Detection of Layout-Purpose TABLE Tags Based on Machine Learning

Hidehiko Okada and Taiki Miura

Kyoto Sangyo University
Kamigamo Motoyama, Kita-ku, Kyoto 603-8555, Japan
hidehiko@ics.kyoto-su.ac.jp

Abstract. To make webpages more accessible to people with disabilities, <table> tags should not be used as a means to layout document content. Therefore, to evaluate the accessibility of webpages, it should be checked whether the pages include layout-purpose <table> tags. Automated precise detection of layout-purpose <table> tags in HTML sources is still a research challenge because it requires further than simply checking whether specific tags and/or attributes of the tags are included in the sources. We propose a method for the detection that is based on machine learning. The proposed method derives a <table> tag classifier that deduces the purpose of a <table> tag: the classifier deduces whether a <table> tag is a layout-purpose one or a table-purpose one. We have developed a system that derives classification rules by ID3. The system derives a decision tree from a set of learning data (<table> tags of which the purposes are known) and classifies <table> tags in webpages under evaluation. Classification accuracy was evaluated by cross validation with 200 test data collected from the Web. Result of the evaluation revealed that 1) the tags can be roughly classified with attribute values of border, number of rows, number of tags that appear ahead of the <table> tag, and the nest of <table> tags (i.e., these attributes are more likely to appear in upper layers in decision trees), and 2) the accuracy rates are about 90% for the 200 test data.

Keywords: Web accessibility, automated checking, <table> tags, machine learning, ID3.

1 Introduction

To make webpages more accessible to people with disabilities, <table> tags should not be used as a means to visually layout document content: layouting contents by <table> tags may present problems when rendering to non-visual media [1],[2]. It is reported that the number of tables on pages doubled from 7 in 2000 to 14 in 2003, with most tables being used to control page layouts [3]. Therefore, to evaluate the accessibility of webpages, it should be checked whether the pages include layout-purpose <table> tags. Several methods and tools have been proposed and developed for web accessibility [4]-[14]. Accessibility tools are listed in [13],[14] and compared in [7]. Some tools detect deeply nested <table> tags that are likely to be layout-purpose ones. Still, a method for automated detection of layout-purpose <table> tags

C. Stephanidis (Ed.): Universal Access in HCI, Part III, HCII 2007, LNCS 4556, pp. 116–123, 2007.

in HTML sources is a challenge: it requires further than simply checking whether specific tags and/or attributes of the tags are included in the sources.

We propose a method for the detection that is based on machine learning. The proposed method derives a <table> tag classifier that classifies the purpose of the tag: the classifier deduces whether a <table> tag is a layout-purpose one or a table-purpose one. We describe our system that implements the proposed method and report a result of experiment for evaluating classification accuracy.

2 Proposed Method

2.1 Basic Idea

<Table> tag is used for (1) expressing data tables as the tags are originally designed for or (2) adjusting the layout of document contents. In the case where a <table> tag is used for the table-purpose, the data in the same row or column are semantically related with each other (e.g., values of the same property for several data items, or values of several properties for the same data item) and the relation can be expressed by row/column headers. On the other hand, in the case where a <table> tag is used for the layout-purpose, the data in the same row or column may not be semantically related. Thus, to deduce the purpose of a <table> tag in a fundamental approach, it should be analyzed whether or not the data in the same row/column of the table are the semantically related ones. To make the analyses automated by a computer program requires a method for semantic analyses of table contents, but the automated semantic analyses with enough precision and independency for any kinds of webpages is hard to achieve.

Our basic idea focuses on machine-readable design attributes of a table instead of analyzing semantics of data in a table. If some common design pattern is found in some attribute values of layout-purpose <table> tags and another common design pattern is found in the attribute values of in table-purpose <table> tags, the two purposes can be discriminated by denoting classification rules based on the design patterns. However, it is unknown whether we can find such design patterns, and even if we can, it will be hard for us to manually investigate common patterns by analyzing design attribute values in a large number of <table> tag instances and denote sufficient rules to precisely classify the tags. In our research, we manually investigate design attributes only: to automatically derive classification rules, we utilize a machine learning method.

Among several kinds of machine learning methods available, we here utilize ID3 [15], a method for deriving a decision tree from a set of data instances. An advantage of a decision tree as a classifier over other forms of classifiers (e.g., a multi-layered neural net) is that classification rules are obtained as tree-formed explicit if-then rules and thus easy to read for human users of the method.

2.2 <Table> Tag Design Attributes for Classification Rules

We first collected and investigated <table> tag instances (webpage HTML sources including <table> tags) and extracted design attributes of <table> tag that seem to contribute to the classification. Webpages were collected from various categories in

Yahoo webpage directory so that the pages were not biased from the viewpoint of page categories. We manually judged the purpose of <table> tag instances in the collected pages. By this way, we collected 200 <table> tags a half of which were layout-purpose ones and the others table-purpose ones. We then extracted common design patterns for each set of <table> tags. The findings were as follows.

- Common design patterns in layout-purpose <table> tags
  - <Table> tags are nested.
  - Some cell(s) in the table include image(s).
  - The number of HTML tags that appear before the <table> tag in the source is small.
  - Some cells in the table are spanned.
  - The width and/or height are/is specified.
- Common design patterns in table-purpose <table> tags
  - The table has visible border lines.
  - The table includes many rows.
  - <Th> tags for row/column headers are included.
  - Table titles are specified.

By denoting classification rules based on these common design patterns, it will be possible to deduce the purposes of <table> tags to a certain extent. To derive the rules in a form of decision tree by ID3, we determined 10 attributes in Table 1.

Table 1. Attributes for Decision Tree

Name	Meaning	Values
border	Whether the table has visible border lines.	Binary
caption	Whether the table has a caption.	
height	Whether the table height is specified.	
img	Whether a cell in the table includes an item of image data.	
nest	Whether the table includes a nested table in itself.	
span	Whether some cells are spanned.	
th	Whether a row/column has a title header for the data in the row/column.	
width	Whether the table width is specified.	
num_tag	The number of HTML tags that appear before the <table> tags in the source.	Positive integer
num_tr	The number of rows in the table.	

- Binary values for the eight attributes of “border”-“width” mean whether or not the table has visible borders, captions, etc. For example, if the border attribute is specified for the <table> tag and the attribute value is 1 or more then border = Y, else border = N.
- A value of the attribute “num_tag” is the number of HTML tags that appear before the <table> tags in the source. In counting the tags, those in the header part are not counted.
- A value of the attribute “num_tr” is the number of rows (<tr> tags) in the table.

2.3 System Configuration for the Proposed Method

Fig. 1 shows the system configuration for our method. In the learning phase, the method derives rules for classifying <table> tags from a set of webpages in which layout/table-purpose <table> tags are included. First, for each <table> tag in the learning data, attribute values are obtained by analyzing the webpage HTML source in which the tag is included. The purpose of each tag in the learning data is given by manual judgments in collecting the learning data. From these data of attribute values and given purposes, classification rules are derived by a machine learning method. In the case where ID3 is applied as the learning method, the rules are obtained as a decision tree (i.e., tree-formed if-then rules). In the checking phase, <table> tags (of which the purpose is unknown) are classified by the rules obtained in the learning phase. Attribute values of each <table> tag for the classification are obtained by the same way as in the learning phase (the <table> attribute analyses module in Fig. 1 is shared by the learning and checking subsystems). The obtained attribute values are applied to if parts of the rules, and the purpose of each <table> is deduced by the rule of which the if part is satisfied.

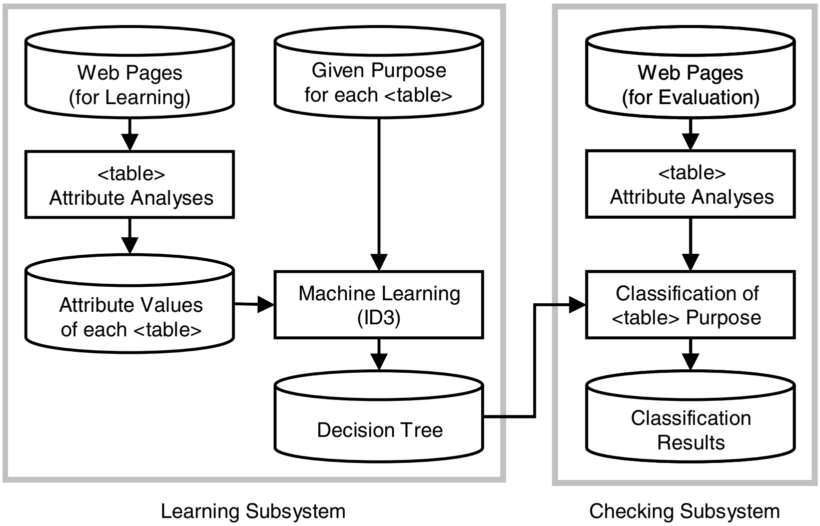

Fig. 1. System Configuration for the Proposed Method

3 Evaluation of the Proposed Method

3.1 Evaluation Method

To evaluate the effectiveness of our method, we investigate the classification accuracy by 10-fold cross validation (CV) with the 200 <table> tags we collected (see subsection 2.2). The 200 tags are randomly divided into 10 groups (G_1, G_2, …, G_{10}).

A decision tree is derived by ID3 applying to 180 <table> tags in 9 groups excluding G_i, and the classification accuracy is checked with 20 <table> tags in G_i (i=1,2,…,10). The accuracy rate is calculated as follows.

$$\text{Accuracy Rate} = (\text{Number of <table> tags correctly classified})/20 \qquad (1)$$

Ten values of the accuracy rate are obtained by a trial of 10-fold CV. We test the 10-fold CV three times and statistically investigate the accuracy rates.

3.2 Decision Trees Obtained by ID3

By the three trials of 10-fold CVs, 30 decision trees are obtained in total. Examples of the decision trees are shown in Tables 2-4 (only the nodes in the depth ≤ 3 are shown). Values in the "No." column are the serial numbers of the nodes where the #0 node is the root node. Tables 2-4 denote parent-child node relationships by indents in the "Rule Element" column. For example, in Table 2, the #1 and #8 nodes are child nodes of the #0 node and the #2 and #5 nodes are child nodes of the #1 node. Values in the "Rule Element" column are the conditions in if-parts of rules. Conditions that appear in a path from the root node to a leaf node are connected with AND. Values in the "Table" and "Layout" columns are the number of table/layout-purpose <table> tags in the learning data included in the node. For example, Table 2 shows the followings.

- The root node includes 89 table-purpose tags and 91 layout-purpose tags (see #0 node).
- Of the 180 <table> tags in the root node,
  - those with border=N are 86. Nine tags are table-purpose ones and the other 77 tags are layout-purpose ones (see #1 node), and
  - those with border=Y are 94. 80 tags are table-purpose ones and the other 14 tags are layout-purpose ones (see #8 node).

Table 2. Example (1) of Decision Trees Obtained in the Experiment

No.	Rule Element	Table	Layout	Total
0	(root)	89	91	180
1	border = N	9	77	86
2	num_tr ≤ 7	1	71	72
3	num_tag ≤ 12	0	66	66
4	num_tag > 12	1	5	6
5	num_tr > 7	8	6	14
6	nest = N	8	1	9
7	nest = Y	0	5	5
8	border = Y	80	14	94
9	nest = N	78	5	83
10	img = N	59	2	61
11	img = Y	19	3	22
12	nest = Y	2	9	11
13	height = N	0	5	5
14	height = Y	2	4	6

- Nodes of which the values in the “Table” column or the “Layout” column are leaf nodes. A leaf node corresponds to an if-then rule. For example, the #3 node denotes that all tags that meet the condition:
  - (border = N) and (num_tr ≤ 7) and (num_tag ≤ 12)
  - are layout-purpose ones. Thus, the following rule is obtained from the #3 node.
  - If (border = N) and (num_tr ≤ 7) and (num_tag ≤ 12) then the tag is a layout-purpose one.

Table 3. Example (2) of Decision Trees Obtained in the Experiment

No.	Rule Element	Table	Layout	Total
0	(root)	89	91	180
1	num_tr ≤6	15	82	97
2	border = N	3	69	72
3	img = N	3	18	21
4	img = Y	0	51	51
5	border = Y	12	13	25
6	nest = N	12	5	17
7	nest = Y	0	8	8
8	num_tr > 6	74	9	83
9	nest = N	72	1	73
10	border = N	9	1	10
11	border = Y	63	0	63
12	nest = Y	2	8	10
13	num_tag ≤7	0	8	8
14	num_tag > 7	2	0	2

Table 4. Example (3) of Decision Trees Obtained in the Experiment

No.	Rule Element	Table	Layout	Total
0	(root)	86	94	180
1	num_tr ≤6	15	84	99
2	num_tag ≤12	3	76	79
3	img = N	3	20	23
4	img = Y	0	56	56
5	num_tag > 12	12	8	20
6	nest = N	12	4	16
7	nest = Y	0	4	4
8	num_tr > 6	71	10	81
9	nest = N	69	1	70
10	border = N	10	1	11
11	border = Y	59	0	59
12	nest = Y	2	9	11
13	border = N	0	8	8
14	border = Y	2	1	3

The examples of decision trees shown in Tables 2-4 are typical ones in the 30 trees obtained. The other trees have similar structures from the root node to nodes in depth three as either of the three examples. The 30 trees reveal that the attributes border, num_tr, num_tag and nest is likely to appear in the upper layers of the tree, i.e., these attributes well contribute to the classification.

3.3 Evaluation of Classification Accuracy

Accuracy rates obtained by the three trials of the 10-fold CVs are shown in Table 5. Ten values of the accuracy rates are obtained by a single trial of CV, and the values in Table 5 are the minimum, maximum, mean and SD values of the ten accuracy rates for each trial. In all of the three trials, the maximum rate is 100% so that all checking <table> tags are correctly classified. The mean values are around 90% and the SD values are small, which supports the effectiveness of our method. Improvements for better values of the minimum accuracy rates are the further research challenges.

Table 5. Classification Accuracy Rates (%)

	1st CV	2nd CV	3rd CV
Min.	85	80	80
Max.	100	100	100
Mean	92	89	94
SD	6.0	7.5	6.7

4 Conclusion

In this paper, we proposed a method for detecting layout-purpose <table> tags in webpage HTML sources. The proposed method utilizes a machine learning method for deriving <table> tag classifiers. We have developed a system that utilizes ID3 as the machine learning method. The system derives a decision tree as the classifier from a set of <table> tag data for learning. Classification accuracy was evaluated by 10-fold CVs with 200 webpages collected from the Web. It is found that the purposes can be roughly discriminated with attributes of border, num_tr, num_tag and nest shown in Table 1: these attributes are likely to appear in upper layers in decision trees. In the experiment with the 200 <table> tags collected, mean accuracy rates were around 90%.

In our future work, we'll research whether machine learning methods other than ID3 (e.g., C4.5, multi-layered neural network, support vector machine) can improve the accuracy. These methods will be applied to our system and classification accuracy rates will be compared within the methods.

References

1. HTML 4.01 Specification. W3C Recommendation http://www.w3.org/TR/html4/
2. Web Content Accessibility Guidelines 1.0. W3C Recommendation http://www.w3.org/TR/WCAG10/
3. Ivory, M.Y., Megraw, R.: Evolution of Web Site Design Patterns. ACM Trans. on Information Systems 23(4), 463–497 (2005)
4. Abascal, J., Arrue, M., Fajardo, I., Garay, N., Tomas, J.: Use of Guidelines to Automatically Verify Web Accessibility. Universal Access in the Information Society 3(1), 71–79 (2004)

5. Cooper, M.: Evaluating Accessibility and Usability of Web Sites. In: Proc. of 3rd Int. Conf. on Computer-Aided Design of User Interfaces (CADUI'99), pp. 33-42 (1999)
6. Brinck, T., Hermann, D., Minnebo, B., Hakim, A.: AccessEnable: A Tool for Evaluating Compliance with Accessibility Standards. In: CHI'2002 Workshop on Automatically Evaluating the Usability of Web Sites (2002) http://simplytom.com/research/AccessEnable_workshop_paper.pdf
7. Brajnik, G.: Comparing Accessibility Evaluation Tools: a Method for Tool Effectiveness. Universal Access in the Information Society 3(3-4), 252–263 (2004)
8. Ivory, M.Y., Mankoff, J., Le, A.: Using Automated Tools to Improve Web Site Usage by Users With Diverse Abilities. IT&Society, vol.1(3), pp. 195–236 (2003) http://www.stanford.edu/group/siqss/itandsociety/v01i03/v01i03a11.pdf
9. Ivory, M.Y.: Automated Web Site Evaluation: Researcher's and Practitioner's Perspectives. Kluwer Academic Publishers, Norwell, MA, USA (2003)
10. Scapin, D., Leulier, C., Vanderdonckt, J., Mariage, C., Bastien, C., Farenc, C., Palanque, P., Bastide, R.: A Framework for Organizing Web Usability Guidelines. In: Proc. of the 6th Conf. on Human Factors & the Web (2000) http://www.isys.ucl.ac.be/bchi/publications/2000/Scapin-HFWeb2000.htm
11. Vanderdonckt, J., Beirekdar, A.: Automated Web Evaluation by Guideline Review. Journal of Web. Engineering 4(2), 102–117 (2005)
12. Beirekdar, A., Keita, M., Noirhomme, M., Randolet, F., Vanderdonckt, J., Mariage, C.: Flexible Reporting for Automated Usability and Accessibility Evaluation of Web Sites. In: Costabile, M.F., Paternó, F. (eds.) INTERACT 2005. LNCS, vol. 3585, pp. 281–294. Springer, Heidelberg (2005)
13. Complete List of Web Accessibility Evaluation Tools, W3C WAI http://www.w3.org/WAI/ER/tools/complete
14. TOOLS for Making Interfaces Accessible. in HCI Bibliography http://www.hcibib.org/accessibility/#TOOLS
15. Quinlan, J.R.: Induction of Decision Trees. Machine Learning 1, 81–106 (1986)