Extracting Web Content by Exploiting Multi-Category Characteristics

Qian Wang[1], Qing Yang[2], Jingwei Zhang[1](✉), Rui Zhou[3], and Yanchun Zhang[4]

[1] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China
1369815448@qq.com, gtzjw@hotmail.com

[2] Guangxi Key Laboratory of Automatic Measurement Technology and Instrument, Guilin University of Electronic Technology, Guilin 541004, China
gtyqing@hotmail.com

[3] Faculty of Science, Engineering and Technology, Swinburne University of Technology, Melbourne, Australia
rzhou@swin.edu.au

[4] Centre for Applied Informatics, Victoria University, Melbourne, Australia
yanchun.zhang@vu.edu.au

Abstract. Extracting web content aims at separating web content from web pages since web content is organized and presented by different HTML templates and is surrounded by various information. Knowing little about template structures and noise information before extraction, the variability of page templates, etc., make the extraction process very challenging to guarantee extraction precision and extraction adaptability. This study proposes an effective web content extraction method for various web environments. To ensure extraction performance, we exploited three kinds of characteristics, visual text information, content semantics(instead of HTML tag semantics) and web page structures. These characteristics are then integrated into an extraction framework for extraction decisions for different websites. Comparative experiments on multiple web sites with two popular extraction methods, CETR and CETD, show that our proposed extraction method outperforms CETR on precision when keeping the same advantage on recall, and also gains 4% improvement over CETD on the average F1-score; especially, our method can provide better extraction performance when facing short content than CETD, and presents a better extraction adaptability.

Keywords: Content extraction · Visual characteristics · Content semantics

1 Introduction

A vast number of websites and web pages produce large-scale and popular web content, which are making great contributions for data-driven applications and novel business modes. Before further exploitation of these text content presented in web pages, an extraction process should be started to conduct data preprocessing for constructing analysis applications on massive web content. The extraction process aims at discovering the concerned visual text information in web pages, removing the surrounding noise and then separating them from web pages. However, websites use various HTML tags to organize and to present their content in a variety of formats, which constitutes the major obstacle for web content extraction. In addition, some extra information along with free web services, such as advertisements, recommended links, etc., increase the difficulty of identifying relevant text accurately and automatically. It is very valuable to establish effective methods and to provide clean web content for both existing and future big data applications.

Most of the existing extraction methods adopt an extraction mode of learning rules first and then extracting, a popular branch is to use supervised or semi-supervised methods to learn the extraction rules on web page structures, such as DOM(Document Object Model) tree [1] [2] or XPath [3], and then to use the output rules to find the extracted objectives. However, the myriad web page templates adopted by different web sites, especially rich formats contributed by flexible and creative users, may cause the learned extraction rules failed. The adaptability is necessary for the contemporary extraction methods except for extraction precision. Both the diversity and the variability of web page structures brought by presentation requirements make a bigger extraction challenge. Compared with extraction mode of rule learning first, the new extraction mode should make an instant extraction decision automatically on the characteristics of the current web pages to address the above challenges.

In this work, our goal for web content extraction is to provide an effective and adaptive extraction decision without explicit extraction rules, which can work well on a wide variety of web sites covering plentiful web page templates and noise. First, we define a novel visual text content characteristic to find some well-marked text nodes as starting points to enlighten the following extraction. And then a path aggregation computation is introduced to locate those text areas, namely a block of continuous text in web pages, this process maybe includes some noise for web content extraction. Finally, content semantics are exploited to remove the above noise and to improve extraction performance, which is very different from the semantics of HTML tags. All the above characteristics are then integrated into an extraction framework for instant extraction decisions.

In summary, this study makes the following contributions:

– We design a framework based on multi-characteristics of web pages to extract web content, which allows to make instant extraction decisions on web pages' own characteristics without any explicit extraction rules and provides extraction adaptability.

- We define two novel characteristics for extraction, namely visual text information and content semantics, especially, content semantics is very different from HTML tag semantics. The two kinds of characteristics are integrated with the DOM model of web pages to support accurate extraction decisions in different web sites.
- We demonstrate the effectiveness of our method by conducting comprehensive and comparative experiments on multiple web sites, which cover different existential situations of web content.

The rest of this paper is organized as follows. In Sect. 2, we review the related work. We present the detailed problem definition for web content extraction in Sect. 3. Section 4 describes the extraction framework and elaborates the characteristics for extracting web content. In Sect. 5, we investigate the effectiveness of our method. Finally, we conclude our work in Sect. 6.

2 Related Work

Web content usually refers to the visual formal text in web pages, such as news, blogs, etc., which do not include advertisements, links and other noise. Web content extraction was firstly put forwards by Rahman et al. [4] in 2001, which has been playing an important role on data collecting and cleaning for big data applications. From the viewpoints of extraction technologies, there are three primary kinds of extraction strategies, rule-based extraction, vision-based extraction and semantics-based extraction.

Rule-based extraction, also named wrapper-induction extraction, often depends on the web page structure contributed by HTML tags, which considers a web page as a DOM tree or a character stream. Valter et al. [5] proposed Roadrunner, which can establish wrappers described by regular expressions automatically on both the similarities and differences of a group of web pages. Furche et al. [6] studied the problems of both extraction robustness and noise resistant, but they need to annotate web pages manually and is not a completely automatic method. Reis et al. [7] put forwards a domain-oriented approach to extract news, whose core is to compute tree edit distance on the DOM model of web pages. Wu et al. [8] defined tag path edit distance and tag path ratios to extract news from web pages. Furthermore, they developed the relevancy between web content layouts and tag paths, and then made tag path feature fusion to extract web content [9]. Rule-based methods need to create different wrappers for different sites, once the web page templates changed, the learned rules will fail. In addition, the pure structure character is not enough to deal with today's complex web environments well.

Vision-based extraction makes full use of web page layouts and other visual cues to improve extraction performance, which also often collaborates with web page structures, such as DOM tree. Cai et al. [10] proposed an extraction algorithm named VIsion-based Page Segmentation(VIPS), which utilizes both the DOM tree and visual layout cues to segment documents and then to find those segmentations containing the required content. Based on the VIPS, Song et al. [11] constructed feature vector for block importance, including spatial features(such as position, size) and content features(such as the number of pictures and links), and considered the blocks with high rankings as extraction objectives. Fernandes et al. [12] also adopted the strategy of ranking the blocks of web pages by computing their weights to extract content. Sun et al. [13] demonstrated the efficiency of text density for content extraction, which introduces text density both under a single node and in a whole web page to work with DOM tree for more accurate extraction. Qureshi et al. [14] developed more statistical and formatting characteristics of web pages on DOM trees, such as text information and links density, font size, style and location, to establish a hybrid extraction model. The vision-based extraction technologies need to make careful observations of visual cues, especially, different formatting preferences often generate different visual features, whose applications still have some constraints.

Semantics-based extraction utilizes the known functions of HTML tags, namely the semantics of tags, to improve extraction performance. Peters et al. [15] integrated the semantics of several specific HTML tags for content extraction, such as $< class >$, $< div >$, $< h1 >$, etc. Both the flexible nested usage of HTML tags and the growing number of HTML tags, such as HTML5 tags, only permit us to exploit the semantics of a part of HTML tags.

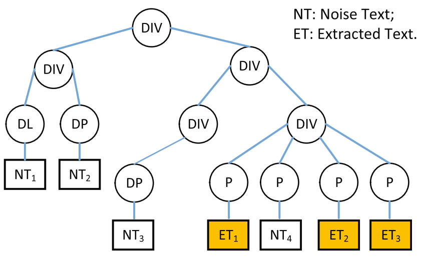
Some further characteristics are also exploited for content extraction. Ortona et al. [16] proposed WADaR, which adopts strategies of joint wrapper and data repair to provide better extraction performance. Weninger et al. [17] proposed Content Extraction via Tag Ratios(CETR), which considers HTML tag ratios as an important feature and designs a clustering technique based on 2-dimension tag ratio to distinguish content and non-content areas in web pages. Uzun et al. [18] introduced a feedback mechanism between the decision tree learning for obtaining the extraction rules and the extraction precision on the specific rules, which aims at getting a tradeoff between extraction efficiency and effectiveness.

Web environments have been in evolution, it is still a challenging work to develop an effective and strongly adaptable content extraction framework. We should exploit further characteristics to respond to the extraction challenges caused by diversified presentation requirements and flexible user formatting, and to design novel methods for accurate and adaptive extraction. Weninger et al. [19] also conduct in-depth discussions on the current web content extraction technologies in 2015, who believe that the evolution of web site formats and practices is a big challenge for content extraction, both rule-based extraction methods and extractors based on heuristic features should be re-examined, new extraction methods should be designed to address future extraction problems.

3 Problem Description

For collecting text content, we focus on the topic-based web pages in this work, namely those pages holding news, blogs, etc., whose core contents are a few paragraphs of text describing a complete story. Usually, a web page can be considered as a character stream or a DOM tree for further processing. To make full use of the structural characteristic of web pages implied by HTML tags, we define a web page on the viewpoint of DOM tree. The DOM tree of a web page is composed of two types of nodes, branch nodes showing web page organization and leaf nodes holding web page information. The values of branch nodes are just HTML tags, the values of leaf nodes usually text or some other visual information, such as links, advertisements, etc.

In a DOM tree, a tag path is a series of ordered HTML tags, which starts from the root node to one leaf node in DOM tree. An effective text node means that the text node located by a tag path holds the required content that should be extracted. A web page can be defined by a set of pairs, $< path, content >$, where *path* corresponds to a tag path, and *content* is just the information held by the leaf node located by *path*. Figure 1a and b illustrate an excerpt of DOM tree and the corresponding definition for the web page.



$< //DIV/DIV/DL, NT_1 >$

$< //DIV/DIV/DP, NT_2 >$

$< // DIV/DIV[2]/DIV/DP, NT_3 >$

$< //DIV/DIV[2]/DIV[2]/P[1], ET_1 >$

$< //DIV/DIV[2]/DIV[2]/P[2], NT_4 >$

$< //DIV/DIV[2]/DIV[2]/P[3], ET_2 >$

$< //DIV/DIV[2]/DIV[2]/P[4], ET_3 >$

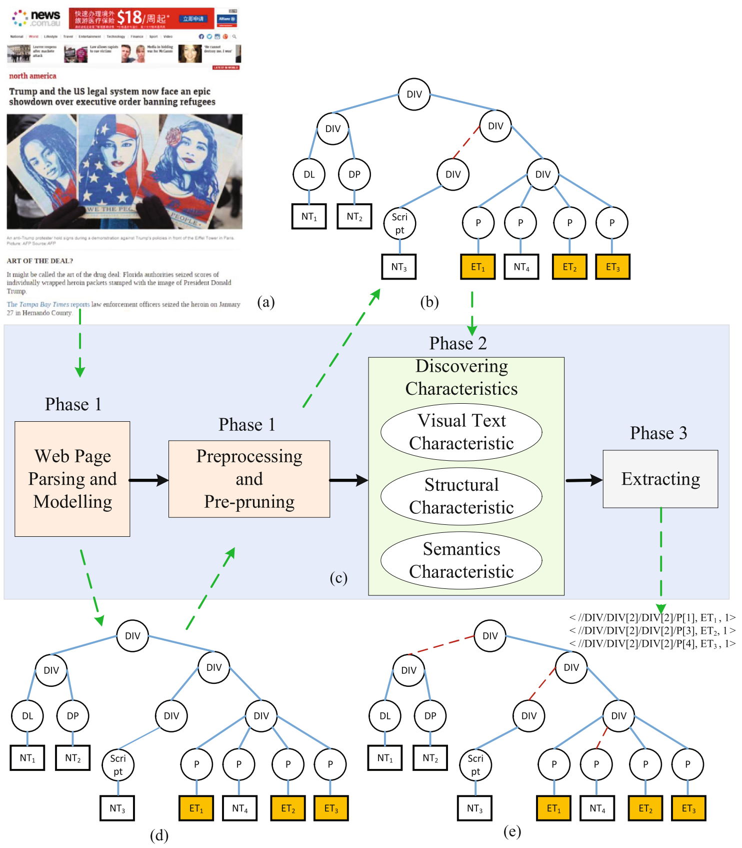
(a) An Excerpt of DOM Tree (b) a formal definition of web page

Fig. 1. An excerpt of DOM tree and the definition of web page

An extraction operation is to make an extraction decision for each pair, a boolean value will be attached for each pair. For example, if the *content* located by *path* is a required content, the pair will be transformed into $< path, content, TRUE >$, otherwise $< path, content, FALSE >$. Given a group of web pages, which can be transformed into a series of pairs $\{p_1, p_2, \cdots, p_n\}$, the content extraction process aims at expanding each pair $p_i =< path_i, content_i >$ into a triple $ep_i =< path_i, content_i, decision_i >$ and then outputs those triples with $decision = 1$, namely $\{ep_i | ep_i =< path_i, content_i, decision_i > \wedge decision_i = 1\}$.

4 Extraction Framework for Web Content

In this section, we will elaborate a three-phase extraction framework, which is comprised of modeling and preprocessing web pages, discovering extraction characteristics and integrating the above characteristics to make extraction decisions. Before extraction, since HTML tags contribute greatly for web page structure except for making information visual, all web pages will be modelled into DOM trees for easy utilization of structural characteristics, and some redundant parts are also pre-pruned for simplifying later computation. Three kinds of characteristic, including visual punctuation mark characteristics, Web page structure characteristic and content semantics similarity, are defined and refined, which are then applied for the following extraction decisions.



(a)
(b)
(c)
(d)
(e)
Phase 1
Web Page Parsing and Modelling
Phase 1
Preprocessing and Pre-pruning
Phase 2
Discovering Characteristics
Visual Text Characteristic
Structural Characteristic
Semantics Characteristic
Phase 3
Extracting
DIV
DL
DP
Script
P
NT1
NT2
NT3
NT4
ET1
ET2
ET3
< //DIV/DIV[2]/DIV[2]/P[1], ET1, 1>
< //DIV/DIV[2]/DIV[2]/P[3], ET2, 1>
< //DIV/DIV[2]/DIV[2]/P[4], ET3, 1>

Fig. 2. Content extraction framework and its input/output

Figure 2 presents the proposed content extraction framework and the input/output in each phase, in which the green dashed arrows represent input or output, the red dashed lines represent that the paths do not locate the required content and should be removed, all leaf nodes with orange filling in DOM trees represent the required content. Figure 2(c) presents the three phases of the extraction process. When a web page shown in Fig. 2(a) is input into the extraction framework, the required content will be output in the third phase, Fig. 2(e) shows the set of required triples and illustrates the final DOM tree, in which all paths marked with red dashed lines are removed, those remaining paths can be used to extract the required content. Figure 2(b) and (d) demonstrate the internal input/output and the corresponding changes of web page.

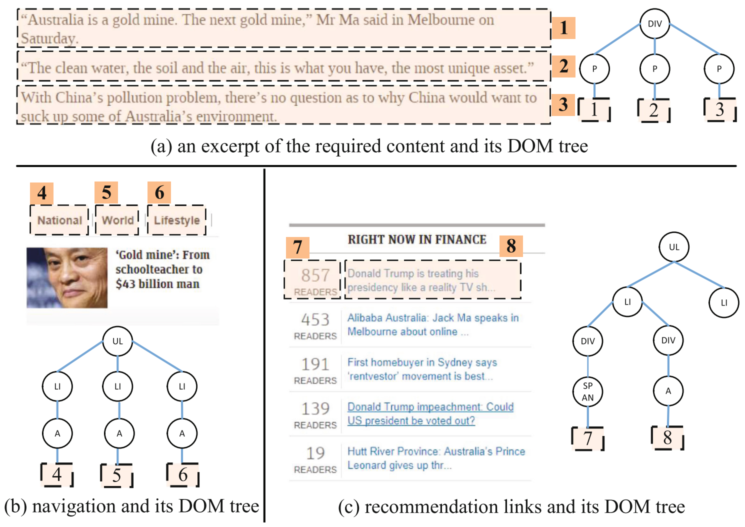
4.1 Constructing DOM Tree and Web Page Preprocessing

Except for presentation functions, HTML tags also provide valuable structural information for web pages, which is very helpful to improve extraction performance. The embedded structure of HTML tags in web pages can be interpreted into a tree structure, which is named DOM model. We use Jsoup parser [20] to interpret web pages into DOM trees. DOM tree is the logical model of web pages, which allows web pages to be processed easily in memory. DOM model transforms each pair of HTML tags into a subtree, such as $< DIV >$ and $< /DIV >$. A traversal operation can be exerted on DOM tree to access all branch and leaf nodes, and then output $< path, content >$ pairs. In addition, we also use regular expressions to remove those unrelated tags and parts from web pages, such as subtrees covered by $< script >$ and $< /script >$, which are not relevant to the required content. The pre-pruning will simplify the web pages, which is demonstrated in Fig. 2(c), (b) and (d).

4.2 Computing Visual Characteristics on Punctuation Marks Appearance

Text density, layouts of web pages and some other visual characteristics have been developed to improve extraction performance, but they have their own constraints. For example, text density can only work well for long text, in fact, from the structural viewpoint of web pages, a paragraph of text is often organized by multiple HTML tag pairs for the purpose of presentation, which separates a paragraph of text into several short text block. Here, we will develop the intrinsic characteristic of text content to aid extraction.

Punctuation marks are necessary for a text to describe some specific things. A topic-related web page often holds several paragraphs of text to tell a complete story, in which punctuation marks are necessary. According to the large number of observations and statistics on web pages, punctuation marks present apparent usage difference between the required content and those areas holding noise, such as navigation, recommendation links, etc. The required text in web pages are usually accompanied with a large number of punctuation marks since they are presented for easy reading. Though noise, such as navigation panels, advertisements, etc., may be presented in text, they contain few punctuation marks. Figure 3 makes an illustration about the punctuation mark characteristics in different areas of a web page. Figure 3(a), (b) and (c) show parts of the required content, navigation and recommendation links respectively and their corresponding DOM trees, punctuation marks are more used in the area holding the required content, but rarely appear in those noise areas. The above observation is an important cue for extraction, we will use the punctuation mark characteristics accompanying with the required content to identify those effective text nodes with distinctive punctuation marks.



"Australia is a gold mine. The next gold mine," Mr Ma said in Melbourne on Saturday.

"The clean water, the soil and the air, this is what you have, the most unique asset."

With China's pollution problem, there's no question as to why China would want to suck up some of Australia's environment.

(a) an excerpt of the required content and its DOM tree

National World Lifestyle

'Gold mine': From schoolteacher to $43 billion man

(b) navigation and its DOM tree

RIGHT NOW IN FINANCE

857 READERS Donald Trump is treating his presidency like a reality TV sh...

453 READERS Alibaba Australia: Jack Ma speaks in Melbourne about online ...

191 READERS First homebuyer in Sydney says 'rentvestor' movement is best...

139 READERS Donald Trump impeachment: Could US president be voted out?

19 READERS Hutt River Province: Australia's Prince Leonard gives up thr...

(c) recommendation links and its DOM tree

Fig. 3. Punctuation marks in different areas of web page

In order to identify effective text nodes with the help of the visual punctuation mark characteristics, we define the Visual Value of Text Characteristics($vvtc$) in Formula 1 to weight the text characteristics of each node in DOM tree, which combines the text length and the punctuation mark weight for each node.

$$vvtc_a = \frac{text\_length_a}{nodes\_num} * \frac{punc\_num_a}{nodes\_num} \tag{1}$$

Here, a is a leaf node in the DOM tree, $text\_length_a$ is the length of the text held by the node a, $punc\_num_a$ is the total number of various punctuation marks in the above text, $nodes\_num$ is the number of all leaf nodes in the DOM tree. A node with high $vvtc$ value indicates its text should be extracted. All nodes can be sorted in descend order on their $vvtc$ value, the top-k nodes can be kept. For example, the $vvtc$ values of node (1) to (8) in Fig. 3 are 1.25, 1.59, 1.56, 0, 0, 0, 0, 0.17 respectively. Node (1) to (3) will be extracted if top-3 nodes are reserved.

The $vvtc$ value cannot be considered as a direct proof for effective text nodes. It can only tell that a node is an effective text node with a high possibility, especially when some nodes with high $vvtc$ value are organized together in a DOM tree, just like those nodes numbered 1 to 3 in Fig. 3(a). In fact, the visual characteristic exerts a strict operation to discover text content, which will cause it cannot cover the complete text content.

4.3 Aggregating Paths to Expand Content Areas

The visual text characteristic only outputs those tag paths locating text content with apparent punctuation marks. Considering the DOM model, a paragraph of text content in web page is often organized in several different text nodes according to the formatting and presentation requirements, which prevents the visual characteristic from discovering all content that should be extracted. For example, the punctuation mark characteristic does not apply for text that is given as special presentations since it is very short and does not have any punctuation marks, such as using $< h >$ for highlights. Especially, the required text nodes are usually neighbors or siblings in DOM tree though it is possible for them to be separated by some other nodes, such as nodes holding image titles.

Since those text nodes holding a paragraph of text are neighbors or nearby, which implies that tag paths locating a paragraph of text have a common prefix, those text nodes nearby the discovered tag paths may also be the expected ones. We can extend the discovered results on the visual text characteristic to cover the whole content area. Our strategy is to aggregate tag paths to discover the maximum common path prefix and then to locate broad content areas. Here a block of content area is just a subtree constructed by some tag paths with a common prefix.

Those tag paths discovered in Sect. 4.2 need to be aggregated for locating the whole content areas. Supposing X and Y are two tag paths, represented as $X =< x_1, x_2, \cdots, x_i >$, and $Y =< y_1, y_2, \cdots, y_j >$, x_i and y_j are tags, the aggregation of X and Y can be defined in Formula 2. $pcs(X, i, Y, j)$ holds the length of common substring between X and Y. The area covered by the resulted tag path, which is constructed by the first $pcs(X, i, Y, j)$ tags of X, will be considered as content area.

$$pcs(X,i,Y,j) = \begin{cases} 0 \qquad\qquad \textbf{if} \quad (i=0) \quad \textbf{or} \quad (j=0), \\ pcs(X,i-1,Y,j-1)+1 \\ \qquad\qquad \textbf{if} \quad (i,j>0) \quad \textbf{and} \quad (x_i = y_j), \\ max(pcs(X,i,Y,j-1), pcs(X,i-1,Y,j)) \\ \qquad\qquad \textbf{if} \quad (i,j>0) \quad \textbf{and} \quad (x_i \neq y_j.) \end{cases} \tag{2}$$

This above extraction operation aims at covering those effective text content with no punctuation marks. Compared with extraction process in Sect. 4.2, tag path aggregation is a generalized process to locate the content areas and in fact loosens the results, which causes it possible to include some noise in the content areas. For example, image title presented in the middle of a paragraph of text will be extracted in this phase.

4.4 Summarizing Content Semantics and Computing Similarity

In general, both the title of web page and the content covered by the $<$ *description* $>$ tag summarize the primary meaning of the whole text content, and both of them play an important role in revealing the content of web page. In order to refine extraction results, we combine SimHash [21] and Hamming distance to compute the similarity between content area and web page title as well as the content covered by the $< description >$ tag to determine whether the area covered by the result tag path in the second stage is the whole content area or should remove some noise.

Firstly, we use NLPIR word segmentation system [22] to get keywords of each text node in the content areas, and then the following process is exerted on those keywords to summarize text content, which is responsible for transforming a paragraph of text into a feature vector. Figure 4 illustrates the whole process.

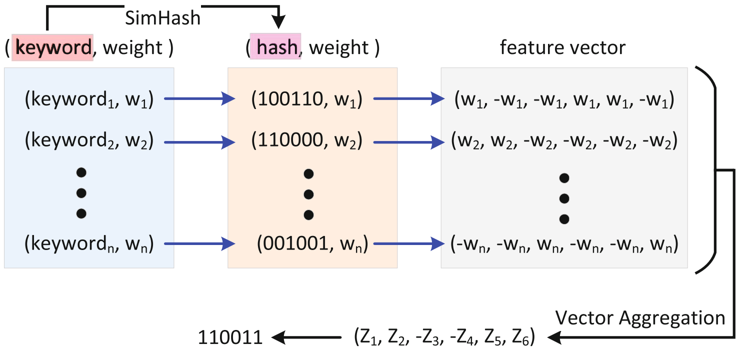


Fig. 4. Summarizing content semantics

(a) computing the weight of keywords. All keywords get their weight in the whole text area by $TF - IDF$ presented in Formula 3, where k represents the k_{st} keyword, N is the number of text paragraphs in the text area, N_K is the number of text paragraphs containing k, L is an empirical constant. And then to construct $(keyword, weight)$ pair set.
(b) computing hash value of keywords. SimHash is applied to get the hash value of each keyword, and then transform $(keyword, weight)$ into $(hash, weight)$.
(c) constructing feature vectors of keywords. For each pair $(hash, weight)$, check each bit of $hash$, and use $weight$ to substitute 1, and $-weight$ for 0.
(d) Outputting the feature vector of content area. Summing all vectors of keywords into a final vector. For each element of the final vector, if its value is greater than 0, then use 1 to replace it; otherwise, 0 is deployed.

$$weight(k) = TF_k * IDF_k = TF_k * [log(\frac{N}{N_k}) + L] \tag{3}$$

Repeating the above process for the concatenation of the title of web page and the content covered by the $< description >$ tag to get the final vector by SimHash. Then Hamming distance is adopted to compute the similarity between two final vectors corresponding to web title as well as the content covered by the $< description >$ tag and each paragraph of text belonging to a leaf node respectively, which will tell how many bits are not consistent in the two final vectors. Apparently, a distance value in the limited range indicates that the paragraph of text has great relevance with web title as well as the content covered by the $< description >$ tag, which should be extracted. A parameter threshold for Hamming distance will be designated for the extraction decisions, which will be discussed in Sect. 5.2.

5 Experiments

In this section, we will firstly discuss the threshold parameters for the chosen characteristics, and then design and conduct experiments on our collected real data sets to investigate the effectiveness of the proposed extraction method. A performance comparison with the popular extraction methods, CETD [13] and CETR [17], is also presented.

Table 1. Web site URLs

No	Website url	Number of web pages	Abbr.
1	news.ifeng.com	100	W_IF
2	news.163.com	100	W_163
3	news.sina.com.cn	100	W_SN
4	news.qq.com	100	W_TC
5	www.huanqiu.com	100	W_HQ
6	news.cctv.com	100	W_CT
7	www.xinhuanet.com	100	W_XH
8	www.chinanews.com	100	W_CN
9	news.sohu.com	100	W_SH
10	www.81.cn	100	W_81

5.1 Experiment Setting and Datasets

We use Java to implement the extraction method, all operations on DOM trees are supported by DOM API in Java. Jsoup [20] Parser is adopted to patch the missed HTML tags and to interpret web pages into DOM trees since it provides very convenient APIs and strong robustness for web pages with complex structures. All experiments run on the following experimental environments, Intel Xeon CPU E3@3.30GHz and 8GB RAM.

We crawled web pages from 10 different web sites and established the data set, which includes 100 different web pages for each web site and covers different web environments. Table 1 lists the URLs of those web sites and their abbreviations as the following labels. Every web page in the data set is firstly interpreted by

Jsoup and then use pre-defined regular expressions to remove irrelevant HTML tags, finally the preprocessed web page is input into our extraction framework to extract the required content.

We use a trivial and boring process, manual XPath expressions combined with artificial confirmation, to extract all the required content accurately and to establish our ground truth for experimental evaluation. Precision($\frac{|ECS|}{|EAS|}$), Recall($\frac{|ECS|}{|GS|}$) and F1-score($\frac{2*|ECS|}{|EAS|+|GS|}$) are used as evaluation metrics. Here, EAS denotes the set of the extraction results from the data set by the proposed method, namely all extracted sentences. ECS denotes the set of the required sentences in the extraction result. GS denotes the set of sentences existing in the ground truth.

5.2 Parameter Analysis

For better extraction performance, we introduce two threshold parameters, T_1 for the Visual Value of Text Characteristics and T_2 for Hamming distance. Especially, T_1 is defined as $\frac{vvtc_a}{MaxVVTC}$, where $MaxVVTC$ is the maximum value of all $vvtc$ values, $vvtc_a$ denotes the $vvtc$ value of the specific node a. A good threshold choice will enhance the extraction results.

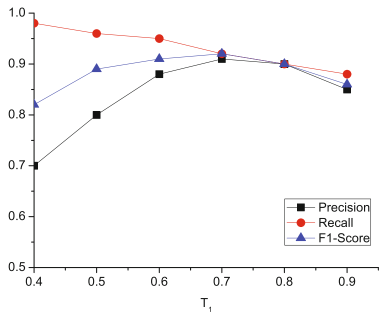


Fig. 5. Recall, Precise and F1-Score on different T_1

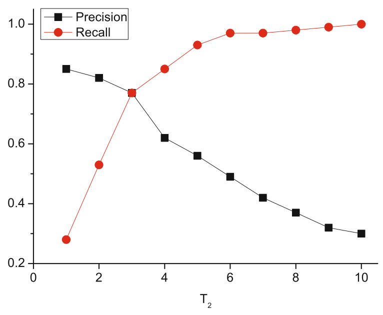


Fig. 6. Recall, Precise and F1-Score on different T_2

We design a group of experiments to reveal the relationships between the thresholds and the extraction performance, whose results are presented in Figs. 5 and 6. The experimental results in Fig. 5 show T_1 should be between 0.6 and 0.8 for a good extraction performance. Figure 6 tells that T_2 should be set to 3 for a tradeoff between precision and recall. In the following experiments, T_1 will be fixed at 0.8 and T_2 at 3.

5.3 Experimental Results and Analysis

For verifying extraction effectiveness, we executed the proposed method(abbr. as VSSE) and compared its performance with the popular extraction methods, CETD [13] and CETR [17]. Our extraction method made a good extraction results, all performance indexes are above 90%, which proves its adaptability for different web environments. The overall extraction performance comparison on all web sites with CETD and CETR are presented in Figs. 7, 8, 9 and 10.

Figure 7 presents the precision comparison, the extraction precision contributed by our method has fully dominated CETR, which is increased by 20% for almost each web site, even up to 35% for some individual web site. For extraction recall, our method and CETR have similar performance, which are presented in Fig. 8. CETR computes the tag ratios per line in web pages, and then extracts content on the basis of clustering, which often brings a big coverage for the required content to cover some noise, this is a key reason for CETR to have a different performance between its precision and recall. The characteristics from visual punctuation marks and the web page structures adopted by our method have the similar function with CETR, but a further extraction decision by content semantics similarity on the extracted items enhances the extraction precision, which also finally contributes to a steady performance on F1-score, the corresponding results are compared in Fig. 9. Compared with CETD, our method presents a better stability on recall though it has a slim advantage on precision, which only won by 2% on average precision. CETD considers the text density as a primary characteristic for extraction decision, the extraction results will present a big error when noise density is over text density in some nodes, such as the No.4 in Fig. 8.

From the experimental results presented in Figs. 7, 8 and 9, we can also see that the extraction performance of our method are very steady on different web sites, but the performance of both CETD and CETR are influenced by the different environments of web sites. Figure 10 makes a more intuitive presentation for their extraction performance by the overall performance on the whole data set. The integration of three different nature of characteristics makes our method work with good adaptability on different web environments.

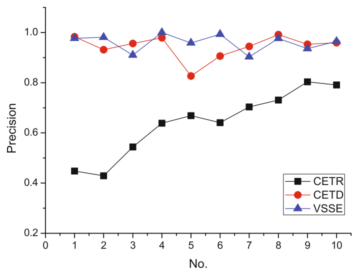


Fig. 7. Precision Comparison

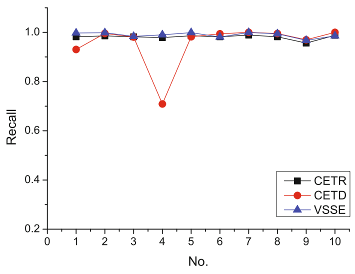


Fig. 8. Recall Comparison

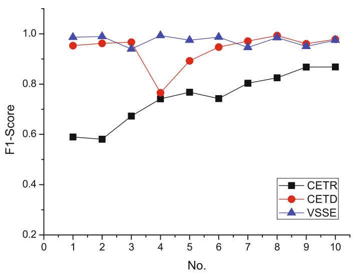


Fig. 9. F1-Score Comparison

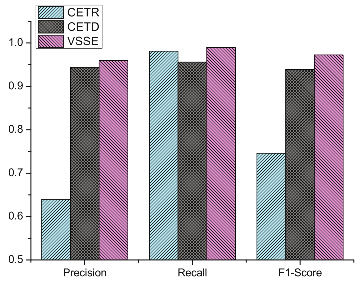


Fig. 10. Overall Performance Comparison

6 Conclusions

This study contributes a novel content extraction method for different web sites, which combines content semantics information with visual text information and web page structures to improve extraction performance. The three kinds of characteristics make a full consideration of the appearance, organizational structure and the semantics consistency of the required content, which provide the extraction solutions with a high precision and recall. Web page structures and content semantics similarity are the inherent characteristics of web pages, which ensure the steady extraction performance. Especially, the proposed method does not depend on the specific semantics of HTML tags, which ensures the extraction adaptability under different web environments. The punctuation mark characteristic depends more on the content organization and presentation formatting, which are very helpful to extract those formal and long content, but when a web page uses more editing tags and more nested structures to present content, its effect will be weakened, web page partition and new visual text characteristics should be considered for performance improvement in future.

Acknowledgments. This work is funded by the National Natural Science Foundation of China (No.61363005, 61462017, U1501252), Guangxi Natural Science Foundation of China(No.2014GXNSFAA118353, 2014GXNSFAA118390), Guangxi Key Laboratory of Automatic Detection Technology and Instrument Foundation(YQ15110), Guangxi Cooperative Innovation Center of Cloud Computing and Big Data.

References

1. Gupta, S., Kaiser, G., Neistadt, D., Grimm, P.: Dom-based content extraction of html documents. In: Proceedings of the 12th International Conference on World Wide Web, WWW 2003, pp. 207–214. ACM, New York (2003)
2. Gupta, S., Kaiser, G.E., Grimm, P., Chiang, M.F., Starren, J.: Automating content extraction of html documents. World Wide Web **8**(2), 179–224 (2005)

3. Zhang, J., Zhang, C., Qian, W., Zhou, A.: Automatic Extraction Rules Generation Based on XPath Pattern Learning. In: Chiu, D.K.W., Bellatreche, L., Sasaki, H., Leung, H., Cheung, S.-C., Hu, H., Shao, J. (eds.) WISE 2010. LNCS, vol. 6724, pp. 58–69. Springer, Heidelberg (2011). doi:10.1007/978-3-642-24396-7_6
4. Alam, H., Rahman, A.F.R., Hartono, R.: Content extraction from html documents. In: Proceedings of 1st International Workshop on Web Document Analysis, WDA2001 (2001)
5. Crescenzi, V., Mecca, G., Merialdo, P.: Roadrunner: towards automatic data extraction from large web sites. In: Proceedings of the 27th International Conference on Very Large Data Bases, VLDB 2001, pp. 109–118. Morgan Kaufmann Publishers Inc., San Francisco (2001)
6. Furche, T., Guo, J., Maneth, S., Schallhart, C.: Robust and noise resistant wrapper induction. In: Proceedings of the 2016 International Conference on Management of Data, SIGMOD 2016, pp. 773–784. ACM, New York (2016)
7. Reis, D.C., Golgher, P.B., Silva, A.S., Laender, A.F.: Automatic web news extraction using tree edit distance. In: Proceedings of the 13th International Conference on World Wide Web, WWW 2004, pp. 502–511. ACM, New York (2004)
8. Wu, G., Li, L., Hu, X., Wu, X.: Web news extraction via path ratios. In: Proceedings of the 22nd ACM international conference on Conference on information & #38; knowledge management, CIKM 2013, pp. 2059–2068. ACM, New York (2013)
9. Gong-Qing, W., Li, L., Li, L., Xindong, W.: Web news extraction via tag path feature fusion using ds theory. J. Comput. Sci. Technol. **31**(4), 661–672 (2016)
10. Cai, D., Yu, S., Wen, J.-R., Ma, W.-Y.: Extracting Content Structure for Web Pages Based on Visual Representation. In: Zhou, X., Orlowska, M.E., Zhang, Y. (eds.) APWeb 2003. LNCS, vol. 2642, pp. 406–417. Springer, Heidelberg (2003). doi:10.1007/3-540-36901-5_42
11. Song, R., Liu, H., Wen, J.-R., Ma, W.-Y.: Learning block importance models for web pages. In: Proceedings of the 13th International Conference on World Wide Web, WWW 2004, pp. 203–211. ACM, New York (2004)
12. Fernandes, D., de Moura, E.S., Ribeiro-Neto, B., da Silva, A.S., Gonçalves, M.A.: Computing block importance for searching on web sites. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM 2007, pp. 165–174. ACM, New York (2007)
13. Sun, F., Song, D., Liao, L.: Dom based content extraction via text density. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, pp. 245–254. ACM, New York (2011)
14. Qureshi, P.A.R., Memon, N.: Hybrid model of content extraction. J. Comput. Syst. Sci. **78**(4), 1248–1257 (2012)
15. Peters, M.E., Lecocq, D.: Content extraction using diverse feature sets. In: Proceedings of the 22nd International Conference on World Wide Web, WWW 2013 Companion, pp. 89–90. ACM, New York (2013)
16. Ortona, S., Orsi, G., Buoncristiano, M., Furche, T.: Wadar: joint wrapper and data repair. Proc. VLDB Endow. **8**(12), 1996–1999 (2015)
17. Weninger, T., Hsu, W.H., Han, J.: Cetr: Content extraction via tag ratios. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 971–980. ACM, New York (2010)
18. Uzun, E., Agun, H.V., Yerlikaya, T.: A hybrid approach for extracting informative content from web pages. Inf. Process. Manage. **49**(4), 928–944 (2013)

19. Weninger, T., Palacios, R., Crescenzi, V., Gottron, T., Merialdo, P.: Web content extraction: a metaanalysis of its past and thoughts on its future. SIGKDD Explor. Newsl. **17**(2), 17–23 (2016)
20. Jsoup. https://jsoup.org/
21. Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing, STOC 2002, pp. 380–388. ACM, New York (2002)
22. Huaping, Z.: Nlpir. http://ictclas.nlpir.org/