The Net for the Graphs: Towards Webgenre Representation for Corpus Linguistic Studies

Alexander Mehler *Rüdiger Gleim*

1 Introduction

In recent years, the Web has become increasingly significant for corpus linguistic research (Baroni and Bernardini 2004; Keller and Lapata 2003; Kilgarriff and Grefenstette 2003; Resnik and Smith 2003; Santamaría et al. 2003). On the one hand, it contains a vast amount of hypertext documents of newly emerging document types (e.g., *conference websites*, *corporate sites*, *electronic encyclopedias*, *hotlists*, *sites of online shops*, *(personal, academic) home pages*, *weblogs* etc.). On the other hand, the Web has become accepted as a common platform for information exchange so that one can find instances of almost any type of electronic text imaginable. This, *in theory*, makes the Web the source of choice when large corpora for studying language varieties are needed. But it also makes it the source of choice when studying the emergence and evolvement of hypertext types. The reason for this assessment is also the main source of difficulties one has to face following this line of research: Web-based hypertext authoring mostly utilizes languages, as for example HTML, CSS and related "standards", in spite of their well-known deficits regarding the separation of structure, content and form. Moreover, these languages do not at all standardize the content-based, functional structuring of websites, neither with respect to the internal structuring of constitu-

tive webpages, nor with respect to page linkage. Rather, the kinds of structuring and linkage observable on the Web emerged spontaneously and rapidly during its short history. These kinds are, of course, not completely determined by the medium or authoring software used, but vary with the different functions and contents they carry and the styles of Web document authors. Nevertheless, hypertextual patterns allow reliable predictions of the functions being manifested. We have no problem distinguishing, for example, a personal academic home page from a conference website not only in terms of content but also in terms of *document structure*.

The Web apparently manifests an evolution of hypertextual patterns *in fast motion* making its various mutations accessible to corpus linguistic studies. This implies that the tremendous differences in structural quality manifested by websites are by no means a venial deficit to be abstracted away by hypertext representation. Rather, this informational variety is an indispensable characteristic of the kind of structure formation under consideration. As a consequence, any approach to representing Web-based patterns of hypertext authoring has to face the task of representing and processing various aspects of informational uncertainty. In other words: The apparatus of probabilistic modeling will be needed in order to model, for example, aspects of structural ambiguity, under-specification and vagueness of structural descriptions of Web-based units, their constituency and dependency structure.

This paper is about prerequisites of representing patterns of Web-based hypertext authoring. Its basic tenet is that *websites* and their constitutive *pages* are instances of *webgenres* (Crowston and Kwasnik 2003; Crowston and Williams 1999, 2000; Dillon and Gushrowski 2000; Orlikowski and Yates 1994; Rehm 2002; Yoshioka and Herman 2000) and their elementary *stages* (Ventola 1987) or *phases* (Eggins 1994) by analogy with texts and their components as instances of genres and generic stages (Martin 1992). We hypothesize a webgenre to be identifiable by means of function

bearing patterns whose variance within the same genre is lower than between different ones. In the following sections, we discuss an indispensable prerequisite for automatically studying this functional variety, namely the download and representation of presumptive webgenre instances on the level of websites.

When it comes to an experiment in corpus-based analysis in this area, one is confronted with a tremendous set of problems. To name only a few of these: How do we identify the extent of a website of a given webgenre? In other words, how do we identify Web-based hypertext borders? What does an appropriate representation model look like which allows one to represent the different kinds of textual and hypertextual structures manifested by websites? How do we deal with flawed website manifestations as a result of, for example, malformed HTML-coding, broken links or missing structural explicitness? How do we make the resulting website representation retrievable for the different tasks in corpus linguistic research?

Since a loss of information occurs every time a website is taken out of its context, answers to these questions have to be carefully considered. We hypothesize that appropriate answers get their validity to the degree to which they clarify the relation of *explicit* (*visible*) or *manifesting* website structure and *implicit* (*hidden*) or *manifested* webgenre structure.

This paper addresses some aspects of representing hypertextual units with a focus on websites as instances of webgenres. The subsequent sections concentrate on representational and technological issues of this task. Starting from a draft of our *conceptual data model* of webgenres, some major problems in website representation are specified in section 2. This relates, amongst other things, to the so called *polymorphism* and *polyfunctionality* of hypertextual units. In section 3 we sketch our *logical data model* which is based mainly on graph theory. Subsequent to this logical specification of the conceptual model, its physical implementation

is presented in section 3 too. We utilize the G*raph* e*X*c*hange* L*anguage* (GXL) (Winter et al. 2002) and thus propose a document schema as an appropriate format for physical data modeling of websites. As this paper focuses on the *explicit* (*visible*) structure of websites, sections concentrate on representing hyperlinks (3.1), the nesting of link, document and linguistic structure (3.2) and structure formation in time (3.3). Section 4 utilizes this model in order to derive constraints for exploratory corpus analyses. Finally, the conclusion gives a prospect on future work. This relates especially to *mining* and representing the implicit genre-specific, functional structure of websites. In summary, the present paper can be seen as a preparatory step towards mining this hidden webgenre structure.

2 Outline of a conceptual model of genre-specific website structuring

According to discourse analysis, distributional patterns vary depending on the functions of the discourses in which they are observed (Biber 1995). Starting from the *weak contextual hypothesis* of Miller and Charles (1991) which says that the similarity of the contextual representations of words contributes to their semantic similarity, one might state that differences of textual form reflect differences in function as far as they are confirmed by a significantly high number of instances and thus are recognizable as text patterns. The main objective of the approach followed by the present paper is to verify this hypothesis in the area of Web-based documents. That is, we expect websites of different genres to be distinguished by the function bearing patterns they manifest. We expect this distinguishability to also hold – although to a minor degree – for the constituents of websites (e.g., webpages) and the sub-functions they serve.

In order to further specify this hypothesis, the concept of *web-*

genre has to be narrowed down. This can be done by abstractly defining a *document class* as a class of textual or hypertextual units which serve the same or related functions and thus manifest similar structures and layout shapes. Different criteria of document class formation relate to different types of access to such functional entireties. If we consider, for example, the composition of classes from an extensional point of view, that is from the point of view of their document elements, we deal with *text sorts* (Heinemann 2000). If we concentrate instead on *situative* or *communicative* criteria of class membership, we deal with *registers* (Biber 1995; Halliday and Hasan 1989) and *genres* (Martin 1992), respectively. In analogy to this, we find references to *hypertext sorts*, *digital genres* and *webgenres* in case of classes of hypertextual documents (Dillon and Gushrowski 2000; Jakobs 2003; Orlikowski and Yates 1994; Rehm 2002). If in contrast to this, class membership is defined in intensional terms, we deal with *text patterns* and *superstructures* as prototypical representations of class members, whose expectation driven production/reception they support (Heinemann 2000; van Dijk and Kintsch 1983). The basis of all these approaches is the notion that structure and shape of (hyper-)textual units vary (though not deterministically) in dependence on the communicative situation or function they manifest. If we focus on structure abstracting from shape or layout, respectively, we deal with the *logical document structure*. As we deal with hypertextual units we speak, more specifically, of the logical *hypertext* document structure.

The taxonomic notion of genre of Yates and Orlikowski (1992), to which the majority of approaches to webgenres refers, aims at genre classifications. A review of the notion of webgenre is given by Firth and Lawrence (2003). They analogously identify the focus of research in this area with classification. Crowston and Williams (2000), for example, identify *hotlists*, *home pages* and *Web server statistics* as original webgenres without precursors in

literary language (Dillon and Gushrowski 2000), whose classification necessarily includes hypertextual genre markers (Crowston and Williams 1999). Consequently, the identification of sufficiently selective markers is seen as one of the main tasks of webgenre analysis (Rehm 2002). An instance of taxonomic genre analysis on the level of webpages is given by Yoshioka and Herman (2000) who analyze a single website by mapping its constitutive pages on a set of genre categories. See also Rehm (2002) who classifies generic modules of single pages.

In addition to the taxonomic notion, the procedural organization of genres is examined in systemic-functional linguistics (Halliday and Hasan 1989; Martin 1992). That is, dependency relations of generic constituents (i.e., stages or phases) and their chronology are studied from the point of view of text type formation (Ventola 1987). This approach is adopted in the present paper since it allows to identify links between pages of the same site as manifestations of webgenre internal structure (Mehler et al. 2004; Mehler and Gleim 2005). This notion is confronted with serious problems of hypertext representation which can all be traced back to the fundamental distinction of visible or manifesting website structure and hidden or manifested webgenre structure. In order to explain this, we start from a four level model of Web-based structure formation, that is of logical hypertext document structure, including the level of elementary building blocks, module types, Web document types and document network types (Mehler and Gleim 2005). *Building blocks* (manifested, for example, by tables or paragraphs) exist only as dependent parts of *module types* which relate to functionally homogeneous sub-functions of Web-based communication (e.g., *call for papers*, *program* or *conference venue* as sub-functions of the spanning function of *Web-based conference organization*).[1] Next, *Web document types* classify Web-

[1] See Storrer (2002) for a definition of the notion of module in the context of hypertext authoring.

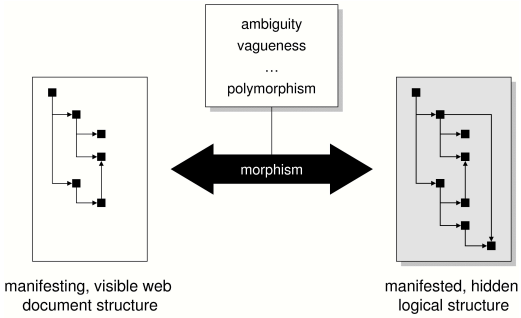

Figure 1. Informational uncertainty of the morphism interrelating manifested and manifesting structure

based manifestations of pragmatically closed acts of Web-based communication, where each of these acts serves a complex function of, for example, *conference organization*, *personal presentation* or *online shopping*. Fourth, *document network types* relate to systems of pragmatically closed, though not necessarily homogeneous communication acts. A document network type is manifested, for example, by a university's website which covers, amongst others, personal academic home pages, project sites and library sites which together contribute to the same corporate identity.

This enumeration might suggest that the levels are deterministically separated without recourse to informational uncertainty. It might also suggest that they directly relate to HTML-elements, webpages, websites and compound websites, respectively. This is, of course, not the case. In fact, there exists a many-to-many relation between functionally specified levels of Web-based communication and their manifestations by means of pages and related expression units (see figure 1), that is, between hidden hypertext document structure and manifesting website structure. Without systematizing the morphism of figure 1 – for more details see Mehler and Gleim (2005); Mehler et al. (2005) –, we only emphasize two aspects of informational uncertainty:

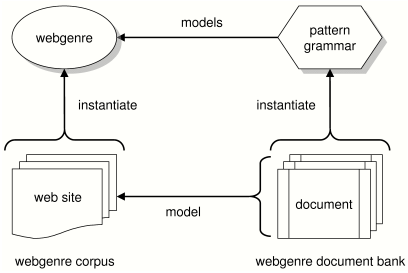

Figure 2. The basic model of document pattern-oriented webgenre analysis

- *Polymorphism* occurs if the same expression unit manifests several categories by means of separate segments. Polymorphism is given when, for example, the same webpage of a conference website provides information about the *call for papers*, the *submission procedure* and *conference registration*, that is, when it manifests two or more functions. Polymorphism results in *multiple categorizations* without being reducible to ambiguity of category assignment since in this case several categories are actually manifested by the same expression unit. Thus, resolving polymorphism cannot be reduced to the task of disambiguating category assignment as applied in machine learning and related areas.

- *Discontinuous manifestation* occurs if the same function or content unit is distributed over several expression units. Discontinuous manifestation results in *flawed* or even *missing categorizations* since in this case the webpages under consideration manifest the focal content/function category only in part. Thus, discontinuous manifestation relates to vagueness.

These two relations constitute a many-to-many relation of *function* (and *content*) units on the one hand and *expression* units on the other hand. As a result of this relation, the function or

content structure of a website is generally *not* directly accessible by just segmenting and subsequently categorizing its constitutive webpages in separation (for more information on this argumentation see Mehler et al. 2005). Moreover, links cannot be directly identified as manifestations of the "staging" of a webgenre or of the ordered progression of its phases and their structuring. This observation makes the representation of a webpage's internal *and* external structure an indispensable prerequisite for any effort in exploring the genre-specific structure of websites.

Figure 2 summarizes our webgenre model presented so far: Webgenres are considered to be manifested by websites (consisting of at least one webpage) whose structure is an informationally uncertain map of the underlying, hidden functional (webgenre) structure. As a consequence, corpora of website representations, henceforth called *webgenre document banks*, are needed, whose document elements map both: the manifesting website structure and the manifested webgenre structure as it is instantiated by the former. Finally, webgenre pattern grammars have to be induced on the basis of the input document banks which allow to classify newly observed instances according to the genre-specific patterns they manifest.

In the next section we present our approach as far as it focuses on the prerequisite of representing websites as expression units. Thus, it concentrates on the representation of explicit, visible website structure leaving the induction of the hidden logical hypertext document structures to future work (cf. Mehler et al. 2005 for a first approach to such an induction algorithm).

3 A text technological view on representing websites

This section outlines the basic building blocks of the format we use for representing websites. It is part of the `HyGraph` system

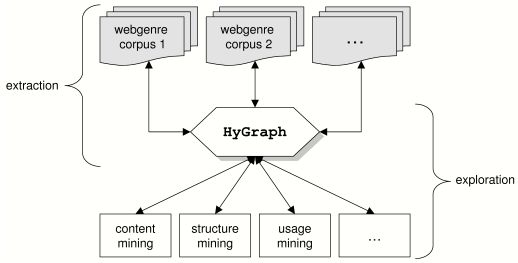

Figure 3. The HyGraph system as a generic Web mining interface for webgenre analysis

(Gleim 2005) which mediates between webgenre corpora and their processing for the various tasks of Web content, structure and usage mining (see figure 3). The **HyGraph** system addresses the following tasks of hypertext document processing: extraction of corpora of websites of certain webgenres; generic representation of Web documents; Web corpus management and maintenance; visualization of Web document structure; unsupervised learning of hypertext graphs.

In this paper we concentrate on the second of these tasks and thus ask for an appropriate representation format. A common framework for representing hypertextual units is graph theory. This relates especially to the area of *directed graphs*.[2] Consequently, various metrics of hypertext structure have been defined on digraphs (Botafogo et al. 1992; Chakrabarti 2002; Furner et al. 1996). However, even simple Web-based units show a structural complexity beyond digraphs. Hyperlinks, for example, often address sections of their corresponding target pages. In such relations, up to four elements can be involved: The source and target page as well as the source and target anchor. It is evident that

[2] A directed graph or digraph G is an ordered pair $G = (V, E)$ of a set V of vertices and a set E of edges where $E \subseteq V^2$. For a detailed introduction to graph theory see Melnikov et al. (1994).

this is only a simple example of many more complex cases where the expressive power of digraphs is exceeded:

- *Link structure:* Website internal and external links have to be identified as well as the graph structures (e.g., sequences, hierarchies and networks of interlinked units) they induce. In section 3.1 we consider different types of hyperlinks and the hierarchical structures they induce and transcend, respectively.
- *Nested structures:* Link classification is a new task in machine learning (Getoor 2003). It asks for representation models which go down to the wording of single pages – comparable to the bag-of-words model, but with the important difference that now graphs of such representations are needed since webpages are embedded as vertices into hypertext graphs. In section 3.2, we consider the HTML-based DOM structure and the text-based logical document structure of single pages as complements of their internal link structure.
- *Time alignment:* Websites are, of course, no stable units, but evolve in time. When they are created, conference websites often only consist of a single page announcing the conference. Then, they gradually grow as the conference approaches. Once it is over, some of the website's sections are removed (e.g., registration), others are added (e.g., conference pictures) before the website is finally deleted. In order to grasp this kind of life cycle-based structure formation, a format is needed which allows identifying different graph representations as being manifestations of the same logical unit *at different points in time*. This is outlined in section 3.3.

It is evident that a rather complex class of graphs is needed

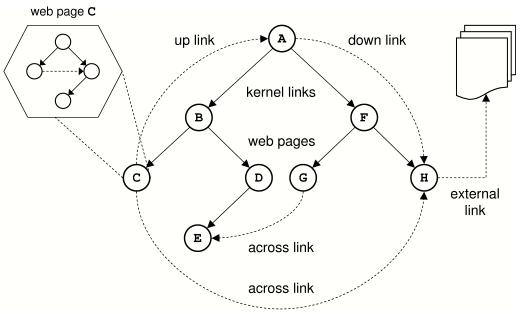

Figure 4. Types of links connecting webpages symbolized as circles

as a *logical data model* in order to meet these requirements for adequate hypertext representation. It should allow to express relations between arbitrary numbers of vertices as well as hierarchical embeddings of graphs into vertices. We utilize the G*raph* e*X*c*hange* L*anguage* (GXL; Winter et al. 2002) as a format of *physical data modeling* in order to serve these needs. We propose using GXL for computer-based storage, maintenance and retrieval of genre-specific website representations. On the level of logical data modeling it corresponds to certain classes of graphs whose usage will also be motivated.

3.1 Representing internal and external link structure

In order to introduce our format of website representation, we start from a simplified model consisting of a directed tree (henceforth called *kernel hierarchy*) rooted by the so called leader in the sense of Eiron and McCurley (2003) (i.e., its "start page") and augmented by across, up and down links which together span a website's *hypertext graph* (see figure 4). In this section, we explain why this hypertext graph is a *hypergraph*, but not just a digraph.

The notion of a kernel hierarchy is exemplified by a conference

website headed by a menu and title page referring to, for example, its call for papers which in turn may be continued by a page on the conference's sessions etc. so that finally a hierarchical structure evolves. It is evident that the kernel hierarchy reflects navigational constraints. That is, the position of a page in this tree can be seen as reflecting the probability to be navigated by a reader starting from the root and following only its kernel links. The welcome page of a corporate website, for example, is far easier to reach than the contact information of the service hotline.

Variable	**Value**
number of websites	1,096
number of webpages	50,943
number of hyperlinks	303,278
maximum depth	23
maximum width	1,035
average size	46
average width	38
average height	3

Table 1. A sample corpus of 1,096 conference and workshop websites

A website's kernel hierarchy is spanned by so called kernel links. Kernel links have to be distinguished from across, up, down, inside and outside links (Amitay et al. 2003; Eiron and McCurley 2003; Routledge et al. 2000), which in the following are defined on the basis of the kernel hierarchy of the hypertext graph (see figure 4):

- *Kernel links* associate dominating nodes with their immediately dominated successor nodes in terms of the kernel hierarchy.
- *Down links* associate nodes with one of their (normally mediately) dominated successor nodes in terms of the kernel hierarchy – possibly parallel to a kernel link.
- *Up links* analogously associate nodes of the kernel hierarchy

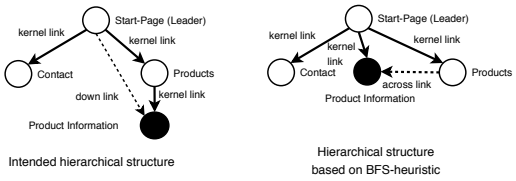

Figure 5. A problem of the heuristics of breadth first search regarding the detection of a website's kernel hierarchy

with one of their (normally mediately dominating) predecessor nodes.

- *Across links* associate nodes of the kernel hierarchy none of which is an (im-)mediate predecessor of the other in terms of the kernel hierarchy.
- *Inside links* are node (i.e., page) internal links.
- *Outside links* associate nodes of the kernel hierarchy with nodes of other websites.

Table 1 lists the frequencies of these link types as found in our test corpus of 50,943 pages of 1,096 conference websites from the fields of computer science and mathematics.

As these types of links are not explicitly tagged, they have to be automatically detected. We use a heuristic method based on a breadth-first search starting with the leader of the input hypertext graph. Consequently, pages directly accessible from the root are mapped onto the second level of the kernel hierarchy and so on until the levels of the leaves are reached. It is easy to conceive cases where this method fails to detect the correct kernel structure. If, for example, a company releases a new product there might be a newsflash on the welcome page of its website which directly links to the product description. In this case, the product description page is rated too high because of being directly accessible from the root.

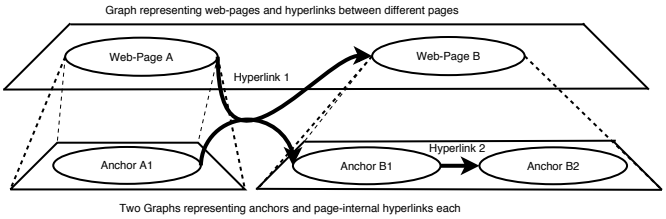

Figure 6. A layer-model of website representation embedding two page representation graphs into a website representation graph

Instead of that it should be located below the "products" page. Figure 5 illustrates this example. In order to solve this problem, it is necessary to have knowledge of the contents and purposes of webpages and of the prototypical structure of the webgenre they instantiate. That is, this example already leads to the level of implicit hypertext document structure.

The picture of website structuring we get from these considerations is that of a hypertext graph representing pages and their links as nodes and edges, respectively. As internal links belong to single pages they are represented as part of these pages' node representations (see figure 6; see also table 2). This model now allows us to introduce the physical data model based on GXL:

- *Graphs* are ordered pairs (V, E) of a vertex set V and an edge set E. In GXL, vertices are referred to as XML-elements named `node`. In the present framework, instances of this element are commonly used to represent single webpages identified by an `ID` (see table 2) and a GXL-attribute named `URI`. Accordingly, instances of the elements `edge` and `rel`(ation) are used to represent links of these nodes (see table 2).

- *Typed graphs* are graphs with typed vertices and edges. Amongst other things, we utilize typing to distinguish anchor and page nodes as well as frame source links from "standard"

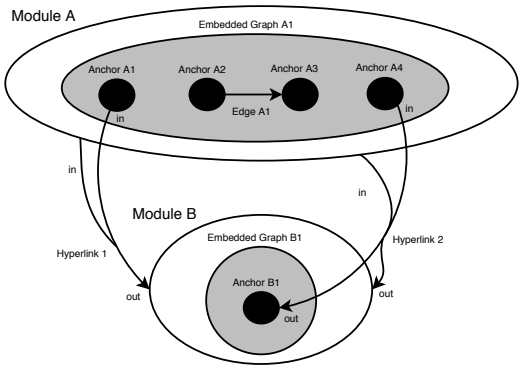

Figure 7. Three cases of page linkage (edge A1, hyperlink 1 and 2)

links. This typing (not to be confused with the distinction of link types above) is manifested by the `type` element and its `xlink:href` attribute. Since we need several type systems to independently classify the same set of hypertext constituents, we also construct attributed graphs.

- *Attributed graphs* are graphs whose nodes and edges are assigned possibly nested bags, sets, tuples or sequences of boolean, integer, real or string valued attributes. We use attributed graphs to model the URL of a webpage as an attribute-value pair and its metatags as a bag of such pairs enclosed by an instance of the GXL-attribute `MetaTags`. Unlike in Mehler et al. (2004), we do no longer map a page's textual content onto a token vector attribute, but map it as a graph on its own (see section 3.2). But we still use attributes in order to type links. That is, links are assigned a GXL-attribute `types` whose values distinguish, amongst others, between *across*, *up*, *down*, *inside* and *outside* links (see table 2).

- *Directed graphs* are graphs whose edges are ordered pairs of nodes, *adjacent from* their source node and *adjacent to* their target node. They are the default means of representing HTML links whose source and target anchors belong to the same webpage, i.e., page internal links (see link `Edge A1` in figure 7). This is done with the help of two attributes assigned to the `edge` element (see table 2): `from` and `to` take the `ID` of the corresponding source and target node anchor as values, respectively. In spite of this preferred usage, `edge` elements, their attributes and content model are not restricted to map HTML links. According to the GXL model of hypergraphs (see the last bullet of this listing), even sophisticated links following the XLink standard can be modeled by means of GXL.

- *Ordered graphs* are directed graphs whose arcs are assigned ordinal numbers reflecting any order dependent on their respective source node. In linguistics, these numbers can be used to model the syntagmatic order of the immediate constituents of the same superordinate node. In hypertext representation, they are analogously used to model the order of links which are adjacent *from* the same node. This order depends on the syntagmatic order of the links' anchors. It is manifested by means of an attribute of name `startorder` or `endorder`, respectively, which is assigned to `rel`(ation)`end` elements of the focal `rel`(ation) element.[3] All `startorder` (`endorder`) attribute values of `rel`(ation)`end`s which are incident from (to) the same node have to define a proper ordering on the `rel`(ations) involved (see table 2).[4]

[3] In the case of edges, the attributes `fromorder` and `toorder` are used instead.

[4] This is not the standard interpretation of both attributes in GXL, but the one which is needed in order to map the order of `rel`(ations) according to the syntagmatic order of the anchor nodes of the hyperlinks they are used to

- *Stratified graphs* are graphs whose nodes embed graphs on their own. In the present framework they serve to model page-internal link structures based on links whose source and target anchors belong to the same page (e.g., `Edge A1` in figure 7). In order to map the internal link structure of a page A, we embed the graph spanned by this structure into the node representing A. This part of the model is in accordance with the paradigm of document-oriented modelling complementing the predominant data-oriented character of GXL. Since page-internal links simply consist of a possibly attributed association of two anchor nodes of the same page, the `edge` element suffices as the GXL analogue of edges in digraphs in order to model this kind of link. In the case of all other links, *hyperedges* of *hypergraphs* are used instead.

- *Hypergraphs* are graphs whose *hyper*edges are subsets of the vertex set V. Hyperedges may also be ordered and directed. This qualifies them for modeling HTML links whose anchors belong to different webpages (see `Hyperlink 2` in figure 7). Table 2 illustrates an instance of the element `rel`(ation) which models a link of two pages (identified by `ModuleA` and `ModuleB`). The content model of the hyperedge in question comprises a `rel`(ation)`end` element targeting at `ModuleA` as its `sourcepage`, a `relend` targeting at `ModuleB` as its `targetpage`, and a `relend` element targeting at the link's source page anchor. Links with a target anchor specification in the URL value of their `href` attribute are modeled as `rel` elements with an additional `relend` element of role `targetanchor` (see link `Hyperlink 2` in figure 7 and table 2). Since relation ends can be extended by any GXL-attribute and since hyperedges of this kind are not restricted regard-

map. Note further that, for the time being, neither the GXL DTD nor the GXL Schema does check compliance to the latter restriction which has to be ensured by the HyGraph system.

ing the number of their targets, they allow modeling any relation of any valency. In other words, hyperedges are the preferred means of representing links, whether simple HTML links or more complex links of the XLink standard.

According to the hypertext graph model presented so far, Web-based hypertexts are represented as typed, attributed, directed, ordered hypergraphs supplemented by graph stratification and markup of the kernel hierarchy. This leaves out how to represent a page's internal content beyond its internal link structure. How this kind of graph embedding is performed is outlined in the next section.

3.2 Nesting hypertext document structures

The previous section focused on link structure representation. We have emphasized that it is necessary to distinguish layers for representing page internal and page external linkage. This leaves unspecified how to represent the remaining building blocks of page structure. At least, this relates to the D*ocument* O*bject* M*odel* (DOM) based representation of a webpage's HTML structure and to its linguistic structure. As far as we deal with the latter, we concentrate on the notion of logical (text) document structure as introduced in Power et al. (2003). An XML-based framework for dealing with logical text document structure is the C*orpus* E*ncoding* S*tandard* (CES; Ide et al. 2000) which we integrate in part into our GXL-based model. The basic tenet for doing this is to have an integrated, encompassing representation of a webpage's internal structure.

DOM related information is extracted from the HTML source of the corresponding input page. In many cases this source cannot be parsed directly because of malformed code. We use the HTMLParser[5] for parsing and correction in order to overcome this

[5] `http://htmlparser.sourceforge.net`

```
<!DOCTYPE gxl SYSTEM "http://www.gupro.de/GXL/gxl-1.0.dtd">
<gxl>
    <graph hypergraph="true" edgemode="directed" id="HyperGraph0">
        <node id="ModuleA">
            <graph id="InternalLinkStructureA1" hypergraph="false" edgemode="directed">
                <node id="AnchorA1"><!--...--></node>
                <node id="AnchorA2"><!--...--></node>
                <node id="AnchorA3"><!--...--></node>
                <node id="AnchorA4"><!--...--></node>
                <!--...-->
                <edge id="EdgeA1" from="AnchorA2" to="AnchorA3">
                    <attr name="types"><set><string>internallink</string></set></attr>
                </edge>
                <!--...-->
            </graph>
        </node>
        <node id="ModuleB">
            <graph id="InternalLinkStructureB1" hypergraph="false" edgemode="directed">
                <node id="AnchorB1"><!--...--></node>
                <node id="AnchorB2"><!--...--></node>
                <!--...-->
            </graph>
        </node>
        <node id="ModuleC">
            <graph id="InternalLinkStructureC1" hypergraph="false" edgemode="directed">
                <node id="AnchorC1"><!--...--></node>
                <!--...-->
            </graph>
        </node>
        <rel id="Hyperlink1">
            <attr name="types"><set><string>kernellink</string></set></attr>
            <relend direction="in" target="ModuleA" role="sourcepage" startorder="1"/>
            <relend direction="in" target="AnchorA1" role="sourceanchor"/>
            <relend direction="out" target="ModuleB" role="targetpage" endorder="1"/>
        </rel>
        <rel id="Hyperlink2">
            <attr name="types"><set><string>downlink</string></set></attr>
            <relend direction="in" target="ModuleA" role="sourcepage" startorder="2"/>
            <relend direction="in" target="AnchorA4" role="sourceanchor"/>
            <relend direction="out" target="ModuleB" role="targetpage" endorder="2"/>
            <relend direction="out" target="AnchorB1" role="targetanchor"/>
        </rel>
        <rel id="Hyperlink3">
            <attr name="types"><set><string>kernellink</string></set></attr>
            <relend direction="in" target="ModuleB" role="sourcepage" startorder="1"/>
            <relend direction="in" target="AnchorB2" role="sourceanchor"/>
            <relend direction="out" target="ModuleC" role="targetpage" endorder="1"/>
        </rel>
    </graph>
</gxl>
```

Table 2. Schematic outline of a sample GXL-based representation of a website (dots indicate omitted content – note that in this and subsequent examples we use descriptive IDs which in runtime experiments are replaced by prefixed numbers)

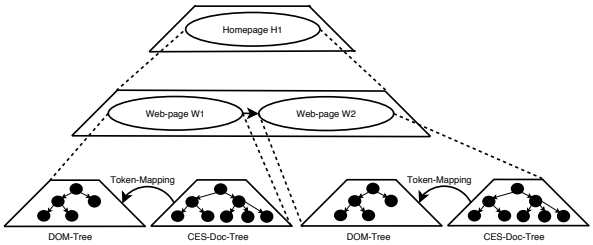

Figure 8. Integrated representation of DOM and LDS structure

problem. It provides an interface to the output DOM which GXL allows to represent as a directed rooted tree. We embed this tree into the node model of the focal page (see table 3 and figure 8). The DOM tree is the main source for deriving a page's internal link structure.

The second level of structure formation concerns the linguistic document structure of a webpage which we assume, for the sake of simplicity, to be representable as a labeled tree – this is, of course, an oversimplification, but serves as a working definition. We follow the approach of Power et al. (2003) and thus represent, amongst others, tokens, sentences, paragraphs and sections as part of a webpage's logical (text) document structure.

For various reasons, the extraction of this linguistic information from a webpage is not trivial. HTML possesses some basic means to represent document structure: for example, `H`-tags can be used to denote headlines and `P`-tags to mark paragraphs. But HTML lacks elements needed for explicitly tagging linguistic elements as, for example, sentences and tokens. Beside this insufficient expressiveness, another drawback is the *tag abuse* problem Barnard et al. 1995) which occurs when HTML tags are misused for layout purposes. Someone might, for example, use a headline tag to highlight a phrase in continuous text. On the other hand, the headline of a chapter could be highlighted by means of a bold

font without using a headline tag. Instead of going into the details of these problems when extracting document structure from DOM trees, we rather discuss the question of how to integrate the latter with representations of logical text document structure. In GXL, both structures can be represented as graphs. However, it would be insufficient not to account for their mapping. Because of differences in scope, we do not map their inner nodes or try to order or even to nest them, but rather focus on a mapping of their elementary text tokens only.[6] We do that by mapping each token of a page's text content model to the most specific node of the DOM tree to which it belongs. Figure 9 illustrates this mapping. In terms of a simplified GXL encoding, this example is outlined in table 3. The internal structure of `Module1` is represented by an additional embedded graph. This graph itself contains two embedded graphs which represent its DOM and logical document structure. Finally, the token-based mapping is manifested by a third graph.

So far, we have augmented our hypertext graph model by means of three component graphs which are nested into the nodes representing the pages whose link, DOM and linguistic structure they model. What is missing is an account of the fact that websites are hypertext documents which allow easy editing and modifications without necessarily losing their object identity. That is, we need to consider the revision process of (logically) the same website. This is outlined in the next section.

3.3 Time-aligned website representations

Web-based hypertexts are dynamic entities which preserve their "object identity" although they may change their gestalt dramatically during their lifespan. Above, the example of a conference

[6] It is easy to see that sentences may contain HTML-lists as list items can obviously contain sentences so that we cannot nest a webpage's logical document structure into its DOM structure nor the other way round.

```
<gxl>
    <graph hypergraph="true" edgemode="directed" id="HyperGraph0">
        <node id="Module1">
            <graph id="DOM-Tree1">
                <attr name="type"><enum>DOM</enum></attr>
                <node id="tag1"><!--...[body]...--></node>
                <node id="tag2"><!--...[h1]...--></node>
                <node id="html-text-1"><!--...[Conference 2005]...--></node>
                <edge from="tag1" to="tag2"/>
                <edge from="tag2" to="html-text-1"/>
                <!--...-->
            </graph>
            <graph id="CES-Doc1">
                <attr name="type"><enum>CES</enum></attr>
                <node id="node1"><!--...[body]...--></node>
                <node id="node2"><!--...[tok]...--></node>
                <node id="ces-orth1"><!--...[Conference]...--></node>
                <node id="ces-orth2"><!--...[2005]...--></node>
                <!--...-->
            </graph>
            <graph id="CES_DOM_Mapping1">
                <attr name="type"><enum>Mapping</enum></attr>
                <edge from="ces-orth1" to="html-text-1"/>
                <edge from="ces-orth2" to="html-text-1"/>
                <!--...-->
            </graph>
            <graph id="InternalLinkStructure1" hypergraph="false" edgemode="directed">
                <attr name="type"><enum>Linkage</enum></attr>
                <!--...-->
            </graph>
        </node>
        <!--...-->
    </graph>
</gxl>
```

Table 3. A sample nesting of webpage structure (dots indicate omitted content)

website was given, where its gestalt ranged from a single page at the time of its creation to possibly several hundred pages as the conference event approaches. At least, the following types of changes interrelating the interleaving website revisions can be distinguished when primarily focusing on webpages:[7]

- A `minor change` of a webpage typically concerns the correction of spelling mistakes or minor reformulations of its wording.
- A `significant change` of a webpage occurs when content

[7] The following listing does not claim to be a complete list of possible website changes.

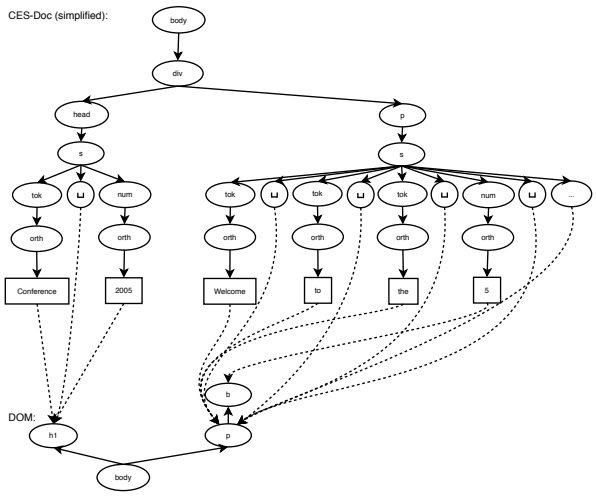

Figure 9. Mapping between text-tokens of DOM and CES representation

is added, removed or rearranged within the page.

- A `layout change` occurs when its layout is changed without actually touching its content.
- The deletion of a webpage is encoded as `deletion`. Analogously, the insertion of a webpage in a subsequent stage of a website's lifespan is encoded as `insertion`.
- A `replacement` of a webpage occurs if its content changes completely.
- The case of a `webpage movement` without replacement occurs if only the URL is changed.
- A `change of link structure` may have its source in webpages linking to the focal one. But also outgoing hyperlinks may have changed.

- The `movement of an entire home page or website` is a special case of a webpage movement. Regarding this type, it is assumed that the structure of the home page itself is not significantly affected.

In the previous sections, we have presented an integrated model of different levels of structure formation starting from a website's link structure down to the DOM structure of elementary pages. These representations are snapshots of Web-based hypertexts at certain points in time. In order to represent the order of these snapshots, we add a further representation layer on top of the existing ones. That is, we introduce a graph whose nodes denote website representations at certain points in time. The chronological ordering of these points in time is mapped by means of an additional directed graph.

The next step is to type the modifications that interrelate neighboring snapshots. We utilize the list of types of modifications presented above. If, for example, the content of a webpage has slightly changed, the respective website representations of the same website are interlinked by a `rel`(ation) of type `minor change`.[8] In the case of deletions and insertions simple `rel`s (i.e., hyperedges) each with only one `rel`(ation)`end` are used instead (see table 4).

This representation does, of course, not express the modification in detail, but it should be sufficient to quickly locate the places where changes occurred in order to analyze them separately. Figure 10 shows an example of a chronologically ordered representation of hypertext snapshots. This example can be encoded in GXL as outlined in table 4.

[8]The automatic detection of such changes is coming into reach by means of the framework of graph similarity measuring (Mehler et al. 2005).

```
<gxl>
    <graph id="snapshots_homepage_h1">
        <node id="snapshot_homepage_h1_2005-08-10">
            <graph id="document_network1">
                <node id="webpage_1_of_2005-08-10"/>
                <node id="webpage_2_of_2005-08-10"/>
                <!--...-->
            </graph>
        </node>
        <node id="snapshot_homepage_h1_2005-09-10">
            <graph id="document_network2">
                <node id="webpage_1_of_2005-09-10"/>
                <node id="webpage_2_of_2005-09-10"/>
                <!--...-->
            </graph>
        </node>
        <rel id="Hyperlink1">
            <attr name="types"><set><string>minor change</string></set></attr>
            <relend direction="in" target="webpage_1_of_2005-08-10" role="source"/>
            <relend direction="out" target="webpage_1_of_2005-09-10" role="target"/>
        </rel>
        <rel id="Hyperlink2">
            <attr name="types"><set><string>deletion</string></set></attr>
            <relend direction="in" target="webpage_2_of_2005-08-10" role="source"/>
        </rel>
        <rel id="Hyperlink3">
            <attr name="types"><set><string>insertion</string></set></attr>
            <relend direction="in" target="webpage_2_of_2005-09-10" role="source"/>
        </rel>
    </graph>
</gxl>
```

Table 4. Schematic outline of a GXL-based website representation (dots indicate omitted content)

4 Towards explorations of linguistic regularities sensitive to hypertext structure

Following the line of argumentation in Mehler (2005) and utilizing the representation model presented so far, we can now refer to website structure as a resource for (i) narrowing down the scope of linguistic pattern exploration and (ii) specifying additional constraints on those events which count as occurrences, co-occurrences, repetitions etc. In order to do that, the concept of a *domain* and, based on that, of a *data pool* have to be defined analogously to Mehler (2005).

In the present context, the notion of a domain is used to classify spans of the logical *hypertext* document structure of websites

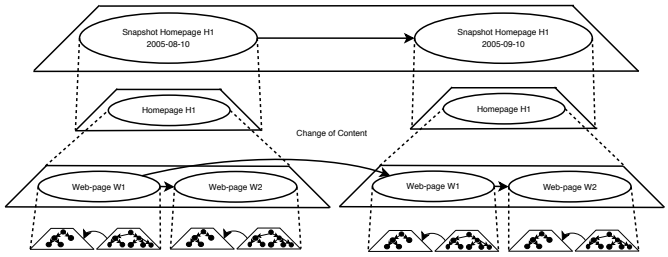

Figure 10. A time ordered website representation

and webpages as well as of the logical *text* document structure of the latter. Consequently, a domain equals, for example, a *module* type, a *Web document* type or a *document network* type. As we focus in this paper on expression units of Web-based communication, domains are seen to be additionally exemplified by the types *website*, *webpage* and all types of building blocks of the DOM and logical text document structure of webpages (e.g., *table*, *paragraph* and *sentence*). Finally, domains are seen to also include any type of spans as they are defined by parts of the kernel hierarchy and of the various levels of structure formation of single pages (e.g., *left subtree of the leader* or *third level of the right subtree of the leader* or *source and target page of an across link*). Thus, domains are used to type website spans in which linguistic data (e.g., co-occurrences) is observed.[9]

In the following definition, these types of spans of websites are referred to as elements d of the set of domains D. Further, if $S(C)$ is the set of all segments of the *webgenre document bank* C according to some segmentation procedure (segmenting, for example, websites into their pages and these pages into their sections, paragraphs and sentences etc.) and $s \in S(C)$ is a segment of type $d \in D$, then this is symbolized as $s \models_{S(C)} d$. If $\boldsymbol{a} \in T$ is a to-

[9]Note that we represent websites by means of GXL which is data-oriented and thus does not directly allow to specify domains using the XPath language.

ken (i.e., a (lexical) text position) instantiating (i.e., mapped onto the) type $a \in V$, we symbolize this as $\boldsymbol{a} \models_T a$. T and V are the set of tokens and types, respectively, so that $T(s)$ and $V(s)$ are analogously the set of tokens and types of the segment s. We now redefine definition 1 of Mehler (2005) and extend it in order to make it applicable to websites:

Definition 1. Let C be a webgenre document bank and $d \in D$ a domain with the set of instances $d(C) = \{s \in S(C) \mid s \models_{S(C)} d\}$ in C. The set of all co-occurrences of any types in segments of the domain d is $\Omega_C^d = \{(\boldsymbol{a}, \boldsymbol{b}) \mid \exists s \in d(C) : \boldsymbol{a}, \boldsymbol{b} \in T(s) \wedge \boldsymbol{a} \trianglelefteq \boldsymbol{b}\}$. The relation $\trianglelefteq$ maps the syntagmatic order of the textual content of the elements (i.e., websites) of C. On the level of websites, this order is based on the depth first order of their component pages according to the kernel hierarchy. On the level of webpages, it is based on the syntagmatic order of their text content. $\boldsymbol{a} \trianglelefteq \boldsymbol{b}$ means that $\boldsymbol{a}$ is a text position (i.e., a token) which linearly occurs before text position $\boldsymbol{b}$. With the help of Ω_C^d several sets can be derived:

1. $\Omega_C^d|_{(a,b)} = \{(\boldsymbol{a}, \boldsymbol{b}) \in \Omega_C^d \mid \boldsymbol{a} \models_T a \wedge \boldsymbol{b} \models_T b\}$ is the set of all co-occurrences of $a, b \in V$ in segments of the domain d, in which a occurs before b.

2. $\Omega_C^d|_{\{a,b\}} = \Omega_C^d|_{(a,b)} \cup \Omega_C^d|_{(b,a)}$ is the set of all co-occurrences of $a, b \in V$ in segments of domain d irrespective of their syntagmatic order.

3. $\Omega_C^d|^x = \{(\boldsymbol{a}, \boldsymbol{b}) \in \Omega_C^d \mid \boldsymbol{a}, \boldsymbol{b} \in T(x)\}$ is the set of all co-occurrences of any types in segment x of the domain d.

4. $\Omega_C^d|^x_{(a,b)} = \{(\boldsymbol{a}, \boldsymbol{b}) \in \Omega|_{(a,b)} \mid \boldsymbol{a}, \boldsymbol{b} \in T(x)\}$ is the restriction of $\Omega_C^d|_{(a,b)}$ to x. Accordingly, $\Omega_C^d|^x_{\{a,b\}} = \Omega_C^d|^x_{(a,b)} \cup \Omega_C^d|^x_{(b,a)}$.

5. $h_{ij} = |\{\boldsymbol{a} \mid \exists (\boldsymbol{b}, \boldsymbol{c}) \in \Omega_C^d|^{x_j} : \boldsymbol{a} \models_T a \wedge (\boldsymbol{a} = \boldsymbol{b} \vee \boldsymbol{a} = \boldsymbol{c})\}|$ is the frequency of $a_i \in V$ in segment x_j of domain d.

Ω_C^d and any set derived from it according to the latter specifications is called *data pool induced by* the corpus C, the domain d and possibly

some additional restrictions separated by |. □

Definition 1 is easily extended by additional co-occurrence restrictions. The restriction which is mostly applied in this context is the frequency restriction used to rule out *hapax legomena* and other low frequency items. Another frequently used restriction refers to the syntagmatic distance of the units to be viewed as co-occurring. These and related restrictions are not formalized in the present paper – we leave that to future work.

Data pools according to definition 1 work as filters which make accessible the linguistic information of Web-based communication as it is distributed over websites. The aim is to make it accessible to the various tasks of exploratory corpus analysis and machine learning by preserving restrictions as they result from the possibly genre-specific structuring of websites. Following this line of argumentation, co-occurrence analyses, for example, no longer need to be restricted to the textual content of *single* pages, but may include co-occurrences of items belonging to different but neighboring pages of the same level of the kernel hierarchy. To give another example: Co-occurrence analyses may be solely based on pages which are interlinked by means of across links. As websites are characterized by the phenomenon of discontinuous manifestation (see section 2) and related aspects of informational uncertainty, such an approach is indispensable when analyzing dependencies of linguistic items which, though they deal with the same topic or manifest the same function, are nevertheless distributed over different pages. This is exemplified by a conference website (e.g., `http://www.ht04.org/`) in which the (conference) program section is distributed over several webpages (i.e., three pages in the case of the latter example) so that there is, for example, no co-occurrence on any of these pages of the types *paper* and *keynote* (except for the menu). The aim of definition 1 is thus to soften or even neutralize such limitations in a way which is grounded in the underlying webgenre structure model.

5 Conclusion

In this paper, we have presented a GXL-based model for the representation of the link structure of websites, the nested structure of their constitutive pages and the alignment of their successive snapshots. This was proposed as a preliminary step to automatically analyzing and representing webgenres as they are instantiated by websites. In Mehler and Gleim (2005), the distribution of hypertext graphs of the genre of conference websites is analyzed based on this framework. In Mehler et al. (2005), the present framework is utilized to derive an algorithm for unsupervised graph learning. In this paper it is demonstrated that the link and DOM structure of websites and pages, respectively, are valuable sources for hypertext categorization. Improvements in this area hinge on improving hypertext representation. As has been shown, this task poses a lot of problems which, we believe, can only be adequately solved by means of machine learning methods *grounded in a webgenre model*. Future work will address these induction methods and their grounding in more detail. Analogously to the algorithm proposed in Mehler et al. (2005), these methods will be settled in the framework of unsupervised graph learning.

References

Amitay, E., Carmel, D., Darlow, A., Lempel, R. and Soffer, A. (2003).The connectivity sonar: Detecting site functionality by structural patterns. *Proceedings of the 14th ACM conference on Hypertext and Hypermedia*, 38-47.

Barnard, D. T., Burnard, L., DeRose, S. J., Durand, D. G. and Sperberg-McQueen, C. (1995). Lessons for the World Wide Web from the text encoding initiative. *Proceedings of the Fourth International WWW Conference "The Web Revolution"*.

Baroni, M. and Bernardini, S. (2004). BootCaT: Bootstrapping corpora and terms from the Web. *Proceedings of LREC 2004*, 1313-1316.

Biber, D. (1995). *Dimensions of register variation: A cross-linguistic comparison*, Cambridge: CUP.

Botafogo, R. A., Rivlin, E. and Shneiderman, B. (1992). Structural analysis of hypertexts: Identifying hierarchies and useful metrics. *ACM Transactions on Information Systems* 10(2), 142-180.

Chakrabarti, S. (2002). *Mining the Web: Discovering knowledge from hypertext data*, San Francisco: Morgan Kaufmann.

Crowston, K. and Kwasnik, B. (2003). Can document-genre metadata improve information access to large digital collections? *Library Trends*.

Crowston, K. and Williams, M. (1999). The effects of linking on genres of Web documents. *Proceedings of the Hawai'i International Conference on System Science*.

Crowston, K. and Williams, M. (2000). Reproduced and emergent genres of communication on the World Wide Web. *The Information Society* 16(3), 201-216.

Dillon, A. and Gushrowski, B. (2000). Genres and the Web: Is the personal home page the first uniquely digital genre? *Journal of the American Society of Information Science* 51(2), 202-205.

Eggins, S. (1994). *An introduction to systemic functional linguistics*, London: Continuum.

Eiron, N. and McCurley, K. (2003). Untangling compound documents on the Web. *Proceedings of the 14th ACM Conference on Hypertext and Hypermedia*, 85-94.

Firth, D. and Lawrence, C. (2003). Genre analysis in information systems research. *Journal of Information Technology Theory and Application* 5(3), 63-87.

Furner, J., Ellis, D. and Willett, P. (1996). The representation and comparison of hypertext structures using graphs. In Agosti, M. and Smeaton, A. (eds.) *Information Retrieval and Hypertext*, 75-96.

Getoor, L. (2003). Link mining: A new data mining challenge. *SIGKDD Explorations Newsletter* 5(1), 84-89.

Gleim, R. (2005). HyGraph: Ein Framework zur Extraktion, Repräsentation und Analyse webbasierter Hypertexte. In Fisseni, B., Schmitz, H., Schröder, B. and Wagner, P. (eds.) *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen*, 42-53.

Halliday, M. and Hasan, R. (1989). *Language, context, and text: Aspects of language in a socialsemiotic perspective*, Oxford: OUP.

Heinemann, W. (2000). Textsorte – Textmuster – Texttyp. In Brinker, K., Antos, G., Heinemann, W. and Sager, S. F. (eds.) *Text- und Gesprächslinguistik. Linguistics of text and conversation*, Berlin: de Gruyter, 507-523.

Ide, N., Bonhomme, P. and Romary, L. (2000). Xces: An XML-based standard for linguistic corpora. *Proceedings of LREC 2002*, 825-830.

Jakobs, E. (2003). Hypertextsorten. *Zeitschrift für germanistische Linguistik* 31(2), 232-252.

Keller, F. and Lapata, M. (2003). Using the Web to obtain frequencies for unseen bigrams. *Computational Linguistics* 29(3), 459-484.

Kilgarriff, A. and Grefenstette, G. (2003). Introduction to the special issue on the Web as corpus. *Computational Linguistics* 29(3), 333-347.

Martin, J. (1992). *English text. System and structure*, Amsterdam: Benjamins.

Mehler, A. (2005). Preliminaries to an algebraic treatment of lexical associations. *Proceedings of the Workshop Learning and Extending Lexical Ontologies at ICML 2005.*

Mehler, A., Dehmer, M. and Gleim, R. (2004). Towards logical hypertext structure – a graph-theoretic perspective. *Proceedings of I2CS '04*, 136-150.

Mehler, A. and Gleim, R. (2005). Polymorphism in generic Web units. A corpus linguistic study. *Proceedings of Corpus Linguistics 2005*, available online at `http://www.corpus.bham.ac.uk/PCLC/`.

Mehler, A., Gleim, R. and Dehmer, M. (2005). Towards structure-sensitive hypertext categorization. *Proceedings of the 29th Annual Conference of the German Classification Society.*

Melnikov, O., Tyshkevich, R., Yemelichev, V. and Sarvanov, V. (1994). *Lectures on graph theory*, Mannheim: BI Wissenschaft.

Miller, G. and Charles, W. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes* 6(1), 1-28.

Orlikowski, W. and Yates, J. (1994). Genre repertoire: The structuring of communicative practices in organizations. *Administrative Science Quarterly* 39(4), 541-574.

Power, R., Scott, D. and Bouayad-Agha, N. (2003). Document structure. *Computational Linguistics* 29(2), 211-260.

Rehm, G. (2002). Towards automatic Web genre identification: A corpus-based approach in the domain of academia by example of the academic's personal homepage. *Proceedings of the Hawai'i International Conference on System Sciences*.

Resnik, P. and Smith, N. (2003). The Web as a parallel corpus. *Computational Linguistics* 29(3), 349-380.

Routledge, L., Bailey, B., van Ossenbruggen, J., Hardman, L. and Geurts, J. (2000). Generating presentation constraints from rhetorical structure. *Proceedings of the 11th ACM Conference on Hypertext and Hypermedia*, 19-28.

Santamaría, C., Gonzalo, J. and Verdejo, F. (2003). Automatic association of Web directories to word senses. *Computational Linguistics* 29(3), 485-502.

Storrer, A. (2002). Coherence in text and hypertext. *Document Design* 3(2), 156-168.

van Dijk, T. and Kintsch, W. (1983). *Strategies of discourse comprehension*, New York: Academic Press.

Ventola, E. (1987). *The Structure of social interaction: A systemic approach to the semiotics of service encounters*, London: Pinter.

Winter, A., Kullbach, B. and Riedinger, V. (2002). An overview of the GXL graph exchange language. In Diehl, S. (ed.) *Software Visualization*, 324-336.

Yates, J. and Orlikowski, W. (1992). Genres of organizational communication: A structurational approach to studying communications and media. *Academy of Management Review* 17(2), 299-326.

Yoshioka, T. and Herman, G. (2000). Coordinating information using genres. Technical Report, MIT Sloan School of Management.