IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON RELIABILITY IN SENSOR-CLOUD SYSTEMS AND APPLICATIONS (SCSA)

Received August 22, 2020, accepted September 11, 2020, date of publication September 15, 2020,
date of current version September 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3024194

VB-PTC: Visual Block Multi-Record Text Extraction Based on Sensor Network Page Type Conversion

JIBING GONG[1,2,3], HEKAI ZHANG[1,2], WEIXIA DU[1,2], HUANHUAN LI[1,2], AND HONGNIAN WEN[4]

[1]School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China
[2]Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province, Yanshan University, Qinhuangdao 066004, China
[3]Key Laboratory for Software Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, China
[4]School of Information Science and Engineering, Shijiazhuang Institute of Railway Technology, Shijiazhaung 050041, China

Corresponding author: Hekai Zhang (hekai_zhang@stumail.ysu.edu.cn)

This work was supported by the Hebei Key Research and Development Project under Grant 20310101d.

ABSTRACT Usually, in addition to the main content, web pages contain additional information in the form of noise, such as navigation elements, sidebars and advertisements. This kind of noise has nothing to do with the main content, it will affect the tasks of data mining and information retrieval so that the sensor will be damaged by the wrong data and interference noise. Because of the diversity of web page structure, it is a challenge to detect relevant information and noise in order to improve the true reliability of sensor networks. In this paper, we propose a visual block construction method based on page type conversion (VB-PTC). This method uses a combination of site-level noise reduction based on hashtree and page-level noise reduction based on linked clusters to eliminate noise in web articles, and it successfully converts multi-record complex pages to multi-record simple pages, effectively simplifying the rules of visual block construction. In the aspect of multi-record content extraction, according to the characteristics of different fields, we use different extraction methods, combined with regular expression, natural language processing and symbol density detection methods which greatly improves the accuracy of multi-record content extraction. VB-PTC can be effectively used for information retrieval, content extraction and page rendering tasks.

INDEX TERMS Dom trees, HTML documents, noise elimination, web data extraction, web mining.

I. INTRODUCTION

With the rapid development of the world wide web(WWW), the internet has become the main source of information that can be accessed in the form of web articles [1]. To improve the reliability of data collection, more and more researchers are relying on cloud processing and decision-making to find useful information from expanded knowledge sources [2], [3]. Content extraction (CE) is a technique used to determine the correct part of an HTML document that contains the main content of a web document [4]. Usually, web pages contain the main content, but this information is also surrounded by some additional information, such as anchor tags, advertisements, and various navigation information. For topic classification, only the main content is very important, and any noise will have an adverse impact on the classification [5], [6]. The sensor may not receive data at all, or the received data may be damaged, or it may be susceptible to errors and interference (such as noise) [7], [8]. Due to the diversity of web structure, CE is a difficult task.

In today's web templates are an important reason to be concerned. Templates account for about 40%–50% of web content [9]. The content of the original misleading search engine, page classification, clustering, link analysis, and other applications that provide advanced text analysis web content appears repeatedly on a website [10], [11]. In addition, in some applications, it is important to evaluate whether the page content has changed.

In recent years, the research on web page segmentation technology has been widely concerned, and has made a wealth of research results [12]. Web page segmentation technology is based on the visual characteristics of people, summarizes some rules of web page segmentation, and then realizes web page segmentation based on these rules

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Zakirul Alam Bhuiyan.

[13], [14]. Since then, many researchers have proposed many improved web page segmentation technologies based on this method [15], [16], but the idea of rule-based segmentation technology has no essential change. At present, there are two main problems in vision based web page segmentation technology. First, the result of web page segmentation is too fragmented, which is not conducive to web page reconstruction; Second, the segmentation rules are too targeted, which is not conducive to the portability of the program [17], [18]. Therefore, how to divide the granularity of web page segmentation, how to make general rules of web page segmentation, and how to improve the robustness of the algorithm are all problems that need further study.

This paper mainly focuses on a multi-record complex page, after noise reduction, the page will be converted to a multi-record simple page. On the basis of making full use of web content information, Dom structure information, and visual information, a more robust and efficient single page web information extraction method, i.e., visual block construction method based on page-type conversion (VB-PTC) is proposed. **Contributions in this paper are as follows:**

- We propose a Site-level and Page-level noise reduction method based on hashTree and linked clusters, respectively. Through the combination of site-level and page-level methods, the template data in the web page is removed to complete the noise reduction processing on the web page, which effectively solves the problem of an inconsistent number of template nodes in the web page template.
- Convert multi-record complex pages to multi-record simple pages, streamline visualization block building rules, and make the algorithm portable.
- In the multi-record page, different extraction methods are used for different field contents, reasonable use of field features, and improve the accuracy of field matching.

The structure of the rest paper is organized as follows: in Section II, we introduce the existing methods of page noise reduction and content extraction. In Section III, we introduced the technical terms and concepts that appeared in the article. In Section IV, the process of page noise reduction based on site-level and page-level methods is described in detail, and in Section V, the rules of visual block construction and the method of multi-record concent extraction are given. In Section VI, we introduce the evaluation index and compare it with the latest algorithm. In Section VII, we summarize and tell the ideas and plans for the future work.

II. RELATED WORK

In this section, we review related work in the following three aspects.

A. DATA COLLECTION

Currently, the commonly used data acquisition method is to construct the page url through regular expressions, and download the page through BeautifSoup etc. [19]. However, this method has two main disadvantages: First, the URL constructed by using regular expressions is not completely accurate, the URL be invalid; Second, this method of generating connections in batches cannot be based on user behavior and needs. Using sensor networks to record user behaviors to obtain page addresses can effectively solve the above-mentioned two problems [20]. Selvaraj *et al.* [21] introduced a new hash and time-based security method for secure data downloads in wireless sensor networks. Ling *et al.* [22] proposed a distributed large-scale wireless sensor network data partition processing model. Large amounts of data can be obtained quickly. However, data acquisition without filtering can lead to data redundancy. Yin *et al.* [23] proposed a new method of vibration monitoring based on a wireless sensor network, which can realize vibration data collection, online detection, and data analysis. The user behavior can be obtained through mouse and keyboard clicks.

B. PAGE NOISE REDUCTION

The detection of noise elements in web pages has become a research field. Based on the technology of eliminating noise, a novel method is introduced in [24]. By identifying and eliminating noise from web pages, the performance of web mining data can be improved. The purpose of this method is to remove noisy content tags and extract the information of the main content tags from web pages. A system has been deployed to detect and eliminate noise in web pages without using any information about the main content. The system uses features such as the length of the anchor text and the number of punctuation marks to identify noise in web pages [25]. Similarly, the model developed in [26] performs noise removal in two phases: feature extraction and clustering. The system uses content extraction algorithm to extract text content. After observing the HTML tags of the extracted content, the system uses the concept of line block to find the distance between any two adjacent lines for noise classification. The system uses a variety of text functions, such as the ratio of text to label, the ratio of anchor text to text, and the density of title, keywords. In addition, the noise element detection of MLP (Multi-Layer Perceptron) is also carried out [27]. The results show that the linguistic features of sentence length play a key role in noise classification. Supervised machine learning method has the limitation of marking the training data set. Therefore, an unsupervised method is developed in [28] to extract noise. The algorithm uses visual features to allocate pages to multiple blocks, and uses a hybrid hash algorithm to calculate the importance of each block. The final filtering of noise is performed using the importance threshold of each block.

Web pages are usually heterogeneous [29]. In order to improve the performance of the mining web, Debnath *et al.* [30] assigns a weight to each block in the web page. This technology uses a compressed tree to construct data, and then calculates the weight of each node. Identify different noises and calculate their importance. In order to extract core information from web pages,

a technology [31]–[33] is introduced. In most cases, web pages contain heterogeneous data, and it is difficult to automatically identify the data. In this method [34], candidate keywords are used to list the content sequence in the web page to which the filter is applied. This is done by using the algorithm of the set theory. Naseer *et al.* [35] compares the famous noise removal techniques of web pages. Their results show that most methods use the Dom tree structure to detect and eliminate noise. The Dom tree structure is used for noise removal in [36]. The system deployed a three-stage algorithm. First, complete feature selection is performed. Next, create the feature Dom tree is created, and finally,use the feature Dom tree for noise detection. Enhanced Dom tree and context features are also tested in [37]–[39] for noise detection.

C. CONTENT EXTRACTION

At present, the advanced method is based on visual information, which pre-renders the target web page through a browser interface or kernel. Then, the web data record is extracted based on the visual law of the web page. Cai *et al.* took the lead in proposing the classic VIPS [40] algorithm, which first extracts all the appropriate page areas from the Dom tree, and then reconstruct the semantic structure of the web page according to these pages and split bars. As an extension of ViNT [41], ViPER [42], and ViDE [43] have also successfully used the visual features of web pages to achieve data extraction. CMDR [44] is to learn the features of multi-record pages through neural network, and combine the MDR method based on the Dom structure information to mine the data area of community forum pages.

Different from the above methods, in recent years, some scholars have proposed to extract web pages by using a convolutional neural network (CNN) method based on the visual similarity of web pages. VIBS [45] applies CNN (Convolutional Neural Network) in image field to the screenshots of web pages, and at the same time uses similar VIPS to extract web pages. The algorithm generates a visual block, and finally combines the results of the two stages to identify the text area of the web page. Gogar *et al.* [46] proposed a text map and added context to divide the visual area of the web page. These methods have high extraction accuracy in the websites which have not seen the model but are similar in vision. However, these methods only consider the characteristics of web snapshot, and do not use the characteristics of web elements, that is, Dom tree nodes. Moreover, the training speed of CNNs is slow, so they are not applicable in practical engineering.

III. PERELIMINARY

A. TEMPLATE DATA

At present the Internet is an extremely rich source of information and data. However, in most web pages, the main content is accompanied by non-information parts, such as navigation menu, link list, header and footer, copyright notice, advertisement, etc. These elements are often referred to as **template data**[1] [47].

Template data is usually defined as non-information part outside the main content of web page, which is usually generated by a machine and repeated on the same web page. Although some elements, such as navigation menus or advertisements, are easily recognized as templates, it may be difficult for other elements to determine whether they are templates according to the previous definitions. For example, in a journal paper page, in addition to the title and abstract of a paper, it also includes *similar article recommendation* generated by the system are also included. These recommended articles contain topics and links to the full text. Obviously, this kind of information with a user's subjective tendency and generated dynamically is not what we need.

The methods of automatically removing template data can be divided into two categories [48]: page-level and site-level. The page-level algorithm processes each web page separately. In the site-level method, multiple pages from the same site are processed at one time. As shown in Figure 1, the black area is web site-level noise, the blue part is page-level noise, and the yellow part is the body content.

B. PAGE TYPE CLASSIFICATION

In order to adapt to people's reading habits, the various components and modules in the web page are usually arranged according to a certain visual rule, that is, the web page layout. According to the complexity, it can be divided into several types [49], as shown in Figure 2. The single-record page is shown in Figure 2 (a). A page contains only one data record. Common single-record pages include news pages, article pages, etc., whose text information occupies a prominent position on the page, and there is less noise information on the web page. The previous methods focused on this type of page and obtained good results. A simple multi-record page is shown in Figure 2 (b). The data records are arranged in a list on the web page, and the list of data records in the page occupies the most important part. Common multi-record pages include catalog pages, search results pages, etc. At present, people are most exposed to multi-record complex pages, as shown in Figure 2 (c). There are various semantic blocks in the page, which are distributed around the area where the data record list is located. This type of page includes a community forum page and a research paper display page.

C. SENSOR NETWORK PAGE DATA ACQUISITION

The CGI (Common Gateway Interface) [50] provides a channel for an external program on the web server. This service endpoint technology enables interactivity between the browser and the server. CGI is a program that running on a

[1]Delete template data types, such as navigation, link lists, copyright notices, template materials, such as: headers and footers, advertisements, and duplicate materials (such as fixed formats containing author information in a paper page).

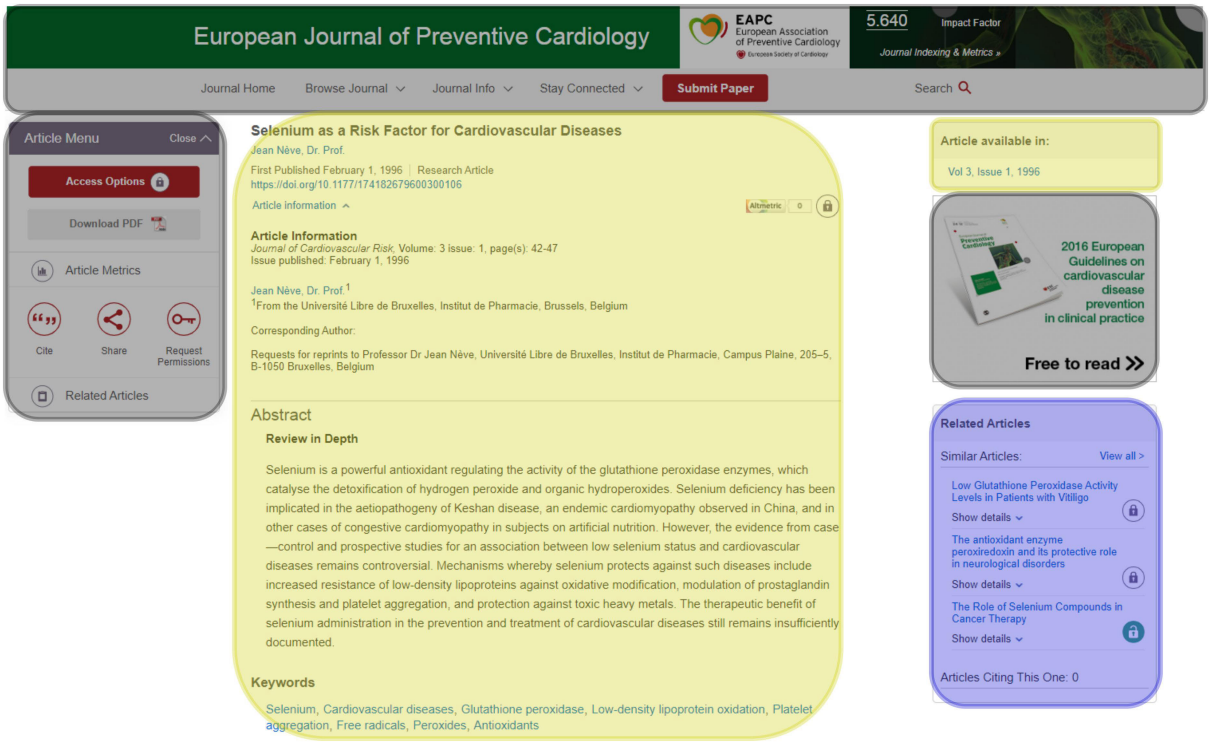

FIGURE 1. Template data and text content display in the page.

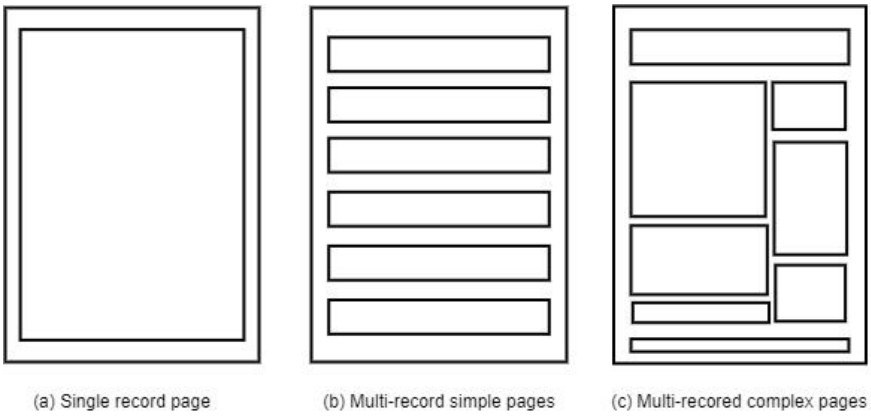

FIGURE 2. Three common types of web pages.

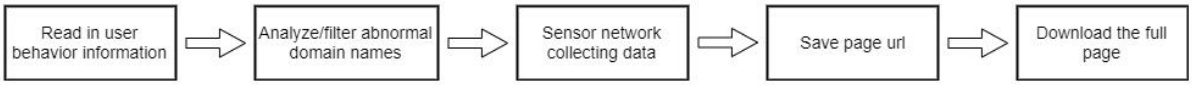

FIGURE 3. CGI data collection process.

web server, that is triggered by the input of a browser, and is a bridge between the server and other programs in the system. A CGI program is an external program, which is an executable file running on the server.

When CGI is used to realize remote sensor pressure data collection, the CGI program can obtain the data by directly accessing the hardware or calling the driver (this article uses the user's click behavior). Just like ordinary data collection, after data collection, the CGI program organizes the data into HTTP streams and sends them to the web server. The web server is responsible for sending it to the client, and finally get the page link that the user has browsed. In order to filter abnormal user behavior, we will set the domain name information, and pages that are not within the scope of this domain name will not be collected. The CGI data collection process is shown in Figure 3.

IV. MULTI-RECORD COMPLEX PAGE TYPE CONVERSION

A. PAGE CLEANING

After getting the HTML code of the page, we do not recommend noise reduction immediately. The page contains a lot of style information and fragment tags, which is a serious obstacle for us to build the Dom tree next. In order to avoid a deeper Dom tree, we first clean and preprocess the page. After many experiments, our method is observed to be suitable for most complex multi-record pages. The page cleaning process is as follows:

- **Delete page notes.** When building a Dom tree, page annotations will be converted into annotation nodes and become part of the Dom tree, which is obviously not what we need, because we cannot get any useful information from the annotations.
- **Delete unused tags.** Useless tags refer to < style >, < script >, < link >, < noscript >, < iframe >, < video >, < picture >, < img >, < form >, < input >, and < meta >. This kind of label mainly contains the style information and non-text information of the page. Deleting these labels is equivalent to rendering the page, so that all the contents of the page are displayed in the form of a single column.
- **Delete the label and merge the text content.** We just need to delete the HTML tags in and keep the text information in the tags. Such labels include < em >, < strong >, < b >, < sub >, < sup >, < i >. These are fragment tags. Fragment tags have no practical significance and are only used for auxiliary text content expression.
- **Delete the empty node.** Empty node is defined as: there is no child node under this node, and there is no text content. There may be a large number of empty nodes in the original HTML page or the HTML page after deleting the above tags, which need to be deleted recursively from the bottom up.

B. SITE-LEVEL NOISE REDUCTION METHOD BASED ON HASHTREE

We use the Dom structure of website pages to search for Dom tree nodes that repeatedly appear on multiple pages of the website to find template data. For each page, we define it as a triple $P = (\Omega, \Phi, \eta)$, where:

- $\Omega = (P_1, P_2, \cdots, P_n)$ represents the node collection of the Dom tree on a given page. There is no overlap between these Dom trees, and each Dom tree P_i can be defined as the triple $P_i = (\Omega_i, \Phi_i, \eta_i)$ described above, so nested loops like this.
- $\Phi = (tag, attrib, text)$ represents the root node information of the current Dom tree, including the tag name of the node, the attribute name and value of the node, and the text content under the node. Since there may be more than one attribute of a node, we take all the attributes owned by the node as the attribute information of the node. The text content is all the text information contained between the start label and the end label of the node.
- η represents the unique identification information generated by Φ. We call it the fingerprint of the node. We use three kinds of information in Φ to generate the fingerprint of a node (that is, a Dom tree whose root node is this node). In this paper, η is calculated by hash algorithm, because the hash function is simple to calculate and fast to run. It is worth noting that in order to avoid the error caused by the irregular writing of HTML, we delete all spaces in the text content during hash calculation, which is based on a large number of experiments.

Figure 4 shows the process of generating the node fingerprint from the sample code. First, we need to get the HTML code of the page. Taking the code in Figure 4 (a) as an example, we construct the Dom tree of the code, as shown in Figure 4 (b). Then, from top to bottom, we traverse each node of the Dom tree hierarchically, extract the label information and content information of the node, as shown in Figure 4 (c), and calculate the fingerprint of each node in the process of traversal. The fingerprint calculation method used in this paper is hash algorithm. After the above operations, we can get a hash tree corresponding to the Dom tree. It is worth noting that this hash tree will not be constructed in the actual operation, but is a logical way of existence.

In order to facilitate site-level to find template data in multiple pages, we choose a pair of pages to compare each time during the experiment. We still adopt the method of hierarchical traversal, at the same time, we compare the fingerprint of each node in two Dom trees, and delete the nodes with the same fingerprint. Due to the characteristics of template data generation, template data will only add sibling nodes in the same layer, but not increase the depth of the tree. For example, there are different numbers of *authors* in two pages, and these authors and author information are siblings of each other. The number of nodes containing author information in the two pages is different. So in the process of comparing nodes, we only need to consider whether the same fingerprint exists in the same layer of nodes.

In the process of page comparison, we also need to set a set of ϕ_i for each page to record the fingerprint of each layer node, where $i = \{1, 2, \cdots, n\}$, n is the number of comparison pages. In the comparison of the two pages, we will judge whether there are duplicate elements in ϕ_1 and ϕ_2, and delete the nodes corresponding to the duplicate elements in the Dom tree to delete the template data in the page. It should be noted that the proposed method effectively solves the problem that the number of template nodes in the web template is not the same, that is, the number of author nodes above is different from the number of other page author nodes.

The site-level noise reduction method for implementing hashtree is shown in Algorithm 1.

C. PAGE-LEVEL NOISE REDUCTION METHODS BASED ON LINK CLUSTERS

The page-level template data deletion method takes a single web page as input. This makes them more flexible and easy to use than site-level methods. The page-level algorithm mainly

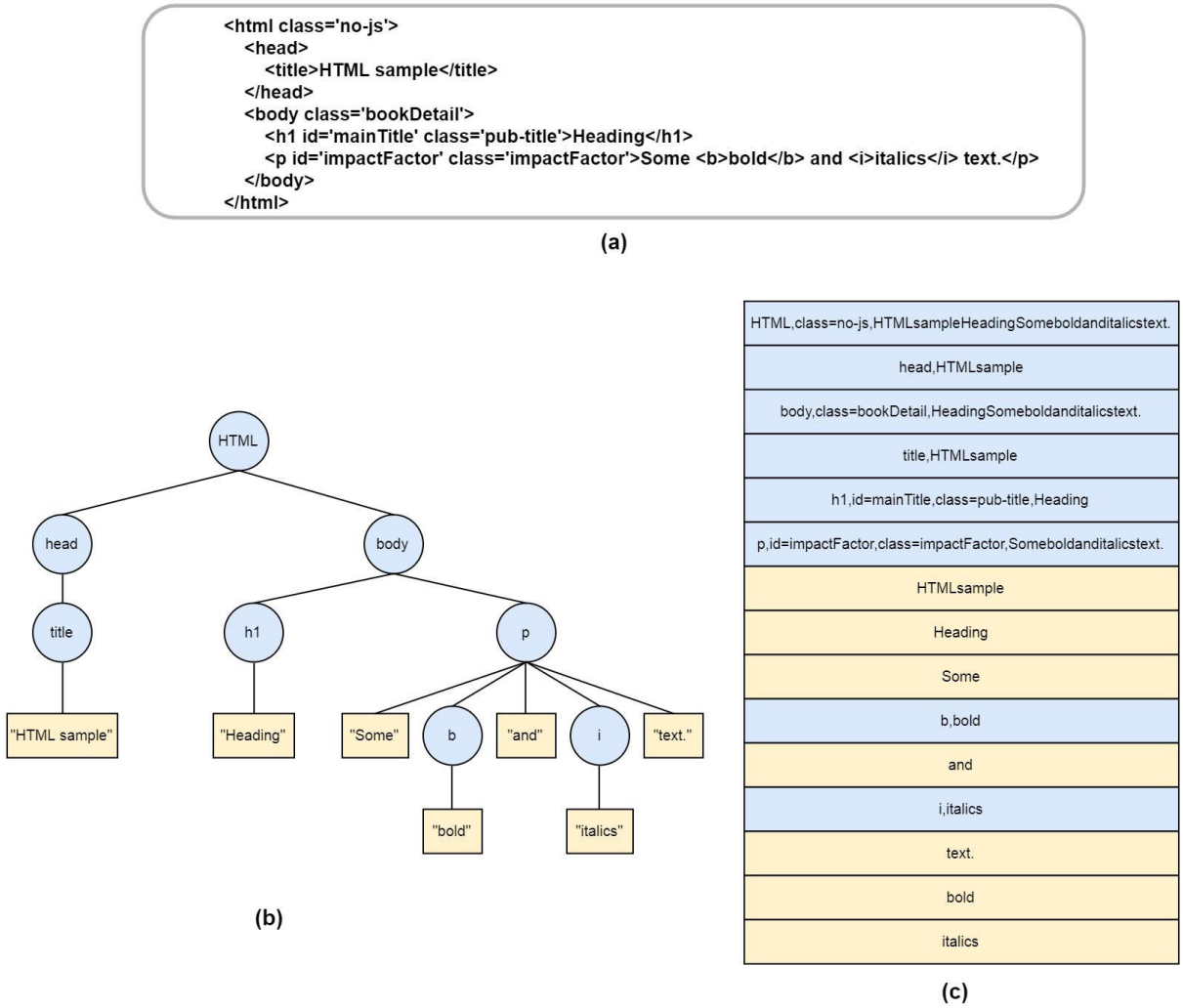

FIGURE 4. The extraction process of node fingerprint.

deletes the dynamic template data in the page, that is, the blue part in Figure 1, such as **recommendation**, **relevant literature**, **guess you like** and so on. However, due to its dynamic nature, it cannot be deleted according to page comparison. But noise links are usually grouped in clusters. Different from text links (such as keywords, author name, etc.), noise links contain longer text content. We propose a text density noise links removal method based on a link cluster.

If node i is a node of the Dom tree, the text density LTD_i based on the link cluster under this node is:

$$LTD_i = \frac{LTG_i}{TG_i} \times \frac{LT_i}{T_i} \times \sqrt{LT_i} = \frac{LTG_i \cdot LT_i^{3/2}}{TG_i \cdot T_i}. \quad (1)$$

where T_i is the total number of text words of node i divided by spaces, TG_i is the number of tags in node i, LTG_i is the number of link tags < a > in node i, and LT_i is the total number of text words in link tags < a > in node i. $\frac{LTG_i}{TG_i}$ is the density of linked clusters, and the more < a > tags in nodes, the more likely it is to be linked clusters. $\frac{LT_i}{T_i}$ is the text density, and the higher the value, the more likely the noise links is. $\sqrt{LT_i}$ is a regularized term, and usually noise links contain longer text content. The text density LTD_i of the nodes is shown in Figure 5. We can find the nodes with the largest text density, but the text density of each node is similar. In many web pages, this method can not accurately locate noise links.

In order to solve the above problems and further improve the search hit rate, a mathematical model is established to expand the difference between text density.

$$score_i = \log(SD) \times LTD_i \times \log 10(LTG_i + 2), \quad (2)$$

where SD is the standard deviation of the node text density. We extract the results of a webpage based on this rating. From Figure 5, we can see that the difference between the text density of the two Dom tree nodes is 42, and after filtering by Equation 2, we can see from Figure 6 shows that the smallest difference between the Dom tree node scores is 1130. Enlarging the difference in text density is more conducive to finding noise links in dynamic template data.

Algorithm 1 Site-Level Page Comparison Noise Reduction Algorithm

Require: Dom tree structure of web page: $DomTree_1, DomTree_2$
Depth of trees in $DomTree_1$ and $DomTree_2$: D_1, D_2
Storage node fingerprint list: ϕ_1, ϕ_2
Ensure: HTML code of site-level page after noise reduction: $HTML_1, HTML_2$
function FingerPrint($Layer$)
    $L \leftarrow \phi$ // Record the hash value of nodes in this layer
    for $node \rightarrow Layer$ **do**
        // Space between words should be removed from text in node
        $t = node.tag + node.attrib + node.text$
        //In this paper, the hash algorithm is used to calculate the unique identification of nodes
        $FP = hash(t)$
        $L.append(FP)$
    end for
    return L
end function

function DelNode($DomTree_1, DomTree_2, i$)
    // Get all nodes of layer i in Dom tree
    $LayerA = DomTree_1[i]$
    $LayerB = DomTree_2[i]$
    // Get the hash fingerprint of the node in this layer
    $hashNode_1 = FingerPrint(LayerA)$
    $hashNode_2 = FingerPrint(LayerB)$
    $H \leftarrow hashNode_1 \bigcap hashNode_2$
    $DomTree_1, DomTree_2 = del(H)$
end function

for $i = 1 \rightarrow min(D_1, D_2)$ **do**
    $DelNode(DomTree_1, DomTree_2, i)$
end for

// Reconvert Dom tree to HTML code
$HTML_1, HTML_2 \leftarrow transformDomTree_1, DomTree_2$
return $HTML_1, HTML_2$

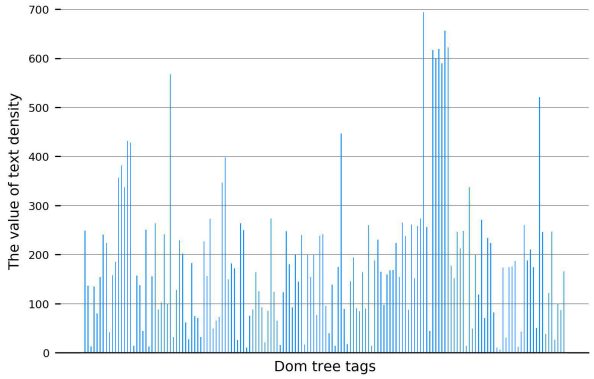

FIGURE 5. Text density of each node in Dom tree.

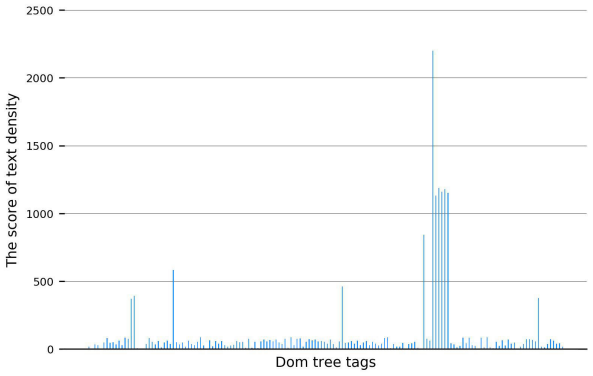

FIGURE 6. Score of each node in Dom tree.

V. BASED ON VISUAL BLOCK CONTENT EXTRACTION

After the page noise reduction processing of Section IV, we successfully converted the multi-record complex page to the multi-record simple page, as shown in Figure 7. All the content in the page is displayed in the page from top to bottom in behavioral units, that is, there is no discrete block, which is more conducive to building visual blocks.

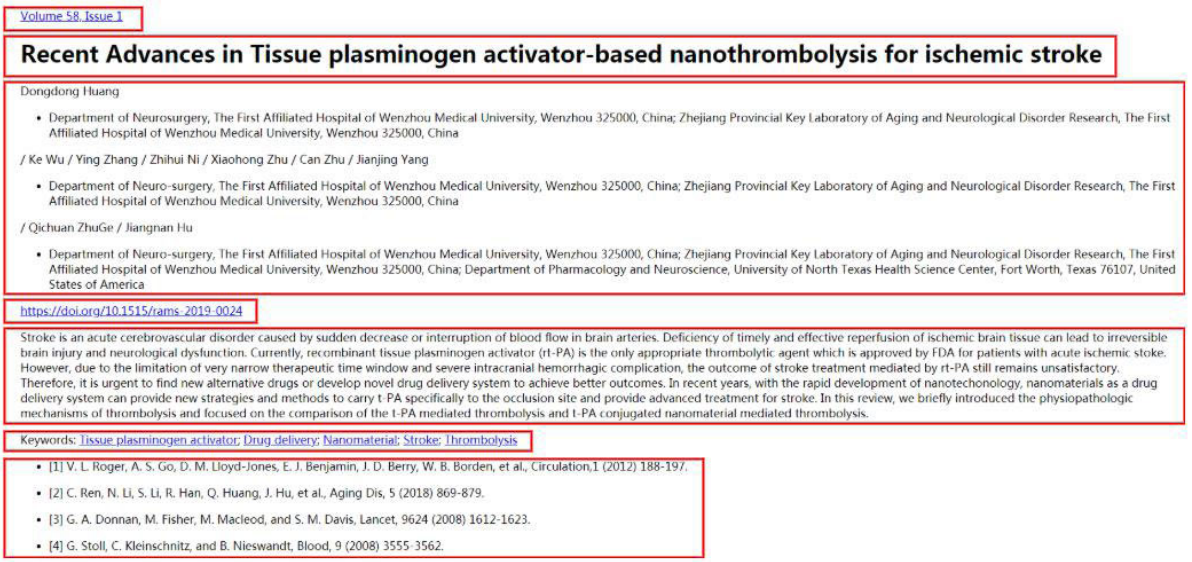

FIGURE 7. Simple multi-record page after noise reduction.

A. MULTI-RECORD CONTENT EXTRACTION

The data record is the web page element that users are really interested in. The useful information in the page is stored in different visualization blocks with the item as the unit. We only need to judge the item type of each visualization block and match it with the corresponding item. In this article, the types of items extracted from the scientific papers page are shown in Section VI-A. We use different extraction methods for different types of items to ensure the correctness of item matching.

B. VISUAL BLOCK BUILDING

In this step, we aim to find all the appropriate visual blocks contained in the current subpage. Generally, each node in the Dom tree can represent a visual block. However, some large nodes, such as < table > and < p > are used for organizational purposes only and are not suitable for representing a single visual block. In these cases, the current node should be further divided and replaced by its child nodes. In order to avoid making specific extraction rules for different web page structures, we simplify the rules for building visualization blocks.

At present, most visualization block building not only uses label and its attributes, but also considers CSS style, node size, and other information. Too many rules are not conducive to the portability of the program. We give some simple inference rules to judge whether the current node should be divided. If a node does not need to be divided, the node block will be extracted. Take the data set we used as an example, and the reasoning rules are as follows:

- <h> tag is a visualization block.
- The text under the < ol >, < ul > tags is a visual block.
- The smallest div that contains text (that is, the div contains text content, but does not contain the rest of the div tags) is a visual block.
- If the gap between the visualization blocks defined above contains text information, the gap is also a visualization block.

Through the above rules, it can help us to build visualization blocks from multi-record simple pages. On the one hand, it embodies the advantages of noise reduction processing proposed to transform multi-record complex pages into multi-record simple pages. On the other hand, it reflects the role of multi-record simple pages in refining the rules for building visualization blocks.

——Regular expression: this method can quickly and accurately get the contents of *volume*, *issue*, *doi*, *keywords* and *published time* in the page. These items have a fixed context in the web page, that is, the item name will be followed by the item content. The name and content of the item are usually placed in a string of labels. The content of the *title* item is usually stored in the < head > tag and can be easily located using XPath.

——Natural language processing (NLP): We use pynlpir[2] method to identify *authors* and *references*. We extract the place name entity, year entity, and number string entity of each visualization block, and calculate the proportion of its number in the visualization block. According to the characteristics of the *authors* item and *reference* item, if the proportion of place name entities in a visualization block is higher, it is more likely to be the content of the *authors* item, because each author has organization information. If a visualization block has a higher proportion of year entities and number string entities, it is more likely to be the content of the *reference* item, which is determined by the format of reference.

——Based on symbol density: the purpose of this method is to find *abstract* items in web pages. The calculation process

[2]https://github.com/tsroten/pynlpir

TABLE 1. Open source algorithms.

Algorithms	Description
Boilerpipe[3] [51]	
-DE	The default extractor based on a decision tree and shallow text features.
- AE	An extractor tuned towards news articles.
- ASE	Extracts only the whole sentences.
- LCE	Extracts the largest text block.
- NWER	An extractor based on predefined wordcount heuristics.
jusText [4] [52]	Designed to preserve mainly text containing full sentences.
NCleaner [5] [53]	
- default	Uses character-level n-gram models as classifiers.
- non-lexical	Relies on non-lexical text features.
Readability [6] [54]	
- .NET	A. NET (C#) port of Readability.
- Python	A Python port of Readability.

TABLE 2. Total dataset.

Number of sites	Number of journals	Number of pages	Number of items
6	32	30528	9

is similar to that in Section IV-C. We calculate the text density TD_i of each visualization block as follows.

$$TD_i = \frac{T_i - LT_i}{TG_i - LTG_i}, \tag{3}$$

The meaning of the symbols is the same as that of Equation 1. However, due to the characteristics of complex pages with multiple item, the results obtained by using text symbol density can not accurately locate abstract items. We found that, unlike other item content, abstract has a lot of punctuation marks. Therefore, we adopt a method of item content acquisition based on symbol density. Suppose sbD_i is the symbol density of the nodes in the visualization block:

$$SbD_i = \frac{T_i - LT_i}{Sb_i + 1}. \tag{4}$$

where Sb_i is the number of symbols for the text in node i. In order to enlarge the difference of the text density of the visualization block, we set up a mathematical model. The formula is as follows:

$$T_{score} = log(SD) * ND_i * \log 10(Pnum_i + 2) * \log SbD_i, \tag{5}$$

where SD is the standard deviation of node text density, $Pnum_i$ is the number of P tags of node i. The content in the visualization block with the largest symbol density score is the *abstract* item.

VI. EXPERIMENTS

Different from other single-record text extraction algorithms, our VB-PTC algorithm is used to effectively deal with multi-item content extraction of multi-record pages. In our experiment, we selected multiple journals owned by multiple publishers and randomly selected the paper page as the test data set. We compare the performance of the algorithm based on VB-PTC with 10 different content extraction algorithms developed in four open source projects. The content extracted by the algorithm is called test data, and the corresponding data is the verification data. The verification data adopts the Scrapy framework and can be accurately positioned to the field content on the page in combination with XPath. These two data sets are used to evaluate the calculation of the indicator formula.

A. DATASET

Most of the previous performance studies used small datasets, which could not guarantee the fairness of the experimental results [55]. In our work, we use a large data set to the experiment. We selected 6 publishers and randomly selected multiple journals from them to collect data from the pages under each journal. The total data is shown in Table 2. The selected journal name, page number, and item type are shown in Table 3. In the selected pages, we can not guarantee that each page has the item in Table 3. There may be some missing item. We will save the missing item as empty strings.

Please note that the web pages collected in the test platform can be displayed correctly by the web browser we use. An example of a page not displaying correctly when some images are displayed as small red crosses. But this will not have any effect on our experimental results.

B. BASELINE

We evaluated 10 different algorithms from four open source projects and compared their performance with the proposed VB-PTC algorithm. The four open source projects are described below. Table 1 lists the details of the 10 selected algorithms.

[3]https://code.google.com/p/boilerpipe
[4]https://code.google.com/p/justext
[5]http://sourceforge.net/projects/webascorpus/ files/NCleaner/NCleaner-1.0
[6]http://code.google.com/p/nreadability, https://github.com/buriy/python-readability

TABLE 3. Page data source display.

Publisher	Journal	Count	Item
Degruyter	Reviews on Advanced Materials Science	90	abstract, authors, keywords, reference, doi, title, volume, issue, published time
	Nordic Pulp Paper Research Journal	174	
	Journal Fur Die Reine and Angewadte Mathematik	230	
	Zeitschrift Fur Physikalische Chemie	1132	
	Open Chemistry	1661	
	Open Physics	1692	
Springer	Journal of The Korean Physical Society	107	
	UROLOGE	1403	
	Irish Journal of Medical Science	851	
	Annali di Matematica Pura ed Applicata	1304	
	Laser Physics	915	
Nature	British Dental Journal	1373	
	European Journal of Clinical Nutrition	801	
	Eye	683	
	Journal of Perinatology	574	
	Nature Reviews Disease Primers	211	
Wiley	Advances in Chemical Physics	1097	
	Chemistry Select	102	
	International Journal of Clinical Practice	320	
	Molecular Informatics	1172	
	Cancer Science	1187	
	Journal of Midwifery Womens Health	217	
Sagepub	European Journal of Preventive Cardiology	2613	
	International Journal of Laboratory Medicine	1216	
	Journal of Chemical Research	1567	
	Journal of Mechanical Engineering Science	1275	
	Scottish Medical Journal	80	
Rsc	Chemical Society Reviews	1534	
	Faraday Discussions	1907	
	Journal of Materials Chemistry	2820	
	Molecular Systems Design Engineering	102	
	Nanoscale Horizons	118	

- **Boilerpipe** [51]: is an algorithm that can remove advertisements and other additional information from HTML, and extract target information (such as body content, publishing time). The basic idea is to obtain a classifier through training to extract the required information.
- **jusText** [52]: The key idea of the jusText algorithm is that long blocks and some short blocks can be classified with very high confidence. All the other short blocks can then be classified by looking at the surrounding blocks. It aims to retain text that mainly contains complete so it is very suitable for creating language resources.
- **NCleaner** [53]: The core component of the NCleaner algorithm consists of two separate character-level n-gram language models for clean and dirty text. These models are applied to each identified text segment in turn. If the dirty model calculates a higher probability than the clean model, the text segment is considered to be a boilerplate and deleted. Otherwise, it is included in the final output.
- **Readability** [54]: calculates the text density of the Dom node, and also calculates the weight of the Dom node according to some common Dom attributes such as id, class, etc. Finally, the corresponding Dom block is analyzed, and the specific text content is extracted.

C. EVALUATION INDICATORS

We use the index of information retrieval, namely recall (R), precision (P), and their harmonic average F_1, to measure the performance of content extraction. We calculate these metrics for each extracted field content as follows:

$$R_i = \frac{|extracted\ relevant\ text|}{|all\ relevant\ text|}. \tag{6}$$

$$P_i = \frac{|extracted\ relevant\ text|}{|all\ extracted\ text|}. \tag{7}$$

$$F_{1i} = \frac{2 \cdot P_i \cdot R_i}{P_i + R_i}, \tag{8}$$

The number of *extracted related text* in Equation 6 and 7 is calculated using difflib,[7] which is a software library. It enables us to calculate the size of the intersection between the text marked as content in the annotation data set and the different text output generated by the algorithm used. Each text is converted into a word or tag sequence, and the longest common subsequence is the length of the extracted related text.

We express the performance of the algorithm as its overall performance of collecting page information, including all the item information in the page. Where j represents the total

[7] http://difflib.codeplex.com

TABLE 4. Comparison between single-record text extraction and baseline algorithm(%).

Algorithm	Precision	Recall	F_1
***Boilerpipe**			
-DE	73.24	71.32	78.26
-AE	77.26	74.58	81.77
-ASE	73.92	68.83	75.24
-LCE	65.45	63.82	74.31
-NWER	66.37	62.44	70.56
***NCleaner**			
-default	75.08	73.32	75.42
-non-lexical	76.41	75.69	78.76
***Readability**			
-.NET	84.73	80.97	86.91
-Python	84.49	80.25	87.33
***jusText**	90.28	87.32	90.35
***VB-PTC**	90.15	85.43	92.17

TABLE 5. Accuracy experiment results of various data sources(%).

Data Source	Precision	Recall	F_1
degruyter	91.86	87.95	95.41
springer	92.35	85.67	90.16
nature	89.17	88.82	90.17
wiley	91.32	85.41	91.64
sagepub	88.65	84.73	91.34
rsc	87.33	89.36	94.32

number of collected items in the page, and i represents a item in the page.

$$F_1 = \frac{1}{j}\sum_{i=1}^{j} F_{1i}. \tag{9}$$

D. EXPERIMENTAL RESULTS ON TEXT EXTRACTION

We compare the proposed algorithm with the baseline algorithm. Since most of the existing algorithms are for single-record pages, we regard the *abstract* item as the text content and compare it with the baseline algorithm. The experimental results are shown in Table 4, and the F_1 of the VB-PTC algorithm in single-record text extraction is as high as 92.17. At least 1.82% higher than that of the baseline algorithm. The values of the VB-PTC algorithm in precision and recall are stable above 90% and 85%, respectively. Although VB-PTC is not score the highest in precision and recall, but F_1 is the highest, which also achieves the purpose of this paper. It shows that other models can not find a perfect balance between precision and recall as done by the VB-PTC algorithm, so that the final F_1 is higher than other algorithm models.

In order to see the stability of the proposed model, we present the abstract item extraction results under different data sources in Table 5. The F_1 of the *abstract* item extraction in different data sources is more than 90%. However, in the actual operation the F_1 is much higher than this value. We will discuss the reason for the error in Section VI-F.

E. EXPERIMENTAL RESULTS ON EXTRACTING THE REMAINING ITEMS

In this part we show the extraction of the VB-PTC model in other items, and the experimental results are shown in Figure 8. We can see that items (*volume, issue, doi, keywords, and published time*) matched by regular expressions have a higher F_1, because usually these items have a better context and can be easily located. Most of them are above 90% and can reach 100% even if the page is well-formed. The item (*authors and reference*) F_1 matched based on natural language processing technology is not very high, which may be the result of huge differences in the text content of the page. The matching method based on symbol density (*abstract*) has a high F_1 and F_1 is more than 90% under different data sources, which shows that our mathematical model is accurate.

F. ERROR ANALYSIS

1) ITEM SIMILARITY BETWEEN PAGES

In some pages, there is also the *citation* section. *Reference* is the other papers cited in this paper, and *citation* is the papers cited in this paper. Generally, the format of the two parts is the same, with a high degree of similarity. Using our natural language processing method to extract the *reference* item, we will also regard the *citation* part as the *reference* item. It is worth nothing that the location of the *reference* part is accurate.

2) THERE IS A DIFFERENCE BETWEEN FETCHING DATA AND VERIFYING DATA

The validation data is located in the item content of the page by using the framework of the summary and the combination

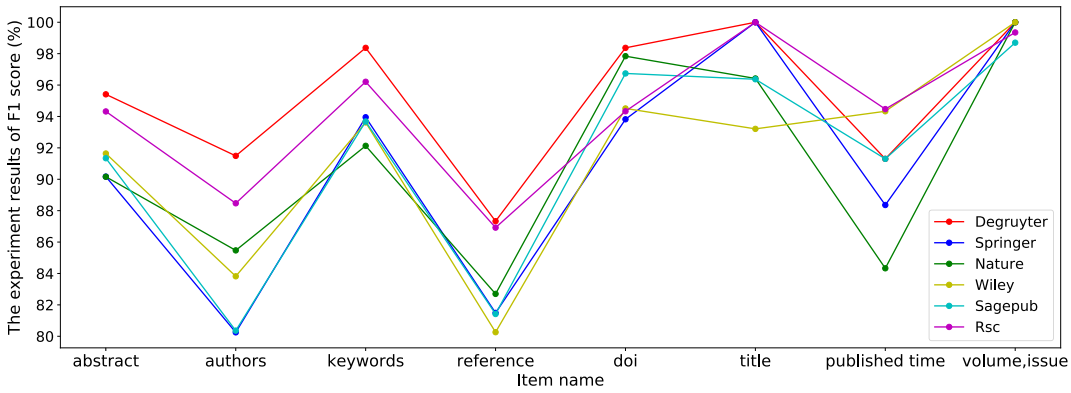

FIGURE 8. Multi-record text extraction experiment results(%).

of the XPath. The text content extracted by our proposed VB-PTC model proposed by us may be different from the validation data in details, for example, the captured data or the data in the validation set will have a space in some places. But the location of the item is accurate.

VII. CONCLUSION AND FUTURE WORK

In this paper, a visual block construction method based on page type conversion VB-PTC is proposed. First, we get the Dom tree structure of the web page, a site-level noise reduction method based on hashtree, and a page-level noise reduction method based on link clusters are used to remove the template data of web pages, and transform the multi-record complex page into the multi-record simple page. Then, use simplified visualization block construction rules are used to generate the visualization blocks corresponding to different word segments. Finally, based on the regular expressions, natural language processing, and symbol density method, the content in the visualization block is detected and matched with the corresponding items. The experiments results show that the VB-PTC algorithm can extract data from unknown and complex multi-record dynamic web pages, and has good generalization ability. At the same time, it can extract the effective information in the efficiently, accurately and without unsupervised from multi-record complex web page.

In a future work, we plan to design more targeted extraction methods for different fields to improve the extraction accuracy of non-standard pages.

ACKNOWLEDGMENT

The authors would like to thank my alma mater Yanshan University and my teachers and classmates for their help in my creation.

REFERENCES

[1] H. Tao, M. Z. A. Bhuiyan, M. A. Rahman, T. Wang, J. Wu, S. Q. Salih, Y. Li, and T. Hayajneh, "TrustData: Trustworthy and secured data collection for event detection in industrial cyber-physical system," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3311–3321, May 2020.

[2] S. Shi, C. Liu, C. Yuan, and Y. Huang, "Multi-feature and dag-based multi-tree matching algorithm for automatic Web data mining," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. Intell. Agent Technol.*, 2014, pp. 118–125.

[3] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, and Q. Yang, "Large-scale hierarchical text classification with recursively regularized deep graph-cnn," in *Proc. World Wide Web Conf., Int. World Wide Web Conf. Steering Committee*, 2018, pp. 1063–1072.

[4] N. Aslam, B. Tahir, H. M. Shafiq, and M. A. Mehmood, "Web-AM: An efficient boilerplate removal algorithm for Web articles," in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, 2019, p. 287.

[5] M. H. Arif, J. Li, M. Iqbal, and H. Peng, "Optimizing XCSR for text classification," in *Proc. IEEE Symp. Service-Oriented Syst. Eng. (SOSE)*, Apr. 2017, pp. 86–95.

[6] H. Peng, J. Li, Q. Gong, Y. Song, Y. Ning, K. Lai, and P. S. Yu, "Fine-grained event categorization with heterogeneous graph convolutional networks," in *Proc. 28th Int. Joint Conf. Artif. Intell.* Palo Alto, CA, USA: AAAI Press, 2019, pp. 3238–3245.

[7] P. Sivakumar, "Effectual Web content mining using noise removal from Web pages," *Wireless Pers. Commun.*, vol. 84, no. 1, pp. 99–121, Sep. 2015.

[8] Z. Zhene, P. Hao, L. Lin, X. Guixi, B. Du, M. Z. A. Bhuiyan, Y. Long, and D. Li, "Deep convolutional mesh RNN for urban traffic passenger flows prediction," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Jun. 2018, pp. 1305–1310.

[9] Y. Liu, H. Peng, J. Guo, T. He, X. Li, Y. Song, and J. Li, "Event detection and evolution based on knowledge base," in *Proc. KBCOM*.

[10] D. Gibson, K. Punera, and A. Tomkins, "The volume and evolution of Web page templates," in *Proc. Special Interest Tracks Posters 14th Int. Conf. World Wide Web*, 2005, pp. 830–839.

[11] Y. Sheng, T. Wu, and X. Wang, "Incorporating term definitions for taxonomic relation identification," in *Proc. 9th Joint Int. Semantic Technol. Conf. (JIST)*. Cham, Switzerland: Springer, 2019, pp. 1–17.

[12] H. Peng, J. Li, S. Wang, L. Wang, Q. Gong, R. Yang, B. Li, P. Yu, and L. He, "Hierarchical taxonomy-aware and attentional graph capsule RCNNs for large-scale multi-label text classification," *IEEE Trans. Knowl. Data Eng.*, early access, Dec. 16, 2020, doi: 10.1109/TKDE.2019.2959991.

[13] F. Viveros-Jiménez, M. A. Sánchez-Pereza, H. Gómez-Adorno, J. P. Posadas-Durán, G. Sidorov, and A. Gelbukh, "Improving the boilerpipe algorithm for boilerplate removal in news articles using HTML tree structure," *Computación y Sistemas*, vol. 22, no. 2, Jul. 2018.

[14] Q. Mao, J. Li, S. Wang, Y. Zhang, H. Peng, M. He, and L. Wang, "Aspect-based sentiment classification with attentive neural turing machines," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 5139–5145, doi: 10.24963/ijcai.2019/714.

[15] Y. Liu, H. Peng, J. Li, Y. Song, and X. Li, "Event detection and evolution in multi-lingual social streams," *Frontiers Comput. Sci.*, vol. 14, no. 5, Oct. 2020.

[16] Y. He, J. Li, Y. Song, M. He, and H. Peng, "Time-evolving text classification with deep neural networks," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2241–2247.

[17] H. Tao, M. Z. A. Bhuiyan, A. N. Abdalla, M. M. Hassan, J. M. Zain, and T. Hayajneh, "Secured data collection with hardware-based ciphers for IoT-based healthcare," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 410–420, Feb. 2019.
[18] M. Z. A. Bhuiyan, J. Wu, G. Wang, Z. Chen, J. Chen, and T. Wang, "Quality-guaranteed event-sensitive data collection and monitoring in vibration sensor networks," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 572–583, Apr. 2017.
[19] J. Bar-Ilan, "Data collection methods on the Web for infometric purposes—A review and analysis," *Entometrics*, vol. 50, no. 1, pp. 7–32, 2001.
[20] M. Kus, "Design of a Web-based educational interface for wireless sensor networks," in *Proc. Int. Symp. Comput. Informat. Math.*, 2013.
[21] R. Selvaraj, T. Marwala, and V. M. Kuthadi, "An efficient Web services framework for secure data collection wireless sensor network," *Brit. J. Political Sci.*, vol. 12, no. 1, pp. 18–31, 2015.
[22] Y. Ling, X. Xie, H. Wang, J. Huang, C. Li, and W. Lai, "A study on wireless sensor network data collection model based on real perception data analysis," in *Proc. Int. Conf. Mechatronics Intell. Robot.*, 2017, pp. 310–315.
[23] K. Yin and C. Zhong, "Data collection in wireless sensor networks," in *Proc. IEEE Int. Conf. Cloud Comput. Intell. Syst.*, Sep. 2011, pp. 98–102.
[24] S. Yu, D. Cai, J.-R. Wen, and W.-Y. Ma, "Improving pseudo-relevance feedback in Web information retrieval using Web page segmentation," in *Proc. 12th Int. Conf. World Wide Web*, 2003, pp. 11–18.
[25] D. Yang and J. Song, "Web content information extraction approach based on removing noise and content-features," in *Proc. Int. Conf. Web Inf. Syst. Mining*, vol. 1, 2010, pp. 246–249.
[26] E. S. Pan, "Boilerplate removal and content extraction from dynamic Web pages," Tech. Rep., 2014.
[27] R. Schäfer, "Accurate and efficient general-purpose boilerplate detection for crawled Web corpora," *Lang. Resour. Eval.*, vol. 51, no. 3, pp. 873–889, Sep. 2017.
[28] R. Uma and B. Latha, "Noise elimination from Web pages for efficacious information retrieval," *Cluster Comput.*, vol. 22, no. S6, pp. 14583–14602, Nov. 2019.
[29] Y. Sheng, Z. Xu, Y. Wang, and G. de Melo, "Murex: Multi-document semantic relation extraction for news analytics," *WWW J.*, vol. 23, no. 3, pp. 2043–2077, 2020.
[30] S. Debnath, P. Mitra, N. Pal, and C. L. Giles, "Automatic identification of informative sections of Web pages," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 9, pp. 1233–1246, Sep. 2005.
[31] G. Poonkuzhali, K. Thiagarajan, K. Sarukesi, and G. V. Uma, "Signed approach for mining Web content outliers," *World Acad. Sci., Eng. Technol., Int. J. Comput., Elect., Automat., Control Inf. Eng.*, vol. 3, no. 8, pp. 2124–2128, 2009.
[32] R. Uma and B. Latha, "Sub-topic modeling-a hierarchy model for topic correlations," *Int. J. Control Theory Appl.*, vol. 9, no. 28, pp. 175–179, 2016.
[33] P. M. Joshi and S. Liu, "Web document text and images extraction using DOM analysis and natural language processing," in *Proc. 9th ACM Symp. Document Eng.*, 2009, pp. 218–221.
[34] E. Cesario, F. Folino, G. Manco, and L. Pontieri, "An incremental clustering scheme for duplicate detection in large databases," in *Proc. 9th Int. Database Eng. Appl. Symp. (IDEAS)*, 2005, pp. 89–95.
[35] N. Aslam, B. Tahir, H. M. Shafiq, and M. A. Mehmood, "Web-AM: An efficient boilerplate removal algorithm for Web articles," in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, Dec. 2019, p. 287.
[36] S. N. Das, P. K. Vijayaraghavan, and M. Mathew, "Eliminating noisy information in Web pages using featured DOM tree," *Int. J. Appl. Inf. Syst.*, 2014.
[37] S. Wu, J. Liu, and J. Fan, "Automatic Web content extraction by combination of learning and grouping," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 1264–1274.
[38] A. K. Oza and S. Mishra, "Elimination of noisy information from Web pages," *Int. J. Recent Technol. & Eng.*, vol. 2, no. 1, pp. 5–11, 2013.
[39] M. Kim, Y. Kim, W. Song, and A. Khil, "Main content extraction from Web documents using text block context," in *Proc. Int. Conf. Database Expert Syst. Appl.*, 2013, pp. 81–93.
[40] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, "Vips: A vision-based page segmentation algorithm," Tech. Rep., 2003.
[41] H. Zhao, W. Meng, Z. Wu, V. V. Raghavan, and C. Yu, "Fully automatic wrapper generation for search engines," in *Proc. 14th Int. Conf. World Wide Web*, no. 14, 2005, pp. 66–75.
[42] K. Simon and G. Lausen, "Viper: Augmenting automatic information extraction with visual perceptions," in *Proc. 14th ACM Int. Conf. Inf. Knowl. Manage.*, no. 14, 2005, pp. 381–388.
[43] W. Liu, X. Meng, and W. Meng, "ViDE: A vision-based approach for deep Web data extraction," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 3, pp. 447–460, Mar. 2010.
[44] F. K. Wai, L. W. Yong, V. L. L. Thing, and V. Pomponiu, "CMDR: Classifying nodes for mining data records with different HTML structures," in *Proc. IEEE Region 10 Conf. (TENCON)*, Nov. 2017, p. 1862.
[45] J. Liu, L. Lin, Z. Cai, J. Wang, and H.-J. Kim, "Deep Web data extraction based on visual information processing," *J. Ambient Intell. Hum. Comput.*, pp. 1–11, Oct. 2017.
[46] T. Gogar, O. Hubacek, and J. Sedivy, "Deep neural networks for Web page information extraction," in *Proc. IFIP Int. Conf. Artif. Intell. Appl. Innov.*, 2016, pp. 154–163.
[47] A. Schulz, J. Lassig, and M. Gaedke, "Practical Web data extraction: Are we there yet?—A short survey," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Oct. 2016, pp. 562–567.
[48] E. Ferrara, P. De Meo, G. Fiumara, and R. Baumgartner, "Web data extraction, applications and techniques: A survey," *Knowl.-Based Syst.*, vol. 70, pp. 301–323, Nov. 2014.
[49] H. A. Sleiman and R. Corchuelo, "A survey on region extractors from Web documents," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 9, pp. 1960–1981, Sep. 2013.
[50] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, Jun. 2004.
[51] C. Kohlschütter, P. Fankhauser, and W. Nejdl, "Boilerplate detection using shallow text features," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, 2010, pp. 441–450.
[52] J. Pomikčlek, "Removing boilerplate and duplicate content from Web corpora," Disertacný Práce, Tech. Rep., 2011.
[53] S. Evert, "A lightweight and efficient tool for cleaning Web pages," Tech. Rep., 2008.
[54] B. Sluban and M. Grčar, "URL tree: Efficient unsupervised content extraction from streams of Web documents," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2013, pp. 2267–2272.
[55] V. Suchomel and J. Pomikálek, "Efficient Web crawling for large text corpora," in *Proc. 7th Web Corpus Workshop*, 2012, pp. 1–5.

JIBING GONG received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, China. He is currently a Professor with the School of Information Science and Engineering, Yanshan University. He is also the Head of the Knowledge Engineering Group (KEG) research team, Yanshan University. His research interests include big data analytics, heterogeneous information networks, machine learning, and data fusion.

HEKAI ZHANG is currently pursuing the master's degree with the Department of Information Science and Engineering, Yanshan University. His research interests include data mining and page automation collection systems.

WEIXIA DU is currently pursuing the master's degree with the Department of Information Science and Engineering, Yanshan University. Her research interests include heterogeneous information networks and recommendation systems.

HUANHUAN LI is currently pursuing the master's degree with the Department of Information Science and Engineering, Yanshan University. He has been involved in research work on NLP and recommendation.

HONGNIAN WEN is currently pursuing the master's degree with the Department of Information Science and Engineering, Shijiazhuang Institute of Railway Technology. Her research interest includes machine learning.