International Journal of Engineering & Technology, 8 (1.4) (2019) 458-470

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Menu Recognition Using Rendered Webpage Information and Machine Learning

Chan Choi[1], Minwoo Park[2], Geunseong Jung[3], Jaehyuk Cha [*]

Department of Computer Science, Hanyang University, Seoul, Korea
[1] chaney89@hanyang.ac.kr:
[2]pmw9027@hanyang.ac.kr:
[3]aninteger@hanyang.ac.kr:
**Corresponding author E-mail: chajh@hanyang.ac.kr:*

Abstract

Among the many components of a webpage, the web menu provides organization as well as information on the primary content of the website. By recognizing the menu existing within a webpage, it is possible to identify the overall structure of the website from the menu's information for crawling and indexing. Therefore, tasks can be performed efficiently without accessing the pages unnecessarily. In addition, a variety of applications could be developed through web menu detection. This paper suggests a method to categorize the web menu within a webpage using machine learning. Rendered attribute values of the web document are extracted with processed data to offer more choices in selecting the machine learning attributes. This paper also proposes and demonstrates a Chrome extension-based webpage document collector that effectively collects the final rendered form of a webpage in a browser and its internal data that are required when performing machine learning for web menu categorization. Lastly, a demo platform that can detect webpage menus in real time is designed based on the learned result.

Keywords: *web menu; machine learning; feature selection; logistic regression*

1. Introduction

Webpages provide a lot of information to users. The main information areas, or the advertisement areas, in a webpage provide various contents that attract the user's interest. However, much of the information on webpages may not be always useful to the user. Many studies are currently being performed on categorizing webpage content before they are provided. Using webpage content analysis for providing the categories or patterns required by a user is known as web content mining [1].

With the advances in the web technology, the configurations of modern webpages are becoming increasingly complex. The core components of a webpage are its menu and the primary content that the particular page aims to deliver. In addition, a standard webpage includes some non-essential information for users such as advertisements. In this regard, many studies are conducted that evaluate the structure of individual webpages [2].

A webpage's menu contains the overall organization of the website and information on its primary content [3]. Crawlers or indexers generally determine a webpage's relevance according to the terms and distribution of hyperlinks in the page. Webpages contain a considerable number of normal terms and hyperlinks [4]. Using the order of crawling as the basis for understanding a website's organization is inefficient. However, when the website's menu is recognized and the information within it is conveyed to the crawler, the crawler classifies the menu node and the website's organization is easy and efficient. Thus, the computation amount reduces, because only the links with high relevance are accessed through the menu's categorization system, and the indexer or crawler does not follow every link to collect the website's information. Various services using artificial intelligence, as well technology that gathers the learning data for each algorithm, are required. The Web is a source of essential information; thus, many studies have been performed on technology for collecting information from web documents in particular [2–4]. Conventional web document collectors only collect the document's original-format file. However, the webpages include a technology that creates content dynamically or has components that change in real time according to the usage environment. When collecting the original file of this type of pages, the actual contents to be rendered are excluded from the file, or it is formed in a different format than the content verified by the user. In addition, the elements rendered asynchronously may have a difference in contents between when they are collected and when they are viewed. Thus, collecting accurate data using only a conventional collector is challenging. We propose a web document collector that effectively collects the final form and its internal data rendered in an actual browser, rather than the document's original file. We design the proposed collector based on a browser extension so that it can operate in a normal developer environment, and we used a Chrome extension for implementation.

This proposed method performs categorization of menus in webpages, using machine learning. Machine learning depends on training sessions in which the system acquires specialized knowledge of a domain, thus making it suitable for extracting specific domain information

Copyright © 2019 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

from a web source.

Machine learning methods require a training stage that provides a part of webpages obtained by a domain specialist from other and the same website, and then manually labeled. Providing examples of webpages in the same domain but with a different structure requires caution. Different templates are typically selected to create webpages with dynamic contents even in the same domain scenario, so the system learns a method of extracting the web contents in this type of context.

Machine learning performs categorization of web contents using logistic regression that corresponds to supervised learning. The core of machine learning is to provide the appropriate properties. The proposed method provides property values that were directly processed for obtaining the properties of web documents rendered to allow more diverse choices when selecting machine learning properties. In addition, we designed a rendered webpage document collector based on the Chrome extension function, required for menu categorization during machine learning. Finally, we designed a demonstration platform that categorizes webpage menus in real time based on the results of machine learning.

2. Related Work

Webpage segmentation refers to grouping and dividing the content in webpages according to an algorithm or a specific standard. Research on webpage segmentation is ongoing according to the development of webpage technology and variety of forms in webpages [5]. Julian proposed an algorithm that categorizes menus using only DOM information [6]. However, the algorithm is inefficient because it does not reflect webpage compositions that are gradually becoming more individualized because values, such as "menu" for the "id" value, are used in the DOM properties.

Web mining analyzes all data obtained from the Web. It recognizes useful pattern information in the data appearing on the Web or stored in websites. Because it obtains information based on the vast amount of data appearing on the Web, it exhibits characteristics applicable to various fields [6].

There are many approaches in Web Page Segmentation. Bar-Yossef and Rajagopalan try to identify template blocks by finding common shingles. Cai et al.'s VIPS algorithm used information after rendering the DOM [2].

Web mining has three forms. The first is web structure mining, and its purpose is to obtain information on the structural elements of websites and webpages. Website structural information refers to a graph structure composed of hyperlinks between webpages. The second form of web mining is web content mining; it is a method of extracting significant content from the pages that constitute an actual website. It is a type of information extraction and is closely related to text mining. In other words, it is a technology that automatically finds useful information from the vast amount of web data available online. The third form is web usage mining; it analyzes the usage patterns of web users. This facilitates understanding the web users' behavior beyond access statistics and recognizing webpage usage patterns.

Web content mining extracts useful data, information, and knowledge from the contents in a webpage [6]. Automatic web searching and indexing tools were created because of lack of structural components, such as hypertext documents, which allow information resources to expand continually on the World Wide Web. Although they are convenient to the user, these tools do not provide structural information, and do not perform category filtering and document analysis. Thus, more intelligent tools for information searching, such as intelligent web agents, have been developed over the past few years. Database and data mining technology have expanded, thereby a high level of service has been achieved for semi-structured data that can be used on the Web. In semi-structured data, most tasks use the HTML structure in the documents, and some use the hyperlink structure between documents for expressing the documents.

While a regular web document refers to a Document Object Model (DOM) that includes the website's configuration information, a rendered web document includes the final form and visual information of the original webpage rendered by a browser [7]. Because the Web is used as a source of core materials, studies on web document collection technology have been actively conducted [8–10]. The characteristic of the existing web document collectors is that they only collect the documents' original form.

Web navigation supports searching for information on the Web, which is composed of hypertext or hypermedia [5]. A website's overall navigation system includes several search items such as global, local, and context. There are three types of web navigation. Hierarchical website navigation is used when navigating from a general to a specific thing. It provides a clear and direct path to all pages in a website. Global website navigation shows the website's highest-level sections or pages. It can be used on each page, and it lists the major content sections or pages of the website.

Local website navigation connects with the webpage's text and connects to other pages within the website. These types of web navigations belong to various forms of web navigation categories and allow efficient webpage navigation.

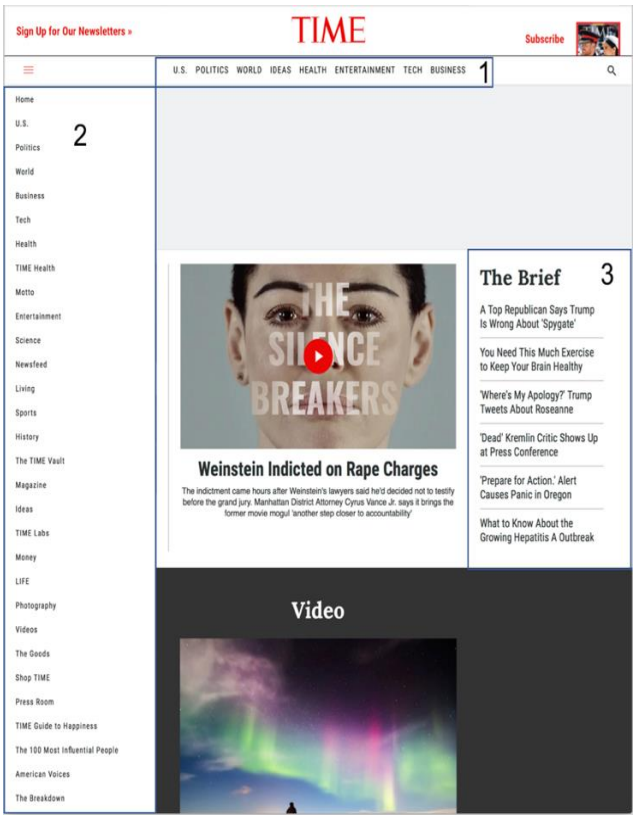


Fig. 1 web navigation types

3. Rendered Web Document Collector

Webpages tend to embed technologies that configure the content dynamically or change the configuration attributes in real time based on the usage environment. When collecting the original form of such a page, there are concerns about missing the content that must be rendered in real time within the file or configuring to a different form from what is visually confirmed. Moreover, the attributes rendered asynchronously may differ in content between the time of collection and time of confirmation. In other words, collecting accurate materials using the existing collectors is challenging. Hence, we designed a web document collector that gathers the final form the embedded information rendered by the browser.

Table 1 Collected site addresses

Wordpress top 100 sites		
www.wordpress.com	www.sitepoint.com	www.webgains.com
www.jquery.com	www.theblaze.com	www.justjared.com
www.xda-developers.com	www.perezhilton.com	www.labnol.org
www.tutsplus.com	www.celebuzz.com	www.firstpost.com
www.smashingmagazine.com	www.studiopress.com	www.rocketnews24.com
www.cbslocal.com	www.css-tricks.com	www.downloadha.com
sedoparking.com	www.template-help.com	www.problogger.net
www.empowernetwork.com	www.thenextweb.com	www.ford.com
www.smartpassiveincome.com	www.failblog.org	www.qualtrics.com
www.zendesk.com	www.vanguardngr.com	www.sixrevisions.com
vimeo.com/	www.lapatilla.com	www.templatic.com
www.dyndns.org	www.infowars.com	www.thesuperficial.com
www.thechive.com	www.quantcast.com	www.publico.es
www.glispa.com	www.hongkiat.com	www.cheetahmail.com
www.arstechnica.com	www.intercambiosvirtuales.org	www.makeuseof.com
www.debonairblog.com	www.searchenginejournal.com	www.gamestlbb.com
www.wnd.com	www.copyblogger.com	www.oneddl.eu
www.dotcomsecrets.com	www.whmcs.com	www.unblog.fr
www.dreamhost.com	www.funpatogh.com	www.allkpop.com
www.sonymobile.com	www.addictivetips.com	www.radio.com
www.support.wordpress.com	www.designmodo.com	www.webdesignerdepot.com
www.html.it	www.wpmudev.org	www.redmondpie.com
www.usafis.org	www.yoast.com	www.envato.com
www.dailycaller.com	www.icanhascheezburger.com	www.noticierodigital.com
www.socialmediaexaminer.com	www.mydealz.de	www.dawn.com
www.tripwiremagazine.com	www.dcwg.org	www.babosas.com

www.diythemes.com	www.imnicamail.com	www.allthingsd.com
www.giveawayoftheday.com	www.serienjunkies.org	www.wpbeginner.com
www.filmifullizle.com	www.hotair.com	www.deadline.com
www.boingboing.net	www.te3p.com	www.forums.wordpress.com
www.speckyboy.com	www.gruenderszene.de	www.creativecommons.org
www.hollywoodlife.com	www.socialspark.com	www.slodive.com
www.blogdetik.com	www.hasoffers.com	

3.1 Data collection

In this study, we collected the menu data from the top 100 WordPress-powered websites ranked by Hacker Target. Almost Web Menu node based on this system consisted of a *'nav'* tag, so we restricted exam to web pages which are based on the WordPress. Hacker Target is a security agency that ranks websites based on Internet traffic scanning and provides security solutions. Their rankings include websites from different countries around the world. Therefore, we collected various menu forms from the websites of different countries.

3.2 Processing mechanism

Among the web browser extensions, we used the Chrome extensions. Web browser extension programs are for a specific web browser to either change an existing feature's function or to add a new feature. Most HTML5-based web browsers, except IE (Internet Explorer), support browser extensions via JavaScript. Chrome extensions can modify, change, and improve the features of the Chrome browser using web technologies such as HTML, JavaScript, and CSS [11]. Chrome extensions have two types of script pages with different task functions. The first is the event page that is a script to manage a specific task or status of apps or extension programs. Event pages can exchange messages with the content page or other extensions. The second is the content script, and it is a JavaScript file written to an existing web document. The content script can access the DOM of existing web documents and modify, write, and delete.

We configured the collector as in Fig. 2. We sequentially executed the eight steps of collection phases. For a semi-automatic collection, the fourth phase can be skipped.

1. The Tab Handler manages events for the tabs and searches the collection list from the List Manager that manages the URL list to be collected.

2. The Tab Handler creates a tab, renders the URL searched in the collection list, and examines the time during which the tab is open. If the tab's opening time exceeds a specific period, Step 8 is executed.

3. After rendering a webpage, the content script is inserted into the target document, and a DOM validator identifies the abnormalities in the created DOM through a DOM parser.

4. The Event Handler processes events such as warning popups that may interfere with the collector's actions.

5. The CORS Handler handles the same-origin policy violation problem for the area to be collected.

6. The DOM Handler modifies the DOM to the user's desired form and collects the materials.

7. The collected materials are connected to the internal or external storage module via DB & File API.

8. The Tab Handler closes the opened tab and executes Step 1.

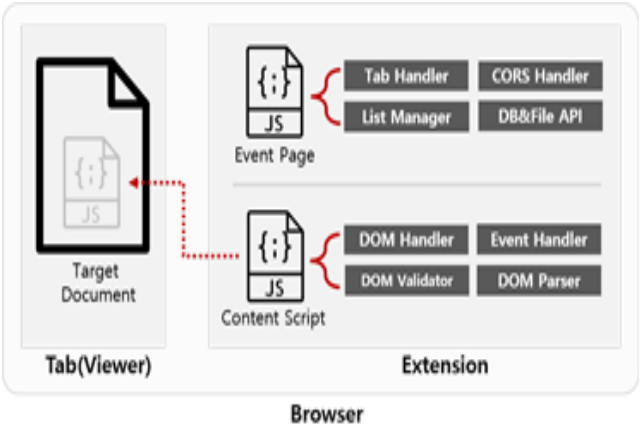


Fig. 2 Structure of collector-based browser extension

3.3 Features of the rendered web menu

Features in machine learning are essential for classification accuracy. Providing various attributes improves the accuracy by forming more diverse combinations during feature selection. First, the Elements in HTML have various properties for representing a webpage [7]. We choose basic properties from the properties of HTMLElement and Element about their size and positions: clientWidth, clientHeight, offsetLeft, offsetTop, offsetWidth, offsetHeight, scollLeft, scrollTop, scrollHeight, and scrollWidth. We also collected properties derive from

DOM structure and hyperlink because the hierarchical relations among the elements and the distribution of textual information could help to identify the main block of content [12]. Finally, we obtained 19 attributes for machine learning for web menu recognition. Table 2 presents the 19 properties that we collected for machine learning that categorizes the web menu. The links in Table 2 refer to anchor nodes.

Table 2 Collected data attributes

Basic Properties	**Explanation**
clientWidth	The actual width of the element (including the padding size)
clientHeight	The actual height of the element (including the padding size)
offsetLeft	X coordinate of body tag reference element
offsetTop	Y coordinate of body tag reference element
offsetWidth	Width of element (Including the size of padding, border, and scrollbars)
offsetHeight	Height of element (Including the size of padding, border, and scrollbars)
scrollLeft	The X coordinate of the element's scroll
scrollTop	The Y coordinate of the element's scroll
scrollHeight	Scroll height of element
scrollWidth	Scroll width of element
Additional Properties	**Explanation**
childNum	The number of child nodes of the element
linkNum	The number of links in child nodes
linkRatio	The ratio of the number of text words in the child node to the number of links
linkRatio2	The ratio of the number of tags to the number of links in child nodes
Depth	Distance of corresponding node by body tag
avgLinkDepth	The average distance between links of the node and child nodes
textWordsLength	The number of text words that child nodes have
childrenSibling TagRatio	Tag match ratio between child nodes
disCenter	The ratio of the distance between the center point of the corresponding node to the center point of the body tag

We collected the numerical attributes, such as the absolute and relative location values, from the information included in the rendered web document. The reason for location information collected was that the menu can be mostly found at a specific location. We collected the number of child node links because the menu should include many links by nature. We calculated the ratio of the number of text words in the child nodes to the number of links by considering the menu's characteristic reduced form. As the menu is often abbreviated and categorized, the content does not include many words [3]. The ratio of the child node's tag number to the number of links indicates that it is not a menu if substantially more tags exist than links [12]. We determined the average level between the links of the corresponding and child nodes by obtaining the level between each node and the links when multiple links existed among the child nodes of the corresponding nodes.

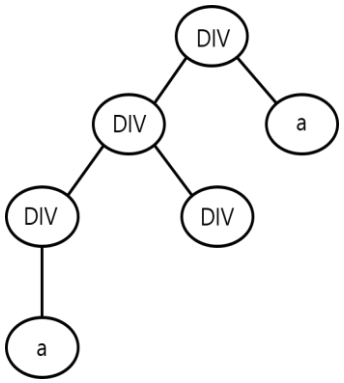


Fig. 4 avgLinkDepth attribute

DIV refers to a block, and a refers to a link. When the current node is a route node, the distance between the route node and node on the right is 1. The distance to a node on the left is 3. Therefore, the average distance between the links of the corresponding node and the child nodes is (1+3)/2; therefore, 2 is the attribute value. For the number of text words, the reason is the same as the case of the ratio of text–word number to the link number of the child nodes. The tag coincidence rate between the child nodes is an item added by considering the form of the menu's organization. This is because the nodes under the menu show a repeatedly coinciding style. When there is one tag, which indicates a leaf node, the tag coincidence index is 1. Therefore, we handled it as an exception when the ratio was collected. Lastly, the distance ratio between the midpoints of the corresponding node and midpoint of the body tag is an attribute representing how far from the midpoint the corresponding node is located. In this paper, we occasionally found many links at the webpage midpoint in the same form as the menu. In those cases, we did not categorize the corresponding element as the menu, but as a core content element that the relevant page aimed to provide. The ratio from Fig. 4 as an example is distance 2 over distance 1. Similarly, we obtained the distance ratio from all

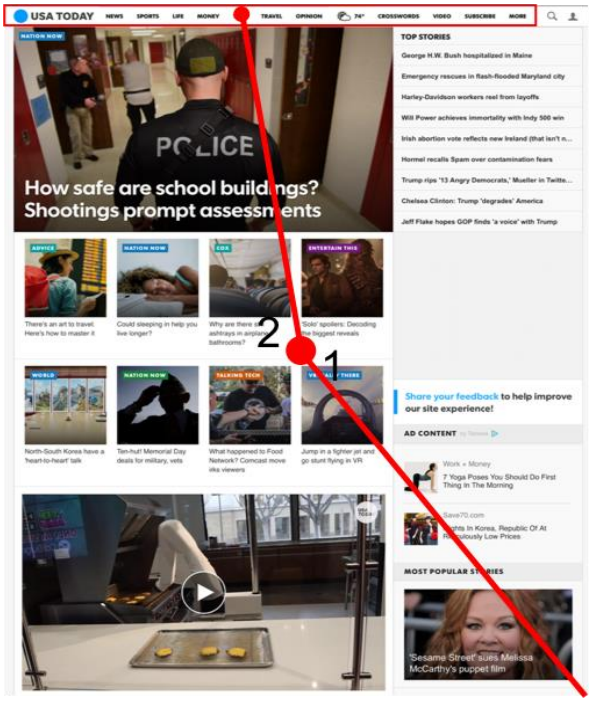


Fig. 5 disCenter attribute

nodes by distance 1.

3.4 Improving the collector performance

The goal of the collector was to quickly collect a large amount of data. The proposed collector periodically checks the time for the collection process to reduce the data collection time. During examination, when the time limit set by the user exceeds, Step 8 of the collection process, described in Section 3, is performed. Through this process, webpages that consume time until rendering is complete are processed to prevent performance reductions. The time limit set by the user affects the amount of data collected. For example, when the time limit setting is less than the universal rendering time of the webpages to be collected, the amount of data collected may become smaller.

In addition, the data collection time is reduced by collecting data in tab units. Table 3 shows the experimental environment for confirming the performance changes according to the number of tabs.

Table 3 Experiment environment

CPU	Intel Dual Core 3.40GHz
RAM	12GB
OS	Window 10 (64bit)
Chrome	58.0.3029.81
Network	Ethernet (1.0Gbps)

We divided the experiments into four conditions for the number of tabs (1, 5, 10, and 20 tabs) and measured the collection time for 100 webpages. We performed the collection task five times for each condition and compared the mean collection times for each condition. The targets for collection in the experiments were rendered web documents from 100 randomly selected webpages. Various methods are available for storing data, and the storage method can affect the collection time comparison. In this experiment, we executed Step 7 from the collection process described in Section 3.1. Fig. 6 shows the experiment results.

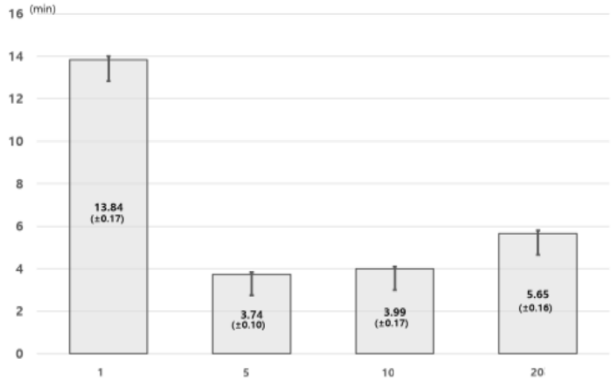


Fig. 3 Data collection time according to the number of tabs

In Fig. 6, when the number of tabs is 5, the collection time is reduced by 381% compared with when the number of tabs was 1. By contrast, when the number of tabs increased to 10, the collection time was not reduced compared with when the number of tabs was

5, and when it was 20, the data collection time actually increased. For this change, two reasons are predicted. First, there may be a bandwidth limit. Performance may not increase even though the number of tabs increases because the maximum capacity for bandwith is fixed. The second reason is that the program's performance improvements may be limited when the computer uses a parallel processor [13]. Thus, verifying the limits to performance improvements through future experiments is necessary.

When the collector is implemented based on a Chrome extension, same-origin policy violation issues can occur. This policy restricts the documents or scripts loaded from one source from interacting with the resources from another source [14]. In this paper, we overcame this issue by using the following procedure:

1. The content script extracts the "src" properties from all of the <img> nodes in the webpage and sends them to the event page as messages.

2. The event page loads the "src" properties sent by the content script on the <canvas> and converts them into data values and sends them back to the content script.

3. Finally, the content script changes the existing "src" properties to the received data addresses

4. Classification and Result

4.1 Data preprocessing

Imbalanced data significantly impacts the performance of most machine learning algorithms. Most existing data are imbalanced, and this significantly affects the performance of machine learning [15]. Data imbalance mainly refers to the issues in data distribution. When data belonging to the majority category are excessively larger than those in the minority category, and when the majority category intrudes upon the minority category, setting the border line for categorization is difficult.

Methods for dealing with imbalanced data are broadly divided as under-sampling and over-sampling methods [16]. Under-sampling, according to a predefined rule, takes a sample of the majority data items, which contains as many items as the number of minority data items. We verified the methods in practice through simple experiments, and in under-sampling, the results showed that the amount of data loss was large, but the numerical accuracy was relatively high because we set the amount of data to match the number of minority category items. However, in real world, expecting this method to categorize web menus in real time based on the learning model is challenging. Over-sampling, following a predefined rule, takes a sample of the minority data items, which contains as many items as the number of majority data items. The weight value of the minority data items increases, and the numerical accuracy tends to be less than that of under-sampling because the minority category data intentionally increases to match the size of majority category data.

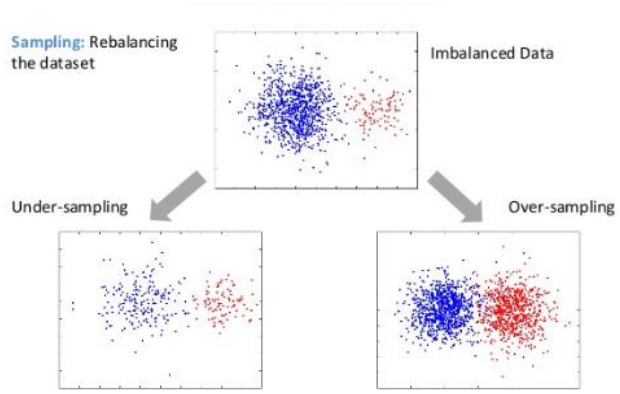


Fig. 4 Over-sampling and Under-sampling

Under-sampling takes a sample of majority category data items as many items as the number of minority category data items according to a predefined rule. This method has an advantage of a short computation time. A disadvantage is that the data sampled from the majority

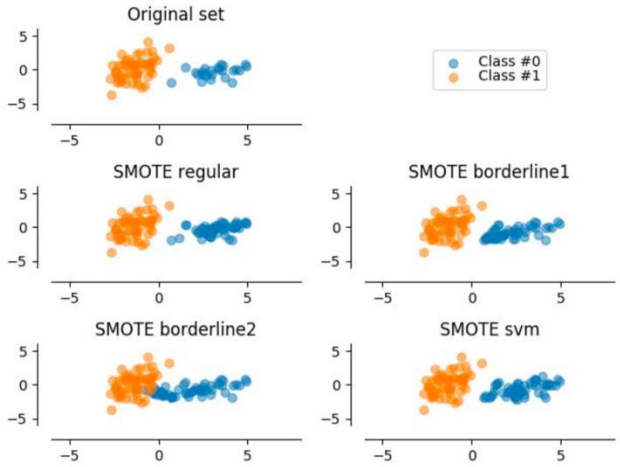


Fig. 5 Techniques in SMOTE

category affects the performance of the categorizer. Under-sampling methods include random under-sampling, near-miss, and Tomek links.

Random under-sampling [17] randomly reduces the amount of data in the majority category to the number of items in the minority category data set. Nearmiss sampling [18] does not sample the entire dataset that belongs to the majority category, but only selects a part of it and takes samples until the numbers of data items in the two categories become identical. This method has three forms. The first method, Nearmiss1, selects the three data from the majority category, which have the closest mean distance to the minority category data items. Of these, the data with the smallest distance are sampled. The second, Nearmiss2, selects the three data from the majority category, which have the farthest mean distance to the minority category data, and it samples all three. The third, Nearmiss3, selects and samples the majority category data, which are the closest to each minority category data.

Tomek links [19] use the distance values between samples. Different categories are connected by making the samples with the minimum distance from each other into pairs. For example, the x pair belongs to the (minority of S), and x belongs to the (majority of S). In addition, x is the distance to x, and when there is no sample with x or x, it is a Tomek link. When two samples form a Tomek link, one of the two samples is noise or both exist near the category border. Therefore, after combining the data, all Tomek links eliminated the overlapping between unwanted categories. The removal process continues until all minimum distance connected pairs are in the same category.

Over-sampling, following a predefined rule, takes a sample of the minority data as many items as the number of majority data category. An advantage is that there is no data lost, so all the data can be used. A disadvantage is that the number of data increases, so the amount of time consumed increases. In this paper, we elucidate the most typical over-sampling method, the SMOTE algorithm. SMOTE creates data that belong to the minority category [20]. First, SMOTE takes a data sample in the minority category, and then it finds the sample's k-nearest neighbors. Then, it finds the difference between the current sample and k-neighbors, and this difference is multiplied by a random value between 0 and 1, and then added to the original sample. This created sample is added to the training data. Thus, SMOTE functions as a method that adds points that are moved slightly from the existing sample while considering the surrounding neighbors.

[21] proposes an over-sampling method called borderline SMOTE. The existing SMOTE algorithm simply performs a sampling from the minority class randomly, but borderline SMOTE focuses on the part that is harder to categorize by expanding the samples that are on the borderline with the other class, as shown in Fig. 8.

4.2 Logistic Regression

Logistic regression, a probability model proposed by D.R. Cox, is a statistical technique used to predict the probability that an event will occur by using a linear combination of independent variables. [22] Logistic regression represents the relationship between dependent and independent variables as a function so that it can be used in later prediction models. It is a type of categorization technique. Logistic regression can be binomial or polynomial. In binomial and polynomial logistic regressions, there are two and more than two results of dependent variables, respectively. Because the data used in this paper has two dependent variables, we describe binomial logistic regression.

The logistic model equation is such that no matter what number the independent variable is, the dependent variables or the result values are always within the range of [0,1]. This is achieved by performing a logit conversion on the odds ratio, shown as follows. The meaning of this equation is the ratio of the probability of success to the probability of failure. A suitable value for x can be substituted in the equation to better understand its meaning.

$$odds = \frac{p(y=1|x)}{1-p(y=1|x)} \tag{1}$$

When p was 0, the above equation was 0, and when p was p, the equation was 1. When p was 1, the equation was p. This indicates that, when p approaches p, the equation moved toward 0, and as p approached 1, it moved toward x, but the range was limited to [0,1].

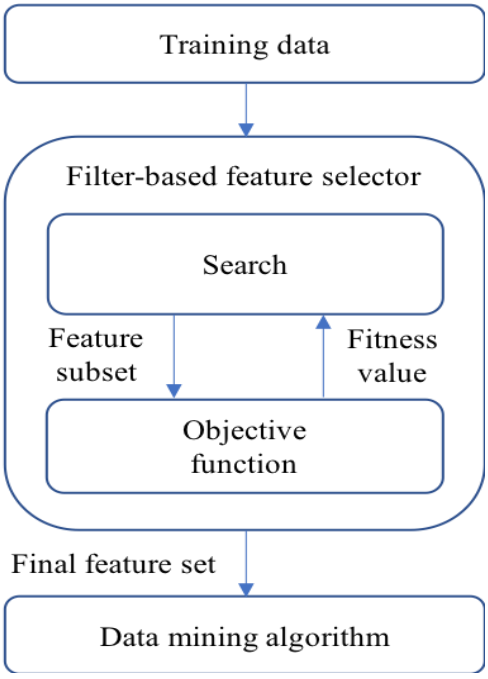


Fig. 6 Filter-based feature selector

$$logit(p) = log\frac{p}{1-p} \tag{2}$$

Therefore, by finding the log of the odds as in the preceding equation, we adjusted the output value range to [0,1] when the input value's range was x. In logistic regression, the logit conversion results are the same as a linear function for x, and they are expressed as follows.

$$log\frac{p}{1-p} = ax + b$$

$$\frac{p}{1-p} = e^{ax+b}$$

$$p = e^{ax+b}(1-p) \tag{3}$$

$$p = \frac{1}{1+e^{-(ax+b)}}$$

Thus, the probability that the independent variable will belong to the category is 1. We generalized this to create the following equation, called the logistic function:

$$p_i = log^{-1}(\beta \cdot X_i) = \frac{1}{1+e^{-\beta \cdot X_i}} \quad (4)$$

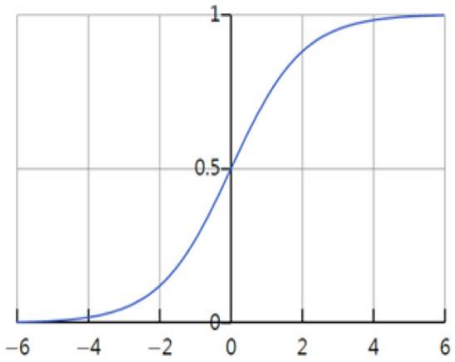


Fig. 7 Logistic function

We used Logistic Regression model to extract the Web menu from Web pages. The goal of the proposed method is to identify the menu element among the components in web pages. Since the logistic regression technique provides us more intuitive and interpretable results, we adopted it to classify the collected data whether they are menu components in web pages

4.3 Technique of Machine Learning

Feature selection is determining the most remarkable combination of many attributes provided during machine learning. In other words, feature selection increases the estimation accuracy by selecting the input variable having a high correlation with the output variable. Available feature-selection techniques include sequential forward selection (SFS), sequential forward floating selection (SFFS), and exhaustive feature selection (EFS).

Feature selection chooses the variables that will be used in machine learning; this is also called variable selection. Feature selection selects input variables that are highly related to out-put variables to increase the accuracy of predictions. Feature selection methods are broadly divided into filter and wrapper methods [23].

The filter method selects a set of good properties from all properties. In this case, it is defined independently from the machine learning algorithm; thus, property sets with maximum machine learning performance are created by using this method. The filtering method provides rankings of the properties' importance, which are criteria to assist the experimenter. The method's independence from the machine learning algorithm indicates that the filtering results do not find the best property sets for machine learning, but provides property rankings to the experimenter, which can indicate the extent of influence of a property and whether the property produces good results in machine learning.

The filter method selects properties independently from the model's performance results as part of the machine learning preprocessing; by contrast, the wrapper method uses the machine learning algorithm when selecting properties, so it selects the property sets with best performance. However, unlike the filter method, the wrapper method uses the machine learning algorithm to select property sets; thus, it takes longer.

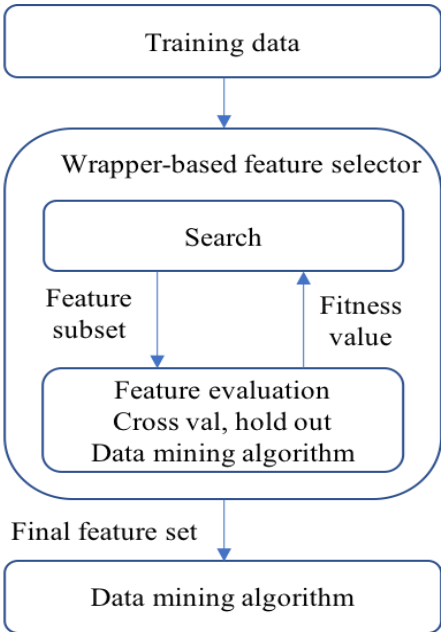


Fig. 11 Wrapper-based feature selector

In this paper, we used the EFS technique that forms every combination of all attributes and uses them as an input to the machine learning algorithm. However, this has the drawback of being time-consuming. However, it has the significant advantage of identifying the best attribute.

We classified the previously described data in two ways. The menu node is set as positive, whereas we configured other nodes as negative for the classification using binomial logistic regression. Among the collected data, we used 80% as learning data and 20% as test data. The graph in Fig. 11 presents the result classified by selecting attributes using the EFS technique after using SMOTE over-sampling on the collected data.

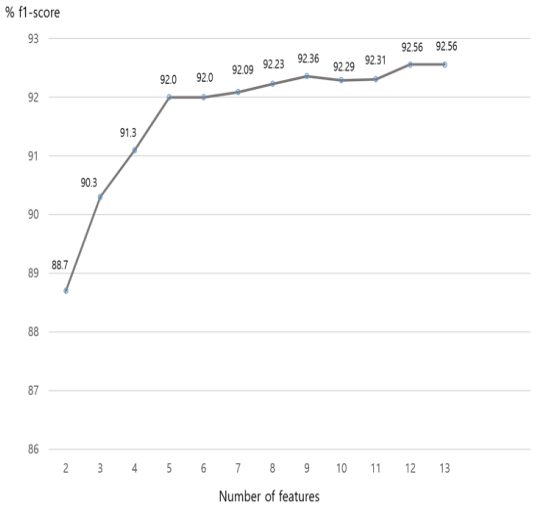


Fig. 12 f1-score of experiment

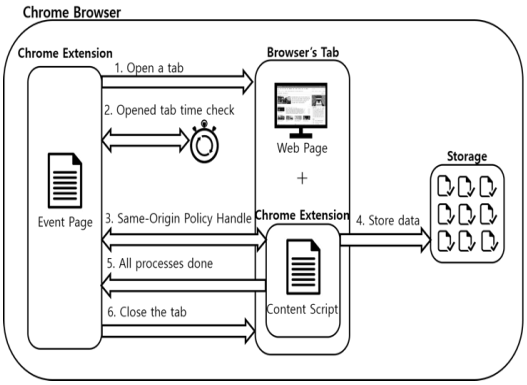


Fig. 14 Configure Menu Classification Check Tool

From the graph, the result increased as the number of attributes selected increased. However, the f1 score decreased when we selected 10 attributes. This indicates that assuming and estimating an unconditional increase is incorrect. When we selected more than 12 attributes, we observed the same performance as for 12 attributes. The combination of attributes with the best performance was that of offsetLeft, offsetWidth, offsetHeight, scrollHeight, clientWidth, clientHeight, linkup, chillum, linkRatio, glidepath, textWordsLength, childrenSibling TagRatio, and disCenter, expecting a 92.56% classification performance. From the experimental result, we found that the disCenter, children Sibling, and avg Link Depth attributes were frequently selected, regardless of the number of selected attributes. Providing processed attributes in addition to the basic attributes had a positive effect.

From the graph, the number of feature selection shows that there is no big change after five. When we selected more than 12 attributes, we observed the same performance as for 12 attributes.

The combination of attributes with the best performance was that of OffsetLeft, offsetWidth, offsetHeight, scrollHeight, clientHeight, linkNum, childrenTagNum, childrenAllTagNum, nowDepth, avgLinkDepth, textWordsLength, childrenSiblingTagRatio, expecting a 70.56% classification performance.

4.4 Real-time menu detector

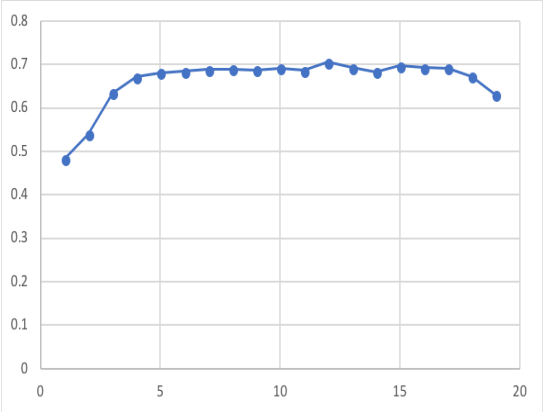


Fig. 13 coefficient of determination of experiment

We designed a demo platform for a real-time menu detector using the preceding experiment and a Chrome extension. The purpose of designing a demo platform was that it improves performance by visually checking the menu detection result and identifying the problems. First, we used a Chrome extension for creating the rendered web document information of the current webpage; we sent this information to the server for logistic regression analysis. As nodes themselves cannot be exchanged, we saved the index value of each node by preprocessing, and the result value was received from the server to compare and distinguish the index and nodes. We analyzed the rendered web document information received for estimating a previously studied logistic regression model before sending the result back to the Chrome extension. The Chrome extension compared the result received from the server with the saved index value to distinguish the node and allow users to confirm it visually.

5. Conclusions

In this paper, we proposed a tab unit collection method to improve performance, and we suggested a technique for extracting web menus within webpages, using machine learning. First, we designed a rendered webpage document collector from a Chrome extension for extracting web menus. In addition, we conducted over-sampling on the collected data t for the machine learning process and performed attribute selection using the EFS technique. Upon adding the self-processed attribute values to the rendered web document attributes, and combining them, we found that the self-processed attribute values were frequently selected. Detection using logistic regression showed a performance result of 92.56%. Lastly, we designed a demo platform for detecting webpage menus in real time from the learned result. This platform provided a base for more accurate analysis and performance improvement by allowing visual confirmation of the classified result. Various other machine-learning algorithms apart from logistic regression and the diverse sampling technique should be used for experiments and comparisons in future research.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (Ministry of Science and ICT) (No. NRF-2016R1A2B4016591).

References

[1] Han J, Cheng H, Xin D, Yan X, Frequent pattern mining: Current status and future directions, *Data Min Knowl Discov*, Vol.15, No.1, (2007), pp.55–86.
[2] Cai D, Yu S, Wen JR, Ma WY, VIPS: a visionbased page segmentation algorithm, *Beijing Miciosoft Res Asia*, (2003), pp.1–29, available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.118.638&rep=rep1&type=pdf.
[3] Insa D, Silva J, Page-Level Webpage Menu Detection, (2016), .
[4] Bar-Yossef Z, Rajagopalan S, Template Detection via Data Mining and Its Applications, Vol.9, (2002), pp.580–591, available online: http://doi.acm.org/10.1145/511446.511522%5Cnhttp://dl.acm.org/ft_gateway.cfm?id=511522&type=pdf.
[5] Lowe D, Gaedke M, Web Engineering: 5th International Conference, ICWE 2005, Sydney, Australia, July 27-29, 2005, Proceedings, (2011), .
[6] Markov Z, Larose DT, Data mining the Web: uncovering patterns in Web content, structure, and usage, (2007), .
[7] Robie J, Ag S, Document Object Model (DOM) Level 3 Core Specification, *W3C Recomm*, No.April, (2004), pp.1–216.
[8] Ochoa Serna J, Design and implementation of a Scraping system for Sport News, (2017), .
[9] Blanvillain O, Kasioumis N, Banos V, BlogForever Crawler, *Proc 4th Int Conf Web Intell Min Semant - WIMS '14*, (2014), pp.1–8, available online: http://dl.acm.org/citation.cfm?doid=2611040.2611067.
[10] Wang J, Guo Y, Scrapy-based crawling and user-behavior characteristics analysis on Taobao, *Proc 2012 Int Conf Cyber-Enabled Distrib Comput Knowl Discov CyberC 2012*, (2012), pp.44–52.
[11] Liu L, Zhang X, Yan G, Chen S, Chrome extensions: Threat analysis and countermeasures, *… Netw Distrib Syst …*, (2012), , available online: https://www.cs.gmu.edu/~sqchen/publications/NDSS-2012.pdf.
[12] Insa D, Silva J, Tamarit S, Using the words/leafs ratio in the DOM tree for content extraction, *J Log Algebr Program*, Vol.82, No.8, (2013), pp.311–25, available online: http://dx.doi.org/10.1016/j.jlap.2013.01.002.
[13] Gorelick M, Ozsvald I, High Performance Python: Practical Performant Programming for Humans, (2014), .
[14] Ruderman J, Same origin policy for JavaScript, *Online Https//Developer Mozilla Org/En/Same Orig Policy JavaScript*, (2009), .

[15] Maimon O, Rokach L, Data Mining and Knowledge Discovery, *Kodo Keiryogaku (The Japanese J Behav*, (1999), .
[16] An A, Cercone N, Huang X, A case study for learning from imbalanced data sets, *Conf Can Soc Comput Stud Intell*, (2001), pp.1–15.
[17] Japkowicz N, Learning from Imbalanced Data Sets: A Comparison of Various Strategies, *AAAI Work Learn from Imbalanced Data Sets*, Vol.68, (2000), pp.10–5.
[18] Mani I, Zhang I, kNN approach to unbalanced data distributions: a case study involving information extraction, *Proc Work Learn from Imbalanced Datasets*, Vol.126, (2003), .
[19] Tomek I, Two modifications of CNN, *IEEE Trans Syst Man Cybern*, Vol.6, (1976), pp.769–72.
[20] Chawla N V., Bowyer KW, Hall LO, Kegelmeyer WP, SMOTE: Synthetic minority over-sampling technique, *J Artif Intell Res*, Vol.16, (2002), pp.321–57.
[21] Han H, Wang W-Y, Mao B-H, Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning, *Int Conf Intell Comput*, (2005), pp.878–87.
[22] Cox DR, The regression analysis of binary sequences, *J R Stat Soc Ser B*, (1958), pp.215–42.
[23] Guyon I, Elisseeff A, An Introduction to Variable and Feature Selection, *J Mach Learn Res*, Vol.3, No.3, (2003), pp.1157–82.
[24] Zhai Y, Liu B, Web data extraction based on partial tree alignment, *... 14th Int Conf World Wide Web*, (2005), pp.76–85, available online: http://dl.acm.org/citation.cfm?id=1060761.
[25] Hearst M a, Palo X, Segmentation Expository Text, *Proc 32nd Annu Meet Assoc Comput Linguist*, No.Hearst, (1994), pp.9–16.
[26] Chakrabarti D, Kumar R, Punera K, A graph-theoretic approach to webpage segmentation, *Proceeding 17th Int Conf World Wide Web - WWW '08*, (2008), pp.377, available online: http://portal.acm.org/citation.cfm?doid=1367497.1367549.
[27] Ferrara E, De Meo P, Fiumara G, Baumgartner R, Web data extraction, applications and techniques: A survey, *Knowledge-Based Syst*, Vol.70, (2014), pp.301–23, available online: http://dx.doi.org/10.1016/j.knosys.2014.07.007.
[28] Kushmerick N, Wrapper induction: efficiency and expressiveness, *Artif Intell*, Vol.118, No.1–2, (2000), pp.15–68.
[29] Debnath S, Mitra P, Pal N, Giles CL, Automatic Identification of Informative Sections of Web-pages, Vol.17, No.i, (2005), pp.1–35.
[30] Chen Y, Ma W-Y, Zang H-J, Detecting web page structure for adaptive viewing on small form factor devices, *Proc 12th Int Conf World Wide Web*, No.49, (2003), pp.225--233, available online: http://research.microsoft.com/pubs/68995/p297-chen.pdf.
[31] Baluja S, Browsing on Small Screens : Recasting Web-Page Segmentation into an Efficient Machine Learning Framework, *Entropy*, No.1, (2006), pp.33–42, available online: http://doi.acm.org/10.1145/1135777.1135788.
[32] Manabe T, Extracting Logical Hierarchical Structure of HTML Documents Based on Headings, *Proc 41st Int Conf Very Large Data Bases*, Vol.8, No.12, (2015), pp.1606–17.
[33] Kao HY, Ho JM, Chen MS, WISDOM: Web Intrapage Informative Structure Mining based on Document Object Model, *IEEE Trans Knowl Data Eng*, Vol.17, No.5, (2005), pp.614–27.
[34] Vieira K, da Silva AS, Pinto N, de Moura ES, Cavalcanti JMB, Freire J, A Fast and Robust Method for Web Page Template Detection and Removal, *Proc 15th ACM Int Conf Inf Knowl Manag*, (2006), pp.258–67, available online: http://doi.acm.org/10.1145/1183614.1183654.
[35] Kohlschütter C, Nejdl W, A densitometric approach to web page segmentation, *Proc 17th ACM Conf Inf Knowl Manag*, (2008), pp.1173–82.
[36] Etzioni O, Banko M, Soderland S, Weld DS, Open information extraction from the web, *Commun ACM*, Vol.51, No.12, (2008), pp.68–74.
[37] Etzioni O, Cafarella M, Downey D, Popescu A-M, Shaked T, Soderland S, et al., Unsupervised named-entity extraction from the web: An experimental study, *Artif Intell*, Vol.165, No.1, (2005), pp.91–134.
[38] Cass TA, Formless forms and paper web using a reference-based mark extraction technique, (1997), .
[39] C. Chan, M. Park, G. Jung, J. Cha "Detecting Menu in Rendered Web Document Using Machine Learning Techniques" International Journal of Artificial Intelligence and Applications for Smart Devices, vol. 5, no. 1, pp.1-6, 2018.