Research and Implementation of Web Crawler Based on Learning Resources

Xiang-wei TAN, Yi ZHANG* and Zheng-jun PAN
South China Institute of Software Engineering, GU, China
*Corresponding author

Keywords: Learning Resources, Web Crawler, Heritrix.

Abstract. Aiming at the current situation that the effective learning resources are difficult to be acquired in college software teaching, it is proposed that the open-source web crawler framework Heritrix be redeveloped to study the function of crawler's network resource identification and data screening and to propose a theme-based Data acquisition program, and on this basis to achieve a network-based reptiles designated theme system to meet the teaching of university software for high-quality learning resources needs.

Introduction

At present, the teaching activities of software major in colleges and universities are more and more inseparable from the Internet. It is reflected in the prerequisite courses for teachers that Internet-related course resources are required to be searched online and students need to have access to the Internet for relevant learning resources after class. However, the resources on the network are not enough at the moment. Many of the resources found do not meet the needs of learning or are less timely or repeatable. The teachers and students spend a great deal of time and energy wasting resources in data collection Activities, leading to the effective implementation of teaching activities bottlenecks. Therefore, the use of network reptiles to help teachers and students improve the efficiency of learning resources acquisition appears to have good value and significance.

Although traditional web crawlers can easily access resources on the network, but lack of custom words are generally blind, many resources are not what users need, and need to spend a lot of time and energy to do data analysis and extraction operations; in addition, now The network learning resources are more and more abundant, and inevitable problems of homogeneity will increase the difficulty of data analysis. In the background of big data, the data update is faster and faster, the volume of data is more and more large, and the invisible Also reduces the effectiveness of traditional web crawlers collecting valid data.

For the traditional web crawler problem mentioned above, this paper proposes a custom-made development scheme based on the open source Heritrix framework. Taking a university-based learning resource sharing platform as an example, this paper designs a web crawler system that can collect thematic learning resource data, In order to improve the efficiency of data collection of learning resources and inject new vitality into the teaching activities of universities under the era of big data.

Heritrix Framework Introduction

Heritrix framework based on the common crawler framework integrates Spring's injection-style development ideas, mainly by the crawler crawler controller CrawlController, pending uri list Frontier, used to crawl task management thread pool ToeThread, used to customize the catch Process tasks to take the contents of the processor chain ProcessChain several major components [1]. The frame structure shown in Figure 1:
The Heritrix framework first reads the information in the crawler-beans.cxml crawling task configuration file, defines the crawling scope, content, seed URI, and grabbing the basic server information, the crawler crawler controller will use non-relational BerckleyDB database to be processed to crawl task list Frontier, when the controller to grasp the establishment of the URI crawling task, remove a free thread from the thread pool ToeThread to carry the task, crawling tasks through the task processor chain in the various processes After the function is built, it starts to crawl the data. After the data is parsed, the newly obtained URI is put into the queue to be fetched, and the loop continues until the URI in the crawl queue is processed.

Key Technology Introduction

Custom Theme Crawler Based on Heritrix Framework

Using the Heritrix framework to customize thematic crawlers is mainly done by extending the corresponding processor of the processor chain, where the Extractor processing extractor Extractor is responsible for extracting the new URI from the webpage. However, Universal Extractor provided by Heritrix can only crawl all the information and the user can not control what Heritrix should catch and what should not be grabbed. In this case, you can consider designing a custom Extractor to implement the crawling theme custom made. Here's how to write a custom Extractor:

First of all, inherit the abstract class Extractor, and override the method extract (CrawlURI curi).

```java
public class LearningResourceExtractor extends Extractor {
    protected void extract(CrawlURI curi) {
        // ...
    }
}
```

Next, the page corresponding to the URI is obtained, the page content is converted into a character string, and the content of the inspection page is traversed. If the content encounters a line that contains a specific identifier, that is, the page link address of the topic is encountered, the URI is intercepted in front of the domain name to form a URL, the link added to the pending list. Because the text talks about crawling themed web crawlers for teaching video resources on campus, the following regular expression is used to describe the URL of the video asset:

```java
Matcher swfParamMatcher = Pattern.compile(".*\w{4}=([^&]+).*video_id=(([^&]+).*eurl=([^&]+).*iurl=([^&]+).*")
.matcher(curi.toString());
swfParamMatcher.find();
```
Grabbed off the relevant topic page parsing JSoup this code-level page parsing tool to complete, after page analysis, teaching resources on the campus website using the `<source>` tag to describe the period `src` attribute is the video link. Use the APIs provided by JSoup to easily get the properties in the specified element node. However, we should pay attention to is to use the jsoup API get method to get the page document is in the absence of cooike state, some sites want to get the video playback address, you need to register to log in to have permission to access, you can consider using httpclient to simulate the user Login status to solve this problem.

**Thematic Relevance Measure**

In order to ensure that the crawled pages and extracted URLs are related to the topics, pages crawled by the crawler need to be analyzed to filter out the pages with low or irrelevant relevance to the topic. The solution proposed in this paper is to calculate the relevance of web pages by improving Pagerank algorithm based on vector space.

The formula for calculating the Pagerank value of a page $A$ is as Formula (1) [2]:

$$PR(A) = \frac{(1 - \lambda)}{N} + \lambda \left( \frac{PR(T_1)}{C(T_1)} + \cdots + \frac{PR(T_n)}{C(T_n)} \right)$$  \hspace{1cm} (1)

Where $P$ represents the total number of pages, $PR(A)$ represents the Pagerank value of page $A$, $PR(T_i)$ represents the Pagerank value of the page $T_i$ linked to $A$, $C(T_i)$ represents the number of outbound links of page $T_i$, $\lambda$ is the damping coefficient, The general range is: $0 < \lambda < 1$ [3].

Next, use the document scoring mechanism in the Lucene framework to establish a similar spatial model, vectorize the web content and matching subject information into an $N$-dimensional space, each vector is one-dimensional, new evaluation The algorithm considers that the smaller the angle between the two vectors, the greater the correlation between the vectors. The cosine of the angle is the score of relevance, the smaller the angle is, the higher the cosine value, the higher the score, the higher the matching degree of information acquisition [4].

We segmented the page into a series of terms, each Term, have to give a Term weight, different Term according to their weight in the document to affect the relevance of the document score calculation. So we regard the term weight of term in all pages as a vector, as Formula (2):

$$\text{Document} = \{\text{term}_1, \text{term}_2, ..., \text{term}_N\}$$
$$\text{Document Vector} = \{\text{weight}_1, \text{weight}_2, ..., \text{weight}_N\}$$  \hspace{1cm} (2)

Similarly, we look at the query as a simple document, but also with a vector to represent, as Formula (3):

$$\text{Query} = \{\text{term}_1, \text{term}_2, ..., \text{term}_N\}$$
$$\text{Query Vector} = \{\text{weight}_1, \text{weight}_2, ..., \text{weight}_N\}$$  \hspace{1cm} (3)

We put all the search for the document vector and query vector into an $N$-dimensional space, each term is one-dimensional. As shown in Figure 2:

![N-dimensional space](image)

Figure 2. N-dimensional space.
By calculating the cosine of angle $\theta$ as the score of correlation, the smaller the angle, the larger the cosine, the higher the correlation. Cosine formula such as formula (4):

$$\text{score}(q, d) = \cos \theta = \frac{V_q \cdot V_d}{|V_q||V_d|}$$  \hspace{1cm} (4)

The following assumptions: The query vectors are $V_q = <w(t_1, q), w(t_2, q), ..., w(t_n, q)>$ The document vectors are $V_d = <w(t_1, d), w(t_2, d), ..., w(t_n, d)>$, The vector space dimension is $n$, which is the length of the union of the query and the document. When a Term does not appear in the query, $w(t, q)$ is zero. When a Term does not appear in the document, $w(t, d)$ is zero. $w$ is the weight, the formula is generally $tf * idf$, where $tf(t \text{ in } d)$ is the frequency at which Term $t$ appears in document $d$, and $idf(t)$ is the number of documents that Term $t$ has appeared.

Since the numerator of the formula is the dot product of two vectors, then $t_1, t_2, ..., t_n$ here only have the non-zero values of the query and the union of the document, only if the query appears or only appears in the document Term of the value of the zero; in the query, very few people will enter the same word in the query, so you can assume that $tf(t, q)$ are 1; $idf$ is the number of Term in the number of documents appeared, Which also includes the query phrase this small document, so $idf(t, q)$ and $idf(t, d)$ is the same, is the total number of documents in the index plus one, when the total number of documents in the index is large enough, Query phrase This small document can be ignored, so you can assume that $idf(t, q) = idf(t, d) = idf(t)$. When calculating the vector length of a document, the weight of each term is no longer taken into account But all for one. The so the cosine formula becomes the formula (5):

$$\text{score}(q, d) = \cos \theta = \frac{1}{\sqrt{\sum_{t \in q} tf(t)^2} \times \sum_{t \in q} idf(t)^2 \times \sqrt{\sum_{t \in d} tf(t)^2}}$$  \hspace{1cm} (5)

According to the formula (1) of the traditional Rank Rank algorithm, the Page Rank algorithm which can increase the content similarity of web page content can be modified into formula (6): [6]

$$\text{PR}(A) = \left(\frac{1 - \lambda}{n} + \lambda \left(\frac{\text{PR}(T_1)}{g(T_1)} + ... + \frac{\text{PR}(T_n)}{g(T_n)}\right)\right) \times f + \text{score}(q, d) \times (1 - f)$$  \hspace{1cm} (6)

Score $(q, d)$ to measure the similarity between the page and the subject; $f$ is the fusion weight coefficient, used to adjust the original Pagerank value and score $(q, d)$ in the operation of the proportion of the value of the range 0-1.

The improved Rank Rank algorithm has a certain ability of distinguishing results due to the combination of the similarity of web content, which can reduce the number of unrelated pages and improve the sorting effect of search results.

Filter the URL Has Been Crawled

In the Heritrix framework, FrontierScheduler is a sub class of the PostProcessor class that adds URL links that have been parsed in Extractor to Frontier for later processing. In order to further improve the recognition ability of the theme crawler, some URLs that have been crawled are filtered and the system intends to use a Bloom filter.

Assuming that there is a set of $n$ elements $S = \{s_1, s_2, ..., s_n\}$, one-to-one mapping processing is performed by $k$ mapping functions $\{f_1, f_2, ..., f_k\}$. $S_j$ $(1 \leqslant j \leqslant n)$ in $S$ is mapped to $k$ values $\{g_1, g_2, ..., g_k\}$, and then the corresponding array of array elements in array $[g_1], array[g_2] ...$. The value of array $[g_k]$ is set to 1. If you want to find out whether an item is in $S$, then you can use the mapping function $\{f_1, f_2, ..., f_k\}$ Get k values $\{g_1, g_2, ..., g_k\}$, and then determine whether the value of the array array is all 1, if all 1, the item in $S$, otherwise item is not included in $S$.

Design and Implementation of the Theme Web Crawler System

The theme crawler system design workflow is as follows: First, take the seeds of URL from the list to be crawled, download the web page from the Internet according to the HTTP protocol and the DNS protocol, and then parse the page to filter the two function modules. For persistent storage, and then
processed to meet the requirements of the URL sent to the list to be crawled for the next download. [5]

The overall system structure shown in Figure 3:

![Figure 3. Theme crawler system functional structure.](image)

Start the Heritrix service and keep it running. The crawler then starts crawling based on the configuration file. The web console can then build crawl tasks, check configuration items, run crawl tasks, and suspend / intercept crawls. Tasks and other operations. Interface shown in Figure 4:

![Figure 4. Heritrix's web console interface.](image)

The crawler download page containing teaching video resources extracted from the URL, into the display layer for display, and provide download and online play function. Retrieve the campus network theme of "Java" related teaching resources shown in Figure 5:

![Figure 5. System background management interface.](image)
System has been tested, the basic functions consistent with the expected, the relevant website can be designated subject resources for effective access.

**Concluding Remarks**
This paper mainly focuses on how to efficiently obtain thematic network resources. By extending the open source Heritrix framework, a solution based on thematic network crawler system is developed, which is good for the research and development of the same type of software system Inspiration

**Acknowledgement**
This research was financially supported by China Guangdong Province "innovation strong school project" research projects.

**Reference**