Network data analysis of crawler general search engine based on Python

Li Yapeng
(College of Urban and Environmental Sciences, Liaoning Normal University, Dalian 116029, China)

Abstract

With the development of information age and the popularization of programming technology, search engine has become a necessity in people's daily life. Most search engines use crawler technology as the core module to return the results of user queries through keywords. However, the explosive growth of network information makes it difficult to locate and locate information. In view of the above problems, this paper takes the general search engine "Baidu Search" as the crawling object on the basis of Python. On the basis of learning and analyzing the principle, core modules and running process of current crawling technology, this paper compares three different data crawling methods to achieve the goal of data crawling. Firstly, this paper gives the principle and workflow of crawler technology, introduces some key technologies in crawler engineering, and focuses on the method of crawling web pages. It is intended to provide reference for future research on the problems and the possibility of improvement.

Key word: General search engine, Python crawler, crawl web

1. General Search Enging work principle

Web crawler can be divided into two kinds: General crawler and focused crawler. General web crawler is to collect web pages and information from the Internet[1~3]. These web pages are used to provide support for the search engine indexing[4,5]. It determines whether the content of the whole engine system is rich and the information is timely. Therefore, the performance of the crawler directly affects the effect of the search engine. Focused crawler is a kind of web crawler program for specific subject requirements. It differs from general search engine crawler in that the focused crawler processes and filters the content when it implements web crawling, so as to ensure that only web pages related to the requirements are crawled.

1.1 Crawling

1.1.1 Basic workflow

The first part selects the seed URLs, which are captured in the URL queue[6]; removed to the crawl URL, DNS analyzes the host IP, URLs, and the corresponding page downloads[7]; the download pages are stored in the database, and these pages have been placed in the URL queue crawl URL. Analyze the URL in the crawled URL queue, analyze other URLs, and put the URL into the queue to be crawled, leading to the next loop (Fig.1).

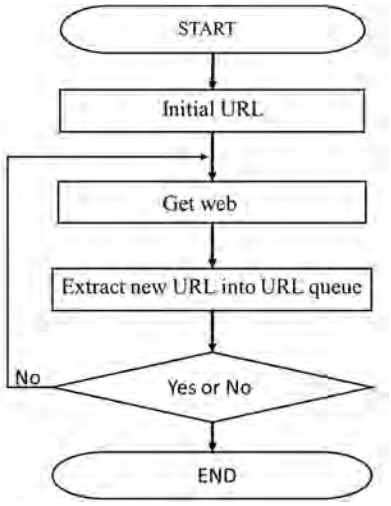

Fig. 1 Workflow diagram

1.1.2 Get URL

The new site offers sites to search engines on its own initiative: (http://yapengli.baidu.com/linksubmit/url), links to new sites on other sites (as far as possible within the scope of search engine crawling), search engines and DNS resolution services (such as DNSPod) cooperate, and new site domain names will be quickly crawled. But search engine spider crawling is typed with certain rules, it needs to comply with some commands or file content, such as the link labeled nofollower, or the Robots protocol. Robots protocol (also known as the crawler protocol, robot protocol, etc.), the full name is "Robots Exclusion Protocol" (Robots Exclusion Protocol), the site through the Robots protocol tells search engines which pages can be crawled, which pages can not be crawled, such as: Taobao: https://www.taobao.com/robots.txt; Tencent: http://ww. W.qq.com/robots.txt.

1.2 Data storage

Search engines crawl through the crawler to the web page and store the data in the original page database. The page data and the user browser which is the same as that of the obtained HTML. Search engine spiders also do a certain amount of duplicate content detection when crawling pages. Once encountering a lot of plagiarized, collected or duplicated content on sites with very low access weight, it is likely that they will not crawl any more.

1.3 Preprocessing

The search engine takes the crawler's pages back and preprocessed various steps. Extract text, Chinese segmentation, noise elimination (such as copyright text, navigation, advertising...), index of processing, link calculation, special document processing etc.. In addition to HTML files, search engines can also capture and index text-based file types such as PDF, WORD, WPS, PPT, TXT, etc. We often see this type of file in search results. But search engines can't handle non-verbal content such as pictures, videos, Flash, or execute scripts and programs.

2. Page analysis

Extraction from the web crawler to grab some data to achieve some purpose. A method of extracting web data: Beautiful, Soup and lxml regular expressions. With the option through the browser, view the page source code, through the Firebug Lite extension (http://getfirebug.com/firebuglite), analysis of web information. Firefox can install the full version of the Firebug browser.

2.1 Three scraping method

2.1.1 Regular expressions

Python regular expression (2.x): https://docs.python.org/2/howto/regex.html can grab data by matching a single page element, but regular expressions often fail if the page changes. A more robust way is to add the parent element of the unique identifier of the target web page to the matching rule[8].

```
import urllib2
import re
def scrape(html):
    area = re.findall('<tr id="places_area__row">.*?<td\s*class=["\']w2p_fw["\']>(.*?)</td>', html)[0]
    return area
if __name__ == '__main__':
    html = urllib2.urlopen('http://example.webscraping.com/view/United-Kingdom-239').read()
```

```
print scrape(html)
```

Generally speaking, the method of regular expression is not suitable for the scene of frequently changing web pages, and it has some problems such as difficult to construct and poor readability.

2.1.2 Beautiful Soup

Beautiful Soup is a Python library that can extract data from HTML or XML files: https://www.crummy.com/software/Beautiful Soup/. Compared with regular expressions, code using Beautiful Soup is easier to construct and understand. Installation module: PIP install beautifullysoup4-i https://mirrors.ustc.edu.cn/pypi/web/simple/Using Beautiful Soup, you first parse the downloaded HTML content into a source document to determine the actual format; then use find () and find_all () to locate the required elements[9].

```
# -*- coding: utf-8 -*-
import urllib2
from bs4 import BeautifulSoup
def scrape(html):
    soup = BeautifulSoup(html, "html.parser")
    tr = soup.find(attrs={'id': 'places_area__row'}) # locate the area row
    # 'class' is a special python attribute so instead 'class_' is used
    td = tr.find(attrs={'class': 'w2p_fw'}) # locate the area tag
    area = td.text # extract the area contents from this tag
    return area
if __name__ == '__main__':
    html = urllib2.urlopen('http://example.webscraping.com/view/United-Kingdom-239').read()
    print scrape(html)
```

2.1.3 Lxml

Lxml is based on the Python encapsulation of libxml2 XML analytic library. Http://lxml.de/, http://lxml.de/installation.html. The CSS selector represents the mode used by the selection element. Compared with the XPath selector, the CSS selector is more succinct. But in the internal implementation of Lxml, the CSS selector is actually converted to an equivalent XPath selector[10].

```
# -*- coding: utf-8 -*-
import urllib2
import lxml.html
def scrape(html):
    tree = lxml.html.fromstring(html)
    td = tree.cssselect('tr#places_area__row > td.w2p_fw')[0]
    area = td.text_content()
    return area
if __name__ == '__main__':
    html = urllib2.urlopen('http://example.webscraping.com/view/United-Kingdom-239').read()
    print scrape(html)
```

2.2 Comparison and analysis

Lxml methods are fast and robust and are usually the best choice for capturing data, while regular expressions and Beautiful Soup are useful only in certain scenarios(Table.1).

3. Conclusions

The amount of information on the web was "explosive" growth today, users need from the vast amounts of information can accurately extract the required information technology. In this context,

Table 1 Comparison and analysis of three Python Crawlers

Grab method	Property	Facility value	Installation
Regular expressions	fast	difficulty	Simple (built-in module)
Beautiful Soup	slow	simple	Simple (pure Python)
Lxml	fast	simple	relative difficulty

Web crawler technology will continue to attract people's attention because of its powerful ability to automatically extract web information.

Reference

[1] Guo Erqiang, Li Bo. Web crawler technology based on Python in large data environment [J].Computer products and circulation, 2017 (12): 82.

[2 Bird S, Klein E, Loper E. Natural Language Processing with Python[M]. Southeast University press, 2010.

[3 Manning C D, Surdeanu M, Bauer J, et al. The Stanford CoreNLP Natural Language Processing Toolkit[C]// Meeting of the Association for Computational Linguistics: System Demonstrations. 2014.

[4] Zhou Zhonghua, Zhang Huiran, Xie Jiang.Python-based data crawler on Sina Weibo [J].Computer Applications, 2014, 34 (11): 3131-3134.

[5] Lianjie, Zhou Xin, Cao Wei, et al. Data Mining Scheme of Sina Weibo [J].Journal of Tsinghua University: Natural Science Edition, 2011, 51 (10): 1300-1305.

[6] Chen Meng.Design and Implementation of a Python-based Sina News Reptile System[J].Modern Information Technology, 2018,2(07): 111-112.

[7] Pan Qiaozhi, Zhang Lei.A Brief Introduction to Python-based Network Crawler Technology in Large Data Environment [J].Network Security Technology and Applications, 2018 (05): 41-42.

[8] Bai Xueli. Analysis of the characteristics and application of Python based reptile technology [J]. Shanxi science and technology, 2018,33 (02): 53-55.

[9] Xiong Chang.Web page data capture and analysis based on Python crawler technology[J].Digital technology and applications, 2017 (09): 35-36.

[10] Fang Jin Tang. Design and implementation of online education platform based on web crawler [D]. Beijing Jiaotong University, 2016.

◆著者紹介

Li Yapeng

「Master of Quaternary Geology, school of Urban and Environmental Sciences, Liaoning Normal University, Dalian,China