Methods for the Semantic Analysis of Document Markup

Petra Saskia Bayerl

Harald Lüngen

Justus-Liebig-Universität
Gießen, Germany

{petra.s.bayerl|
harald.luengen}
@uni-giessen.de

Daniela Goecke

Andreas Witt

Universität Bielefeld
Bielefeld, Germany

{daniela.goecke|
andreas.witt}
@uni-bielefeld.de

ABSTRACT

We present an approach on how to investigate what kind of semantic information is regularly associated with the structural markup of scientific articles. This approach addresses the need for an explicit formal description of the semantics of text-oriented XML-documents. The domain of our investigation is a corpus of scientific articles from psychology and linguistics from both English and German online available journals.

For our analyses, we provide XML-markup representing two kinds of semantic levels: the *thematic* level (i.e. topics in the text world that the article is about) and the *functional* or *rhetorical* level. Our hypothesis is that these semantic levels correlate with the articles' *document structure* also represented in XML. Articles have been annotated with the appropriate information. Each of the three informational levels is modelled in a separate XML document, since in our domain, the different description levels might conflict so that it is impossible to model them within a single XML document.

For comparing and mining the resulting multi-layered XML annotations of one article, a Prolog-based approach is used. It focusses on the comparison of XML markup that is distributed among different documents. Prolog predicates have been defined for inferring relations between levels of information that are modelled in separate XML documents. We demonstrate how the Prolog tool is applied in our corpus analyses.

Categories and Subject Descriptors

I.7.2 [**Document Preparation**]: Document and Text Processing — Markup Languages

General Terms

Theory, Experimentation

Keywords

XML, Semantic Analysis, Prolog, Information Extraction

1. INTRODUCTION

Conventionally, the semantics of a document grammar has been specified by non-formal means:

- naming markup elements and attributes appropriately
- inserting comments into the document grammar
- providing an external documentation such as the TEI Guidelines (cf. [25])

Recently, a number of semantic description languages have been developed in the context of semantic web activities, such as RDF/RDFS, OIL, SHOE, and the Topic Map standard, which so far have mainly been used as a supplement for XML schema grammars to ensure interoperability of *data-oriented* XML documents. A semantic approach to both *data-oriented* and *text-oriented* XML markup has been presented (cf. [19]) analysing the actual semantic relations among *text objects* (titles, sections, paragraphs, etc.) which are different and more specific than the hierarchical relationships expressible in a DTD. Instead of data-oriented documents our project focusses on the semantics of text-oriented document structures, such as the structure of scientific articles, in terms of text semantics, that is relations among the concepts that the text is about.[1]

In particular, we are interested in what kind of semantic information is regularly associated with structural markup. Such knowledge can help improve information retrieval applications for marked-up texts. To investigate the relationship between structural markup categories and semantic concepts on an empirical basis, we compiled a corpus of articles from scientific journals and annotated them according to a shortened variant of the DocBook DTD ([31], see section 3.1).

We assume that there are essentially two types of semantic information associated with the contents of a scientific article: firstly, the *thematic* structure of the article, describing

the *text world* or *problem space* (cf. [27]) that is referred to by the article; secondly, the article's *functional* or *rhetorical structure*, i.e. the rhetorical relations that hold between the discourse units of the article. To represent these, we resorted to *Rhetorical Structure Theory* (RST, cf. [16]). Unlike thematic structure, the rhetorical relations appearing in a text are largely independent of the text type.

The method of relating the thematic and rhetorical structure of an article to its structural markup is to provide XML markup also for the thematic and the rhetorical structure and to annotate a corpus of scientific articles on these three levels. For comparing and mining the resulting multi-layered XML documents we use an approach developed within the project *Sekimo* where universal linguistic functions (coreference in particular) and their linguistic realisations in typologically different languages are investigated.[2] An overview on this approach with relation to the Sekimo project is given in [22]. A detailed description of the approach and its implementation is given in section 4.

In the remainder of the article we describe the underlying text basis or corpus we use for our analyses as well as the methodological approach and its actual realisation. Some sample analyses will be presented to get a better insight of the potentials of our approach.

2. CORPUS

The primary aim, when compiling the corpus, was to create a collection of documents of one text type, in this case scientific articles, with a broad range of possible instances showing enough variety to ensure representative results.

2.1 Composition

Since we assumed that the scientific discipline chosen influences the degree of standardisation especially with regard to representational (structural) and rhetorical issues, we decided to include articles from a field with a traditionally highly standardised article structure (psychology) and from another field with a less standardised one (linguistics). Besides disciplinary background, we hypothesised that the language of an article may play a role in the representation of document structures. Hence, papers in English as well as German were included.

The actual corpus consists of 158 articles proportioned as follows:

- English, psychological: 60 experimental, 18 review
- English, linguistic: 30 experimental, 3 review
- German, linguistic: 47 without further classification

The annotation scheme described below is ongoing and presently completed for 15 articles.

2.2 Sampling strategies

The ranking of the Institute for Scientific Information (ISI), where journals are listed according to their actual importance or impact factor, was taken as a starting point for sampling English psychological and English linguistic documents. Lists for psychological and linguistic journals (as of September 2002) were obtained and checked for electronic availability. From every accessible journal, three articles out

of successive volumes (e.g. 2002, 2001, 2000) were chosen at random.

Due to the fact that German journals are not included in the ISI-rankings, alternative sampling methods had to be found for German articles. Only a very small amount of German psychological journals are available via internet, so these have been excluded from the corpus so far. An appropriate amount of German linguistic articles, however, could be obtained from the online-journal 'Linguistik Online' which hosts a large archive of publicly accessible electronic documents.

2.3 Pre-processing of documents

Originally, documents were obtained either as pdf or as html files. Both formats were converted to text format and automatically provided with skeletal XML markup, e.g. with segment annotations as described below. Annotations on the different levels are provided semi-automatically by human annotators using commercial XML processing tools like XMetal and XMLSpy, or O'Donnell's RST-Tool (cf. [18]). During the whole process the quality of the annotation is controlled by checking inter-rater reliability (cf. [5]) and intra-individual consistency (coder drift).

3. ANNOTATION LEVELS

In the following we discuss the rationale behind our three levels of text annotation (structural, thematic, rhetorical) and the XML markup for annotating our corpus on each of these levels.

The starting points for creating our annotation schemas were mainly those schemas developed by [12] and [28]. In our view, however, both schemas as well as others that focussed on scientific articles (e.g. [13, 29]) had certain drawbacks. Often these schemas were not informative enough for the annotations we had in mind. Moreover, most of them did not distinguish between the different types of information known from text-linguistic theories (e.g. along the lines of the criteria for textuality *coherence* and *intentionality* in [6]). In [9] and [11] a separation of levels is suggested, too, where besides layout and grammatical issues, content vs. functional structures are a major concern. Whereas content structure addresses the propositional content of texts (corresponding to our *thematic* structure), functional structure deals with their illocutionary aspects.

As we hypothesised that different types of information would result in different markup structures including overlaps, we decided to employ three separate schemas, each representing one type of information.

3.1 Structural level

As a representant of structural markup we chose the DocBook DTD, which is a standard originally developed for technical documentation (cf. [31]), e.g. manuals, but is recently also used in academic writing. We defined a proper subset of the DocBook DTD containing element types that are relevant for scientific articles such as <sect1>, <footnote>, <table>, and their respective subelements. This subset was extended by 14 additional logical elements, such as <toc> defined in a separate XML-schema[3], which can be included

---
[3]Usually a table of contents is generated from the structural markup pertaining to sections, but when annotating printed text, the table of contents has to be marked explicitly.

via namespace references. The annotators can thus choose from 61 structural markup elements.[4] A sample annotation can be seen in Figure 1.

3.2 Thematic level

To represent the thematic structure of an article, we developed a schema which includes concepts (called *topics*) such as *hypothesis, method*, or *dataCollection*, and semantic relations between them (e.g. *subtopic-of* or *has-property*). The annotation of documents with this schema results in the thematic structure of a text representing its thematic progression. The topics are hierarchically structured, which leads to a tree-shaped schema, assuming that each topic is either a node or a leaf (see Figure 2).

Higher levels of the schema are considered more global topics applicable to scientific articles in general (e.g. *problem, background*), whereas lower levels become more and more specific, finally denoting topics associated with specific disciplines (e.g. *quality_msr,sizeFinal,responseRate*). In this respect, the thematic schema is related to van Dijk's concept of *macrostructures* [8].

The inventory of the topics of a scientific article depends to a great degree on the discipline considered. In accordance with the complexity of scientific documents, our schema currently lists 120 topics applicable to articles from the fields of psychology or linguistics.

The thematic schema represents a canonical order of topics, i.e. in a particular article topic occurrences may deviate from it in several ways. For instance, it is to be expected that several topics will not be present in an instance. Moreover, the order of topic occurrences may differ from the one described in the underlying schema.

For the annotation these considerations led to the adoption of an XML-structure in which topics are represented as attribute values instead of nested elements (see Figure 1). Only the status of topics as *nodes* or *leaves* is represented by the elements `group` and `segment`, respectively. The resulting XML-structure is thus a shallow one, but the original hierarchical structure can be reconstructed via the ID/IDREF-mechanism of the attributes `parent` and `id`, similar to O'Donnels XML representation of rhetorical structure trees (see below).

Each text segment (basically corresponding to a sentence) is annotated with one topic. Any text segment may be thematically annotated, including those in abstracts, footnotes, or captions. Furthermore, text elements such as pointers to tables, figures, bibliography entries, etc. are annotated thematically.

3.3 Rhetorical level

The nature of rhetorical relations between discourse segments is argumentative, or pragmatic, i.e. involves the relationship between author and reader. To analyse such relations we employed the Rhetorical Structure Theory (RST) developed bei Mann and Thompson in the 1980s (e.g. [15, 16]).

RST assigns a text a hierarchical structure where the constituents (*text spans*) in turn consist of (usually two) smaller text spans, between which a rhetorical relation holds where one is the *nucleus* and the other is the *satellite* of that re-

---

```
<segment id="16" parent="71" relname="span"> The impending
  &quot;crisis&quot; and the debate over how to reform the civil
  litigation system have been prominent topics in the news
  media.</segment>
<segment id="17" parent="56" relname="span"> From the now
  infamous McDonald&apos;s coffee spill case to litigation
  against Ford and Firestone for injuries caused by tire tread
  separation to tobacco litigation, high stakes civil cases have
  become familiar staples of our media diet</segment>
<segment id="18" parent="17" relname="evidence"> (see e.g., Are
  lawyers burning America, 1995; Budiansky, 1995; Church, 1986;
  Langley, 1986; Stossel, 1996).5</segment>
...
<group id="56" type="span" parent="16" relname="elaboration" />
...
<group id="71" type="span" parent="84" relname="span" />
```

Figure 3: XML representation of rhetorical structure

lation. Examples of rhetorical relations are *concession*, *evidence*, and *elaboration*. We employ 31 of the 33 relations in the `ExtMT.rel` relation set provided with O'Donnell's RST annotation tool (cf. [18]), which is based on the set of 23 relations in [16]. We further defined 10 relations on the basis of those employed by [4] and what we considered necessary for the analysis of scientific articles.

For the annotation of rhetorical structures to (parts of) scientific articles we employ the RST-Tool by Mick O'Donnell (cf. [18]). This tool provides a graphical interface where links between text spans can be drawn, i.e. the RST representation tree can be built and relations can be annotated by drag and drop mouse operations. The tool conveniently stores such a structure in well-formed XML, which we then convert into valid XML according to O'Donnells RST.dtd that comes with the tool. The XML structure is shallow, basically consisting of instances of the elements `group` and `segment` which are linked via the ID/IDREF attributes `parent` and `id` that encode the hierarchical structure (see Figure 3).

4. METHOD

In this section we present an approach for querying our XML corpus, focussing on comparing the structural and the thematic annotation levels (i.e. in this paper, queries involving the rhetorical structure will not be discussed for reasons of space). Since we use the term annotation *level* to refer to an abstract level of analysis (such as the level of morphology in a linguistic grammar), we introduce the term annotation *layer* to refer to the actual realisation of the annotation in e.g. XML. In our case, each of the three levels is realised as a single XML annotation layer. For other domains, however, it might be suitable to model one level in more than one annotation layer, or one annotation layer may serve several levels.

Thus, the focus of our method is on the comparison of different annotation layers of the same document[5], whereas most existing query languages place their emphasis on the analysis of single layer annotations (cf. [2], ch. 2). The XML data model can be seen as a tree, so that hierarchical relations like inclusions between elements can easily be queried. Existing query languages normally do not provide a con-

---

[4]The structural markup schema was designed in collaboration with the project A2/*HyTex*, DFG-Forschergruppe 437/*Texttechnologische Informationsmodellierung*

[5]A related approach which is specialised on linguistic data is proposed in the Nite project (cf. [3]).

```
<sect1>
 <title>INTRODUCTION</title>
 ...
 <para>The impending "crisis" and the debate over how to reform the civil litigation system have been prominent topics in the news
  media. From the now infamous McDonald's coffee spill case to litigation against Ford and Firestone for injuries caused by tire
  tread separation to tobacco litigation, high stakes civil cases have become familiar staples of our media diet (see e.g., Are lawyers
  burning America, 1995; Budiansky, 1995; Church, 1986; Langley, 1986; Stossel, 1996).
  <footnoteref linkend="i5">5</footnoteref>
 </para>
```

```
<group id="g1" parent="r01" topic="content"/>
<group id="g2" parent="g1" topic="problem"/>
<group id="g3" parent="g1" topic="evidence"/>
<group id="g6" parent="g2" topic="background"/>
...
<segment id="s23" parent="g6" topic="history_bck">The impending "crisis" and the debate over how to reform the civil litigation system
 have been prominent topics in the news media. </segment>
<segment id="s24" parent="g6" newtopic="illustration_bck" litref="s341 s349 s351a s389 s423" footnoteref="s33a">From the now infamous
McDonald's coffee spill case to litigation against Ford and Firestone for injuries caused by tire tread separation to tobacco
litigation, high stakes civil cases have become familiar staples of our media diet (see e.g., Are lawyers burning America, 1995;
Budiansky, 1995; Church, 1986; Langley, 1986; Stossel, 1996).5 </segment>
```

Figure 1: Structural (above) and thematic (below) annotation of the same text segment
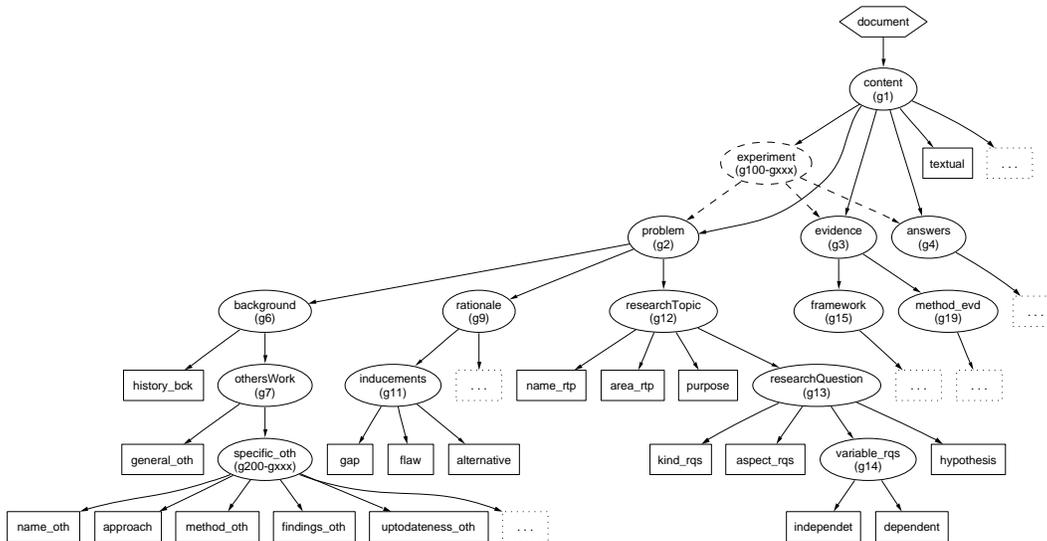

Figure 2: A fraction of the thematic schema

nection of distributed annotations. Moreover, overlapping elements cannot be modelled directly, as a parallel view on the primary data is needed to model the overlaps. Such a parallel view is realised in our approach by extending the XML data model with information about the absolute position of the PCDATA contained by a node. This approach is known as position-based indexing, a technique used for indexing large text data-bases (cf. [21]). In our project the positional information allows for a connection of the different annotation layers.

To represent our data model for a parallel view on multiple annotations, we use the programming language Prolog. Prolog is often used in computational linguistic applications and is compatible with SGML and XML as well. The transformation of XML annotations into a Prolog fact base allows for all kinds of queries that can be formulated in Horn clause logic, a subset of first order predicate logic. A Prolog-based system for text retrieval of SGML data, for instance, has been developed by [23]. A more recent approach using Prolog for modelling XML annotations was put forward by [26].

4.1 Modelling multi-level annotations

In general it is assumed that markup added to a document structures the text hierarchically and that markup elements are compatible with each other. This assumption is often referred to as the OHCO-thesis (ordered hierarchy of content objects, cf. [7]). This point of view, however, does not hold for text data, especially not for linguistic analyses of language data (cf. [24]).[6] Paragraph and page structure, or morpheme and syllable structure are examples of description levels for which a non-hierarchical ordering often occurs. The thematic structure of a document and its logical structure in terms of text objects such as sections, paragraphs,

---

[6]The OHCO-thesis also is critically discussed in [20].

etc. cannot necessarily be made compatible in a hierarchical ordering, either.

In the same way elements of different levels may overlap or mutually include each other. Even when it is possible to find a common hierarchy that models the separate levels, the distribution of these levels on different annotation layers allows for different treatments of the diverse types of information which is sometimes desired. A separation of annotation levels into several layers is also reasonable if there is no or only little knowledge about how to define a common annotation layer, e.g. by a document grammar. Such knowledge can instead be extracted from the data by correlating the elements of the different layers.

Since a parallel view on the data is necessary to model overlapping units, the XML data model is augmented with the start and end positions of annotated sequences of primary data. Thus, the primary data serve as an absolute reference which is independent of the actual annotation layer. By *primary data* we refer to the raw PCDATA without any markup, i.e. the characters without annotation tags. The markup information encoded on the different annotation layers is interpreted as meta data with respect to the primary data (cf. [32]). It is now possible to compare different annotation layers in order to correlate text sequences explicitly.

A text sequence is uniquely defined by its start and end position within the primary data. Given identical primary data, it is easy to extract and compare the annotations that include a text sequence that is defined by its start and end position. Characters instead of words have benn chosen as smallest units, thus the start and end positions are character positions. The use of characters as smallest units allows e.g. linguistic analysis such as a morphological analysis which refers to subword units, i.e. parts of words. Therefore, single instances of elements defined by the document grammar can be compared, or conclusions about element classes, i.e. propositions that are true for all instances of an element, can be drawn.

4.2 Implementation of the translation from XML to Prolog

To analyse text data annotated in the way desribed above we have implemented queries in Prolog. In order to apply these queries to XML data, the XML data has to be translated into Prolog facts. These facts serve as input for the Prolog program that is used for the analysis. The translation is described in this section, the Prolog program is described in section 5.

The Prolog-to-XML translation is done by a Python script called *xml2prolog.py*. The script recursively traverses the DOM tree of the XML files. As input either a single XML file or several XML files can be passed. Python was chosen as programming language to convert XML to Prolog because of its reasonable XML and Unicode support. Due to its simple syntax and extensive standard library, fast software development is possible.[7]

The output of the script is a Prolog database. The resulting Prolog facts encode both the XML tree structure and the sequential ordering of elements with respect to the primary data. In the second step, they serve as input data for

the Prolog program `SeIT.pl`.[8]

The translation of an XML element results in a Prolog fact with 5 arguments (node/5), the translation of an attribute in a Prolog fact with 6 arguments (attr/6).

Result of an element translation:
node(AnnotationLayer, Start, End, Node, element(ElementName)).

- **AnnotationLayer** is the name of the annotation layer (a unique identifier, e.g. the file name).
- **Start** is the starting point in the primary data.
- **End** is the last position in the primary data.
- **Node** is the node in the DOM tree of the respective layer.
- **element(ElementName)** is the element name.

Result of an attribute translation:
attr(AnnotationLayer, Start, End, Node, AttributeName, Value).

- **AnnotationLayer**, **Start**, **End**, and **Node** are defined as for node/5.
- **AttributeName** is the attribute name.
- **Value** is the value of the attribute.

The output can be varied by specifying additional parameters in order to:

1. test if the XML files share the same primary data, i.e. if the primary data are identical.[9] Given a DTD for the XML files, the XML annotations are validated and those whitespace sequences are ignored that are not specified as relevant in the DTD. In case the primary data are not identical, the position of the first deviating character and its context are written to the output device.

2. generate Prolog facts for each character of the primary data for each annotation layer:
node('thm',0,1,[1,1,1],pcdata('A')). The first argument is the layer name, the second and third argument indicate the start and end position of the character. The fourth position is the node in the DOM tree, and the last argument contains the character at this position.

3. generate a Prolog fact for each character of the primary data only once, i.e. independent of the number of annotation layers: pcdata_node(0,1,'A'). The first two arguments indicate the start and the end position. The third argument contains the character at this position.

Figure 4 shows a fraction of the Prolog database for the example given in Figure 1.

5. SAMPLE ANALYSES

The functionality of the Prolog predicates of `SeIT.pl` serves three aims:

---

[7]The *xml2prolog*-script and a documentation can be downloaded at http://www.text-technology.de/sekimo/internet-praesentation/prolog.html

[8]*SeIT* is an abbreviation for *Semantic Inference Tool.*

[9]In case of white space differences between a closing tag and an opening tag the script tries to expand or shorten this whitespace and assumes the files' primary data to be identical.

- querying statistics about the inventory of annotation elements
- querying information about element instances
- querying information about element classes

In the current project, queries are formulated on the Prolog command line, output is written either to the standard output device or into an output file. Alternatively, interfaces to the programming languages C, C++ or Java exist in SWI Prolog.

In the following subsections, we present sample queries to the Prolog database of Figure 4. This database represents the scientific article[10] annotated on the structural (layer `doc`) and thematic level (layer `thm`) as shown in Figure 1.

```
node('doc', 0, 78286, [1], element('article')).
node('doc', 0, 92, [1, 1], element('title')).
node('doc', 92, 149, [1, 2], element('section')).
node('doc', 92, 149, [1, 2, 1], element('para')).
node('doc', 115, 116, [1, 2, 1, 1], element('footnoteref')).
node('doc', 44787, 46152, [1, 9, 3, 5], element('sect3')).
node('doc', 44787, 44812, [1, 9, 3, 5, 1], element('title')).
node('doc', 44812, 46152, [1, 9, 3, 5, 2], element('para')).
...
attr('thm', 7379, 7415, [1, 113], 'newtopic', 'aspect_rtp').
attr('thm', 7415, 7562, [1, 114], 'topic', 'findings_oth').
attr('thm', 7415, 7562, [1, 114], 'parent', 'g7').
attr('thm', 7562, 7737, [1, 115], 'litref', 's433').
attr('thm', 7737, 7967, [1, 116], 'topic', 'findings_oth').
attr('thm', 7967, 8122, [1, 117], 'topic', 'findings_oth').
attr('thm', 8122, 8265, [1, 118], 'topic', 'method_oth').
...
```

Figure 4: Listing of a fraction of the Prolog database for the structural (above) and thematic (below) level.

5.1 Statistics

The statistics provide a first survey over the annotated data. They permit an interpretation of the data even if no information on the structure of the data is known or if one has not worked with the annotation inventory yet. Statistics for the `doc` sample instance are given in Figure 5. They show for instance that the lowest section structuring element used on the `doc` layer is `sect3`, occurring 3 times.

5.2 Comparison of annotation layers

Further predicates are implemented for the comparison of annotation layers. Relations between sequences that arise from the different start and end positions are widely used in the field of Artificial Intelligence for relating temporal units to each other (cf. [1]). Recently, these relations have been applied for a structural interpretation of XML documents annotated on different layers (cf. [10], [30]). Robert C. Miller ([17]) describes in detail all possible relations between contiguous segments of text.

Regarding the relations that exist between XML elements of two annotation layers, either the element instances can be analysed or the element classes as a whole can be considered. In the first case single text sequences (defined by their start and end positions) are compared with each other. In

```
get_statistics(doc).

Number of Nodes : 232
Number of different Elements : 12
Number of Attributs : 9
Number of different A/V-pairs : 21
-------------------------------------------
Different elements and their occurrences :
abstract      1
article 1
bibliography  1
bibliomixed   103
footnote      7
footnoteref   7
para    80
sect1   7
sect2   4
sect3   3
section 2
title   16
-------------------------------------------
Attribute       # occurrences   # different values
ID              7               7
Lang            1               1
linkend         7               7
-------------------------------------------
```

Figure 5: Part of the statistics for the structural annotation (layer `doc`)

the latter case the observations about all instances of the elements under consideration are summarised.

5.2.1 Comparison of element instances

For every element that is defined in the document grammar, several instances in the XML document may exist, i.e. several sequences of the primary data are marked with this element.

Given two text sequences i of annotation layer $L1$ and j of annotation layer $L2$, i is labelled with element I and j is labelled with element J. $S(i)$ and $E(i)$ indicate the start and end position of sequence i, $S(j)$ and $E(j)$ indicate the start and end position of sequence j respectively.

When comparing the start and end position of two sequences, the following relations may be observed:

- identity of start and end position: i and j share the same start and end points
- independence: the ranges of i and j are independent of each other
- inclusion: j contains i completely (or vice versa)
- start point identity: special case of inclusion, i and j share starting point, j includes i
- end point indentity: special case of inclusion, i and j share end point, j includes i
- end point is starting point: j begins when i ends (or vice versa)
- overlap: ranges of i and j overlap

Thus, for two instances i and j, when i is annotated with element I and j is annotated with element J, there are at least fifteen possible relations, depending on whether the sequence i precedes j or i follows j, e.g. the inclusion relations holds either if $S(i) < S(j) \wedge E(j) < E(i)$ or if $S(j) < S(i) \wedge E(i) < E(j)$. Figure 6 illustrates the possible relations.[11]

---

[10]J.K. Robbenolt and Christina A. Studebaker. News media reporting on civil litigation and its influence on civil justice decision making. In *Law and Human Behaviour*, 27(1):5-27, 2003.

[11]The relation *identity* can be seen as a special case of the inclusion relation, too, i.e. mutual inclusion. However, no hierarchical structure between the two elements exists as it is the case for the other variants of the inclusion relation.
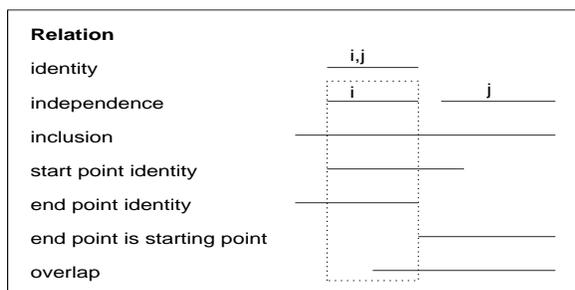
Figure 6: Possible Relations for sequences i and j

5.2.2 Predicates

The predicate **chk_relation/6** provides an analysis of relations between an element in layer *L1* and an element in layer *L2*. With the help of two variant predicates, the elements may be combined with an attribute-value specification in layer *L1* (**chk_relation/7**) or in both layer *L1* and layer *L2* (**chk_relation/8**). **chk_relation/7** is used for the following example:

chk_relation(Rel,Element1,Layer1,
Element2,[Attrib2, Value2],Layer2,L).

- **Rel** is one of the relations described above.
- **Element1** is the element name of annotation layer **Layer1**.
- **Element2** is the element name of annotation layer **Layer2**.
- **Attrib2** is the attribute name of Layer2, **Value2** is the attribute's value.

Output is a list **L** that contains the number of occurrences where the relation holds for the given elements and attribute-value specification. In addition, the total number of occurrences of the element or element plus attribute-value specification themselves are computed. This enables the developer to decide if the relation holds for all occurrences of the elements or if additional relations have to be investigated.

Figure 7 shows the results for querying which topic specification on elements of type `segment` on the layer `thm` occurred included in `sect3`-elements on the layer `doc`. Since the value of the attribute `topic` on layer `thm` is given as the variable `T`, the result is a listing of all topics that occurred within a section-structuring element `sect3` on the layer `doc`. Judged by the number of segments with identical topic assigments, the topics `assumption` (i.e. a general assumption within the framework described), `findings_oth`, and `method_oth` (i.e findings from and methods used in previous studies) were extensively discussed in one or more `sect3`-sections.

In line 11 in Figure 7, for example, it is stated that there are 9 instances of `segment` elements with the attribute specification `topic="method_oth"` on the layer `thm` that are included within instances of `sect3` elements on layer `doc`. Since there are only three instances of `sect3` altogether, there may (but need not) be counter-examples i.e. instances of `sect3` on the layer `doc` that do not include any `segment` with the specification `topic="method_oth"`. And since there are 14 instances of `segment` with the attribute-value specification `method_oth` altogether, there definitely must be some

counter-examples on the layer `thm`, i.e. instances of these that do not occur within `sect3`. Thus, for some relations, it might be desirable to list separately those element instances for which a relation holds and those elements instances for which a relation does not hold (from both layers). The predicate **chk_occurrence** permits this kind of query.[12]

chk_occurrence(Rel,E1,Layer1,E2,[Attrib2,Value2],
Layer2,L,CounterEx1, CounterEx2).

- **L** is a list containing all element instances for which the relation **Rel** holds.
- **CounterEx1** is the list of counter-examples in **Layer1**, i.e. the list of all element instances for which **Rel** does not hold.
- **CounterEx2** is the list of counter-examples in **Layer2**, i.e. the list of all element instances for which **Rel** does not hold.

All other variables are defined as for **chk_relation/7**.

An example of a **chk_occurrence/9**-query to our `doc-thm`-database, asking for occurrences of the element `segment` plus the attribute-value specification `topic="method_oth"` on the layer `thm` within elements of type `<sect3>` on the layer `doc`, is shown in Figure 8.

```
?- chk_occurrence(included_B_in_A,sect3,doc,segment,
                  [topic,method_oth],thm,L,C1,C2).
L  = [(31314, 36280), [1, 9, 3, 3], (34093, 34188), [1, 255]
      (36280, 44787), [1, 9, 3, 4], (38841, 38913), [1, 280]
      (36280, 44787), [1, 9, 3, 4], (38913, 39175), [1, 281]
      (36280, 44787), [1, 9, 3, 4], (39758, 40015), [1, 284]
      (36280, 44787), [1, 9, 3, 4], (40738, 40862), [1, 288]
      (36280, 44787), [1, 9, 3, 4], (40862, 41030), [1, 289]
      (36280, 44787), [1, 9, 3, 4], (42277, 42412), [1, 298]
      (36280, 44787), [1, 9, 3, 4], (42412, 42511), [1, 299]
      (36280, 44787), [1, 9, 3, 4], (42511, 42610), [1, 300]]

C1 = [[sect3, doc, [1, 9, 3, 5], (44787, 46152)]]

C2 = [[segment, thm, [1, 118], (8122, 8265)]
      [segment, thm, [1, 119], (8265, 8395)]
      [segment, thm, [1, 201], (24346, 24536)]
      [segment, thm, [1, 223], (28319, 28437)]
      [segment, thm, [1, 332], (49106, 49356)]]
```

Figure 8: Listing of instances of a relation and counter-examples: chk_occurrence/9

Interestingly, 9 of the 14 sequences annotated with the topic `method_oth` occurred within the first two nodes with element type `sect3`, which also happen to be adjacent sibling nodes, namely `[1, 9, 3, 3]` and `[1, 9, 3, 4]`. The variable `C1` is instantiated with a list containing one counter-example on layer `doc`, i.e. one `sect3` node that does not contain a `segment` specified for `topic="method_oth"` (counter-examples are given by their node IDs and start and end positions in the primary data). The list `C2` contains the counter-examples on the layer `thm`, i.e. those instances of `segment` elements specified for `topic="method_oth"` that are not included within `sect3` elements on layer `doc`.

---

[12]chk_occurrence/8 for comparing an element on layer *L1* with an element on layer *L2*, chk_occurrence/9 for comparing an element plus one attribute value specification on layer *L1* with an element on layer *L2*, and chk_occurrence/10 for comparing an element plus one attribute value specification on layer *L1* with an element plus one attribute-value specification on layer *L2*.

```
?- chk_relation(included_B_in_A,sect3,doc,segment,[topic,T],thm,L).
L = [[[sect3, segment, [topic, approach_oth], 1], [sect3, 3], [segment, [topic, approach_oth], 7]]
     [[sect3, segment, [topic, assumption], 15], [sect3, 3], [segment, [topic, assumption], 47]]
     [[sect3, segment, [topic, dataAnalysis_oth], 1], [sect3, 3], [segment, [topic, dataAnalysis_oth], 1]]
     [[sect3, segment, [topic, evaluation_oth], 1], [sect3, 3], [segment, [topic, evaluation_oth], 2]]
     [[sect3, segment, [topic, findings_oth], 32], [sect3, 3], [segment, [topic, findings_oth], 112]]
     [[sect3, segment, [topic, futureResearch], 2], [sect3, 3], [segment, [topic, futureResearch], 24]]
     [[sect3, segment, [topic, gap], 2], [sect3, 3], [segment, [topic, gap], 10]]
     [[sect3, segment, [topic, kind_rtp], 1], [sect3, 3], [segment, [topic, kind_rtp], 2]]
     [[sect3, segment, [topic, kind_rtp_oth], 3], [sect3, 3], [segment, [topic, kind_rtp_oth], 5]]
     [[sect3, segment, [topic, method_oth], 9], [sect3, 3], [segment, [topic, method_oth], 14]]
     [[sect3, segment, [topic, procedure_oth], 2], [sect3, 3], [segment, [topic, procedure_oth], 2]]
     [[sect3, segment, [topic, relation_oth], 1], [sect3, 3], [segment, [topic, relation_oth], 3]]
     [[sect3, segment, [topic, theoreticalBasis], 3], [sect3, 3], [segment, [topic, theoreticalBasis], 12]]]
```

Figure 7: Listing of results for chk_relation/7 for cases where an element `sect3` on the layer `doc` includes any element `segment` on the layer `thm` where the attribute `topic` is specified

However, querying information of the kind *What structural element includes which topics in article X* is not really sufficient to yield statistically interesting results. First, we would like to know not only that e.g. the topic `method_oth` occurred within a `para`, but also at which structural position such a `para` is situated. The importance of sentence and paragraph position information for identifying topics (in this case article topics) was pointed out e.g. in [14], and using the structural markup in our corpus we are able to provide structural position information for *all* elements of the markup. Thus, by means of an XSLT counting script we have enriched the structural markup with information such as (relative) element position among all siblings of the same type, stored in additional attributes such as `POSINFO1`. This enables us to formulate `SeIT.pl` predicates querying for occurrences of topics in relation to elements with their structural positions like `/article[1]/sect1[2]/para[1]` (which reads: first paragraph under the second sect1 in the whole article), as in

chk_relation(included_B_in_A,sect2,['POSINFO1',_], doc,segment,[topic,T],thm,L).

That way, the semantics of an XML element such as `sect3` may be differentiated according to context and reference in the sense of [19]. Some results in terms of frequency lists for a subcorpus are presented in Tables 1 and 2).

Occurrences of	
method_oth	at **Structural position**
17	/article[1]/sect1[1]
14	/article[1]/sect1[5]
7	/article[1]/sect1[4]
5	/article[1]/sect1[2]
2	/article[1]/sect1[3]
0	/article[1]/sect1[6]
0	/article[1]/sect1[7]
0	/article[1]/sect1[8]
0	/article[1]/sect1[9]
45	/article[1]

Table 1: Frequency list for *method_oth* at different structural positions in a subcorpus with 15 scientific articles

5.2.3 Comparison of element classes

The relations and predicates described in the previous sec-

	Occurrences at
Topic	/article[1]/sect1[2]/para[1]
educational	4
assumption	3
procedure	3
gap	2
illustration_frm	2
sample_oth	2
...	...
total:	33

Table 2: Frequency list for topics occurring at the structural position `/article[1]/sect1[2]/para[1]` in a subcorpus with 15 scientific articles

tions focus on the analysis of text sequences, i.e. single element instances. For a more comprehensive comparison of two annotation layers, further information about whole element classes is necessary, i.e. on relations between all instances of two element classes. This information might e.g. be needed in order to decide whether two separate annotation layers can be combined into a single XML document and in order to define the hierarchical structure between elements of different layers.

When examining all instances of two element classes it is – in the majority of cases – not possible to state one single relation that holds for all instances but only a list of relations that hold between subsets of the two element classes. To increase the probability to find unique relations between element classes, we have defined four more general *meta relations* each of which subsumes several of the relations described earlier:

- **Independence** For all instances i and j of the classes I and J, only the relations of independence and end point = starting point hold.

- **Identity** For all non-independent instances of i and j, only the identity relation holds.

- **Inclusion** For all non-independent instances of i and j only the relations identity, inclusion, start point identity, or end point identity hold.

- **Overlap** For all non-independent instances of i and j, only the overlap relation holds.

Two predicates have been defined to analyse these meta relations.

- **chk_metarelation_el/6** outputs the corresponding meta relation for an element class A and a class of element plus attribute-value specification B.
- **chk_metarelation_layer/3** outputs the corresponding meta relations for all element classes of two annotation layers.

Figure 9 illustrates a query after a meta relation for our example, i.e. for the element `sect3` on the layer `doc` compared with the element `segment` specified for `topic="method_oth"`. The result is [`inclusion_B`], meaning that such segments on the layer `thm` never extend across `sect3` elements on the layer `doc`.

```
?- chk_metarelation_el(sect3,doc,segment,
                       [topic,method_oth],thm,Metarelation).

  Metarelation = [inclusion_B]
```

Figure 9: Listing of meta relations: chk_metarelation_el/6

6. CONCLUSION AND PROSPECTS

We have presented a corpus-based approach to analysing the semantics of structural markup categories for text-oriented XML-documents, in particular, scientific articles annotated according to the DocBook DTD. Apart from the structural markup, the articles in our corpus are provided with XML annotations on two further, semantic levels, namely the thematic level and the rhetorical level. Each level corresponds to one annotation layer stored in a separate XML document instance containing the same primary data, since it cannot be guaranteed that our description levels are compatible with each other (i.e. that element instances on different layers do not overlap). For comparing the different layers, i.e. for being able to state relations between element instances on the different layers, we employ the Prolog-based query system `SeIT.pl` for multi-level annotations, which provides a single view on distributed annotations and is able to model overlaps. Possible relations between text sequences in element instances on different annotation layers are *identity*, *inclusion*, *overlap*, *independence*, and special cases of these. We presented Prolog predicates for querying relations between elements and for checking occurrences (in terms of byte offsets of the identical PCDATA) of instances of such relations, and we demonstrated how these are applied to our corpus. We showed how structural position information inserted in the structural markup can be correlated with thematic structure and reveal dependencies between certain topics and certain structural positions. Moreover, we presented predicates for the analysis of meta relations i.e. relations between classes of elements. Presently, the meta relations are defined very strictly in `SeIT.pl`, e.g. the meta relation *identity* only holds if for all non-dependent instances of element i on layer $L1$ and of element j on layer $L2$, the original identity relation holds. But we want to give more detailed statements about element classes (on each level) whose elements stand in different relations. Thus we intend to cluster such element instances into element class subsets for which meta relations still hold.

In the project *Semantic of generic document structures,* we also intend to establish further dependencies such as frequencies of typical DocBook subtree configuration occurrences within certain topics and also between grammatical (i.e. non-semantic) features (such as lemmata n-gram frequencies or occurrence of past vs. present tense) and thematic and structural markup. The result will be a collection of *text-type parameters* modelled in a text-type ontology which contains links to structural markup categories. We believe that such a knowledge base can be used for (semi-)automatic semantic web annotations. We also intend to evaluate whether the approach can improve the performance of applications that require a knowledge of the textual semantics, such as document indexing, discourse parsing, or information retrieval, where only structural markup is available in the input.

The tools described in this paper are available for public download: http://www.text-technology.de/sekimo/internet-praesentation/prolog.html.

7. ADDITIONAL AUTHORS

Additional author: Daniel Naber (Universität Bielefeld, Germany, email: `daniel.naber@t-online.de`).

8. REFERENCES

[1] J. Allen and G. Ferguson. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4(5):531–579, 1994.

[2] A. Bonifati. *Reactive Services for XML Repositories.* PhD thesis, Politecnico di Milano, 2002.

[3] J. Carletta, J. Kilgour, T. O'Donnell, S. Evert, and H. Voormann. The NITE object model library for handling structured linguistic annotation on multimodal data sets. In *Proceedings of the EACL Workshop on Language Technology and the Semantic Web (3rd Workshop on NLP and XML, NLPXML-2003)*, Budapest, Hungary, 2003.

[4] L. Carlson, D. Marcu, and M. E. Okurowski. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the 2nd SIGDIAL Workshop on Discourse and Dialogue*, Eurospeech 2001, Denmark, 2001.

[5] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurements*, 20:37–46, 1960.

[6] R.-A. de Beaugrande and W. U. Dresseler. *Einführung in die Textlinguistik.* Niemeyer, 1981.

[7] S. DeRose, D. Durand, and E. Mylonas. What is text, really? *Journal of Computing in Higher Education*, 1(2):3–26, 1990.

[8] T. Dijk, editor. *Macrostructures: An interdisciplinary study of global structures in discourse, interaction, and cognition.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1980.

[9] M. Dimter, editor. *Textklassenkonzepte heutiger Alltagssprache. Kommunikationssituation, Textfunktion und Textinhalt als Kategorie alltagssprachlicher Textklassifikation.* Niemeyer, Tbingen, 1981.

[10] P. Duruseau and M. Brook O'Donnell. Concurrent markup for XML documents. In *XML Europe 2002 - Conference Proceedings*, 2002.

[11] E. Gülich. Textsorten in der Kommunikationspraxis. In W. Kallmeyer, editor, *Kommunikationstypologie.*

Handlungsmuster - Textsorten - Situationstypen, pages 15–46. IDS, Düsseldorf, 1996.

[12] N. Kando. Text-level structure of research papers: Implications for text-based information processing systems. In *Proceedings of the British Computer Society Annual Colloquium of Information Retrieval Research*, pages 68–81, 1997.

[13] E. D. Liddy. The discourse-level structure of empirical abstracts: An exploratory study. *Information Processing and Management*, 27(1):55–81, 1991.

[14] C.-Y. Lin and E. Hovy. Indentifying topics by position. In *Proceeedings of the 5th Conference on Applied Natural Language Processing (ANLP)*, pages 283–290, Washington, D.C., 1997.

[15] W. C. Mann, C. M. Matthiessen, and S. A. Thompson. Rhetorical Structure Theory and text analysis. Research Report ISI/RR-89-242, 1989. Information Science Institute, Marina del Rey, CA.

[16] W. C. Mann and S. A. Thompson. Rhetorical Structure Theory: Toward a functional theory of text organisation. *Text*, 8(3):243–281, 1988.

[17] R. C. Miller. *Lightweight Structure in Text*. PhD thesis, Computer Science Department, School of Computer Science, Carnegie Mellon University, May 2002.

[18] M. O'Donnell. RSTTool 2.4 – A markup tool for Rhetorical Structure Theory. In *Proceedings of the International Natural Language Generation Conference (INLG'2000)*, pages 253 – 256, Mitzpe Ramon, Israel, 2000.

[19] A. Renear, D. Dubin, C. M. Sperberg-McQueen, and C. Huitfeldt. Towards a semantics for xml markup. In R. Furuta, J. I. Maletic, and E. Munson, editors, *Proceedings of the 2002 ACM Symposium on Document Engineering*, pages 119–126. Association for Computing Machinery, 2002.

[20] A. Renear, E. Mylonas, and D. Durand. Refining our notion of what text really is: The problem of overlapping hierarchies. In *Research in Humanities Computing*. Oxford University Press, 1996.

[21] R. Sacks-Davis, T. Dao, J. A. Thom, and J. Zobel. Indexing documents for queries on structure, content, and attributes. In *Proceedings of the International Conference on Digital Media Information Bases*, pages 236–245, Nara, Japan, Nov. 1997.

[22] F. Sasaki, C. Wegener, A. Witt, D. Metzing, and J. Pönninghaus. Co-reference annotation and resources: A multilingual corpus of typologically diverse languages. In *Proceedings of the 3nd International Conference on Language Resources and Evaluation (LREC-2002)*, pages 1225–1230, Las Palmas, Spain, 2002.

[23] B. Schröder. Pro-SGML: Ein Prolog-basiertes System zum Textretrieval. In G. Heyer and C. Wolff, editors, *Linguistik und neue Medien*, pages 205–216. DUV, 1998.

[24] G. F. Simons. The nature of linguistic data and the requirements of a computing environment for linguistic research. In J. Lawler and H. Aristar-Dry, editors, *Using Computers in Linguistics: A Practical Guide*, pages 10–25. Routledge, London, 1998.

[25] C. M. Sperberg-McQueen and L. Burnard, editors. *Guidelines for Electronic Text Encoding and Interchange (TEI P3)*. Text Encoding Initiative, Chicago and Oxford, 1994.

[26] C. M. Sperberg-McQueen, D. Dubin, C. Huitfeldt, and A. Renear. Drawing inferences on the basis of markup. In *Proceedings of Extreme Markup Languages 2002*, Montreal, Canada, 2002.

[27] S. Teufel. *Argumentative Zoning: Information Extraction from Scientific Text*. PhD thesis, University of Edinburgh, 1999.

[28] S. Teufel, J. Carletta, and M. Moens. An annotation scheme for discourse-level argumentation in research articles. In *Proceedings of EACL*, 1999.

[29] R. Trigg and M. Weiser. TEXTNET: A network-based approach to text handling. *ACM Transactions on Office Information Systems*, 4(1):1–23, 1986.

[30] T. Trippel, F. Sasaki, B. Hell, and D. Gibbon. Acquiring lexical information from multilevel temporal annotations. In *8th European Conference on Speech Communication and Technology*, Geneva, Switzerland, to appear, September 2003.

[31] N. Walsh and L. Muellner. *DocBook: The Definitive Guide*. O'Reilly, 1999.

[32] A. Witt. Meaning and interpretation of concurrent markup. In *Joint Conference of the ALLC and ACH (ALLCACH2002)*, Tübingen, Germany, 2002.