International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:1, No:8, 2007

Application of a Similarity Measure for Graphs to Web-based Document Structures

Matthias Dehmer, Frank Emmert Streib, Alexander Mehler, Jürgen Kilian and Max Mühlhäuser

***Abstract*— Due to the tremendous amount of information provided by the World Wide Web (WWW) developing methods for mining the structure of web-based documents is of considerable interest. In this paper we present a similarity measure for graphs representing web-based hypertext structures. Our similarity measure is mainly based on a novel representation of a graph as linear integer strings, whose components represent structural properties of the graph. The similarity of two graphs is then defined as the optimal alignment of the underlying property strings. In this paper we apply the well known technique of sequence alignments for solving a novel and challenging problem: Measuring the structural similarity of generalized trees. In other words: We first transform our graphs considered as high dimensional objects in linear structures. Then we derive similarity values from the alignments of the property strings in order to measure the structural similarity of generalized trees. Hence, we transform a graph similarity problem to a string similarity problem for developing a efficient graph similarity measure. We demonstrate that our similarity measure captures important structural information by applying it to two different test sets consisting of graphs representing web-based document structures.**

***Keywords*— Graph similarity, hierarchical and directed graphs, hypertext, generalized trees, web structure mining.**

I. Introduction

MEASURING the similarity between graphs is a challenging problem, especially for graphs of large orders the computation of similarity measures is quite involved [15], [20]. Many similarity measures of graphs are based on isomorphic relations and subgraph isomorphism [15], [20], respectively. Because it is well known that the subgraph isomorphism problem is NP-complete, the complexity of the underlying graph similarity measure is considered to be unacceptable for practical use [16]. Apart from measures that are based on isomorphic relations, there are methods in order to measure the structural similarity of trees. For example SELKOW [13] generalized the well known string edit distance [8], [14] to trees. Algorithmic improvements of tree alignment techniques for applications in Bioinformatics were proposed by, e.g., JIANG et al. [9] and ZHANG et al. [19]. But for the most graph similarity problems these methods are not suitable because the topology of the underlying graphs is more complex than the topology of trees. For this reason, we will present a new method for measuring the similarity of *labeled/unlabeled, hierarchical and directed graphs*. The term *hierarchical* means that there is an underlying directed rooted tree which forms a tree hierarchy. The main idea of our new method consists of two steps: (i) derivation of property strings from a graph and (ii) evaluating the alignment of the corresponding strings representing graphs. This is uniquely possible, because we restrict ourself to hierarchical graphs. In the following we call these graphs *generalized trees*. This graph class has been first introduced by MEHLER et al. [11] and is defined by

Definition 1.1: Let $\mathcal{H} = (\hat{V}, E_1)$ be an directed rooted tree. Let $m_{\hat{V}} : \hat{V} \longrightarrow A_{\hat{V}}$ be a vertex labeling function and $A_{\hat{V}}$ denote a vertex alphabet. The vertex set $\hat{V}$ is defined by

$$\hat{V} := \{v_{0,1}, v_{1,1}, v_{1,2}, \ldots, v_{1,\sigma_1}, v_{2,1}, v_{2,2}, \ldots, v_{2,\sigma_2}, \ldots, v_{h,1}, v_{h,2}, \ldots, v_{h,\sigma_h}\}$$

where $v_{i,j}$ denotes the j-th vertex on the i-th level, $0 \leq i \leq h$, $1 \leq j \leq \sigma_i$. h denotes the depth of $\mathcal{H}$ and σ_i is the number of vertices on level i. The edge set $\hat{E} := \hat{E}_1 \cup \hat{E}_2 \cup \hat{E}_3 \cup \hat{E}_4$ is defined as [11]:

- $(\hat{E}_1)$ forms the edge set of the underlying directed rooted tree $\mathcal{H}$.
- $(\hat{E}_2)$ *Up-edges* associate analogously nodes of the tree hierarchy with one of their (dominating) predecessor nodes.
- $(\hat{E}_3)$ *Down-edges* associate nodes of the tree hierarchy with one of their (dominated) successor nodes in terms of that tree hierarchy.
- $(\hat{E}_4)$ *Cross-edges* associate nodes of the tree hierarchy, none of which is an (immediate) predecessor of the other in terms of the tree hierarchy.

If $\hat{E}_2$, $\hat{E}_3$ and $\hat{E}_4 \neq \emptyset$ then $\hat{\mathcal{H}} = (\hat{V}, \hat{E}, m_{\hat{V}}, A_{\hat{V}})$ denotes a generalized tree. If we set $A_{\hat{V}} = \emptyset$, the generalized tree is unlabeled (see Fig. (1)).

For the comparison of hypertext graphs there are simple *graph theoretic indices* (see Section (II)) like *Multiplicity* defined by WINNE et al. [17]. Multiplicity is defined as the ratio of the edge cut set of two graphs to the number of all possible edges. Because of this definition it is obivous that Multiplicity is not suitable for a comparison of the overall structure of a graph. In contrast to those simple indices we design in Section (III) a powerful parametric model for measuring the structural similarity of web-based document structures representing generalized trees. We solve this graph similarity problem by first transforming the generalized trees into linear structures. Then we apply a dynamic programming [1] algorithm for determining optimal alignments of the corresponding property

Matthias Dehmer is with the Technische Universität Darmstadt, 64289 Darmstadt, Germany, e-mail: dehmer@informatik.tu-darmstadt.de. Frank Emmert-Streib is with the Stowers Institute for Medical Research, 1000 E. 50th Street, Kansas City, MO 64110, USA, e-mail: fes@stowers-institute.org. Alexander Mehler is with the Universität Bielefeld, 33501 Bielefeld, Germany, e-mail: Alexander.Mehler@uni-bielefeld.de. Jürgen Kilian is with the Technische Universität Darmstadt, 64289 Darmstadt, Germany, e-mail: kilian@noteserver.org. Max Mühlhäuser is with the Technische Universität Darmstadt, 64289 Darmstadt, Germany, e-mail: max@informatik.tu-darmstadt.de.

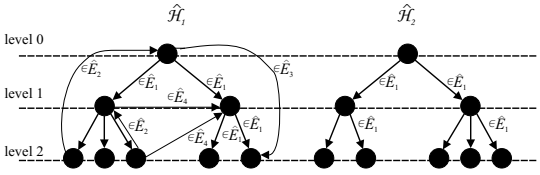

Fig. 1. $\hat{\mathcal{H}}_1$ shows a generalized tree and his edge types fulfilling Definition (1.1). In contrast $\hat{\mathcal{H}}_2$ represents an ordinary directed rooted tree, which consists only of (bold) edges $e \in \hat{E}_1$. An edge $e \in \hat{E}_1$ overjumps always just one level, $e \in \hat{E}_3$ overjumps at least one level, $e \in \hat{E}_4$ does not necessarily overjump a level.

strings. On the basis of the values of the resulting alignments we construct a graph similarty measure d.

This paper is organized as follows: In the next section we repeat some graph theoretic measures for the structural analysis of hypertext structures introduced so far in the context of web-mining. We present in Section (III) our similarity measure mathematically and apply it in Section (IV) together with agglomerative clustering to two different data sets, each one containing web-based document structures. This paper finishes in Section (V) with a summary and conclusions.

II. Graph Theoretic Measures for Structural Analysis of Hypertext Structures

In order to motivate our new method for measuring the similarity of generalized trees we first look at known measures for the structural analysis of hypertexts. We will see, that those measures are not suitable for a similarity-based structural analysis of graph-based documents, because they can not capture enough information that allows a meaningful clustering of web-based documents. In the field of structural analysis of hypertexts there are many measures that are called *indices* describing structural properties of hypertexts [2], [17]. The characteristic property of an index is that the described structural property, e.g., connectedness, is mapped on a normalised measured value. For example, Botafogo et al. [2] defined the well known graph theoretic measure *Compactness* of a directed hypertext graph $\mathcal{H}$ as

$$C := \frac{(|V|^2 - |V|) \cdot \mathcal{K} - \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} c_{ij}}{(|V|^2 - |V|) \cdot \mathcal{K} - (|V|^2 - |V|)} \in [0,1],$$

which expresses how well connected a hypertext graph is. Here,

$$(c_{ij})_{ij} := \begin{cases} w_{ij} & : \text{if } w_{ij} \text{ exists} \\ \mathcal{K} & : \text{else,} \end{cases}$$

denotes the converted distance matrix and w_{ij} denotes the shortest path from v_i to v_j. $\mathcal{K}$ defines the *conversion constant*, $|V|$ denotes the number of vertices of $\mathcal{H}$ and Botafogo et al. [2] set $\mathcal{K} = |V|$. From definition of C we conclude: $C = 1$ iff $\mathcal{H}$ is completely connected, $C = 0$ iff $\mathcal{H} = (V, \emptyset)$. Now, assume for two hypertext graphs $\mathcal{H}_1$, $\mathcal{H}_2$ holds $C_1 \approx C_2$. It is clear that the graph structures can be noticeable different. Therefore the index C is not suitable for determining intervals which contain similar hypertext structures. Hence, it is not possible to derive quality features, like "positive navigation behavior" from a certain value $C^\star \in [0,1]$.

In contrast to this, the unsupervised learning approach we suggest extends and improves the concept of graph theoretic indices mentioned significantly. Our new approach for analyzing web-based document structures consists of two steps:

1) Developing a method for measuring the similarity of generalized trees. Up to now there are no contributions in the area of the structural analysis of hypertext data that present results in terms of measuring the similarity of document structures by comparing the overall graph structure.
2) Application of multivariate data analysis methods, e.g., clustering methods.

There are already known approaches determining the similarity of web-based documents, e.g., [5], [10]. Cruz et al. [5] presented results about the structural similarity of DOM-trees [4] based on TDFA (*Tag Frequency Distribution Analysis*). Joshi et al. [10] proposed an approach for measuring the structural similarity of web-based documents representing DOM-trees on the basis of a *bag of path model*. However, our similarity measure is completely different from all these contributions, in the sense that it captures more structural information of a graph.

III. Similarity Measuring of Generalized Trees

In this section we design a new similarity measure d for measuring the structural similarity of generalized trees. In the following we consider the unlabeled case, because the transition to the labeled case is possible by minor modifications (see Section (IV)).

The main idea of this similarity measure is based on the derivation of property strings for each generalized tree and then to align the property strings representing our generalized trees by a *dynamic programming* technique [1] (see Fig. (2)). From the resulting alignment one obtains a value of the scoring function, which is minimized during the alignment process. The similarity of two generalized trees will be expressed by a cumulation of local similarity functions which weighs two types of alignments: *out-degree* and *in-degree* alignments on a generalized tree level. Since we are examining hierarchical graphs, we take a closer look at the out-degree and in-degree sequences (on a level i), induced by the vertex sequences $v_{i,1}, v_{i,2}, \ldots, v_{i,\sigma_i}$ and their edge relations (see Fig. (2)). Now, the more similar with respect to a *cost function* α the out-degree and in-degree alignments on the levels i are, the more similar is the common structure of the graphs. Define $r_k^{\hat{\mathcal{H}}^k} := v_{0,1}^{\hat{\mathcal{H}}^k}, k \in \{1,2\}$, and let $\hat{\mathcal{H}}^1$ be a given graph and $v_{i,j}^{\hat{\mathcal{H}}^1}$, $0 \le i \le h_1$, $1 \le j \le \sigma_i$ denote the j-th vertex on the i-th level of $\hat{\mathcal{H}}^1$, analogous to $v_{i,j}^{\hat{\mathcal{H}}^2}$ for $\hat{\mathcal{H}}^2$. Then the problem of determining the structural similarity between $\hat{\mathcal{H}}^1$ and $\hat{\mathcal{H}}^2$ is equivalent to computing the optimal alignment of

$$S_1 := r_1^{\hat{\mathcal{H}}^1} \circ v_{1,1}^{\hat{\mathcal{H}}^1} \circ v_{1,2}^{\hat{\mathcal{H}}^1} \circ \cdots \circ v_{h_1,\sigma_{h_1}}^{\hat{\mathcal{H}}^1}, \quad (1)$$

$$S_2 := r_2^{\hat{\mathcal{H}}^2} \circ v_{1,1}^{\hat{\mathcal{H}}^2} \circ v_{1,2}^{\hat{\mathcal{H}}^2} \circ \cdots \circ v_{h_2,\sigma_{h_2}}^{\hat{\mathcal{H}}^2}, \quad (2)$$

with respect to a *cost function* α which evaluates the alignments. Here, we distinguish different types of alignments:

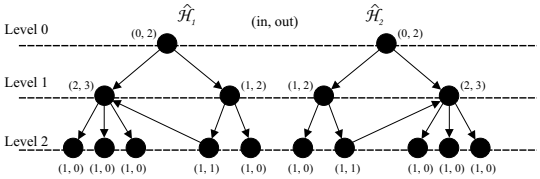

Fig. 2. Two generalized trees with their property strings. As an example the property string in terms of out-degrees of $\hat{\mathcal{H}}_1$ on level 1 equals "3 ∘ 2". The symbol ∘ denotes usual string concatenation.

(v, v) (vertex-vertex), $(-, v)$ (gap-vertex), and $(v, -)$ (vertex-gap). In order to determine the optimal alignment between two given graphs, we consider the Sequences (1), (2) where $S_k[i]$ denotes the i-th position of the sequence S_k and it holds $S_1[n] = v_{h_1,\sigma_{h_1}}^{\hat{\mathcal{H}}^1}, S_2[m] = v_{h_2,\sigma_{h_2}}^{\hat{\mathcal{H}}^2}$, $\mathbb{N} \ni n, m \geq 1$, $S_k[1] = r_k^{\hat{\mathcal{H}}_k}$, $k \in \{1, 2\}$. The algorithm with the complexity $O(|\hat{V}_1| \cdot |\hat{V}_2|)$ for finding the optimal alignment of S_1 and S_2 generates a matrix $(\mathcal{M}(i, j))_{ij}, 0 \leq i \leq n, 0 \leq j \leq m$. Now, we define the optimal alignment on the basis of the following dynamic programming algorithm [8]:

$$\begin{aligned} \mathcal{M}(0,0) &:= 0, \\ \mathcal{M}(i,0) &:= \mathcal{M}(i-1,0) + \alpha(S_1[i], -) \ : \ 1 \leq i \leq n, \\ \mathcal{M}(0,j) &:= \mathcal{M}(0,j-1) + \alpha(-, S_2[j]) \ : \ 1 \leq j \leq m, \end{aligned}$$

$$\mathcal{M}(i,j) := \min \begin{cases} \mathcal{M}(i-1,j) + \alpha(S_1[i], -) \\ \mathcal{M}(i,j-1) + \alpha(-, S_2[j]) \\ \mathcal{M}(i-1,j-1) + \alpha(S_1[i], S_2[j]), \end{cases}$$

for $1 \leq i \leq n$, $1 \leq j \leq m$. In order to evaluate the alignments on each level, we defined [6] the functions

$$\gamma^{out} = \gamma^{out}(i, \sigma_1^{out}, \sigma_2^{out})$$

and

$$\gamma^{in} = \gamma^{in}(i, \sigma_1^{in}, \sigma_2^{in}),$$

$\sigma_k^{out}, \sigma_k^{in} \in \mathbb{R}, k \in \{1, 2\}$ in an natural way and constructed a similarity measure $d \in [0, 1]$ (on the basis of these functions). $\gamma^{out}, \gamma^{in}$ are two-parametric functions, which detect the similarity of an outdegree and indegree alignment (on a level i). Finally, if we assume a set of units U and a mapping $\phi : U \times U \longrightarrow$ [0,1], we called ϕ a backward similarity measure if it satisfies the conditions

$$\phi(u, v) = \phi(v, u), \forall\, u, v \in U$$

and

$$\phi(u, u) \geq \phi(u, v), \forall\, u, v \in U.$$

Now we state the key result which has been proven in [6] for measuring the similarity for generalized trees.

Theorem 3.1: Let $\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2$, $0 \leq i \leq \rho$, $\rho := \max(h_1, h_2)$.

$$d(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2) := \frac{\prod_{i=0}^{\rho} \gamma^{fin}(i, \sigma_1^{out}, \sigma_2^{out}, \sigma_1^{in}, \sigma_2^{in})}{\frac{\sum_{i=0}^{\rho} \gamma^{fin}(i, \sigma_1^{out}, \sigma_2^{out}, \sigma_1^{in}, \sigma_2^{in})}{\rho+1}}, \quad (3)$$

is a backward similarity measure, where γ^{fin} is defined as

$$\begin{aligned} \gamma^{fin} &= \gamma^{fin}(i, \sigma_1^{out}, \sigma_2^{out}, \sigma_1^{in}, \sigma_2^{in}) \\ &:= \zeta \cdot \gamma^{out} + (1 - \zeta) \cdot \gamma^{in}, \quad \zeta \in [0, 1]. \end{aligned}$$

By construction we have $d(\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2) \in [0, 1]$. As a summary we note that our algorithm measures the similarity of two generalized trees by applying the technique of sequence alignments to outdegree and indegree sequences (on a level i). These alignments have both global and local significance. On the one hand, the sequence alignments will be implemented in a global sense, to compute the optimal alignment between the sequences S_1 and S_2. On the other hand, the alignments will be evaluated on the levels of the generalized trees by the function γ^{fin}. We note that the presented algorithm is suitable for the comparison of large generalized trees, because its complexity is enormously better then the complexity of methods which deal with isomorphic relations.

IV. Experimental Results

We applied our method to web-based document structures representing generalized trees and, hence, to a problem from web structure mining. To perform the evaluation, we used a corpus T_C which contains 500 conference websites from mathematics and computer science due to Mehler et al. [11]. Starting from conference calendar websites, Mehler et al. created the corpus with a java application that collects the conference links. Based on this set of links, we extracted the websites from the web by `HyGraph` [7]. Our evaluation is based on the two following steps:

1) Examining the cumulative similarity distribution of T_C on the basis of website structures, that is we represent a conference website $w \in T_C$ as a unlabeled generalized tree. This examination is impossible without a meaningful similarity measure which covers the complex graph structure of two graph-based documents. Thereby, the interpretation of the cumulative similarity distribution leads us to various applications.
2) Application of agglomerative clustering to the obtained similarity matrix where the web-based documents are represented by their `DOM`-structure [4]. Here, a `DOM`-tree is represented by a labeled generalized tree.

Definition 4.1: In terms of T_C we define data classes $D_1 - D_5$ which are manifest by the following parameter spectra:

- D_1: $\zeta = 1.0$ (solely alignments of Kernel-edges); parameter settings:
$$\sigma_{out}^1 = 1.0, \sigma_{out}^2 = 2.0, \sigma_{in}^1 = 1.0, \sigma_{in}^2 = 2.0.$$
- D_2: $\zeta = 0.3$; parameter settings:
$$\sigma_{out}^1 = 1.0, \sigma_{out}^2 = 1.0, \sigma_{in}^1 = 1.0, \sigma_{in}^2 = 1.0.$$
- D_3: $\zeta = 0.5$; parameter settings:
$$\sigma_{out}^1 = 1.0, \sigma_{out}^2 = 1.0, \sigma_{in}^1 = 1.0, \sigma_{in}^2 = 1.0.$$
- D_4: $\zeta = 0.5$; parameter settings:
$$\sigma_{out}^1 = 3.0, \sigma_{out}^2 = 3.0, \sigma_{in}^1 = 3.0, \sigma_{in}^2 = 3.0.$$
- D_5: $\zeta = 0.5$; parameter settings:
$$\sigma_{out}^1 = 2.0, \sigma_{out}^2 = 2.0, \sigma_{in}^1 = 2.0, \sigma_{in}^2 = 2.0.$$

In general the computation of the cumulative similarity distribution of a corpus of graph-based hypertext structures opens new perspectives, e.g.:

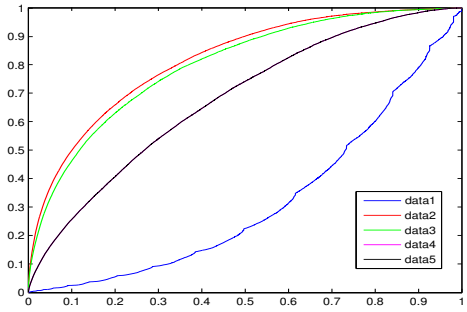

Fig. 3. The x-axes corresponds to the values of $d \in [0,1]$ and the y-axes represents the cumulative similarity distributions for D_1-D_5.

- The distribution in terms of structural similarity of graph-based hypertexts is unknown, because there is no efficient graph similarity measure available, which capture enough structural information for measuring the structural similarity. Our graph similarity measure d together with the cumulative similarity distribution provides a better understanding of web-based hypertexts and their interactions.
- Suppose we have a test corpus T of graph-based hypertexts from a specific web-genre, e.g., conference websites. MEHLER et al. described in [12] the interesting problem of computing a structural prototype of T by a graph median. As a preprocessing step for the *graph prototyping* [12] we can decide, on the basis of the cumulative similarity distribution, how structurally different the web-based document structures are.

Here, we compute the cumulative similarity distribution of T_C based on Definition (4.1) in order to examine the navigation strategies in terms of a web-genre. We assume that a generalized tree reflects all possible navigation paths of a graph-based conference website. The computation and interpretation of the cumulative similarity distribution of T_C leads us to the question how different the navigation strategies within the specific web-genre are. As example, we choose the web-genre of conference websites. In order to discuss the cumulative similarity distribution of T_C (see Fig. (3)) we note that the data classes D_1-D_5 are manifest by the same corpus T_C. We obtain a certain data class by only varying the parameters mentioned in Definition (4.1). Now, by varying the parameters we find the parameter tuple $(\zeta, \sigma^1_{out}, \sigma^2_{out}, \sigma^1_{in}, \sigma^2_{in})$ which captures enough structural information during the similarity measuring. In the following we notice that the plot of class D_1 differs in principle from the plots of data classes D_2-D_5. We recognize that, e.g., 20% of the conference websites have already the similarity value $d \leq 0.5$. Unlike 90% of the conference websites in D_2 have the similarity value $d \leq 0.5$. In summary, we conclude from Fig. (3), that the similarity values of the conference websites in D_1 were significantly higher compared to the conference websites of data classes D_2-D_5. This is plausible, because the conference websites in D_1 are treated solely as rooted trees without Cross-edges, Up-edges and Down-edges. Hence, the main part of the conference websites of D_1 is significantly less structurally different than

TABLE I
RESULTS OF PERFORMANCE EVALUATION OF THE CLUSTERING PROCEDURE CONCERNING W_1.

C_i	cluster	precision	recall
C_1	staff web pages	84%	99%
C_2	lecture announcements	76%	91%
C_3	summary web pages	65%	92%
C_4	lecture materials	85%	60%
C_5	Download pages	72%	84%

TABLE II
RESULTS OF PERFORMANCE EVALUATION OF THE CLUSTERING PROCEDURE CONCERNING W_2.

C_i	Cluster	precision	recall
C_1	Staff web pages	99%	99%
C_2	summary web pages	99%	99%
C_3	seminar announcements	75%	99%
C_4	lecture announcements	93%	93%
C_5	techn. doc. pages	50%	99%

the websites of the remaining data classes. In terms of D_2-D_5 the situation is inverted: In consideration of all types of conference websites the main part of the graph-based hypertext structures are structurally dissimilar on the basis of d. The plot of D_4 equals the plot of D_5. Finally we notice that for the data classes D_2-D_5 the main part of all possible navigation strategies are very different within our web-genre. This is reflected by psychological features of hypertext navigation, e.g.:

- Certain strategies of treatment
- Existence of previous knowledge
- Specific user preferences

The next application consists of an evaluation of a set of websites $\{W_1, W_2\}$[1] which are treated as sets of their DOM-structures [4]. Now we have to modify our similarity measure d because our generalized trees are vertex labeled by HTML-markups. We define in a similar way

$$\gamma^{fin}_m(i, \hat{\sigma}^1_{out}, \hat{\sigma}^2_{out}, \hat{\sigma}^1_{in}, \hat{\sigma}^2_{in}) := (1-\zeta_m) \cdot \gamma^{fin}(i, \hat{\sigma}^1_{out}, \hat{\sigma}^2_{out}, \hat{\sigma}^1_{in}, \hat{\sigma}^2_{in}) + \zeta_m \cdot \gamma_m(i, \sigma_m),$$

where $\zeta_m \in [0,1], \sigma_m \in \mathbb{R}$ and $\gamma_m(i, \sigma_m)$ state a cumulative function which detects the similarity of vertex markup alignment on a level i. With $\zeta_m = 0.5$ ($\sigma^1_{out} = 1.0, \sigma^2_{out} = 2.0, \sigma^1_{in} = 1.0, \sigma^2_{in} = 2.0$) and on the basis of a simple scheme which expresses the similarity relations between vertex labels, we apply an *agglomerative clustering method*[2] [3] to the obtained similarity matrices $(d_{ij})_{ij}$, $1 \leq i \leq |W_1|$, $1 \leq j \leq |W_1|$ and $(d_{ij})_{ij}$, $1 \leq i \leq |W_2|$, $1 \leq j \leq |W_2|$, $d_{ij} \in [0,1]$. $|W_1|$ and $|W_2|$ denote the number of extracted DOM-structures from W_1, W_2, respectively.

As a result we found cluster C_i which contain graph-based document structures where the C_i manifest structure

[1] http://www.algo.informatik.tu-darmstadt.de, http://www.sec.informatik.tu-darmstadt.de.

[2] We used Average Linkage [3].

types, e.g., staff web pages, and special structured template web pages. We evaluated the performance of clustering with the well know measures *recall* $= \frac{|M_r \cap M_g|}{|M_r|}$ and *precision* $= \frac{|M_r \cap M_g|}{|M_g|}$ [3]. Here, M_r denotes the number of all relevant documents and M_g the number of documents found, based on a Cluster C. The results in the Tables (I), (II) show clearly that the agglomerative clustering method created type classes, which contain structurally meaningful web pages. The high precision values express that the web pages found in a cluster C_i are actually relevant and the high recall values state that the web pages, which are relevant for Cluster C_i, were actually found. In comparison with the remaining precision values the cluster C_5 in Table (II) achieves a lower precision value. However, C_5 has a high recall value. That means, (i) some web pages found related for C_5 are not relevant and (ii) all relevant web pages for C_5 were found. Regarding to recall the situation for C_4 in Table (I) is reversed. Hence, we have shown that d is successfully applicable for the structural filtering of web pages.

V. Conclusions

We started pointing out the problems that graph theoretic indices have in capturing important structural information of graphs. For this reason we designed a new method to measure the structural similarity of generalized trees. Our similarity measure is based on the representation of generalized trees as linear strings. We call these strings property strings, because their components represent structural properties of the generalized trees. The similarity of two generalized trees is then defined as the optimal alignment of the corresponding property strings of the generalized trees. From this definition it is clear, that our similarity measure is also different from measures, which are based on isomorphic relations [15], [20]. We demonstrated in Section (IV) that the similarity measure d has important applications in web structure mining [3]. We have shown that the cumulative similarity distribution provides useful information about the test corpus T_C. On the basis of Definition (4.1) we answered the question how structurally different the conference websites of T_C are. In our experiment in Section (IV) the corpus T_C comprised 500 generalized trees and the average number of nodes per tree was 23. Because our similarity measure is a parametric measure depending on $(\zeta, \zeta_m, \sigma^1_{out}, \sigma^2_{out}, \sigma^1_{in}, \sigma^2_{in}, \hat{\sigma}^1_{out}, \hat{\sigma}^2_{out}, \hat{\sigma}^1_{in}, \hat{\sigma}^2_{in})$ we are able to emphasize different structure types of generalized trees during the evaluation of the alignment. For example, by setting $\zeta = 1$ we consider a unlabeled hypertext structure as a directed rooted tree. More precisely, we align only the out-degree property strings induced by edges from the underlying directed rooted tree. If we set $\zeta = 0$, we align the property strings induced by in-degree sequences only. In most of the cases we used $\zeta = \frac{1}{2}$, which weighs in- and out-degree sequences equally, $\gamma^{fin} = \frac{\gamma^{out}}{2} + \frac{\gamma^{in}}{2}$. In the case of labeled generalized trees we considered in Section (IV) websites as sets of their DOM-trees ($|W_1| + |W_2| = 270$). We applied agglomerative clustering to the obtained similarity matrix and found type classes, which contain structurally meaningful web pages. We evaluate the clusters found by comparing them with labels manually assigning to the web pages and find high recall and high precision values. That means web pages found in a cluster C_i are actually relevant and web pages which are relevant for cluster C_i were actually found. Hence, our similarity measure d is applicable to filter web pages according to their structural organization. The interpretation of our results from the filtering process is that similar DOM-trees tend to have a similar content and layout elements. Furthermore, clusters tell us something about the meaning of web pages. Altogether we have found document groups which are comparable on the basis of their structure types.

References

[1] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957
[2] R. A. Botafogo, B. Shneiderman: *Structural analysis of hypertexts: Identifying hierarchies and useful metrics*, ACM Trans. Inf. Syst. 10 (2), 1992, 142-180
[3] S. Chakrabarti: *Mining the Web. Discovering Knowledge from Hypertext Data*, Morgen and Kaufmann Publishers, 2003
[4] S. Chakrabarti: *Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction*, Proc. of the 10th International World Wide Web Conference, Hong Kong, 2001, 211-220
[5] I. F. Cruz, S. Borisov, M. A. Marks, T. R. Webb: *Measuring Structural Similarity Among Web Documents: Preliminary Results* , Lecture Notes In Computer Science, Vol. 1375, 1998
[6] M. Dehmer, *Strukturelle Analyse web-basierter Dokumente*, Ph.D Thesis, Department of Computer Science, Technische Universität Darmstadt, 2005, unpublished
[7] R. Gleim: *HyGraph – Ein Framework zur Extraktion, Repräsentation und Analyse webbasierter Hypertextstrukturen*, Beiträge zur GLDV-Tagung 2005, Bonn/Germany, 2005
[8] D. Gusfield: *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*, Cambridge University Press, 1997
[9] T. Jiang, L. Wang, K. Zhang: *Alignment of trees - An alternative to tree edit*, Theoretical Computer Science, Elsevier, Vol. 143, 1995, 137-148
[10] S. Joshi, N. Agrawal, R. Krishnapuram, S. Negi,: *Bag of Paths Model for Measuring Structural Similarity in Web Documents*, Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2003, 577-582.
[11] A. Mehler, M. Dehmer, R. Gleim: *Towards logical hypertext structure. A graph-theoretic perspective*, Proc. of I2CS'04, Guadalajara/Mexico, Lecture Notes in Computer Science, Berlin-New York: Springer, 2004
[12] A. Mehler, R. Gleim, M. Dehmer: *Towards structure-sensitive hypertext categorization*, to appear in: Proceedings of the 29-th Annual Conference of the German Classification Society, 2005
[13] S. M. Selkow: *The tree-to-tree editing problem*, Information Processing Letters, Vol. 6 (6), 1977, 184-186
[14] T. F. Smith, M. S. Waterman: *Identification of common molecular subsequences*, Journal of Molecular Biology, Vol. 147 (1), 1981, 195-197
[15] F. Sobik, *Graphmetriken und Klassifikation strukturierter Objekte*, ZKI-Informationen, Akad. Wiss. DDR, Vol. 2 (82), 1982, 63–122
[16] J. R. Ullman, *An algorithm for subgraph isomorphism*, J. ACM, Vol. 23 (1), 1976, 31–42
[17] P. H. Winne., L. Gupta, J. C. Nesbit: *Exploring individual differences in studying strategies using graph theoretic statistics*, The Alberta Journal of Educational Research, Vol. 40, 177-193, 1994
[18] A. Winter: *Exchanching Graphs with GXL*, http://www.gupro.de/GXL
[19] K. Zhang, D. Shasha: *Simple fast algorithms for the editing distance between trees and related problems*, SIAM Journal of Computing, Vol. 18 (6), 1989, 1245-1262
[20] B. Zelinka, *On a certain distance between isomorphism classes of graphs*, Časopis pro pěst. Mathematiky, Vol. 100, 1975, 371–373