Chapter 11
Mining Graph Patterns in Web-Based Systems: A Conceptual View

Matthias Dehmer and Frank Emmert-Streib

11.1 Introduction

The task of applying Data Mining methods [38] to web-based hypertexts is often referred to as Web Mining [16]. In view of the steadily increasing complexity of web data sources and the huge amount of information available online, Web Mining has been an important and fruitful research topic [16, 46]. Generally, Web Mining can be divided into the following categories:

1. Web Content Mining: Web Content Mining provides methods for automatically extracting information from web-based data sources. Important problems are data extraction and analysis by using, e.g., Text Mining methods [53].
2. Web Structure Mining: Web Structure Mining deals with exploring structural properties of web-based hypertexts, e.g., investigating internal and external link structures of web-based documents [16] or exploring hypertext structure types using graph-based models [55]. Moreover, there are a lot of earlier contributions rooted in complex network theory [29] dealing with analyzing mathematical growth-properties of the web graph and web subgraphs by using stochastic models [1, 34, 40, 48, 63]. Often, these methods aim to improve web-search and information extraction algorithms in Web Mining [14, 45].
3. Web Usage Mining: Web Usage Mining [73] deals with exploring and analyzing patterns reworked from web logs to analyze behavior of hypertext users. Such an analysis can be in particular useful to optimize business websites, to analyze their quality and to detect effectiveness features, see, for example [64].

In this chapter, we put the emphasis on discussing methods (in the context of Web Structure Mining) to analyze graph-based hypertext patterns. To tackle our problem, we discuss a graph-theoretic framework for exploring graph-based patterns representing web-based hypertext structures. Besides modeling document structures as

M. Dehmer (✉)
Institute of Discrete Mathematics and Geometry, Vienna University of Technology, Vienna, Austria; Institute for Bioinformatics and Translational Research, Hall in Tyrol, Austria
e-mail: matthias.dehmer@univie.ac.at; mdehmer@geometrie.tuwien.ac.at; Matthias.Dehmer@umit.at

A. Mehler et al. (eds.), *Genres on the Web*, Text, Speech and Language Technology 42, DOI 10.1007/978-90-481-9178-9_11,

graphs [57] that means that in the sense of consistent graph similarity measuring, we apply a method to measure the structural similarity of graphs (see Section (11.4.2)) to approach problems in Web Structure Mining, for example:

1. Computing the cumulative similarity distribution Θ of a web genre [50] corpus containing graph-based documents (see Section 11.5). A possible interpretation of Θ addresses the important question how structurally distributed the graph-based documents in the given corpus are.
2. Structural filtering of web-based units: By measuring the structural similarity of the document structures and then applying clustering techniques, we obtain clusters which contain structurally similar web-based units.

The main contribution of this conceptual chapter is to shed light on the task of automatically analyzing web genre data by using a method for structurally comparing graph-based hypertexts [18, 22, 27]. We use the term "web genre" and "web genre data" in the sense of Mehler et al. [59] where web genres are considered as hypertext types, see, e.g. [59, 65]. Also, we want to emphasize that we do not use the vector space model [31, 52] to represent a web-based document structure [18, 24]. Instead, we use a special graph class called generalized trees (GTs) [25, 57] for modeling our web-based documents [57].

Basically, Mehler focuses on webgenres not from the point of view of a bag-of-features model [56]. Rather, this approach conceives instances of webgenres as complex signs that have a characteristic structure due to their membership to a certain genre. This contrasts genre modeling with topic modeling in Information Retrieval [2] where a topic is represented by a set of lexical units that are typically used to manifest that topic. Rather, Mehler's approach is linguistic in the sense that instances of a certain text type are seen to have a characteristic topical structure *and* a characteristic generic structure. Take the example of a newspaper article in contrast to, say, a personal letter: although in both cases the universe of topics is certainly open, we can nevertheless expect that instances of both types depart with respect to the topical areas they typically deal with. Moreover, the differences between these text types are also manifested in structural terms: the structure of a letter significantly differs from that of most newspaper articles. *So why not exploring text structure [28], document structure [62] or even layout structure [75] to get insights into the webgenre (or hypertext type) of a webpage or of a website?*

Interestingly, many webgenre models oversee this structural source of the characteristics of webgenres. Consequently, they tend to rely on some extension or simply on some application of the bag-of-features or vector space model. However, such an approach disregards a central characteristic of web units as instances of webgenres, that is, their hyperlink structure, which is genuine web-based. From this point of view, a website is seen to be identifiable as an instance of a webgenre by means of its hypertextual structure – beyond its textual structure. Mehler et al. [59] have shown that because of many aspects of informational uncertainty this hypertextual structure is – by analogy to its textual counterpart – not immediately accessible: neither can we simply read-out this structure from HTML tags or URLs, nor is it manifested by hyperlinks only. Rather, this *hidden* hypertext document structure needs first to be explored as this is done with its counterpart in the form of document structure [62].

In this paper, we propose a structural approach of webgenres and webgenre classification that builds upon a webgenre-related hypertext structure model. More specifically, we utilize a certain graph model (in the form of generalized trees) that has been found to be the structural kernel of many complex linguistic aggregates [54]. Our task is to add a computational model that deals with this class of graphs as a model of webgenre structure. In this sense, we propose an algorithmic model that integrates a recent structural model of linguistic units by example of webgenres with their computational processing.

The graph similarity-based approach we want to discuss in this chapter operates on generalized trees representing hierarchical and directed graphs. We notice that generalized trees are more general than ordinary rooted trees because a generalized tree contains an ordinary rooted tree as a special case. For practical applications, this implies that a generalized tree captures more structural information of the underlying document structure than an usual DOM-tree [15] represented by a directed rooted tree. The classical DOM-tree model has been also applied for measuring the structural similarity of underlying hypertext structures by [13, 42].

The chapter is organized as follows: Section 11.2 presents some mathematical preliminaries. In Section 11.3, we briefly discuss the problem of deriving structural properties of graphs to characterize them structurally. Besides outlining existing methods for measuring the similarity of web-based document structures in Section 11.4, this section also discusses a graph similarity-method that operates on generalized trees. In Section 11.5, we outline resulting applications in Web Structure Mining and Web Usage Mining. The chapter finishes with a short summary in Section 11.6.

11.2 Mathematical Preliminaries

First, we introduce some mathematical preliminaries [25, 37, 39].

Definition 1 $G = (V, E), |V| < \infty, E \subseteq \binom{V}{2}$ is called a finite undirected graph. $G = (V, E), |V| < \infty, E \subseteq V \times V$ represents a finite directed graph.

Definition 2 Let $G = (V, E)$ be a graph. $\tilde{G} = (\tilde{V}, \tilde{E})$ is called a subgraph iff $\tilde{V} \subseteq V$ and $\tilde{E} \subseteq E$. Moreover, if it holds $\tilde{E} = E \cap (\tilde{V} \times \tilde{V})$, then we call $\tilde{G}$ the induced subgraph of G.

Definition 3 An isomorphism class denotes the set of graphs which are isomorphic to a given graph G.

Definition 4 A tree is a connected, acyclic undirected graph. A tree $T = (V, E)$ with a distinguished vertex $r \in V$ is a rooted tree. r is called the root of the tree. The level of a vertex v in a rooted tree T equals the length of the path from r to v. The maximum path length d from the root r to any vertex in the tree is called the depth of T. A leaf is a vertex incident to exactly one edge in a tree.

Definition 5 Let $G = (V, E)$ be a finite, directed graph. Then, we define the following sets and quantities:

$$\begin{aligned}\mathcal{N}^{+}(v) &= \{w \in V \backslash \{v\} \,|\, (v, w) \in E\},\\ \mathcal{N}^{-}(v) &= \{w \in V \backslash \{v\} \,|\, (w, v) \in E\},\\ \delta_{\text{out}}(v) &= |\mathcal{N}^{+}(v)|,\\ \delta_{\text{in}}(v) &= |\mathcal{N}^{-}(v)|.\end{aligned}$$

We call $\delta_{\text{out}}(v)$ and $\delta_{\text{in}}(v)$ out-degree and in-degree of $v \in V$, respectively.

Definition 6 A directed acyclic graph T is called a *directed rooted tree* if there is an unique vertex r satisfying $\delta_{\text{in}}(r) = 0$ from which any other vertex of T is reachable by a unique path.

Definition 7 Let $T = (V, E_1)$ be a directed rooted tree. The vertex set is defined by

$$V := \{v_{0,1}, v_{1,1}, v_{1,2}, \ldots, v_{1,|V_1|}, v_{2,1}, v_{2,2}, \ldots, v_{2,|V_2|}, \ldots, v_{d,1}, v_{d,2}, \ldots, v_{d,|V_d|}\}, \tag{11.1}$$

and we assume $|V| < \infty$. $|L|$ denotes the cardinality of the level set $L = \{l_0, l_1, \ldots, l_d\}$. The surjective mapping $\mathcal{L} : V \longrightarrow L$ is called a multi level function that assigns to every vertex an element of the level set L. It holds $d = |L| - 1$. $v_{i,j}$ denotes the j-th vertex on the i-th level, $0 \leq i \leq d$, $1 \leq j \leq |V_i|$. $|V_i|$ denotes the number of vertices on level i. The edge set $E_{GT} := E_1 \cup E_2 \cup E_3 \cup E_4$ of a finite generalized tree $H = (V, E_{GT})$ is defined as [57]:

- E_1 forms the edge set of the underlying directed rooted tree T. These edges are called Kernel-edges.
- E_2: *Up-edges* associate analogously vertices of the tree hierarchy with one of their (dominating) predecessor vertices.
- E_3: *Down-edges* associate vertices of the tree hierarchy with one of their (dominated) successor vertices in terms of that tree hierarchy.
- E_4: *Across-edges* associate vertices of the tree hierarchy, none of which is an (immediate) predecessor of the other in terms of the tree hierarchy.

Figure 11.1 shows a generalized tree exemplarily.

Definition 8 We define some metrical properties of graphs. $d(u, v)$ denotes the distance between $u \in V$ and $v \in V$ representing the minimum length of a

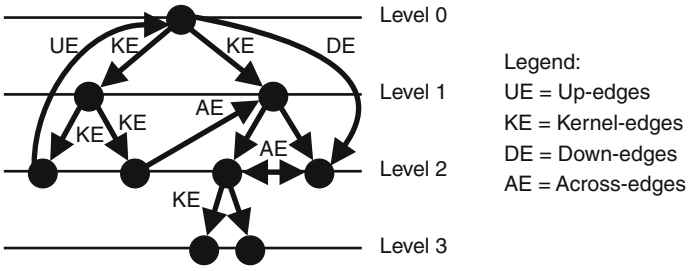

Fig. 11.1 A generalized tree with its edge types

path between u, v. Note that $d(u, v)$ is an integer metric. We call the quantity $\sigma(v) = \max_{u \in V} d(u, v)$ the eccentricity of $v \in V$. $\rho(G) = \max_{v \in V} \sigma(v)$ and $r(G) = \min_{v \in V} \sigma(v)$ is called the diameter and radius of G, respectively.

11.3 Structural Graph Measures

Graphs can be considered as powerful and generic models to describe complex relational objects which appear in a large number of scientific areas, e.g., computer science, chemistry, sociology, cognitive sciences and biology [17, 33, 76]. Apart from using graphs for modeling real world problems, an important problem is also to quantify structural information by inferring structural properties of a graph in question. This problem addresses the task of characterizing graphs based on graph measures. To give a short overview on such structural network measures, we present the listing as follows:

1. Degree distributions $P(i)$, e.g., see [29].
2. Exponent of degree distributions, i.e., it holds $P(i) \sim i^{-\gamma}$, e.g., see [29].
3. Total number of vertices $|V|$ and edges $|E|$.
4. Distance matrix $(d(v_i, v_j))_{v_i, v_j \in V}$.
5. Metrical properties of graphs, e.g., $\sigma(v)$, $\rho(G)$ and $r(G)$, e.g., see [70].
6. Clustering coefficient, modularity and network motifs, e.g., see [3, 8].
7. Vertex centrality measures, e.g., see [9, 51, 76].
8. Eigenvector measures, e.g., see [47, 51].

Another method to characterize graphs is based on quantifying structural information using information-theoretic measures. This problem relates to determine the structural complexity of a graph. Entropic measures to determine the so-called structural information content of a graph have been developed by [7, 6, 19, 20, 30]. A task that is also related to determine structural features of graphs is to identify stylistic properties. For example, a stylistic property can be understood as a characteristic structural feature of a graph that manifests a graph class, e.g., a hierarchy, an undirected edge set, a directed edge set etc. To identify such features exemplarily, we consider Fig. 11.2. The depicted graphs from different application domains manifest different styles of graphs. More precisely, graph (A) represents a directed rooted tree to model a DOM-structure. Graph (B) shows a more complex website structure representing a generalized tree. Graph (C) is a chemical structure represented by an undirected and vertex labeled graph. A different definition of a style that aims to compare such styles structurally (this lead to a generalization of the classical graph similarity problem [26]) has been already expressed in [26]. In [26], a style was defined as a set of graphs with impressed structural properties. Finally, we compared the styles by using a method which is based on the definition of a median graph [26, 58].

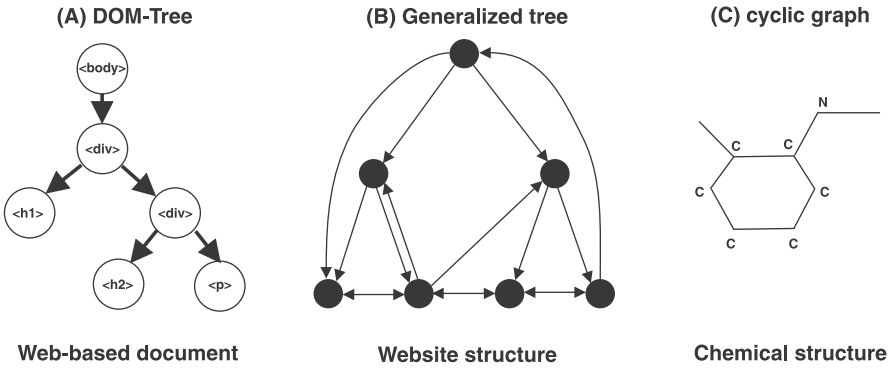

Fig. 11.2 Graph styles from different application domains

11.4 Graph Similarity Measures for Web Mining

11.4.1 Classical Similarity and Distance Measures for Graphs

The problem of measuring the similarity (or distance) between structures representing networks occur in numerous scientific disciplines [5, 13, 22, 68]. Usually, graph similarity measures are based on incorporating structural features of given graphs, e.g., degree sequences, subgraphs, and other metrical properties of graphs [70]. Also, the task of measuring the structural similarity of graphs is often referred to as *graph matching* [12]. There exist basically two major paradigms for matching graphs structurally which have been intensely discussed in the scientific literature: *exact graph matching* and *inexact graph matching* [12].

Exact graph matching is mainly based on the principle of finding a graph or a subgraph of a given graph that matches a graph or subgraph structure of an other graph exactly. With other words, one has to determine if two graphs are isomorphic [39], i.e., structurally equivalent. It is known that even classical graph similarity measures belonging to the exact graph matching paradigm are based on determining isomorphic and subgraph isomorphic relations, see, e.g., [43, 71, 72, 77]. A prominent example of a classical graph metric represents the well-known Zelinka-distance [77]; two graphs are more similar, the bigger the common induced (isomorphic) subgraph is. This implies that graphs which have a large common induced subgraph have a small distance and vice versa. It is worth mentioning that Zelinka [77] was the first who introduced such a measure for unlabeled graphs of same order. The key result is as follows [71, 72, 77].

Theorem 1 *Let* $H = (V_H, E_H)$ *and* $G = (V_G, E_G)$ *be unlabeled graphs without reflexive and multiple edges and it holds* $|V_H| = |V_G| = n$. $\overline{SUB}_m(H)$ *denotes the set of induced subgraphs of order* m. $H^\star$ *denotes the isomorphism classes of such graphs in which* H *lies and let*

$$SUB_m(H) := \{H^\star | \, H \in \overline{SUB}_m(H)\}. \tag{11.2}$$

$SUB_m(H)$ is just the set of isomorphism classes in which the induced subgraphs of H with order m lie. Then,

$$d_Z(H, G) := n - SIM(H, G), \tag{11.3}$$

is a graph metric, where

$$SIM(H, G) := \max\{m | SUB_m(H) \cap SUB_m(G) \neq \emptyset\}. \tag{11.4}$$

A more general version of this theorem was introduced by Sobik [71, 72]. The following assertion states that the measure $d_S(H, G)$ for determining the structural similarity of arbitrary and also labeled graphs represents a graph metric.

Theorem 2 *Let $H := (V, E, f_V, f_E, A_V, A_E)$ be a finite and labeled graph. A_V, A_E denote finite, non-empty vertex and edge alphabets and $f_V : V \to A_V$, $f_E : E \to A_E$ the associated vertex and edge labeling functions. Now, let H and G be finite, labeled graphs of arbitrary orders, respectively. Then,*

$$d_S(H, G) := \max\{|H|, |G|\} - SIM(H, G)\} \tag{11.5}$$

is a graph metric.

Now, we want to briefly discuss inexact graph matching. The most prominent measure from inexact graph matching is the so-called *graph edit distance* (GED) developed by Bunke [10]. It can be considered as a powerful extension of the Levenshtein-distance [49]. GED is mainly based on the idea to define graph edit operations such as insertion or deletion of an edge/vertex or relabeling of a vertex along with costs associated with each such operation [10]. Moreover, Bunke [10] calls an optimal inexact match a sequence of edit operations which transforms a graph G into H by producing minimal transformation costs. If $m_1, m_2, \ldots, m_n$ are assumed to be all possible transformations mapping G to H, then the optimal inexact match [10] m' is defined by

$$c(m') = \min\{c(m_i) | \, 1 \leq i \leq n\}. \tag{11.6}$$

Finally, the graph edit distance between two graphs is the minimum cost associated with a sequence of edit operations. Further, the optimal error-correcting graph isomorphism is defined as the resulting isomorphism after obtaining this optimal sequence of edit operations [10]. The original result of Bunke [10] can be now expressed as follows.

Theorem 3 *Let $d(H, G)$ be the costs for determining the optimal inexact match between H and G. Then, $d(H, G)$ is a graph metric.*

Many other graph similarity or distance measures and methods can be found in, e.g. [4, 17, 44, 60, 67, 71, 72].

11.4.2 Graph Similarity Measures Based on Trees

In this section, we outline graph similarity measures applied to web-based document structures. As follows, we express a listing of graph similarity measures which have been applied to DOM-trees [13]:

1. Similarity measures which are based on tree edit measures, e.g., see [41, 69, 74].
2. Similarity measures based on the frequency of tag labels, e.g., see [13].
3. Similarity measures based on Fourier transformation, e.g., see [32].
4. Similarity measures based on path similarity, e.g., see [42].

A major problem of these measures is that they only operate on ordinary rooted trees which do not capture the structural information properly represented by a complex hyperlink structure associated to a graph-based document. Especially the measures based on tag frequencies, see, e.g., [13] are restrictively interpretable because a rearrangement of the tag order does not necessarily imply a variation of the corresponding similarity measure. Moreover, the sketched measures do not provide the option to emphasize certain structural properties when measuring the structural similarity of graphs because the measures are non-parameterized. In contrast, parameterized similarity measures would give us the possibility to learn the parameters by using appropriate data sets. In Section 11.4.3, we express the definition of such a parameterized measure for determining the structural similarity of generalized trees. An in-depth treatment of graph similarity measures can be found in [11, 12, 18, 22].

11.4.3 Structural Similarity of Generalized Trees

This section aims to repeat the construction principle of a method for measuring the structural similarity of generalized trees, see, e.g. [18, 22, 27]. The main construction steps can be stated as follows [18, 22, 27]:

- We start with two generalized trees, H_1 and H_2.
- Derive their formal string representations and transform them into linear integer strings which are called property strings.
- Perform string alignments of the derived property strings by using a dynamic programming (DP) algorithm. From each such alignment (on each level i), a similarity score will be obtained.
- By cumulating up the derived similarity scores, a final graph similarity measure can be obtained. Hence, the problem of comparing two generalized trees structurally is then equivalent with determining optimal property string alignments.

These key steps are also visualized in Fig. 11.3. We start repeating the construction by stating some definitions [18, 21, 22].

Definition 9 Let X be a set. A positive function $s : X \times X \longrightarrow [0, 1]$ is called similarity measure if

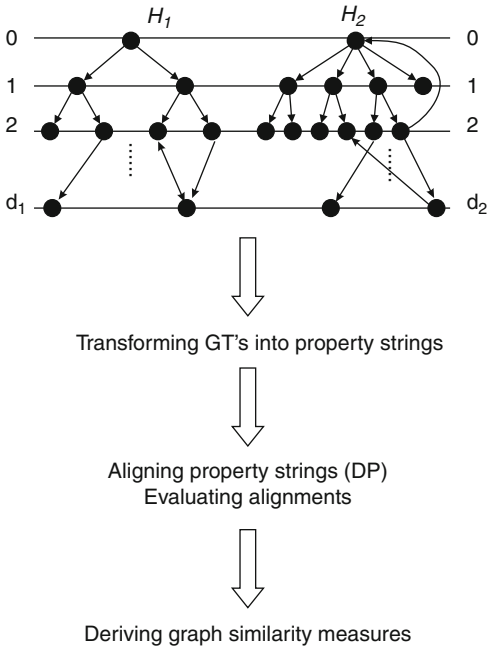

Fig. 11.3 Key steps to infer a graph similarity measure for generalized trees

- $s(x, y) > 0 \quad \forall x, y \in X.$
- $s(x, y) = s(y, x) \quad \forall x, y \in X.$
- $s(x, y) \leq s(x, x) = 1 \quad \forall x, y \in X.$

Definition 10 Let X be a set. A positive function $\omega : X \times X \longrightarrow [0, 1]$ is called distance measure if

- $\omega(x, y) \geq 0 \quad \forall x, y \in X.$
- $\omega(x, y) = \omega(y, x) \quad \forall x, y \in X.$
- $\omega(x, x) = 0 \quad \forall x \in X.$

Definition 11 Let H be a generalized tree. We call the set

$$S^H := \left\{ v^H_{0,1}, v^H_{1,1} \circ v^H_{1,2} \circ \cdots \circ v^H_{1,|V_1|}, \ldots, \circ v^H_{d,1} \circ v^H_{d,2} \circ \cdots \circ v^H_{d,|V_d|} \right\}, \quad (11.7)$$

the formal string representation of H. The symbol $\circ$ denotes usual string concatenation.

Definition 12 Let H be a generalized tree. We call

$$S^H_{\text{out}} := \left\{ \delta_{\text{out}} \left(v^H_{0,1} \right), \delta_{\text{out}} \left(v^H_{1,1} \right) \circ \delta_{\text{out}} \left(v^H_{1,2} \right) \circ \cdots \circ \delta_{\text{out}} \left(v^H_{1,|V_1|} \right), \ldots, \right.$$
$$\left. \circ \delta_{\text{out}} \left(v^H_{d,1} \right) \circ \delta_{\text{out}} \left(v^H_{d,2} \right) \circ \cdots \circ \delta_{\text{out}} \left(v^H_{d,|V_d|} \right) \right\}, \quad (11.8)$$

the set of out-degree property strings and

$$S_{\text{in}}^{H} := \{\delta_{\text{in}}\left(v_{0,1}^{H}\right), \delta_{\text{in}}\left(v_{1,1}^{H}\right) \circ \delta_{\text{in}}\left(v_{1,2}^{H}\right) \circ \cdots \circ \delta_{\text{in}}\left(v_{1,|V_1|}^{H}\right), \ldots, \\ \circ\delta_{\text{in}}\left(v_{d,1}^{H}\right) \circ \delta_{\text{in}}\left(v_{d,2}^{H}\right) \circ \cdots \circ \delta_{\text{in}}\left(v_{d,|V_d|}^{H}\right)\}, \tag{11.9}$$

the set of in-degree property strings of H.

Define $r_k^{H^k} := v_{0,1}^{H^k}, k \in \{1, 2\}$. Let H^1 be a given GT and $v_{i,j}^{H^1},\ 0 \leq i \leq d_1,\ 1 \leq j \leq \sigma_i$ denotes the j-th vertex on the i-th level of H^1. Analogously, this also holds for $v_{i,j}^{H^2} \in H^2$. As mentioned above, the task of measuring the structural similarity between H^1 and H^2 is equivalent to determine the optimal alignment of

$$S_1 = v_{0,1}^{H^1} \circ v_{1,1}^{H^1} \circ v_{1,2}^{H^1} \circ \cdots \circ v_{d_1,\sigma_{d_1}}^{H^1},$$
$$S_2 = v_{0,1}^{H^2} \circ v_{1,1}^{H^2} \circ v_{1,2}^{H^2} \circ \cdots \circ v_{d_2,\sigma_{d_2}}^{H^2},$$

with respect to their associated property strings and to a cost function α. $S_k[i]$ denotes the i-th position of the sequence S_k and it holds $S_1[n] = v_{d_1,\sigma_{d_1}}^{H^1}, S_2[m] = v_{d_2,\sigma_{d_2}}^{H^2},\ \mathbb{N} \ni n, m \geq 1,\ S_k[1] = r_k^{H_k},\ k \in \{1, 2\}$. The algorithm for finding the optimal alignment of S_1 and S_2 generates a matrix $(\mathcal{M}(i, j))_{ij},\ 0 \leq i \leq n,\ 0 \leq j \leq m$. We find that its time complexity is $O(|\hat{V}_1| \cdot |\hat{V}_2|)$, see [18, 23]. To determine optimal alignment of the derived property strings, we state the following algorithm [18, 23]:

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

and

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

for $1 \leq i \leq n,\ 1 \leq j \leq m$. Here, the derived property strings will be aligned on two levels: globally and locally. To evaluate the alignments, we need the preliminary assertion as follows.

Lemma 1 *Let* $\omega(x, y) := 1 - e^{-\frac{1}{2}\frac{(x-y)^2}{\sigma^2}}$. $\omega : \mathbb{R} \times \mathbb{R} \longrightarrow [0, 1]$ *is a distance measure.*

Proof From the definition of $\omega(x, y)$ we infer $\omega(x, y) \in [0, 1],\ \forall x, y \in \mathbb{R}$ and $\omega(x, x) = 1 - 1 = 0,\ \forall x \in \mathbb{R}$. Since $(x - y)^2 = (y - x)^2,\ \forall x, y \in \mathbb{R}$, the symmetry condition holds.

Now, we define

$$\alpha^{\text{out}}\left(v_{i_1,j_1}^{H^1}, v_{i_2,j_2}^{H^2}\right) := \begin{cases} \omega^{\text{out}}\left(\delta_{\text{out}}\left(v_{i_1,j_1}^{H^1}\right), \delta_{\text{out}}\left(v_{i_2,j_2}^{H^2}\right), \sigma_{\text{out}}^1\right) & : \quad i_1 = i_2 \\ +\infty & : \quad \text{else}, \end{cases}$$

$0 \leq i_k \leq d_k$, $1 \leq j_k \leq \sigma_{i_k}$, $k \in \{1, 2\}$, where $\omega^{\text{out}}(x, y, \sigma_{\text{out}}^k) := 1 - e^{-\frac{1}{2}(x-y)^2/(\sigma_{\text{out}}^k)^2}$, $x, y, \sigma_{\text{out}}^k \in \mathbb{R}$, and

$$\alpha^{\text{out}}\left(v_{i,j_1}^{H^1}, -\right) := \omega^{\text{out}}\left(\delta_{\text{out}}\left(v_{i,j_1}^{H^1}\right), \xi, \sigma_{\text{out}}^2\right),$$
$$\alpha^{\text{out}}\left(-, v_{i,j_2}^{H^2}\right) := \omega^{\text{out}}\left(\xi, \delta_{\text{out}}\left(v_{i,j_2}^{H^2}\right), \sigma_{\text{out}}^2\right).$$

$\xi > 0$ prevents an alignment between two leaves being better evaluated as an alignment between a leaf and a gap ("–") [22]. By $\omega^{\text{in}}\left(x, y, \sigma_{\text{in}}^k\right) := 1 - e^{-\frac{1}{2}(x-y)^2/\left(\sigma_{in}^k\right)^2}$, we define analogously $\alpha^{\text{in}}\left(v_{i_1,j_1}^{H^1}, v_{i_2,j_2}^{H^2}\right)$, $\alpha^{\text{in}}\left(v_{i,j_1}^{H^1}, -\right)$ and $\alpha^{\text{in}}\left(-, v_{i,j_2}^{H^2}\right)$.

To evaluate the alignments of the property strings locally (i.e., on each generalized tree level), we express the mapping [18, 22]

$$\text{align}\left(v_{i,j_1}^{H^1}\right) := \begin{cases} v_{i,j_2}^{H^2} & : \quad \text{align}^{-1}\left(v_{i,j_2}^{H^2}\right) = v_{i,j_1}^{H^1} \\ - & : \quad \text{else}. \end{cases}$$

For $v_{i,j_1}^{H^1}$, the mapping determines the vertex $v_{i,j_2}^{H^2}$ during the trace-back [18]. Moreover, we define the functions

$$\gamma_{H^k}^{\text{out}}(i) := \frac{\sum_{j=1}^{\sigma_i^k} \hat{\alpha}_{\text{out}}\left(v_{i,j}^{H^k}, \text{align}\left(v_{i,j}^{H^k}\right)\right)}{\sigma_i^k},$$
$$\gamma_{H^k}^{\text{in}}(i) := \frac{\sum_{j=1}^{\sigma_i^k} \hat{\alpha}_{\text{in}}\left(v_{i,j}^{H^k}, \text{align}\left(v_{i,j}^{H^k}\right)\right)}{\sigma_i^k},$$

$k \in \{1, 2\}$, which provide similarity values of the alignments of out-degree and in-degree property strings. Finally, by analogously defining the functions $\hat{\alpha}_{\text{out}}$ and $\hat{\alpha}_{\text{in}}$, we obtain the normalized and cumulative functions

$$\gamma^{\text{out}}\left(i, \hat{\sigma}_{\text{out}}^1, \hat{\sigma}_{\text{out}}^2\right) := 1 - \frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^1} \hat{\alpha}^{\text{out}}\left(v_{i,j}^{H^1}, \text{align}\left(v_{i,j}^{H^1}\right)\right) \right\}$$
$$- \frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^2} \hat{\alpha}^{\text{out}}\left(v_{i,j}^{H^2}, \text{align}\left(v_{i,j}^{H^2}\right)\right) \right\}, \quad (11.10)$$

and

$$\gamma^{\text{in}}\left(i, \hat{\sigma}^1_{\text{in}}, \hat{\sigma}^2_{\text{in}}\right) := 1 - \frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^1} \hat{\alpha}^{\text{in}}\left(v_{i,j}^{H^1}, \text{align}\left(v_{i,j}^{H^1}\right)\right) \right\} - \frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^2} \hat{\alpha}^{\text{in}}\left(v_{i,j}^{H^2}, \text{align}\left(v_{i,j}^{H^2}\right)\right) \right\}, \quad (11.11)$$

which detect the similarity of an out-degree and in-degree alignment on a level i. $\hat{\sigma}^1_{\text{out}}, \hat{\sigma}^2_{\text{out}}$ and $\hat{\sigma}^1_{\text{in}}, \hat{\sigma}^2_{\text{in}}$ are the parameters of $\hat{\alpha}^{\text{out}}$ and $\hat{\alpha}^{\text{in}}$, respectively. By using the defined quantities, it can be proven that the resulting comparative measure is a graph similarity measure (i.e., the measure satisfies the properties of Definition (9)) [18, 22].

Theorem 4 *Let H_1, H_2 be two generalized trees, $0 \leq i \leq \mu$, $\mu := \max(d_1, d_2)$. Then,*

$$s(H_1, H_2) := \frac{(\mu + 1)}{\sum_{i=0}^{\mu} \gamma^{\text{fin}}\left(i, \hat{\sigma}^1_{\text{out}}, \hat{\sigma}^2_{\text{out}}, \hat{\sigma}^1_{\text{in}}, \hat{\sigma}^2_{\text{in}}\right)} \prod_{i=0}^{\mu} \gamma^{\text{fin}}\left(i, \hat{\sigma}^1_{\text{out}}, \hat{\sigma}^2_{\text{out}}, \hat{\sigma}^1_{\text{in}}, \hat{\sigma}^2_{\text{in}}\right), \quad (11.12)$$

is a graph similarity measure where γ^{fin} is defined by

$$\gamma^{\text{fin}} = \gamma^{\text{fin}}\left(i, \hat{\sigma}^1_{\text{out}}, \hat{\sigma}^2_{\text{out}}, \hat{\sigma}^1_{\text{in}}, \hat{\sigma}^2_{\text{in}}\right) := \zeta \cdot \gamma^{\text{out}} + (1 - \zeta) \cdot \gamma^{\text{in}}, \quad \zeta \in [0, 1].$$

11.5 Applications

In the following, we outline existing and future applications of our presented approach which we have stated in Section 11.4.3. Here, we represent websites as a graph-based model [57] where we map each document structure to a generalized tree. In [22], a family of graph similarity measures was evaluated based on a corpus containing 500 conference websites from mathematics and computer science created by Mehler et al. [57]. Finally, the conference websites were inferred from the web and transformed into generalized trees by using the tool `HyGraph` [35, 36].

One of the main ideas is to apply a comparative analysis to a corpus consisting of graph-based web units. Now, for automatically analyzing web genre data, we propose the following evaluation steps:

1. Because the graph similarity measure outlined in Section 11.4.3 is parameterized, one can emphasize structural features of the graphs under consideration when measuring their structural similarity [27, 22]. This can be done by varying the parameters $\left(\zeta, \hat{\sigma}^1_{\text{out}}, \hat{\sigma}^2_{\text{out}}, \hat{\sigma}^1_{\text{in}}, \hat{\sigma}^2_{\text{in}}\right)$. For example in [27, 22], we have shown

that by setting ζ equal to 1 or 0, we either consider the alignments of out-degree or in-degree property strings only. To set $\zeta = \frac{1}{2}$ means that we weight the out-degree and in-degree property strings equally [22, 27].

2. We calculate the complete similarity matrix by computing the pairwise similarity scores of the given generalized trees. For this, we use the graph similarity measure presented in Section 11.4.3 with a fixed parameter set [22]. Moreover, we can compute the so-called cumulative similarity distribution Θ usually depicted as a two-dimensional plot. Θ can be used for expressing the percentage of generalized trees which possess a similarity value less or equal $s \in [0, 1]$ and, hence, to answer the question how structurally different the document structures of a given corpus are [22, 27]. Generally, we consider the study of Θ as a preliminary step for automatically analyzing web genre data that already led to a better understanding of the problem of comparing web-based hypertexts structurally [18, 22, 27].
3. Starting from a computed similarity matrix, one can additionally apply multivariate analysis methods, e.g., clustering techniques to filter web-based documents. By determining such clusters one identifies websites of similar structure, i.e., these clusters contain structurally similar web pages [18].

From the just outlined steps, it should be clear that this approach can also be used for analyzing data sets of hypertext structures inferred from other Web Mining areas. For example, if it would be possible to transform weblog data sets into sets of generalized trees, we could apply the approach analogously. This would result to novel applications in Web Usage Mining. In [18, 27] it has been sketched that the focus of such a study would be to analyze the navigation behavior of hypertext users [61, 66]. Generally, navigation patterns can be described by graphs [61, 66]. Particularly in our case, we would describe those by generalized trees. Each cluster we could determine by using the above stated approach then contains generalized trees which reflect a similar navigation behavior of a specific user. As we have already outlined in [18, 27], a possible interpretation of these clusters can lead to study psychological features of hypertext users.

11.6 Conclusion

The main goal of this conceptual chapter was to present an approach for automatically analyzing web genre data representing graphs. Instead of using the well-known vector space model for modeling document structures, we applied a graph-based representation model proposed by Mehler et al. [57]. A notable feature of this model is that the document structures represented by generalized trees capture more structural information than DOM-trees [18, 36, 57]. In Section 11.4.2, we briefly reviewed methods to measure the structural similarity of web-based documents which operate on tree structures only. In contrast to this, in Section 11.4.3 we repeated an approach for measuring the structural similarity of generalized trees. A key feature of this method is that the graphs will be transformed into linear integer

strings. By applying a string alignment algorithm, we weighted these alignments and finally derived a graph similarity measure for generalized trees. Hence, we solved a graph similarity problem by transforming it into a string similarity problem. Section 11.5 presented an overview of possible evaluation steps for automatically analyzing web genre data representing graphs. Moreover, existing applications of this approach were discussed.

Acknowledgments We are thankful to Alexander Mehler for fruitful discussions on this topic.

References

1. Albert, R., H. Jeong, and A.L. Barabási. 1999. Diameter of the world wide web. *Nature* 401:130–131.
2. Baeza-Yates, R., and B. Ribeiro-Neto, eds. 1999. *Modern information retrieval*. Reading, MA: Addison-Wesley.
3. Barabási, A.-L., and Z.N. Oltvai. 2004. Network biology: Understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113.
4. Basak, S.C., V.R. Magnuson, G.J. Niemi, and R.R. Regal. 1988. Determining structural similarity of chemicals using graph-theoretic indices. *Discrete Applied Mathematics* 19:17–44.
5. Batagelj, V. 1988. Similarity measures between structured objects. In *Proceedings of an International Course and Conference on the Interfaces between Mathematics, Chemistry and Computer Sciences*. Dubrovnik, Yugoslavia.
6. Bonchev, D. 1979. Information indices for atoms and molecules. *MATCH* 7:65–113.
7. Bonchev, D. 1983. *Information theoretic indices for characterization of-chemical structures*. Chichester: Research Studies Press.
8. Bornholdt, S., and H.G. Schuster. 2003. *Handbook of graphs and networks. From the genome to the Internet*. Weinheim: Wiley-VCH.
9. Brandes, U., and T. Erlebach. 2005. *Network analysis*. Lecture Notes in Computer Science. Heidelberg: Springer.
10. Bunke, H. 1983. What is the distance between graphs? *Bulletin of the EATCS* 20:35–39.
11. Bunke, H. 2000a. Recent developments in graph matching. In *Proceedings of the 15th International Conference on Pattern Recognition* 2:117–124.
12. Bunke, H. 2000b. Graph matching: Theoretical foundations, algorithms, and applications. In *Proceedings of Vision Interface 2000*, 82–88. Montreal, Canada.
13. Buttler, D. 2004. A short survey of document structure similarity algorithms. In *International Conference on Internet Computing*, 3–9. Los Vegas, Nevada, USA.
14. Carrière, S.J., and R. Kazman. 1997. Webquery: Searching and visualizing the web through connectivity. *Computer Networks and ISDN Systems* 29(8–13):1257–1267.
15. Chakrabarti, S. 2001. Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction. In *Proceedings of the 10th International World Wide Web Conference*, May 1–5, 211–220. Hong Kong.
16. Chakrabarti, S. 2002. *Mining the web: Discovering knowledge from hypertext data*. San Francisco, CA: Morgan Kaufmann.
17. Cook, D., and L.B. Holder. 2007. *Mining graph data*. Weinheim: Wiley-Interscience.
18. Dehmer, M. 2006. *Strukturelle analyse web-basierter Dokumente. Multimedia und Telekooperation*. Wiesbaden: Deutscher Universitäts Verlag.
19. Dehmer, M. 2008a. Information-theoretic concepts for the analysis of complex networks. *Applied Artificial Intelligence* 22(7 and 8):684–706.
20. Dehmer, M. 2008b. Information processing in complex networks:graph entropy and information functionals. *Applied Mathematics and Computation* 201:82–94.

21. Dehmer, M., and F. Emmert-Streib. 2007. Structural similarity of directed universal hierarchical graphs: A low computational complexity approach. *Applied Mathematics and Computation* 194:7–20.
22. Dehmer, M., and A. Mehler. 2007. A new method of measuring similarity for a special class of directed graphs. *Tatra Mountains Mathematical Publications* 36:39–59.
23. Dehmer, M., A. Mehler, and R. Gleim. 2004. Aspekte der Kategorisierung von Webseiten. In *Proceedings des Multimediaworkshops der Jahrestagung der Gesellschaft für Informatik*, eds. P. Dadam und M. Reichert, Lecture Notes in Computer Science, vol. 2, 39–43, Berlin: Springer.
24. Dehmer, M., F. Emmert-Streib, and J. Kilian. 2006. A similarity measure for graphs with lowcomputational complexity. *Applied Mathematics and Computation* 182:447–459.
25. Dehmer, M., A. Mehler, and F. Emmert-Streib. 2007. Graphtheoretical characterizations of generalized trees. In *Proceedings of the International Conference on Machine Learning: Models, Technologies & Applications (MLMTA'07)*. Las Vegas, NV.
26. Dehmer, M., F. Emmert-Streib, and T. Gesell. 2008. A comparative analysis of multidimensional featuresof objects resembling sets of graphs. *Applied Mathematics and Computation* 196:221–235.
27. Dehmer, M., F. Emmert-Streib, A. Mehler, and J. Kilian. 2006. Measuring the structural similarity of web-based documents: A novel approach. *International Journal of Computational Intelligence* 3(1):1–7.
28. Dimter, M. 1981. *Textklassenkonzepte heutiger Alltagssprache*. Tübingen: Niemeyer.
29. Dorogovtsev, S.N., and J.F.F. Mendes. 2003. *Evolution of networks. From biological networks to the internet and WWW*. Oxford: Oxford University Press.
30. Emmert-Streib, F., and M. Dehmer. 2007. Information theoretic measures of UHG graphs with low computational complexity. *Applied Mathematics and Computation* 190:1783–1794.
31. Ferber, R. 2003. *Information retrieval. Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web*. Heidelberg: dpunkt.verlag.
32. Flesca, S., G. Manco, E. Masciari, L. Pontieri, and A. Pugliese. 2002. Detecting structural similarities between XML documents. In *Proceedings of the International Workshop on the Web and Databases (WebDB 2002)*. Madison, Wisconsin, USA.
33. Foulds, L.R. 1992. *Graph theory applications*. New York, NY: Springer.
34. Gibson, D., R. Kumar, K.S. McCurley, and A. Tomkins. 2007. Dense subgraph extraction. In *Mining graph data*, eds. D. Cook and L.B. Holder, 411–441. Hoboken, NJ: Wiley-Interscience.
35. Gleim, R. 2004. Integrierte Repräsentation, Kategorisierung und Strukturanalyse Webbasierter Hypertexte. Master's thesis, Technische Universität Darmstadt, Fachbereich Informatik, Sept 2004.
36. Gleim, R. 2005. HyGraph: Ein Framework zur Extraktion, Repräsentation und Analyse webbasierter Hypertexte. In *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen. Beiträge zur GLDV-Tagung 2005 in Bonn*, eds. B. Fisseni, H.-C. Schmitz, B. Schröder, and P. Wagner, 42–53. Frankfurt a.M.: Lang.
37. Halin, R. 1989. *Graphentheorie*. Berlin: Akademie Verlag.
38. Han, J., and M. Kamber. 2001. *Data mining: Concepts and techniques*. New York, NY: Morgan and Kaufmann Publishers.
39. Harary, F. 1969. *Graph theory*. Reading, MA: Addison Wesley Publishing Company.
40. Huberman, B., and L. Adamic. 1999. Growth dynamics of the world-wide web. *Nature*, 399:130.
41. Jiang, T., L. Wang, and K. Zhang. 1994. Alignment of trees – an alternative to tree edit. In *CPM '94: Proceedings of the 5th Annual Symposium on Combinatorial Pattern Matching*, 75–86, London: Springer-Verlag.
42. Joshi, S., N. Agrawal, R. Krishnapuram, and S. Negi. 2003. A bag of paths model for measuring structural similarity in web documents. In *KDD '03: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 577–582, New York, NY.

43. Kaden, F. 1982. Graphmetriken und Distanzgraphen. *ZKI-Informationen, Akademie der Wissenschaften der DDR* 2(82):1–63.
44. Kaden, F. 1986. Graphmetriken und Isometrieprobleme zugehöriger Distanzgraphen. *ZKI-Informationen, Akademie der Wissenschaften der DDR* 1(P6):1–100.
45. Kleinberg, J.M. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46(5):604–632.
46. Kosala, R., and H. Blockeel. 2000. Web mining research: A survey. SIGKDD explorations: Newsletter of the Special Interest Group (SIG) on knowledge discovery & data mining, *ACM* 2(1):1–15.
47. Koschützki, D., K.A. Lehmann, L. Peters, S. Richter, D. Tenfelde-Podehl, and O. Zlotkowski. 2005. Clustering. In *Centrality indices*, eds. U. Brandes and T. Erlebach, Lecture Notes of Computer Science, 16–61. Berlin: Springer.
48. Kumar, R., P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tompkins, and E. Upfal. 2000. The web as a graph. In *PODS '00: Proceedings of the 19th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 1–10, New York, NY: ACM Press.
49. Levenstein, V.I. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics – Doklady* 10(8):707–710, Feb 1966.
50. Lindemann, C., and L. Littig. 2010. Classification of web sites at super-genre level. In *Genres on the web: Computational models and empirical studies*, eds. A. Mehler, S. Sharoff, and M. Santini, Text, Speech and Language Technology. Dordrecht: Springer.
51. Mason, O., and M. 2007. Verwoerd. Graph theory and networks in biology. *IET Systems Biology* 1(2):89–119.
52. Mehler, A. 2001. *Textbedeutung. Zur prozeduralen Analyse und Repräsentation struktureller Ähnlichkeiten von Texten*, volume 5 of *Sprache, Sprechen und Computer/Computer Studies in Language and Speech*. Frankfurt a. M.: Peter Lang.
53. Mehler, A. 2004. Textmining. In *Texttechnologie. Perspektiven und Anwendungen*, eds. H. Lobin and L. Lemnitzer, 83–107. Tübingen: Stauffenburg.
54. Mehler, A. 2009. Generalized shortest paths trees: A novel graph class applied to semiotic networks. In *Analysis of complex networks: From biology to linguistics*, eds. M. Dehmer and F. Emmert-Streib, 175–220. Weinheim: Wiley-VCH.
55. Mehler, A. 2010. Structure formation in the web. toward a graphtheoretical model of hypertext types. In *Linguistic modelling of information and markup languages*, eds. A. Witt and D. Metzing, 225–247. Dordrecht: Springer.
56. Mehler, A., and R. Gleim. 2006. The net for the graphs – towards webgenre representation for corpus linguistic studies. In *WaCky! Working papers on the web as corpus*, eds. M. Baroni and S. Bernardini, 191–224. Bologna: Gedit.
57. Mehler, A., M. Dehmer, and R. Gleim. 2004. Towards logical hypertext structure – A graph-theoretic perspective. In *Proceedings of the Fourth International Workshop on Innovative Internet Computing Systems (I2CS '04)*, eds. T. Böhme and G. Heyer, Lecture Notes in Computer Science, vol. 3473, 136–150, Berlin/New York: Springer.
58. Mehler, A., R. Gleim, and M. Dehmer. 2005. Towards structure-sensitive hypertext categorization. In *Proceedings of the 29th Annual Conference of the German Classification Society*, LNCS, Mar 9–11. Universität Magdeburg, Berlin/New York, NY: Springer.
59. Mehler, A., R. Gleim, and A. Wegner. 2007. Structural uncertainty of hypertext types. An empirical study. In *Proceedings of the Workshop "Towards Genre-Enabled Search Engines: The Impact of NLP"*, 30 Sept 2007, 13–19, in conjunction with RANLP 2007. Borovets, Bulgaria.
60. Messmer, B.T., and H. Bunke. 1998. A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(5):493–504.
61. Noller, S., J. Naumann, and T. Richter. 2001. LOGPAT – Ein webbasiertes Tool zur Analyse von Navigationsverläufen in Hypertexten. http://www.psych.uni-goettingen.de/congress/gor-2001
62. Power, R., D. Scott, and N. Bouayad-Agha. 2003. Document structure. *Computational Linguistics* 29(2):211–260.

63. Raghavan, P. 2000. Graph structure of the web: A survey. In *LATIN 2000: Theoretical Informatics. Proceedings of 4th Latin American Symposium*, 123–125. Punta del Este, Uruguay.
64. Rahm, E. 2002. Web usage mining. *Datenbank-Spektrum* 2(2)75–76.
65. Rehm, G. 2007. *Hypertextsorten. Definition – Struktur – Klassifikation*. Norderstedt: Books on Demand.
66. Richter, T., J. Naumann, and S. Noller. 2003. Logpat: A semi-automatic way to analyze hypertext navigation behavior. *Swiss Journal of Psychology* 62:113:120.
67. Schädler, C. 1999. Die Ermittlung struktureller Ähnlichkeit undstruktureller-Merkmale bei komplexen Objekten: Einkonnektionistischer Ansatz und seine Anwendungen. PhD thesis, Technische Universität Berlin.
68. Scsibrany, H., K. Karlovits, W. Demuth, F. Müller, and K. Varmuza. 2003. Clustering and similarity of chemical structures represented by binary substructure descriptors. *Chemometrics and Intelligent Laboratory Systems* 67:95–108.
69. Selkow, S.M. 1977. The tree-to-tree editing problem. *Information Processing Letters* 6(6):184–186.
70. Skorobogatov, V.A., and A.A. Dobrynin. 1988. Metrical analysis of graphs. *MATCH* 23:105–155.
71. Sobik, F. 1982. Graphmetriken und Klassifikation strukturierter Objekte. *ZKI-Informationen, Akademie der Wissenschaften der DDR* 2(82):63–122.
72. Sobik, F. 1986. Modellierung von Vergleichsprozessen auf der Grundlage von Ähnlichkeitsmaßen für Graphen. *ZKI-Informationen, Akademie der Wissenschaften der DDR* 4:104–144.
73. Spiliopoulou, M. 2000. Web usage mining for web site evaluation. *Communications of the ACM* 43(8):127–134.
74. Tai, K.C. 1979. The tree-to-tree correction problem. *Journal of the ACM* 26(3):422–433. ISSN 0004-5411.
75. Waltinger, U., A. Mehler, and A. Wegner. 2009. A two-level approach to web genre classification. In *Proceedings of the 5th International Conference on Web Information Systems and Technologies (WEBIST '09)*, 23–26 Mar 2009. Lisboa.
76. Wasserman, S., and K. Faust. 1994. *Social network analysis: Methods and applications*, Structural Analysis in the Social Sciences. Cambridge, MA: Cambridge University Press.
77. Zelinka, B. 1975. On a certain distance between isomorphism classes of graphs. *Časopis pro ˇpest. Mathematiky* 100:371–373.