Published in: Mehler, Alexander/Kühnberger, Kai-Uwe/Lobin, Henning/Lüngen, Harald/
Storrer, Angelika/Witt, Andreas (eds.): Modeling, Learning, and Processing of Text-
Technological Data Structures. - Berlin/Heidelberg: Springer, 2011. pp. 197-218.
(Studies in Computational Intelligence 370)

Chapter 11
Integrated Linguistic Annotation Models and Their Application in the Domain of Antecedent Detection

Andreas Witt, Maik Stührenberg, Daniela Goecke, and Dieter Metzing

Abstract. Seamless integration of various, often heterogeneous linguistic resources in terms of their output formats and a combined analysis of the respective annotation layers are crucial tasks for linguistic research. After a decade of concentration on the development of formats to structure single annotations for specific linguistic issues, in the last years a variety of specifications to store multiple annotations over the same primary data has been developed. The paper focuses on the integration of the knowledge resource *logical document structure information* into a text document to enhance the task of automatic anaphora resolution both for the task of candidate detection and antecedent selection. The paper investigates data structures necessary for knowledge integration and retrieval.

11.1 Introduction

Anaphora Resolution (AR) describes the process of identifying the correct antecedent for a given anaphoric element and, in general, consists of three steps: (1) identification of anaphoric elements, (2) creation of a candidate set for each anaphora and (3) detection of the correct antecedent from the candidate set. In this paper we will focus on the second and third step and we will investigate the question how to create an appropriate candidate set.

In recent approaches that define anaphora resolution as a pairwise decision, the candidate set is created by choosing all candidates that precede a given anaphora

Andreas Witt
Institut für Deutsche Sprache, Zentrale Forschung, R5, 6 - 13, D-68016 Mannheim, Germany
e-mail: witt@ids-mannheim.de

Maik Stührenberg · Daniela Goecke · Dieter Metzing
Bielefeld University, Faculty of Linguistics and Literary Studies, Universitätsstraße 25,
D-33615 Bielefeld, Germany
e-mail: {maik.stuehrenberg,daniela.goecke,
dieter.metzing}@uni-bielefeld.de

or by using a fixed search window (e.g. in terms of sentences) and by collecting all discourse entities in this window [e.g. 52, 41, 46, 58]. Taking all preceding candidates into account works well for small texts, however for long texts this might lead to large candidate sets. The definition of an appropriate size of the search window is important inasmuch as a small window leads to errors due to the fact that the search window does not cover the correct antecedent at all and as a large window leads to large candidate sets which increases the possibility of preferring a wrong candidate over the correct one (for a discussion of the window size's impact on precision and recall values see [52]). Furthermore the computational effort increases due to the large number of candidates.

We argue that the approach of a fixed search window is not appropriate for long texts and thus not for all text types but that the search window has to be *flexible* in order to include the correct antecedent but to exclude those candidates that are least likely. How can we decide on the likelihood of an antecedent candidate? Current approaches of anaphora resolution are learning based, i.e. the likelihood of antecedent candidates is trained on a set of positive and false examples. However, in these approaches the candidate set is either created by taking all preceding discourse entities into account or by using a fixed window; for a given candidate set the most likely candidate is chosen. In our approach we investigate constraints in order to create an appropriate search window. We decide against a fixed search window due to two reasons:

1. With a fixed search window only antecedents can be found that lie within the given window size.
2. The search window cannot be enlarged arbitrarily as the size of the candidate list has negative impact on the resolution process.

Previous corpus investigation shows that linear distance between anaphora and antecedent is an important factor when creating an antecedent candidate set. Figure 11.1 shows linear distance of anaphoras of pronominal as well as of non-pronominal type in the corpus under investigation. The majority of pronominal anaphoras find their antecedents at a small distance whereas non-pronominal anaphoras find their antecedents even across large distances: For 26.8% of all non-pronominal anaphoras, the antecedent is found at a distance of two or more paragraphs. These anaphoras form 20.9% of all anaphoras occurring in the corpus under investigation. Previous investigations of the same corpus regarding the size of the search windows focused on linear distance in terms of discourse entities rather than sentences or paragraphs. For 50% of the direct anaphoric relations and 55.78% of the indirect anaphoric relations, the anaphoric element finds it antecedent within a distance up to 15 discourse entities (see [17]).

In this paper we will investigate how to resolve those anaphoras whose candidates lie outside a fixed window of one paragraph or 15 discourse entities. We will investigate the impact of hierarchical structure, especially logical document structure (LDS), on anaphora resolution. Why logical document structure might help to resolve anaphoric relations? The term logical document structure refers to the structure of a text in the sense of its formal composition and is in contrast to the

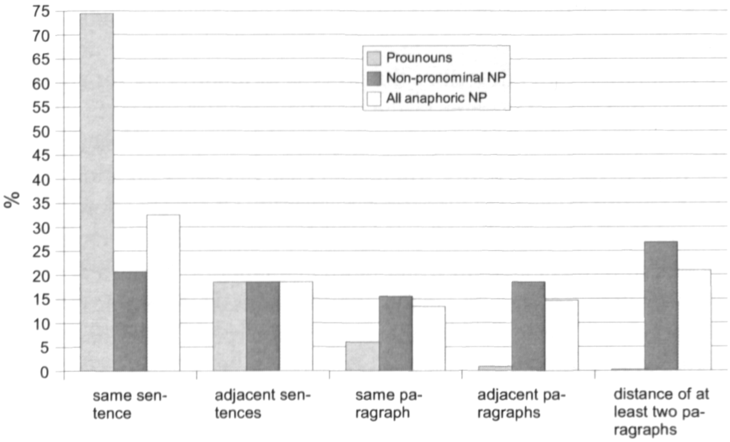

Fig. 11.1 Linear distance of anaphora and antecedent.

text's contentual composition, e.g. in terms of Introduction, Body or Conclusion. These contentual categories are realized by categories of the LDS, i.e. introduction, body and conclusion are realized as separate sections. The influence of the logical document structure on the choice of an antecedent might either be a direct influence on the markables (or antecedent life span) or an influence on the search window (see [15]). Thus, we investigate how to describe accessibility of antecedent candidates both in terms of linear as well as in terms of hierarchical distance. Accessibility is of special interest as linear distance between anaphora and antecedent might be large. The term *linear distance* is based on text structure and refers to syntagmatic distance between anaphora and antecedent in terms of words, discourse entities, sentences or paragraphs. *Hierarchical distance* describes distance between anaphora and antecedent on the basis of a hierarchical structure in terms of a tree structure, for example as found in discourse structure or logical document structure.

The remainder of the article is structured as follows: In Section 11.2, we provide the theoretical background of anaphora resolution and describe our categorial framework of anaphoric relations. In Section 11.3 we give an overview of logical document structure, describe the annotation of LDS and formulate our research questions regarding the use of LDS for anaphora resolution. In Section 11.4 we present annotation models that allow the investigation of different types of information and in Section 11.5 we will present the results of a corpus study investigating the use of LDS for anaphora resolution.

11.2 Anaphora Resolution

Anaphora Resolution (AR) describes the process of identifying for a given anaphoric element its correct antecedent in the previous textual context. The anaphoric element picks up its antecedent linguistically. In case of coreference, anaphora and antecedent refer to the same entity whereas in case of cospecification the anaphoric element picks up its antecedent linguistically but the two expressions are not coreferent. According to the relations that hold between the discourse entities, anaphora can be divided into direct anaphora and indirect anaphora. For direct anaphora, the antecedent is explicitly mentioned in the previous context (Example (1)) whereas for indirect anaphora the antecedent is not mentioned explicitly but has to be inferred from the context (Example (2)).

(1) I met a man yesterday. He told me a story.
(Example taken from [7], p. 414)

(2) I looked into the room. The ceiling was very high.
(Example taken from [7], p. 415)

Apart from the distinction of direct/indirect anaphora, discourse referents may be coreferent or not. In Example (1) the linguistic units "a man" and "he" are cospecified and refer to the same entity whereas "the room" and "the ceiling" in Example (2) do not although they are closely related due to world knowledge.

In this article we will investigate both direct and indirect anaphora as well as pronominal and definite description anaphora. We will focus on the question how to detect possible antecedent candidates from the set of discourse referents and how to select the correct one from the candidate set. The question how to detect possible candidates is of special interest as the linear distance between anaphora and antecedent might be large thus leading to a large set of candidates when using a fixed search window. In order to resolve anaphoric relations different types of information are needed. Information on discourse structure and referential accessibility is needed apart from information on POS, congruency, grammatical function and linear distance.

The corpus study is based on a corpus of German scientific articles that have been annotated manually for anaphoric relations. The annotation scheme comprises two primary relation types (direct and indirect anaphora) and a set of secondary relation types both for direct as well as for indirect anaphora. The annotation scheme is described in detail in [16]. The annotation has been done using the annotation tool SERENGETI [11] and has been checked for inter-annotator-agreement using kappa values [18]. Additional information for the resolution process has been added to the corpus by annotating the data automatically using the dependency parser MACHINESE SYNTAX[1] which provides lemmatization, POS information, dependency structure, morphological information and grammatical function. Based on this information, discourse entities have been detected automatically afterwards by identifying nominal heads (i.e. nouns or pronouns) and their pre-modifiers. Information

[1] http://www.connexor.eu/technology/machinese/machinesesyntax/

on logical document structure has been provided by the partner project C1 (see also Section 11.3.3).

11.3 Logical Document Structure

The aim of this section is to describe logical document structure as a structuring means of texts. LDS is a hierarchical structure: An article consists of sections which consist of subsections which consist of paragraphs. Furthermore, LDS describes the structure of a text – not its realization in a given medium, i.e. different realizations refer to the same structuring elements, e.g. paragraph boundaries or footnotes. Paragraph boundaries can be realized by line breaks with indentation or by blank lines (with or without following indentation). In print media, footnotes are often found at the bottom of the page whereas in hypertexts they are found at the end of the text and are linked via hyperlinks. In the next subsections we will provide a formal description of logical document structure and give an overview how LDS can be used for linguistic tasks.

11.3.1 What Is Logical Document Structure?

Formally, LDS forms a tree structure: Each section can contain several adjacent subsections, i.e. there are no overlapping arcs, and each subsection has exactly one parent section that contains it. Figure 11.2 shows the typical structure of a scientific article.

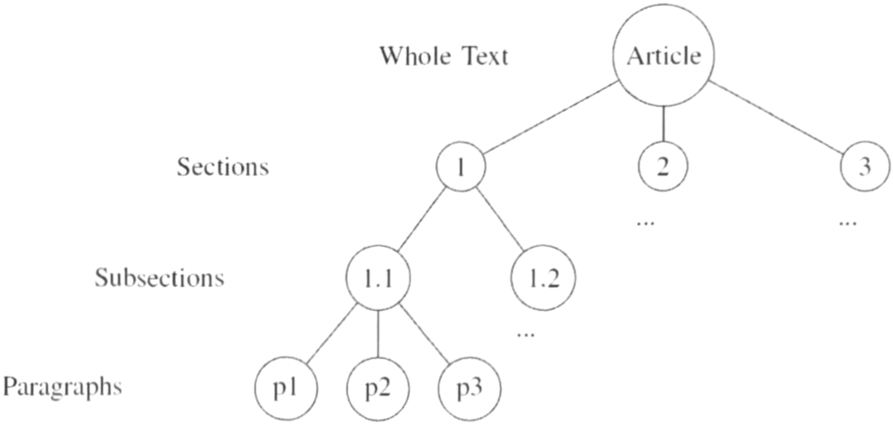

Fig. 11.2 Logical document structure of an article.

For a formal description of logical document structure as a tree, we follow the definition of [1]:

1. A single node by itself is a tree. This node is also the root of the tree.
2. Suppose n is a node and $T_1, T_2, ..., T_k$ are trees with roots $n_1, n_2 ..., n_k$ respectively. We can construct a new tree by making n the parent of nodes $n_1, n_2 ..., n_k$. In this tree n is the root and $T_1, T_2, ..., T_k$ are the subtrees of the root. Nodes $n_1, n_2 ..., n_k$ are called the *children* of node n.

(*ibid.* p. 75)

Knowledge about the expressiveness and complexity of LDS is important as it determines the means to describe and to annotate LDS in linguistic data. In terms of information modeling, the structuring elements shown in Figure 11.2 form a properly nested tree and thus follow the model of an ordered hierarchy of content objects (*OHCO*, cf. [10]). Each properly nested tree can be annotated using XML since the underlying formal model of XML is the tree – although extensions to this rule may apply according to the document grammar formalism that is used to define a specific markup language: e.g. DTDs are considered as tree-equivalent (extended) context-free grammars [cf. 22, p. 199] and [cf. 38, for a further discussion]. Any given XML annotation can be accessed by using XML tools: XPATH to traverse the tree and XSLT for further analyses. However, apart from their textual content, texts do contain objects that have to be converted into a tree structure in order to be annotated using XML, e.g. tables (cf. [31], p. 55ff). The application of LDS for linguistic tasks as well as its annotation for the corpus under investigation is described in the next subsections.

11.3.2 Application of Logical Document Structure for Linguistic Tasks

Information on logical document structure is applied for different linguistic tasks, e.g. language generation or genre detection. In this article we investigate the question whether LDS can be applied for the task of anaphora resolution.

Regarding language generation, [37] apply LDS (*abstract document structure* following the authors' terminology) in order to describe the abstract representation of a text – in contrast to its rhetorical structure or its realization (rendering). Whereas rhetorical structure is used to model the semantic content of a text, abstract document structure is used to model the hierarchical structure of textual entities. Abstract document structure is realized as a text using appropriate layout.

[34] describe the generation of referring expressions in hierarchically structured domains. [33] applies this framework for the domain of documents. Each document can be described as a hierarchical domain due to its hierarchical structure. For the task of language generation, a referring expression should allow for an easy identification of its referent. For hierarchically structured domains, information on the domain can be used to improve referring expressions in order to reduce the amount of search necessary to identify the referent. For a given document item, it

is necessary to identify the amount of information that is necessary to detect the referent, e.g. in order to refer appropriately to a picture item, information is needed whether the picture is located in the actual section or in another section.

Regarding anaphora resolution, the influence of the LDS on the choice of an antecedent might be either (a) a direct influence on the discourse entities (or antecedent life span), (b) an influence on the selection of a candidate according to the anaphora's and antecedent's position regarding the LDS or (c) an influence on the search window (comparable to different window sizes according to the NP type of the anaphora) (see also [15]). The first type is related to the fact that discourse entities "only serve as antecedents for anaphoric expressions within pragmatically determined segments" (cf. [52], p. 549).

Regarding LDS, previous investigation shows that some discourse entities are more prominent throughout the whole document than others, e.g. markables occurring in the abstract of a text might be accessible during the whole text whereas markables that occur in a list item or in a footnote-structure are less likely to be an antecedent for anaphoric elements in the main text. For a corpus of 4323 anaphoric relations 65.3% of all anaphora-antecedent-pairs are located in the same segment. Regarding the remaining anaphora-antecedent-pairs, we expect markables described in hierarchically higher elements (e.g. in a subsection) to be much more prone to finding their antecedents in structuring elements of a higher level (i.e. in a section) than in a preceding but hierarchically lower segment (i.e. in a preceding subsubsection). Thus, the influence on the search window may either enlarge the search window, i.e. the antecedent may be located outside the standard window (e.g. located in the whole paragraph or in a preceding one), or may narrow the search window, e.g. due to the start of a new chapter or section. Apart from defining an appropriate search window, the position of an antecedent candidate within a paragraph gives hints as to how likely that candidate is chosen as the correct one: 50.2% of the antecedents in the corpus are located paragraph-initial and 29.1% are located paragraph-final whereas only 20.2% are located in the middle of the paragraph. Thus, information on LDS might give information regarding the search window and for selecting the correct antecedent from a set of candidates (see also [47]).

In the following we will analyze how to apply these findings to antecedent detection and we will investigate the following research questions:

1. How are anaphora and antecedent located regarding LDS?
2. Does the position of the anaphora/the antecedent regarding LDS give hints for the antecedent choice?
3. Is it possible to define the search window by using information on the position of the anaphora/the antecedent?

In the next sections we will describe the annotation of LDS and the integration of different annotations layers for the task of analyzing their interrelationship and investigating the research questions.

11.3.3 XML-Annotation of Logical Document Structure

In order to investigate the influence of LDS on anaphora resolution we analyze a corpus regarding the research questions formulated above. The corpus under investigation has been annotated manually for anaphoric relations, additional information on lemmatization, POS, dependency structure, morphology and grammatical function as well as on discourse entities has been added afterwards (cf. Section 11.2). This information together with the annotation of the layer of logical document structure forms the basis for our analyses.

Apart from a set of newspaper articles that have been annotated in our project, we had the possibility to use an extensive set of annotations for scientific articles that have been annotated in the partner project C1. The annotation of the corpus data is based on an annotation scheme that has been defined by the partner projects C1 and B1 and which forms a subset of the DocBook annotation scheme with additional elements from (X)HTML. A detailed description of the annotation scheme as well as of the annotation procedure is given in [30, 28] and we will only give a brief overview here. For the annotation, a subset has been chosen from the complete set of elements from the DocBook standard (cf. [53]) which has been originally developed for technical documentation. The subset has been chosen in order to ease annotation by using only those elements that are needed for annotating the corpus of scientific articles. Another set of elements has been defined in order to describe elements that are not contained in the set of DocBook elements, e.g. elements for a table of contents which – in a standard DocBook creation process – is not annotated but created automatically from DocBook annotations. These elements are defined in a separate XML namespace. Another set of elements comprises XHTML-elements in order to describe e.g. link elements already annotated in the original corpus data. Altogether a set of 45 DocBook-elements and another 13 logical elements has been used for the annotation process. The annotation set thus comprises elements for describing the hierarchical structure of texts according to author, abstract, sections, paragraphs, footnotes, lists, list items, bibliography, tables, captions and the like.

The different annotation layers have been combined using markup unification which allows the combination of two XML annotation layers into a new XML instance [55]. The analysis of the research questions is based on the unified annotation data. This data is stored in a generic format that allows for creating different output formats, e.g. a candidate list (see Section 11.4.3), and for analyses using XSLT and XQuery. Different approaches for the integration of resources are presented in the next section.

11.4 Integration of Resources

In linguistic research often using only a single linguistic annotation layer is inadequate for dealing with specific tasks. This inadequacy does not only occur when one has to handle different linguistic levels, but can arise when working on a single representation level, e.g. [35] describe the problems when annotating multiword units on different lexical representation levels. Usually, annotation of linguistic representation levels is generated by linguistic resources, such as parsers, taggers, and the like. The integration of different resources is a crucial problem and, since the application of most linguistic resources results in heterogeneous output formats, i.e., XML instances following different document grammars that are only suitable for the given linguistic aspect this resource is aimed at, usually one encounters the problem of combining these different annotation layers that are all based on the same primary data. In this section we will present approaches to this problem.

11.4.1 Representation Formats

XML-based markup languages follow the formal model of a tree, i.e., the data that is structured by means of such a markup language is organized hierarchically as a tree (to be more specific: as a single tree) [59], similar to the above-mentioned OHCO model. Dealing with multi-dimensional annotation (i.e. multiple trees) and – as a result – with possibly overlapping structures is one of the key problems when working with XML-based annotation formats.

In the last years a variety of approaches has been developed to cope with overlapping structures. These proposals can be mainly divided into three categories: non-XML based approaches, XML-related approaches and XML-based approaches. The classic approach for dealing with multiple annotation layers is the use of separate documents or twin documents as [59] call them (if they share some annotation, the so-called sacred markup). [9] presents several formats that have been developed over the past years and that allow overlapping markup, starting from SGML's CONCUR feature [20] – a reimplementation approach named XCONCUR has been made by [21, 39, 56] –, over TEI milestones and fragmentation [5] and different standoff (i.e. the markup is separated from the primary data and stored in a separate document, [5, 51]) approaches up to specifications that leave the XML path, such as the Layered Markup and Annotation Language (LMNL, cf. [49, 8]) in conjunction with Trojan milestones following the HORSE (Hierarchy-Obfuscating Really Spiffy Encoding) or CLIX model. [44] discusses similar approaches for the formal representation of overlapping markup, adding colored XML [27] and the tabling approach described by [14] to the set of already stated proposals. Again, [59] compare state of the art in overlapping markup approaches, including alternatives to XML's data model (e.g. a directed acyclic graph structure (GODDAG, [45]) over the XML inherent tree) and its notation. In addition, the Prolog fact base approach discussed by [54, 55] or adding delay nodes to the XQuery 1.0 and XPath 2.0 Data Model (XDM) as virtual representation of nodes proposed by [29] allowing different nodes to share children describe other non-XML based specifications. Furthermore, XML-based specifications that follow the Annotation Graph paradigm [4], such as NITE [6], the Potsdamer Austauschformat für Linguistische Annotationen (PAULA, cf. [12, 13]), the Graph-based Format for Linguistic Annotations (GrAF, cf. [25]) developed by ISO/IEC TC37 or the Sekimo Generic Format (SGF) and its successor XStandoff (cf. section 11.4.2 and [47]) have been developed as well.

In case of using the classic approach of separate or twin documents the primary data (or source data, i.e. the textual data that is to be annotated) is saved together with a single annotation layer in separate files. Since only a single tree hierarchy is saved per file no overlapping structures occur. However, this approach might present problems in respect to the fact that the primary data is saved several times redundantly and analyzing relations between elements derived from different annotation layers may be cumbersome when dealing with multiple files without a linking element between the annotations. Although it is possible to use the character stream of the primary data as coordinates to align different annotation layers (cf. [55]), often changes to the primary data are introduced during the annotation process (in terms of added or deleted whitespace) raising further issues.

The Text Encoding Initiative proposes different XML-based solutions for dealing with complex markup (as multi dimensional markup is sometimes called): apart from stand-off markup, [5, chapters 16.9 and 20.4] there are milestone elements (empty elements that can be used as boundary markers, [5, chapter 20.2]) or fragmentations and joints (i.e., a series of elements is used in which each represent only a portion of the virtually larger element, [5, chapter 20.3]). In addition [57] describes a system that adopts TEI's feature structures [5, chapter 18] as a meta-format for representing heterogenous complex markup. The TEI tag set for feature structures supports a method for a general purpose data structure. A feature structure is built up of a `fs` element (feature structure) with an optional `type` attribute containing various instances of `f` elements (feature). Each `f` element bears a `name` attribute containing the feature's name. Possible child elements of the `f` element – apart from other feature structures (`fs`) – can be `binary`, `symbol`, `numeric` or `string` elements, allowing differentiation of the feature's value. Apart from the `string` element, which stores the value as its textual content, each of the named elements use a `value` attribute for this purpose. This simple mechanism can be used as a very general representation system. As an extension, feature and feature-value libraries can be established for re-using feature structure components in different instances. Re-entrant feature structures and Collections (complex feature structures) can be used as well. The connection between the primary data and the feature structure annotation(s) can be established by various linking mechanism described in the TEI guidelines (e.g. standoff techniques or XML ID/IDREF). For a concrete example of use cf. [57].

While most of the before-mentioned approaches target at the representation of multi-dimensional annotation, their usage in validating and analyzing multiple annotation layers is restricted: The non XML-based formats such as TeXMECS, LMNL or XCONCUR lack the support for XML's companion specifications such as XPath, XSLT or XQuery (although development has been started for an API for XCONCUR, [40] and query languages for overlapping markup have been proposed by [27, 23, 24, 2, 3]). Another problem arises by the fact that document grammars for validating overlapping markup structures are in the proposal state only, e.g. the Rabbit/Duck grammars proposed by [43] for GODDAG structures/TexMECS, XCONCUR-CL [39] or Creole (Composable Regular Expressions for Overlapping Languages etc., [50]) an extension to RELAX NG [26] developed in the LMNL community. For these reasons, if validating complex markup is an issue, it is easier to stick with XML-based approaches that can make use of the full scale of XML processing tools.

In the following section we will present the format developed in the Sekimo project for analyzing complex markup, the Sekimo Generic Format (SGF).

11.4.2 Sekimo Generic Format and XStandoff

The Sekimo Generic Format has been developed at Bielefeld University during the second phase of the Sekimo project. It is an XML-based successor of the Prolog fact base format for the storage and analysis of multiple annotated texts described in [55] and [19]. It follows a standoff annotation approach but combines all annotation levels that belong to the same primary data in a single XML instance, following the formal model of a multi-rooted tree. In fact, it is possible to store not only the annotations belonging to a single corpus item but several different corpus entries together with their respective annotation and metadata, the resources used during the annotation process and the document editing history.

The basic set-up of an SGF instance is quite simple: it consists of the primary data (either included in the instance or as a reference to an external file – or even multiple files when dealing with diachronic or multi-modal corpora), the segmentation (in terms of character position when dealing with texts, in terms of time spans or frames when dealing with non-textual primary data) and its annotation layers. In addition, optional metadata can be inserted at various positions and a log can be used for saving the document history (i.e. added, modified or deleted annotation elements).[2]

In contrast to other pivot formats such as TEI's feature structures, PAULA or GrAF, SGF tries to maintain as much of the original annotation format as possible, i.e. the only changes that are made concern the deletion of text nodes and the addition of the `sgf:segment` attribute that links to the corresponding `sgf:segment` element (via XML ID/IDREF) that defines the character span in the primary data storing the textual data part that is annotated by this specific element. A second distinguishing feature is that SGF usually stores all information, i.e. primary data, its segmentation and all respective annotation layers, in a single instance. SGF is flexible enough to allow in addition the use of multiple files or – at the opposite range – the storage of a whole corpus together with the resources used in its creation process in a single file. A graphical overview of an SGF instance is shown in Figure 11.3. As one can see, the original annotation layers (one containing POS annotation, the second a logical document structure) remain intact, including their

[2] The log functionality was primarily designed for the web-based annotation tool *Serengeti* but can be used in every other environment to track the changes that have been made to an SGF instance.

respective element hierarchy and attributes (the latter not shown in the simplified graphical overview).[3]

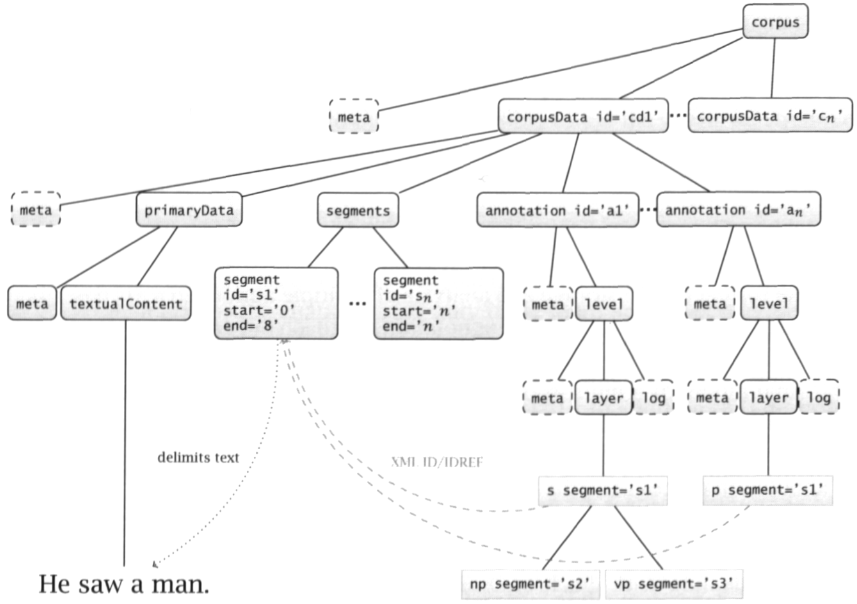

Fig. 11.3 A graphic overview of an SGF instance.

The format as such is designed as a set of XML schema files. In addition, there are converter scripts available as well, allowing the transformation of a single inline annotation into an SGF instance (`inline2SGF`), the merging of SGF annotation levels regarding the very same primary data input (`mergeSGF`), the deletion of SGF annotation levels (`removeLevel`) and a conversion from SGF to inline annotation using TEI milestone elements (`SGF2inline`).

We use SGF for two purposes: first, as a storage and exchange format that can be used in the web-based annotation tool *Serengeti* that has been developed in our project, and second, as a basis for corpus analysis, such as the relationship between elements of the logical document structure layer and anaphoras or antecedents respectively. For the latter it is possible to use standard XML related tools such as XSLT or XQuery to process and query SGF instances. Furthermore, it is possible to extract the reasonable parts from the culminated information stored in an SGF instance that are crucial for a specific task, which is shown in section 11.4.3.

A more detailed description of SGF can be found in [47], for a discussion of its use in the Anaphoric Bank project cf. [36] (in this volume).

The currently developed successor of SGF, called XStandoff (for both *extended* and *extensible* standoff format), introduces some changes to both the format and

[3] A real SGF instance is shown in [36] (in this volume).

the accompanied toolkit (see [48] for a detailed description), including the support for differentiating between containment and dominance relations in XML annotations (see [42] for a discussion) and an `all` namespace that can be used to subsume elements that are present in different annotation layers, amongst others. As a result, XStandoff is capable of expressing GODDAG structures (including cross-layer validation) while maintaining full compatibility to the XML standard. SGF and XStandoff are available under the GNU Lesser General Public License (LGPL v3)[4].

11.4.3 Antecedent Candidate List

Given the representation formats described in the previous section, the annotation layers for anaphoric relations and logical document structure (see Sections 11.2 and 11.3.3) are converted to SGF and can be analyzed afterwards. For the application domain of anaphora resolution, a set of candidates is identified via an XSLT script for each anaphoric relation and each anaphora together with its candidate set is stored in a candidate list (see Listing 11.1 for a shortened example of a candidate list).

The candidate list consists of several `semRel` elements each containing one anaphora element and several `antecedentCandidate` elements. Information on the relation type between the anaphora and its correct antecedent is stored as attribute information in the `semRel` element. The `anaphor` element describes properties of the anaphoric element as well as information on the correct antecedent, the `antecedentCandidate` elements store information on the antecedent candidates. All information is stored in terms of attributes. Congruency information is stored in the attributes `num` and `gen`. Additional information is given for part of speech (`pos`, `npType`), grammatical function (`syntax`), dependency structure (`dependHead`, `dependValue`) and lemma of head noun (`lemma`). The position of an element is described as position within the sentence (`sentencePos`), within the paragraph (`paraPos`) and in terms of its position regarding the whole document (`sentencePosition`, `position`). For all `antecedentCandidate` elements distance information in terms of sentences and discourse entities is added (`sentenceDistance`, `deDistance`). Information on the hierarchical structure of the respective candidates is stored as attribute information `c1-docPath`. The attribute `c1-docAnteHierarchy` stores information on the hierarchical relation between anaphora and antecedent candidate, in Listing 11.1 anaphora and correct antecedent are located in the same paragraph, i.e. their discourse entity elements are siblings. For the process of anaphora resolution each anaphora-candidate-pair is interpreted as a feature vector which is used for training a classifier (see also [41, 46, 58]). A detailed description of the candidate list creation process as well as of the XSLT processing script is given in [47].

[4] See `http://www.xstandoff.net` for downloads, example annotations and further details.

Listing 11.1 Example candidate list. Shortened and manually revised output

```
<?xml version="1.0" encoding="UTF-8"?>
<candidateList
  xmlns:sgf="http://www.text-technology.de/sekimo"
  xmlns:chs="http://www.text-technology.de/sekimo/chs"
  xmlns:c1-doc="http://www.text-technology.de/do-gi-docbook"
  <!-- [...] -->
  maxDeDistance="15"
  filename="ling-deu-010-sgf-c1doc.xml">
 <semRel relationID="sr71" type="cospecLink" subtype="ident"
     phorIDRef="de258" antecedentIDRefs="de249">
  <anaphor deID="de258" deType="nom" pos="N" syntax="@NH" lemma="
      monitoring-prozess" dependHead="w932" dependValue="mod"
      npType="pureNP" num="PL" gen="MSC" cas="DAT" sentencePos="
      6/6" paraPos="10/23" sentenceParaPos="2/4" position="241"
      sentencePosition="38" c1-docPath="/article[1]/sect1[3]/para
      [2]">Monitoring-Prozessen</anaphor>
  <!-- [...] -->
  <antecedentCandidate correctAntecedent="yes" deID="de249"
      deType="nom" pos="N" syntax="@NH" lemma="monitoring-prozess
      " dependHead="w914" dependValue="subj" npType="pureNP" num=
      "PL" gen="MSC" cas="NOM" sentencePos="2/4" paraPos="2/23"
      sentenceParaPos="1/4" position="233" sentencePosition="37"
      c1-docPath="/article[1]/sect1[3]/para[2]" deDistance="8"
      sentenceDistance="1" c1-docAnteHierarchy="siblings">
      Monitoring-Prozesse</antecedentCandidate>
  <!-- [...] -->
  <antecedentCandidate deID="de252" deType="nom" pos="N" syntax="
      @NH" lemma="aufmerksamkeits#fokus" dependHead="w910"
      dependValue="mod" npType="defNP" num="SG" gen="MSC" cas="
      DAT" sentencePos="4/4" paraPos="4/23" sentenceParaPos="1/4"
       position="235" sentencePosition="37" c1-docPath="/article
      [1]/sect1[3]/para[2]" deDistance="6" sentenceDistance="1"
      c1-docAnteHierarchy="siblings">m Aufmerksamkeitsfokus</
      antecedentCandidate>
  <!-- [...] -->
 </semRel>
</candidateList>
```

11.5 Results of a Corpus Study

The corpus annotated during the project has a total size of 14 documents, divided into six German scientific articles with complex document structure and eight German newspaper articles. The corpus comprises 3084 sentences with 55221 tokens and 11459 discourse entities. We've annotated 4185 anaphoric relations (3223 direct and 962 indirect).

The corpus under investigation consists of five German scientific articles, its size and information on anaphoric relations are given in Table 11.1. In our analyses we focus on semantic relations with only one antecedent due to the fact that relations with more than one antecedent only play a minor role (see column *#SemRels > 1*

Ante in Table 11.1). Pronominal anaphoras tend to find their antecedent at a small distance that almost always lies within a distance of 15 discourse entities (DE henceforth) therefore we focus our research questions on non-pronominal anaphoras (see Figure 11.4).

Table 11.1 Overview on the corpus.

Text	#Token	#DE	#Antecedent DE	#Anaphora DE	#Anaphoric relations	#Anaphoric rel. (1 Ante)	#Anaphoric rel. (>1 Ante)
ld-003	12423	2619	996	1347	1358	1311 (96.54%)	47 (3.46%)
ld-010	2248	501	139	174	183	177 (96.72%)	6 (3.28%)
ld-012	6467	1189	342	465	489	484 (98.98%)	5 (1.02%)
ld-014	9385	1529	424	496	500	488 (97.6%)	12 (2.40%)
ld-016	9286	1773	307	395	405	394 (97.28%)	11 (2.72%)
Σ	39809	7611	2208	2877	2935	2854 (97.24%)	81 (2.76%)

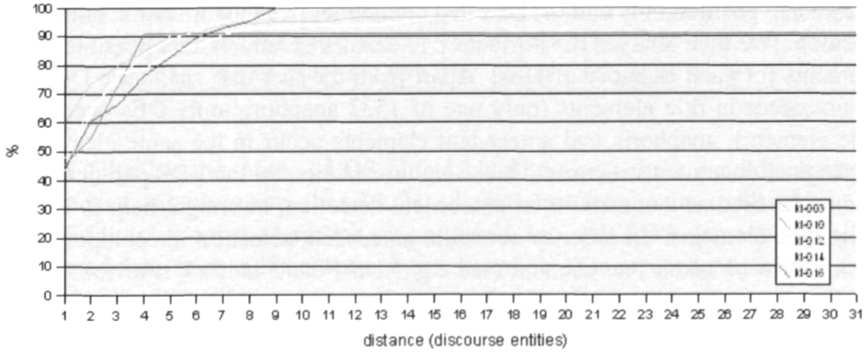

Fig. 11.4 Distance between pronominal anaphora and antecedent in discourse entities.

Distance information for non-pronominal anaphoras shows that linear distance is greater than for pronominal anaphoras (Figure 11.5). Both pronominal and non-pronominal anaphoras show fairly homogeneous behavior among the different texts which supports the assumption that distance information is stable among different texts of the same text type even if text length varies among these texts.

Figure 11.5 shows for a distance baseline of 15 discourse entities that – for the different texts – a minimum of 48.27% and a maximum of 61.81% of all anaphoras find their antecedents within this search window. We will now investigate our research questions:

1. How are anaphora and antecedent located regarding LDS?
2. Does the position of the anaphora/the antecedent regarding LDS give hints for the antecedent choice?

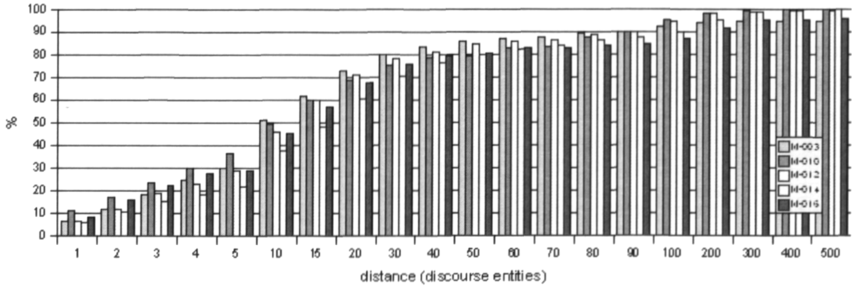

Fig. 11.5 Distance between non-pronominal anaphora and antecedent in discourse entities.

3. Is it possible to define the search window by using information on the position of the anaphora/the antecedent?

Regarding questions (1) and (2) we classify the discourse entities (DEs) occurring in the scientific articles into four categories: (1) DEs that are both in anaphora and in antecedent position, (2) DEs that are in anaphora position only, (3) DEs that are in antecedent position only and (4) DEs that are neither in anaphora nor in antecedent position. We then analyze the attribute `c1-docPath` and extract possible parent elements for each category of DEs. Apart from the fact that anaphoric DEs tend to not occur in title elements (only one of 1532 anaphoric-only DEs occurs in a title element), anaphoric and antecedent elements occur in the same elements of the logical document structure. Thus, sole information on the LDS parent element cannot be used as a constraint on antecedent detection, but might help to identify anaphoric elements. In order to constrain antecedent selection the complete LDS path has to be taken into account (see e.g. LDS-Filter3 below). Previous corpus evidence regarding the choice of DEs located in footnote elements or list items can be confirmed by the corpus under investigation. LDS information cannot be used as a hard constraint in a sense that the occurrence of a DEs in a given LDS structure prohibits the antecedent to be in antecedent position. Nevertheless, LDS can serve as a weak constraint when comparing structures of competing antecedent candidates.

Regarding question (3) we define a search window of 15 discourse entities to be the baseline and we compare different LDS filters against this baseline. Table 11.3 shows the results of the tests for anaphoric relations with non-pronominal anaphora. For the baseline we simply collect all discourse entities that are within a distance of no more than 15 discourse entities. As the annotation scheme used for corpus creation only allows non-pronominal discourse entities in antecedent position the number of candidates in each set might be smaller than 15. Therefore, we define candidate sets of exactly 15 elements by adding candidates to the baseline set. For each of the LDS filters we create candidate sets of exactly 15 elements, too.

The first LDS filter (Table 11.3: LDS-Filter1) selects antecedent candidates according to their values for the attribute `c1-docAnteHierarchy`. This attribute has different values according to the relationship of `c1-docPath`-values of anaphora and antecedent. The value *siblings* is chosen if the DE-elements of anaphora and antecedent have the same parent element and the value *same* describes LDS paths that contain exactly the same types of elements (e.g. /article[1]/sect1[1]/para[10] – /article[1]/sect1[3]/para[3]). The value *ante-ancestor* is chosen if the parent element of the antecedent DE is an ancestor to anaphora DE's parent element (e.g. /article[1]/sect1[1]/para[10] – /article[1]/sect1[1]/para[10]/emphasis[1]), the value *ana-ancestor* is chosen accordingly. If none of the above values hold, *yes* indicates the anaphora's LDS path to be longer than the ancestor's path, *no* indicates the opposite. LDS-Filter1 chooses candidates with values *sibling, same* and *yes* according to the corpus findings given in Table 11.2: *sibling, same* and *yes* cover most of the anaphoric relations.

Table 11.2 LDS-Hierarchy of anaphora and antecedent.

	ld-003	ld-010	ld-012	ld-014	ld-016
Total	1311	177	484	488	394
siblings	751 (57.28%)	83 (46.89%)	277 (57.23%)	278 (56.97%)	255 (64.72%)
same	276 (21.05%)	28 (15.82%)	154 (31.82%)	125 (25.61%)	97 (24.62%)
ante-ancestor	31 (2.36%)	4 (2.26%)	5 (1.03%)	4 (0.82%)	1 (0.25%)
ana-ancestor	10 (0.76%)	1 (0.56%)	1 (0.21%)	10 (2.05%)	10 (2.54%)
yes	168 (12.81%)	40 (22.6%)	37 (7.64%)	39 (7.99%)	22 (5.58%)
no	75 (5.72%)	21 (11.86%)	10 (2.07%)	32 (6.56%)	9 (2.28%)

LDS-Filter2 is the same as LDS-Filter1 but filters only those candidates whose DE-distance value is greater than 15, thus the baseline set remains unfiltered.

LDS-Filter3 keeps the baseline set unfiltered, too. All LDS elements *glosslist*[5] are filtered from the candidate set as these do not occur in antecedent position, candidates with values *sibling, same* and *yes* are chosen afterwards.

Table 11.3 Results of LDS filters.

	ld-003	ld-010	ld-012	ld-014	ld-016
#Anaphoric Relations	1153 (100%)	166 (100%)	380 (100%)	370 (100%)	265 (100%)
	Coverage for different test cases				
(1) Baseline: deDistance≤15	61.67%	59.64%	59.21%	47.57%	56.6%
(2) CL Size=15 (no LDS-Filter)	65.39%	60.24%	62.89%	51.35%	61.51%
(3) CL Size=15 (LDS-Filter1)	61.49%	61.45%	64.21%	52.16%	61.13%
(4) CL Size=15 (LDS-Filter2)	65.13%	60.84%	63.16%	52.43%	62.26%
(5) CL Size=15 (LDS-Filter3)	65.22%	62.05%	63.16%	52.43%	62.26%

[5] This element was introduced to annotate definition and glossary lists (containing glossary items and the respective definition) which may be found in some of the scientific documents. Since anaphoras should not occur between an anaphora in the running text and an antecedent in a glossary definition we can safely apply the filter.

Table 11.3 shows for each scientific article and for each of the test cases the amount of anaphoric relations for which the correct antecedent candidate is contained in the candidate list. The results show that LDS filters do only play a minor role in the creation of an appropriate candidate set. The antecedent candidate set of 15 elements with no LDS filters is only slightly outperformed by the LDS filters. In fact, LDS-Filter1 decreases the amount of correct antecedents found as it filters correct antecedents that would have been found within the search window. LDS-Filter3 filters candidates whose value of the `c1-docPath`-attribute never occur in antecedent position. However, as distance between anaphora and antecedent can be very large, filtering for single candidates does not much improve the coverage. We can draw the conclusion, that LDS as a hard constraint cannot close the gap to full coverage of antecedent candidates in long texts. We argue to enlarge the candidate set to an appropriate size (see Figure 11.5) and to apply weak constraints in order to choose the correct antecedent from the set of candidates, e.g. based on information as given in Table 11.2.

11.6 Conclusion

The research described in this chapter started with the initial assumption that typical language technological tasks would benefit from considering not only textual content but also additional information that quite often is available in digital documents. Unfortunately, however, the results of our investigations do support our initial assumptions only weakly. The minor effects found led us to the conclusion not to use logical document structure as an absolute constraint for the (non-)accessibility of anaphora antecedents but using it only as an additional resource that might improve the task of anaphora resolution slightly. Moreover, we believe that a whole bunch of additional information sources could be taken into account to improve applications of language technology. (see also [32]) To enhance the accessibility of the diverse information types we propose to make this information available together with the text document in a standardised way. XStandoff could be used as an annotation technique that allows doing this in a powerful way.

Acknowledgments

The work presented in this paper is part of the project A2 Sekimo of the distributed Research Group *Text-technological Modelling of Information* funded by the German Research Foundation. We thank the partner project C1 SemDok for providing their annotation layers of logical document structure.

References

[1] Aho, A.V., Hopcroft, J.E., Ullman, J.D.: Data Structures and Algorithms. Addison-Wesley, Reading (1983)

[2] Alink, W., Bhoedjang, R., de Vries, A.P., Boncz, P.A.: Efficient XQuery Support for Stand-Off Annotation. In: Proceedings of the 3rd International Workshop on XQuery Implementation, Experience and Perspectives, in Cooperation with ACM SIGMOD, Chicago, USA (2006)

[3] Alink, W., Jijkoun, V., Ahn, D., de Rijke, M.: Representing and Querying Multi-dimensional Markup for Question Answering. In: Proceedings of the 5th EACL Workshop on NLP and XML (NLPXML 2006): Multi-Dimensional Markup in Natural Language Processing, EACL, Trento (2006)

[4] Bird, S., Liberman, M.: Annotation graphs as a framework for multidimensional linguistic data analysis. In: Proceedings of the Workshop "Towards Standards and Tools for Discourse Tagging", Association for Computational Linguistics, pp. 1–10 (1999)

[5] Burnard, L., Bauman, S. (eds.): TEI P5: Guidelines for Electronic Text Encoding and Interchange. published for the TEI Consortium by Humanities Computing Unit, University of Oxford, Oxford, Providence, Charlottesville, Bergen (2007)

[6] Carletta, J., Evert, S., Heid, U., Kilgour, J.: The NITE XML toolkit: data model and query language. Language Resources and Evaluation 39(4), 313–334 (2005)

[7] Clark, H.: Bridging. In: Johnson-Laird, P.N., Wason, P.C. (eds.) Thinking: Readings in Cognitive Science, pp. 411–420. Cambridge University Press, Cambridge (1977)

[8] Cowan, J., Tennison, J., Piez, W.: LMNL update. In: Proceedings of Extreme Markup Languages, Montréal, Québec (2006)

[9] DeRose, S.J.: Markup Overlap: A Review and a Horse. In: Proceedings of Extreme Markup Languages (2004)

[10] DeRose, S.J., Durand, D.G., Mylonas, E., Renear, A.H.: What is text, really? Journal of Computing in Higher Education 1(2), 3–26 (1990)

[11] Diewald, N., Goecke, D., Stührenberg, M., Garbar, A.: Serengeti - webbasierte annotation semantischer relationen. appears in: LDV-Forum GLDV-Journal for Computational Linguistics and language Technology (2009)

[12] Dipper, S.: Xml-based stand-off representation and exploitation of multi-level linguistic annotation. In: Proceedings of Berliner XML Tage 2005 (BXML 2005), Berlin, Deutschland, pp. 39–50 (2005)

[13] Dipper, S., Götze, M., Küssner, U., Stede, M.: Representing and Querying Standoff XML. In: Rehm, G., Witt, A., Lemnitzer, L. (eds.) Datenstrukturen für linguistische Ressourcen und ihre Anwendungen. Data Structures for Linguistic Resources and Applications. Proceedings of the Biennial GLDV Conference 2007, pp. 337–346. Gunter Narr Verlag, Tübingen (2007)

[14] Durusau, P., O'Donnel, M.B.: Tabling the overlap discussion. In: Proceedings of Extreme Markup Languages (2004)

[15] Goecke, D., Witt, A.: Exploiting logical document structure for anaphora resolution. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy (2006)

[16] Goecke, D., Stührenberg, M., Holler, A.: Koreferenz, Kospezifikation und Bridging: Annotationsschema. Research group Text-technological Modelling of Information, Universität Bielefeld, Fakultät für Linguistik und Literaturwissenschaft, & Georg-August-Universität Göttingen, Seminar für Deutsche Philologie (2007)

[17] Goecke, D., Stührenberg, M., Wandmacher, T.: A hybrid approach to resolve nominal anaphora. LDV Forum – Zeitschrift für Computerlinguistik und Sprachtechnologie 23(1), 43–58 (2008)

[18] Goecke, D., Stührenberg, M., Witt, A.: Influence of text type and text length on anaphoric annotation. In: ELRA (ed.) Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, Morocco (2008)

[19] Goecke, D., Lüngen, H., Metzing, D., Stührenberg, M., Witt, A.: Different views on markup. distinguishing levels and layers. In: Witt, A., Metzing, D. (eds.) Linguistic Modeling of Information and Markup Languages. Contributions to Language Technology, pp. 1–21. Springer, Heidelberg (2010)

[20] Goldfarb, C.F.: The SGML Handbook. Oxford University Press, Oxford (1991)

[21] Hilbert, M., Schonefeld, O., Witt, A.: Making CONCUR work. In: Proceedings of Extreme Markup Languages (2005)

[22] Hopcroft, J., Motwani, R., Ullman, J.: Introduction to Automata Theory, Languages, and Computation, 2nd edn. Addison-Wesley, Reading (2000)

[23] Iacob, I.E., Dekhtyar, A.: Processing XML documents with overlapping hierarchies. In: JCDL 2005: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries, pp. 409–409. ACM Press, New York (2005)

[24] Iacob, I.E., Dekhtyar, A.: Towards a query language for multihierarchical xml: Revisiting xpath. In: Proceedings of the 8th International Workshop on the Web & Databases (WebDB 2005), Baltimore, Maryland, USA, pp. 49–54 (2005)

[25] Ide, N., Suderman, K.: GrAF: A Graph-based Format for Linguistic Annotations. In: Proceedings of the Linguistic Annotation Workshop, Association for Computational Linguistics, Prague, Czech Republic, pp. 1–8 (2007)

[26] ISO/IEC 19757-2:2003, Information technology – Document Schema Definition Language (DSDL) – Part 2: Regular-grammar-based validation – RELAX NG (ISO/IEC 19757-2). International Standard, International Organization for Standardization, Geneva (2003)

[27] Jagadish, H.V., Lakshmanany, L.V.S., Scannapieco, M., Srivastava, D., Wiwatwattana, N.: Colorful XML: One hierarchy isn't enough. In: Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD 2004), pp. 251–262. ACM Press, New York (2004)

[28] Langer, H., Lüngen, H., Bayerl, P.S.: Text type structure and logical document structure. In: Proceedings of the ACL 2004 Workshop on Discourse Annotation, Barcelona, pp. 49–56 (2004), `http://www.uni-giessen.de/germanistik/ascl/dfg-projekt/pdfs/aclws.pdf`

[29] Le Maitre, J.: Describing multistructured XML documents by means of delay nodes. In: DocEng 2006: Proceedings of the 2006 ACM symposium on Document engineering, pp. 155–164. ACM Press, New York (2006)

[30] Lenz, E.A., Lüngen, H.: Dokumentation: Annotationsschicht: Logische Dokumentstruktur. Research group Text-technological Modelling of Information, Universität Dortmund, Institut für deutsche Sprache und Literatur, & Justus-Liebig-Universität Gießen, Fachgebiet Angewandte Sprachwissenschaft und Computerlinguistik (2004)

[31] Lobin, H.: Informationsmodellierung in XML und SGML. Springer, Heidelberg (2000)

[32] Metzing, D.: Diskurs-Anaphern. Texttechnologische Informationsmodellierung und benachbarte linguistische Forschungskontexte. In: Marello, C., Hölker, K. (eds.) Dimensionen der Analyse von Texten und Diskursen, LIT Verlag (to appear 2011)

[33] Paraboni, I.: Generating references in hierarchical domains: the case of document deixis. PhD thesis, Information Technology Research Institute, University of Brighton (2003)

[34] Paraboni, I., van Deemter, K., Masthoff, J.: Generating referring expressions: Making referents easy to identify. Computational Linguistics 33(2), 229–254 (2007)

[35] Pianta, E., Bentivogli, L.: Annotating Discontinuous Structures in XML: the Multiword Case. In: Proceedings of LREC 2004 Workshop on "XML-based richly annotated corpora", Lisbon, Portugal, pp. 30–37 (2004)

[36] Poesio, M., Diewald, N., Stührenberg, M., Chamberlain, J., Jettka, D., Goecke, D., Kruschwitz, U.: Markup infrastructure for the anaphoric bank: Supporting web collaboration. In: Mehler, A., Kühnberger, K.U., Lobin, H., Lüngen, H., Storrer, A., Witt, A. (eds.) Modelling, Learning and Processing of Text-Technological Data Structures. Springer, Berlin (2011)

[37] Power, R., Scott, D., Bouayad-Agha, N.: Document structure. Computational Linguistics 29(2), 211–260 (2003)

[38] Rizzi, R.: Complexity of context-free grammars with exceptions and the inadequacy of grammars as models for xml and sgml. Markup Languages – Theory & Practice 3(1), 107–116 (2001)

[39] Schonefeld, O.: XCONCUR and XCONCUR-CL: A constraint-based approach for the validation of concurrent markup. In: Rehm, G., Witt, A., Lemnitzer, L. (eds.) Datenstrukturen für linguistische Ressourcen und ihre Anwendungen. Data Structures for Linguistic Resources and Applications. Proceedings of the Biennial GLDV Conference 2007. Gunter Narr Verlag, Tübingen (2007)

[40] Schonefeld, O.: A simple API for XCONCUR. In: Proceedings of Balisage: The Markup Conference, Montréal, Québec (2008)

[41] Soon, W.M., Lim, D.C.Y., Ng, H.T.: A machine learning approach to coreference resolution of noun phrases. Computational Linguistics 27(4), 521–544 (2001)

[42] Sperberg-McQueen, C., Huitfeldt, C.: Markup discontinued discontinuity in texmecs, goddag structures, and rabbit/duck grammars. In: Proceedings of Balisage: The Markup Conference, Balisage Series on Markup Technologies, vol. 1 (2008)

[43] Sperberg-McQueen, C.M.: Rabbit/duck grammars: a validation method for overlapping structures. In: Proceedings of Extreme Markup Languages (2006)

[44] Sperberg-McQueen, C.M.: Representation of overlapping structures. In: Proceedings of Extreme Markup Languages (2007)

[45] Sperberg-McQueen, C.M., Huitfeldt, C.: GODDAG: A data structure for overlapping hierarchies. In: King, P., Munson, E.V. (eds.) PODDP 2000 and DDEP 2000. LNCS, vol. 2023, pp. 139–160. Springer, Heidelberg (2004)

[46] Strube, M., Müller, C.: A machine learning approach to pronoun resolution in spoken dialogue. In: ACL 2003: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, Morristown, NJ, USA, pp. 168–175 (2003)

[47] Stührenberg, M., Goecke, D.: SGF – an integrated model for multiple annotations and its application in a linguistic domain. In: Proceedings of Balisage: The Markup Conference, Montréal, Québec (2008)

[48] Stührenberg, M., Jettka, D.: A toolkit for multi-dimensional markup: The development of SGF to XStandoff. In: Proceedings of Balisage: The Markup Conference, Montréal, Québec, Balisage Series on Markup Technologies, vol. 3 (2009)

[49] Tennison, J.: Layered markup and annotation language (LMNL). In: Proceedings of Extreme Markup Languages, Montréal, Québec (2002)

[50] Tennison, J.: Creole: Validating overlapping markup. In: Proceedings of XTech 2007: The Ubiquitous Web Conference, Paris, France (2007)

[51] Thompson, H.S., McKelvie, D.: Hyperlink semantics for standoff markup of read-only documents. In: Proceedings of SGML Europe 1997: The Next Decade –Pushing the Envelope, Barcelona, pp. 227–229 (1997)

[52] Vieira, R., Poesio, M.: An empirically based system for processing definite descriptions. Computational Linguistics 26(4), 539–593 (2001)

[53] Walsh, N., Muellner, L.: Doc-Book: The Definitive Guide. O'Reilly, Sebastopol (1999)

[54] Witt, A.: Meaning and interpretation of concurrent markup. In: Proceedings of ALLC-ACH 2002, Joint Conference of the ALLC and ACH, Tübingen (2002)

[55] Witt, A., Goecke, D., Sasaki, F., Lüngen, H.: Unification of XML Documents with Concurrent Markup. Literary and Lingustic Computing 20(1), 103–116 (2005)

[56] Witt, A., Schonefeld, O., Rehm, G., Khoo, J., Evang, K.: On the lossless transformation of single-file, multi-layer annotations into multi-rooted trees. In: Proceedings of Extreme Markup Languages, Montréal, Québec (2007)

[57] Witt, A., Rehm, G., Hinrichs, E., Lehmberg, T., Stegmann, J.: SusTEInability of linguistic resources through feature structures. Literary and Linguistic Computing (2009) (to appear)

[58] Yang, X., Su, J., Zhou, G., Tan, C.L.: Improving pronoun resolution by incorporating coreferential information of candidates. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004), Barcelona, Spain (2004)

[59] Zacchiroli PMFVS: Towards the unification of formats for overlapping markup. New Review of Hypermedia and Multimedia 14(1):57–94 (2008)