ICIC Express Letters
Part B: Applications
Volume **8**, Number **3**, March **2017**

ICIC International ©2017 ISSN 2185-2766
pp. **491–497**

MEASUREMENT OF WEB STRUCTURAL SIMILARITY BASED ON SUBTREE MATCHING

Shengnan Zhang, Kun Yang and Lianqiang Niu

School of Information Science and Engineering
Shenyang University of Technology
No. 111, West Shenliao Road, Shenyang 110870, P. R. China
zsncjr@sina.com; 1339774751@qq.com; niulq@sut.edu.cn

Received August 2016; accepted November 2016

Abstract. *Measurement of Web structural similarity is the key step in the clustering process of Web pages. This paper proposes a simple and efficient algorithm for measuring structural similarity in Web pages based on subtree matching. Firstly, according to a certain rule, Document Object Model (DOM) trees of Web pages are clipped so as to eliminate the redundant structural information. Secondly, by defining the rulers of non-repeated optimal matching and transformation, the similarity of subtrees is combined to the one of their root trees, in which the impact of the depth and the width of subtrees on the similarity is adequately considered. Experiments show that the proposed algorithm can distinguish similarity of Web structure more accurately and reasonably, and furthermore, improve the accuracy of Web pages clustering.*
Keywords: Structural similarity, Subtree matching, DOM tree, Web page clustering

1. **Introduction.** The massive amount of Web pages on the Internet provide enormous data resources, and how to make full use of the useful information contained in Web pages has always been a research hotspot. Since Web pages are semi-structured and can be designed flexibly, it brings the heterogeneity of Web pages. In numerous applications of Web information processing, the similarity measurement of Web structure is an important technical supporting, which is widely used in the information extraction [1-4], pattern extraction [5,6], search engine based on clustering [7], and so on.

[8] presented a method for measuring similarity of Web structure based on computing the tree edit distance, but it is not suitable for dealing with DOM trees of HTML with complex structures and nested relations and also has the high time complexity. In the works [9-11], the measurement algorithms, in which the structural similarity of trees were converted to compute the matching degree between the sets of tree paths, were discussed. Although their implementation is simple, the capacity of distinguishing Web pages with lower structural similarity is very poor; on the contrary, the simple tree matching algorithm proposed in [12] created similarity criterion by using the trees with same nodes in HTML document and had a higher distinguishing ability, but for the similar pages it cannot achieve the higher similarity. [13] combined the tags in each layer of DOM tree into a string at first and used it to calculate the edit distance between corresponding layers in two DOM trees, then the weighted sum of the distance of each layer was regarded as the structural similarity of two trees. However, the above methods do not accurately reflect the difference between trees, and they are a bit weak in precision. [14] calculated similarity by using repeated and optimal matching of subtrees, whose advantage is to avoid missing any match information and identify the similar Web pages with a higher accuracy, but it has the over-matching.

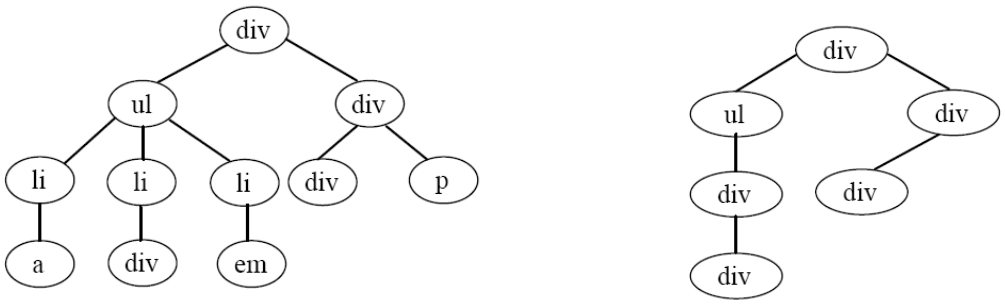
In order to better distinguish the Web pages with different similar degree, this paper proposes a measuring Web structural similarity algorithm that combines the classifying,

clipping of Web tags and the optimal matching of subtrees. Firstly, we classify the Web tags according to their influence on the Web structure, and then use a certain rule to clip the tags with less influence so as to improve the matching accuracy and efficiency of algorithm. Secondly, the clipped DOM tree with tags is optimally matched by a non-repeated way, and then the similarities between subtrees are converted to the ones between rooted trees, which makes the calculation of Web structural similarity more accurate and efficient.

2. Classification and Clipping of Web Tags.

2.1. Classification of Web tags.

Document Object Model (DOM) is a commonly used method that represents and processes HTML or XML document, its basic idea is to convert all the nodes in a document to a tree, which is known as DOM tree, and furthermore the relevant modification, calculation and information extraction can be done through DOM tree. For example, the structural similarity between two Web pages can be replaced by the one of two DOM trees. Figure 1(a) shows a simple DOM tree structure of Web page.



(a) Original tree (b) Clipped tree

FIGURE 1. DOM tree and clipped DOM tree

In the existing algorithms of measuring similarity based on tags, all tags have the same status. In fact, different from the general tree structure, the function of tags in DOM tree has large discrepancy, a large number of tags have little or no impact on the structure layout of Web pages, such as <head>, <script>, <li> and <a>. When the tags are used for structural similarity comparison without discriminating, it will lead to error matching between nodes with the same names, reduce importance of structural tags and affect the accuracy of the similarity. In consequence, we need to classify the tags according to its influence on Web structure.

Definition 2.1. *The tag, which cannot induce the structural change of Web pages and just marks the content is referred to as descriptive tag, i.e., Text Tag.*

Definition 2.2. *The tag that describes the Web structure is called structured tag, i.e., Block Tag. Every tag not belonging to the Text Tag is a Block Tag.*

Table 1 lists all of the Text Tags and part of the Block Tags.

By distinguishing different types of tags, we can clip the tags and its subtrees appropriately when calculating the similarity of web structure.

2.2. Clipping tags of Web pages.

In the actual DOM tree, all the Text Tags cannot be clipped unconditionally; this is because Block Tags may appear in the subtrees of Text Tags. For example, in Figure 1 is one of the child tags of <li>. Although <li> belongs to Text Tag, cutting <li> will lead to the loss of important structural information in this branch. Therefore, we need to define the corresponding clipping rule.

TABLE 1. Text tags and part of block tags

Text Tag	Block Tag
a abbr acronym address area aside b base basefont bdi bdo big br caption cite col colgroup dd del dfn dt dir em embed figcaption font h1-h6 head hr i img ins kbd keygen legend li link mark menuitem meta meter noframes noscript optgroup option output p param pre q rp rt s samp script small source span strike strong style sub summary sup td tfoot th thead time title tr track tt u var wbr	applet article audio body button canvas center code datalist div dl fieldset figure footer form frame frameset header html iframe section select table tbody textarea ul video

Definition 2.3. *If a tag belongs to Text Tag and there is no Block Tag in its subtrees, then the tag (or node) can be clipped, otherwise it is not.*

Definition 2.3 embodies the clipping rule of tag nodes.

Rule 1. *When traversing DOM tree, if tag A can be clipped, then it and its corresponding subtrees will be removed; otherwise it will not be processed.*

According to Rule 1, clipping algorithm of DOM tree can be implemented in two steps. First, annotate the nodes according to the Definition 2.3 when creating a DOM tree. Then, clip the DOM tree according to the annotation.

Figure 1(b) shows the clipped structure of DOM Tree in Figure 1(a), from which we can find that the clipped DOM tree not only maintains the structural information of Web page, but also eliminates the redundancy greatly, and highlights the importance of structural tags. In this paper, the clipped DOM is called DOM structure tree.

3. **Similarity Measurement of Web Structure.** Web pages based on the same template usually consist of multiple regional blocks, each region can be represented as a subtree, and the sequence of each subtree in the rooted tree is fixed. Since DOM structure tree maintains the characteristics of the original DOM tree, we can measure the similarity of Web structure by comparing the similarity of subtrees between DOM structure trees. In this paper, by defining the rulers of optimal matching and transformation, the similarity of subtrees is combined to the one of their root trees. In order to avoid over-matching, every node in arbitrary layer of DOM structure trees will not match with other nodes as they have got its optimal matching.

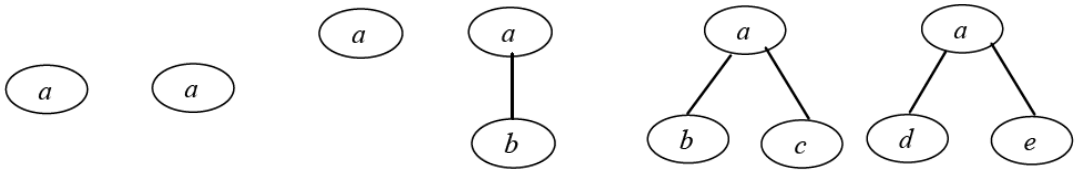
Definition 3.1. *Let non-empty rooted tree $T = (R, TC)$ be the DOM structure tree of a Web page. R is the root node of T, $TC = \{T_1, T_2, \ldots, T_n\}$ is the set of subtrees of T. For arbitrary non-empty rooted tree T_A and T_B, the similarity between them can be defined as follows:*

$$\mathrm{sim}(T_A, T_B) = \begin{cases} 0, & R_A \neq R_B \\ 1/(\max(\mathcal{D}_A, \mathcal{D}_B) \cdot \max(\mathcal{L}_A, \mathcal{L}_B)), & R_A = R_B \wedge (\tau = 0 \wedge \mathcal{C} = 0) \\ \mathrm{opt\_sim}(TC_A, TC_B), & R_A = R_B \wedge \tau > 0 \end{cases} \quad (1)$$

where $\mathcal{D}_\ell$ and $\mathcal{L}_\ell$ are the depth and the number of leaf nodes of tree T_ℓ respectively, $\ell = A, B$; $\mathcal{C} = \max(n_A, n_B)$, n_A and n_B are the number of child nodes in root node R_A and R_B respectively; τ is the number of matching pairs between TC_A and TC_B, $\tau = 0$ means that there is no matching subtree in the subtrees set, the similarities between all the subtrees are 0. Figure 2 shows three cases for $\tau = 0$.

Case (a) meets $\mathcal{C} = 0$, case (b) and case (c) meets $\tau = 0$. According to the second one of Formula (1), the similarity of two trees are 1, 1/2 and 1/4 respectively.

In Formula (1), if the root nodes of tree T_A and T_B are different, their similarity is 0. When the root nodes are same, $\mathcal{C} = 0$ means that at least two trees have no child nodes;



(a) No subtree (b) One having subtree (c) Each having subtrees

FIGURE 2. Three cases for $\tau = 0$

on this occasion, the operator $1/(\max(\mathcal{D}_A, \mathcal{D}_B) \cdot \max(\mathcal{L}_A, \mathcal{L}_B))$ ensures that the similarity of two trees is 1, and if one of them gets more child nodes and the structural difference between them gets bigger, a lower similarity can be obtained.

Function $\text{opt\_sim}(TC_A, TC_B)$ refers to implementing optimal matching between the subtrees sets TC_A and TC_B, and then converting the similarities of subtrees to the ones of the root trees based on a certain rule. The transformational rule is shown as Formula (2):

$$\begin{cases} \text{opt\_sim}(TC_A, TC_B) = \dfrac{1}{2}(\text{s\_d}(TC_A, TC_B) + \text{s\_w}(TC_A, TC_B)) \\ \text{s\_d}(TC_A, TC_B) = \dfrac{\sum_{i=0}^{n_A} \mathcal{D}_{Ai} \times \max_{j \in \{0,1,\ldots,n_B\} - \mathcal{N}_B} \text{sim}\,(TC_{Ai}, TC_{Bj})}{2 \times \max\left(\sum_{i=0}^{n_A} \mathcal{D}_{Ai}, \sum_{i=0}^{n_B} \mathcal{D}_{Bi}\right)} \\ \qquad + \dfrac{\sum_{j=0}^{n_B} \mathcal{D}_{Bj} \times \max_{i \in \{0,1,\ldots,n_A\} - \mathcal{N}_A} \text{sim}\,(TC_{Bj}, TC_{Ai})}{2 \times \max\left(\sum_{i=0}^{n_A} \mathcal{D}_{Ai}, \sum_{i=0}^{n_B} \mathcal{D}_{Bi}\right)} \\ \text{s\_w}(TC_A, TC_B) = \dfrac{\sum_{i=0}^{n_A} \mathcal{L}_{Ai} \times \max_{j \in \{0,1,\ldots,n_B\} - \mathcal{N}_B} \text{sim}\,(TC_{Ai}, TC_{Bj})}{2 \times \max\left(\sum_{i=0}^{n_A} \mathcal{L}_{Ai}, \sum_{i=0}^{n_B} \mathcal{L}_{Bi}\right)} \\ \qquad + \dfrac{\sum_{j=0}^{n_B} \mathcal{L}_{Bj} \times \max_{i \in \{0,1,\ldots,n_A\} - \mathcal{N}_A} \text{sim}\,(TC_{Bj}, TC_{Ai})}{2 \times \max\left(\sum_{i=0}^{n_A} \mathcal{L}_{Ai}, \sum_{i=0}^{n_B} \mathcal{L}_{Bi}\right)} \end{cases} \tag{2}$$

where $\mathcal{N}_A$ and $\mathcal{N}_B$ are subtree sets of TC_A and TC_B respectively, in which all subtrees have been matched with a subtree in the other set.

The optimal matching refers to selecting a pair of subtrees with maximum similarity from the subtrees not being matched optimally in two sets of subtrees in turn until all the subtrees in one set have been matched. Hence, each subtree has only one match tree at most.

The algorithm is described as follows:

Step1. Compare the root nodes of two trees. If they are different, the similarity of two trees is set to zero, then end the algorithm; Otherwise, enter the next step;

Step2. Count the number of child nodes of two trees respectively, and calculate the similarity between each pair of subtrees in two sets of subtrees. If both of two trees have no child nodes or no matching subtrees, the similarity of trees can be calculated by the operator $1/(\max(\mathcal{D}_A, \mathcal{D}_B) \cdot \max(\mathcal{L}_A, \mathcal{L}_B))$, then end the algorithm; Otherwise, enter the next step;

Step3. According to the optimal matching rule, obtain the optimal matching subtree for each subtree and take it as a reference of computing similarity between two trees, enter the next step;

Step4. Take the depth $\mathcal{D}_A$ and the number of leaf nodes $\mathcal{L}_A$ of subtrees as the weights when the similarities of subtrees are converted to the ones of root trees, then calculate and return the similarity of two trees based on Formula (2).

In the above transformation of similarity by the recursive process, the algorithm considers the influence of structural complexity of subtrees on the similarity, namely, the more complex the structure is, the greater the proportion of similarity is, contrarily, it gets much lower. $\mathcal{D}_A$ portrays the complexity of subtree's depth, and $\mathcal{L}_A$ reflects the complexity of its breadth. The algorithm considers the weighted value s_d(TC_A, TC_B) and s_w(TC_A, TC_B) of $\mathcal{D}_A$ and $\mathcal{L}_A$ as the features of subtrees, and then average them as the final similarity of two trees. For ensuring the similarity to satisfy the symmetry, two matches, namely from T_A to T_B and from T_B to T_A are carried out respectively in the calculation for two features.

The range of the similarity gained by Formula (1) is $[0, 1]$. The closer value is to 1, the higher similarity is, contrarily, the lower similarity is.

4. Experiment and Analysis.

In order to verify the validity of the proposed method, the corresponding algorithm is implemented with Java language program, and it is also compared with other related algorithms, which include matching algorithm based on traditional tree path model (TreePaths, [10]), simple tree matching algorithm (STM, [12]), and the algorithm based on optimal free matching of subtrees (OMF, [14]).

The test data set includes 500∗6 Web pages selected from Baidu Encyclopedia (hereafter referred to as B. E), Youth Network (hereafter referred to as Y. N), CSDN, Blog Garden (hereafter referred to as B. G), Pacific and Sina News (hereafter referred to as S. N). Web pages of different Web sites are generated by respective templates, which have obvious differences in structure and can be used to illustrate the applicability of the various algorithms.

4.1. Calculation and comparison of similarity.

In experiments, the similarity between DOM tree structures of six kinds of clipped sample Web pages was calculated, Table 2 shows the average similarity between different types of Web structures. Here, the specific values of our algorithm are given; due to the limit of space, only the numerical ranges of similarity of other algorithms (TreePaths, STM, and OMF) are demonstrated.

From Table 2, we can know that the proposed algorithm is more accurate than other methods in calculating the structural similarity of Web pages. In our method, for the Web pages with similar structure, the similarity is higher, on the contrary, the similarity of different categories is lower, and the reason is that our algorithm takes account of the difference of width and breadth between the trees. However, the similarity in [12] only reflects the ratio of the sum of subtrees similarity and the number of tree nodes in essence, which leads to the lower value in calculating the similar Web pages. In [10] and

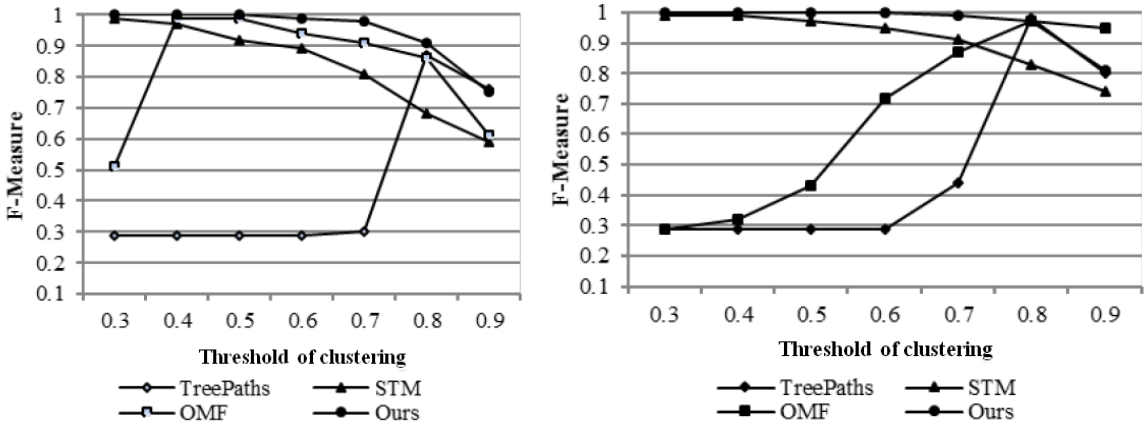
Table 2. Similarity of DOM trees calculated by different algorithms

	B. E		Y. N		CSDN		B. G		Pacific		S. N	
	Ours	Others	Ours	Others	Ours	Others	Ours	Others	Ours	Others	Ours	Others
B. E	0.86	0.7∼0.9	0.15	0.12∼0.71	0.08	0.10∼0.73	0.03	0.10∼0.73	0.14	0.11∼0.74	0.09	0.24∼0.74
Y. N	0.15	0.12∼0.71	0.99	0.99	0.08	0.15∼0.72	0.05	0.16∼0.70	0.20	0.11∼0.70	0.15	0.20∼0.73
CSDN	0.08	0.1∼0.73	0.08	0.15∼0.72	0.93	0.94∼0.99	0.23	0.27∼0.80	0.08	0.08∼0.72	0.10	0.17∼0.79
B. G	0.03	0.1∼0.73	0.05	0.16∼0.70	0.23	0.27∼0.80	0.90	0.90∼0.94	0.08	0.10∼0.74	0.07	0.15∼0.78
Pacific	0.14	0.11∼0.74	0.20	0.11∼0.70	0.08	0.08∼0.72	0.08	0.10∼0.74	0.91	0.6∼0.86	0.17	0.14∼0.74
S. N	0.09	0.24∼0.74	0.15	0.20∼0.73	0.10	0.17∼0.79	0.07	0.15∼0.78	0.17	0.14∼0.74	0.99	0.98∼0.99

[14], because of existing excessive matching, the similarity is high for different types of Web pages. Other experiments also illustrate that the similarity of the algorithms can be obviously improved by using clipped DOM tree.

4.2. **Comparison of Web clustering effect.** In order to verify the clustering effect of Web pages, one hundred pages were randomly selected from each of the above six categories Websites, the agglomerative hierarchical clustering was adopted for clustering sample pages, and the average distance between classes was used to measure the distance between clusters. The threshold range of the classes distance in our algorithm is set between 0.3 and 0.9, and the interval of threshold is 0.1. In addition, F-Measure was used as evaluation criteria of clustering effect, and the value of F-Measure is between 0 and 1, the larger the value is, the more accurate clustering is.

In the same way, this paper also gives the clustering experiments of non-clipped Web pages and clipped pages respectively. Figure 3 demonstrates the results.



(a) Clustering effect of non-clipped DOM tree (b) Clustering effect of clipped DOM tree

FIGURE 3. Comparison of clustering effect using different algorithms

From the above comparison charts, we know that F-Measure value under different threshold in our algorithm is higher than other three algorithms, and the resulting value is kept in a high range. In addition, after clipping DOM trees, we may improve the clustering effect within a certain threshold range.

5. **Conclusions.** Different from the similarity of general trees and graphs, Web pages not only contain similar structure but also a lot of non-structural elements. However, the non-structural information affects both the judgment of tree structure, statistics of nodes, and the matching accuracy of Web pages. The clipping rule presented in this paper can effectively eliminate the influence of non-structural information. At the same time, because of the great difference in structures of subtrees, with sole consideration of node number and path length, we just can simply distinguish between the similar or dissimilar Web pages and the accuracy of similarity calculation is poor. The proposed similarity criterion adequately considers the complete structure information described by the depth and width of trees; as a result, the Web pages with different similar degree of structure can be better distinguished; furthermore, the non-repeated optimal matching between subtrees presented by the algorithm also effectively avoid the excessive matching, and then give a more accurate similarity measurement.

REFERENCES

[1] D. C. Reis, P. B. Golgher, A. S. Silva et al., Automatic Web news extraction using tree edit distance, *Proc. of the 13th International Conference on World Wide Web*, New York, pp.502-511, 2004.

[2] C. Zheng, Y. Fu and L. Yu, Template-based information automatic extraction of Web, *Application Research of Computers*, vol.26, no.2, pp.570-572, 2009.
[3] Z. A. Chen and Z. Y. Zhou, Fast Web automatic text extraction algorithm based on template, *Application Research of Computers*, vol.26, no.7, pp.2646-2649, 2009.
[4] S. H. Yang, H. L. Lin and Y. B. Han, Automatic data extraction from template-generated Web pages, *Journal of Software*, vol.19, no.2, pp.209-223, 2008.
[5] C. Kim and K. Shim, Text: Automatic template extraction from heterogeneous Web pages, *IEEE Trans. Knowledge & Data Engineering*, vol.23, no.4, pp.612-626, 2011.
[6] J. Zheng, X. Wang and F. Li, Research on automatic generation of extraction patterns, *Journal of Chinese Information Processing*, vol.18, no.1, pp.48-54, 2004.
[7] Z. Lin, I. King and M. R. Lyu, PageSim: A novel link-based similarity measure for the World Wide Web, *Proc. of 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, Hong Kong, pp.687-693, 2006.
[8] K. C. Tai, The tree-to-tree correction problem, *Journal of the ACM*, vol.26, no.3, pp.422-433, 1979.
[9] S. Joshi, N. Agrawal, R. Krishnapuram et al., A bag of paths model for measuring structural similarity in Web documents, *Proc. of the 9th International Conference on Knowledge Discovery and Data Mining*, Washington, pp.577-582, 2003.
[10] H. Liao, Y. Yang, Z. Jia et al., An improved Web structural similarity based on matching algorithm of tree path, *Journal of Jilin University (Science Edition)*, vol.50, no.6, pp.1199-1203, 2012.
[11] Y. Wang, Z. Wang and F. Ye, Research of improved tree path model in Web page clustering, *Computer Science*, vol.42, no.5, pp.109-113, 2015.
[12] X. He and Z. Xie, Structural similarity measurement of Web pages based on simple tree matching algorithm, *Journal of Computer Research and Development*, no.z3, pp.1-6, 2007.
[13] R. Li, J. Zeng and S. Zhou, Improved Web page clustering algorithm based on partial tag tree matching, *Journal of Computer Applications*, vol.30, no.3, pp.818-820, 2010.
[14] M. Q. Song and R. X. Song, Research of structural similarity of Web pages based on HTML, *Journal of the China Society for Scientific and Technical Information*, vol.30, no.2, pp.160-165, 2011.