Design of Data Crawling System Based on Python - Taking House Information Crawling as an Example

Hongxia Mao

School of Computer and Software, Jincheng College, Sichuan University, Chengdu 611731, Sichuan, China

Abstract: ***The extensive application of Internet technology has led to the explosive growth of network resources. Finding the required data in the massive data is a time-consuming and labor-intensive thing. Housing information is one of the hot topics of national concern, and the use of web crawling technology can quickly and accurately obtain housing information from various platforms. This article uses Python language combined with web crawling technology to design a house information data crawling system, which includes modules such as URL manager, webpage download, webpage analysis, data collection, and data storage. Successfully saved the house information and pictures on the target website through the operation of the system.***

Keywords: Python; Data crawling; Anti crawling strategy.

1. INTRODUCTION

With the rapid development of Internet technology, information technology has developed rapidly, especially the rapid growth and accumulation of data resources on the Internet at a huge speed, and the explosive growth of network resources [1]. It will be more and more difficult to quickly and accurately find valuable information in massive data. Therefore, web crawlers have emerged, which can accurately and efficiently crawl the required data from target websites according to their own needs [2]. Data crawling can bring some burden to websites, so different websites have adopted corresponding anti crawling strategies. The data crawling system periodically analyzes the target website to study the anti crawling mechanism, in order to ensure that the data crawling system can operate normally and crawl the required data [3]. Wu, Z. et al. (2024) introduces an improved Markov model with enhanced state transition mechanisms and multimodal data integration, applicable in any predictive modeling tasks involving complex sequences and dynamic data [4]. Wu, Z. (2024). introduces a novel combination of REEGWO, CNN, and BiLSTM, significantly improving the optimization of deep learning parameters, applicable in fields requiring advanced time series forecasting [5].

At present, housing information is an important topic of concern for the people. People have a high enthusiasm for paying attention to the prices of new houses, second-hand houses, and rental houses. However, major platforms have data barriers that cannot cover all housing information [6-7]. Therefore, building a system that can crawl housing information on the internet is particularly important. This article uses Python as the programming language for the data collection system to crawl house information from the network [8-14].

2. CRAWLER PRINCIPLE

A web crawler is a program whose main purpose is to download web pages on the Internet to the local and extract relevant data. Web crawlers can automatically browse information on the internet and then download and extract the necessary data according to specified rules. The architecture of a basic web crawler is shown in Figure 1.

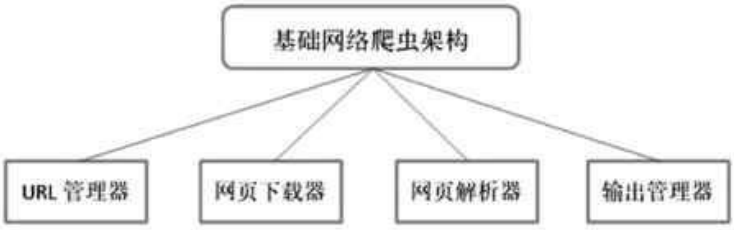

Figure 1: Architecture of Basic Web Crawler

URL Manager: Manage the URLs to be crawled to prevent duplicate crawling and circular crawling.

Web page downloader: This is a component for downloading web pages. It is used to download the web pages corresponding to the URLs on the Internet to local places. It is one of the core parts of the crawler.

Web parser: This is a component that parses web pages, used to extract valuable data from web pages, and is another core part of web crawlers.

Output Manager: This is a component that stores information and is used to output parsed content to files or databases.

3. DESIGN OF BUILDING DATA CRAWLING SYSTEM

This article focuses on crawling house data from Q House website. The data crawling system will crawl the basic information of website links, second-hand houses, rental houses, and new houses in various cities on Q House website, and save the crawled house information.

3.1 Analysis of Web URL Management

Open Q House website using Google Chrome, enter the second-hand housing link, click on the location to obtain the URL of the corresponding city, try to change the city, and observe that the URL of the corresponding second-hand housing website is regular. As shown in Figures 2 and 3:

Figure 2: URL of Beijing website

Figure 3: Website URL in Foshan area

Changing cities, it was found that the URL setting pattern of the website is: https://city pinyin qfang. com/sale。

Q House website can display up to 30 housing information on the corresponding webpage for each city. Click on the next page, and the webpage URL will change accordingly. The pattern of change is: the URL for each city entering is: ht tp://foshan qfang. com/sale,

Click on the second page URL as follows: http://foshan. qfang. com/ sale/f2。 As shown in Figure 4.

Figure 4: Housing in Foshan Area Page 2

Therefore, the construction rule of the URL on page n is: http://foshan. qfang. com/sale/fn。

Therefore, the code for the URL for flipping house information on Q House website is:

```
pre_url = 'http://foshan.qfang.com/sale/f'
for x in range(1,11):
    url = pre_url + str(x)
```

3.2 Web Page Downloads

Requests is a library that is used when writing crawler code. Requests inherit all the features of urllib2. Re questions supports HTTP connection maintenance and connection pooling, session maintenance using cookies, file uploading, automatic encoding of response content, and automatic encoding of internationalized URLs and POST data. The code to download a webpage using Requests is as follows:

```
pre_url = 'http://foshan.qfang.com/sale/f'
for x in range(1,11):
    url = pre_url + str(x)
    html = requests.get(url, headers=headers)
```

3.3 Web Page Parsing

By using the developer mode of Google Chrome, you can locate the location of the data to be crawled in the web page source code and extract the XPath path, as shown in Figure 5.

Figure 5: Front end code for website housing information

Call the spider function to obtain the house information of the corresponding page. The code is shown in Figure 6:

```
def spider(url):
    '''爬虫函数'''
    selector = download(url)

    house_list = selector.xpath("//*[@id='cycleListings']/ul/li")
    for house in house_list:
        xiaoqu = house.xpath('div[1]/p[1]/a/text()')[0]
        huxing = house.xpath('div[1]/p[2]/span[2]/text()')[0]
        mianji = house.xpath('div[1]/p[2]/span[4]/text()')[0]
        weizhi = house.xpath('div[1]/p[3]/span[2]/a[1]/text()')[0]
        zongjia = house.xpath('div[2]/span[1]/text()')[0]
        #构造详情页URL
        house_url = ('http://beijing.qfang.com'
                     + house.xpath('div[1]/p[1]/a/@href')[0])
        sel = download(house_url)
        house_year = sel.xpath("//div[@class='housing-info']/ul/li[2]/div/ul/li[3]/div/text()")[0]
        mortgage_info = sel.xpath("//div[@class='housing-info']/ul/li[2]/div/ul/li[5]/div/text()")[0]
        #构造要写入文件的数据项
        item = [xiaoqu,huxing,mianji,weizhi,zongjia,house_year,mortgage_info]
        #写入文件
        data_writer(item)
        print('正在抓取',xiaoqu)
```

Figure 6: Code for crawling house information

3.4 Output Save

Save the crawled property information to a CSV file named afang _foshan, as shown in Figure 7.

```
def data_writer(item):
    with open('qfang_foshan.csv', 'a',encoding='utf-8',newline='') as csvfile:
        writer = csv.writer(csvfile)
        writer.writerow(item)
```

Figure 7: Code for saving housing information

Save the image information of the house in binary format, as shown in the code in Figure 8.

```
def image_saver(url,xiaoqu):
    '''
    图片保存函数
    param url: 图片网页URL
    param xiaoqu: 图片小区名称
    :return: 无
    '''
    img = requests.get(url,headers = headers)
    with open('./Qfang_image/{}.jpg'.format(xiaoqu),'wb') as f:
        f.write(img.content)
```

Figure 8: Code for saving property images

4. COPING WITH ANTI CRAWLING STRATEGIES

Websites usually implement anti crawling mechanisms against web crawlers, with the most common anti crawling techniques being using headers and user behavior based anti crawling.

The most common anti crawling strategy is to extract headers from user requests. Many websites will detect the user agent of headers. In response to this anti crawling mechanism, this article adds headers to the crawler code when designing the crawler code, assigning the user agent of the browser to the headers of the crawler, so that the website server can recognize the client's operating system and version, CPU type, browser and version, browser rendering engine, browser language, browser plugins, etc., in order to respond to anti crawling strategies.

The website crawled in this article determines whether it is a crawler by detecting user behavior and performing multiple similar operations on the same account in a short period of time. In response to this situation, control the crawler code to randomly wait a few seconds after each request before making the next request.

5. CONCLUSION

This article studies the design of a data crawling system using Python language. Through the implementation of relevant crawling techniques and code, the house data of Q House website is crawled. The design of a data crawling system has been implemented, covering web URL management, web download, web analysis, data extraction, data saving, image saving, and handling of website anti crawling mechanisms.

REFERENCES

[1] Iquebal, A. S., Wu, P., Sarfraz, A., & Ankit, K. (2023). Emulating the evolution of phase separating microstructures using low-dimensional tensor decomposition and nonlinear regression. MRS Bulletin, 48(6), 602-613.

[2] Wang, W., & Osaragi, T. (2024). Lognormal distribution of daily travel time and a utility model for its emergence. Transportation research part A: policy and practice, 183, 104058.

[3] Z. Ren, "A Novel Feature Fusion-Based and Complex Contextual Model for Smoking Detection," 2024 6th International Conference on Communications, Information System and Computer Engineering (CISCE), Guangzhou, China, 2024, pp. 1181-1185, doi: 10.1109/CISCE62493.2024.10653351

[4] Wu, Z., Wang, X., Huang, S., Yang, H., & Ma, D. (2024). Research on Prediction Recommendation System Based on Improved Markov Model. Advances in Computer, Signals and Systems, 8(5), 87-97.
[5] Wu, Z. (2024). presents an innovative integration of the REEGWO algorithm with CNNs and BiLSTM networks, enhancing deep learning model optimization, which can be applied to other areas requiring improved hyperparameter tuning and sequential data prediction.
[6] Shen, Z. (2023). Algorithm Optimization and Performance Improvement of Data Visualization Analysis Platform based on Artificial Intelligence. Frontiers in Computing and Intelligent Systems, 5(3), 14-17.
[7] Ji, H., Xu, X., Su, G., Wang, J., & Wang, Y. (2024). Utilizing Machine Learning for Precise Audience Targeting in Data Science and Targeted Advertising. Academic Journal of Science and Technology, 9(2), 215-220.
[8] Ma, Y., Shen, Z., & Shen, J. (2024). Cloud Computing and Hyperscale Data Centers: A Comparative Study of Usage Patterns. Journal of Theory and Practice of Engineering Science, 4(06), 11-19.
[9] Yuan, B., & Song, T. (2023, November). Structural Resilience and Connectivity of the IPv6 Internet: An AS-level Topology Examination. In Proceedings of the 4th International Conference on Artificial Intelligence and Computer Engineering (pp. 853-856).
[10] Yuan, B., Song, T., & Yao, J. (2024, January). Identification of important nodes in the information propagation network based on the artificial intelligence method. In 2024 4th International Conference on Consumer Electronics and Computer Engineering (ICCECE) (pp. 11-14). IEEE.
[11] Lin, Z., Wang, Z., Zhu, Y., Li, Z., & Qin, H. (2024). Text Sentiment Detection and Classification Based on Integrated Learning Algorithm. Applied Science and Engineering Journal for Advanced Research, 3(3), 27-33.
[12] Wang, Z., Zhu, Y., Li, Z., Wang, Z., Qin, H., & Liu, X. (2024). Graph neural network recommendation system for football formation. Applied Science and Biotechnology Journal for Advanced Research, 3(3), 33-39.
[13] Lu, Q., Guo, X., Yang, H., Wu, Z., & Mao, C. (2024). Research on Adaptive Algorithm Recommendation System Based on Parallel Data Mining Platform. Advances in Computer, Signals and Systems, 8(5), 23-33.
[14] Wu, X., Wu, Y., Li, X., Ye, Z., Gu, X., Wu, Z., & Yang, Y. (2024). Application of adaptive machine learning systems in heterogeneous data environments. Global Academic Frontiers, 2(3), 37-50.