A Research on the Method of Fine Granularity Webpage Data Extraction of Open Access Journals

Zhao Huaming,[1] Zhao Xiaomin,[2] Zhangzhe[3]

1（National Science Library, Chinese Academy of Sciences, China 100190）

2（Business school, Jilin University ,China 130062）

3（Software college, Nanyang Normal University ,China 473061）

Abstract: In light of the problems existing during the fine granularity webpage data extraction of open access journals (OAJ), e.g. complicated webpage structure, relative insufficiency of page markup node hierarchy and vague fine granularity data structure, this paper puts forward a new solution to fine granularity web information extraction by combining the webpage markup language processing technology and natural language processing (NLP) technology, in reference to the mainstream methods of deep web information extraction at present. During the formulation of data extraction rules, the new method can reduce the human intervention while the quality of data extraction is assured, by such tools as customization of semi-automatic data interaction, intelligent matching, simulation of extraction results, visualized validation, etc. At last, the paper selects three OAJ websites randomly to conduct the data extraction test. The test results show that this method can achieve a quality extraction of the fine granularity webpage data, and also effectively integrate the fine granularity webpage data of OAJ and form services.

Key words: fine granularity, open access, data-mining, deep-web, extraction

1. Foreword

The open access (OA) journal is a type of network academic resources available for users free of charge, and also a type of important OA resources without any access restriction [1]. As the movement in OA academic resources develops rapidly across the world, the OA academic information resources are becoming one of important academic research resources. More and more researchers focus on how to effectively mine and utilize the OA academic resources. The OAJ academic information resources usually refer to the fine granularity DEEP WEB resources. Meanwhile, webpage data extraction constitutes a part of basic research in web information mining. Hence, this paper researches the fine granularity data extraction of OA academic journals, analyzes and summarizes the existing mainstream methods of deep web information extraction, proposes a fine granularity web information extraction solution which integrates the webpage markup language processing technology and NLP technology, in combination of the OAJ webpage features. Meanwhile,

this paper puts forward the key ideas, steps and processes of realization, and validates the data extraction effect of this method in cases.

2. Relevant Work

2.1 Deep web information extraction method

Deep web data extraction, also known as deep web information gathering, is one of research highlights in the field of data mining. In recent years, a few methods of information extraction based on wrapper (Consisting of a series of extraction rules and computer codes to apply these rules, the wrapper is a program which extracts the required information especially from the specific information source and return the results [2]). According to the different location techniques in use, it can be classified into two categories:

(1) The extraction method based on the webpage markup language processing technology: This method is to locate and extract information according to the webpage structure. That is, before the information extraction, this method parses a web document into a document object model (DOM) tree, and then compares the edition distance of all nodes to confirm the target data area. At last, the data records in the target data area are recognized. The whole process is actually realized by the operation of translating the webpage data extraction into DOM tree. The typical system with this technology is the wrapper system MDR [3] from the Department of Computer Science, University of Illinois, Chicago, USA. Such wrappers as SG-WRAP[4], EXALG[5], XPath-Wrapper [6] and WDE [7] are further improved on the basis of MDR. For instance, SG-WRAP adds the semantic description of data to support XML document; EXALG deduces precisely the extraction rules according to the discovery frequency and definition of different roles of the calculation-typed symbols; XPath-Wrapper borrows the W3C standard XPath language to further confirm the data target; WDE introduces the weighting factor in computing the value of tree editing distance of nodes, specific to the features of less-structured webpage type. RoadRunner[8] and DeLa[9] utilize the grading mode of DOM tree to discover the nesting structure of list webpage, and successfully extract the target data lines constituting the backend data tables.

(2) The extraction method based on the processing technology of visualization characteristic: This method applies the webpage visualization characteristics in the location of target data. The target data located through the application of visualization characteristics of webpage is to make up for some disadvantages in purely depending on the webpage marking language, e.g. weak semantic representation of webpage markup language, random syntactic structure, composition of target data possibly with webpage markup symbols, etc. Similarly, the typical system with this technology is DEPTA [10-11] from the University of Illinois. In addition, ViPER[12] and ViNTs[13] also apply the similar technology. At the same time, they both locate the target data by the webpage markup language processing technology and the processing

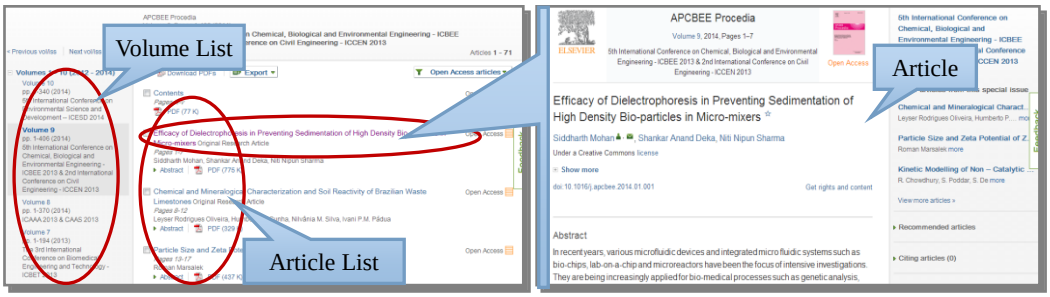
technology of visualization characteristics. On the webpage markup language processing technology, they profit from MDR approach, by which the visualization characteristics of webpage is used for segmentation of data area and distribution of authority. However, the difference is that ViPER takes into account the horizontal and longitudinal visualization characteristics, while ViNTs only focuses on the horizontal visualization characteristics. ViDRE[14] designs a target data location method which is purely dependent on the visualization characteristics, without any relevance to the webpage markup language. It applies VIPs (Vision-based Page Segmentation) algorithm [15] to generate the visualization characteristics tree of data blocks for webpage, and completes the location and extraction of target data in the visualization characteristic tree.

However, the above systems only can differentiate, acquire and process the webpage records, rather than further analyze and process fields/items - the smaller information units included in the content of the webpage records. As a result, the granularity of extracted information is so coarse that it cannot meet the higher requirement of professional applications for field analysis, e.g. open access resource of academic journals, which requires extracting the title, author, affiliation, year of publication, page number and other more detail information.

2.2 Analysis on Features of OAJ websites

The OAJ website pertains to deep web network resource, which is generally achieved by the technology of active webpage. With the uniform template, a vast majority of website contents are extracted dynamically from database according to different parameters and generated automatically, having an obvious hierarchical structure. The URL in the same OAJ website often presents a clear tree-shaped hierarchical structure as well.

Through the observation and analysis on a number of pages from different OAJ websites, this paper concludes that: (1) List page, including the volume list page, and the article list page, generally contains a few records, each of which is generated on the basis of the same webpage template, with a basically same data format; the volume list page generally includes the information such as year, volume No., journal name, etc.; the article list page generally include the information such as author, article name, journal name, key words, etc., and part of webpage also include the abstract of this record. (2) Detail page describes all detailed information of a specific literature record, of which partial information coincides with the information of such record in the list page. Figure 1 shows two categories of page of OAJ "AASRI Procedia": list page and detail page. The category of such page can be also recognized clearly through URL links. Where the URL of list page contains the words "/science/journal" and the URL of detail page contains the words "/science/article", these two categories of page pertain to the science literatures, presenting a very clear hierarchical structure.



（a）list page　　（b）detail page

Figure 1 Instance diagram of page of OAJ "AASRI Procedia"

Through observation and analysis on large amounts of page source codes from different OAJ websites, it is not difficult to find that: the granularity of DOM tree on the webpage is coarse, while the node of partial page markup tree is relatively weak so that it cannot reveal the backend data completely in the form of fields. Hence, in face of fine granularity field-level information extraction, we should consider applying the natural language data processing technology as the supplement, in a vision to precisely segment and extract the fine granularity field-level data. According to the above features of OAJ websites as we've observed, this paper makes the following four assumptions: There are a great number of webpage generated by the same template under the node of the same directory of URL tree; the webpage layout generated by templates is basically consistent; the data structure generated by templates is consistent roughly; URL text in the webpage is to describe and further expand the theme contents of target webpage.

Considering that the OAJ webpage is featured with strong regularity, similar data format, uniform webpage and data template, this paper puts forward a fine granularity web information extraction solution which integrates the webpage markup language processing technology and NLP technology, with the OAJ websites as research object and the extraction of web field-level data as the goal, and presents the specific realization, including: data preprocessing module, semi-automatic template tagging module based on browser, customization module for data extraction templates, automatic template relevance module (automatic prompt), visualized validation module, data extraction module, etc.

3. Field-based Fine Granularity Webpage Data Extraction Method

This paper mainly extracts the following attributes of OAJ literature information: author, article title, journal name, key words, year of publication, volume No., etc. According to the difference of page contents, a successful extraction requires extracting at least three fields or attributes, such as key words, author & affiliation and abstract for the detail page; journal name, year, volume No. and volume link for the volume list page; title, author and literature link for the article list page. The complete extractor system framework is as shown in Figure 2, including two processes: template training process for page data extraction rules and web information extraction process.

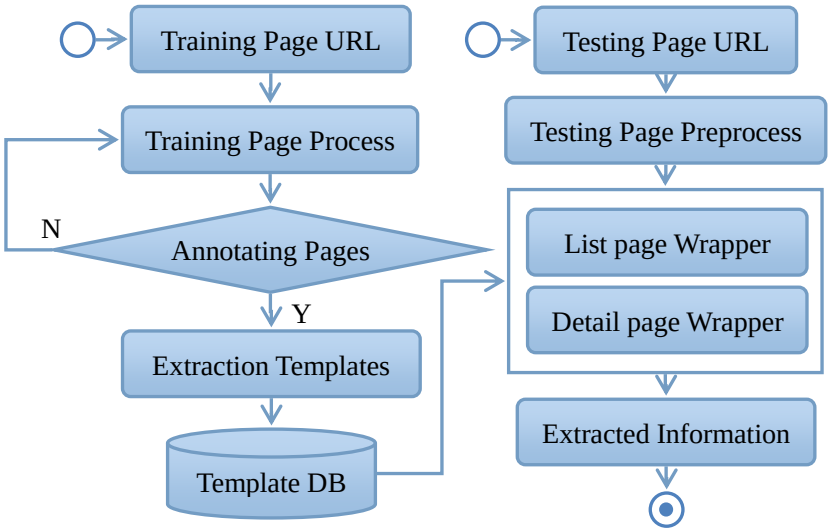


Figure 2 Overall framework of deep-Web fine granularity information extraction

Template training process of data extraction rules (customization process of wrapper) is described as follows:

(1) Data preprocessing, e.g. denoising and data standardization/normalization processing, etc.;

(2) Customization and operation of extraction rules based on interactive interface: intelligent matching, increase/modification of rules, extraction result simulation, etc.

(3) Extraction result verification: To confirm the field extraction rules or templates;

(4) Export <Path, string expression> of the field to the backend server (database).

Web information extraction process is described as follows:

(1) Follow the updates of target journal pages and launch the task scheduling of journal data updates;

(2) Acquire/gathering the update data of target journal pages and execute the standardized preprocessing;

(3) Confirm whether it is the list page of volume, the article list page or the detail page automatically according to the page features, select the corresponding extraction rules (wrapper), extract the relevant information, save the extracted information in the backend database and export the statistical information of extraction results.

3.1 Template training process of data extraction rules

In the training of extraction rule templates, the new basic seed page is added first. After the system acquires the URL of the new seed page, the system backend executes the automatic match between URL routing information of the new seed page and that of the

existing seed page in the template library, through the intelligent template correlation module, and feedbacks the page rule templates with high similarity and field extraction rules to the frontend interface, so as to accelerate the template customization and accuracy of data extraction rules for the new seed page; then, based on the interface of semi-automatic template extraction, the system generates the extraction rules for each field of the new seed page by the interactive method, to form the field-level data extraction rule templates for the new seed page; at last, the system will validate the accuracy of new data extraction rule template by the visualization template validation tool, if an extract result is correct, the new template will be saved in the backend journal page template library, while for those incorrect templates, the system will continue to modify their extraction rules until they are correct.

The generation process of template library of page data extraction rules include four subprocesses and one interactive tool: (1) Data preprocessing subprocess; (2) Intelligent template correlation subprocess; (3) Rule template customization subprocess; (4) Extraction result validation subprocess; and the semi-automatic customization tool for data extraction templates based on graphical interface.

3.1.1 Semi-automatic customization tool for data extraction templates

Generally, the page structure and data template are varying from OAJ and OAJ websites. It needs the manual assistance in the customization and management of data extraction template library, so the quality of data extraction can be assured. To reduce the error rate and workload, this system develops a semi-automatic customization tool for data extraction templates to simulate the data extraction effect, supplementarily confirm the extraction rules and generate the extract templates directly through webpage, including: simulation of data gathering results, simulation of data preprocessing results, simulation of extraction results, intelligent matching, and other effects and functions as shown in Figure 3.

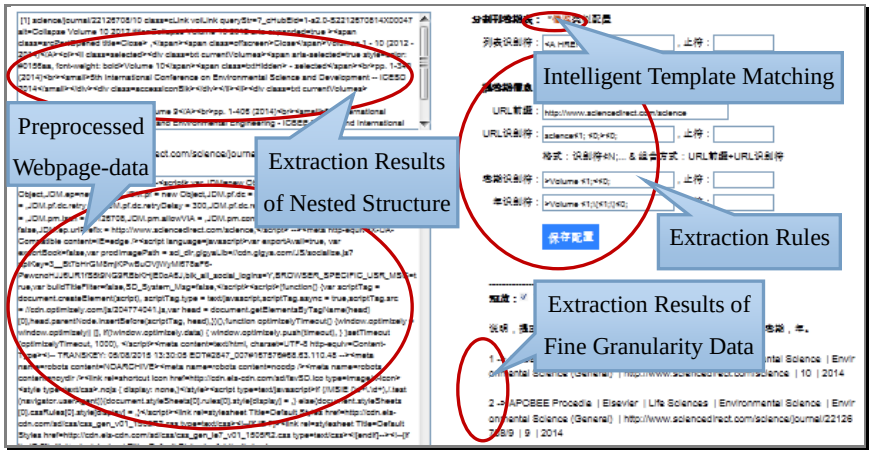


Figure 3 Schematic diagram of semi-automatic customization tool for data extraction templates based on the graphical interface

3.1.2 Training data preprocessing

The work of training data preprocessing mainly refers to gathering and denoising of training page data, including two aspects: (1) Standardized data processing, which is to check whether the page data includes any separator or identifier in the system default use based on ensuring the integrity of acquired data; if any, it needs to check and remove the unnecessary markup or placeholder in the HTML document through the replacement processing. The work is actually completed before the interactive interface of semi-automatic template tagging is presented, and the interactive interface only presents the processed results. (2) Optimal matching of gathering engine. Two data gathering engines are available in this system. One is the default data gathering engine based on HTTPCLIENT tool, with fast data gathering; the other is the data gathering engine based on HTMLUNIT tool, which supports the gathering of JAVASCRIPT and other dynamic loading data, with a relatively low speed of data gathering. The system operator determines an appropriate data gathering engine through checking whether the items of key field in the acquired data are complete. The system will save the relevant configuration automatically, and apply the appropriate gathering engine during the subsequent actual data gathering. The data preprocessing process is as shown in Figure 4.

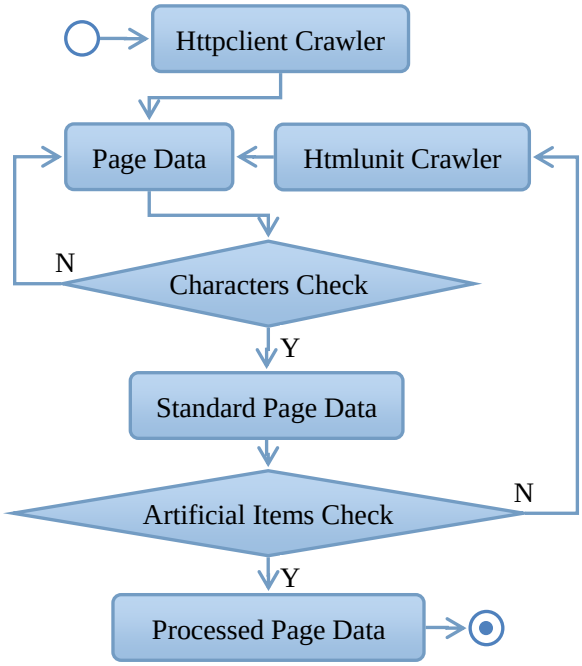


Figure 4 Training data processing flowchart

3.1.3 Information path location

It is very critical to annotate the path for the relevant data record or field, whether the list page or detail page. The path location doesn't need the human intervention under the

optimal conditions. Among the above-mentioned wrappers based on the webpage markup language processing technology and based on the visualization characteristic processing technology, a vast majority of them can realize automatic extraction, without human intervention. They can process not only the webpage with multi-layer, nested list structure but also the complicated data, such as DeLa and DEPTA; however, it fails to process the decomposition containing the information of multiple fields under DOM tag nodes, so the location accuracy needs to be further improved.

The information path location can be achieved by multiple location basis, such as: (1) Location according to the tag location information: this location method is not accurate as it is always changing along with the location information of tag; (2) Path acquisition according to the class attribute of tags: generally speaking, the class attribute may correspond to a category of nodes; (3) Location according to id attribute of tags: some nodes have id attribute and remain exclusive generally in a specific page, so the location is accurate precisely; (4) Use of fixed texts in the page: Such texts are of good stability generally, few of which is changed during the revision, so these texts usually function as prompt texts of structured data to locate the data with a higher requirement for precision.

During the actual data extracting, considering that the webpage markup nesting of partial list pages is complicated while the detail page is abundant in contents, complicated in format and hard to be located, this system rapidly locates the tag nodes and data separators, and precisely locates the data record blocks by integrating a browser extension tool (DOM Inspector) and the semi-automatic customization tool for data extraction templates in the system. The algorithm procedure is as shown in Figure 5.

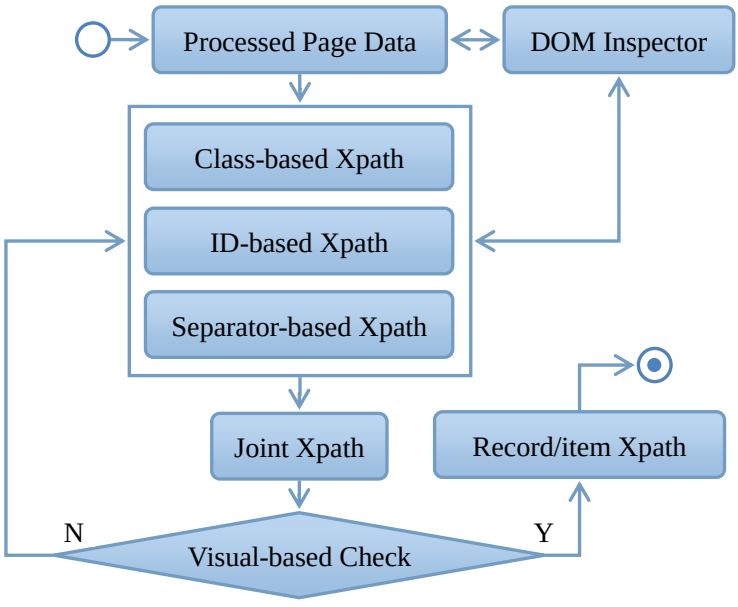


Figure 5 Information path location algorithm flowchart

3.1.4 Rule template customization

Upon passing the test, the data location rules for key fields form the extraction template for specific journals together with the journal information and field information. The path of fine granularity field-level data tagging (XPath), string expression, field and other information are submitted to the backend server for saving. As there are two types of page, it needs to generate two types of page wrapper, namely, list page wrapper and detail page wrapper.

For detail page, the structure of matching template shall be: ＜ITEM XPATH：ITEM STRING EXPRESSION/ID[;…],(SEPARATOR),(IDENTIFIER)＞.

Among them, ITEM XPATH is to locate to the path of a certain field node; ITEM STRING EXPRESSION is to extract the string expression of the field contents; SEPARATOR and IDENTIFIER are optional rules, of which the former is used to separate the repeatable fields, while the latter is used to replace the abnormal characters of the data.

For the list page, the location needs to be made to the record and then specific fields therein. The structure of matching template should be: ＜LISTRECORD XPATH：LISTRECORD STRING EXPRESSION＞＜ITEM XPATH：ITEM STRING EXPRESSION[;…],(SEPARATOR),(IDENTIFIER)＞

In this system, the extraction can precisely converge to the required field information in the form of iteration. In most cases, 3-4 iterations can help achieve the satisfactory effect. The string expression can be one/more page identifiers or the combination of one/more page identifiers and special characters.

3.1.5 Intelligent correlation of rule template and seed page

Observation reveals that an open access data source generally has a few OAJ resources. These journal resources are highly similar to one another in terms of page DOM tree structure, URL structure, page data structure, etc. The data extraction templates relating to these journals can be shared in the training process of similar, new seed pages. Hence, when a new seed page is added to the system, the system will execute the URL similarity matching with the existing extraction templates of seed page in the template library, to form the intelligent correlation and interaction between rule templates and seed page, and reduce the human intervention and errors during the template customization. The related algorithm is specified as below:

Input:page seed URL;//a new page URL.

Output:<Seed, Template>; //the seed similar to certain template.

Begin

Extract parameters of the Top N template site similar to the seed URL, including:items, items_string_expression, listrecords, separator, identifier, listrecords _string_expression, etc.;

For items in template_items order by similarity do

Get all items rules;

If seed.items_rules == null Create_items_rules(seed, seed.default_items_rules);

Get listrecords rules in template_listrecords;

If seed.listrecords_rules == null Create_listrecords_rules(seed, seed.default_ listrecords_rules);

End

3.2 Extraction stage

Through the training stage of data extraction rule templates, the system has obtained the URL of target journal seed and its related data extraction template. At the stage of data extraction, the system can achieve the automated data extraction of target journals through task dispatcher program and in combination of existing seed URL and templates. The major functions include: data update check module, data extraction module.

3.2.1 Data update check module

Observation reveals that the updates of OAJ page generally include: data update of new journals and update of page data structure. The simplest method of checking the update of target page data is to translate the page data into the hashed value with the fixed length output based on the HASH algorithm and then make a contrast, namely, to compare the changes of MD5, the hashed value of page data at the different time. The same value explains there is no change in page data; otherwise, the page data is changed and updated. The extraction rules in the template library can be applied to further confirm whether the page update is for journal data or page data structure, and the rationality can be validated by verifying the extraction results. If, in the list page, the number of records can be extracted normally, the page update is for journals; if the extracted data length is abnormal in the detail page, such as the date length of year of publication, the page update is for data structure of the page, whereby the data extraction rule template shall be re-customized.

In order to automatically check the data updates and changes, the system will automatically download the page data of target journals and check the data changes, through the time-based task dispatcher program and URL value of target journals saved in the template library; in case of any change, the system will notify the system backend to launch the data update task and assign the event-based task dispatcher program to complete the updating progress, after saving the related parameters; the system will automatically complete the extraction and data saving of update data of target journals, through the event-based task dispatcher program and the data update module in the template library. The algorithm procedure is as shown in Figure 6.

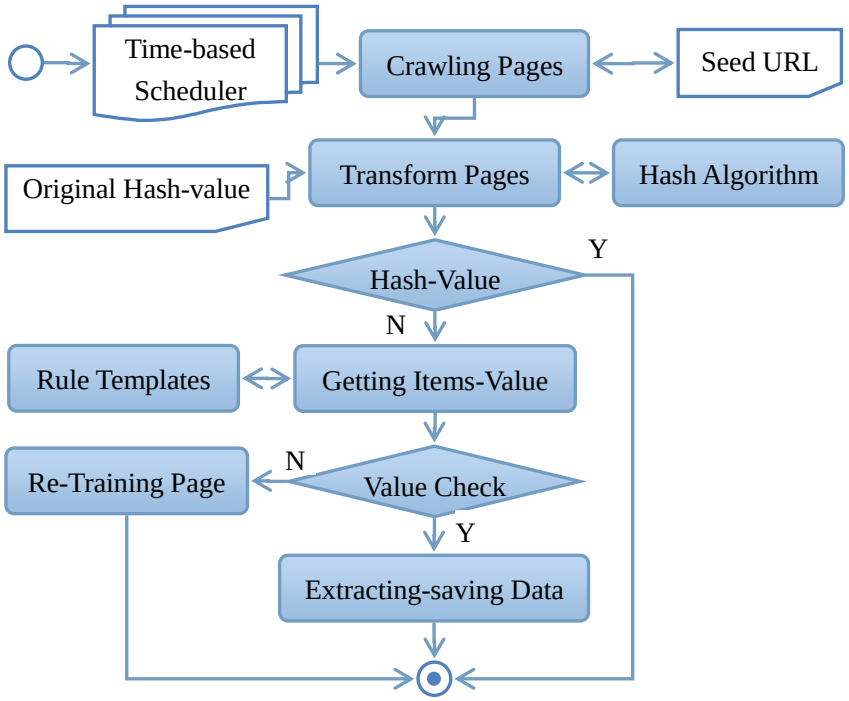


Figure 6 Data update monitoring algorithm flowchart of OAJ websites

3.2.2 Data extraction module

Upon detecting a new data processing task of target journal updates, this system will first select an appropriate gathering engine according to the gathering template of target journals, acquire the page update data of target journals and execute the standardized data preprocessing; then, the system will automatically confirm whether it is list page, article page or detail page according to the features of the page, and select an appropriate wrapper to analyze and extract the related page data; at last, the system will save the extracted data in the backend database.

The analysis and extraction process of page data is varying from wrappers. If the extracted webpage is the list page or article page, the major process is as below: (1) Locate the record through XPath, the nesting information block path; (2) Locate to a certain field node by field XPath; (3) Extract the final text after filtering the content through string expression. For detail page, the step (1) is skipped.

4. Case validation

To validate the validity of the method of OAJ fine granularity webpage data extraction, this paper randomly samples three OAJ resources of two websites from OA journal resource sites at home and abroad: AASRI Procedia, APCBEE Procedia and Slovak Journal of Civil Engineering (SJCE) to conduct a test. Meanwhile, with regard to the fine granularity extraction of key fields of the above three journals, this paper conducts a statistical evaluation and test. These fields include: paper title, author, affiliation, year of publication, key words and abstract, covering some key data of the list page and detail page. The

experimental data is as shown in Table 1 and 2. The experimental effect is subject to the measurement indicators, namely, precision ratio, recall rate and weighted geometric mean F index of precision rate and recall rate. The precision rate is the proportion of correct effects among all the acquired answers. The recall rate refers to the proportion of correct effects among all the answers (including those obtained answers and those that should be ignored). The computing formula is as below:

PRE = A/(A + B) (1)

REC = A/(A + C) (2)

F = (WEIGHT × WEIGHT + 1. 0) ×PRE×REC /(WEIGHT × WEIGHT ×PRE + REC) (3)

In the formula (1) and (2), A represents the number of related webpage information as extracted thereto. B represents the number of unrelated webpage information as extracted thereto; C represents the number of related webpage information which is not extracted. In the formula (3), WEIGHT is the relative weight of recall rate and precision rate. When WEIGHT is equal to 1, both aspects are important; when WEIGHT is more than 1, the precision rate is more important; when WEIGHT is less than 1, the recall rate is more important. In this experiment, the precision rate of data extraction is more important. Thus, the value of WEIGHT is set to 2.

Table 1 OAJ acquisition and its statistical & evaluation results

Journal	Number of correct extraction	Non-extracted number	Number of questionable extraction	Total number of extraction	Number of actual pages	Recall rate (rec)	Precision rate (pre)	F index
AASRI	381	0	0	381	381	100%	100%	100%
APCBEE	516	0	0	516	516	100%	100%	100%
SJCE	104	0	0	104	104	100%	100%	100%

Table 2 Fine granularity information extraction of OAJ and its statistical & evaluation results

Fine granularity information	Number of correct exhibition	Non-extracted number	Number of questionable extraction	Total number of extraction	Number of actual pages	Recall rate (rec)	Precision rate (pre)	F index
Title	1001	0	0	1001	1001	100%	100%	100%
author	998	0	3	1001	1001	100%	99.7%	99.76%
year	1001	0	0	1001	1001	100%	100%	100%
keyword	994	0	7	1001	1001	100%	99.3%	99.44%

abstract	989	0	12	1001	1001	100%	98.8%	99.04%
affiliation	998	0	3	1001	1001	100%	99.7%	99.76%

Experimental results show that it is of higher precision rate and recall rate to extract OAJ list page and detail page of by the method of OAJ fine granularity webpage data extraction The extraction effect of title, author and year of publication is slightly better than that of affiliation, key words and abstract, which is attributed to the hierarchy definition of webpage DOM. Those articles that the author fails to extract are concerned with the information of organization and table of contents. There is no data made by the author. Experiments show that the DOM definition of the article/list page is better than that of detail page, which also agrees with the actual situation. The detail page is richer in contents and more complicated in structure, and applies more data structures or identifiers, e.g. standard format of literature writing and record (MARK), especially the data of references. In addition, Javascript is indeed more influential to the acquisition effect, so the acquisition quality is relatively low.

5. Conclusion

The main difficulty in extracting the OAJ fine granularity webpage data is how to locate and extract the fine granularity data. However, the key to solve the problem is to adopt different data location and data extraction rules to form data extraction templates according to different types of page. As the settlement of such problems are challenged by such facts as complicated page structure, less hierarchy of page markup nodes, and vague fine granularity data structure, it is hard to solve it by the traditional methods based on the webpage markup language processing technology and based on the processing technology of visualization characteristic. To this end, this paper puts forward a new solution to fine granularity web information extraction that integrates webpage markup language processing technology and NLP technology, on the basis of analyzing the features of OAJ websites and summarizing the major extraction methods for DEEP WEB sites. The results show that this new method can extract the fine granularity page data with high quality, and effectively integrate the OAJ fine granularity webpage data and form the services. Surely of course, this system still needs to be further improved, such as: gathering engine optimization of dynamic script page, optimization of citation data extraction precision, data quality control, etc., all of which will be a direction in which this system will continue to concern and improve.

References:

[1] Frandsen T F. The integration of open access journals in the scholarly communication system: three science fields[J]. Information Processing and Management，2009, 45(1): 131-141.

[2] Eikvil, L.Information Extraction from World Wide Web – A Survey[R].Norwegian Computing Center, 1999.

[3] LIUB, GROSSMANR, ZHAIY.Mining data records in Web pages[C] //KDD2003, 2003:601-606.

[4] MENG Xiaofeng, LU Hongjun, etal.SG-WRAP:a schema guided wrapper generator data engineering[C] //Proceedings of 18th International Conference on Data Engineering, 2002.

[5] ARASU A, GARCIA-MOLINA H.Extracting structured data from Webpages[C] /ACMSIGMOD Conference, 2003.

[6] ANTONT.XPath-Wrapperinductionbygeneralizingtreetraversalpatterns[C] //LWA2005, 2005: 126-133.

[7] PARK J, BARBOSA D.Adaptive record extraction from Web pages[C] //WWW2007, 2007:1335-1336.

[8]CRESCENZIV, MECCAG, MERIALDOP.RoadRunner:towards automatic data extraction from large Websites[C] //VLDB2001:109-118.

[9] WANGJ, LOCHOVSKYFH.Data extraction and label assignment for Web databases[C] // Proceedings of the 12th International Conference on WorldWideWeb,2003: 187-196.

[10] ZHAI Yanhong, LIU Bing.Automatic wrapper generation using tree matching and partial tree alignment[C] //2006 American Association for Artificial Intelligence, 2006.

[11] ZHAI Yanhong, LIU Bing.Web data extraction based on partial Tree alignment[C] //WWW 2005, 2005.

[12] SIMON K, LAUSEN G.ViPER:augmenting automatic information extraction with visual perceptions[C] //Proceedings of the2005 ACM International Conference on Information and

Knowledge Management(CIKM ’05), Germany, 2005.

[13] ZHAO Hongkun, MENG Weiyi.Fully automatic wrapper generation for search engines[C] //WWW 2005, Japan, 2005.

[14] LIU Wei, MENG Xiaofeng, MENG Weiyi.Vision-based Web data records extraction [C] //Ninth International Workshop on the Web and Databases(WebDB2006), Chicago, 2006.

[15] CAID, YU S, WEN J, etal.Extracting content structure for Webpages based on visual representation [C] //APWeb,2003:406-417.