id value contains web_content_. The link is in the lanmul in the span tag under the h3 tag. The publishing time is in the span tag with a class value tim. The parsing code is shown in Fig.4.
4

3.3. Storage of crawling data
First, open the file to be saved by using the open function. The parameter 'a' represents the opening and writing. If there is already content in the file, it will be attached at the end of the file. As shown in Fig.5.

```python
fd = open('yangshi-linyi-href.csv', 'a', encoding='utf-8', newline='')
```

**Fig.5** The open function opens the file code

Using the csv.writer().writerow() writes data to the CSV file. As shown in Fig.6.

```python
writer = csv.writer(fd)
for item in data:
    writer.writerow(item)
fd.close()
```

**Fig.6** Write data code to the CSV file

After all data is written, close the file using the close() function.

3.4. Get the source code of news page
First of all, use the read_csv() function in pandas function package to import the data in the CSV file, and assign the previously crawled link to the url. Then, through get(url,headers) in the requests package, a request is sent to the web page pointed to by the url to obtain the source code of the web page, and the content obtained is named response. The code is shown in Fig.7 and Fig.8.

```python
data = pandas.read_csv('yangshi-linyi-href.csv', header=None)
for item in data[0]:
    url = item
    html, response_code = get_page(url)
```

**Fig.7** Import data code
def get_page(url):
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0)
    }
    response = requests.get(url, headers=headers)
    response.encoding = response.apparent_encoding
    html = response.text
    return html, response.status_code

3.5. Access to news content
Using the etree.HTML(html.text) to parse the web source code. After that, use XPath () to filter to get
the data to be crawled, including the title, source and body content of the news. The code is shown in Fig. 10.

html = etree.HTML(HTML)
title = html.xpath('//h1/text()')
source = html.xpath('//meta[@name="source"]/text()')
content = html.xpath('//div[@class="content_area"]/text()')

If the above code does not find the relevant content, it will be stored as 404, and the content that has
not been crawled can be manually crawled as required. The code is shown in Fig.11.
if title==[]:
    data = zip(['404'], ['404'], ['404'])
    return data
if source==[]:
    data = zip(['404'], ['404'], ['404'])
    return data
if content==[]:
    data = zip(['404'], ['404'], ['404'])
    return data

Fig.11 Code after crawling failure

3.6. Storages of news data
Use csv.writer().writerow() writes the crawled data in 3.5 to the CSV file. This is the end of the crawl process.

4. Experimental result
This paper has 159 news about Linyi national logistics hub for business service, 872 news on Shanghai national logistics hub for business service, 321 on Yiwu national logistics hub, 122 on Ganzhou national logistics hub for business service, and 240 on Shenzhen national logistics hub. (Tab.1)

|                | Linyi | Shanghai | Shenzhen | Yiwu | Ganzhou |
|----------------|-------|----------|----------|------|---------|
| CCTV           | 95    | 290      | 30       | 245  | 90      |
| CNR News       | 59    | 571      | 206      | 74   | 32      |
| China News Network | 5    | 11       | 4        | 2    | 0       |
| Total          | 159   | 872      | 240      | 321  | 122     |

Tab.1 Results of crawler experiment

5. Conclusion
At present, in the era of big data, information has a considerable role. Information has the characteristics of large quantity, rich content and different quality of information, so how to quickly obtain a large number of relevant information from news websites has become a problem to be solved. This paper designs a crawler program by using Python language to crawl the logistics situation of Shenzhen in Linyi, Shanghai, Yiwu, Ganzhou and Shenzhen. In the process of crawler getting data, the code still has some room for improvement. For example: during the running process, some news content did not crawl down due to the different format of the web source code. In the future, improvements will be made to the defects of the crawler.

References
[1] Can L, Jianyu R, Wei L and Qiangqiang Z 2018 Personalized recommendation oriented crawling and displaying system of educational news Software Engineering 40 38-40+34
[2] Zhenliao L, Yingying Z and Yanjing W 2013 Intelligent grabbing technology for news network Control Engineering of China 20 707-710
[3] SiWen C and Haiyan L 2019 Web news oriented crawler development and hot news event analysis Geomatics & Spatial Information Technology 42 110-113+118

[4] Chengcheng W 2018 Data information crawler technology based on Python Electronics World 03 208-209