An Event Data Extraction Method Based on HTML Structure Analysis and Machine Learning

Chenyi Liao♯1, Kei Hiroi*2, Katsuhiko Kaji%3, Nobuo Kawaguchi♯*4
♯Graduate School Engineering, Nagoya University
Furo-cho Nagoya Japan
1liao@ucl.nuee.nagoya-u.ac.jp
*Institute of Innovation for Future Society, Nagoya University
Furo-cho Nagoya Japan
2k.hiroi@ucl.nuee.nagoya-u.ac.jp
%Faculty of Information Science, Aichi Institute of Technology
3kaji@aitech.ac.jp
4kawaguti@nagoya-u.jp

Abstract—**This paper proposes an event data extraction method that extracts business event data, such as coupons, tickets, sales campaigns, etc., from a homepage or blog of shops and pushes them to users. Users no longer need to browse their favorite shops' homepage one by one. The method supports comprehensiveness and effectiveness for event data obtainment. This proposition works into two tasks: web page block segmentation and event data identification. The first task segments the web page into blocks. Each of the blocks includes information, such as title, notification, date, etc. relating to event information. Many related works suppose web page block segmentation based on specific tags, vision, function, etc. In this research, we propose a web page block segmentation method based on HTML document structure analysis. The second task is used to identity event data from segmented blocks. We propose a method to implement event data identification based on machine learning. We show the results of a verification experiment. Experimental data are from 96 shops located in two underground shopping streets UNIMALL and ESCA, at a train station in the city of Nagoya (Japan). Because the event data identification depends on the Japanese language, this method is available for all the Japanese home page.**

Keywords—Web Mining, Text Classification, Knowledge Processing, Ubiquitous Computing

I. Introduction

Recently, event data, such as exhibition or party notifications, are massed on the Internet. In previous research, we have developed a temporal-spatial visualization web system named Event.Locky[1][2], which obtains structured data from event search APIs, such as ATND and CONNPASS, and visualizes them on a web page. However, some valuable business event data of shopping streets, such as coupons, tickets, sales campaigns, etc., are published on a shop ' s homepage or blog. This event data cannot be obtained from existing event search APIs because the event data is recorded by the HTML document that is made up of half-structured data. It is impossible for an information system to directly use that data. On the other hand, users who need that event information have to access each of the websites one by one. Therefore, it is challenging to support comprehensiveness and effectiveness without an autonomous extraction system. For those reasons, in this research, we develop an autonomous event data extraction system that extracts event data from homepages or blogs of shops and delivers it to users. The official homepage URLs from which we extract event data are obtained from the official website[1] of Nagoya train station. This proposition takes a two-task approach using web page block segmentation and event data identification.

The first task is web page block segmentation. A block refers to an area on a web page. The web page is written in HTML document. Therefore, a block is also a HTML code fragment within the HTML document. A block must include information such as title, notification, date, etc. of an event. Much of the related research that we introduce in section II, proposes web page block segmentation based on specific tags, vision, function, etc. In this research, we propose a web page block segmentation method from the view of HTML document generation. The server-side script loops search results from a database and dynamically generate each of them into HTML code fragments using the same HTML code fragment template. Therefore, by matching the approximate HTML code fragment in a HTML document, the blocks can probably be segmented.

The second task is event data identification. By observing web page block segmentation, the block is converted into text. However, with the exception of event data, many non-significant data also remain. This includes elements such as the logo, navigation bar, copyright information, etc. Therefore, it is necessary to identify event data from all the data. After doing this, the task converts the data into a text classification problem. We propose a semantic classifier to identity event data or non-event data based on machine learning. For higher dimensions such as word vector space model (more than 10,000 dimensions) classification, we adopt Support Vector Machine (SVM) to support this task.

In this paper, section II introduces some related works and analyses each of their applications. Section III shows the consideration and algorithm of web page block segmentation that we propose. Section IV implements event data identification using SVM. Section V evaluates experimental results of webpage block segmentation and event data identification.

[1]Nagoya Train Station: http://www.meieki.com/station_sa.php

Section VI is the conclusion.

II. Related Works

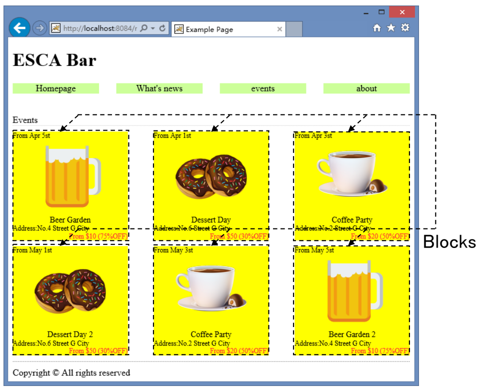
Web page block segmentation technique is used to segment a whole web page into multiple blocks that contain informative data. In the early researches[3][4], the informative block is segmented by observing some specified tags, such as $\langle table \rangle$ or $\langle H1 \rangle$ to $\langle H3 \rangle$. However, at that stage, many web pages were designed by a regular structure, such as table layout. Up until now, most web sites adopt div+css layout, which is more flexible. Web page blocks may be described by $\langle div \rangle$ or other common tags. The tag-based segmentation methods have been unable to adapt to current web site layouts.

Kovacevic[5] et al. propose a page-layout based web page block segmentation method in page layout: top(header), bottom(footer), left, right, center. This method may be effective for prim layout of most web pages. However, it cannot be used in all web pages. Especially, in the case of fashion, the homepage of a shop is most likely designed unconventionally. In this case, the page-layout based method is not useful.

Cai[6][7] et al. propose a vision-based page segmentation (VIPS). It simulates human vision to segment web page blocks, which distinguish different parts of the web page, such as lines, blanks, images, colors, etc. However, the result of VIPS is also a tree structure. It is still difficult to estimate the level of tree node in which the event data are. On the other hand, a list of titles in a div element may be estimated as one block by VIPS, rather than segmented into individual titles. Therefore, it is not a proper fit for event data extraction. To summarize, none of the above webpage block segmentation methods consider the web page block segmentation task from the view of web page generation.

From web page blocks segmentation, aside from event data, some non-significant data also remain. For this reason, an event data identification method is essential. Lan[8] et al. propose a method to eliminate non-significant data from web pages. The method compares several web pages in a web site. The non-significant data usually have approximate contents and presentation styles, such as navigation panels, copyright and privacy notices, in a website. However, the underlying condition is that web pages must be built applying the same template in the website. Also, the crawler has to crawl more than two web pages from a website. A single web page cannot be processed with this method. In this research, all the data besides event data are defined as non-significant data. Therefore, a semantic classifier may be the optimal selection for this task. The classifier is used to classify segmented text into event data or non-significant data.

For the text classification task, some of the most frequently used methods such as the Naive Bayes classifier[9]and the k-Nearest Neighbors(k-NN)[10] classifier are proposed. The Naive Bayes classifier measures the probability that a text belongs to event or not. The probability is constituted by factors that calculate weather each word in the text belongs to an event. Therefore, it is difficult to use the combination of words for precision improvement. The process time of Navie Bayes classifier is linear with respect to the size of the training set. If the training set is huge, the process time may be extremely long. K-NN measures cosine similarity between


http://localhost:8084/r
Example Page

ESCA Bar

Homepage What's news events about

Events

From Apr 5st
Beer Garden
Address:No.4 Street G City
From $10 (75%OFF)

From Apr 1st
Dessert Day
Address:No.6 Street G City
From $50 (30%OFF)

From Apr 3st
Coffee Party
Address:No.2 Street G City
From $20 (50%OFF)

From May 1st
Dessert Day 2
Address:No.6 Street G City
From $50 (30%OFF)

From May 3st
Coffee Party
Address:No.2 Street G City
From $20 (50%OFF)

From May 5st
Beer Garden 2
Address:No.4 Street G City
From $10 (75%OFF)

Copyright © All rights reserved

Fig. 1. A Web Page Sample

the target text and each of the tagged training texts. Then, it identifies the target text through the minimum cosine similarity tagged training text to determine which is event text. However, k-NN is lengthy if the training set is huge. In addition, k-NN has the characteristic of over-fitting under noise and error of the training set.

Many researches[11][12] validate the availability of Support Vector Machine(SVM) for text classification. From f1-measure results, the performance of SVM is higher than other classification models, such as Bayesian model, decision tree, neural network, etc. In particular, SVM shows higher performance when it processes higher dimension classifications, such as text classification (more than 10,000 dimensions) and avoids over-fitting more than other methods. In this research, we adopt SVM for event data identification.

III. Webpage Block Segmentation

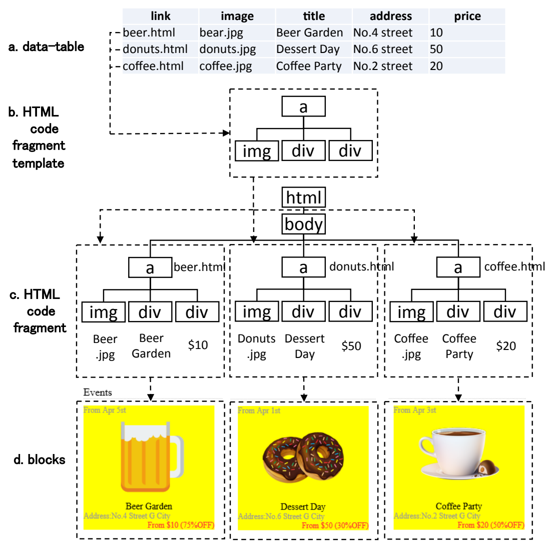
Event data blocks in a web page share a similar structure. As shown in Fig.1, there is a web page sample, which includes a banner, a navigation bar, a content (event) area and the copyright information. On the web page, each of the event data blocks (denoted by pointing arrows) has the date, image, title, address, price, etc. They are structured in a similar structure. Therefore, we propose an event data block extraction method through matching similar structures on the web page. In this section, we begin by explaining some terminologies. After that, we explain the consideration of our web block segmentation method. Lastly, we illustrate specific methods and algorithms.

HTML code fragment: An HTML code fragment is a part of an HTML document. Since an HTML document consists of a tree structure, an HTML code fragment may be a sub-tree in the HTML document.

Block: A block is an area of the web page. It is a HTML code fragment in an HTML document. It must contain informative information. In this research, a block of event data must contain the complete information of the event, such as title, notification, date, etc.

HTML code fragment template: An HTML code fragment template is a container. It is an HTML code fragment without text. It has the same DOM[2] structure as HTML code

[2]DOM: http://www.w3.org/DOM/



link	image	title	address	price
beer.html	bear.jpg	Beer Garden	No.4 street	10
donuts.html	donuts.jpg	Dessert Day	No.6 street	50
coffee.html	coffee.jpg	Coffee Party	No.2 street	20

Fig. 2. Example of Blocks Generation

fragments.

Dynamic web page: A dynamic web page is a web page whose construction is controlled by an application server processing server-side scripts, such as JSP, PHP, etc. It operates a database and dynamically generates the web page. When an user accesses a website in a browser, the server-side script analyzes the requested URL and its parameters. It searches results that the user requests from the database. When the database returns results, the server-side script applies those results into a predefined HTML code fragment template to generate them into HTML code fragments. Then, the server-side script groups these HTML code fragments into an HTML document. Finally, the server-side script pushes this HTML document to the user as a web page. Ordinarily, generated web page uses a extension, such as ".jsp" or ".php" etc. Recently, in consideration of user-friendly and search engine optimization, more and more server-side scripts generate a web page cache with the extension ".htm" or ".html", which is called analog static. From the view of user, the analog static web page seem like a static html web page. However, essentially, the analog static web page is belong to dynamic web page. Either dynamic web page or analog static web page, the server-side scripts structure a system named Content Management System(CMS)[13], such as the WordPress[3]. Almost the shop homepage is adopting CMS system. In our experiment, in 69 homepages, more than 66 of them are adopting CMS system.

Event data blocks sharing a similar structure can be explained from the view of block generation. Fig.2 shows a process of block generation. Data is structured and stored in a database as a data table. Each row includes informative information. When the CMS searches the data, each of them is applied to the same HTML code fragment template. They are generated into different blocks with the same DOM structure and shown on the web page. Therefore, some blocks in a webpage probably have similar DOM structure. It is possible to segment these blocks by matching DOM structure.

[3]WordPress: https://wordpress.org/

TABLE I. Web Blocks Segmentation Algorithm

	Web Blocks Segmentation Algorithm
1	initialize $Q = \{E_r\}, i = 0$
2	do $E_c = Q[i]$
3	$i = i + 1$
4	if $equal(E_c, E_c^L)$ or $equal(E_c, E_c^R)$
5	$output(E_c)$
6	else
7	$Q = Q + S_c$
8	until $Q[i] is null$

TABLE II. Function $equal(E_1, E_2)$

	Function $equal(E_1, E_2)$
1	String S_1 = $BreadthFirstSearch(E_1)$
2	String S_2 = $BreadthFirstSearch(E_2)$
3	Return $S_1 == S_2$

Since the HTML 2.0 standard, body elements are divided into block elements (block-level elements), such as $\langle P \rangle$, $\langle table \rangle$, and inline elements, such as $\langle a \rangle$, $\langle img \rangle$. Block elements normally start (and end) with a new line when displayed in a browser. They change the web page layout. Inline elements are normally displayed without line breaks. They do not impact the web page layout[4]. In this research, we only use block elements to match DOM structure.

The structure of an HTML document is a multi-tiered tree and an ordered tree. A current block element E in an HTML document is represented as $E = \{L, R, S\}$. $L = \{E_L\}$ and $R = \{E_R\}$ are the left and right brother elements of the current element E. $S_c = \{E_{S1}, E_{S2}, ..., E_{Sn}\}$ is a finite set of E ' s son elements. We propose the block segmenting from a large area to a small area. Therefore, the HTML document scan is top-down, from the root element to leaf elements. We obtain a breadth-first search to scan each E in body elements.

As shown in table I, for breadth-first search, we initialize a queue $Q = \{E_R\}$ with a single element E_R, which is the root element of the HTML document, and initialize a cursor i of Q from 0. Then set the current element E to $Q[i]$. The cursor i increases by 1. At row 4, the functions $equal(E, E_L)$ and $equal(E, E_R)$ have two parameters – the current element E and its left (and right) brother elements $E_L(E_R)$. It is used to compare two sub-tree DOM structures of E and E_L, as well as E and E_R. If the sub-tree DOM structures of E and E_L, or sub-tree structures of E and E_R are equal, the output current element $Q = \{E\}$ as a block; If not, all the son elements S of E are pushed into the tail of the queue Q. Loop to row 2, until $Q[i]$ is null.

Then we explain the function $equal(E_1, E_2)$. As shown in table II Function $equal(E_1, E_2)$ is used to compare the sub-tree DOM structures of the two elements E_1 and E_2. It scans each sub-element of E_1 and E_2 and converts them to string S_1 and S_2. If S_1 is equal to S_2, the function returns true; If not, the function returns false.

Up to now, all the blocks of the HTML document are outputted. However, besides event data, many non-significant data, such as the navigation bar, copyright information, etc., remain. Therefore, we identify the event data from the remaining data using machine learning.

[4]Block and Inline Elements: http://www.w3schools.com/html/html_blocks.asp

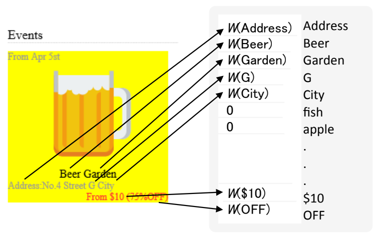


Fig. 3. Example of Word Vector Space Model Mapping

IV. Event Data Identification

Since many non-significant data remain in the segmented blocks, each one of extracted blocks should be classified using machine learning. In this research, the event data identification task converts to event data classification problem with two classes – event or non-event. We obtain SVM to implement event data classification. SVM[14] is a supervised machine learning model for classification and regression analysis. It employs high performance to solve small sample data and higher-dimensional text classification tasks[15].

First, because the input value of SVM is a row vector with a floating number, input text must be mapped to a word vector space model. For implementing word vector space model mapping, a dictionary is utilized. The dictionary includes all the words. Each of the words corresponds to a unique ID with the floating number. In this research, we use a dictionary called MeCab-IPADIC[16] for word vector space model mapping. Therefore, as shown in equation 1, the dimensionality n of word vector space model $\vec{V}$ is predefined to the dictionary $size(D)$.

$$\vec{V} = [w_1, w_2, ..., w_n] \qquad n = size(D) \tag{1}$$

As shown in equation 1, when a word w_i in the word vector space model appears in input text T, its value is set to 1.0; otherwise, the value is set to 0.0.

$$w_i = \begin{cases} 1.0 & w_i \in T \\ 0.0 & w_i \notin T \end{cases} \tag{2}$$

Fig.3 shows an example of word vector space model mapping. Words are set to 1.0 in a word vector space model if they appear in an event block, and others are set to 0.0. It should be noted that some weight technologies are recommended in related works such as $tf-idf$ etc. Soucy[17] et al. propose some weight technologies to improve classification accuracy. The availability of weight technology is proven when it solves multiclass text classification problems. However, in this research, the event data classification is a binary classification task. SVM can find support vectors of text instead of the idf method. Furthermore, because the length of input text is not long, The effect of tf is not necessary. Therefore, we do not adopt weight technologies in this research.

Additionally, different from English, the Japanese language is a non-space-separated language in which words are not explicitly denoted with a whitespace character. Therefore, a morphological analysis system is used to segment text into words. We use KUROMOJI[5] for Japanese morphological analysis.

[5]kuromoji: http://www.atilika.org/

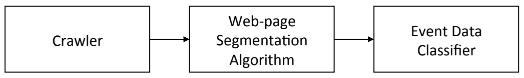


Fig. 4. Structure of Experiment System

Second, we implement a SVM classifier. We use an open source library SVM, LIBSVM[6], developed by Chang et al[18]. Two parameter SVM-formulation and kernel-function need to be set before using LIBSVM. SVM formulation is used to find the optimization separation hyperplane between a positive sample and negative sample. For classification tasks with balanced training samples, the C-SVM is the most frequently used SVM-formulation.

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i \\ & y_i(w^T\phi(x_i)+b) \geq 1-\xi_i, \\ & \xi_i \geq 0, i = 1,\ldots,l. \end{aligned} \tag{3}$$

C-SVM with a parameter C is a penalty factor for outliers. We do an optimal parameter search to decide it in section V. A kernel function is used to map sample vector to a higher-dimension. Because most text classification tasks are linearly separable, we set a linear kernel function for it as follows[15][19][20].

$$K(x_i, x_j) = x_i^T x_j \tag{4}$$

The linear kernel function does not map text vector space onto a higher-dimension. It is the fastest option.

V. Evaluation Experiment

In this section, we design an experimental system for evaluating the efficiency of a web block segmentation algorithm and event data identification. As shown in Fig.4, the experimental system is divided into three steps.

The crawler is developed by some existing sample in JAVA interfaces. It is used to download web pages from the Internet as HTML documents. Downloaded web pages are stored locally and sent to the web page block segmentation algorithm.

For web page block segmentation algorithm evaluation, we select the homepages of 96 shops as the experimental objects. The 96 shops are located in UNIMALL and ESCA, two underground shopping streets at a railway station of Nagoya city, Japan. We evaluate the recall rate using the following equation.

$$Recall = \frac{extracted\ number\ of\ event\ blocks}{actual\ number\ of\ event\ blocks} \tag{5}$$

The recall rate is proportionate to the extracted number of event blocks in actual number of event blocks. The results are shown in TABLE III. 69 homepages are available from the 96 shops The others are off-line or without event data. As a result, the recall rate is 82.14%. This result is considered acceptable for information recommendation. According to analyses of missed segmented samples, there are the following two reasons.

[6]LIBSVM: http://www.csie.ntu.edu.tw/ cjlin/libsvm/

TABLE III. RESULT OF WEB BLOCK SEGMENTATION

url	event blocks	CallBack
http://www.komeda.co.jp/ (50 Homepages) http://www.akakura.jp/	378	100.00%
http://www.n-rs.co.jp/	30	96.67%
http://www.peakmanager.com/pcweb/248	28	96.43%
http://www.kikuchi-megane.co.jp/index.html	11	90.91%
http://suppondo.co.jp/	10	90.00%
http://www.pokkacreate.co.jp/	913	89.05%
http://www.honeys.co.jp/	12	83.33%
http://www.hokennomadoguchi.com/	6	83.33%
http://www.world.co.jp/soup/	5	80.00%
http://www.ukiya.co.jp/top_sozai/top.htm	15	40.00%
http://www.hokennomadoguchi.com/	6	00.00%
https://www.facebook.com/sukikoto.unimall	3	00.00%
http://www.bodywork.co.jp/	5	00.00%
http://chronos.chicappa.jp/	6	0.00%
http://www.riochain.co.jp/	1	00.00%
http://www.r-p-s.net/	3	00.00%
http://www.uniqlo.com/jp/	1	00.00%
http://www.schiatti.co.jp/	1	00.00%
http://www.leilian.co.jp/	2	00.00%
http://www.erina-t.com/	6	00.00%
	average	82.14%

Asynchronous contents cannot be recognized by the crawler: Some shops show their Facebook or Twitter page on their homepage using frames, such as $\langle iframe \rangle$. Some homepages load data from servers by AJAX. As data are loaded in an asynchronous method after accessing the homepage, the current crawler is not able to obtain this asynchronous data. Some research proposes a GUI-Less Browser[21] to solve this problem. It simulates a browser process and returns a complete HTML document after data loading. In the future, we will use GUI-Less Browser to support AJAX and frames.

Anomalous HTML structure cannot be extracted: Some shops create homepages manually. This probably makes minute structural differences in each event web page block. On the other hand, some event HTML blocks have small differences, such as a "new marker" or a "more button". As we are using exact matching for comparing the HTML structure, the structure with even a small difference cannot be extracted. We will try similarity matching methods to support this problem in the future.

For event data classification algorithm evaluation, we manually filter 23,761 segmented block texts as a test sample. We evaluate the classification algorithm considering three performance indexes – Precision, Recall and F1-score as shown in the following equation.

$$\begin{aligned} Precision &= \frac{correctly\ classified\ number\ of\ event\ data}{classified\ number\ of\ event\ data} \\ Recall &= \frac{correctly\ classified\ number\ of\ event\ data}{actual\ number\ of\ event\ data} \\ F_1 &= 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \end{aligned} \tag{6}$$

We do an optimal parameter search to decide C in C-SVM. The value of C is looped from 2^{-5} to 2^{5} each $2^{\frac{1}{2}}$, and get the three performance indexes above. As shown in Fig.5, precision rate and recall rate are marked by the left Y-axis. The F1-score is marked by the right Y-axis The X-axis is i in $C = 2^{i}$. As

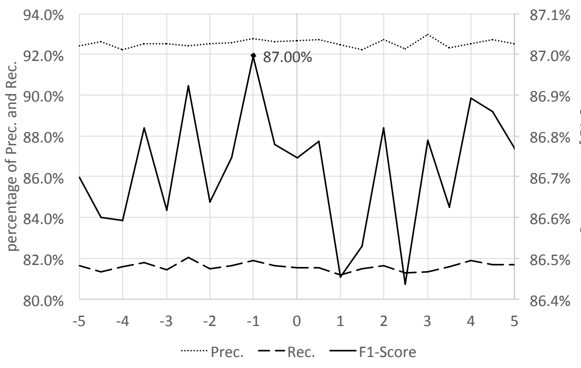


Fig. 5. Optimal Parameter Search for C

TABLE IV. RESULT OF OPTIMAL PARAMETER SEARCHING FOR C

C	Prec.	Rec.	F1
-5	92.41%	81.65%	86.70%
-4.5	92.61%	81.33%	86.60%
-4	92.23%	81.61%	86.59%
-3.5	92.53%	81.77%	86.82%
-3	92.53%	81.42%	86.62%
-2.5	92.40%	82.06%	86.92%
-2	92.49%	81.48%	86.64%
-1.5	92.55%	81.63%	86.75%
-1	92.79%	81.88%	**87.00%**
-0.5	92.62%	81.63%	86.78%
0	92.66%	81.54%	86.75%
0.5	92.74%	81.55%	86.79%
1	92.45%	81.19%	86.45%
1.5	92.23%	81.50%	86.53%
2	92.70%	81.64%	86.82%
2.5	92.29%	81.28%	86.44%
3	92.99%	81.36%	86.79%
3.5	92.30%	81.61%	86.63%
4	92.52%	81.91%	86.89%
4.5	92.72%	81.70%	86.86%
5	92.53%	81.69%	86.77%
5.5	92.43%	81.54%	86.64%

shown in TABLE IV, we find an optimal parameter when C is 2^{-1}.

Another experience evaluates relativity between the size of the test sample and the three performance indexes when C is set into 2^{-1}. We adopt cross validation that sets 90% as the training set and 10% as the test set. The numbers of test sample loops from 1,000 to 23,761 each 1,000. As shown in Fig.6, the X-axis is the size of test sample. F1-score increases with the size of test sample. TABLE V shows detailed data of the event data classifier evaluation. When the size of the test sample increases to 23,761, the F1-score reaches a maximum 86.997%. This is a higher accuracy result of text classification. It is proved to be an effective event data classifier. Otherwise, from Fig.6, we find the F1-score may continually increase with the size of the training set. In the future, we aim to implement some new methods to obtain more training data, using crowdsourcing, for example, and reduce the noise in the training data. On the other hand, the shop homepage owner may publish their events information in embed metadata, such as microdata, microformates, RDFa etc. In the future, we try to design a method to defined those active metadata that is higher availability.

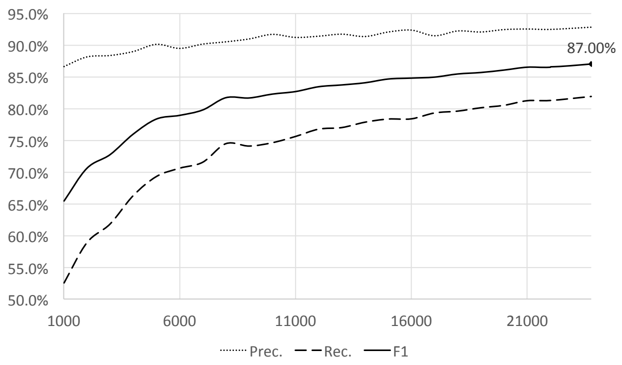


Fig. 6. Evaluation Result of Event Data Classifier

TABLE V. Evaluation Result of Event Data Classifier

Sample Size	Prec.	Rec.	F1
1000	86.60%	52.50%	65.37%
2000	88.11%	58.86%	70.57%
3000	88.33%	61.81%	72.73%
4000	88.96%	66.28%	75.96%
5000	90.09%	69.29%	78.33%
6000	89.45%	70.59%	78.91%
7000	90.13%	71.54%	79.76%
8000	90.49%	74.46%	81.70%
9000	90.94%	74.08%	81.65%
10000	91.66%	74.61%	82.26%
11000	91.18%	75.59%	82.66%
12000	91.37%	76.74%	83.42%
13000	91.69%	76.99%	83.70%
14000	91.32%	77.84%	84.04%
15000	92.03%	78.33%	84.63%
16000	92.33%	78.36%	84.78%
17000	91.44%	79.29%	84.93%
18000	92.20%	79.58%	85.43%
19000	92.03%	80.11%	85.66%
20000	92.41%	80.49%	86.04%
21000	92.50%	81.22%	86.49%
22000	92.44%	81.25%	86.49%
23000	93.01%	81.57%	86.91%
23761	92.79%	81.88%	**87.00%**

VI. Conclusion

In this research, we propose an event data extraction method that extracts business event data, such as coupons, tickets, sales campaign, etc., from the homepage or blog of shops and pushes them to users. Users no longer need to browse their favorite shops ’ homepage one by one. It supports comprehensiveness and effectiveness for event data obtainment. This proposition works into two tasks: web page block segmentation and event data identification. The first task is a method that segments the web page into blocks. Each of the blocks includes informative information, such as title, notification, date, etc. relating to event information. We propose a web page block segmentation method based on HTML document structure analysis. The second task is used to identity event data from the segmented blocks. We propose a method to implement event data identification based on machine learning. We show the results of the verification experiment. Experimental data are obtained from 96 shops located within two underground shopping streets – UNIMALL and ESCA, at a train station in the city of Nagoya (Japan). From the evaluation results of the web page block segmentation algorithm and event data classifier, we prove this method is an efficient method for event data extraction. Because the event data identification depends on the Japanese language, this method is available for all the Japanese home page.

References

[1] Chenyi L., Katsuhiko K., Kei H., Nobuo K.: Design and Implementation of Event Information Summarization System, CDS workshop of COMPSAC, 2014.

[2] Chenyi L., Katsuhiko K., Kei H., Nobuo K.: Development of Event Information Summarization System Based on SpatioTemporal-Information, DICOMO, 2014. (In Japanese)

[3] Lin, S.-H., Ho, J.-M., Discovering Informative Content Blocks from Web Documents, In Proceedings of ACM SIGKDD'02, 2002.

[4] Crivellari, F., Melucci, M., Web Document Retrieval Using Passage Retrieval, Connectivity Information, and Automatic Link Weighting–TREC-9 Report, In The Ninth Text REtrieval Conference (TREC 9), 2000.

[5] Kovacevic, Milos, et al. Recognition of Common Areas in a Web Page Using Visual Information: a possible application in a page classification. In: Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on. IEEE. p. 250-257, 2002.

[6] Cai D., Yu S., Wen J.-R., Ma W.-Y., VIPS: a visionbased page segmentation algorithm, Microsoft Technical Report, MSR-TR-2003-79, 2003.

[7] Cai, Deng, et al. Extracting content structure for web pages based on visual representation. In: Web Technologies and Applications. Springer Berlin Heidelberg. p. 406-417, 2003.

[8] Lan Y.; Bing L.; Xiaoli L.. Eliminating noisy information in web pages for data mining. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. p. 296-305, 2003.

[9] Lewis, David D. Naive (Bayes) at forty: The independence assumption in information retrieval. In: Machine learning: ECML-98. Springer Berlin Heidelberg. p. 4-15, 1998.

[10] Yang, Yiming; LIU, Xin. A re-examination of text categorization methods. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM. p. 42-49, 1999.

[11] Joachims, Thorsten. Text categorization with support vector machines: Learning with many relevant features. Springer Berlin Heidelberg, 1998.

[12] Serastiani, Fabrizio. Machine learning in automated text categorization. ACM computing surveys (CSUR), 34.1: 1-47, 2002.

[13] Rockley, Ann; Kostur, Pamela; MANNING, Steve. Managing enterprise content: A unified content strategy. New Riders, 2003.

[14] Cortes, Corinna; Vapnik, Vladimir. Support-vector networks. Machine learning, 20.3: 273-297, 1995.

[15] Joachims, Thorsten. Transductive inference for text classification using support vector machines. In: ICML. p. 200-209, 1999.

[16] Taku K., Kaoru Y., Yuji M.: Applying Conditional Random Fields to Japanese Morphological Analysis, Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004), pp.230-237, 2004.

[17] Soucy, Pascal; MINEAU, Guy W. Beyond TFIDF weighting for text categorization in the vector space model. In: IJCAI. pp. 1130-1135, 2005.

[18] Chih-Chung C.; Chih-Jen L.: LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2.3: 27, 2011.

[19] Huma L., Craig S., John Shawe-Taylor, Nello Cristianini, Chris Watkins: Text Classification using String Kernels, The Journal of Machine Learning Research, Volume 2, pp. 419-444, March 2002.

[20] Pilaszy, Istvn. Text categorization and support vector machines. In: The proceedings of the 6th international symposium of Hungarian researchers on computational intelligence. 2005.

[21] Bruns, Andreas; KORNSTADT, Andreas; WICHMANN, Dennis. Web application tests with selenium. Software, IEEE, 26.5: 88-91, 2009.