Crawler Technology Based on Scrapy Framework

Wu Hejing
East University of Heilongjiang
Heilongjiang, China
e-mail: 499917928@qq.com

Abstract—With the development of the times and the popularization of scientific and technological products, the Internet has become inseparable from our lives, and search engines have become a daily necessity of people.In view of the growing needs。 **This topic requires the design of a prototype crawler system based on Scrapy framework. The specific requirements and contents are as follows: analyzing the structure and rules of the target website, looking for data items that need to be crawled. Based on Scrappy framework, a crawler prototype program is implemented by customizing crawling rules.Select the appropriate database for data access and analysis.**

Keywords-Creeper; Scrapy; framework; Python; Cookie

I. INTRODUCTION

With the development of the times and the popularization of scientific and technological products, the Internet has become inseparable from our lives, and search engines have become a daily necessity of people. Users can search information by inputting keywords into search engines to find information related to keywords. But with the explosive growth of network information, it becomes more difficult to find the desired information accurately.

In order to meet the growing needs, this paper chooses Scrapy, an open-source crawler framework based on Python, to crawl the "knowledge", and to learn and analyze the principle and running process of the crawler. On this basis, a prototype program of web crawler is implemented, and data crawling and storage are completed.

Firstly, this paper introduces the development process of the crawler, the working principle and classification of the crawler and the grasping strategy, and focuses on the current popular Cookie and its corresponding Session and Robots protocol.

Secondly, the use of Scrapy framework is introduced in detail. Using Scrapy framework to develop crawlers, the process and implementation details of developing crawlers by Scrapy are introduced in detail.

Finally, the crawler is tested and the results of crawling are shown.

II. WORKING PRINCIPLE

In a word, a crawler is a script or program that can get information and save it. The first step is to send a request to the target web page or website, and then get the response from the server.

Universal crawler is an important part of search engine. Its main function is to collect web pages on the Internet, then save them and process them.

Focus on crawlers, crawlers for specific needs. When it crawls a web page, it filters the first content and grabs the web page information related to the requirements.

DOI: 10.21307/ijanmc-2019-056

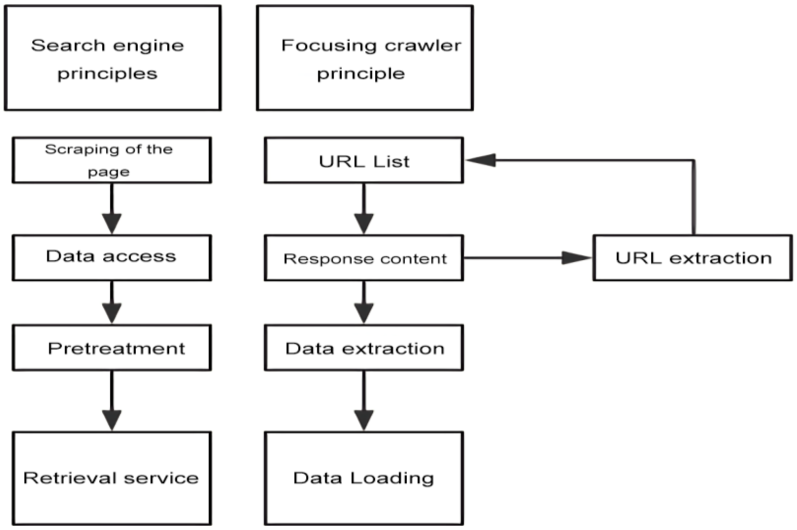


Figure 1. The Difference between Universal and Focused Reptiles

The crawler workflow is similar to the principle of ordinary users accessing web pages. When a user opens a web page, the browser will send a request to the server visiting the site, and the server will respond to the request and return it to the browser Response. The browser will parse the Response to display the web page[9]. The general crawler framework is shown in Fig. 2below.

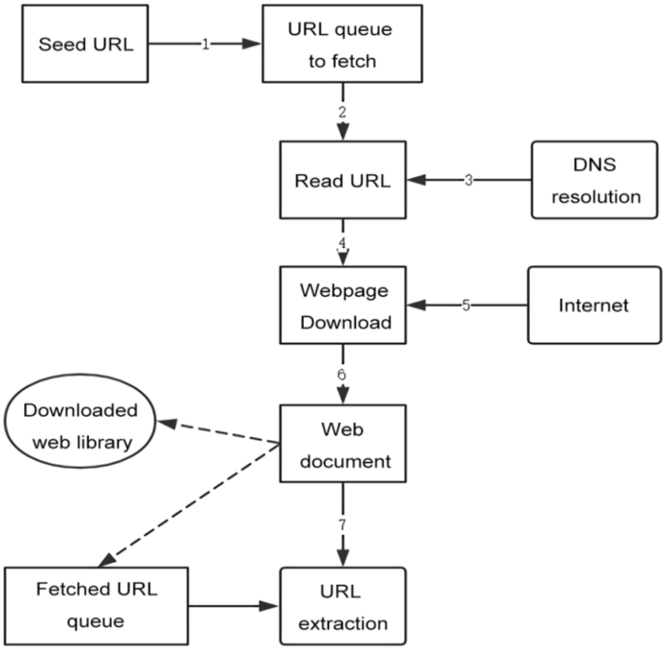


Figure 2. Universal crawler framework process

First of all, select some sites in the Internet, and take it as a starting point.Put these starting points into the queue to be grabbed, perform the queue out operation, and read out the queue elements.Resolve the URL of the target site through DNS.DNS will convert the domain name to the corresponding IP. The Downloader Downloads the target page through the server.The URL of the download page will be extracted.Reduplicate the crawl URL queue.The URL of the crawl URL queue continues to loop until the waiting URL queue is empty.

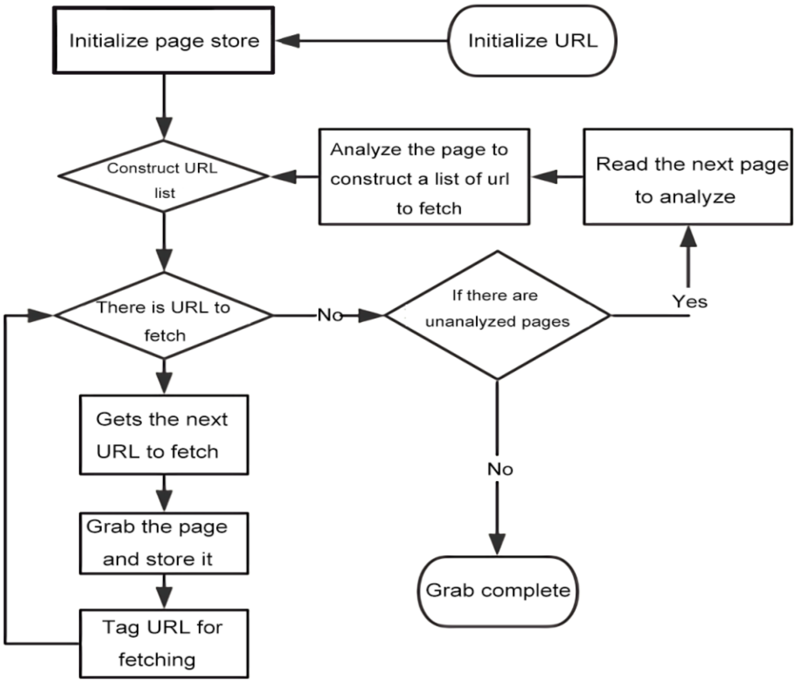
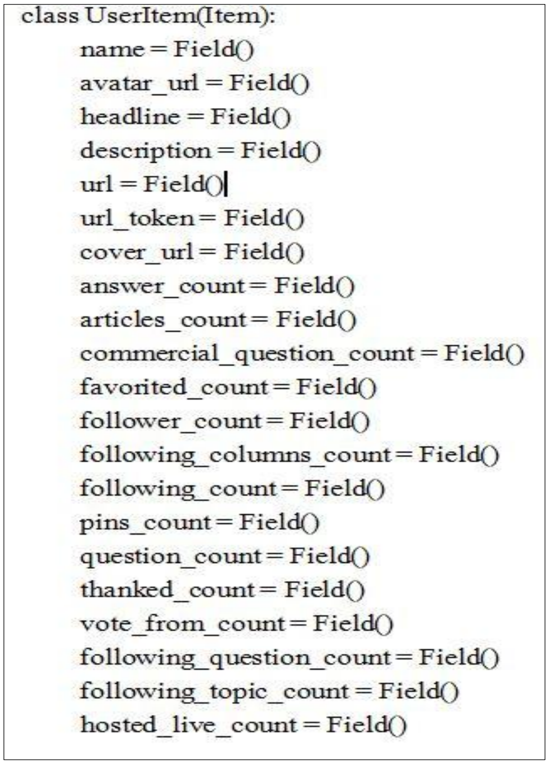

Initialize page store
Initialize URL
Construct URL list
Analyze the page to construct a list of url to fetch
Read the next page to analyze
There is URL to fetch
No
If there are unanalyzed pages
Yes
Gets the next URL to fetch
No
Grab the page and store it
Grab complete
Tag URL for fetching

Figure 3. Network crawler flow chart

III. DETAILS OF CRAWLER IMPLEMENT

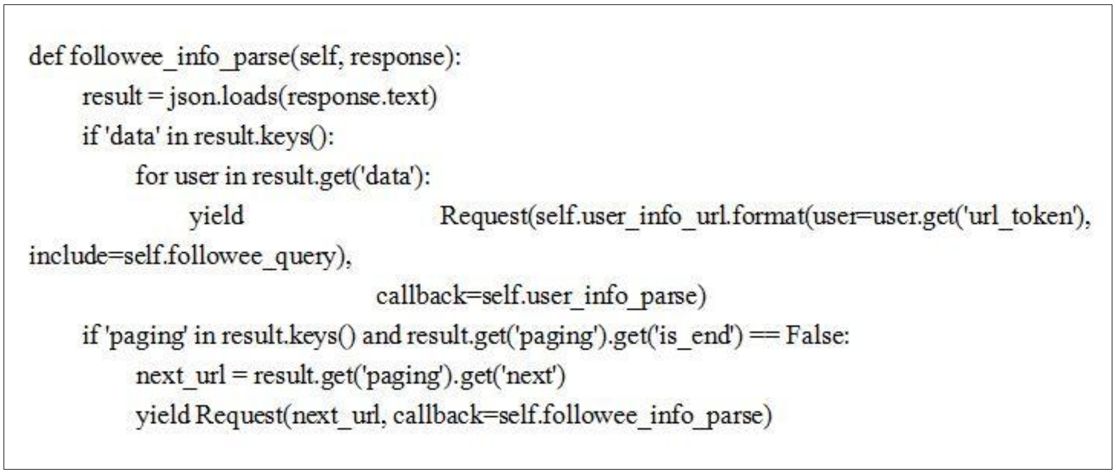
Data items are obtained by debugging web pages that know the user interface.The fieldin Python can accept almost any data type.

Follower_info_parse has two functions: first, it can initiate requests for user information through the attention list; second, it has the function of turning over pages. By parsing the response, it can obtain all users of the current target attention list and obtain detailed information of users. There is also the function of page scheduling to get the list of concerns on the next page. Further requests are then retrieved recursively for circular crawling. Followee_info_parse, which can request user's detailed information through fan list, also has the function of turning pages. Its implementation logic is exactly the same as follower_info_parse, except that the object of the request is different. One is to request detailed information from the person concerned, the other is to request detailed information from the person concerned with the current user.



```
class UserItem(Item):
        name = Field()
        avatar_url = Field()
        headline = Field()
        description = Field()
        url = Field()
        url_token = Field()
        cover_url = Field()
        answer_count = Field()
        articles_count = Field()
        commercial_question_count = Field()
        favorited_count = Field()
        follower_count = Field()
        following_columns_count = Field()
        following_count = Field()
        pins_count = Field()
        question_count = Field()
        thanked_count = Field()
        vote_from_count = Field()
        following_question_count = Field()
        following_topic_count = Field()
        hosted_live_count = Field()
```

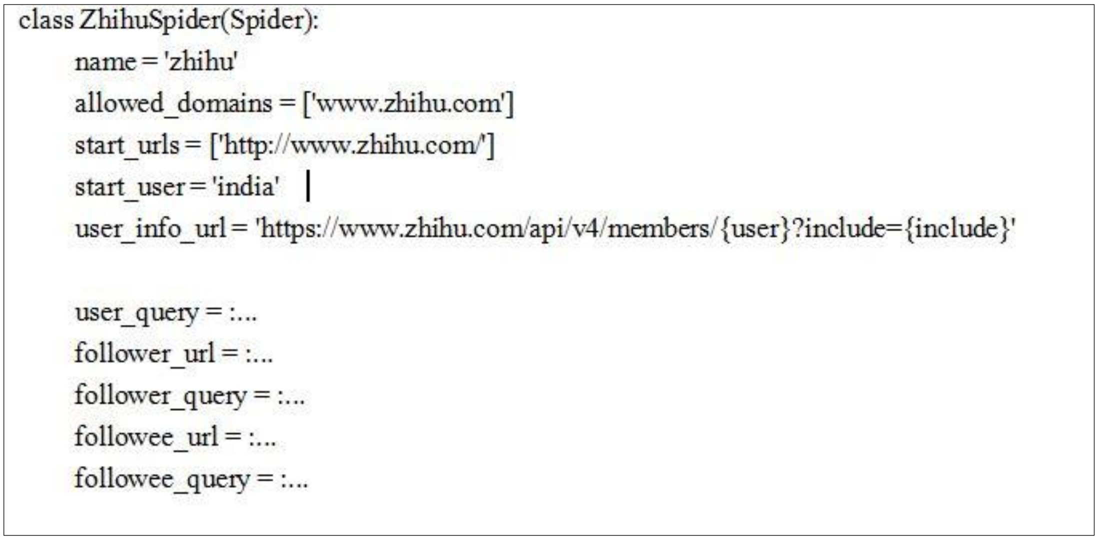
Figure 4. Field definitions in item.py file



```
def followee_info_parse(self, response):
        result = json.loads(response.text)
        if 'data' in result.keys():
                for user in result.get('data'):
                        yield                   Request(self.user_info_url.format(user=user.get('url_token'),
include=self.followee_query),
                                                callback=self.user_info_parse)
        if 'paging' in result.keys() and result.get('paging').get('is_end') == False:
                next_url = result.get('paging').get('next')
                yield Request(next_url, callback=self.followee_info_parse)
```

Figure 5. Followee_info_parse method

Spiders.py is the core of web crawling module and an important part of the whole project. It defines the core business logic. Followee_info_parse, which can request user's detailed information through fan list, also has the function of turning pages. Its implementation logic is exactly the same as follower_info_parse, except that the object of the request is different. One is to request detailed information from the person concerned, the other is to request detailed information from the person concerned with the current user.



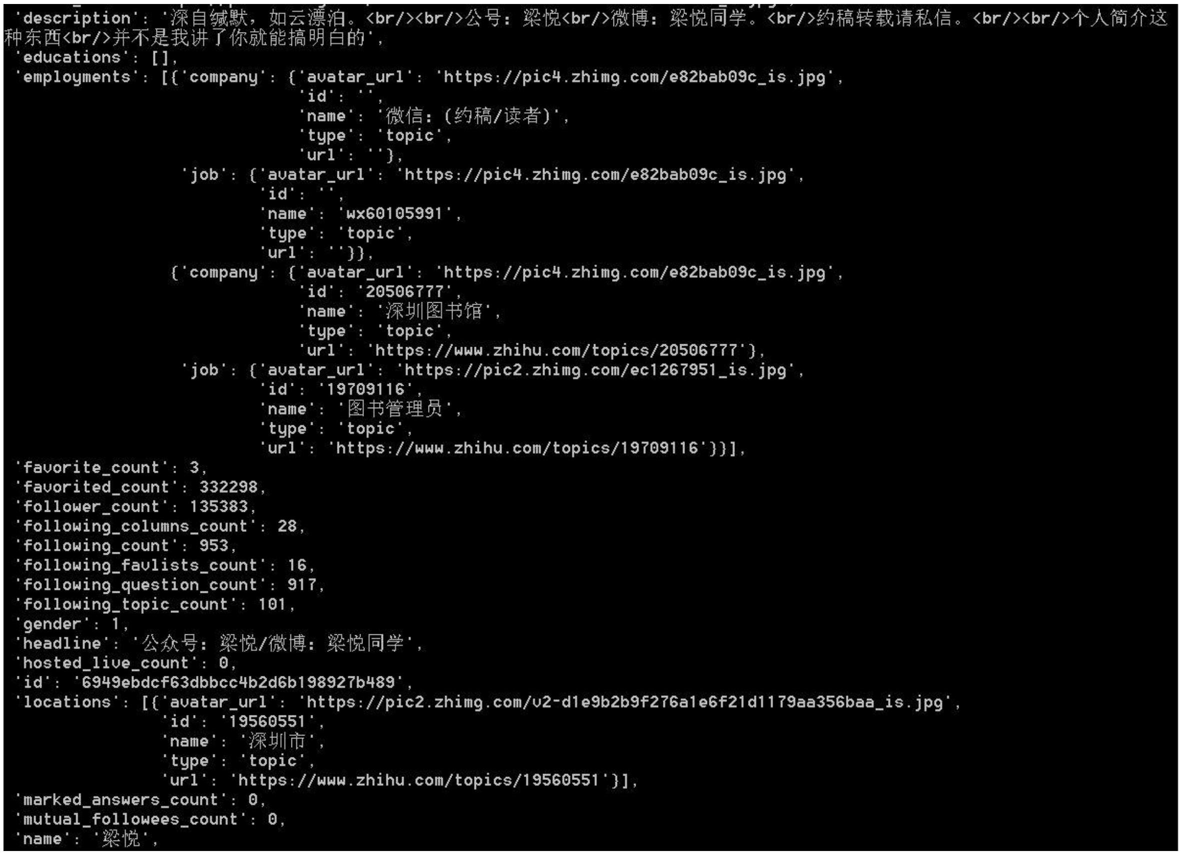
```
class ZhihuSpider(Spider):
    name = 'zhihu'
    allowed_domains = ['www.zhihu.com']
    start_urls = ['http://www.zhihu.com/']
    start_user = 'india'
    user_info_url = 'https://www.zhihu.com/api/v4/members/{user}?include={include}'

    user_query = :...
    follower_url = :...
    follower_query = :...
    followee_url = :...
    followee_query = :...
```

Figure 6. ZhihuSpider

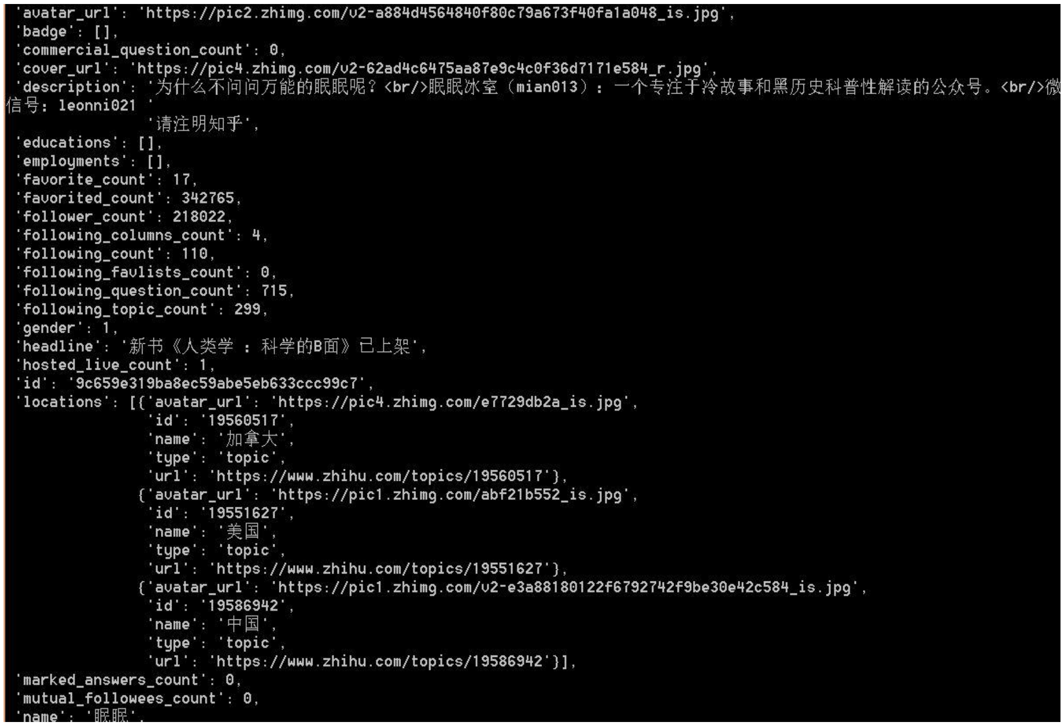
IV. RUNNING STATUS AND TESTING

After testing, the crawler capture data of a single host can reach 400,000 users per day. The crawling speed can be artificially controlled by setting it in the code.



```
 'description': '深自缄默，如云漂泊。<br/><br/>公号：梁悦<br/>微博：梁悦同学。<br/>约稿转载请私信。<br/><br/>个人简介这
种东西<br/>并不是我讲了你就能搞明白的',
 'educations': [],
 'employments': [{'company': {'avatar_url': 'https://pic4.zhimg.com/e82bab09c_is.jpg',
                              'id': '',
                              'name': '微信：(约稿/读者)',
                              'type': 'topic',
                              'url': ''},
                  'job': {'avatar_url': 'https://pic4.zhimg.com/e82bab09c_is.jpg',
                          'id': '',
                          'name': 'wx60105991',
                          'type': 'topic',
                          'url': ''}},
                 {'company': {'avatar_url': 'https://pic4.zhimg.com/e82bab09c_is.jpg',
                              'id': '20506777',
                              'name': '深圳图书馆',
                              'type': 'topic',
                              'url': 'https://www.zhihu.com/topics/20506777'},
                  'job': {'avatar_url': 'https://pic2.zhimg.com/ec1267951_is.jpg',
                          'id': '19709116',
                          'name': '图书管理员',
                          'type': 'topic',
                          'url': 'https://www.zhihu.com/topics/19709116'}}],
 'favorite_count': 3,
 'favorited_count': 332298,
 'follower_count': 135383,
 'following_columns_count': 28,
 'following_count': 953,
 'following_favlists_count': 16,
 'following_question_count': 917,
 'following_topic_count': 101,
 'gender': 1,
 'headline': '公众号：梁悦/微博：梁悦同学',
 'hosted_live_count': 0,
 'id': '6949ebdcf63dbbcc4b2d6b198927b489',
 'locations': [{'avatar_url': 'https://pic2.zhimg.com/v2-d1e9b2b9f276a1e6f21d1179aa356baa_is.jpg',
                'id': '19560551',
                'name': '深圳市',
                'type': 'topic',
                'url': 'https://www.zhihu.com/topics/19560551'}],
 'marked_answers_count': 0,
 'mutual_followees_count': 0,
 'name': '梁悦',
```

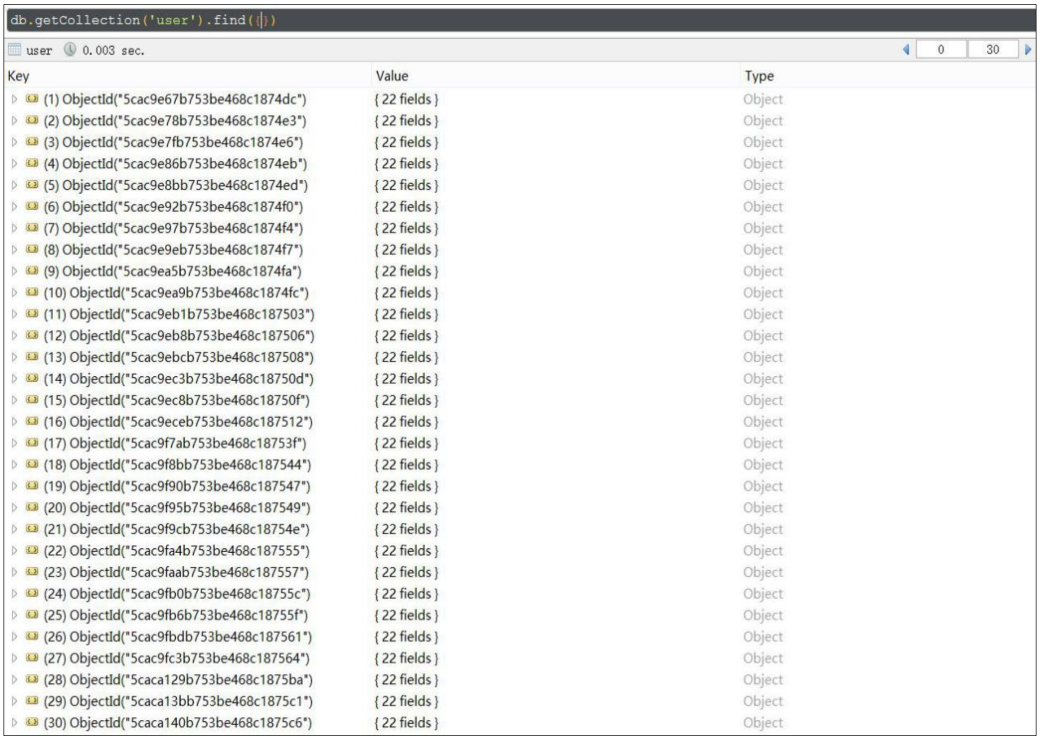
Figure 7. Screenshots of crawling 1



```
 'avatar_url': 'https://pic2.zhimg.com/v2-a884d4564840f80c79a673f40fa1a048_is.jpg',
 'badge': [],
 'commercial_question_count': 0,
 'cover_url': 'https://pic4.zhimg.com/v2-62ad4c6475aa87e9c4c0f36d7171e584_r.jpg',
 'description': '为什么不问问万能的眠眠呢？<br/>眠眠冰室（mian013）：一个专注于冷故事和黑历史科普性解读的公众号。<br/>微
信号：leonni021 '
               '请注明知乎',
 'educations': [],
 'employments': [],
 'favorite_count': 17,
 'favorited_count': 342765,
 'follower_count': 218022,
 'following_columns_count': 4,
 'following_count': 110,
 'following_favlists_count': 0,
 'following_question_count': 715,
 'following_topic_count': 299,
 'gender': 1,
 'headline': '新书《人类学 ：科学的B面》已上架',
 'hosted_live_count': 1,
 'id': '9c659e319ba8ec59abe5eb633ccc99c7',
 'locations': [{'avatar_url': 'https://pic4.zhimg.com/e7729db2a_is.jpg',
                'id': '19560517',
                'name': '加拿大',
                'type': 'topic',
                'url': 'https://www.zhihu.com/topics/19560517'},
               {'avatar_url': 'https://pic1.zhimg.com/abf21b552_is.jpg',
                'id': '19551627',
                'name': '美国',
                'type': 'topic',
                'url': 'https://www.zhihu.com/topics/19551627'},
               {'avatar_url': 'https://pic1.zhimg.com/v2-e3a88180122f6792742f9be30e42c584_is.jpg',
                'id': '19586942',
                'name': '中国',
                'type': 'topic',
                'url': 'https://www.zhihu.com/topics/19586942'}],
 'marked_answers_count': 0,
 'mutual_followees_count': 0,
 'name': '眠眠',
```

Figure 8. Screenshots of crawling 2

After the program runs, it gets a database named "zhihu" and stores all the information in the user table.



```
db.getCollection('user').find({})
```

user 0.003 sec. 0 30

Key	Value	Type
(1) ObjectId("5cac9e67b753be468c1874dc")	{ 22 fields }	Object
(2) ObjectId("5cac9e78b753be468c1874e3")	{ 22 fields }	Object
(3) ObjectId("5cac9e7fb753be468c1874e6")	{ 22 fields }	Object
(4) ObjectId("5cac9e86b753be468c1874eb")	{ 22 fields }	Object
(5) ObjectId("5cac9e8bb753be468c1874ed")	{ 22 fields }	Object
(6) ObjectId("5cac9e92b753be468c1874f0")	{ 22 fields }	Object
(7) ObjectId("5cac9e97b753be468c1874f4")	{ 22 fields }	Object
(8) ObjectId("5cac9e9eb753be468c1874f7")	{ 22 fields }	Object
(9) ObjectId("5cac9ea5b753be468c1874fa")	{ 22 fields }	Object
(10) ObjectId("5cac9ea9b753be468c1874fc")	{ 22 fields }	Object
(11) ObjectId("5cac9eb1b753be468c187503")	{ 22 fields }	Object
(12) ObjectId("5cac9eb8b753be468c187506")	{ 22 fields }	Object
(13) ObjectId("5cac9ebcb753be468c187508")	{ 22 fields }	Object
(14) ObjectId("5cac9ec3b753be468c18750d")	{ 22 fields }	Object
(15) ObjectId("5cac9ec8b753be468c18750f")	{ 22 fields }	Object
(16) ObjectId("5cac9eceb753be468c187512")	{ 22 fields }	Object
(17) ObjectId("5cac9f7ab753be468c18753f")	{ 22 fields }	Object
(18) ObjectId("5cac9f8bb753be468c187544")	{ 22 fields }	Object
(19) ObjectId("5cac9f90b753be468c187547")	{ 22 fields }	Object
(20) ObjectId("5cac9f95b753be468c187549")	{ 22 fields }	Object
(21) ObjectId("5cac9f9cb753be468c18754e")	{ 22 fields }	Object
(22) ObjectId("5cac9fa4b753be468c187555")	{ 22 fields }	Object
(23) ObjectId("5cac9faab753be468c187557")	{ 22 fields }	Object
(24) ObjectId("5cac9fb0b753be468c18755c")	{ 22 fields }	Object
(25) ObjectId("5cac9fb6b753be468c18755f")	{ 22 fields }	Object
(26) ObjectId("5cac9fbdb753be468c187561")	{ 22 fields }	Object
(27) ObjectId("5cac9fc3b753be468c187564")	{ 22 fields }	Object
(28) ObjectId("5caca129b753be468c1875ba")	{ 22 fields }	Object
(29) ObjectId("5caca13bb753be468c1875c1")	{ 22 fields }	Object
(30) ObjectId("5caca140b753be468c1875c6")	{ 22 fields }	Object

Figure 9. Database screenshots

V. CONCLUSION

When the crawler development is completed, it should be tested. Testing is a very important step. First of all, we need to know the performance of the crawler through testing, check whether the crawler has problems, and whether it can crawl the required data. Secondly, we should explore the anti-crawler strategy of the target website and improve the crawler. Finally, check the data that has been crawled to see if it achieves the expected goal of the project.The crawler system can also be extended, there are many technologies not added to it, and then added to it is the requirements of the enterprise level. In the process of writing this system, I consulted a lot of information about Scrapy. Scrapy framework is a new thing for me. New APIs and libraries. Fortunately, I have done some crawler projects before, which is not particularly difficult for me.

ACKNOWLEDGMENT

This paper is about the scientific research project of Heilongjiang Oriental University in 2019, "Implementation of Reptiles Based on Python Scrapy Framework", project number HDFKY190109.

REFERENCE

[1] Jing Wang,YuchunGuo. Scrapy-Based Crawling and User-Behavior Characteristics Analysison Taobao[P]. Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2012 International Conference on,2012.

[2] James W. McGuffee. Non-profit geographically constrained locator[J]. ACM SIGCASComputers and Society,2015,45(2).

[3] Yuhao Fan. Design and Implementation of Distributed Crawler System Based on Scrapy[J].IOP Conference Series: Earth and Environmental Science,2018,108(4).

[4] Shen Jie, Li Yifan. Application of Web crawler system in cloud media [J]. China Cable Television, 2018 (05): 595-597.

[5] Zhang Jin. Research on Web crawler technology based on Hadoop platform [D].Nanjing University of Posts and Telecommunications, 2017.

[6] Zhao Fen, Lei Zhenzhen, Yang Xiaoyun, Su Pengju and Wang Shunye.Based on Baidu Tieba College Students'Network Public Opinion Analysis [J]. Computer Knowledge and Technology, 2018, 14 (28): 227-229.

[7] Ding Zhongxiang, Yang Yanhong, DuYanming. Design and implementation of video information crawling based on Scrappy framework [J]. Journal of Beijing Printing Institute, 2018, 26 (09): 92-97.

[8] Xie Zhu. Emotional Tendency Analysis for Chinese Short Texts [D].Hunan University, 2018.

[9] Wei Chengcheng. Data Information Crawler Technology Based on Python [J]. Electronic World, 2018 (11): 208-209.

[10] Ye Xiqiezhong. Research and Implementation of Tibetan Text Automatic Classification Based on Web [D]. Qinghai University for Nationalities, 2014.

[11] ZhongJiajun. Research on Copyright Infringement Recognition of News Aggregation Platform [D].Lanzhou University, 2018.