IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 9, 2018, accepted October 18, 2018, date of publication October 23, 2018,
date of current version November 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2877592

Title-Based Extraction of News Contents for Text Mining

ZHEN TAN[1], CHUNHUI HE[1], YANG FANG[1], BIN GE[1,2], AND WEIDONG XIAO[1,2]
[1]Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China
[2]Collaborative Innovation Center of Geospatial Technology, Wuhan 430079, China

Corresponding author: Zhen Tan (tanzhen08a@nudt.edu.cn)

This work was supported by NSFC under Grant 71690233 and Grant 71331008.

ABSTRACT As a vital measure to obtain valuable information and intelligence, web news are flooding all corners of the Internet anytime, anywhere. Traditionally, templates or hand-designed features are utilized to extract the content from web pages, but these models have higher time cost and lower extensibility. Recently, many scholars leverage DOM-tree-based or text-density-based models to extract the contents which have better extensibility and lower time cost, but most of them are hard to extract the content accurately and completely and are easy to introduce the noises. In this paper, we propose a title-based web content extracting model TWCEM to extract the contents of each web page, which leverage the title information to extract the web content. Compared with other extraction model, TWCEM can filter the noises effectively and locate the content positions more accurately. In this experiment, we evaluate the proposed model on real-life websites, and TWCEM achieves state-of-the-art results and outperforms its competitors on both extraction performance and time cost.

INDEX TERMS Title-based extraction model, web news, data mining, information extraction.

I. INTRODUCTION

With the development of the Internet, more than ten millions of news are published, uploaded and shared every day, and the news websites have already become a vital measure for people to obtain concerned information. For example, CNN news is visited about 95 million times every month, and the top 5 most popular news websites and their estimated unique monthly visitors are shown in Table 1.[1]

Many researchers want to collect a large amount of news to analyze the contents and then obtain valuable information and intelligence, for example, the high-cleaned input information will improve the quality of news summarization effectively [1], [2]. But for each news page, the corresponding contents are embedded in a HTML document which includes lots of noises [3], e.g., navigation panels, advertisements, related news links and etc.[2]

Initially, many scholars proposed various methods which leverage templates and hand-designed features to extract the news pages [4], e.g., Crescenzi and Mecca *et al.* [5] and Arocena and Mendelzon *et al.* [6] leverage the hand-designed features to extract web information which will cost a large amount of time to design features and are hard to be applied in other domains. Soderland [7] and Laender *et al.* [8] trained latent web features with hand-labeled documents, however labeling documents requires a lot of manpower, all of which are hard to be applied on large-scale information extraction. RoadRunner [9] and NET [10], leverage template to extract the features which have strong constraints on web page structures.

Recently, DOM-tree-based extraction models [11]–[13] are proposed which leverage the structure of HTML documents to locate the position of correct information. For example, Cai *et al.* [14] introduce the vision page segmentation with DOM-tree to extract the content, and

TABLE 1. The top 5 most popular news websites and their estimated unique monthly visitors.

Popular News Website	Unique Monthly Visitors
Yahoo! News	175 million
Google News	150 million
HuffingtonPost	110 million
CNN	95 million
New York Times	70 million

[1]http://www.ebizmba.com/articles/news-websites

[2]A sample of the news page on CNN is shown in Figure 1, where the contents in red box are correct and the contents in blue box are noises.

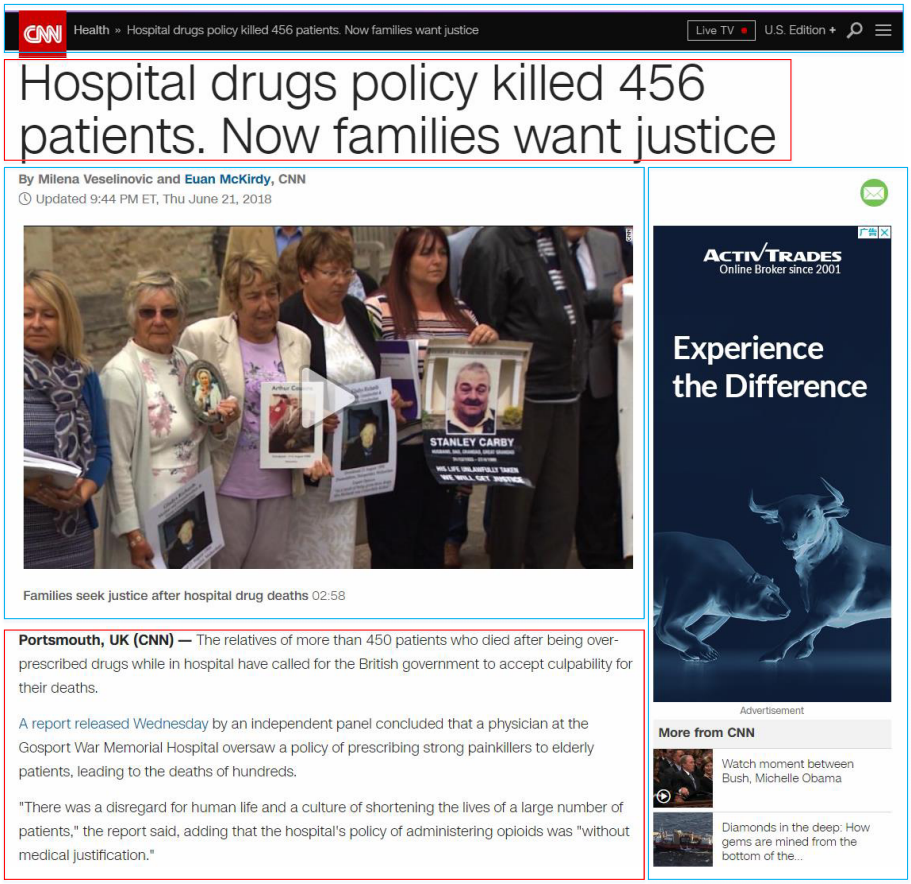


FIGURE 1. A sample of news page on CNN.

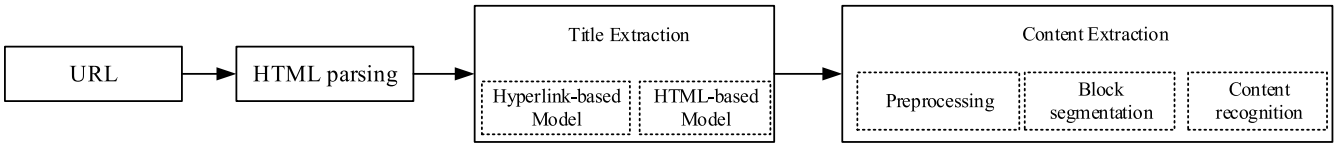


FIGURE 2. The framework of information extraction.

Zheng *et al.* [15] propose an improved method which considers each item in the DOM-tree as a visual block. Compared with template-based models, the DOM-tree-based extraction models achieve competitive results but they need stable web structure which are not suitable for the complex and various web environment. To address the issues, content-based models are proposed which omit the structure of web page and utilize the text features to extract contents. CETR [16] and CEPR [17] utilize the *HTML tag ratio* and *path features* to extract news content, respectively. MCSTD [18] leverages maximum continuous sum of text density methods to extract web contents which can collect the correct content with high-density property. All the content-based extraction models have good performance and high extensibility, but are easy to miss some crucial contents and introduce many noises e.g., navigation panels, advertisements, related news links etc. Hence, the key point is how to use the document features to extract the contents in a comprehensive and accurate way.

In this paper, we propose a title-based web content extracting model, namely **TWCEM**, which leverages the title to locate the position of news contents. A sketch of the model framework is provided in Figure 2. Distinct from existing models where title features are under-represented, **TWCEM** leverages title-based features which can filter the noises in web pages and collect correct contents effectively. Besides, the LCS algorithm is utilized to locate the start and end positions simultaneously, which can decrease the time cost.

1) CONTRIBUTIONS

To summarize, the main contributions of this paper are as follows:

- We propose a novel title-based model **TWCEM** to extract the content of each web page which can filter the noises effectively and locate the content position more accurately.
- Compared with other models, content recognition algorithm is proposed to locate the start and end content positions simultaneously, which can improve the precision of information extraction and decrease the time cost.
- We experimentally evaluated the proposed model on real-life web pages, and **TWCEM** outperforms its competitors on both extraction effectiveness and time cost.

2) ORGANIZATION

We discuss and analyze the related work in Section II and then introduce our model, along with the model analysis in Section III. Afterwards, experimental studies and results analysis are presented in Section IV, followed by conclusion and future work in Section V.

II. RELATED WORK

Web Information Extraction (**WIE**) is first proposed on an automatic content extraction evaluation conference which aims to extract structured and semi-structured data from news pages [19]. In this section, we introduce several related works [4], [20] published in recent years which obtain the state-of-the-art results. According to different extraction mechanisms, we divide these models into two parts: rule-based web information extraction and feature-based web information extraction.

A. RULE-BASED WEB INFORMATION EXTRACTION

1) HANDCRAFTED WEB INFORMATION EXTRACTION

Handcrafted WIE models, e.g., Minerva [5], Web-OQL [6], W4F [21], XWRAP [22] and etc., leverage the hand-designed features to extract web information where the users need to master profound programming skills, have a good knowledge of dataset and output formats, and understand each extraction rule. As a consequence, these models can extract the information precisely in a special domain. But, obviously, these models need high time cost to construct, and due to the poor extensibility they are hard to be applied in other domains.

2) SUPERVISED WEB INFORMATION EXTRACTION

Supervised WIE, e.g., WHISK [7], DEByE [8], SoftMealy [23], SRV [24], PPWIE [25], [26] and etc., utilize the hand-labeled documents to train the latent features, and then these latent features are leveraged to extract news contents from web pages. Compared with handcrafted WIE, supervised WIE systems own higher extensibility and need less programming skills. But it is obvious that these models need large-scale professional annotation and it is hard to apply them on open information extraction.

3) SEMI-SUPERVISED WEB INFORMATION EXTRACTION

IEPAD [27], OLERA [28], Thresher [29] and etc., which are typical semi-supervised WIE systems, leverage hand-labeled training datasets to learn extraction rules. Compared with supervised WIE, semi-supervised WIE systems need lower annotating work which can improve the effectiveness. However, due to the constrain of prior knowledge, these models are not suitable for other application domains.

4) UNSUPERVISED WEB INFORMATION EXTRACTION

Unsupervised WIE systems, e.g., RoadRunner [9], EXALG [30], DeLa [31], DEPTA [32] and NET [10], consider that similar web pages have similar structures. Hence the key point in unsupervised WIE is to detect an original and common template for each website. However, in practice, these models turn out to be limited, because we need to extract contents from more than ten thousands of websites, if there are any changes in any websites, the template will extract wrong contents.

B. FEATURE-BASED WEB INFORMATION EXTRACTION

Recently, apart from rule-based web information extraction models, there are several models which utilize feature-based algorithms to extract the information in each web page.

1) DOM-BASED EXTRACTION MODELS

DOM-based extraction models [11]–[13] leverage the structure of HTML documents to locate the position of correct information. These models can remove the noises, e.g, advertising and hyper-link groups, effectively. Nevertheless, the establishment of DOM-tree has a high requirement for the HTML. Especially, the construction and traversal of the tree have high time complexity, besides the tree traversal method also differs depending on the HTML tags.

2) VISION-BASED EXTRACTION MODELS

The basic idea of vision-based extraction models [14], [15], [33] is that the core information in each HTML is displayed with significant position or style which can be used to extract news content. Cai *et al.* [14], first introduce the vision page segmentation with DOM-tree to extract news pages.

Zheng *et al.* [15] propose an improved method which considers each item in the DOM-tree as a visual block. A series of visual features in each block is composited to evaluate the importance of each item. Wang *et al.* [33] utilize machine learning methods to combine visual features which have good generalization performance, but have high time and memory-space complexities.

3) BLOCKING-BASED EXTRACTION MODELS

CETR [16] utilizes the HTML tag ratio to extract news contents which has good performance and high extensibility. CEPR [17] leverages path features to measure the importance of each node, and then the contents in important nodes are

regarded as correct news content. Besides, CEPR utilizes *Gaussian* smoothing method to weight the tag path edit distance which improves the time cost and reduces content extraction performance. However, CEPR will miss critical information when the block distribution function has more than one sudden changing point. MCSTD [18] utilizes maximum continuous sum of text density methods to extract web content which can collect the correct contents with high-density property.

4) OTHER EXTRACTION MODELS

CETD [34] combines DOM-tree and text density to extract news content which achieve good performance. CLG [35] utilizes the DOM tree to train a machine learning model and then uses a grouping technology to further filter out noisy data. CCB [36] leverages content code vector to judge whether a block is meaningful or not.

Compared with rule-based information extraction models, feature-based models [37] have higher precision and better extensibility which can be applied on on-line information extraction systems effectively. But all the above feature-based models have high time complexity and miss important information when the text distributions are frequently changed. In next section, we propose a novel title-based extraction model which can extract the web information efficiently, and locate the start and end positions accurately.

III. TITLE-BASED WEB NEWS EXTRACTION MODEL

Given a web news set $\mathcal{D}$ in *HTML* format, a document d, $d \in \mathcal{D}$. Our model aims to extract the correct news content c and removes the noises, e.g., navigation panels, advertisements, related news links etc. We propose a title-based Web content extracting model, namely **TWCEM**, which can filter the noises in web pages and collect correct contents effectively. In this section, we first process the HTML parsing and then extract the title of each document. Finally, the title features are leveraged to extract news contents.

A. HTML PARSING

Before processing, we first need to remove the extra tags and identifiers with *Regular Expression* (RE), and the procedure can be seen in Algorithm 1.

Algorithm 1 HTML Parsing

Input: HTML document d.
Output: Candidate text contents c.

$r_1 = RemoveScriptandCSS(d)$;
$r_2 = RemoveAnnotationandSpecialChar(r_1)$;
$c = RemoveTags(r_2)$;

First, we remove the contents between (1) tags $< script >$ and $< /script >$, (2)tags $< style >$ and $< /style >$, which denotes the asynchronous request information and *Cascading Style Sheets* (CSS) in HTML, respectively. In addition, the annotations, blanks and special characters should also be removed in the documents. Finally, all the tags which can be described as $< .*? >$ are removed and text contents are reserved. After preprocessing, we can obtain the pure text content with lots of noises.

B. TITLE EXTRACTION

Title is a summarization of news contents and has strong semantic correlation with the news contents. Hence it is pivotal to extract titles from HTML documents which can be leveraged to extract corresponding news contents. In most cases, a title can be easily captured, because the position of the title in HTML is relatively static, e.g., hyperlink tags, meta tags of DOM-tree and so forth. In our model, we utilize both hyperlink and DOM-tree to extract the correct title.

1) HYPERLINK-BASED TITLE EXTRACTION

For each web news page, under most circumstances, it is easy to obtain the URL and corresponding description information from the home page, and the description information can be regarded as the candidate title. If the description information is missing or consists of abnormal characters, we leverage the information included in the web page to extract the title.

2) HTML-BASED TITLE EXTRACTION

In a HTML document page d, we first convert d into a DOM-tree where non-leaf and leaf nodes denote tags and contents, respectively. Hence, the processing of HTML documents can be achieved through the operation of the DOM-tree. Firstly, we leverage DOM-tree to extract the meta title m_t in tag $< meta >$; secondly, the DOM-tree is utilized to capture candidate titles c_t in head tags $< hi >$[3], where $i = 1, 2, \cdots, 6$; thirdly, we calculate the similarity between meta title and candidate titles with *Edit Distance* (Levenshtein Distance) [38], and the candidate title with the *minimum* edit distance is chosen as the correct title.

Formally, for an extracted meta title $m_t = \{m_1, \cdots, m_i, \cdots, m_{l_1}\}$ and each candidate title $c_t = \{c_1, \cdots, c_j, \cdots, c_{l_2}\}$, where the subscripts i and j denote the i-th and j-th characters in m_t and c_t, respectively, l_1 and l_2 denote the length of m_t and c_t, respectively. The edit distance between m_t and c_t can be described as

$$edit[i][j] = \begin{cases} 0 & i = 0,\ j = 0 \\ j & i = 0,\ j > 0 \\ i & i > 0,\ j = 0 \\ \min(edit[i-1][j] + 1, & \\ edit[i][j-1] + 1, & i > 0,\ j > 0 \\ edit[i-1][j-1] + 1 + flag), & \end{cases} \tag{1}$$

[3] In the international standard of HTML5.1, the tag <*hi*> are leveraged to store the title information, and the corresponding website is https://www.w3schools.com/html/html_headings.asp

where

$$flag = \begin{cases} 0, & m_i = c_j \\ 1, & m_i \neq c_j. \end{cases}$$

The c_j with minimum edit distance is considered as the correct title. If both meta title and candidate title cannot be extracted, we leverage regular expressions to match the Tags [$id^{\wedge} = title$], [$id\$ = title$] and [$class^{\wedge} = title$], the extracted text strings in these tags are regarded as the correct title. The pseudo code of title extraction is shown in Algorithm 2.

Algorithm 2 Title Extraction

Input: Home page H, news page d.
Output: Title t of news page.

$c_t \leftarrow HyperLinkExtraction(H)$;
if $c_t = Null$ **or** $c_t = AbnormalCharString$ **then**
  $DOMTree \leftarrow HTMLtoDOMTree(d)$;
  $c_tList \leftarrow TagsExtraction(h1, h2, \cdots, h6)$;
  if $c_tList = Null$ **or** $c_tList = AbnormalCharString$ **then**
    $c_t \leftarrow IDExtractionFromDOM(DOMTree)$;
    if $c_t = Null$ **or** $c_t = AbnormalCharString$ **then**
      return *Null*
    else
      $t \leftarrow c_t$;
  else
    $m_t \leftarrow MetaTitleExtraction(DOMTree)$;
    $t \leftarrow MinEditDistance(m_t, c_tList)$;
else
  $t \leftarrow c_t$;
return t

C. CONTENT EXTRACTION

After extracting the title, we leverage the results of title extraction to extract news contents. For any available URL with a title, the procedure of content extraction can be divided into three parts. We first obtain the HTML document and remove the noises with HTML processing tools, and then segment the initial content with *segmentation algorithm*, along with *Cblock extraction*. Afterwards, *forward and reverse positioning algorithms* are leveraged to extract the contents.

Definition 1 (Row): In the HTML document, we leverage $\backslash n$ to divide news content, and the content between any two $\backslash n$ denotes a row.

Definition 2 (Ctext): Ctext is the pure text information included in HTML document, and Ctext = $\{Ctext_1, \cdots, Ctext_i, \cdots, Ctext_l\}$, where $Ctext_i$ denotes the i-th row in the Ctext and l denotes the length of Ctext.

Definition 3 (Cblock): For each $Ctext_i$, taking the K-row context around it (forward or reverse), and then through combining $Ctext_i$ and K-line context to obtain i-th Cblock. Formally, we have forward Cblock = $[Ctext_i, Ctext_{i+1}, \cdots, Ctext_{i+K}]$ or reverse Cblock = $[Ctext_i, Ctext_{i-1}, \cdots, Ctext_{i-K}]$.

Definition 4 (Length of Cblock L_C): For each Cblock, the length of Cblock L_C is the number of characters where the invalid characters are removed, e.g., $\backslash t$ and space character.

Definition 5 (Cblock Distribution Function $f_{Cblock}(row)$): For $f_{Cblock}(row)$, x-axis denotes the row number, and y-axis denotes the the length of Cblock.

1) PREPROCESSING

Similar to title extraction, we remove the noises and ineffective information in HTML document, and then obtain the *Ctext*. Then, as input, the *Ctext* is put into the *segmentation algorithm* to extract valid *Cblock*s.

2) BLOCK SEGMENTATION

For the purpose of extracting news contents, we need to segment the *Ctext* and obtain the *Cblocks*. Because the $Ctext_i$ with short contents often consists of navigation tabs, firstly, we filter the $Ctext_i$ with length smaller than δ_1, where δ_1 denotes the minimum effective length of $Ctext_i$. Then, after analyzing a large amount of news contents, we find that the number of punctuations in news content is much larger than the number of punctuations in other blocks, hence we also filter the $Ctext_i$ whose number of punctuations is less than δ_2, where δ_2 denotes the smallest number of punctuations contained in each effective $Ctext_i$. The Block segmentation pseudo is shown in Algorithm 3.

Algorithm 3 Block Segmentation

Input: *Ctext*.
Output: Cblock distribution function $f_{Cblock}(row)$, Cblock content list *CCL*

$f_{Cblock}(row) \leftarrow 0$;
$CCL \leftarrow \emptyset$;
$j \leftarrow 0$;
for $i = 0$; $i < l$; $i++$ **do**
  $L \leftarrow Length(Ctext_i)$;
  if $L < \delta_1$ **then**
    Continue;
  else
    $C \leftarrow NumberofPunctuation(Ctext_i)$;
    if $C \geq \delta_2$ **then**
      $f_{Cblock}(j) = f_{Cblock}(i) + C$;
      $CCL[j].add(Ctext_i)$;
      $j++$;
    else
      Continue;
return $f_{Cblock}(row)$, CCL;

3) CONTENT RECOGNITION

After segmenting *Ctext* and gaining the *Cblock*, we first judge whether the *Cblock* is *NULL* or not. If the *Cblock* is *NULL*, we consider that the news page can not include

Algorithm 4 Content Recognition

Input: *Cblock* distribution function $f_{Cblock}(row)$, *Cblock* content list CCL, title t of news page.
Output: News content N_C

```
if Length(CCL) = 0 then
    N_C = Null;
    return N_C;
row_start ← 0; row_end ← Max(row);
for i = 1; i < L_C/2; i++ do
    L_LCS ← LCS(CCL[i], t);
    if L_LCS ≥ δ3 then
        row_start ← i;
        Break;
    else
        Continue;
for i = L_C − 1; i > L_C/2; i−− do
    L_LCS ← LCS(CCL[i], t);
    if L_LCS ≥ δ3 then
        row_end ← i;
        Break;
    else
        Continue;
for i = row_start; i < row_end; i++ do
    N_C.add(CCL[i]);
return N_C;
```

effective text information. Otherwise, *Longest Common Subsequence* (LCS) algorithm is utilized to find the start and end positions of news contents. In detail, for each *Cblock* $C = \{c_1, \cdots, c_i, \cdots, c_m\}$ and title $t = \{t_1, \cdots, t_j, \cdots, t_n\}$, we calculate the *LCS* between C and t from front to back, and

$$LCS[i][j] = \begin{cases} 0 & i = 0 \ or \ j = 0 \\ LCS[i-1][j-1] + 1 & i, j > 0 \ c_i = t_j \\ \max\{LCS[i][j-1], & \\ LCS[i-1][j]\} & i, j > 0 \ c_i \neq t_j. \end{cases}$$

If the result L_{LCS} in position i is larger than the threshold δ_3 at the first time, where δ_3 denotes the smallest number of same words contained in both title and news content. We consider that position i is the start position of the news contents. Identically, we find the end position of the news content from back to front. The contents between start and end positions are regarded as the news contents. In practice, we locate the start and end positions simultaneously, which can decrease the time cost and enhance the extracting performance.

D. ANALYSIS

Compared with other extracting models, our system has higher accuracy and lower time cost. In detail,

- Compared with text DOM-tree-based models [34] which are hard to find correct information when the structures of web pages are complex and variable. TWCEM can omit the complex analytical procedure and capture the news contents directly.
- Compared with text-density-based models [17], [18] which miss critical information when the *Cblock* distribution function $f_{Cblock}(row)$ has more than one sudden changing points. TWCEM utilizes the title to locate the start and end positions which extract overall efficacious information in web pages.
- Besides, due to the calculation of the correlation between title and contents, TWCEM also can capture the correct information and filter the noises on various situations, as the web page may only include pictures (videos) information, or the web page may include a larger amount of invalid texts when compared with the correct contents.
- TWCEM owns lower time complexity and higher performance compared with other models, furthermore, utilizing the LCS algorithm on the start and end position simultaneously can decrease the time cost effectively.

IV. EXPERIMENTS AND ANALYSIS

In this section, our model, TWCEM, is evaluated and compared with several other models which have been shown to achieve competitive extraction performance.

A. DATA SETS AND PERFORMANCE METRICS

TWCEM is an on-line system which was run on a standard ThinkServer, the detail software and hardware configurations are shown in Table 2. Since no training is required, all the web pages in each corpus are considered as the test data.

TABLE 2. Software and hardware configuration.

Software or hardware	Model or size
CPU	Inter(R) Core(TM) i7 3.0GHz*8
Memory size	128G
Hard disk size	4T
Operation system	CentOS7
Programming language	Java

1) DATA SETS

Three datasets are leveraged to verify the performance of TWCEM, which are the CleanEval dataset from CETR [16], news dataset from news websites and Microblog dataset from blogs and forums. In detail,

- **CleanEval Dataset** [39]. CleanEval includes various components of each web page, e.g., identifying lists, paragraphs and titles which are leveraged to verify the performance of information extraction task. Besides, CleanEval contains 723 English and 714 Chinese news which can verify the robustness of different models.
- **News Dataset**. News dataset contains diverse web news from 8 different English and China news websites: NY Post, Suntimes, BBC, NYTimes, Yahoo, China daily

News, Sina News, and Xinhuanet. Each website includes 50-300 web pages which are chosen randomly.

- **Microblog DataSet**. Notice that information extraction technology for one application domain is difficult to be reused into another different application domain [40]. In order to verify the re-usability of TWCEM algorithms, we selected 600 Weibo web pages which contains short contents for experiment and analysis. The 600 Weibo web pages were from Tencent Weibo, Sina Weibo, and Sohu Weibo, with 200 test pages each.

We show the detailed information of data sets in Table 3.

TABLE 3. Dataset statistics on news sites and microblog sites.

Web Page	#Web sites	#Pages
NY Post	http://www.nypost.com/	300
Suntimes	http://www.suntimes.com/	300
NYTimes	http://www.nytimes.com/	150
BBC	http://www.bbc.com/	300
Yahoo!	http://www.yahoo.com/	300
Sina News	http://www.sina.com.cn/	300
China daily News	http://www.people.com.cn/	300
Xinhuanet	http://www.xinhuanet.com/	300
Tencent Weibo	http://t.qq.com/	200
Sina Weibo	http://weibo.com/	200
Sohu Weibo	http://t.sohu.com/	200

2) PERFORMANCE METRICS

Following CETR [16], we utilize precision (Prec), recall (Rec) and F-measure (F1) to evaluate the performance of news content extraction, which are shown as follows:

$$Precision = \frac{|S_e \cap S_l|}{|S_e|}$$

$$Recall = \frac{|S_e \cap S_l|}{|S_l|}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

where S_e denotes the set of extraction results and S_l denotes the set of hand-labeled results.

3) IMPLEMENTATION

For parameter setting, we selected the parameter δ_1 among $\{\delta_1 | 0 < \delta_1 < 10, \delta_1 \in \mathbf{N}\}$, the parameter δ_2 among $\{\delta_2 | 0 < \delta_2 < 10, \delta_2 \in \mathbf{N}\}$, the parameter δ_3 among $\{\delta_3 | 0 < \delta_3 < 10, \delta_3 \in \mathbf{N}\}$, where $\mathbf{N}$ denotes natural number set. The optimal configurations were $\delta_1 = 4$, $\delta_2 = 1$ and $\delta_3 = 2$ on CleanEval dataset; $\delta_1 = 4$, $\delta_2 = 1$ and $\delta_3 = 2$ on News dataset; and $\delta_1 = 2$, $\delta_2 = 1$ and $\delta_3 = 1$ on Microblog dataset.

B. EVALUATIONS

1) EXPERIMENTS RESULTS

The results are shown in Table 4, Table 5 and Table 6 where the best results are marked in bold and the second ranked results are marked in underlined. From the tables, we can see that

- For F1-score results, on almost all of the dataset, TWCEM achieves state-of-the-art results, especially on Sina News dataset, the value of F1-score is up to 97.59%. Besides, the average results is up to 89.96% which can prove the performance of our model.
- Comparing News datasets with Microblog datasets, the News datasets which include long contents have higher F1-score. The direct reason is that News datasets contain continuous long Cblock which can be easily recognized. Besides, compared with News dataset, Microblog datasets contain limited contents and more noises which makes it hard to locate the position of correct contents.
- On both long and short contents, compared with other models, TWCEM can extract the correct contents more accurately which can prove that leveraging titles to locate the content position can improve the accuracy and robustness effectively. Specifically, on Tencent Weibo, the F1-score is up to 83.06% which achieves an improvement of 37.6% over the top-2 model CETR.
- For recall results, CETR achieves the best results, the direct reason is that CETR leverages *tags ratio* with cluster algorithm to extract contents which collects the contents with lower tags ratio, but it is hard to filter the noises in HTML documents effectively.
- Compared with other baselines, TWCEM also achieves state-of-the-art precision results, on NYTimes and NYPost datasets, the values are up to 99.15% and 98.89%, respectively. The outstanding performance makes it easy to be applied on open information extraction.

2) PARAMETER ANALYSIS

In addition, we also analyze influence of different values of δ_1, δ_2 and δ_3 on different dataset, and the results are shown in Figure 3, from which we can observe that

- In Figure 3(a), on CleanEval and News datasets, when $\delta_1 = 4$, TWCEM achieves state-of-the-art F1-score. On Mircoblog dataset, TWCEM achieves the best F1-score when $\delta_1 = 2$. The direct reason is that Mircoblog dataset includes relatively smaller and more trivial contents than other datasets, hence the filter of δ_1 can remove the effective content easier.
- On all datasets in Figure 3(b), TWCEM obtains outstanding F1-score when $\delta_2 = 1$, which can prove that the attribute of including punctuations or not is a pivotal evaluation index to extract news contents.
- In Figure 3(c), TWCEM obtains the best F1-score when $\delta_3 = 2$ on CleanEval and News datasets and $\delta_3 = 1$ on Mircoblog dataset, That is to say, compared with long news content, short mircoblog websites need less information to locate the content position.

TABLE 4. F1-score results (%).

DataSet	CETD	CLG	MCSTD	CETR	CEPR	TWCEM
CleanEval-En	75.15	75.40	83.62	**88.30**	79.60	86.64
CleanEval-Zh	74.06	73.42	77.08	83.35	78.45	**83.42**
NYPost	73.29	74.36	61.62	58.19	82.73	**87.73**
Suntimes	76.75	83.10	75.73	82.20	87.06	**89.49**
NYTimes	84.76	86.68	82.16	91.14	87.72	**95.63**
BBC	75.36	83.47	80.19	72.77	82.20	**88.41**
Reuters	83.04	83.38	82.58	71.73	86.30	**89.69**
Yahoo!	85.37	85.05	85.80	83.79	87.33	**96.09**
Sina News	86.99	87.64	88.44	86.22	91.59	**97.59**
China daily	80.43	81.16	83.77	38.28	87.76	**91.70**
Xinhuanet	77.65	82.78	87.52	83.32	83.05	**92.54**
Tencent Weibo	44.34	48.65	38.35	79.36	18.49	**82.72**
Sina Weibo	41.14	55.19	46.75	57.98	19.68	**83.06**
Sohu Weibo	60.77	61.04	49.37	87.16	92.65	**94.67**
Average	72.79	75.81	73.07	75.98	76.04	**89.96**

TABLE 5. Precision results (%).

DataSet	CETD	CLG	MCSTD	CETR	CEPR	TWCEM
CleanEval-En	85.23	89.54	86.13	89.42	95.96	**96.13**
CleanEval-Zh	79.21	81.23	85.24	79.60	85.23	**87.25**
NYPost	74.26	76.62	55.34	42.68	98.28	**98.89**
Suntimes	77.85	78.21	76.12	70.82	98.10	**98.26**
NYTimes	89.65	95.52	92.15	88.98	**99.73**	99.15
BBC	74.21	94.35	94.68	60.11	97.83	**98.85**
Reuters	88.51	92.16	95.23	58.48	98.13	**98.19**
Yahoo!	84.42	92.74	92.34	72.67	97.83	**98.65**
Sina News	86.21	91.15	89.69	77.04	**98.57**	98.54
China daily	79.54	85.13	85.47	24.40	95.11	**98.85**
Xinhuanet	75.21	84.29	91.18	72.10	94.72	**97.54**
Tencent Weibo	54.42	55.63	75.67	66.37	94.25	**96.23**
Sina Weibo	38.21	54.56	70.54	41.81	86.84	**92.25**
Sohu Weibo	77.64	85.37	94.22	81.00	96.00	**97.19**
Average	76.04	82.61	84.57	66.11	95.47	**96.86**

TABLE 6. Recall results (%).

DataSet	CETD	CLG	MCSTD	CETR	CEPR	TWCEM
CleanEval-En	67.21	65.12	81.26	**87.20**	68.00	78.85
CleanEval-Zh	69.53	66.98	70.35	**87.48**	72.67	79.91
NYPost	72.34	72.23	69.51	**91.40**	71.43	78.84
Suntimes	75.69	88.64	75.34	**97.95**	78.25	82.15
NYTimes	80.37	79.33	74.12	**93.40**	78.29	92.35
BBC	76.54	74.84	69.55	**92.17**	70.88	79.96
Reuters	78.21	76.13	72.89	**92.75**	77.01	82.54
Yahoo!	86.34	78.54	80.12	**98.94**	78.86	93.66
Sina News	87.79	84.39	87.23	**97.89**	85.54	96.66
China daily	81.33	77.54	82.13	**88.76**	81.46	85.52
Xinhuanet	80.25	81.32	84.15	**98.68**	73.94	88.02
Tencent Weibo	37.41	43.23	25.68	**98.66**	10.25	72.54
Sina Weibo	44.56	55.84	34.96	**94.56**	11.10	75.54
Sohu Weibo	49.92	47.50	33.45	**94.33**	89.53	92.27
Average	70.54	70.83	67.20	**93.87**	67.66	84.20

C. COMPLEXITY ANALYSIS

To compare the time and memory-space complexity of different models, we illustrate the average times of information extraction on each dataset in Table 7. From Table 7, we observe that

- Compared with other baselines, TWCEM model has obvious advantage on time costs. Specifically, the average time of our model is 0.34 which is only one third of the time cost on model CEPR, which can make our model to be applied on large-scale information extraction more easily.
- Compared with news dataset, on all models, the time cost on Microblog dataset is much lower, and the average time is only 0.19 on Sohu Weibo. The direct reason

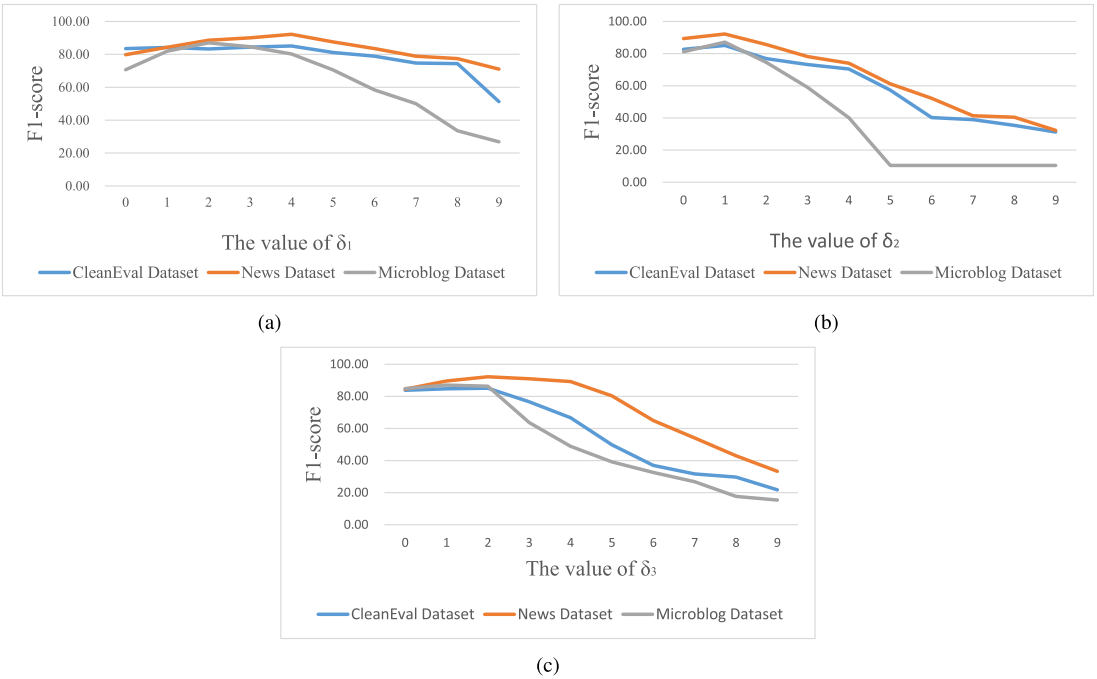


FIGURE 3. Hyperparameter analysis. (a) F1-score on different δ_1. (b) F1-score on different δ_2. (c) F1-score on different δ_3.

TABLE 7. The average times of information extraction (s).

DataSet	CETR	CEPR	TWCEM
CleanEval-En	0.99	1.02	**0.35**
CleanEval-Zh	0.93	0.95	**0.39**
NYPost	0.92	0.94	**0.33**
Suntimes	1.17	1.18	**0.41**
NYTimes	1.23	1.25	**0.42**
BBC	1.28	1.33	**0.44**
Reuters	1.01	1.03	**0.33**
Yahoo!	0.98	0.97	**0.32**
Sina News	1.21	1.25	**0.40**
China daily	0.95	0.99	**0.29**
Xinhuanet	0.99	1.05	**0.36**
Tencent Weibo	0.68	0.71	**0.25**
Sina Weibo	0.77	0.78	**0.22**
Sohu Weibo	0.67	0.68	**0.19**
Average	0.98	1.01	**0.34**

is that the News in Microblog only contain short-text contents with lower number of *Cblock*s, hence these models only need to spend lower time to segment the contents.

V. CONCLUSION

In this paper, we propose a novel title-based information extraction model **TWCEM** to extract the contents of each web page which can filter the noises effectively and locate the content positions more accurately. We verify the proposed model on various real-life web pages, and **TWCEM** achieves state-of-the-art results on both extraction effectiveness and time cost.

We will explore the following future works:

- Besides the title information, the news content also can be leveraged to calculate the correlation between different *Cblock*s which can improve the precision of location.
- For the websites with different languages, we will set adaptive preprocessing measure to extract the news pages which can also improve the performance of the extraction.

REFERENCES

[1] O. Shapira, H. Ronen, M. Adler, Y. Amsterdamer, J. Bar-Ilan, and I. Dagan, "Interactive abstractive summarization for event news tweets," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Copenhagen, Denmark, Sep. 2017, pp. 109–114. [Online]. Available: https://aclanthology.info/papers/D17-2019/d17-2019

[2] R. A. Chongtay, M. Last, and B. Berendt, "Responsive news summarization for ubiquitous consumption on multiple mobile devices," in *Proc. 23rd Int. Conf. Intell. User Interfaces (IUI)*, Tokyo, Japan, Mar. 2018, pp. 433–437, doi: 10.1145/3172944.3172992.

[3] D. Gibson, K. Punera, and A. Tomkins, "The volume and evolution of Web page templates," in *Proc. 14th Int. Conf. World Wide Web (WWW)*, Chiba, Japan, May 2005, pp. 830–839, doi: 10.1145/1062745.1062763.

[4] M. I. Varlamov and D. Y. Turdakov, "A survey of methods for the extraction of information from Web resources," *Program. Comput. Softw.*, vol. 42, no. 5, pp. 279–291, 2016, doi: 10.1134/S0361768816050078.

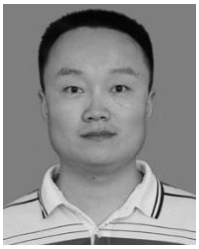
[5] V. Crescenzi and G. Mecca, "Grammars have exceptions," *Inf. Syst.*, vol. 23, no. 8, pp. 539–565, 1998, doi: 10.1016/S0306-4379(98)00028-3.

[6] G. O. Arocena and A. O. Mendelzon, "WebOQL: Restructuring documents, databases, and webs," *Theory Pract. Object Syst.*, vol. 5, no. 3, pp. 127–141, 1999.

[7] S. Soderland, "Learning information extraction rules for semi-structured and free text," *Mach. Learn.*, vol. 34, nos. 1–3, pp. 233–272, 1999, doi: 10.1023/A:1007562322031.

[8] A. H. F. Laender, B. A. Ribeiro-Neto, and A. S. da Silva, "DEByE—Data extraction by example," *Data Knowl. Eng.*, vol. 40, no. 2, pp. 121–154, 2002, doi: 10.1016/S0169-023X(01)00047-7.

[9] V. Crescenzi, G. Mecca, and P. Merialdo, "RoadRunner: Towards automatic data extraction from large Web sites," in *Proc. 27th Int. Conf. Very Large Data Bases (VLDB)*, Roma, Italy, Sep. 2001, pp. 109–118. [Online]. Available: http://www.vldb.org/conf/2001/P109.pdf
[10] B. Liu and Y. Zhai, "NET—A system for extracting Web data from flat and nested data records," in *Proc. 6th Int. Conf. Web Inf. Syst. Eng. (WISE)*, New York, NY, USA, Nov. 2005, pp. 487–495, doi: 10.1007/11581062_39.
[11] W. Li, Y. Dong, R. Wang, and H. Tian, "Information extraction from semi-structured Web page based on DOM tree and its application in scientific literature statistical analysis system," in *Proc. IITA Int. Conf. Services Sci., Manage. Eng. (SSME)*, Zhangjiajie, China, Jul. 2009, pp. 124–127, doi: 10.1109/SSME.2009.59.
[12] W. G. Siqueira and L. A. Baldochi, "Leveraging analysis of user behavior from Web usage extraction over DOM-tree structure," in *Proc. 18th Int. Conf. Web Eng. (ICWE)*, Caceres, Spain, Jun. 2018, pp. 185–192, doi: 10.1007/978-3-319-91662-0_14.
[13] L. Shi, C. Niu, M. Zhou, and J. Gao, "A DOM tree alignment model for mining parallel data from the Web," in *Proc. 21st Int. Conf. Comput. Linguistics 44th Annu. Meeting Assoc. Comput. Linguistics*, Sydney, NSW, Australia, Jul. 2006, pp. 489–496. [Online]. Available: http://aclweb.org/anthology/P06-1062
[14] D. Cai, X. He, J.-R. Wen, and W.-Y. Ma, "Block-level link analysis," in *Proc. 27th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Sheffield, U.K., Jul. 2004, pp. 440–447, doi: 10.1145/1008992.1009068.
[15] S. Zheng, R. Song, and J.-R. Wen, "Template-independent news extraction based on visual consistency," in *Proc. 22nd AAAI Conf. Artif. Intell.*, Vancouver, BC, Canada, 2007, pp. 1507–1511. [Online]. Available: http://www.aaai.org/Library/AAAI/2007/aaai07-239.php
[16] T. Weninger, W. H. Hsu, and J. Han, "CETR: Content extraction via tag ratios," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, Raleigh, NC, USA, Apr. 2010, pp. 971–980, doi: 10.1145/1772690.1772789.
[17] G. Wu, L. Li, X. Hu, and X. Wu, "Web news extraction via path ratios," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, San Francisco, CA, USA, Oct./Nov. 2013, pp. 2059–2068, doi: 10.1145/2505515.2505558.
[18] K. Sun *et al.*, "Web content extraction based on maximum continuous sum of text density," in *Proc. Int. Conf. Asian Lang. Process. (IALP)*, Tainan, Taiwan, Nov. 2016, pp. 288–292, doi: 10.1109/IALP.2016.7875988.
[19] G. R. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel, "The automatic content extraction (ACE) program tasks, data, and evaluation," in *Proc. 4th Int. Conf. Lang. Resour. Eval. (LREC)*, Lisbon, Portugal, May 2004, pp. 1–4. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2004/pdf/5.pdf
[20] C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan, "A survey of Web information extraction systems," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1411–1428, Oct. 2006, doi: 10.1109/TKDE.2006.152.
[21] A. Sahuguet and F. Azavant, "Building intelligent Web applications using lightweight wrappers," *Data Knowl. Eng.*, vol. 36, no. 3, pp. 283–316, 2001, doi: 10.1016/S0169-023X(00)00051-3.
[22] L. Liu, C. Pu, and W. Han, "XWRAP: An XML-enabled wrapper construction system for Web information sources," in *Proc. 16th Int. Conf. Data Eng.*, San Diego, CA, USA, Feb./Mar. 2000, pp. 611–621, doi: 10.1109/ICDE.2000.839475.
[23] C.-N. Hsu and M.-T. Dung, "Generating finite-state transducers for semi-structured data extraction from the Web," *Inf. Syst.*, vol. 23, no. 8, pp. 521–538, 1998, doi: 10.1016/S0306-4379(98)00027-1.
[24] D. Freitag, "Information extraction from HTML: Application of a general machine learning approach," in *Proc. 15th Nat. Conf. Artif. Intell. 10th Innov. Appl. Artif. Intell. Conf. (AAAI)*, Madison, WI, USA, Jul. 1998, pp. 517–523. [Online]. Available: http://www.aaai.org/Library/AAAI/1998/aaai98-073.php
[25] G. Wu and X. Wu, "Extracting Web news using tag path patterns," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Macau, China, Dec. 2012, pp. 588–595, doi: 10.1109/WI-IAT.2012.107.
[26] X. Wu, F. Xie, G. Wu, and W. Ding, "Personalized news filtering and summarization on the Web," in *Proc. IEEE 23rd Int. Conf. Tools Artif. Intell. (ICTAI)*, Boca Raton, FL, USA, Nov. 2011, pp. 414–421, doi: 10.1109/ICTAI.2011.68.
[27] C.-H. Chang and S.-C. Lui, "IEPAD: Information extraction based on pattern discovery," in *Proc. 10th Int. World Wide Web Conf. (WWW)*, Hong Kong, May 2001, pp. 681–688, doi: 10.1145/371920.372182.
[28] C.-H. Chang and S.-C. Kuo, "OLERA: Semisupervised Web-data extraction with visual support," *IEEE Intell. Syst.*, vol. 19, no. 6, pp. 56–64, Nov. 2004, doi: 10.1109/MIS.2004.71.
[29] A. W. Hogue and D. R. Karger, "Thresher: Automating the unwrapping of semantic content from the world wide Web," in *Proc. 14th Int. Conf. World Wide Web (WWW)*, Chiba, Japan, May 2005, pp. 86–95, doi: 10.1145/1060745.1060762.
[30] A. Arasu and H. Garcia-Molina, "Extracting structured data from Web pages," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, San Diego, CA, USA, Jun. 2003, pp. 337–348, doi: 10.1145/872757.872799.
[31] J. Wang and F. H. Lochovsky, "Data extraction and label assignment for Web databases," in *Proc. 12th Int. World Wide Web Conf. (WWW)*, Budapest, Hungary, May 2003, pp. 187–196, doi: 10.1145/775152.775179.
[32] Y. Zhai and B. Liu, "Structured data extraction from the Web based on partial tree alignment," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 12, pp. 1614–1628, Dec. 2006, doi: 10.1109/TKDE.2006.197.
[33] J. Wang *et al.*, "Can we learn a template-independent wrapper for news article extraction from a single training site?" in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Paris, France, Jun./Jul. 2009, pp. 1345–1354, doi: 10.1145/1557019.1557163.
[34] F. Sun, D. Song, and L. Liao, "DOM based content extraction via text density," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, Beijing, China, Jul. 2011, pp. 245–254, doi: 10.1145/2009916.2009952.
[35] 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 (WWW)*, Florence, Italy, May 2015, pp. 1264–1274, doi: 10.1145/2736277.2741659.
[36] T. Gottron, "Content code blurring: A new approach to content extraction," in *Proc. 19th Int. Workshop Database Expert Syst. Appl. (DEXA)*, Turin, Italy, Sep. 2008, pp. 29–33, doi: 10.1109/DEXA.2008.43.
[37] X. Zhao, C. Xiao, X. Lin, W. Zhang, and Y. Wang, "Efficient structure similarity searches: A partition-based approach," *VLDB J.-Int. J. Very Large Data Bases*, vol. 27, no. 1, pp. 53–78, 2018, doi: 10.1007/s00778-017-0487-0.
[38] M. Neuhaus and H. Bunke, *Bridging the Gap between Graph Edit Distance and Kernel Machines* (Series in Machine Perception and Artificial Intelligence), vol. 68. Singapore: World Scientific, 2007, doi: 10.1142/6523.
[39] M. Baroni, F. Chantree, A. Kilgarriff, and S. Sharoff, "Cleaneval: A competition for cleaning Web pages," in *Proc. Int. Conf. Lang. Resour. Eval. (LREC)*, Marrakech, Morocco, May/Jun. 2008, pp. 1–7. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2008/summaries/162.html
[40] E. Ferrara, P. D. Meo, G. Fiumara, and R. Baumgartner, "Web data extraction, applications and techniques: A survey," *Knowl.-Based Syst.*, vol. 70, pp. 301–323, Nov. 2014, doi: 10.1016/j.knosys.2014.07.007.



ZHEN TAN was born in Hami, China, in 1991. He received the master's degree from the National University of Defense Technology, China, in 2014, where he is currently pursuing the Ph.D. degree. His research mainly focuses on knowledge representation and information extraction.

CHUNHUI HE was born in Yongzhou, China, in 1991. He received the master's degree from Xiangtan University, China, in 2017. He is currently with the National University of Defense Technology. His research interests include information extraction and data mining.

YANG FANG was born in Cixi, China, in 1993. He received the bachelor's degree from the National University of Defense Technology, China, in 2016, where he is currently pursuing the master's degree. His research mainly focuses on RDF graphs and knowledge representation.

BIN GE was born in Qingdao, China, in 1979. He received the Ph.D. degree from the National University of Defense Technology, China. He is currently a Vice Professor, and also holds a conjunction research position at the Collaborative Innovation Center for Geospatial Information Technology, China. His research interests include knowledge graph and information extraction.

WEIDONG XIAO was born in Harbin, China, in 1968. He received the Ph.D. degree from the National University of Defense Technology (NUDT), China. He is currently a Full Professor and the Head of the Department of Information System Engineering, NUDT, and also holds a conjunction research position at the Collaborative Innovation Center for Geospatial Information Technology, China. His research interests include big data analytics and social computing.