

PAPER • OPEN ACCESS

## Multilingual Focused Crawler System based on Web Content Extraction and Path Configuration

To cite this article: Jie Wang *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **569** 052030

View the [article online](#) for updates and enhancements.

# Multilingual Focused Crawler System based on Web Content Extraction and Path Configuration

Jie Wang<sup>1,2</sup>, Sanhong Deng<sup>1,2,\*</sup>, Lijuan Wang<sup>3</sup>

<sup>1</sup> School of Information Management, Nanjing University, Nanjing 210023, China

<sup>2</sup> Jiangsu Key Laboratory of Data Engineering and Knowledge Service (Nanjing University), Nanjing 210023, China

<sup>3</sup> Geological Survey of Jiangsu Province, Nanjing 210018, China

\* sanhong@nju.edu.cn

**Abstract.** The multilingual focused crawler system combines web content extraction with path configuration to make use of their advantages and achieve automatic collection of network information in multiple languages. Firstly, system selects foreign language keywords according to crawling webpage language and Chinese keywords, and uses initial link to obtain webpage information. Then, it uses path configuration information or web content extraction algorithm based on the distribution line block to get webpage content, and adopts rules or configuration information to acquire new links, published time and title. Next, keywords are used to filter irrelevant information. Finally, results are presented as a list. When users use focused crawler system, the webpage path information can be configured or not according to requirements, and the collected network resources can also be searched or filtered.

## 1. Introduction

With the rapid development of technology, network resources are exploding exponentially. How to get interesting information to users from massive resources is a problem to be solved. Because of this, web crawlers come into being [1]. According to initial URL set, crawlers extract information and new links by parsing of web page, and add links to URL list, so that they can continue to crawl other web pages. The general crawlers get all information from web pages, which not only result in content irrelevant to user needs, but also reduce crawler efficiency. Focused crawlers can collect relevant information based on keywords, and then improve accuracy. However, many focused crawlers fetch data only for specific website, and cannot extend existing modules to suit other scenarios, which makes them less suitable [2]. Meanwhile, crawlers generally extract web content based on regular expressions or XPath. When they are applied to multiple sites, many rules need to be set. This leads to less flexibility of crawlers.

In this paper, we combine web content extraction with path configuration to design a multilingual focused crawler system based on keywords, and apply it to collect the Belt and Road national mineral resources. In system, users need to provide keywords in five languages, including Chinese, English, Japanese, Russian and Arabic, and initial links, as well as website language. If path configuration of web content is provided, system obtains it according to this information. Otherwise web content extraction algorithm will be used. Finally, system displays information in the form of a list on the front end.

## 2. Related work

Crawlers play a very important role in the acquisition of network information resources. We firstly



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/3.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

review focused crawlers research, and then introduce web content extraction algorithm.

Chakrabarti [3] proposed focused crawlers in 1999. They used the classifier and selector to get relevant web page URLs. After that, the research on focused crawlers began to rise. Uzun [4] divided webpage into blocks, and extracted hyperlinks, texts, menus from each block. Finally, they implemented intelligent crawler. Safran [5] used URL text, anchor text, parent pages, and text around links to predict page relevance and designed crawler based on Naive Bayes. Some researchers used graph theory, decision tree and semantic-based methods to achieve focused crawlers. Du [6] proposed context-based concept map to store contextual knowledge, which was web history information of users click. Then, they calculated similarity between concepts in the graph and matched unvisited pages to get the most relevant links. Li [7] applied decision tree to anchor text of hyperlinks to prioritize unvisited URLs. Dong [8] used dictionary-based ontology learning method to realize adaptive semantic focused crawlers.

After getting network information, how to extract web content is a key step in data processing. There are mainly three methods, which are template-based, statistics-based and machine learning. Sluban [9] found that web documents from same source usually had the common template, so they used URL Tree to extract web content. According to characteristics of webpage content layout, Ji [10] used starting block and ending block to collect text information. Liang [11] first extracted webpage tag features with noise information, and then used SVM to acquire content.

On the whole, focused crawlers have rarely research on multilingual information acquisition and combining content extraction with path configuration. Therefore, we start from multilingual keywords, use path configuration and content extraction, and finally build focused crawler system.

### 3. Focused crawler system

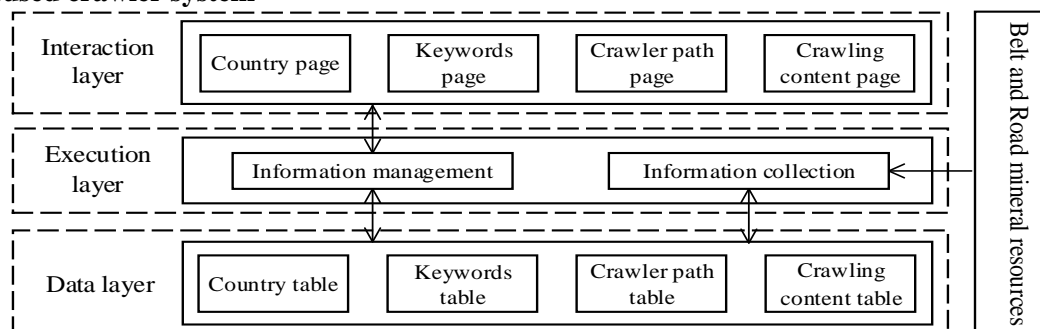


Figure 1. Multilingual focused crawler system framework.

Figure 1 is the framework of focused crawler system. In interaction layer, users can configure country, keywords, crawler path and crawling content. Then system transfers information to execution layer. Execution layer is business logic implementation part. It processes user requests and collects network resources. Data layer is database, which stores information and is called by execution layer. In system, each table in data layer has a one-to-one correspondence with page in interaction layer. The core parts of system are information management and information collection module, which are introduced next.

### 3.1. Information management module

**3.1.1. Country configuration module.** This module is to fill in information for countries that need to collect network resources, including country name and region. System divides 67 countries in the Belt and Road into 7 regions and stores them in country table.

**3.1.2. Keywords configuration module.** System sets 5 language keywords, which are Chinese, English, Japanese, Russian and Arabic. When configured, system demands Chinese keyword as primary key, and only one can be filled in. Other language keywords are the translation of Chinese keywords. Because of polysemy, other language keywords can be filled in multiples and separated by commas.

**3.1.3. Crawler path configuration module.** It is used to configure specific location of information to be crawled in the webpage, including the following parts.

- (1) **Initial Links:** Required field. System can continuously obtain new links according to initial links.
- (2) **Language:** Required field. It includes Chinese, English, Japanese, Russian and Arabic. Language is configured according to webpage language.
- (3) **Region:** Required field. It is from country configuration module.
- (4) **Country:** Required field. It comes from country configuration module. System displays corresponding countries according to the selected region.
- (5) **News Page Path:** Optional field. This path refers to the location of news list in web page and uses XPath to fill in.
- (6) **Next Page Path:** Optional field. It refers to next page location of news list and uses XPath to fill in. For news page and next page path, system demands that paths be either filled in or not filled in.
- (7) **Web Content Path:** Optional field. This path refers to the location of web page content, which uses XPath to fill in.
- (8) **Published Time Path:** Optional field. It refers to published time location in web page and uses XPath. System requires that web content and published time path be either filled in or not filled in.
- (9) **Keywords:** Required field. System shows Chinese keywords, and user needs to select at least one keyword for crawler configuration. After that, system obtains other language keywords according to the selected Chinese keywords and the language set.

**3.1.4. Crawling content configuration module.** It mainly displays obtained information to users in the form of a list. It can filter information according to release status and crawling time, as well as providing retrieval function. In addition, users can modify obtained title, published time and content.

### 3.2. Information collection module

Information collection module gets network resources based on crawler path configuration. It's designed by Scrapy and realizes concurrent crawling [12]. Its work includes four aspects: (1) URL deduplication. (2) Getting new links from web pages to achieve circular crawling. (3) Extracting web content, title and published time. (4) Filtering information by keywords. The process of this module is shown in Figure 2.

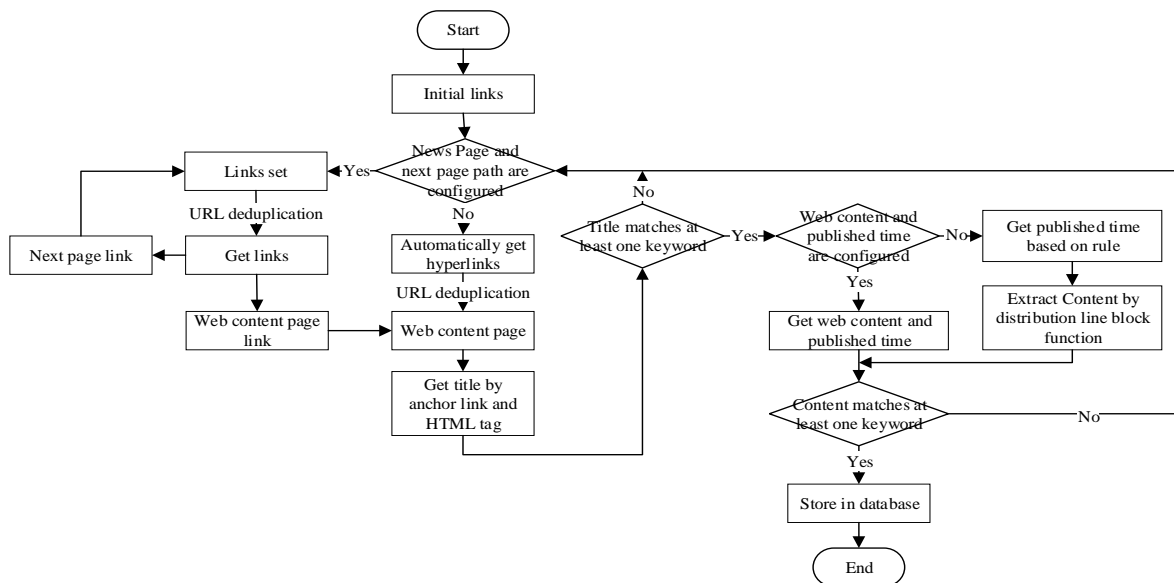


Figure 2. Information collection module.

**3.2.1. URL deduplication.** System uses memory-based approach to de-duplicate URLs. Crawler firstly gets existing URLs from database. Then it compares crawled webpage URLs with them to achieve URL filtering. In addition, system uses `allowed_domains` in Scrapy to set a list of domain names, so that URL whose domain name is not in the list can be removed.

**3.2.2. New links extraction.** In system, we set two methods to get new links.

(1) When both news page and next page path are configured, system can quickly get new links based on XPath and add them to URL list.

(2) When news page and next page path are not set, system applies Scrapy's Selector to automatically obtain new links from hyperlink and src attribute of iframe. Since some links may be files, videos or pictures, system filters them by means of rules. Meanwhile, links may not be complete in some websites, for example, `/a?i=3, ././a/b`. So they need to be standardized to get complete links. System uses Python's `urljoin` to combine current webpage URL with incomplete link to get full link starting with http.

**3.2.3. Web information extraction.** There are two ways to obtain web content and published time.

(1) If users set web content and published time path, system obtains information according to XPath, and then removes HTML tags by regular expressions to get text.



(2) If web content and published time path are not configured, system automatically extracts information. For web content, system adopts text extraction method based on the distribution line block function [13]. The algorithm firstly removes HTML tags. Then it selects  $K$  rows as one block and calculates total number of characters in each block. Next, it makes distribution function of block length and block number. Finally, the block content between sudden rise point and sudden drop point in distribution function is taken as web content. System uses the rule-based approach to get published time. For the year, there may be "Heisei" in Japanese website and Islamic calendars may be used in Arabic website, so system transforms them to standard year form. For the month, system establishes a unified month dictionary for English, Russian and Arabic, then converts month to standard numeric form by matching. In the end, system unifies published time into Year-Month-Day form, and stores it in database.

System combines anchor links and HTML tags to get title. If new link contains anchor text, it is used as web page title. If not, system sequentially extracts the `<title>`, `<h1>` and `<h2>` in corresponding webpage of new link, and uses first obtained text as title.

**3.2.4. Web Information filtering.** System filters web information based on keywords provided by users. Firstly, system gets foreign language keywords corresponding to Chinese keywords based on provided language. Then, the obtained title is matched with keywords, and web information will be stored as long as one keyword appears in title. If title does not match any keywords, system will match according to web content. Similarly, if web content matches at least one keyword, then save it, otherwise filter it.

## 4. System application

In this paper, focused crawler system is applied to collect the Belt and Road mineral resources. System interfaces are designed by Django's Admin, which are shown in Figure 3-6. The retrieval function is provided in country, keywords and crawling content configuration page. System adopts fuzzy matching to retrieve information based on search terms. The country and crawling content configuration page also provide filtering capabilities. The related information can be acquired by clicking filtering conditions.

System employs Scrapy to deploy focused crawlers to server, then the http command can be used to start and stop crawlers. Therefore, users can control crawlers by asynchronous request mode in crawler path configuration page. System sets progress bar and crawling state for each crawler. When the crawler is running, progress bar will be in scrolling state and crawling state will be . When it is completed, progress bar will reach 100% and crawling state will change to . System applies Ajax to achieve these changes, so that users can view crawler state without refreshing the browser.

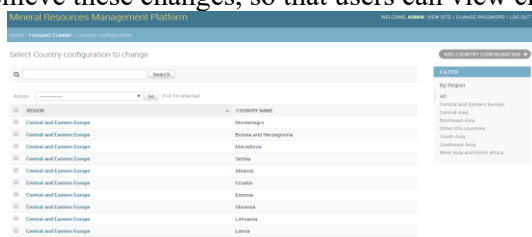


Figure 3. Country configuration page.

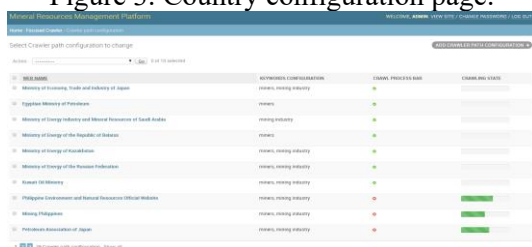


Figure 5. Crawler path configuration page.

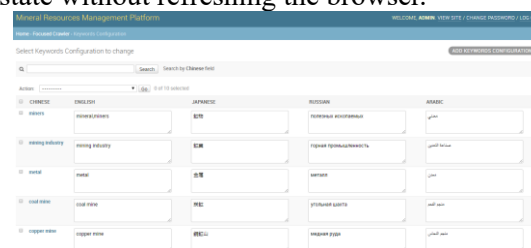


Figure 4. Keywords configuration page.

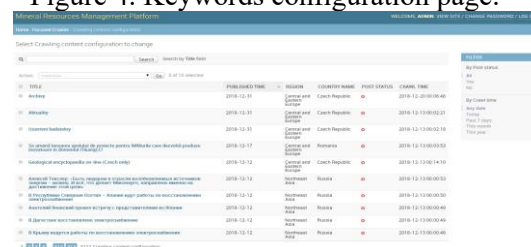


Figure 6. Crawling content configuration page.

## 5. Conclusion

Web path configuration can improve extracted content accuracy and crawling efficiency, but it requires manual settings and the cost is high. Web content extraction algorithm not only has higher accuracy, but also reduces manual participation. Therefore, we combine them in order to take their advantages. Finally, we realize multilingual focused crawler system based on web content extraction and path configuration by using five language keywords. System uses asynchronous method to start and stop crawlers. It can also obtain webpage title, content, published time in real time and display results in a list form.

However, system still has some shortcomings, including the following points: (1) It does not consider title and web content similarity when deduplicating. (2) It does not dynamically change IP, which making the stability of crawlers not high enough. (3) More rich visual analysis of text data, such as topic and time analysis. In the future work, we will improve above points.

## Acknowledgments

This work is supported by Key Project of the Experimental Teaching Reform of the “13th Five-Year Plan” of Nanjing University (SY201919).

## Reference

- [1] Kumar, M., Bhatia, R., & Rattan, D. (2017). A survey of Web crawlers for information retrieval. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(6), 1-45.
- [2] Lu, M., Wen, S., Xiao, Y., Tian, P., & Wang, F. (2017). The design and implementation of configurable news collection system based on web crawler. In *2017 3rd IEEE International Conference on Computer and Communications*. pp. 2812-2816.
- [3] Chakrabarti, S., Berg, M. V. D., & Dom, B. (1999). Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31(11-16), 1623-1640.
- [4] Uzun, E., Serdar Güner, E., Kılıçaslan, Y., Yerlikaya, T., & Agun, H. V. (2014). An effective and efficient Web content extractor for optimizing the crawling process. *Software: Practice and Experience*, 44(10), 1181-1199.
- [5] Safran, M. S., Althagafi, A., & Che, D. (2012). Improving relevance prediction for focused Web crawlers. In *2012 IEEE/ACIS 11th International Conference on Computer and Information Science*. pp. 161-166.
- [6] Du, Y. J., Pen, Q. Q., & Gao, Z. Q. (2013). A topic-specific crawling strategy based on semantics similarity. *Data & Knowledge Engineering*, 88, 75-93.
- [7] Li, J., Furuse, K., & Yamaguchi, K. (2005). Focused crawling by exploiting anchor text using decision tree. In *Special interest tracks and posters of the 14th international conference on World Wide Web*. pp. 1190-1191.
- [8] Dong, H., & Hussain, F. K. (2014). Self-adaptive semantic focused crawler for mining services information discovery. *IEEE Transactions On Industrial Informatics*, 10(2), 1616-1626.
- [9] Sluban, B., & Grčar, M. (2013). URL Tree: Efficient unsupervised content extraction from streams of web documents. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. pp. 2267-2272.
- [10] Ji, X., & Zhong, C. (2015). Blocking-Based Information Extraction Algorithm for Webpage of News. *Computer Applications and Software*, 32(04), 317-322.
- [11] Liang, D., Yang, Y., & Wei, Z. (2018). Information Extraction of Web Pages Based on Support Vector Machine. *Computer and Modernization*, 9, 21-26+31.
- [12] Han, B., Ma M., & Wang, D. (2019). Research on Crawler and Anti-reptile Based on Scrapy Framework. *Computer Technology and Development*, 29(2): 1-5.
- [13] Chen, X. (2009). General web page text extraction algorithm based on block distribution function. <http://code.google.com/p/cx-extractor/>.