An Informative DOM Subtree Identification Method from Web Pages in Unfamiliar Web Sites

Masanobu TSURUTA, Hiroyuki SAKAI, Nonmembers, and Shigeru MASUYAMA, Member

SUMMARY We propose a method of informative DOM subtree identification from a Web page in an unfamiliar Web site. Our method uses layout data of DOM nodes generated by a generic Web browser. The results show that our method outperforms a baseline method, and was able to identify informative DOM subtrees from Web pages robustly.

key words: informative region identification, Web document, DOM, layout analysis

1. Introduction

The number of Web pages are dramatically increasing all over the world in recent years. In general, Web pages contain clutters, i.e., contents not related to the topic of the Web page. A screenshot of a typical Web page of a Japanese newspaper is shown in Fig. 1. In Fig. 1, some advertisements, link-lists and a common header region are clutters. On the other region, there are contents that related to the Web page’s topic. We call this region as an informative region. For making efficient use of Web pages, information retrieval and mining are important technologies. Identification of informative regions is positioned as an indispensable preprocessing for information retrieval and mining.

For this task, a number of algorithms have been proposed. Finn et al. proposed a method [1] that regards a Web page as a sequence of text and tag tokens, and extracts the informative content area by maximizing the number of text tokens in the body text area, and tag tokens in the navigation area. It is able to extract only text data and is not able to apply to a Web page like a photo gallery.

Debnath [2] proposed a method for extracting informative regions using a number of Web pages of a Web site. However, this method needs to crawl the unfamiliar Web site that includes the Web page, in order to extract informative regions. Crawling is too costly preprocessing for extracting informative regions of only one Web page.

Cai et al. [3] proposed a segmentation algorithm VIPS based on the layout of a Web page. VIPS uses some visual information of DOM nodes, i.e., the background color, font size, the size of DOM node and the various layout information, for segmentation of a Web page. Moreover, Song et al. [4] proposed a machine learning method for predicting the importance of blocks divided by VIPS. This method accurately predicts the importance of regions in Web pages. However, this method depends on segmentations of VIPS. If the segmentation was failed, this method would not work appropriately.

Gupta et al. [5] proposed a method for removing clutters from Web page. This method uses the URL blacklist of advertisements distributed on the Web and some heuristics for detecting link-lists. This method is not able to apply to undefined types of clutters, i.e., advertisement regions that includes a lot of non-anchor text.

We propose an informative DOM subtree identification method. Our method is able to identify an informative region of a Web page in an unfamiliar Web site, without crawling or segmentation. Moreover, our method is able to apply to a Web page like a photo gallery, and to eliminate undefined types of clutters since this method mainly uses positional information.

2. Proposed Method

Our proposed informative DOM subtree identification method uses the DOM tree generated by an actual Web browser and the positional information of DOM nodes in a window of the browser. Our method consists of two phases: generation of a weight-map and estimation of an informative DOM subtree of a Web page.

Manuscript received July 2, 2007.
Manuscript revised October 17, 2007.

The authors are with the Department of Knowledge-based Information Engineering, Toyohashi University of Technology, Toyohashi-shi, 441–8580 Japan.
a) E-mail: tsuruta@smlab.tutkie.tut.ac.jp
b) E-mail: sakai@smlab.tutkie.tut.ac.jp
c) E-mail: masuyama@tutkie.tut.ac.jp
DOI: 10.1093/ietisy/e91-d.4.986

*The Document Object Model is a standard interface that will allow programs to access the content, structure and style of documents. More information is at http://www.w3.org/DOM/
2.1 Generation of the Weight Map

A weight-map is generated from the answer set. Answer set $N_g$ consists of a number of pairs of Web pages and an informative region annotated by human annotators obeying the following rules: In a Web page, annotate a DOM node that completely includes the informative DOM subtree, and includes only indivisible non-informative DOM nodes. Note that the following steps are automatically executed.

**Step 1** Obtain the layout data of DOM nodes in a Web page from a rendered object tree produced by an actual Web browser. Obtain offset values of all DOM nodes from `nsIDOMNSHTMLElement`, or compatible interfaces, by opening the target Web page with a browser such as Firefox. Transform their relative coordinates to absolute ones.

**Step 2** Generate a weight-map. The weight-map is a grid over a window of a browser that has $x \times y$ cells, and each cell $c$ has the following weight $W(c)$.

$$W(c) = \frac{\sum_{d \in N_g} E(c, d)}{|N_g|}$$

(1)

Here, $N_g$ is the answer set defined in Step 1, for generating the weight-map, $d$ is a Web page in $N_g$. $E(c, d)$ is a score of $c$ in $d$.

Assignment of $E(c, d)$ is described in Fig. 2. $E(c, d) = \alpha$ when $c$ is covered only by a DOM node annotated as an answer by human annotator ("ANSWER"), $-1$ when $c$ is covered only by brothers or non-direct ancestors of ANSWER ("ERROR"), otherwise 0, where $\alpha$ is a parameter for balancing the weight-map. For example, in Fig. 3, when node A was ANSWER, nodes D, E and G are ERRORs of A.

2.2 Identification of an Informative DOM Subtree

As candidates of an objective DOM subtree, our method extracts the set of DOM nodes that are block-level elements and have a larger area than that of a cell from a rendered Web page. Then, $score(e)$ of each DOM node $e$ in $d$ is calculated by the weight-map. Here,

$$score(e) = \sum_{c \in C(e)} W(c)$$

(2)

where $C(e)$ is a set of cells in the weight-map covered by $e$. At last, the DOM node that maximizes $score(e)$, is identified as the root of the most informative DOM subtree. For example, in Fig. 4, $score(\Delta_1) = -3.5$, $score(\Delta_2) = 9$.

3. Evaluation by Experiments

Our goal is to design a method that extracts an informative region from a Web page without crawling the Web site to which the target Web page belongs. Our proposed method uses a weight-map that generated from an answer set that annotated by human annotators. We evaluated, by experiments, our method and a baseline method for the following three measures.

3.1 Measures

In the task of informative region identification, there are two significant criteria, to eliminate clutters and not to miss
some informative region. To examine whether the proposed method satisfies these criteria, we use three measures for evaluation; the accuracy Acc:

\[ \text{Acc} = \frac{|N_{\text{correct}}|}{|N_{\text{answer}}|}, \]

the average Cov of information coverage:

\[ \text{Cov} = \frac{\sum_{d \in N_{\text{descendant}}} \frac{I(e_d(d))}{I(e_a(d))}}{|N_{\text{descendant}}|}, \]

and the average Covn of clutter coverage:

\[ \text{Covn} = \frac{\sum_{d \in N_{\text{ancestor}}} \frac{I(e_d(d)) - I(e_a(d))}{I(d) - I(e_a(d))}}{|N_{\text{ancestor}}|}, \]

where,

\[ I(e) = \text{Char}(e) + 10 \cdot \text{Obj}(e), \]

Char(e) is the number of characters contained in the subtree of e. Obj(e) is the number of DOM nodes whose tagName's are IMG or EMBED in the subtree of e. A tagName is the attribute of a DOM node that returns the name of HTML element represented by the DOM node, i.e., IMG, EMBED, and DIV.

Here, e_d(d) is a DOM node estimated as the root of informative DOM subtree of d; e_a(d) is the DOM node annotated as the root of informative DOM subtree of d. N_{\text{answer}} is the answer set, N_{\text{correct}} is the set of Web pages when e_d(d) was an ANSWER, N_{\text{ancestor}} is the set of Web pages when e_d(d) was an ancestor node of ANSWER and N_{\text{descendant}} is the set of Web pages when e_d(d) was a descendant node of an ANSWER.

In Cov and Covn, we calculate the weighted number of displayed objects that are characters and images in a DOM node e by I(e). In Covn, I(d) indicates that of a root of the DOM tree generated by parsing of a Web page d.

Acc shows the fraction of the number of Web pages where evaluated method succeeds in identifying informative DOM subtree, in the answer set. If an estimated DOM subtree is the descendant of an answer node, we use Cov that is the proportion of the number of displayed objects in an estimated DOM subtree to that of the DOM subtree annotated as the informative region of Web pages. Meanwhile, if an estimated DOM subtree is the ancestor of an answer node, we use Covn, the average of the number of displayed objects in clutters in an estimated DOM subtree. The method that has higher Acc, Cov values and lower Covn value is the good one.

3.2 Answer Sets

As an answer set N_{\text{answer}} for generating a weight-map, we used the search results from Japanese social bookmark service Hatena::Bookmark that consists of 62 Japanese Web pages. Each of Web pages in N_{\text{answer}} has a DOM node annotated as the root of informative DOM subtree. Used queries are “animal” and “recipe.”

An answer set N_{\text{answer}} for evaluation consists of 191 annotated Web pages of search results from Hatena::Bookmark. Used queries are “music,” “lifehacks,” “business,” “政治 (politics),” “あとで読む (for later reading).” Note that since there was no Web site in both corpora, we removed some Web pages that we could not discover any informative regions (e.g., any top-pages of search engine.)

3.3 Parameters

In these experiments, we used GtkMozEmbed\(^\dagger\) as a browser. We set the window size of the browser to 800×960 pixels for minimizing body margins of the rendered Web page. The weight-map window size was set to 800×4000 pixels, and the size of a cell was set to 20×20 pixels. Constant \(\alpha\) was set to 0.7. \(\alpha\) was obtained by a closed test for the answer set A for generating a weight-map. Actual experiments use the weight-map generated by this \(\alpha\).

3.4 Baseline Method

We compared our method with a baseline method based on the ratio of HTML tags in a Web page\(^\dagger\dagger\). Although this baseline method is similar to that of Finn et al. [1], this method can extract an informative DOM subtree. This method extracts a DOM node e that maximizes Char(e)

\[ |N_{\text{correct}}| \quad 65 \quad 24 \]
\[ |N_{\text{descendant}}| \quad 36 \quad 90 \]
\[ |N_{\text{ancestor}}| \quad 92 \quad 56 \]
\[ |N_{\text{answer}}| \quad 0 \quad 21 \]

<table>
<thead>
<tr>
<th>Table 1 Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

\(^\dagger\)http://www.w3.org/TR/DOM-Level-2-Core/core.html#ID-745549614
\(^\dagger\dagger\)http://www.mozilla.org/unix/gtk-embedding.html
\(^\dagger\dagger\dagger\)http://blog.zuzara.com/2006/06/06/84/
from candidates that consist of the DOM nodes satisfying $\frac{\text{Tags}(e)}{\text{Char}(e)} \leq 0.1$ in a Web page. The extracted node $e$ is identified as the root of the informative DOM subtree of the Web page. Here, $\text{Tags}(e)$ is the number of HTML tags in $e$ represented by HTML.

3.5 Experimental Results

Experimental results are shown in Table 1, where, $N_{\text{error}}$ is the set of Web pages that the estimated node was an ERROR node. The generated weight-map trimmed to $40 \times 60$ cells is plotted in Fig. 5.

4. Discussions

In all three evaluation measures we considered, the proposed method outperformed the baseline. Its high Cov and low Covn values show the reliability in the task of an informative region. It is remarkable that our method did not estimate any ERROR DOM subtree as informative one. This indicates that our method was robust.

The baseline method estimates many descendants of ANSWER as informative DOM subtrees. Moreover, estimated DOM subtrees were small. These make value of Cov low. For example, the screenshot of an entry at Flickr.com is shown in Fig. 6. In this Web page, the baseline method was failed, because the amount of text in an informative region was less than general Web pages. However, our method identifies the picture and comments of users in an estimated region. Although our method also miss the right sidebar that has some metadata of the picture, the region was more informative than that estimated by the baseline method.

In the generated weight-map shown in Fig. 5, the brightest cells frequently covered by the informative region exists at central of the weight-map. The blocks that consists of darker cells are at the top, the far left and right. These dark blocks are covered by clutters that are banners of Web sites and navigation menus. On the other hand, there is not remarkable information at the bottom, because the heights of Web pages are all different. Hence, some DOM subtree estimated by the proposed method includes clutters at the bottom of the Web page.

The proposed method could fail to identify an informative DOM subtree of a Web page in the answer set. Because the proposed method strongly depends on layout data, we might use more features and apply some machine learning methods for estimating an informative DOM subtree to cope with this problem. However, this case does not happen in this experiment.

For future work, the normalization of positional information of DOM nodes may attain higher accuracy. The DOM subtrees estimated by our proposed method sometimes include clutter at the bottom because of using absolute coordinates of DOM nodes. However, simple normalization of coordinates might miss the information of clutters that locate at the top of Web pages.

5. Conclusions

We proposed an informative region identification method from a Web page in an unfamiliar Web site. Our proposed method estimates a DOM subtree as the informative region by using a weight-map generated by positional information of DOM subtree and the answer set annotated by a human annotator. Experimental results showed that our proposed method outperformed the baseline method based on the ratio of text and HTML tags.

Acknowledgment

This work was partially supported by Toyohashi University of Technology Global COE Program “Frontiers of Intelligent Sensing” and Grant-in-Aid “Scientific Research on Priority Areas: New Horizons in Computing” (B) (2) 16092213.

References