Image Information Collection System Based on Python Web Crawler Technology

Dong Jin*
Department of art and design, Taiyuan University, Taiyuan, Shanxi, China
**Corresponding Author.*

Abstract

Collecting data from the Internet is the key to solve the problem of data sources. This paper studies the image information collection system based on Python web crawler technology.This paper studies and develops a data acquisition system based on Python web crawler technology, which realizes the automatic collection of subject data. In this paper, we use urllib, beautiful soup, threading library to design and develop a system model framework including data crawling, exception handling, robots protocol management and multithreading management modules. Through the application of specific cases, this paper introduces the data acquisition process. Experimental data show that compared with the traditional manual data acquisition, the proposed method greatly improves the work efficiency.

Keywords*: Python, web crawler, image information collection, data fusion.*

I. Introduction

Under the background of big data, all walks of life need data support. How to get the data they are interested in in in the vast amount of data? In the aspect of data search, the search engine has made great progress. But for some special data search or complex search, it can not be well completed, the data using search engine can not meet the needs, network security, product research, all need data support [1]. There is no ready-made data on the network, so we need to manually search, analyze, refine and format the data to meet the needs [2-3]. Using web crawler can automatically complete the work of data acquisition and summary, which greatly improves the work efficiency.

Web crawler, also known as web spider, is a program that can automatically extract web information according to established rules [4]. It imitates the browser to send HTTP requests to access network resources and automatically obtain the web data that users need [5-6]. There are some web crawlers of targeted websites, such as QQ space crawler, which can capture 4 million logs, talks, personal information and other data a day; Zhihu crawls for high quality answers on various topics; Taobao product price comparison directional crawler crawls goods, reviews and sales data [7].

Python is an object-oriented, interpretive, high-level programming language with dynamic semantics. Its syntax is simple and clear, and it has rich and powerful class libraries. Python language supports 100000 function libraries covering all fields of information technology. Relying on open source rapid development, it has formed the world's largest programming community. In the programming language ranking released by IEEE in July 2017, python ranks first, and Python based applications are also brilliant in various fields of computer. Python includes excellent web crawler framework and parsing technology [8]. Python language is easy to use and provides crawler related modules such as urllib, requests, beautiful soup, scrapy, etc. Urllib module provides a high-level interface to get data from the world wide web. Requests simulate the browser to automatically send http / HTTPS requests and get data from the Internet. Beautiful soup parses HTML / XML pages to get the data users need. In this paper, we build a picture crawler program based on Python's requests beautiful soup technology to quickly crawl Baidu Post Bar

ISSN: 0010-8189

pictures, and save these pictures in the local, which is convenient for users to browse offline and further use.

II. Working principle of web crawler and function of Python crawler technology module

Web crawler is a program or script that can automatically capture Internet data according to certain rules. The Web crawler starts to analyze the web page from the homepage or designated page of the web site through the network request to obtain the required content, and continues to enter the next web page through the link address in the web page until all the web pages of the web site are traversed or the stop condition set by the crawler is met. The Python language third-party network request library Requests simulates a browser to automatically send HTTP/HTTPS requests and obtain data from the internet [9-10]. The HTML/XML page obtained by BeautifulSoup can capture the required data for users. Beautiful Soup automatically converts the input document into Unicode encoding and the output document into utf-8 encoding, thus saving programming time.

2.1 The working principle of the network crawler

Web crawler crawls the page is to simulate the process of using the browser to obtain the page information. The crawling process generally includes the following four steps:

(1) Simulate a browser to initiate a request: Send a request to the server through the target URL. The request header generally contains the request type, cookie information and browser type information;

(2) Get the response of the server page: under the normal response of the server, the user will receive the response of the requested web page, which generally includes HTML, JSON string or other binary format data (such as video, picture), etc;

(3) Get page content parsing: use the corresponding parser or conversion method to process the obtained webpage content, such as parsing HTML code with webpage parser. If it is binary data (such as video and picture), it will be saved to a file for further processing;

(4) Storage data: the data obtained from web page parsing can be stored in files such as CSV, JSON, text and pictures, or in databases such as SQLite, MySQL or mongodb.

2.2 Python third-party library Requests module

Requests is an HTTP library written in Python language and licensed by Apache 2. Urllib2 module and httplib module in Python standard library provide most of the required HTTP functions. Requests uses urllib3 module, which supports HTTP connection maintenance and connection pool, supports cookie to maintain session, supports file uploading, supports automatic determination of response content coding, and supports automatic coding of internationalized URL and POST data.

Install the Requests module through pip command ($pip install requests). Urllib provides a series of functions for operating URL, and the request module of urllib can easily access and grab URL (Uniform Resource Locator) content. The commonly used function methods of urllib.request module are shown in Table 1. After using the requests method, a response object will be returned to store the contents of the server response, such as r.status_code (response status code), r.text (response body in string mode, which will be automatically decoded according to the character encoding of the response header), r.json (JSON decoder built r.json(Requests), r.content (response body in byte mode, which will automatically decode gzip and reject compression for you), and so on

Table 1 Main methods in urllib.request module

Method	Function
urllib.request.urlopen(url[,data[,proxies]])	Open a URL and return a file object. You can perform file operations such as read (), readLine (), readlines (), leno (), close (), info (), geturl () on the file object.
urllib.request.urlretrieve(url,filename,mine_hdrs)	Download the html file located by url to your local hard disk. Urlretrieve () returns a binary (le name, mine _ hdrs)
urllib.request.Request(url)	Use request to build a complete URL request with added headers information.

2.3 Beautiful Soup module of Python third-party library

Beautiful Soup is an HTML/XML parser written in python, which can process irregular tags and generate parse trees, and provides simple Python functions to navigate, search and modify parse trees. Install ($ pip install beautiful soup 4) beautiful soup module through pip command. BeautifulSoup transforms HTML document into a tree structure, each node is a Python object, and all objects can be classified into four kinds, as shown in Table 2.

Table 2 Objects in Beautiful Soup Module

Target	Function
Tag	Tag object defines a variety of functions and attributes, such as name and attribute. Each object has its own name. The name attribute can be referenced by the comma operator, and the name attribute can be modified by assigning a string directly. Attributes attribute: tag < B class = "test"> this is the test begin... "> has an attribute of" class "with the value of" test ".
NavigableString	The NavigableString class is used to install strings in tags, which are encoded in unicode, and can be replaced by string function replace_with.
BeautifulSoup	The beautifulsoup attribute represents all the contents of a document. It can be used as a tag object and contains a special attribute name with a value of "[document]", such as beautifulsoup (HTML,'html. Parser ') to parse HTML documents.
Comment	Comment property is a special type of NavigableString object. when it appears in HTML document, the Comment object will be output in a special format.

2.4 Advantages of web crawler written in Python

(1) The language is simple, easy to learn and easy to use. Writing a good Python program feels like writing an article in English, although the English requirements are very strict! This pseudo code nature of Python is one of its greatest advantages. It allows you to focus on solving problems instead of understanding the language itself.

(2) It's easy to use and doesn't need heavy ide. Python only needs a sublimetext or a text editor to develop most small and medium-sized applications.

(3) Scrapy is a powerful crawler framework. Scrapy is an application framework written for crawling website data and extracting structural data. It can be applied in a series of programs including data mining, information processing or storage of historical data.

(4) Powerful network support library and HTML parser, using the network support library reqaests, write less code, can download the web page. Using the web page parsing library beautiful soup, we can easily parse each tag of the web page, and then combine with regular expressions to easily grab the content of the web page.

(5) Very good at doing text processing string processing: Python contains common text processing functions, support regular expressions, can easily handle text content.

2.5 Web crawling strategy

In the web crawler system, the URL queue to be crawled is a very important part. The order of the URLs in the URL queue to be crawled is also a very important problem, because it involves the first page to be crawled, and then which page to be crawled. The way to determine the order of these URLs is called a crawling strategy. Web crawling strategies can be divided into depth first, breadth first and best first:

(1) breadth-first search strategy, its main idea is to start from the root node, first traverse the current level of search, then proceed to the next level of search, and so on. This strategy is mostly used on topic crawlers, because the closer the web page is to the initial URL, the greater the topic relevance. (2) Depth-first search strategy. The main idea of this strategy is to find leaf nodes from root nodes, and so on. In a web page, select a hyperlink, and the linked web page will perform depth-first search to form a single search chain. When there are no other hyperlinks, the search ends. (3) The best priority search strategy, which calculates the similarity between the URL description text and the target web page, or the relevance between the URL description text and the topic, and selects the valid URL to crawl according to the set closed value.

2.6 Web crawler module

According to the working principle of web crawler, a general crawler framework is designed, and its structure diagram is shown in Figure 1.

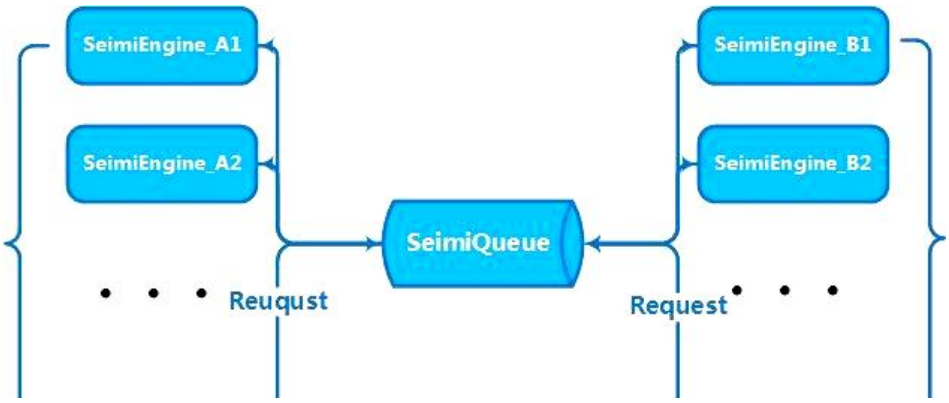

Fig 1:Crawler framework

The basic workflow of web crawler is as follows: (1) firstly, select some carefully selected seed URLs; (2) Put these URLs into the URL queue to be grabbed; (3) Take out the URLs to be crawled from the URL queue to be crawled, download the web pages corresponding to the URLs, send the downloaded web pages to the data analysis module, and then put these URLs into the crawled URL queue. (4) Analyze the webpage data transmitted by the download module, extract the interesting data through regular expression, transmit the data to the data cleaning module, then parse other U RLs in it, and transmit the URL to the URL scheduling module. (5) The U RL

scheduling module receives the U RL data transmitted by the data analysis module, firstly compares these URL data with the crawled URL queue, discards it if it is a crawled URL, and puts the URL into the URL queue to be crawled according to the search strategy of the system. (6) The whole system loops in steps 3-5 until all URLs in the URL queue to be crawled have been completely crawled, or the system actively stops crawling and the loop ends. (7) Organize the cleaning data, and store the data into the database in a standardized format. (8) According to the user's preference, the crawling results are read out from the database and displayed to the user in the form of text and graphics.

III. Post Bar Picture Crawler Program Design

Baidu Post Bar is the largest Chinese community in the world. Post Bar is a theme communication community based on keywords. Post Bar combines with search engine to build an online communication platform, so that people who are interested in the same topic can get together and communicate and help each other conveniently. Design a crawler to crawl the pictures of Meitu Bar in Baidu Post Bar (http://tieba.baidu.com). When running the crawler, prompt the user to input the url of the website to crawl, and modify the request header information. The simulated browser uses get requests for posts in Post Bar in turn. After entering the posts, it finds all the picture tags according to the rules, obtains the url of the picture resources in the posts, and downloads them to the local storage in turn. After crawling all the posts, it presses enter to exit. In the middle of running, it can also use ctrl+c to forcefully exit the program.

Based on Python's requests-BeautifulSoup technology, this paper constructs a picture crawler program, uses Requests to simulate a browser to request a webpage, uses random to generate random numbers to select a simulated browser, uses Python's built-in standard HTML parsing library supported by BeautifulSoup to parse the data returned by the requested webpage, and uses urllib.request.urlretrieve () to download pictures and various network requests.

3.1 Reptile preparation

The development of image crawler program uses Python version 3.6, mainly uses urllib's requests module, beautiful soup module and random module. The module is the definition program file containing variables, functions or classes. Before using the module, import these modules through import. Two global variables, null and true, are defined and initialized to avoid Python reporting errors when null and true appear in the URL.

3.2 Simulate browser to visit website

The crawler simulates the browser sending http / HTTPS requests and getting data from the Internet. User agent is a part of HTTP protocol and a part of request header information. Random is a module of Python standard library. Users can directly call random's method random. RandInt (a, b) to generate an integer within a specified range, where parameter a is the lower limit and parameter B is the upper limit. The lower limit must be less than the upper limit. This function can be used to generate random integers for selecting simulated browsers. When visiting the website, the version and type of browser used by users are provided to the server through the user agent, and the python crawler is simulated as a browser by rewriting the user agent. The following code simulates different browser models and enters Baidu tieba home page.

3.3 Enter the core code of post crawling pictures

The core code of the crawler main program is as follows: access the website to obtain web page data by using urllib's requests and urlopen () methods, parse the obtained web page data with BeautifulSoup's find_all (), grab the picture file with picture tag in the post, and download and save it on the local disk. After running the image crawler program, crawl the downloaded post bar and store the image as shown in Figure 2.

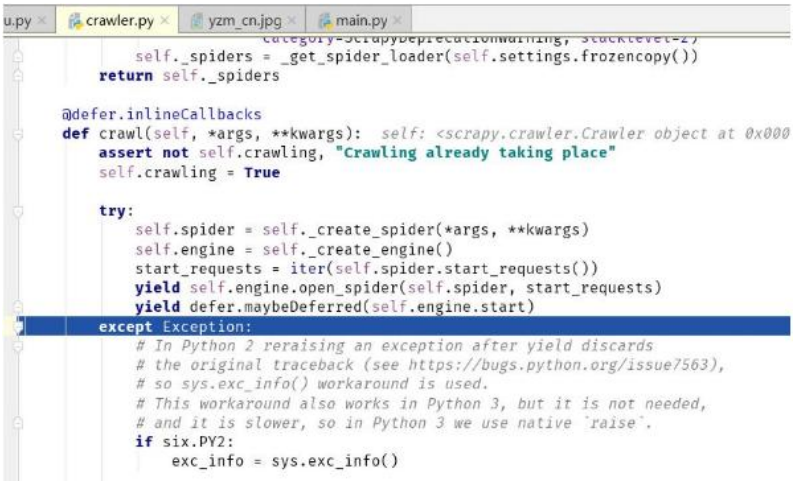

Fig 2: Baidu Post Bar pictures

IV. System module

The whole system consists of six modules: crawler control module, web download module, web parsing module, URL scheduling module, data cleaning module and data display module. These modules cooperate with each other to complete the function of network data capture.

(1) The main control module is mainly to complete some initialization work, generate seed URLs, put these URLs into the URL queue to be crawled, start the webpage downloader to download the webpage, then parse the webpage, extract the required data and URL address, enter the work cycle, control the workflow process of each module, and coordinate the work between each module.

(2) The main function of webpage download module is to download webpages. But there are several cases, for anonymous access to the web page, you can directly download, for the need for authentication, you need to simulate the user login and then download, for the need for digital signature or digital certificate to access the website, you need to obtain the corresponding certificate, load into the program, after verification, you can download the web page. The network is rich in data, for different data, different download methods are needed. After the data download is completed, the downloaded Web page data is transferred to the web page parsing module, and the URL address is put into the crawled URL queue.

(3) The main function of the web page parsing module is to extract the information that meets the requirements from the web page and transfer it to the data cleaning module, extract the URL address and transfer it to the URL scheduling module. In addition, it also extracts the data that meets the specific requirements through regular expression matching or direct search, and transfers the data to the data cleaning module.

(4) The URL scheduling module receives the URL addresses from the web page parsing module, and then compares these URL addresses with the URL addresses in the crawled URL queue. If the URL exists in the crawled URL queue, it discards these URL addresses. If it does not exist in the crawled URL queue, it puts the URL into the corresponding location of the URL address to be crawled according to the web page crawling strategy adopted by the system.

(5) The data cleaning module receives the data from the web page analysis module, and the data extracted by the web page analysis module is generally messy or non-standard data, which requires cleaning and sorting these data, sorting these data into data with a certain format, and then storing these data in the database.

(6) Data display module, according to user needs, statistics the data in the database, the statistical results are displayed in the form of text or graphics, the statistical results can also be stored in different formats of files (such as word document, PDF document, or excel document), permanently saved.

V. Conclusion

In this paper, the working principle of web crawler and the related technical modules of Python building crawler are studied, and the functional usage of modules urllib, BeautifulSoup and random of Python building crawler is discussed. Taking the construction of Baidu Post Bar picture crawler as an example, this paper expounds in detail the process of using Python's Requests-BeautifulSoup technology to build a picture crawler program to capture the pictures of Baidu Post Bar from the aspects of crawler preparation, simulating browser landing on the website, defining the picture crawling function, entering the post bar parsing webpage, crawling the picture storage and so on. Experimental results show that Python-based Requests-BeautifulSoup technology can quickly and effectively build a picture crawler program to automatically parse and crawl web page picture data.

Now has entered the era of big data, all walks of life have demand for data, for some ready-made data, you can get or buy it for free through the network, for some non ready-made data, you need to write a specific web crawler, search, analyze and convert it into the data you need. Web crawler meets this demand, and Python is easy to learn, has a ready-made crawler framework, powerful network support library, text processing library, can quickly meet the specific functions of the web crawler.

References

[1] Yang Yi, Bian Yuan, Zhang Tianqiao. Network Security Situation Awareness Based on Machine Learning. Computer Science and Application, 2020, 10 (12): 8
[2] Li Zhiyong. Hierarchical Network Security Threat Situation Quantitative Assessment Method. Communication World, 2016, 23: 70-70
[3] Hu Wenji, Xu Mingwei. Analysis of Secure Routing Protocols for Wireless Sensor Networks. Journal of Beijing University of Posts and Telecommunications, 2006, 29 (s1): 107-111
[4] Wei Yonglian, Yi Feng, Feng Dengguo, Yong W, Yifeng L. Network Security Situation Assessment Model Based on Information Fusion. Computer Research and Development, 2009, 46 (3): 353-362
[5] Xu Guoguang, Li Tao, Wang Yifeng. A Network Security Real-time Risk Detection Method Based on Artificial Immune. Computer Engineering, 2005,31 (12): 945-949
[6] Jiang Wei, Fang Binxing, Tian Zhihong. Network Security Evaluation and Optimal Active Defense Based on Attack Defense Game Model. Acta Computer Sinica, 2009, 32 (004): 817-827
[7] Miao Yongqing. Stochastic Model Method and Evaluation Technology of Network Security. China Science and Technology Investment, 2017, 4: 314
[8] Yi Hua Zhou, Wei Min Shi, Wei Ma. Research on Computer Network Security Teaching Mode for Postgraduates Under the Background of New Engineering. Innovation and Practice of Teaching Methods, 2020, 3 (14): 169
[9] Bao Xiuguo, Hu Mingzeng, Zhang Hongli. Two Quantitative Analysis Methods for Survivability of Network Security Management Systems. Acta Communication Sinica, 2004, 25 (9): 34-41
[10] Li Weiming, Lei Jie, Dong Jing. an Optimized Real-time Network Security Risk Quantification Method. Acta Computa Sinica, 2009 (04): 793-804