The Workshop Programme

Section I: Theory

Chair: Andreas Witt

09.00-09.40 **Dan Cristea and Cristina Butnariu: "Hierarchical XML Layers Representation for Heavily Annotated Corpora"**

09.40-10.20 **Hagen Langer, Harald Lüngen and Petra Saskia Bayerl: "Towards Automatic Annotation of Text Type Structure: Experiments using an XML Annotated Corpus and Automatic Text Classification Methods"**

10.20-11.00 **Peter Fankhauser and Elke Teich: "Multiple perspectives on text using multiple resources: experiences with XML processing"**

11.00-11.30 **coffee break**

Section II: Applications I

Chair: Ulrich Heid

11.30-12.10 **Stefanie Dipper, Lukas Faulstich, Ulf Leser and Anke Lüdeling: "Challenges in Modelling a Richly Annotated Diachronic Corpus of German"**

12.10.-12.50 **Emanuele Pianta and Luisa Bentivogli: "Annotating Discontinuous Structures in XML: the Multiword Case"**

12.50-14.00 **lunch**

Section III: Applications II

Chair: Peter Wittenburg

14.00-14.40 **Marion Freese: "Enabling xComForTable Mapping to the Linguistic Annotation Framework"**

14.40-15.20 **X. Artola, A. Díaz de Ilarraza, N. Ezeiza, K. Gojenola, A. Sologaistoa and A. Soroa: "EULIA: a graphical web interface for creating, browsing and editing linguistically annotated corpora"**

15.20-16.00 **coffee break**

Section IV: Software

Chair: Amy Isard

16.00-16.40 **Stefanie Dipper, Michael Götze and Manfred Stede: "Simple Annotation Tools for Complex Annotation Tasks: an Evaluation"**

16.40-17.20 **Peter Wittenburg, Hennie Brugman, Daan Broeder and Albert Russel: "XML-Based Language Archiving"**

17.20-18.00 **Thomas Schmidt: "Transcribing and annotating spoken language with EXMARaLDA"**

Workshop Organisers

Andreas Witt
andreas.witt@uni-bielefeld.de
Ulrich Heid
uli@ims.uni-stuttgart.de
Henry S.Thompson
ht@cogsci.ed.ac.uk
Jean Carletta
J.Carletta@edinburgh.ac.uk
Peter Wittenburg
peter.wittenburg@mpi.nl

Workshop Programme Committee

Jean Carletta, University of Edinburgh, UK
Ulrich Heid, University of Stuttgart, Germany
Nancy Ide, Vassar College & Loria, USA & France
Henning Lobin, Justus-Liebig-Universität Gießen, Germany
Dieter Metzing, Bielefeld University, Germany
Joakim Nivre, Växjö University, Sweden
Vito Pirrelli, Istituto di Linguistica Computazionale del CNR, Pisa, Italy
Gary Simons, SIL International, Taxas, USA
Laurent Romary, Loria, France
C. M. Sperberg-McQueen, W3C (MIT), USA
Henry S. Thompson, University of Edinburgh, UK
Jun'ichi Tsujii, University of Tokyo, Japan
Andreas Witt, Bielefeld University
Peter Wittenburg, MPI for Psycholinguistics Nijmegen, Netherlands

Table of Contents

Author Index

Hierarchical XML Layers Representation for Heavily Annotated Corpora

Dan Cristea and **Cristina Butnariu**
University "Al. I. Cuza" of Iaşi
Faculty of Computer Science
and
Institute for Theoretical Computer Science
Romanian Academy – the Iaşi Branch
{dcristea,cris}@infoiasi.ro

Abstract

The paper proposes a scheme for hierarchical representation of XML annotation standards. The representation allows individual work on documents displaying partial fitness in markings, mixing of annotated documents observing or not the same standard, as well as concurrent annotation. The approach allows access to different annotations of a corpus, with minimal representation overhead, which also facilitates accommodation of different, even incompatible, annotations of the same data. Two methods to build a hierarchical representation of annotation standards are shown, one allowing explicit declarations and the other inferring the hierarchy from a set of consistently annotated documents. Merging and extraction operations, which produce derived documents from existing ones are described. A system that implements the formal declarations of the hierarchy and the operations over it is presented.

1. Introduction

The more deeply the linguistic research, the more sophisticated the annotation required. Recently, since XML has become a *de facto* standard for the representation of annotated corpus resources (Ide, Bonhomme, and Romary, 2000), the sophistication of types of processing over texts, speech or multi-media documents resulted in the production of over-crowded marked documents. Annotation in corpora is not only used to record experts' view on specific linguistic phenomena, but also to store intermediate results in pipe-line NLP architectures and to post NLP results on the Web (Cunningham *et al*., 2002). But not always and not for each step in a processing chain are all layers of annotation useful. Usually an NLP step uses as input a document conforming to a certain annotation standard to which it adds another layer of annotation. Also, a human expert uses a tool to annotate a certain document. During the annotation process the expert can make use of some previous annotation layers that, through an interactive tool, can help the task at hand. In all cases, there are reasons to consider that, for a specific task (automatic processing or manual annotation), some existing markings in the input document are useful while others are not and, therefore, could be obscured. Examples of corpora use of this kind are corpus annotation in teamwork and re-usage of annotated corpora. A certain research task in teamwork could require individual experts or software modules, each exhibiting specific knowledge, to annotate at different layers the same original document. Individual results should be merged together in an attempt to compose a document that includes all involved layers. Another example of mixed annotation is given by research tasks that employ existing corpora, to which supplementary annotation layers are added. Heavily annotated corpora obtained in these ways could then be used to draw inter-layer correlations.

The paper reconsiders and enhances a hierarchical scheme to represent annotation standards, proposed in (Cristea *et al*., 1998), to which a processing machinery is added. Annotation standards are represented in a hierarchy, which enables multiple views over a document. Navigation within the hierarchy observes inheritance criteria. The approach allows access to different annotations of a corpus, with minimal representation overhead, which also facilitates accommodation of different (and sometimes incompatible) annotations of the same data. The approach prefers a standoff encoding scheme to an embedded one (Thompson and McKelvie, 1997). Potentially, the original hub (empty annotation) document resides in an URL that could be different from the one on which the annotation is added. Then, any annotation brackets around a piece of text can be recorded separate from the flesh data through their beginning and end character offsets onto the original text. As such, the hub string, identical in all documents, serves as the absolute system of reference.

We show how relations between different markings can be described in the hierarchy and how the directed acyclic graph representation can accommodate circular dependencies between annotations standards. Two methods to build such a graph are shown, one allowing explicit declarations and the other inferring the hierarchy from a set of consistently annotated documents. The case of concurrent annotations over the same hub document is discussed and a solution for contradictory (overlapping) representations is proposed. Finally, we introduce a set of operations that simplify an existent annotated document and combine two different annotations over the same hub document into a unique one.

2. The Hierarchy – a Lattice Representation

In our approach, different layers of annotation over a corpus are codified as a hierarchy of annotation standards (directed acyclic graph, or DAG). A node in the hierarchy is described according to the following syntax:

```
<standard name=″standard-name″
parents=″list-of-parents″>
  <tag name="tag-name" attributes="list-of-
attributes"/>
  …
  <ref source-tag="tag-name" source-
attribute="attribute-name" target-tag="tag-
name" target-attribute=″attribute-name″>
  …
</standard>
```

A standard (node) name is a unique symbol in the hierarchy. A standard inherits all features of all its parents. To avoid conflicts, in the present implementation no preference inheritance criteria are given, which means that the features belonging to the parents of a node are supposed to be orthogonal. Features which are new to a standard, vis-à-vis of those inherited, are defined in between `<standard></standard>` brackets by any number of `<tag>` and `<ref>` labels. A `<tag>` label records a new XML element tag. It has a name (label) and a list of attributes. A `<ref>` label records a semantic relation (dependency) between two annotation standards. It describes a reference between an attribute, called `source-attribute`, belonging to an XML tag, called `source-tag`, of the current standard (the one that contains the `ref` description) and another attribute, called `target-attribute`, belonging to another XML tag, called `target-tag` of a superior standard[1]. A standard *A* is superior to a standard *B* if and only if there is a path from *B* to the root of the hierarchy that passes through *A*.

We say that *a node A subsumes a node B in the hierarchy* (therefore *B* is a descendent of *A*) if and only if:

- any tag-name of *A* is also in *B*;
- any attribute in the list of attributes of a tag-name in *A* is also in the list of attributes of the same tag-name of *B*;
- any semantic relation which holds in *A* also holds in *B*;
- either *B* has at least one tag-name which is not in *A*, and/or there is at least one tag-name in *B* such that at least one attribute in its list of attributes is not in the list of attributes of the homonymous tag-name in *A*, and/or there is at least one semantic relation which holds in *B* and which doesn't hold in *A*.

As such, a hierarchical relation between a node *A* and one descendent *B* describes *B* as an annotation standard which is more informative than *A* and/or defines more semantic constrains.

Figure 1 displays an example of a declaration of a hierarchy of linguistic annotations. The definition builds a lattice, as that in Figure 2, which intends to describe different layers of annotation useful in many NLP applications. `ST-ROOT` represents the "empty" annotation (no tags), therefore describing the hub document of free text. Immediately under this trivial standard, three standards, `ST-TOK`, `ST-SEG` and `ST-PAR` are placed. `ST-TOK` is intended to identify tokens, as words and punctuation, and to mark words' lemmas, `ST-SEG` marks borders between elementary discourse units (*edu*s), like in (Marcu, 2000), and `ST-PAR` simply marks paragraphs. `ST-POS` is placed under `ST-TOK`. This standard does not contribute with new tags to the `TOK` labels inherited but adds the part-of-speech information through its attribute `pos`. The standard `ST-POS` is a parent for both `ST-NP` and `ST-VP`, which are supposed to mark noun phrases (NPs) and verb phrases (VPs), respectively. Tags of these kinds indicate also the heads of the corresponding compounds, as `id`s of `TOK` tags corresponding to the headwords. The `ref` definitions specify that the `head-id` attribute of the `NP` and `VP` tags should be filled with values of the `id` attribute of the `TOK` tags. Then, `ST-COREF`, placed under `ST-NP`, is a standard, which intends to mark anaphoric links between co-referential NPs. It supplements the `NP` tag with a `coref` attribute. The `ref` definition evidences the constraint that a `coref` attribute of an anaphoric NP indicates the `id` attribute of the antecedent NP. `ST-SEG-NP-VP` is a standard of an annotation, which marks simultaneously noun phrases, verb phrases and discourse units boundaries. It adds no new markings to those inherited from its three parents. Finally, `ST-COREF-IN-SEG` is a standard in which the coreferences and segment boundaries are marked, while `ST-PAR-SEG-NP-VP` adds the paragraph layer annotation to the markings for NPs, VPs and *edu*s.

```
<?xml version="1.0"?>
<ROOT>
<standard name="ST-ROOT"/>
<standard name="ST-TOK" parents="ST-ROOT">
  <tag name="TOK" attributes="id lemma"/>
</standard>
<standard name="ST-POS" parents="ST-TOK">
  <tag name="TOK" attributes="pos"/>
</standard>
<standard name="ST-NP" parents="ST-POS">
  <tag name="NP" attributes="id head-id"/>
  <ref source-tag="NP" source-
attribute="head-id" target-tag="TOK"
target-attribute="id"/>
</standard>
<standard name="ST-VP" parents="ST-POS">
  <tag name="VP" attributes="id head-id"/>
  <ref source-tag="VP" source-
attribute="head-id" target-tag="TOK"
target-attribute="id"/>
</standard>
<standard name="ST-COREF" parents="ST-NP">
  <tag name="NP" attributes="coref"/>
  <ref source-tag="NP" source-
attribute="coref" target-tag="NP" target-
attribute="id"/>
</standard>
<standard name="ST-SEG" parents="ST-ROOT">
  <tag name="SEG" attributes="id"/>
</standard>
<standard name="ST-SEG-NP-VP" parents="ST-
SEG ST-NP ST-VP"/>
<standard name="ST-PAR" parent="ST-ROOT">
  <tag name="PAR" attributes="id"/>
</standard>
<standard name="ST-COREF-IN-SEG"
parents="ST-SEG ST-COREF"/>
<standard name="ST-PAR-SEG-NP-VP"
parents="ST-PAR ST-SEG-NP-VP"/>
</ROOT>
```

Figure 1: Declarations of a hierarchy of annotation standards

[1] There is no a-priory motivation for which to call one attribute source and another target, apart from the fact that, usually, the target attribute is the `id` attribute of the target tag. Moreover all target attributes belong to nodes placed upper in the hierarchy.

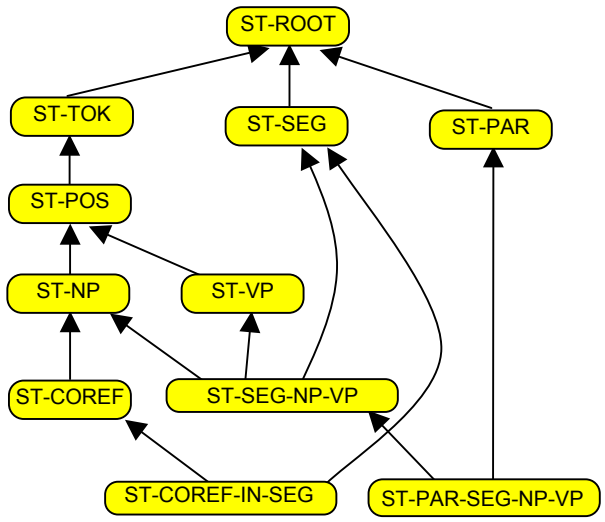


Figure 2: The hierarchy of annotation standards

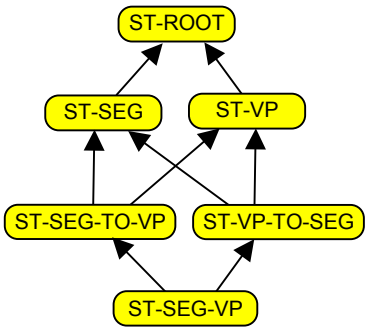
Figure 3 presents an example of a portion of George Orwell's novel "Ninety Eighty Four" annotated conforming to the `ST-COREF-IN-SEG` standard.

```
<?xml version="1.0" encoding="ISO-8859-1"
?>
<ROOT>
<SEG id="0">
 <NP head-id="2" id="0">
  <TOK id="2" pos="N"
lemma="Winston">Winston</TOK>
 </NP>
  <TOK id="3" pos="V" lemma="be">was</TOK>
  <TOK id="4" pos="ING"
lemma="dream">dreaming</TOK>
  <TOK id="5" pos="PREP" lemma="of">of</TOK>
 <NP head-id="7" id="2">
   <NP head-id="6" id="1" coref="0">
    <TOK id="6" pos="PRON"
lemma="he">his</TOK>
   </NP>
   <TOK id="7" pos="N"
lemma="mother">mother</TOK>
  </NP>
  <TOK id="8" pos="PUNCT">.</TOK>
</SEG>
<SEG id="1">
 <NP head-id="9" id="3" coref="0">
  <TOK id="9" pos="PRON"
lemma="he">He</TOK>
 </NP>
 <TOK id="10" pos="V"
lemma="must">must</TOK>
 <TOK id="11" pos="PUNCT">,</TOK>
</SEG>
<SEG id="2">
 <NP head-id="12" id="4" coref="0">
  <TOK id="12" pos="PRON"
lemma="he">he</TOK>
 </NP>
 <TOK id="13" pos="V"
lemma="think">thought</TOK>
 <TOK id="14" pos="PUNCT">,</TOK>
</SEG>
<SEG id="3">
 <TOK id="15" pos="V"
lemma="have">have</TOK>
 <TOK id="16" pos="EN" lemma="be">been</TOK>
  <NP head-id="20" id="5">
   <TOK id="17" pos="NUM"
lemma="ten">ten</TOK>
   <TOK id="18" pos="CC" lemma="or">or</TOK>
   <TOK id="19" pos="NUM"
lemma="eleven">eleven</TOK>
   <TOK id="20" pos="A"
lemma="years_old">years_old</TOK>
  </NP>
</SEG>
<SEG id="4">
 <TOK id="21" pos="ADV"
lemma="when">when</TOK>
  <NP head-id="23" id="6" coref="2">
    <NP head-id="22" id="5" coref="0">
     <TOK id="22" pos="PRON"
lemma="he">his</TOK>
   </NP>
   <TOK id="23" pos="N"
lemma="mother">mother</TOK>
 </NP>
 <TOK id="24" pos="V" lemma="have">had</TOK>
 <TOK id="25" pos="EN"
lemma="disappear">disappeared</TOK>
 <TOK id="26" pos="PUNCT">.</TOK>
</SEG>
<SEG id="5">
 <NP head-id="27" id="7" coref="2">
 <TOK id="27" pos="PRON"
lemma="she">She</TOK>
 </NP>
 <TOK id="28" pos="V" lemma="be">was</TOK>
  <NP head-id="36" id="9" coref="2">
   <TOK id="29" pos="DET" lemma="a">a</TOK>
   <TOK id="30" pos="A"
lemma="tall">tall</TOK>
   <TOK id="31" pos="PUNCT">,</TOK>
   <TOK id="32" pos="N"
lemma="statuesque">statuesque</TOK>
   <TOK id="33" pos="PUNCT">,</TOK>
   <TOK id="34" pos="ADV"
lemma="rather">rather</TOK>
   <TOK id="35" pos="A"
lemma="silent">silent</TOK>
   <TOK id="36" pos="N"
lemma="woman">woman</TOK>
   <TOK id="37" pos="PREP"
lemma="with">with</TOK>
    <NP head-id="39" id="8">
     <TOK id="38" pos="A"
lemma="slow">slow</TOK>
     <TOK id="39" pos="N"
lemma="movement">movements</TOK>
   </NP>
  </NP>
 <TOK id="44" pos="PUNCT">.</TOK>
</SEG>
</ROOT>
```

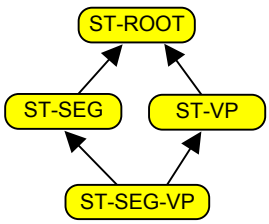
Figure 3: Example of annotation

3. Representing Circular References

From the above subsumption definition it follows that a standard *B* that includes references to tags belonging to another standard *A* should be placed under *A* in the hierarchy. But what if *A* refers *B* as well? Imagine, for instance, an annotation standard in which we have VPs (verb phrases) and SEGs (elementary discourse units) and we want to record for each VP the unit it belongs to, and in each SEG the head VP. Following the above observation, this would raise circularities, which are not acceptable in a DAG structure. Suppose ST-SEG and ST-VP are the standards corresponding to the initial markings that contain the SEG tags, respectively the VP tags, and neither of the two includes references to the other. Because we do not want to modify the existing standards, we can place a new standard, say ST-SEG-TO-VP, under both ST-SEG and ST-VP, is which the SEG tags contain an attribute `head` pointing the `id` of the head VP. Similarly, a standard ST-VP-TO-SEG, placed also under ST-SEG and ST-VP, will enrich the VP tags with an attribute, say `belongs-to`, pointing the `id` of the surrounding SEG. Finally, a standard ST-SEG-VT, child of both ST-SEG-TO-VP and ST-VP-TO-SEG, would inherit both attributes `head` and `belongs-to` from its parents without adding anything else. The result is a hierarchy as that in Figure 4a, whose corresponding description is given in Figure 5a. However, if the intermediate standards ST-SEG-TO-VP and ST-VP-TO-SEG are not useful by themselves, they can be deleted without any loss, such that only the final ST-SEG-VT be kept, child of both ST-SEG and ST-VP, as in Figure 4b, and the description given in Figure 5b. The circular-like constraints appear in the two `ref` declarations of the ST-SEG-VT standard.



a.



b.

Figure 4: Variants of hierarchical representations: without (a.) and with (b.) circular patterns of `ref` constraints

It follows that the representation of XML standards that we propose is not in contradiction with some constrains which can have circular patterns.

```
<ROOT>
<standard name="ST-ROOT"/>
<standard name="ST-SEG" parents="ST-ROOT">
  <tag name="SEG" attributes="id"/>
</standard>
<standard name="ST-VP" parents="ST-POS">
  <tag name="VP" attributes="id"/>
</standard>
<standard name="ST-SEG-TO-VP" parents="ST-
SEG ST-VP">
  <tag name="SEG" attributes="head"/>
  <ref source-tag="SEG" source-
attribute="head" target-tag="VP" target-
attribute="id"/>
</standard>
<standard name="ST-VP-TO-SEG" parents="ST-
SEG ST-VP">
  <tag name="VP" attributes="belongs-to"/>
  <ref source-tag="VP" source-
attribute="belongs-to" target-tag="SEG"
target-attribute="id"/>
</standard>
<standard name="ST-SEG-VP" parents="ST-SEG-
TO-VP ST-VP-TO-SEG"/>
</ROOT>
```

a.

```
<ROOT>
<standard name="ST-ROOT"/>
<standard name="ST-SEG" parents="ST-ROOT">
  <tag name="SEG" attributes="id"/>
</standard>
<standard name="ST-VP" parents="ST-POS">
  <tag name="VP" attributes="id"/>
</standard>
<standard name="ST-SEG-VP" parents="ST-SEG
ST-VP">
  <tag name="SEG" attributes="head"/>
  <tag name="VP" attributes="belongs-to"/>
  <ref source-tag="SEG" source-
attribute="head" target-tag="VP" target-
attribute="id"/>
  <ref source-tag="VP" source-
attribute="belongs-to" target-tag="SEG"
target-attribute="id"/>
</standard>
</ROOT>
```

b.

Figure 5: Declarations of the hierarchies in Figure 4

4. Automatic Classification

In order to interact with an existing hierarchy, one should be able to automatically place a new document within it. Two things are important here: compatibility of names and detection of semantic relations.

The first problem deals with name-spaces: in order for a document to be compared against a hierarchy it should be compatible with the tag and attribute names populating the hierarchy. If the annotations in the new document are semantically identical with those in the hierarchy but there exist name mismatches, compatibility can be achieved by

a translation mechanism. More complex compatibility adjustments can be obtained by working on values, as for instance exploding a range of values of an attribute into new attribute-value pairs.[2]

The semantic-relations problem deals with recognizing domain values intersection. The identity of values of two attributes or their intersection cannot be certified otherwise but by explicit declaration. Automatic detection is always prone to errors, which can be generated by fortuitous value fitness.

In our system, the classification module takes a hierarchy and an XML document and classifies the document within the hierarchy. The header of the document should declare the list of the semantic relations as a collection of `<ref>` records, enclosed in a pair of brackets `<semantic-relations>` … `</semantic-relations>`, each having the same syntax as in a hierarchy declaration:

```
<semantic-relations>
  <ref source-tag="tag-name" source-
attribute="attribute-name" target-tag="tag-
name" target-attribute="attribute-name"/>
  …
</semantic-relations>
```

The classification process proceeds in two steps. First the document to be classified is parsed and a collection of `<tag>` and `<ref>` declarations, having the same syntax as in the hierarchy declaration, is compiled. The `<tag>` records are computed by collecting all tags and their corresponding attributes of the XML elements, and the `<ref>` records – by simply reading the `<semantic-relations>` declarations in the header. Let's call this computed collection of `<tag>` and `<ref>`, the *witness collection*. Also, let's call the proper and inherited features of a node – the *node collection*.

The witness collection is matched against the node collections of the hierarchy, from top to down, starting in the root node. The classification of the witness collection down the hierarchy, generally follows the programming by classification paradigm (Mellish&Reiter, 1993). We say that the witness collection *satisfies the restrictions of a node collection* of the hierarchy (or *is classified under that node*) if the features of the node collections represent a subset of the features of the witness collection, therefore if all (`name`, `attributes`) pairs of the `<tag>` declarations of the node, and all (`source-tag`, `source-attribute`, `target-tag`, `target-attribute`) quadruples of the `<ref>` declarations of the node, proper and inherited, are part of the witness collection as well. In this way the witness collection "falls" down the hierarchy reaching certain levels, possibly more than just one. Below those levels, the witness collection cannot be classified any more under none of the nodes found there. The set of all down-most nodes the witness collection is classified under forms *a superior borderline*. In order to fulfil the process, *an inferior borderline* must also be determined. Two cases are possible: a). there is a set of nodes in the hierarchy which all have as parents the set of nodes on the superior borderline and only these nodes and all the corresponding node collections satisfy the witness collection. Then, the inferior borderline is given by the set of these nodes, and a new node should be included between the superior borderline and the inferior borderline (see figure 6a). b). either there is a set of nodes in the hierarchy which have as parents the set of nodes on the superior borderline and only these nodes, but none of these node collections satisfy the witness collection, or no common child of the superior borderline can be found. Then, the inferior borderline is not defined in the hierarchy and a leaf node is created and placed as child of the nodes belonging to the superior borderline (see figure 6b).

When the classification is completed, the search should continue beyond the nodes placed above the classified node in order to find other nodes that could be in a subsumption relation with the classified node. If such nodes are found, they should be added to the list of parents of the classified node.

5. Concurrent Annotations

Two annotations are called *concurrent* if they intend to represent the same linguistic phenomenon from different perspectives, therefore possibly resulting in different solutions.

Below we give two examples where concurrent annotations are needed:

a). **Same standard, different target documents**. Often, in order to validate an annotated corpus, different teams receive the same task and their work is compared. In case of agreement, the common solution is adopted with a high trust. In case of mismatch, either the controversial versions are given to a third judge, who is asked to decide in favour of one of the two solutions, or the subjects are persuaded to negotiate for an agreement. In these cases one would like to compile a unique document that keeps the common annotations, while indicating also the concurrent parts and the corresponding individual markings. In annotation tasks of these kinds it is likely that the agreed parts be significantly larger than the concurrent parts.

b). **Different standards and documents**. Suppose a corpus has do be syntactically annotated with respect to two distinct linguistic theories. In this case, two standards have to be considered. It is not impossible to imagine a certain research task, for instance comparing a phrase structure and a dependency structure, tempting to check whether a certain clausal constituent is in a given constituency relation within the sentence and in a certain functional dependency with respect to the main verb. It is likely that a pair of two such documents have no identical parts. However, it is also likely that a granularity border exists, up from where the two documents have the same structure and down from where the solutions are different. Then, in order to demonstrate the two approaches over the

[2] The morpho-syntactic descriptions, for example, use complex attribute-value pairs (as `msd="Ncmso"`), which can be expanded into a set of elementary features (`pos="noun" type="common" gender="masculine" number="singular" case="obligue"`).

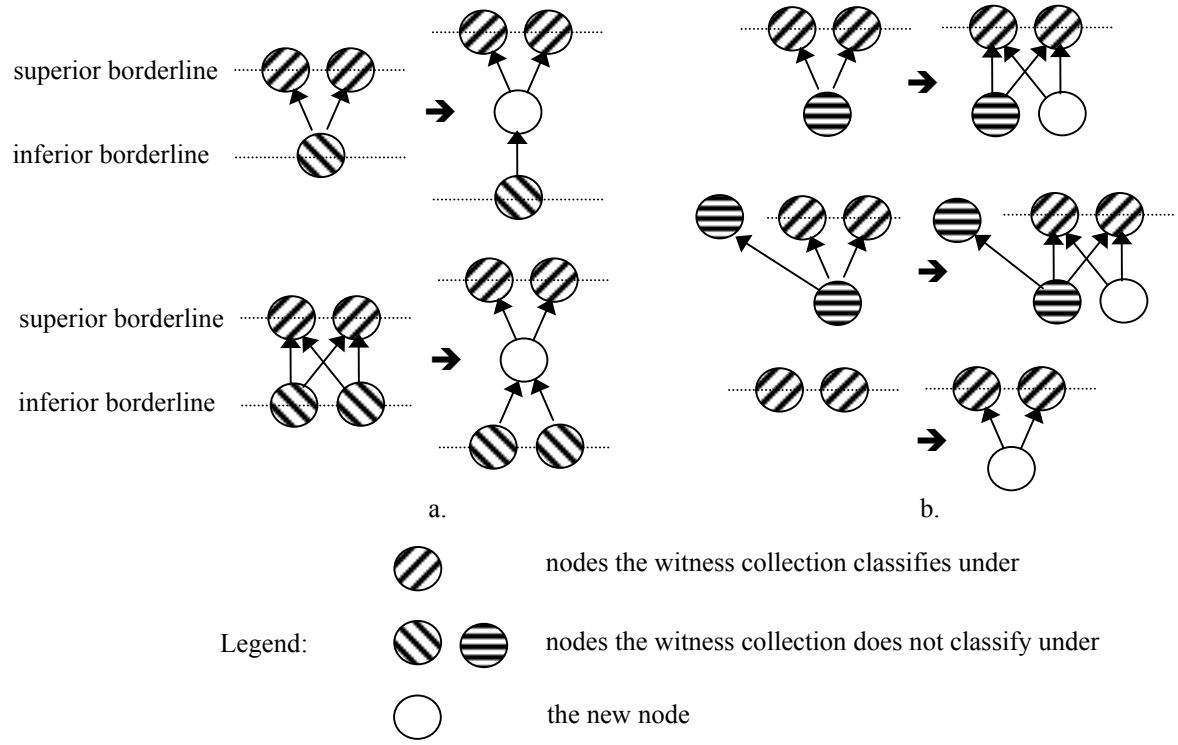


Figure 6: Examples of final classification

same text, a common layer of annotation should indicate at least this granularity limits (sentence, clause, etc.).

Viewed from the perspective of the hierarchical graph representation, documents conforming to concurrent annotation cannot be combined in a common standard, at least for the reason that the target XML documents would contain crossing markings. However, supposing a unique document is preferable to two different versions (for the reasons exposed above), one should allow for both common and concurrent annotations in the same document. Solutions to accommodate concurrent annotations have been previously proposed (Wit, 2002), (Sperberg-McQueen&Huitfeldt, 1999). In our system we represent concurrency in annotation as follows:

```
<concurrent>
  <version id="1" author=… etc…>
    <marking1>text1</marking1>…
    <marking1>text2</marking1>
  </version>
  <version id="2" author=… etc…>
    <marking2>text3</marking2>…
    <marking2>text4</marking2>
  </version>
</concurrent>
```

where `text1+text2=text3+text4`.

6. Operations within the hierarchy

Following the discussion above, our system implements a set of operations, as described in this section.

The *initialize-hierarchy* operation takes a document, headed by a `semantic-relations` statement, and builds a trivial hierarchy formed by the ROOT node (the empty annotation) and one standard corresponding to the annotation in the document.

The *classify* operation takes an existing hierarchy and a document, headed by a `semantic-relations` statement, and classifies the document with respect to the hierarchy, as described in section 4. It will end either by naming an existing standard in the hierarchy to which the document fully observes, or by placing a new standard in a certain place within the hierarchy. As such, building of a hierarchy can be done two ways: *ad-hoc*, by manually declaring it, when there is sufficient a-priori knowledge over a full range of corpus annotations, already existent or to-be-created, as shown in section 2; or *corpus-driven*, by an *initialize-hierarchy* command followed by any number of *classify* commands, when a range of annotated documents are used to inseminate a hierarchy. To note that in this case it is not compulsory for all annotated documents from which the hierarchy is triggered be replica of the same hub document. When annotation conventions are consistent within the collection of documents, different hub documents can be used to incrementally build a hierarchy of annotation standards.

Given a graph of annotation standards and documents annotated corresponding to these standards, all having the same hub document, *merge* and *extract* operations can be defined. A *merge* combines two documents having identical hubs and corresponding to two distinct nodes of the hierarchy, which are not in a subsumption relation, and produces, on one hand, another node in the hierarchy, descendant of the two input nodes, and, on the other, the

corresponding document which contains the union of the annotation tags of the two originating documents. An *extract* applies the reverse operation, extracting from a document, corresponding to a certain node, a document conforming to one of the node's ascendants in the hierarchy.

Finally, there are two types of *concurrent-checks*. One receives a standard name and two XML files, annotated versions of the same hub document, both supposed to observe the standard, and produces a file in which the annotation differences are put in evidence, as described in section 5, example a. The second receives two standard names and two files corresponding to these standards, and produces a difference file, as described in section 5, example b.

7. Conclusions

We described a data structure and a system aimed at facilitating the definition and exploitation of annotation standards over corpora. The system, interpreting the hierarchy definition declarations and implementing the described operations, has been built in Java. It is freely available and can be downloaded from the address: http://consilr.info.uaic.ro/~pic/lc.

As further developments, we intend to supplement the described operations with others, which will finally configure a complex environment, provided with a graphical interface, for working with annotated corpora. This environment could include, for instance, visualization of the hierarchy and interactive operations over it, including the deletion of nodes under some restrictions, unification of two hierarchies, cutting of a sub-hierarchy from an existing one, etc. To unify different annotation with identical or close semantics, we also intend to complement the tag and attribute names with a declarative semantic description. The final goal is to provide automatic conversion from an annotation name space to another, when the associated tags are semantically equivalent. This will aim at keeping a strict control over annotation standards, avoiding the proliferation of tag and attribute names.

We have acquired a collection of corpora, all based on George Orwell's "Ninety Eighty Four" novel as hub document, in both English and Romanian, on which the program was tested. In particular, a discourse parsing application, at present under development, makes heavy use of the merging operations on a rich hierarchy of standards. Also all resources used for the development of the Romanian WordNet, under the Balkanet and Balkanet-MEC projects (Tufis, Cristea, Stamou, 2004), have been classified in a unique hierarchy with the described system.

The described system can help efforts oriented towards the standardisation of language resources. To give an example, we intend to describe all resources which have been created or will be created for the Romanian language, and which are deposited on the site of the Consortium for the Romanian Language Technology, in an NLP-dedicated unique hierarchy. Using this hierarchy, each document will be assigned to a node, whose corresponding standard it observes.

Acknowledgements

The research presented in this paper has been partly supported by the EC funded IST-2000-29388 Balkanet project, and the Balkanet-MEC project funded by the Romanian Ministry of Education and Research.

References

Cristea, D., Ide, N. and Romary, L. (1998). Marking-up multiple views of a Text: Discourse and Reference, in *Proceedings of the First International Conference on Language Resources and Evaluation*, Granada.

Cunningham, H., Maynard, D., Bontcheva, K. and Tablan,V. (2002). GATE: A framework and graphical development environment for robust NLP tools and applications, in *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*.

Ide, N., Bonhomme, P., Romary, L. (2000). XCES: An XML-based Standard for Linguistic Corpora, in *Proceedings of the Second Language Resources and Evaluation Conference (LREC)*, Athens, Greece.

Sperberg-McQueen, C.M. and Huitfeldt, C. (1999). GODDAG: A Data Structure for Overlapping Hierarchies, in *Principles of Digital Document Processing*, Munich, Berlin, Springer Verlag. Early draft presented at the ACH-ALLC Conference in Charlottesville.

Marcu, D. (2000). The Theory and Practice of Discourse Parsing and Summarization. The MIT Press.

Mellish, C. and Reiter, E. (1993). Using Classification as a Programming Language, in *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-1993)*, Volume 1, pages 696-701. Morgan Kaufmann.

Thompson, H. and McKelvie, D. (1997). Hyperlink semantics for standoff markup of read-only documents, in *Proceedings of SGML Europe'97*, Barcelona.

Tufiş, D., Cristea, D. and Stamou, S. (2004). BalkaNet: Aims, Methods, Results and Perspectives. A General Overview, to appear in *Romanian Journal of Information Science and Technology*, Romanian Academy, Bucharest.

Witt, A. (2002). Meaning and interpretation of concurrent markup, in *Proceedings of ALLCACH2002, Joint Conference of the ALLC and ACH*, Tübingen.

TOWARDS AUTOMATIC ANNOTATION OF TEXT TYPE STRUCTURE: EXPERIMENTS USING AN XML-ANNOTATED CORPUS AND AUTOMATIC TEXT CLASSIFICATION METHODS

Hagen Langer[*], **Harald Lüngen**[†], **Petra Saskia Bayerl**[†]

[*] Justus-Liebig-Universität, Gießen
and Universität Osnabrück, Germany
hagen.langer@web.de

[†] Justus-Liebig-Universität, Gießen
Otto-Behaghel-Strasse 10D
35390 Giessen, Germany
{harald.luengen, petra.s.bayerl}@uni-giessen.de

Abstract

Scientific articles exhibit a fairly conventionalized structure in terms of topic types such as *background*, *researchTopic*, *method* and their ordering and rhetorical interrelations. This paper describes an effort to make such structures explicit by providing a corpus of German linguistic articles with XML markup according to a text type schema defining 21 topic type categories. The corpus is further augmented with XML annotations on a grammatical level and a logical structure level. The efficiency of an automatic annotation of text type structure is explored in experiments that apply general, domain-independent automatic text classification methods to text segments and employ features from the raw text level and the corpus annotations on the grammatical level. The results indicate that some of our topic types are successfully learnable.

1. INTRODUCTION

Nowadays the majority of scientific articles is published digitally in electronic libraries, on CDROMs/DVDs, and notably in the W3, e.g. on conference or journal sites, in online archives, or on researchers' home pages. The vast amount of information that is available via the new media at practically every point in time has afforded new techniques for goal-oriented search and retrieval, amongst other things, of scientific articles. Besides simple search via character strings in texts, many techniques require phases of preprocessing articles, e.g. by automatic classification, summarization or generation of metadata. One such technique is the CiteSeer approach of providing access to scientific articles on the W3 via automatically generated citation networks (Giles, Bollacker, & Lawrence, 1998). Another one is the categorization of whole articles into thematic categories such as scientific disciplines and sub-disciplines either automatically (e.g. Sebastiani, 2001) or manually.

This paper describes an approach to analyzing, annotating, and evaluating a corpus of scientific articles according to text type categories on a thematic level using XML technology and methods from automatic text categorization.

The text type structure of an article instantiates components, or *topic types* of research papers such as *background*, *researchTopic*, *method*, which are related by a canonical ordering and typical rhetorical relations, all of which constitute characteristic features of the text type, or genre, of scientific articles. Topic types have elsewhere been called *text level categories* (Kando, 1997), or *zones* (Teufel, Carletta, & Moens, 1999).

Building an annotation tool for thematic structure involves automatic classification of segments into topic type categories, thus we additionally provide XML annotations on other levels of information, namely grammar (syntax and morphology), and logical structure (structural positions according to DocBook markup, cp. Walsh & Muellner, 1999), that can provide features for the classification task. The current aim is thus to examine the correlations between thematic structure and the other levels of analysis to identify linguistic and structural features that constitute topic types. The overall goal of the project SemDoc is to design an empirically based thematic text type ontology that can be used for improved information retrieval/extraction, automatic text summarization and for making scientific articles available to the Semantic Web by automated annotation.[1]

In the remainder of this paper, a characterization of our corpus and the methods of analysis and feature extraction using XML annotations on multiple layers as well as automatic text segment classification experiments, will be presented.

2. TEXT TYPE STRUCTURE

Text type schemas representing text-level structure of scientific articles have been devised previously, for instance in the context of automatic text summarization. In Teufel (1999) (see also Teufel et al., 1999), a schema of the seven "argumentative zones" *BACKGROUND, OTHER, OWN, AIM, TEXTUAL, CONTRAST, BASIS* is employed for classifying the sentences of a scientific article and choosing the most suitable sentences for a summary of the article. In Kando (1997), a hierarchical schema with 51 bottom-level text constituent categories is presented that are similar to our topic types discussed below and were used for manually annotating sentences in Japanese research papers. In

[1] http://www.uni-giessen.de/germanistik/ascl/dfg-projekt/

two experiments, the usefulness of such an annotation in full-length text searching and passage extraction is explored and it is found that it could improve the results. It is also reported that several studies "indicated the feasibility of automatic assignment of categories using surface level natural language processing" (Kando, 1997, p.4). Our text type schema is based on these two approaches but occupies a middle ground between their sizes by including 21 bottom-level topic types, which are supposed to represent the typical structure of texts in the text type of scientific articles. Our aim was to develop an informative schema while sorting out categories we considered primarily functional (like 'Reason for...') and including only purely thematic categories. Moreover, we hypothesized that these 21 topic types could be well distinguished by structural and surface linguistic criteria. The schema is depicted in Figure 1. The edges can be interpreted to represent the *part-of* relation such that a type lower in the hierarchy is a part of the immediately dominating, more global type in terms of text type structure. The order of the categories represents a canonical, expected order of topic types in a scientific article. The text type schema was initially encoded as an XML Schema grammar where topic types are represented by elements that are nested such that the XML structure reflects the structure of the text type structure tree (Figure 2).

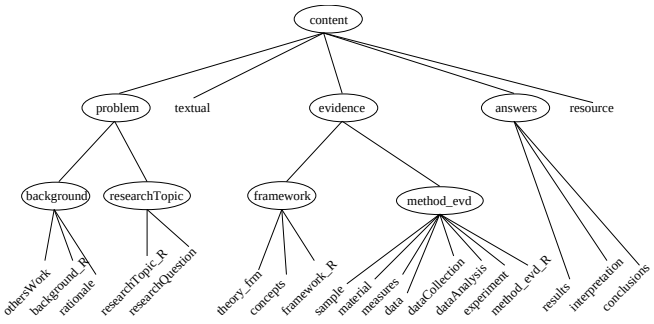


Figure 1: Text type schema

```
<xs:element name="problem">
  <xs:complexType>
    <xs:sequence>
       <xs:element name="background" minOccurs="0">
          <xs:complexType>
             <xs:sequence>
                <xs:element name="othersWork"
                            type="xs:string"
                            minOccurs="0"/>
                <xs:element name="background_R"
                            type="xs:string"
                            minOccurs="0"/>
             </xs:sequence>
          </xs:complexType>
       </xs:element>
       ...
```

Figure 2: XML Schema grammar (extract) for the text type schema

3. DATA COLLECTION

For the analyses and experiments described in this paper, a corpus of 47 research articles from the discipline of linguistics was collected. The articles were taken from the German online journal 'Linguistik Online'[2], from the volumes 2000-2003. The articles have an average length of 8639 word forms and deal with subjects as diverse as syntax and morphology, chat analysis, and language learning. To verify the validity of our approach for more than one discipline, we have also collected and analyzed 60 scientific articles from the field of psychology, however, these will not be considered in this report.

3.1. Annotation levels

Following Bayerl, Lüngen, Goecke, Witt, and Naber (2003), we distinguish between annotation *levels* and annotation *layers*. An annotation level is a chosen level of information that is initially independent of an annotation design, such as the morphology and syntax levels in linguistics. Annotation layer, in contrast, refers to the realization of an annotation as, for instance, XML markup. There need not be a 1:1-correspondence between annotation levels and layers. We would argue that, for example, in the layer defined by the XHTML DTD, at least one logical level and one layout level are integrated. Conversely, one annotation level may be distributed across several layers. As for the three annotation levels in our setting, one (the structural level) was realized as an independent layer, and two (thematic and grammatical level) were realized in one single annotation layer.

3.1.1. Thematic level

As sketched in section 2., the thematic text type schema represents, amongst other things, an expected canonical order of topic types in a scientific article. Yet, the order of topics in a specific article instance may deviate from it and render an XML instance annotated accordingly invalid. Thus we derive a flat version of the hierarchical XML schema by means of an XSLT style sheet. In the flat XML schema for the thematic annotation layer (called THM), topic types are represented as attribute values of elements called <`group`> and <`segment`>, instead of names of nested elements. The empty <`group`> elements represent topic types that corresponded to the nodes (as opposed to leaves or terminal categories) in the original tree of topic types. The original hierarchical structure is represented via the ID/IDREF attributes `id` and `parent`, similar to O'Donnell's XML representation of rhetorical structure trees (O'Donnell, 2000). Each text segment (a thematic unit, often but not always corresponding to a sentence) is annotated with one terminal topic type, including segments from abstracts, footnotes, or captions. An extract from a THM annotation can be seen in Figure 3.[3]

The HTML files containing the articles were automatically stripped off their markup, segmented and provided with skeletal markup according to the flat THM schema. Two annotators then had to fill in the attribute values for the topic types of segments using the XML spy editor.

[2] `http://www.linguistik-online.de/`

[3] The extract, which is also shown in Figure 4, is taken from Bühlmann (2002)

```
<segment id="s196" parent="g4" topic="results">In den
  Texten ist sehr oft nicht klar, ob ein Maskulinum nur
  auf Männer oder auch auf Frauen referiert.
</segment>
<segment id="s197" parent="g4" topic="interpretation">
  Wichtige Fragen, die die LeserInnen an den Text haben,
  bleiben somit unbeantwortet. Die Politik wird durch den
  fast durchgehenden Gebrauch des generischen Maskulinums
  als "Männersache" dargestellt, Frauen werden, auch wenn
  sie vorhanden sind, selten sichtbar gemacht. Zudem wird
  auch mit geschlechtsspezifisch männlichen Wörtern wie
  Gründerväter der Gedanke an Männer evoziert.
</segment>
```

Figure 3: THM annotation (extract)

```
<sect2>
 ...
 <para  POSINFO1="/article[1]/sect1[4]/sect2[4]/para[3]">
  In den Texten ist sehr oft nicht klar, ob ein Maskulinum
  nur auf Männer oder auch auf Frauen referiert. Wichtige
  Fragen, die die LeserInnen an den Text haben, bleiben
  somit unbeantwortet. Die Politik wird durch den fast
  durchgehenden Gebrauch des generischen Maskulinums als
  "Männersache" dargestellt, Frauen werden, auch wenn sie
  vorhanden sind, selten sichtbar gemacht.
 </para>
 <para POSINFO1="/article[1]/sect1[4]/sect2[4]/para[4]">
  Zudem wird auch mit geschlechtsspezifisch männlichen
  Wörtern wie Gründerväter der Gedanke an Männer evoziert.
 </para>
 ...
</sect2>
```

Figure 4: Annotation according to DocBook (extract)

3.1.2. Structural level

Although the linguistic articles in our corpus are originally provided with HTML markup, HTML itself was not considered as an annotation layer in our scheme, as HTML is known to be a hybrid markup language including a mixture of logical, functional and layout categories. For representing a purer logical structure, the HTML annotations were converted to DocBook markup (Walsh & Muellner, 1999). The DocBook standard was originally designed for technical documentation, but has recently been applied to other scientific writing and is devoid of layout elements such as `font` or `br` in HTML.

We did not employ the whole, very large official DocBook DTD, but designed a new XML schema with a subset of 45 original DocBook elements plus 13 new logical elements not conforming to the DocBook standard which were nevertheless needed for our purposes (for example `tablefootnote`, `toc`, and `numexample`). This reduced XML schema for DocBook was developed in collaboration with HyTex project at the University of Dortmund[4], to keep the number of admissible elements manageable.

Since the segmentation on the THM layer is into thematic units, whereas the DocBook elements pertain to logical structure units, we did not want to constrain the THM annotation layer to be fully compatible with the DocBook level, requiring that a possible integrated THM-DocBook annotation layer would always yield well-formed XML. Thus the DocBook annotation was realized as a separate XML layer (called DOC).

The annotations were obtained using a perl script that provided the raw DocBook annotations from the HTML marked up texts, and the XML spy editor for validation and manually filling in DocBook-elements that have no correspondences in HTML. In addition, structural position attributes were added by means of an XSLT stylesheet. These 'POSINFO' attributes make explicit the position of an element in the XML DOM tree of the document instance. The aim is to exploit the position information in the automatic classification of thematic segments in the future. The DocBook annotation of the extract shown in Figure 3 can be seen in Figure 4.

3.1.3. Grammatical level

For an annotation of morphological and syntactic categories to word form tokens in our corpus, the commercial tagger Machinese Syntax by Connexor Oy was employed. This tagger is a rule-based, robust syntactic parser available for several languages and based on Constraint Grammar (Karlsson, Voutilainen, & Heikkilä, 1995) and Functional Dependency Grammar (Tapanainen & Järvinen, 1997). It provides morphological, surface syntactic, and functional tags for each word form and a dependency structure for sentences, and besides is able to process and output "simple XML". DTDs for the tag set and for the XML output format are supplied with the software.

Since all annotations provided by Machinese Syntax pertain to word forms (dependency structure is realized through ID/IDREF-attributes on word form tags), no conflicts in terms of element overlaps may arise between our THM annotation layer and a potential CNX annotation layer. Speaking in terms of the XML-based multiple layer annotation paradigm (Goecke, Naber, & Witt, 2003), the only meta-relation besides independence that may hold between THM-<`segment`> elements and CNX-tagging elements is *inclusion*. Therefore, the THM and CNX annotations could be integrated into one single annotation layer. This way, not only special tools for the analysis of multiple-layer annotations (Goecke et al., 2003; Bayerl, Lüngen, Gut, & Paul, 2003), but also the available query languages for querying information contained in single annotation layers, like XQuery[5], can be adopted for inferring correlations between topic types and grammatical features. Since the current version of Machinese Syntax is able to process only "simple XML", that is, XML without attributes, we implemented two XSLT style sheets, one of which converts our THM-annotations into attribute-free XML by integrating all attribute-value specifications into the names of their respective elements, and another one which reconverts the attribute-free annotations enriched by the CNX-tagging into complex XML.

Out of the large CNX-tag set documented in an extensive manual, we have selected a set of 15 tags (henceforth called CNX-15) that were judged to be valuable for automatic assignment of topic types to text segments of scientific articles. CNX-15 also includes simplified tag specifications that came as a bundle of tags in the original CNX output, cf. the following listing of the CNX-15 tags and their range of values in Table 1.

A third XSLT stylesheet acts as a filter and converter on the integrated THM-CNX annotations to output the THM

[4] `http://www.hytex.info`

[5] `http://www.w3.org/TR/xquery/`

segments plus their CNX-15 tagging in a THM-CNX target format designed for extracting statistics and feature vectors for the automatic classifier.

#	CNX-15 Tag	range of values
1	text	(string)
2	lemma	(lower case string)
3	cmp-head	(lemma of head constituent; lower case string)
4	depend	(dependency category, e.g. `loc`, `dur`, `frq`, i.e. adverbial of location, duration, frequency)
5	pos	N, V, A, ...
6	comparison	POS, SUP
7	nnum	SG, PL (singular or plural of nominal categories
8	numeral	CARD, ORD
9	pers	SG1, SG2, SG3, PL1, PL2, PL3
10	modal	MODAL (modal auxiliary)
11	fin	INF, IMP, SUBJUNCTIVE, PRES, PAS
12	ncomb	N+
13	unknown	<?>
14	aux	AUX
15	passive	PASS

Table 1: CNX-15 tags derived from the Machinese Syntax tag set

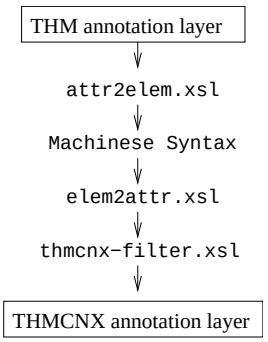


Figure 5: Augmenting THM annotations with grammatical tags

4. AUTOMATIC TEXT SEGMENT CLASSIFICATION EXPERIMENTS

In order to evaluate the feasibility of an automatic annotation of scientific articles according to our THM annotation level introduced in section 3.1.1., we applied different automatic text classification methods to text segments as shown in Figure 3.

The different configurations in the experiments were motivated by the following questions:

- Are thematic structures (of scientific articles) learnable using only general, domain-independent methods?
- Which kind of information (plain text, POS-tag information, morphological analysis, knowledge about the context) has impact on the classification accuracy?
- Which kind of classifier (e.g. KNN, Rocchio) performs best on this task?
- Are there particular topic types which are significantly easier to detect than others?

Our focus has been on the more general question, to which degree thematic structures are learnable, rather than the question how to develop a specialized classifier for the particular task of categorizing text segments according to our text type schema. Thus, in contrast to related work (Teufel, 1999), we restricted ourselves to general and domain-independent classification and pre-processing methods, which are, in principle, also applicable to any other kind of text.

In contrast to standard text classification tasks (Sebastiani, 2002, cf.), the pieces of text in our experiments are much smaller than ordinary documents, and sometimes consist only of a single word of phrase as in the case of headlines. On the other hand, the text segments to be classified appear in a context, which is an additional source of information, not available in case of the standard problem of document categorization.

4.1. Vector Representation

The classification experiments have been carried out at different levels of linguistic analysis:

- inflected word forms (from the raw text)
- stems (from the `lemma` annotation as shown in Table 1)
- part-of-speech patterns (from the `pos` annotation as shown in Table 1)
- head-lemma (the `cmp-head` annotation as shown in Table 1)

Some of these levels of description have also been used in combination (e.g. part-of-speech patterns combined with head lemmata).

For the purpose of our classification experiments each text segment has been represented as a (sparse) probability distribution vector of its units (*units* are e.g. inflected words, POS tags). Feature vectors have been generated directly from the THM-CNX annotation layer introduced above. We did not use TF*IDF or other feature weighting methods, but the bare probability of a term, given its segment[6]. Neither did we employ stop word lists or frequency-based filtering in order to reduce the feature space.

[6]TF*IDF weighting had been used in preliminary experiments, but did not improve the results.

4.2. KNN Classification

The basic idea of the K nearest neighbor (KNN) classification algorithm is to use already categorized examples from a training set in order to assign a category to a new object. The first step is to choose the K nearest neighbors (i.e. the K most similar objects according to some similarity metric, such as cosine) from the trainings set. In a second step the categorial information of the nearest neighbors is combined, in the simplest case, by determining the majority class.

The version of KNN classification, adopted here, uses the *Jensen-Shannon divergence* (also known as *information radius* or *iRad*) as a (dis-)similarity metric:

$$\text{iRad}(q,r) = \tfrac{1}{2}[D(q\|\tfrac{q+r}{2}) + D(r\|\tfrac{q+r}{2})]$$

where $D(x\|y)$ is the Kullback-Leibler divergence (KL divergence) of probability distributions x and y:

$$D(x\|y) = \sum_{i=1}^{n} x(i)(log(x(i)) - log(y(i)))$$

iRad ranges from 0 (identity) to $2log2$ (no similarity) and requires that the compared objects are probability distributions.

Let be $N_{O,C} = \{n_1, \ldots, n_m\}$ $(0 \leq m \leq K)$ the set of those objects among the K nearest neighbors of some new object O that belong to a particular category C. Then the score assigned to the classification $O \in C$ is

$$\text{score}(O,C) = \sum_{j=1}^{m} \text{iRad}(O,n_j)^E.$$

Depending on the choice of E, one yields either a simple majority decision (if $E = 0$), a linear weighting of the iRad similarity (if $E = 1$), or a stronger emphasis on closer training examples (if $E > 1$). Actually, it turned out that very high values of E improved the classification accuracy. Finally, the KNN scores for each segment were normalized to probability distributions, in order to get comparable results for different K and E, when the KNN classifications get combined with the bigram model (see section 4.3., below).

4.3. Bigram Model

The bigram model gives the conditional probability of a topic type T_{n+1}, given its predecessor T_n.

For a sequence of segments $s_1 \ldots s_m$ the total score $\tau(T, s_i)$ for the assignment of a topic type T to s_i is the product of the bigram probability, given the putative predecessor topic type (i.e. the topic type T' with the highest $\tau(T', s_{i-1})$ computed in the previous step), and the normalized score of the KNN classifier. The total score of the topic type sequence is the product of its τ scores.

4.4. Training and Evaluation

For the classification experiments we used the data collection described in section 3.. For each test document the bigram model and the classifier were trained with all other documents. The overall size of the data collection was 47 documents. Thus, each classifier and each bigram model has been trained on the basis of 46 documents, respectively. The total number of segments was 7330. 23 different classes[7] have been manually assigned to the segments of the sample[8]. The number of features varied by the respective choice of data representation: The total number of stems was 33,000 (about 400,000 tokens), the total number of POS tag types was 14.

Additional experiments have been carried out using a simplified Rocchio classifier. This classifier computes the centroid vector for each class and assigns the category of the centroid vector that has the least iRad distance relative to the segment in question.

4.5. Results

We performed several hundred classification tests with different combinations of data representation, classification algorithm, and classifier parameter setting. Table 2 summarizes some results of these experiments, table 3 shows the precision and recall values of the K-13-E-40 classifier with bigram model (last line in table 2) for each topic type. For illustrative purpose, we also included a configuration, where all other segments (i.e. including those from the same document) were available as training segments ('KNN*' in the butlast line of table 2).

classifier	data	K	E	accuracy classifier	accuracy classifier + bigram
KNN	head	13	45	39.433	42.050
KNN	POS	20	40	40.328	41.751
KNN	stem	17	45	38.959	41.196
Rocchio	POS+head	-	1	36.099	20.876
KNN*	POS+head	13	40	54.416	-
KNN	POS+head	13	40	43.812	45.872

Table 2: Results

The standard deviation across topic types of about 24 (both for recall and precision) indicates that the "learnability" of topic types differs enormously. The topic type `resource` has been learned almost perfectly, while other topic types (e.g. `material`) have no recall, at all.

4.6. Discussion

The data collection used for the classification experiments is restricted in many respects: one language (German), one type of document (scientific article), one thematic domain (linguistics), one thematic ontology, and only 46 training documents. Thus, the results of our experiments can only give a rough idea of the lower bound of the accuracy that can be achieved by the application of general,

[7] The classes void_C and void_meta in Table 2 were non-thematic labels assigned to incomplete segments and metadata in the corpus (such as author, affiliation, and acknowledgements), respectively. Thus, they are not part of the abstract thematic schema depicted in Figure 1.

[8] The number of training examples per class ranges from 5 (`experiment`) to 1643 (`resource`). 3 classes have less than 10 training examples.

class	recall	precision
background_R	16.346	23.944
concepts	1.639	5.770
conclusions	29.602	25.813
data	6.195	25.000
dataAnalysis	0.442	3.846
dataCollection	0.000	0.000
experiment	0.000	0.000
framework_R	30.914	23.842
interpretation	15.209	21.277
material	0.000	0.000
measures	0.000	0.000
method_evd_R	5.556	40.000
othersWork	72.673	31.311
rationale	0.000	0.000
researchQuestion	23.296	75.926
researchTopic_R	34.163	45.619
resource	97.018	93.490
results	27.343	24.895
sample	0.000	0.000
textual	29.750	40.067
theory_frm	0.000	0.000
void_C	0.000	0.000
void_meta	67.083	83.420

Table 3: Recall and precision

domain-independent classification methods to this particular kind of document. The upper bound (e.g. if larger training sets are available) still remains unclear. Additionally, the classification experiments reported in this paper are, to our knowledge, the first attempt to apply domain-independent machine learning methods to the problem of identifying the topic types of text segments. Because of the novelty of the approach, there are no "baseline" results that can serve as a standard.

Besides the limitations, stated above, there are some interesting results:

- The accuracy of the best configuration is close to 50%
- The choice of the classification algorithm seems to play an important role (Rocchio vs. KNN).
- The POS-tag distribution of text segments turned out to be nearly as informative as the "bag-of-words" representation.
- The usage of a bigram model improved the accuracy results in almost all configurations.
- The variance of classification accuracy across topic types is extremely high.

5. CONCLUSION AND PROSPECTS

Many applications, e.g. in the context of the Semantic Web, require rich and fine-grained annotations on linguistic levels. In this paper we presented a multiple layer approach to the semantic, grammatical and structural annotation of scientific articles. We carried out experiments on automated annotation of text segments with topic types, using general and domain-independent machine learning methods. We achieved an average accuracy of almost 50%. Although the results probably suffer from limitations of our data collection (small sample size, restricted thematic domain), our main conclusion is that at least some of the topic types of our hierarchy are successfully learnable. Other classification algorithms (e.g. support vector machines), feature selections methods, and/or larger training sets may yield further improvements. Our future work will focus on the integration of structural position information from the DOC annotation layer, usage of additional information from deep syntactic analyses, and the question to which degree our results are generalizable to other thematic domains and languages.

References

Bayerl, P. S., Lüngen, H., Goecke, D., Witt, A., & Naber, D. (2003). Methods for the semantic analysis of document markup. In *Proceedings of the ACM symposium on document engineering (DocEng 2003)*. Grenoble.

Bayerl, P. S., Lüngen, H., Gut, U., & Paul, K. (2003). Methodology for reliable schema development and evaluation of manual annotations. In *Workshop notes for the workshop on knowledge markup and semantic annotation, second international conference on knowledge capture (K-CAP 2003)* (p. 17-23). Sanibel, Florida.

Bühlmann, R. (2002). Ehefrau Vreni haucht ihm ins Ohr... Untersuchungen zur geschlechtergerechten Sprache und zur Darstellung von Frauen in Deutschschweizer Tageszeitungen. *Linguistik Online, 11*. (`http://www.linguistik-online.de`)

Giles, C. L., Bollacker, K., & Lawrence, S. (1998). CiteSeer: An automatic citation indexing system. In I. Witten, R. Akscyn, & F. M. Shipman III (Eds.), *Digital libraries 98 - the third ACM conference on digital libraries* (pp. 89–98). Pittsburgh, PA: ACM Press.

Goecke, D., Naber, D., & Witt, A. (2003). Query von Multiebenen-annotierten XML-Dokumenten mit Prolog. In U. Seewald-Heeg (Ed.), *Sprachtechnologie für die multilinguale Kommunikation. Beiträge der GLDV-Frühjahrstagung, Köthen 2003* (Vol. 5, p. 391-405). Sankt Augustin: gardez!-Verlag.

Kando, N. (1997). 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* (p. 68-81).

Karlsson, F., Voutilainen, A., & Heikkilä, J. (Eds.). (1995). *Constraint grammar: a language-independent system for parsing unrestricted text* (Vol. 4). Berlin and N.Y.: Mouton de Gruyter.

O'Donnell, M. (2000). RSTTool 2.4 – A markup tool for Rhetorical Structure Theory. In *Proceedings of*

the international natural language generation conference (INLG'2000) (pp. 253 – 256). Mitzpe Ramon, Israel.

Sebastiani, F. (2001). Organizing and using digital libraries by automated text categorization. In L. Bordoni & G. Semeraro (Eds.), *Proceedings of the AI*IA workshop on artificial intelligence for cultural heritage and digital libraries* (p. 93-94). Bari, Italy.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1-47.

Tapanainen, P., & Järvinen, T. (1997). A non-projective dependency parser. In *Proceedings of the 5th conference on applied natural language processing* (p. 64-71). Washington D.C.

Teufel, S. (1999). *Argumentative zoning: Information extraction from scientific text.* Unpublished doctoral dissertation, University of Edinburgh.

Teufel, S., Carletta, J., & Moens, M. (1999). An annotation scheme for discourse-level argumentation in research articles. In *Proceedings of EACL.*

Walsh, N., & Muellner, L. (1999). *DocBook: The definitive guide.* O'Reilly.

ACKNOWLEDGEMENT

This work was supported by the German Research Foundation (DFG) in the context of research group nr. 437 *Texttechnologische Informationsmodellierung*.

Multiple perspectives on text using multiple resources: experiences with XML processing

Peter Fankhauser[*], **Elke Teich**[**]

[*]Fraunhofer Integrated Publication and Information Systems Institute (IPSI)
Dolivostr. 15, 64293 Darmstadt, Germany
fankhaus@ipsi.fraunhofer.de

[**]Darmstadt University of Technology
Department of English Linguistics
Hochschulstr. 1, 64289 Darmstadt, Germany
teich@linglit.tu-darmstadt.de

Abstract

We present a system for the linguistic exploration of lexical cohesion that uses two existing natural language resources, the Brown Corpus and the Princeton WordNet. Both are represented in XML. The system computes potential lexical chains for the texts in the corpus using a measure of semantic neighbourhood and constraints on the text. This process is implemented in XSLT/XPath. The results can be inspected from various perspectives by means of an XML-browser. We report on our experiences with XML-processing in this application, discussing both the advantages and the shortcomings of relying on XML as the sole representational and programming paradigm.

1. Introduction

Using XML for the representation of corpora with multiple layers of annotation has become an implicit standard in both computational and corpus linguistics (cf. TEI, XCES).[1] It is also a widely established practice to produce a particular type of annotation with a special-purpose tool (e.g., a tagger for part-of-speech annotation) that outputs XML or maps its output (e.g., a TSV) onto XML. The annotation result can thus figure as an independent layer that can potentially be related to other independently produced layers of annotation (e.g., shallow syntax, named entities, co-reference). The properties of XML as a data structure and the available XML-processing functionalities, including database support if needed, can then conveniently be used to ensure the integrity of an integrated multi-layer corpus (validation, consistency checking, alignment of layers etc.; cf. (Teich et al., 2001)). However, there are still a number of unresolved issues when it comes to the *deployment* of a multi-layer resource encoded in XML, be it for the purpose of linguistic analysis or further computational processing. Notably, there are no readily available solutions for corpus inspection and query. Basing query on XML-related standards, such as XSLT, XQuery and/or XPath means using general-purpose programming languages for a specialized task, where often it would be more straightforward to work with special-purpose languages, such as regular expressions. Dedicated query languages, on the other hand, are more often than not restricted to one layer of annotation.

In this paper, we report on our experiences in building a system for exploring lexical cohesion that is implemented using solely XML and XSLT/XPath. We briefly introduce the concept of lexical cohesion and describe the present application (Section 2). We then introduce the two existing resources we have used to annotate text with potential lexical-cohesive ties – the SEMCOR version of the Brown Corpus (Landes et al., 1998) and the Princeton WordNet (Miller et al., 1990; Fellbaum, 1998) – and give a detailed account of the XML-processing of these resources and the resulting multi-layer representation (Section 3). Finally, we critically assess the use of XML for a complex corpus application of the present kind (Section 4).

[1]TEI: `www.tei-c.org.P4X`;
XCES: `www.cs.vassar.edu/XCES`

2. The application

2.1. The concept of lexical cohesion

Lexical cohesion is the central device for making texts hang together experientially, defining the aboutness of a text (field of discourse) (cf. (Halliday and Hasan, 1976, Chapter 6)). Along with reference, ellipsis/substitution and conjunctive relations, lexical cohesion is said to formally realize the semantic coherence of texts, where lexical cohesion typically makes the most substantive contribution (according to (Hasan, 1984), around fifty percent of a text's cohesive ties are lexical).

The simplest type of lexical cohesion is *repetition*, either simple string repetition or repetition by means of inflectional and derivational variants of the word contracting a cohesive tie. The more complex types of lexical cohesion rely on the system of semantic relations between words, which are organized in terms of *sense relations* (synonymy, hypernymy, hyponymy, antonymy, meronymy) (cf. (Halliday and Hasan, 1976, 278-282)). Potentially, any occurrence of repetition or of relatedness by sense can form a cohesive tie, i.e., not every instance of semantic relatedness between two words in a text does necessarily create a cohesive effect.

While detailed analyses of small samples of text (e.g., (Hoey, 1991)) bring out some tendencies of how lexical cohesion is achieved, large amounts of texts annotated for lexical ties are needed as a basis for empirically testing those tendencies. Carrying out manual analyses of large amounts of text in terms of lexical cohesion is very labor-intensive, however, and the level of inter-annotator agreement is typically not satisfactory. Thus, an automatic procedure is called for. The first attempt in this direction is due to (Morris and Hirst, 1991), using Roget's Thesaurus as a basis. This idea is taken up again by (Barzilay and Elhadad, 1997) in the context of text summarization. Barzilay and Elhadad implement a fully-automatic procedure using WordNet for detecting lexical-semantic relations between words. Related words are put into lexical chains and the strongest chains are then used to extract key sentences from the text for a summary. Apart from simple repetition (extra-strong relation), the relatedness criteria Barzilay and Elhadad use are all to do with the systemic relations between words in the WordNet hierarchy, where synonymy and near-synonymy, hypernymy/hyponymy, meronymy/holonymy count as strong relations. For summarization, the lexical chains computed on this basis are good indicators of the key sentences in a text; but for a full-blown cohesion analysis, constraints acting in the text have to be taken into account as well. One such constraint is the distance of two sense-related words in a text. For example, if a word *object* in sentence 1 of a text with 30 sentences is related by co-hyponymy to a word *subject* in sentence 20, a cohesive effect between the two is unlikely.

Uncovering the exact workings of lexical cohesion requires exploration of the interplay of constraints given by the *system* of word senses and those imposed by *instantiation*, i.e., by the text. Apart from the distance between words in a text, other factors potentially playing a role are *part-of-speech* (Are there any particular parts-of-speech that contract lexical ties significantly more often than others?), *specialized vs. general vocabulary* (Which type of vocabulary participates in the most substantive lexical chains?) and the *register* and *genre* of a text (In a given register/genre, are there any patters of lexical cohesion, e.g., hyponymy-hypernymy, holonymy-meronymy, that occur significantly more often than others? cf. (Teich and Fankhauser, 2004)).

To be able to address such issues has been the motivation for developing the system described in this paper.

2.2. Exploring lexical cohesion

The system we have built for exploring lexical cohesion allows the application of different kinds of constraints on the computation of potential lexical chains that act on WordNet and the concrete text and presents the mark-up result in three different perspectives.

Using the semantic relations represented in WordNet (see Figure 1 for an example), we compute the semantic neighbourhood for every sense-tagged word, and for each synonym set (synset) in the semantic neighborhood we determine the first subsequent word that maps to the synset. In addition, we also take into account *lexical repetition*

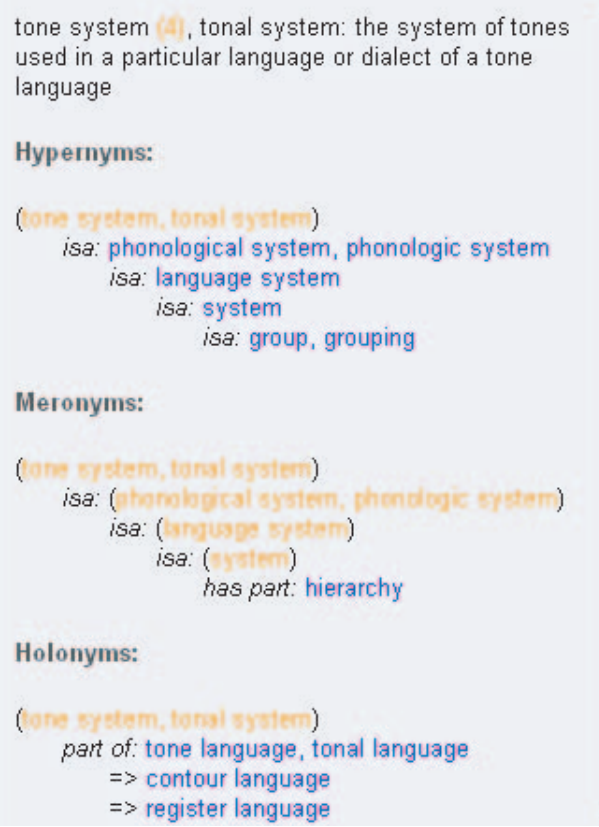


Figure 1: Example from WordNet hierarchy

(same part-of-speech and lemma, but not necessarily same synset), and *proper noun repetition*. For each potential cohesive tie computed on this basis, additional constraints can be applied, both on WordNet and on the actual text. The ones acting on WordNet include the distance of a word from a root in the WordNet hypernymy hierarchy and the branching factor of the underlying semantic relation. The ones acting on the text concern the distance between two words in terms of number of intervening sentences and part-of-speech. These are options that can be set by the user (see Figure 2), who can thus create alternative mark-ups of a text in terms of lexical cohesion.

The resulting mark-up can then be inspected from three perspectives. In the *text view* (Figure 3), each lexical chain is highlighted with an individual color, in such a way that chains starting in succession are close in color. This view can give a quick grasp on the overall topic flow in the text to the extent that it is represented by lexical cohesion.

The *chain view* (Figure 4) presents chains as a table with one row for each sentence, and a column for each chain ordered by the number of words contained in it. This view also reflects the topical organization fairly well by grouping the dominant chains closely. Finally, the *tie view* displays for each word all its (direct) cohesive ties together with their properties (kind, distance, etc.). This view is mainly useful for checking the automatically determined ties in detail.

In addition, all views provide hyperlinks to the WordNet classification for each word in a chain to explore its semantic neighborhood. Moreover, some statistics, such as the number of sentences linking to and linked from a sen-

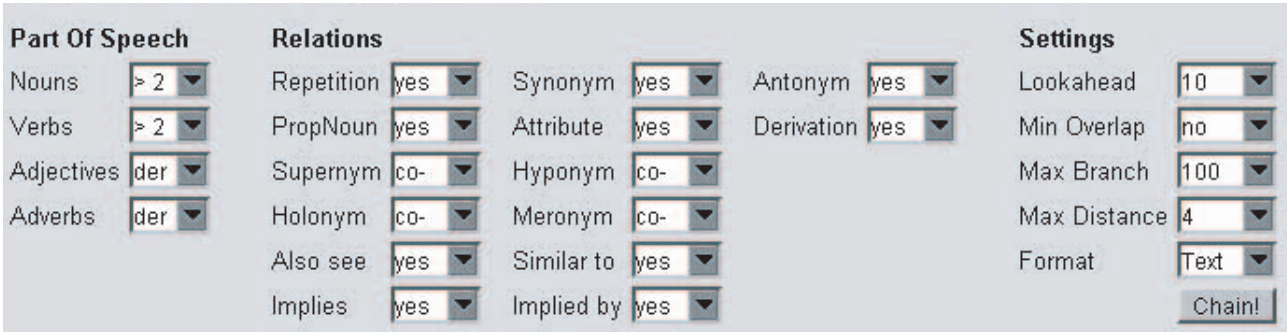


Figure 2: Options for cohesion analysis

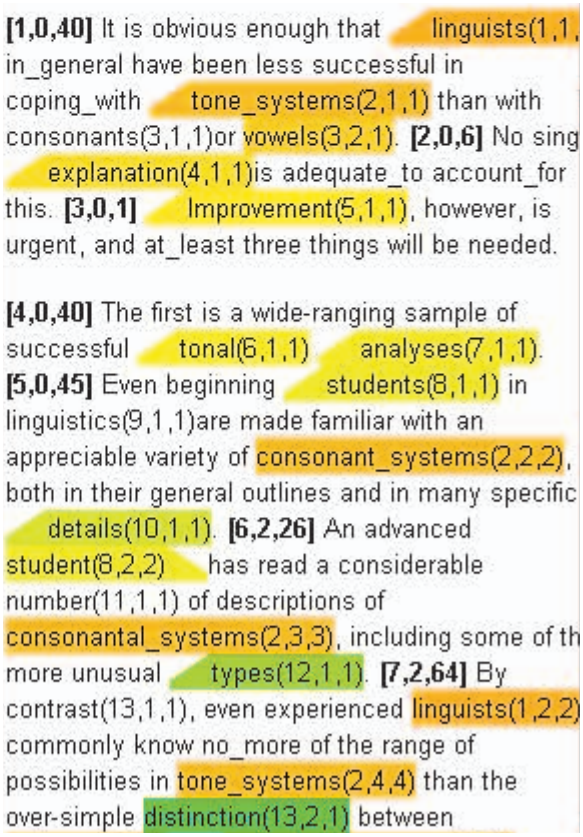


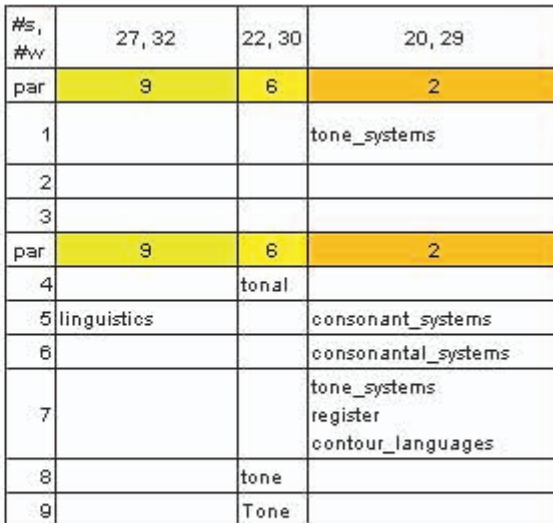
Figure 3: Text view on annotated text

tence, and the relative percentage of ties contributing to a chain are presented. These and some other statistics can then also be exported to a standard statistics package, such as MS Excel or SPSS.

3. XML representation and processing

3.1. Resources

The resources we have used for building the present system are the SEMCOR version of the Brown Corpus and the Princeton WordNet version 1.6. SEMCOR is a multi-layer corpus in that it is marked-up for document structure (headers, paragraphs etc) part-of-speech, lemmata and word senses (according to WordNet). It contains 352 documents with about 2.000 tokens from various registers. In



#s, #w	27, 32	22, 30	20, 29
par	9	6	2
1			tone_systems
2			
3			
par	9	6	2
4		tonal	
5	linguistics		consonant_systems
6			consonantal_systems
7			tone_systems register contour_languages
8		tone	
9		Tone	

Figure 4: Chain view on annotated text

186 documents, nouns, verbs, adjectives and adverbs are sense-tagged with respect to WordNet 1.6. Figure 5 shows an extract represented in XML. Each token is represented as an element `<wf>` with a part-of-speech attribute `pos`; sense-tagged tokens are in addition equipped with attributes `lem` (lemma), `wnsn` (WordNet sense number) and `lexsn` (lexical sense number), which can be used to identify the corresponding synset in WordNet. In addition, `<wf>` elements have an optional attribute `ln` which refers to the line number of the ICAME version of the Brown Corpus (ICAME2, 1999). This attribute is used to include the bibliographic (authors, source, and parts) and typographic (headings and lists) information available from the ICAME version for presentation.

The WordNet version we have used is WordNet 1.6. This version contains about 100.000 synsets, which contain synonyms (120.000 words and 175.000 tokens), and are connected with about 130.000 relations to represent *hyponymy*, various kinds of *meronymy*, *antonymy*, etc. Figure 6 shows the synset for "tone_system" represented in XML, which explicates the rather contrived structure of the original ASCII format. `<synset>` elements have a unique identifier `offset`, and contain `<words>`, `<relations>`, and `<glosses>`. The con-

```
<?xml version="1.0"?>
<contextfile concordance="brown1">
 <context filename="br-j34" paras="yes">
   <p pnum="1">
     <s snum="1">
<wf pos="PRP" ln="0010">It</wf>
<wf pos="VB" lem="be" wnsn="1" lexsn="2:42:03">is</wf>
<wf pos="JJ" lem="obvious" wnsn="1" lexsn="3:00:00">obvious</wf>
<wf pos="RB" lem="enough" wnsn="1" lexsn="4:02:00">enough</wf>
<wf pos="IN">that</wf>
<wf pos="NN" lem="linguist" wnsn="1" lexsn="1:18:01">linguists</wf>
<wf pos="RB" lem="in_general" wnsn="1" lexsn="4:02:00">in_general</wf>
<wf pos="VBP" ot="notag">have</wf>
<wf pos="VB" lem="be" wnsn="1" lexsn="2:42:03">been</wf>
<wf pos="RB" lem="less" wnsn="1" lexsn="4:02:00">less</wf>
<wf pos="JJ" lem="successful" wnsn="1" lexsn="3:00:00">successful</wf>
<wf pos="IN">in</wf>
<wf pos="VB" lem="cope_with" wnsn="1" lexsn="2:41:00">coping_with</wf>
<wf pos="NN" lem="tone_system" wnsn="1" lexsn="1:10:00">tone_systems</wf>
<wf pos="IN">than</wf>
<wf pos="IN" ln="0020">with</wf>
<wf pos="NN" lem="consonant" wnsn="1" lexsn="1:10:00">consonants</wf>
<wf pos="CC">or</wf>
<wf pos="NN" lem="vowel" wnsn="1" lexsn="1:10:00">vowels</wf>
<punc>.</punc>
     </s>
<!-- ... -->
  </p>
 </context>
</contextfile>
```

Figure 5: SEMCOR example

```
<synset offset="n05323594" lex_filenum="10" ss_type="1">
  <words count="2">
    <word lex_id="00">tone_system</word>
    <word lex_id="00">tonal_system</word>
   </words>
  <relations count="002">
    <hypernym ref="n05323217" pos="1" src="0" trg="0"/>
    <part_holonym ref="n05168390" pos="1" src="0" trg="0"/>
  </relations>
  <glosses>
    <gloss>the system of tones used in a particular
    language or dialect of a tone language</gloss>
  </glosses>
</synset>
```

Figure 6: WordNet example

tent of a <word> element together with its attribute lex_id, and the attributes lex_filenum and ss_type of their <synset> match with the combination of the attributes lemma and lexsn of <wf> elements in SEMCOR. Relations refer to other synsets by means of the attribute ref, lexical relations in addition use the attributes src and trg to identify their source word and target word.

The most important design principle for both XML representations has been source fidelity. The XML representation of the Brown Corpus is a straightforward translation from the original SGML format using James Clark's SX Parser. The XML representation of WordNet 1.6 simply explicates the implicit structure of the original ASCII representation rather than translating it to another data model such as RDF or OWL (Melnik and Decker, 2001), which bears the risk of losing source information such as word order or the attributes lex_filenum and lex_id, which are required for matching tokens in the corpus with synsets in WordNet. An arguable disadvantage of this approach is that one can not deploy special purpose corpus query languages such as TIGERSearch (König et al., 2003) or special purpose reasoners such as FaCT (Horrocks, 1999) for the thesaurus. However, as described in the next section, thesaurus-based lexical cohesion analysis requires general purpose querying and programming, which is not adequately covered by special purpose tools.

3.2. Processing

Thesaurus-based lexical cohesion analysis requires the interplay of the corpus with the thesaurus. The basic means for lexical cohesion analysis are so called lexical chains, which consist of words that are related by a lexically cohesive tie. Such lexical chains can be computed in one pass through a text as follows (Doran et al., 2004; Teich and Fankhauser, 2004). Initially the set of lexical chains is empty. Then for each word, the lexical chain which contains a related word is determined. If no such chain exists, the word starts a new chain, otherwise it is included into the existing chain. The resulting lexical chains are disjoint, i.e., each word occurs in at most one chain.

While this basic algorithm is fairly straightforward, the difficulty lies in distinguishing lexically cohesive ties from spurious ties. Both WordNet and the structured Brown Corpus provide some potential factors that can help in ruling out spurious ties:

- Specificity and part-of-speech: A specific noun like "tone_system" is more likely to contract a lexically cohesive tie than a general verb like "be".
- Kind of the semantic relationship: *Repetition* and *synonymy* form stronger ties than *hypernymy* or *meronymy*.
- Strength of the relationship: The direct *hypernym* "phonologic_system" forms a stronger cohesive tie with "tone_system" than the remote *hypernym* "system".
- Distance in text: Words with many intervening words, sentences, or paragraphs are less likely to contract a cohesive tie than close words.

However, these factors can only provide clues. The specificity of a word can only be approximated by the depth of the corresponding synset in the WordNet hypernym hierarchy, likewise the strength of a relationship is only roughly represented by the length and branching factor of the corresponding path in WordNet. Also, the influence of the distance in text on the strength of a cohesive tie depends on the type of text. Therefore, rather than using a fixed set of constraints on these factors, we allow to constrain them individually for the analysis of particular texts.

To this end we proceed in two phases (see Figure 7). In the first phase (P1), all words are equipped with all potential direct cohesive ties by determining for each synset in the semantic neigborhood of the word at hand the next word in the text that belongs to the synset. The semantic neighborhood of a synset is computed as the transitive closure over related synsets guided by regular path expressions. For example, meronym is defined by the regular path expression hypernym* (part_meronym+ | substance_meronym+) | hyponym* member_meronym+, in order to reach direct meronyms as well as part_meronyms and substance_meronyms inherited from hypernyms and member_meronyms inherited from hyponyms.

Figure 8 shows the output of this phase. Each sense-tagged <wf> element is annotated with a unique attribute id and an attribute depth which contains the depth of the corresponding synset in the WordNet hypernym hierarchy. The element relations contains all potential cohesive

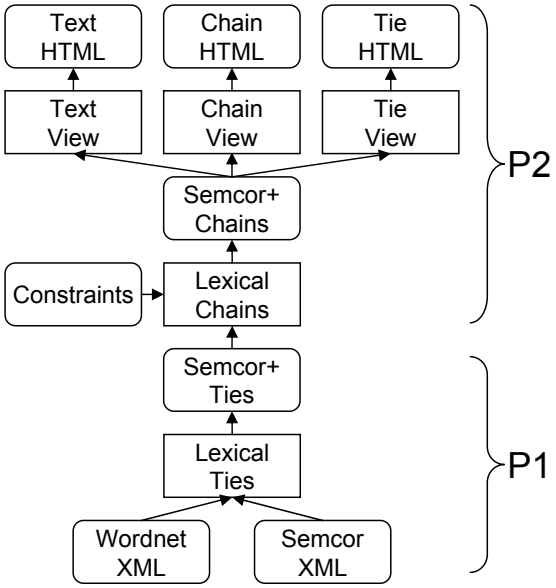


Figure 7: Dataflow Architecture

```
<wf pos="NN" lemma="tone_system" wnsn="1" lexsn="1:10:00"
    id="NC134D5B" depth="4">
  <relations>
    <synonym d="0" b="1" dist="6" id="N6E56CD5B"/>
    <repetition dist="6" b="1" d="0" id="N6E56CD5B"/>
    <holonym d="2" b="2" dist="6" id="N6FFACD5B"/>
    <hypernym d="3" b="84" dist="8" id="N719F8D5B"/>
    <holonym d="1" b="1" dist="11" id="N75A78D5B"/>
    <hypernym d="1" b="1" dist="42" id="N2C2D4D5B"/>
  </relations>
  <content>tone_systems</content>
</wf>
```

Figure 8: Output of Phase 1

ties. Each relation is named by its kind, and has attributes `id` pointing to the next word, `d` and `b` giving the length and branching factor of the underlying path in WordNet, and `dist` giving the number of intervening sentences to the next word.

Both the computation of the semantic neighborhood of synsets as well as the computation of direct cohesive ties have been implemented in XSLT. The implemented XSLT stylesheet consists of about 700 lines of code, of which 250 lines are used for the definition of the automata for the regular path expressions to determine the semantic neighborhood. Altogether the expressive power of XSLT, which essentially is a fully expressive functional programming language, has proven to be quite sufficient. However, for computing the transitive closure without excessive recursion, we have extended the underlying XSLT processor with means to update variables, and for efficient navigation in WordNet we have used the indexing facilities of the underlying XML database system (Infonyte DB (Weitzel et al., 2003; Huck et al., 1999)) via XPath extension functions. With these extensions, the current implementation takes about 5 minutes on an average PC to annotate a text with about 1.500 sense-tagged tokens with all potential cohesive ties.

The second phase (P2) implements the actual chaining algorithm. Also this phase is realized as an XSLT stylesheet, which is parameterized to express simple constraints for ruling out unwanted cohesive ties. The parameters specify for each part-of-speech the minimum depth of potentially cohesive words, the kinds of semantic relationships to be included or explicitly excluded, the maximum length and branching factor of a relationship, and the maximum number of intervening sentences between two constituents of a cohesive tie (see again Figure 2). Using these parameters, the lexical chains are computed in one pass through the text, further annotating the cohesive `<wf>` element with an identifier corresponding to the chain it belongs to. In a second pass the chains are visualized via one of the views introduced in Section 2.2. by transforming the annotated XML representation to HTML. The underlying stylesheet consists of about 1.600 lines of code, half of which is devoted to generating the different views. Because this phase does not require the excessive computation of transitive closures for determining the semantic neighborhood of words, the current implementation takes about 5 to 10 seconds to generate one of the cohesion analysis views for a particular set of constraints.

The user interface for specifying the constraints on lexical cohesion and for inspecting the generated views has been realized on the basis of iReader (Fankhauser and Fitzner, 2003), a configurable browser for XML documents and databases.

4. Summary and discussion

We have presented a system for exploring lexical cohesion that draws on two existing natural language resources, a multi-layer corpus (SEMCOR) and an electronic thesaurus (Princeton WordNet). Lexical cohesion analysis is a prime example of interactivity issues that can only be solved with richly annotated data. It is an application that brings out rather complex requirements on both corpus representation and corpus processing: Not only do we have to deal with the representation of a multi-layer corpus, but also, we have to account for the interplay of constraints imposed by the linguistic system (here: the organization of words in terms of sense relations) and from instantiation, i.e., from the text proper (e.g., distance between words in the text) (cf. Section 2.).

Not surprisingly, all used resources and interim results could be faithfully and adequately represented in XML, which largely facilitated the development, validation, and analysis of results. Perhaps more surprisingly, also the rather complex reasoning and querying required for lexical cohesion analysis could be implemented with only moderate effort using the now stable general purpose XML processing standards XSLT and XPath, for which scalable and robust storage and processing technology that can deal with fairly large data volumes (here altogether about 500 MB) is available. XSLT 1.0 is not perfect. Apart from updatable variables (recall Section 3), also support for user defined XPath functions and a better support for different data types is missing. Both XSLT 2.0 and XQuery 1.0 will include these features, which will allow to improve the factorization of the implementation. The availability of powerful general purpose XML query and processing tools does not necessarily make special purpose corpus query languages for a special purpose XML vocabulary obsolete. However,

such query languages should be seamlessly integrated into general purpose languages, e.g., as extension libraries of XQuery, and not duplicate part of the functionality of XSLT and XQuery.

5. References

Barzilay, Regina and Michael Elhadad, 1997. Using lexical chains for text summarization. In *Proceedings of ISTS 97, ACL*. Madrid, Spain.

Doran, William, Nicola Stokes, Joe Carthy, and John Dunnion, 2004. Comparing lexical chain-based summarisation approaches using an extrinsic evaluation. In P. Soijka, K. Pala, P. Smrz, C. Fellbaum, and P. Vossen (eds.), *Proceedings of the 2nd Global WordNet Conference, January 20-23, 2004*. Brno, Czech republic: Masaryk University, pages 112–117.

Fankhauser, Peter and Christoph Fitzner, 2003. Rich interactive publications with XSLT and little else. In *Proceedings of XML 2003*. Philadelphia.

Fellbaum, Christiane (ed.), 1998. *WordNet: An electronic lexical database*. Cambridge: MIT Press.

Halliday, MAK and Ruqaiya Hasan, 1976. *Cohesion in English*. London: Longman.

Hasan, Ruqaiya, 1984. Coherence and cohesive harmony. In J. Flood (ed.), *Understanding Reading Comprehension*. Delaware: International Reading Association, pages 181–219.

Hoey, Michael, 1991. *Patterns of lexis in text*. Oxford: Oxford University Press.

Horrocks, Ian, 1999. FaCT and iFaCT. In P. Lambrix, A. Borgida, M. Lenzerini, R. Möller, and P. Patel-Schneider (eds.), *Proceedings of the International Workshop on Description Logics (DL'99)*.

Huck, Gerald, Ingo Macherius, and Peter Fankhauser, 1999. PDOM: Lightweight persistency support for the document object model. In *Proceedings of the 1999 OOPSLA Workshop Java and Databases: Persistence Options.*. ACM, SIGPLAN.

ICAME2, 1999. *ICAME Collection of English Language Corpora*. The HIT Centre, University of Bergen, Norway. [CD-ROM], edited by Hofland K. and A. Lindeberg and J. Thunestvedt, 2nd edition.

König, Esther, Wolfgang Lezius, and Holger Voormann, 2003. *TIGERSearch User's Manual*. IMS, University of Stuttgart, Stuttgart.

Landes, Shari, Claudia Leacock, and Randee I. Tengi, 1998. Building semantic concordances. In Christiane Fellbaum (ed.), *WordNet: An electronic lexical database*. Cambridge, MA: MIT Press, pages 199–216.

Melnik, Sergey and Stefan Decker, 2001. WordNet RDF representation. `http://www.semanticweb.org/library/`.

Miller, George A., Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller, 1990. Introduction to WordNet: An on-line lexical database. *Journal of Lexicography*, 3(4):235–244.

Morris, Jane and Graeme Hirst, 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

Teich, Elke and Peter Fankhauser, 2004. WordNet for lexical cohesion analysis. In P. Soijka, K. Pala, P. Smrz, C. Fellbaum, and P. Vossen (eds.), *Proceedings of the 2nd Global WordNet Conference, January 20-23, 2004*. Brno, Czech republic: Masaryk University, pages 326–331.

Teich, Elke, Silvia Hansen, and Peter Fankhauser, 2001. Representing and querying multi-layer corpora. In *Proceedings of IRCS Workshop on Linguistic Databases*. University of Pennsylvania, Philadelphia, USA.

Weitzel, Tim, Thomas Tesch, and Peter Fankhauser, 2003. A scalable approach to processing large XML data volumes. In *Proceedings of Americas Conference on Information Systems 2003 (AMCIS'2003)*. Tampa, Florida.

Challenges in Modelling a Richly Annotated Diachronic Corpus of German

Stefanie Dipper[1], **Lukas Faulstich**[2], **Ulf Leser**[2], **Anke Lüdeling**[1]

[1]Institut für deutsche Sprache und Linguistik
{Stefanie.Dipper,Anke.Luedeling}@rz.hu-berlin.de
[2]Institut für Informatik
{faulstic,leser}@informatik.hu-berlin.de

Humboldt-Universität zu Berlin
Unter den Linden 6, D-10099 Berlin

Abstract

This paper presents the design and architecture of a diachronic corpus of German. We describe the corpus architecture with a focus on the use and restrictions of XML as the data exchange and storage format. In our approach, a relational database will supplement the XML representation to support sophisticated search and presentation facilities. This is a report about ongoing work; the architecture presented here is being developed in a pilot study.

1. Introduction

This paper describes the design and architecture of a diachronic corpus of German with texts from the 9th century (Old High German) to the present (Modern German). This corpus will be built by the large-scale Germany-wide project Deutsch.Diachron.Digital (henceforth DDD).[1]

We describe the corpus architecture of DDD with a focus on the use and restrictions of XML as the data exchange and storage format. We argue that a corpus based on a collection of XML files is not sufficient to support sophisticated search and presentation and that therefore a relational database with an information retrieval extension serves our needs better. We plan to use a graph-based representation for the corpus and to provide powerful import/export methods to support various XML-based formats.

Historical texts are of interst to scholars in many fields (historical linguistics, theoretical linguistics, philology, history, philosophy, ...). However, although many historical texts (manuscrips and early prints) have been digitized in a number of projects (for example, TITUS[2], Bibliotheca Augustana[3], MHDBDB[4]; for an overview, see Kroymann et al., 2004), the historical corpus situation for German is not satisfying: There are no common standards for digitization (this pertains to the question of the best source—manuscript or edition—as well as to the level of diplomaticity and the quality of collation), header information, or annotation on any level. Projects often do not conform to existing standards such as TEI (Sperberg-McQueen and Burnard, 2001) or XCES (Ide et al., 2000). There are no common search interfaces. Many of the digitized texts are not available to a wider audience.

As a reaction to this, DDD aims at creating a generally available, unified resource with common standards and search programs for scholars in the above mentioned fields as well as for interested laypeople. The architecture needs to be highly flexible to cover all the requirements.

The paper is structured as follows. We first present the DDD project and its requirements. We then describe related corpus projects and, finally, give a description of the implementation concept, addressing the general architecture (the data model), import and export methods, and the XML-based representation (the exchange format).

2. Corpus Architecture

The architecture of the DDD corpus has to satisfy different types of requirements: (i) requirements specific to diachronic corpora, (ii) requirements due to the heterogeneity of the corpus, and (iii) requirements due to different types of users. These requirements call for a flexible corpus architecture on the one hand, and for maximal standardization in digitization and annotation on the other hand.

2.1. Requirements for Diachronic Corpora

Our corpus is a historical corpus in that it deals with older texts; moreover, it is a diachronic corpus because it comprises texts from different language periods. Both properties come with requirements that differ from the requirements of corpora consisting of texts from one language period only.

Multi-modality For many historical linguistic research questions, it is necessary to refer to the manuscript facsimiles. Therefore, some parts of the corpus will be aligned by page or line to manuscript facsimiles.

For teaching purposes it is sometimes instructive to hear older texts read out. Hence, certain texts will be aligned to sound files.

[1]The project developed from a Germany-wide initiative (at the moment 15 universities are involved) and is in its beginning phase, with the final funding decision still pending. The architecture presented here is being developed in a pilot study within the Forschungsverbund Linguistik-Bioinformatik, financed by the Senatsverwaltung für Wissenschaft, Forschung und Kultur, Berlin. See `http://korpling.german.hu-berlin.de/ddd/`. Due to previous work of the project partners, DDD can start out with a considerable amount of digitized texts, which are partially annotated, by varying types of information.

[2]`http://titus.uni-frankfurt.de/indexe.htm`

[3]`http://www.fh-augsburg.de/~harsch/augusta.html`

[4]`http://mhdbdb.sbg.ac.at:8000/index.html`

Integration of external resources The corpus will be linked to external resources, like, e.g., the electronically available lexicons for Middle High German[5].

Multi-linguality, alignment The corpus is multi-lingual. First, although there is a continuous development from Old High German to Modern German, there are enough differences between the periods to speak of several languages here. Second, there are many texts, especially in the Old High German period, that contain Latin parts. Some texts are direct (interlinear) translations of Latin, others are interpretations of Latin texts. The interlinear translations are especially interesting for research on word order: any difference in word order between the original Latin text and the German translation points to strong constraints of the Old High German grammar. This means that we need a word-to-word alignment (more precisely: an alignment of n to m words) between Old High German and Latin in these texts, cf. the example in Figure 1, taken from *Tatian* α 2,7 (Sievers, 1961).

Besides text-internal alignments as in the above example, alignments between different texts will be made as well. Examples are alignments between different manuscript versions of the same story (e.g. the various manuscripts recounting the Nibelungenlied), between different editions of the same manuscript, or between a manuscript and its edition.

Another type of example is the alignment of corresponding words of different periods. The purpose of such alignments is to trace the changes a word undergoes in the course of time. For instance, *imbizs* (Old High German) corresponds to *inbizze* (Middle High German), which finally evolved into *Imbiss* in Modern German.

The alignment will be encoded by means of 'hyper lemmas'. A hyper lemma is a set comprising the (normalized) lemmas of different periods that correspond to each other. The lemmas then are linked to the actual words occuring in the text.[6]

Structural annotation The layout of old texts may bear important linguistic information. For instance, words are often split in two parts by line breaks, and it is an open research question how often the location of such breaks coincides with syllable or morpheme boundaries.

Therefore, the texts in the corpus will be structurally annotated, both graphically (marking lines, pages, etc.) and logically (verses, sentences, etc.). Note that this leads to conflicting annotation hierarchies.

Smallest reference unit The token=graphemic word-based encoding of modern corpora is not directly applicable to historical texts.

Historical texts make heavy use of abbreviations, e.g. the character sequence *er* is often replaced by a ˜, as in *d*˜ (= *der* 'the'). Such abbreviations will be spelt out in the normalized, unabbreviated word form. Ideally the normalization allows for a reconstruction of the abbreviation sign ˜ and the corresponding, spelt-out characters.

h	a i	z	a	n		(Alemanic, diplomatic)
h	ê	z	a	n		(Middle Franconian, diplomatic)
h	e i	z	a	n		(Normalization)

Figure 2: Character alignment of dialectal and normalized form

A further example is provided by orthographic variations as they occur in different dialects (which may indicate differences in phonetics and/or phonology). For instance, *ai* in the Alemanic dialect usually corresponds to *ê* in the Middle Franconian dialect. The normalized form (which abstracts away from dialectal variation) uses *ei* to encode this sound. The characters corresponding to each other should be aligned, as sketched in Figure 2.

To model these requirements appropriately, the smallest units of reference in the corpus representation must be single characters. Further possible applications of such a character-based annotation include the encoding of initials and ligatures (paleography), linebreaks, and alliteration.

This has the additional advantage that morpheme boundaries can be annotated and the differences between graphemic and lexical word can be easily marked.

Meta-annotation The annotation of historical texts is at best semi-automatic. This means that often several annotators work on the same text. It is useful to keep record of the annotation task by means of meta-annotations. Meta-annotations refer to other annotations and encode information such as the annotator of the referenced annotation, the date of annotating, or the tool applied in the annotation task. Comments can be added to any annotation unit in the same way.

2.2. Requirements Due to Corpus Heterogeneity

The DDD corpus is heterogeneous with regard to the depth of annotation and its composition.

Annotation depth Depending on the research question, the requirements with regard to annotation of a corpus differ. While for many linguistic questions, rich annotation is desirable, there are philological and lexicographic questions where corpus size may be more important than annotation depth. To satisfy both requirements as best as possible, the depth and type of annotation will differ within the DDD corpus. This must be accounted for by the corpus architecture which should allow the user to select homogeneously annotated sub-corpora as a basis for further research.

The corpus will be composed of three subcorpora, which we call the extension corpus, the core corpus, and the presentation corpus (cf. Figure 3). The extension corpus consists of texts that are annotated only with structural information (see below) and header information (based on the standards TEI (Sperberg-McQueen and Burnard, 2001) and XCES[7] (Ide et al., 2000)), encoding bibliographic information. In addition, the core corpus will be annotated

[5] `http://gaer27.uni-trier.de/MWV-online/MWV-online.html`

[6] The question of which lemmas of different periods correspond to each other is a difficult one, involving aspects of semantics (word meaning), morphology, etc. The DDD project aims at defining clear criteria for hyper lemmas.

[7] `http://www.xml-ces.org/`

Et	*non*	*erat*	*illis*	*filius,*	*eo quod*	*esset*	*Elisabeth*	*sterilis*	(Latin)
inti	*ni*	*uuard*	*in*	*sun,*	*bithiu uuanta*	*Elisabeth*	*uuas*	*unberenti*	(OHG)
and	not	was	them	son	because	Elisabeth	was	sterile	(Gloss of OHG)

'and they did not get a son because Elisabeth was infertile'

Figure 1: Word-to-word alignment of a multi-lingual source text (Latin – Old High German), Tatian

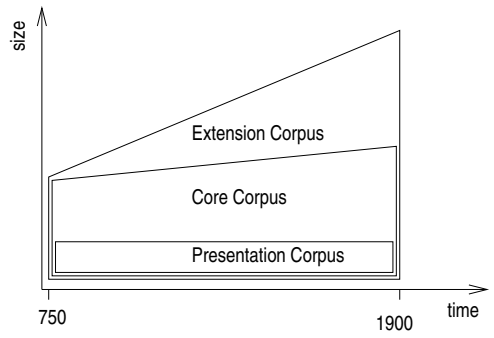


Figure 3: Composition of the planned DDD corpus

with (normalized) lemma information, part-of-speech tags and inflectional morphology.[8] Other annotation levels, e.g. information structure or syntax, can be added to texts from either subcorpus. Some texts—the presentation corpus—will be aligned to manuscript facsimiles or sound files. All types of annotation will be based on existing standards, if available (e.g., STTS (Schiller et al., 1999) for part-of-speech tagging, TIGER (Brants et al., 2002) for syntax annotation), which, of course, will have to be adapted to the special requirements of historical and diachronic data.

DDD intergrates a lot of already digitized material[9], which has to be brought to a common quality standard and annotated.

Corpus composition The corpus composition differs for the different language periods. The older language periods (Old High German, Old Saxon) can be digitized almost completely, while in the newer periods the corpus needs to be balanced with respect to a number of parameters like region, genre, dialect, etc.

Additional texts will be added in the course of the project. Hence, the corpus architecture must allow the addition of new texts. At the same time, it must be possible to identify reference corpora for each period.

2.3. User Requirements

The DDD corpus addresses scholars in many fields, e.g., linguists, lexicographers, philologists, historians. The needs of these user groups differ with respect to (i) search facilities, (ii) the presentation of the corpus, and (iii) export options.

[8]The annotation of historical texts heavily depends on manual work. In this paper, we do not address the issue of how the information will be annotated.

[9]This material was digitized in different projects at partner universities and, among others, includes (parts of) the TITUS corpus, the Bonner Mittelhochdeutsch corpus, and the Digital Middle High German Text Archive.

Search facilities Lexicographers search for the use and collocations of a word or word form. In a diachronic corpus, they can also look at meaning change or form change. For instance, at about 900 AD the word *imbizs* meant 'delicious meal', whereas the corresponding present-day form *Imbiss* means 'snack'. For lexicographic purposes, we therefore need full-text searches and collocation extraction.

In contrast, linguists often search for annotated information such as morphology, part of speech, syntax, e.g. to investigate the change of word order in German. This usually involves complex, cross-level queries.

We are convinced that the requirements of the prospective user groups cannot be satisfied by providing a single search interface. Therefore, we envisage the provision of at least two levels of searching: one simple full-text search including only the digitized text, and another interface providing access to the full annotation.

Corpus presentation Similarly, the ideal visual presentation of the corpus depend on the type of user. The texts will be represented with or without annotation or with selected annotation types only. In addition to a Web interface, presentation with external viewers (e.g., PDF) should be supported.

Export options The corpus (and search results) will be represented by a primary XML exchange format. This allows the user to further process and manipulate the data by external tools.

An XML format will also be used as the exchange format for annotated texts within the project. However, external editors and annotation tools may require or produce documents in different formats.

Standards compliance Finally, existing linguistic and IT standards should be applied wherever possible to facilitate access to the corpus (including future access), to ease the application of external tools, to make data reuse possible, and to allow for comparison and exchange with other corpora.

3. Related Corpus Projects

DDD is inspired by research on historical corpora, multi-lingual corpora, and multi-modal corpora. We first give a broad overview before going into more technical detail.

Historical corpora DDD cannot be modelled directly after existing historical corpora of other languages because most of them are smaller and made with a specific purpose in mind (literary goals or linguistic goals, but not both). For example, the diachronic part of the Helsinki Corpus (roughly 1 million words), which was originally collected for research on variation and language change

(Rissanen et al., 1993), is now annotated linguistically with part-of-speech information and syntax[10]. Other historical corpora that have more annotation levels encompass only a certain language period (like the Lancaster Newsbook corpus, which contains 17th century newsbooks[11]) or are even more specific (compare the corpus of the Nibelungenlied[12]). However, even though the overall corpus architecture cannot be directly copied, existing historical corpora are the basis for many decisions concerning the different annotation layers.

Multi-lingual corpora DDD shares many of the problems of multi-lingual corpora, in that we need alignment between texts and also within the same text. As mentioned above, some of the texts are direct interlinear translations from, e.g., Latin to Old High German. Here we need word-to-word alignment within the same text.

In many cases we have different manuscripts of the same text (as the manuscripts A–C of the Nibelungenlied)—these need to be aligned as well. The problem goes further: in order to track lexical change, all of the texts in the corpus need a common 'normalized' lemma layer (the 'hyper lemma' annotation).

Multi-modal corpora DDD can be modelled on multi-modal corpora, which have the task of connecting different representations of the same utterance—for example, a spoken sentence with its transliteration and the gestures that were made while speaking—with each other and with annotation layers.[13] Each representation and annotation layer is represented in a different file, resulting in a multi-layer stand-off annotation. Roughly spoken, all files are connected via reference to a common base line (or time line for speech data).

The data model for DDD (which is presented in detail in Sec. 4.2.) is inspired by this architecture, but generalizes it by permitting multiple time lines for the same text: a diplomatic rendering of the original text serves as a base line for the graphemic view of the text (volumes, pages, lines, graphemic words), whereas the logical view of the text (chapters, sections, paragraphs, sentences, lexical words) refers to the time line of a normalized version of the text. Both time lines are aligned by annotations that link graphemic with lexical words (cf. Figure 6). Each further representation or annotation layer (normalization, part-of-speech tags, structural inormation, etc.) can refer to either one of these base lines or to annotations within other layers. In XML, each annotation layer can be stored in a separate file which uses XPointer URLs to refer to a base line. In this way, we can deal with conflicting hierarchies, different modes of representation (text, graphics, speech) as well as with the fact that not all texts in the corpus are annotated in the same depth.

[10] `http://www.ling.upenn.edu/mideng/`

[11] `http://www.ling.lancs.ac.uk/newsbooks/`

[12] `http://www.blb-karlsruhe.de/blb/blbhtml/nib/uebersicht.html`

[13] For examples of multi-modal corpora, see the SmartKom corpus (`http://www.smartkom.org/`) or the GeM corpus (`http://www.fb10.uni-bremen.de/anglistik/langpro/projects/gem/newframe.html`); see also below.

4. Implementation Concept

To repeat the above requirements: the DDD corpus is multilingual, multi-modal, and has to support different and varying annotation levels. The smallest unit of annotation is the character and DDD has to support conflicting hierarchies. The corpus must be searchable with intuitive and, at the same time, powerful search tools that can search on all annotated levels.

We first present the overall architecture of DDD in Sec. 4.1., where it is specified that the DDD corpus is stored in a central database. The data model that will serve as the basis for structuring the corpus within the database is introduced in Sec. 4.2. We then focus on how texts can be exported from / imported into the database (Sec. 4.3.). In Sec. 4.4., we present the XML formats that will be used for exchange with other project partners and for publication of the corpus.

4.1. System Architecture

We propose a Web-based system architecture where users search or browse the DDD Web server via standard Web browsers (cf. Figure 4). For digitization and annotation, the project partners can download XML files using a Web browser, apply external tools to these files, and upload the modified XML files again to the DDD Web server. External tools may also communicate directly with Web services offered by the DDD Web server. For instance, a lemmatization tool might access a lexicon at the DDD server via a Web service.

The Web server routes user requests to a module of the application logic tier, which in turn communicates with the relational database system storing the corpus. The application logic comprises several search interfaces, import and export converters, and administrative modules for access control, diagnosis, etc.

The corpus itself is stored in a relational database system containing a full-text retrieval component. Compared to the storage of a corpus in a flat file, this yields several advantages:

- Sophisticated search facilities on text, header data, and annotations: full-text search can be combined with search criteria on header data; complex conditions on annotations and information referenced by annotations can be formulated; etc.
- Extensive support for statistical analysis in modern SQL: SQL:1999 and SQL:2003 (Türker, 2003) incorporate several statistical operators developed for data warehousing applications, which can be used for analyzing large sets of annotations.
- More natural representation of non-hierarchical data (cf. Sec. 4.2.): in XML, non-hierarchical relationships must be expressed using ID-references, which have to be handled by special means.
- Independence from document formats: in Sec. 4.3. we show that various import and export formats can be

supported without imposing the restrictions of a particular format to the database.

- Multi-user capabilities: relational database systems can support a large number of concurrent users. In particular, they are able to successfully handle conflicts arising from concurrent write operations.
- Robustness, scalability, maturity: modern database systems provide excellent means for recovery and backup. They can be easily extended to cope with increasing demands for storage and throughput.
- Longevity: by using industry standards such as SQL, the chance to ensure long-term operation of the DDD corpus is increased.

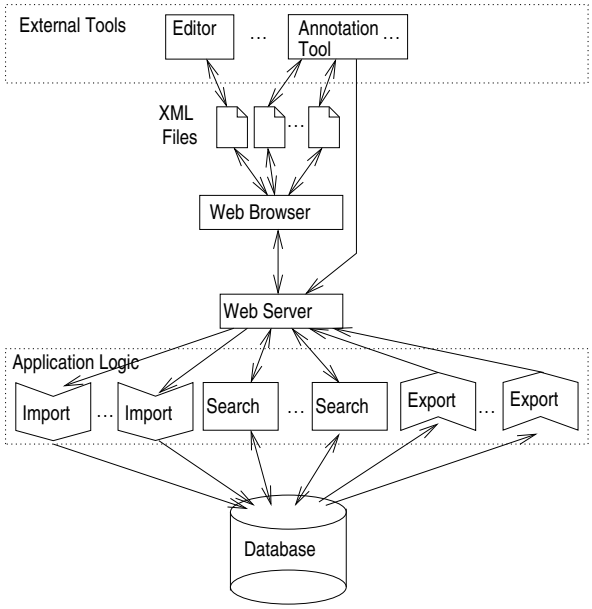


Figure 4: System Architecture of DDD

4.2. Data Model

Although the DDD corpus will be stored in a relational database, we restrict ourselves to a specialized data model for annotated texts rather than using the relational data model in its full generality.

There are two popular data models for multi-modal corpora: the annotation graph (AG) model (Bird and Liberman, 2001) and ordered directed acyclic graphs (ODAGs), such as the NITE object model (NOM) (Carletta et al., 2003). Annotation graphs model annotations as arcs that connect time points on the time axis of a signal. Annotation graphs can be stored easily in relational databases and searched efficiently by translating queries into SQL. However, the AG model has some shortcomings. For instance, parent-child relationships cannot be represented in AGs without extending the data model with special child/parent arcs (Teich et al., 2001). Without this extension, the dominance relation between a non-branching node and its only child is not encoded. Meta-annotations or alignments cannot be represented directly but need to be expressed by introducing equivalence classes (i.e., annotations are linked by assigning them identical attribute values).

The ODAG-based NOM does not share these limitations. Annotations are represented by nodes. Annotation values are stored in form of node attributes. The domination relation between nodes is modeled explicitly by parent-child relationships. Each node may refer to a span of the underlying text. In this case, the child nodes must refer to non-overlapping text spans contained in the span of their parent node. The order of child nodes must correspond to the order of their spans in the underlying text.

We have two requirements which go beyond the NITE model: (i) we want to represent the whole corpus within the same data structure to enable cross-references between texts, and (ii) we want to permit complex annotation values, which cannot be represented as node attributes, a need that has been recognized also in (Brugman and Wittenburg, 2001).

For DDD, we propose a data model based on ODAGs that is presented in Figure 5. Two prominent features of our model are:

- A collection of texts is associated with each ODAG. This collection comprises the source texts of the corpus and, in addition, notes, comments, and other free-text annotations. Every node may reference a span in some text associated with the ODAG. This generalizes the NOM where all texts (“signals”) are synchronized and references cannot point to a specific text. As in NOM, the span referenced by a node must be contained in the spans referenced by its ancestor nodes. Moreover, the spans referenced by the children of a node must be disjoint and the textual order of the spans must be consistent with the order of the child nodes.
- Annotations with complex values are seen as relationships between ODAG nodes. A complex annotation is a node with a child that marks a region of the source text, and one or more children representing (facets of) the annotation value. Alignments are annotation nodes having several children referring to the source text(s).

It is straightforward to use this data model as a generic database schema. However, this approach lacks efficiency. We plan to investigate the efficiency of object-relational features offered by SQL:1999 and SQL:2003. These features can be used to organize nodes by name into a hierarchy of tables that store each node together with its attributes. Parent-child relationships have to be stored in bridge tables since – differing from XML document trees – they are of cardinality $m : n$.

This approach has the advantage that we can represent the whole corpus as a single ODAG under a single root node. Each annotated text is represented by a subgraph that is rooted in a child of the corpus root node. In Figure 6, the structure of a prototype ODAG for the DDD corpus is sketched.

4.3. Import/Export Methods

For presentation, exchange, and support of existing tools, an XML representation of the ODAG is necessary.

An XML document can be modeled as an ordered tree, which is a special case of an ODAG. However, the ODAG

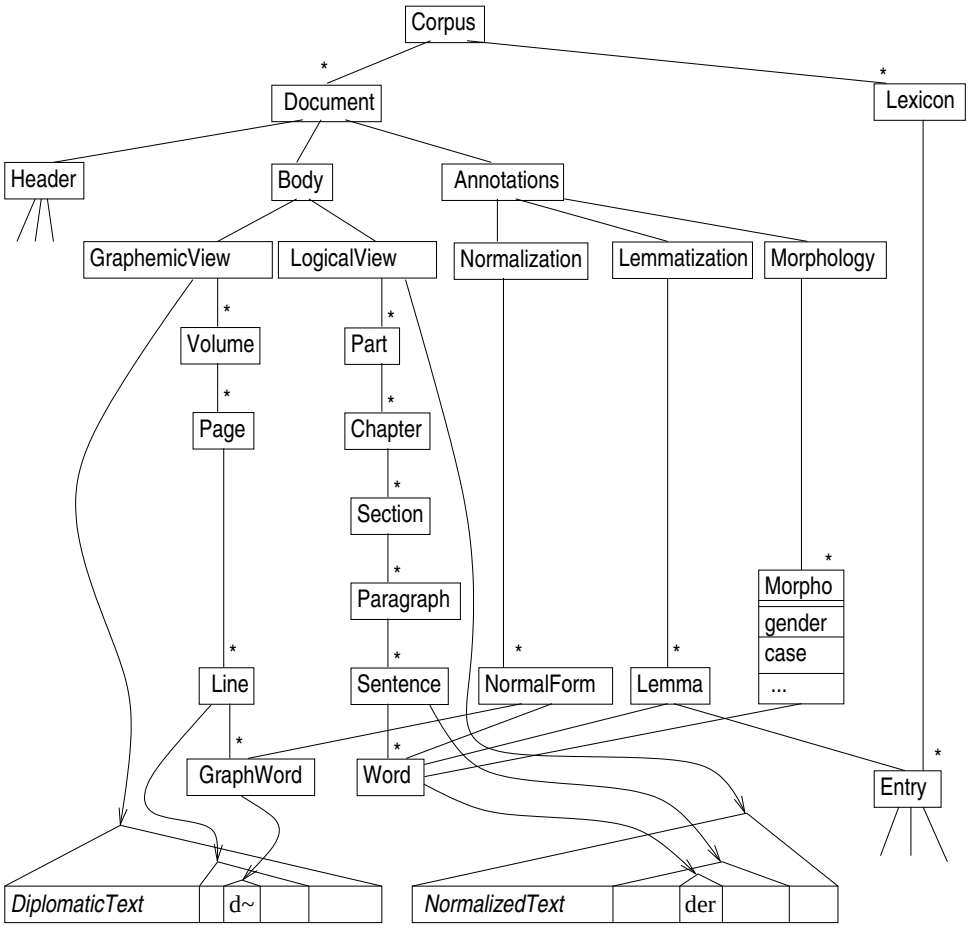


Figure 6: Prototype of DDD corpus schema

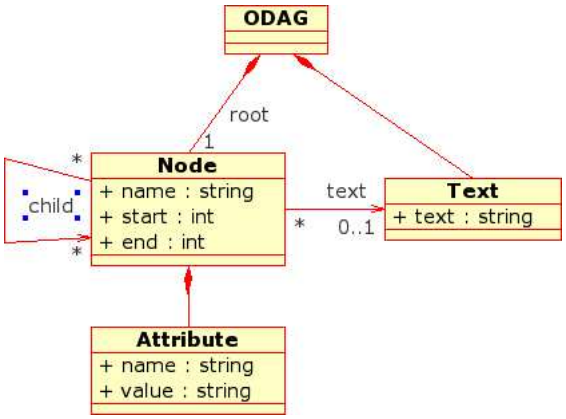


Figure 5: UML model of the DDD datamodel

stored in the database is in general not a tree. Hence, to export an XML document from the database, a tree has to be generated on the basis of the stored ODAG. Moreover, we may want to include only certain annotation layers in the XML document or would like to use names for document elements and attributes that differ from the names used in the database. When importing XML documents into the database, an inverse transformation has to be performed.

Hence, a powerful and flexible method for transforming a source ODAG into a target ODAG is needed to support the import and export of XML documents into/from the database. Requirements for this transformation method are:

Projection: Only certain annotation layers may be needed in the target ODAG. For instance, one might want to present only the physical structure of a document on a Web page.

Selection: The source ODAG may be restricted to a certain part of a text. For instance, a Web page might present only a single chapter of a text. Conversly, an XML document created by an external tool may contain tool-internal data that is to be excluded from import into the database.

Folding and Rearrangements: The target ODAG may contain different transformations of the same part of the source ODAG. For instance, we may want to generate an HTML document that presents each text line as a table whose rows correspond to different annotation layers. This requires a different transformation of the same line for each annotation layer.

Derived Attributes and Elements: Complex annotation values may be derived from separate simple attributes. For instance, the complex STTS part-of-speech value *VAINF* ('verb, auxiliary, infinitive' (Schiller et al., 1999)), as used, e.g., in the TIGER corpus (Brants et al., 2002), may be derived from the

atomic part-of-speech value *verb* and the specifications *type=auxiliary* and *inflection=infinite*.

Interchange between Element Content and Attributes: Values of annotations that are stored as attributes within the database may be represented as element content in an XML document and vice versa.

Context-Sensitivity: Nodes may be transformed in different ways, depending on the context. For instance, a word node may be copied verbatim in the context of a sentence, while in the context of a grammatical annotation, the word node is transformed into an XPointer reference.

Addressing: IDs or XPointer URLs identifying document elements or addressing text regions need to be generated (on export) and resolved (on import) to support stand-off annotation formats.

Encoding/Decoding of Conflicting Hierarchies: Although our primary exchange format avoids the problem of conflicting hierarchies by using separate annotation files, we need to support other formats to represent several hierarchies within the same document. Several solutions for this problem have been proposed by the TEI. From these, we plan to support at least milestones, fragmentation, and virtual joins:

Milestones: creating milestones means replacing subsequent elements (e.g., pages) by empty milestone elements (e.g., page breaks). Decoding of milestones means reconstructing annotations spanning the regions separated by the milestones.

Fragmentation: annotations with a lower priority must be split into several parts at the borders of higher-priority annotations. On import, these annotation fragments must be merged again into single annotations.

Virtual Joins: virtual joins are based on fragmentation but have additional IDREF attributes linking the fragments.

4.3.1. Generic Mapping

To apply existing transformation technology for XML, we use a generic mapping between ODAGs and XML document trees that replicates all shared nodes in the ODAG. Node IDs are generated and stored in an extra noderef attribute to keep track of the original nodes. Text referenced by a node is inserted as PCDATA, interleaved with the document elements representing the children of the node. In addition, the span of the referenced text is described by the attributes text (URI to text), start, end (span).

To facilitate the creation of XML documents for import into the database, redundant content need not to be reproduced. Document elements sharing the same noderef attribute are unified into a shared ODAG node. Empty document elements with a noderef attribute are treated as node references. However, all non-empty elements referring to the same node must have the same content. The unmarked text of the XML document is extracted, concatenated, and an appropriate reference to a span of this text is added to each node unless the node refers explicitly to a text span using the text, start, end attributes. Nodes with such explicit span references may omit the referenced text content, in order to reduce redundancy.

4.3.2. XML Transformation

The XML document resulting from the generic ODAG-to-XML mapping can be transformed in various target formats using general purpose transformation methods like XSLT[14] or STX (Streaming Transformations for XML[15]). XSLT is quite expressive and satisfies most of our requirements in a natural way.

The encoding of conflicting hierarchies by XSLT is not straight-forward, but can be implemented using the timing attributes to select and clip elements of a subordinate hierarchy (see Figure 7). However, this is quite inefficient.

4.3.3. Need for a High-Level Transformation Language

Both methods for encoding conflicting hierarchies result in quite complex and verbose stylesheets that are hard to write manually. This problem could be solved by developing a more high-level transformation language.

Moreover, the ODAG stored in the database can become arbitrarily large. To make the XSLT-based approach scalable, only a subgraph of the database ODAG containing the information to be included in the target document should be exported. An XPath expression could be used to select a node set. The selected subgraph would then be formed by all nodes reachable from this node together with a new synthetic root node. Further specification options might be useful to control the replication of shared nodes depending on the path used to reach them.

Hence it would make sense to define a new transformation language that is better suited to our requirements. This language might be implemented either by generating XSLT stylesheets with additional parameters controlling the generic transformation or by a specialized transformation mechanism of its own.

4.4. Exchange Format

We distinguish two XML-based exchange formats. The first one will be used within the project as the exchange format for annotated texts. This format is the result of the generic mapping described in Sec. 4.3.1., which maps ODAGs stored in the database to XML document trees. This redundant representation separates conflicting hierarchies and the various annotation layers. It uses node and span references to keep track of shared nodes and the alignment of annotations with the underlying texts. For import, redundant content may be omitted.

The second format represents the 'external' exchange format. This format will serve as the official exchange format, wich is made available to the research community. It will be XCES-compliant—which means that, with the current version of XCES[16], not all encodings of the DDD corpus can be represented adequately.

[14] `http://www.w3.org/Style/XSL/`
[15] `http://stx.sourceforge.net/`
[16] `http://www.xml-ces.org/`

5. Conclusion

In this paper, we presented the architecture for a large-scale, diachronic, multi-modal corpus for German. We first sketched the diverse requirements for digitization and annotation that result from the type of data, the different user groups, and their research questions.

In a pilot study, we developed a flexible corpus architecture to answer these requirements. The corpus will be represented by an ODAG and stored in a relational database. An XML-based representation will be derived from the ODAG representation, which serves as the exchange format within the project. In addition, an XCES-compliant XML representation will be made available for research purposes.

In our architecture, the smallest units of reference are characters. There are two time lines: first, the diplomatic text, focusing on physical properties of the source text; second, the normalized text, focusing on logical properties. The corresponding annotations often result in conflicting hierarchies. To find a suitable representation and efficient methods of manipulation for these hierarchies will be a major point in our future work.

6. References

Bird, Steven and Mark Liberman, 2001. A formal framework for linguistic annotation. *Speech Communication*, 33(1,2):23–60. `http://arxiv.org/abs/cs/0010033`.

Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith, 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories, Sozopol*.

Brugman, Hennie and Peter Wittenburg, 2001. The application of annotation models for the construction of databases and tools: Overview and analysis of MPI work since 1994. In *IRCS Workshop on Linguistic Databases*. `http://www.ldc.upenn.edu/annotation/database/papers/Brugman_Wittenburg/20.2.brugman.pdf`.

Carletta, Jean, Jonathan Kilgour, Timothy O'Donnell, Stefan Evert, and Holger Voormann, 2003. 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)*.

Ide, Nancy, Patrice Bonhomme, and Laurent Romary, 2000. XCES: An XML-based standard for linguistic corpora. In *Proceedings of the Second Language Resources and Evaluation Conference (LREC)*.

Kroymann, Emil, Sebastian Thiebes, Anke Lüdeling, and Ulf Leser, 2004. Übersicht über diachrone Korpora. Technical report, Institut für Informatik, Humboldt-Universität zu Berlin. `www.linguistik.hu-berlin.de/ddd/publikation/HistorischeKorpora.pdf`.

Rissanen, Matti, Merja Kytö, and Minna Palander-Collin (eds.), 1993. *Early English in the Computer Age*. Mouton de Gruyter.

Schiller, Anne, Simone Teufel, Christine Stöckert, and Christine Thielen, 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS. Kleines und großes Tagset. Universitäten Stuttgart and Tübingen, `http://www.ims.uni-stuttgart.de/projekte/corplex/TagSets/stts-1999.pdf`.

Sievers, Eduard (ed.), 1961. *Tatian. Lateinisch und altdeutsch mit ausführlichem Glossar*. Paderborn: Schöningh, 2nd edition.

Sperberg-McQueen, C. M. and Lou Burnard (eds.), 2001. *TEI P4: The XML Version of the TEI Guidelines*, chapter 31: Multiple Hierarchies. Text Encoding Initiative. `http://www.tei-c.org.uk/P4X/NH.html`.

Teich, Elke, Silvia Hansen, and Peter Fankhauser, 2001. Representing and querying multi-layer corpora. In *Proceedings of the IRCS Workshop on Linguistic Databases*. University of Pennsylvania, Philadelphia.

Türker, Can, 2003. *SQL:1999 & SQL:2003 - Objektrelationales SQL, SQLJ & SQL/XML*. Heidelberg: dpunkt.verlag.

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <!-- ... -->
  <xsl:template match="line">
    <xsl:variable name="clipStart" select="number(@start)"/>
    <xsl:variable name="clipEnd" select="number(@end)"/>
    <line no="{@no}">
      <xsl:apply-templates select="ancestor::document/structure/logical/part/verse[
(number(@start) &lt; $clipEnd) and (number(@end) &gt; $clipStart)]">
        <xsl:with-param name="text" select="$text"/>
        <xsl:with-param name="clipStart" select="$clipStart"/>
        <xsl:with-param name="clipEnd" select="$clipEnd"/>
      </xsl:apply-templates>
    </line>
  </xsl:template>

  <xsl:template match="verse">
    <xsl:variable name="start">
      <xsl:choose>
        <xsl:when test="number(@start) &lt; $clipStart">
          <xsl:value-of select="$clipStart"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select="number(@start)"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <xsl:variable name="end">
      <xsl:choose>
        <xsl:when test="number(@end) &gt; $clipEnd">
          <xsl:value-of select="$clipEnd"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select="number(@end)"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <verse no="{@no}">
      <xsl:value-of select="substring($text,$start + 1, $end - $start)"/>
    </verse>
  </xsl:template>
</xsl:stylesheet>
```

Figure 7: Detail of an example XSLT stylesheet that implements the fragmentation of verses within lines, using the timing attributes start and end. In each line element all overlapping verse elements are included and clipped at the borders $clipStart and $clipEnd of the line element. The text of the (clipped) verse element is computed as a substring of the unmarked text stored in $text.

Annotating Discontinuous Structures in XML: the Multiword Case

Emanuele Pianta and Luisa Bentivogli

ITC-irst
Via Sommarive 18, 38050 Povo (Trento) - Italy
{pianta,bentivo}@itc.it

Abstract

In this paper, we address the issue of how to annotate discontinuous elements in XML. We will take discontinuous multiwords as a case study to investigate different annotation possibilities, in the framework of the linguistic annotation of the MEANING Italian Corpus.

1. Introduction

The basic data structures of XML are trees. This makes XML very suitable for linguistic annotation, as trees are a very common formalism used for various linguistic representation levels. Syntactic trees are the most clear example of such linguistic representations. Trees can also be used to represent text divisions (sentences, paragraphs, sections, chapters), the structure of the content (e.g. the RST structure, see Mann & Thomson, 1987) or the graphical layout of the text (see Bateman et al., 2002).

Unfortunately there are at least two tasks that challenge the expressiveness of XML as a formalism for linguistic annotation based on trees, which by definition, unlike graphs, don't allow branches to overlap. The first problematic task is *including annotations for multiple or alternative linguistic representations* within the same XML document. The problem is that an XML document can contain only one tree structure, whereas different representation levels can require distinct, partially overlapping trees. For instance a content unit can include the first paragraph of a text and only half of the second one. This means that the branches that encompass the first content unit will cross the branches that encompass the first and second paragraphs. Another example is given by poetry. If we want to represent within the same XML document both the structure of the poem as a sequence of lines and the division in sentences we quickly run in problems, because a line can span over two parts of sentence. The problem is even more acute if we want to include alternative representations for the same linguistic level in the same XML document. In this case the probability that alternative representations lead to overlapping tree branches is even higher. Thus, if the various linguistic representation levels do not fit in one tree, it will be very difficult or impossible to keep different levels of linguistic annotation within the same XML document.

A second group of phenomena which may be difficult to represent through a tree-based formalism such as XML, are *discontinuous units* or *long distance dependences*. Examples of discontinuous units are non-contiguous multiwords, or -in German- separable verbs, whereas long distance dependences are exemplified by the coupling of a pronoun with its textual antecedent(s).

Apparently in this second group of phenomena we are dealing with only one representation level, so we would not expect to incur the problems caused by the necessity to include distinct representation levels. However, on closer inspection, it turns out that in all non trivial annotation tasks, more than one linguistic level is involved in the representation. Even if our aim is annotating a text at one level such as the syntax or the pronoun antecedents, in most cases we also need to represent within the same XML document at least one other linguistic level which is the division of the text in an ordered sequence of graphical words. Given the necessity to include in any linguistic annotation also information about the basic sequence of graphical words, the representation of any linguistic relation involving two non continuous graphical words is problematic for a tree-based formalism such as XML.

This happens even at the most basic linguistic levels, such as lexical annotation. The sequence of graphical words in a text can be represented with a flat tree in which each leaf corresponds to a word. However, if we want to represent in the same document the fact that two non-contiguous graphical words belong to the same lexical unit, then the necessity arises to use overlapping tree branches. Thus, also this second series of representation difficulties are explained by the expressive restrictions of XML as a tree-based formalism, that is a formalism which does not allow for a natural representation of multiple overlapping hierarchies.

2. Related work

The class of problems we are dealing with has been addressed in various ways in the literature on text annotation. One clarifying formulation of the problem describes it as the difficulty of annotating both the logical and the layout (o physical) structure of the same text. These two (tree-)structures may differ in various ways, which can be described in terms of node duplication, removal/addition, reordering, break-out (Murata, 1995).

Note that in principle the annotation problems we have presented so far could at least partly be solved by resorting to SGML, where the CONCUR feature allows for specifying multiple DTDs and associated tagging on a single document instance (Sperberg-McQueen & Huitfeldt, 1999). Unfortunately the CONCUR feature is not available in XML (Clark, 1997). Also, CONCUR is an optional feature of SGML and is not supported by all SGML processors (Sperberg-McQueen & Burnard, 2001). Finally, if a solution to the problem at stake can be found within the XML formalism, this should be preferred because of the expectation that XML documents are easier to be processed, and that more and more XML-aware tools are made available to the text annotation community in the near future.

Actually, a number of approaches have been proposed to allow for multiple overlapping annotations within XML. Sperberg-McQueen & Burnard (2001) provide a detailed description of the characteristics, the advantages and disadvantages of such approaches. Let us mention here what we think are the most important approaches:

- *Multiple encoding* of the same information. This is straightforward but redundant and bears the risk of introducing non-alignments between different but interrelated annotations, when one annotation is updated and the other not.
- Use of *milestone elements*, that is empty elements, marking the beginnings and endings of spans of text, for instance.: <start-span id='w1'/> … <end-span idref='w1'/>. This has the disadvantage that the structure of the "ghost" annotation based on milestones needs to be reconstructed with ad hoc procedures.
- *Stand-off annotation*, that is the annotation of a text is kept separate from the text itself; special pointers are used in the annotation to refer to the specific text elements which are the object of the annotation. This comes in two variants, as the annotation can be kept in a separate section of the same document, or in a separate file. See for instance the annotation format used in GATE (Cunningham et al., 2002).

Among these three approaches, in this paper we will prefer the stand-off approach, as we think that this guarantees the best compromise between advantages (elegance and clearness of the representation, conceptual simplicity of the processing) and disadvantages (physical discontinuity between the text and the annotation, complexity of the pointer processing).

As a final introductory remark, let us underline that none of these approaches to overlapping representations comes without a cost or some contraindication. As the TEI Guidelines put it, "non-nesting information poses fundamental problems for any encoding scheme, and it must be stated at the outset that no solution has yet been suggested which combines all the desirable attributes of formal simplicity, capacity to represent all occurring or imaginable kinds of structures, suitability for formal or mechanical validation, and clear identity with the notations needed for simpler cases" (Sperberg-McQueen & Burnard, 2001).

In the rest of this paper we will consider in more detail one of the cases of overlapping annotation, that have been mentioned above, that is the annotation of discontinuous multiwords. We will investigate different annotation possibilities and present the pros and cons of each of them. From such an analysis we will see that also the lexical representation level, apparently the simplest linguistic representation level, can be problematic, and requires principled solutions. More specifically we will see that lexical representation involves three more fine-grained levels, that are *tokens*, *potential words*, and *multiword expressions*.

3. The multiword case study

The term multiword is used to denotate various kinds of lexical units. Within this paper we will use it to refer to both idioms and restricted collocations. We will exemplify the problems that arise when annotating discontinuous multiwords by considering the Italian multiword "andarci piano" (Eng. "take it easy") within the following sentence:

IT: Coi superalcolici bisogna *andarci* veramente *piano*.
EN: People should *take it* really *easy* with liquors

As a first thing, this example shows that the level of graphical words can interact in interesting ways with other lexical analysis levels. On the one hand, graphical words can correspond to multiple lexicographic words, that is the kind of units that are listed as headwords in a dictionary. In the example above, the graphical word "coi" is the non-concatenating combination of a preposition (con, *with*), and an article (i, *the-plur*), whereas the graphical word "andarci" corresponds to a verb (andare, *to go*) and a clitic pronoun (ci, *there*). Composite words, such as *coi* and *andarci*, occur because two contiguous words can undergo phonological adjustment phenomena when they happen to occur one after the other in a text. Some of the adjustments are optional: for instance "coi" could be substituted by the original two words "con i", whereas the sequence of two words from which "andarci" is generated, that is "andare" and "ci", cannot occur without contraction in an Italian sentence.

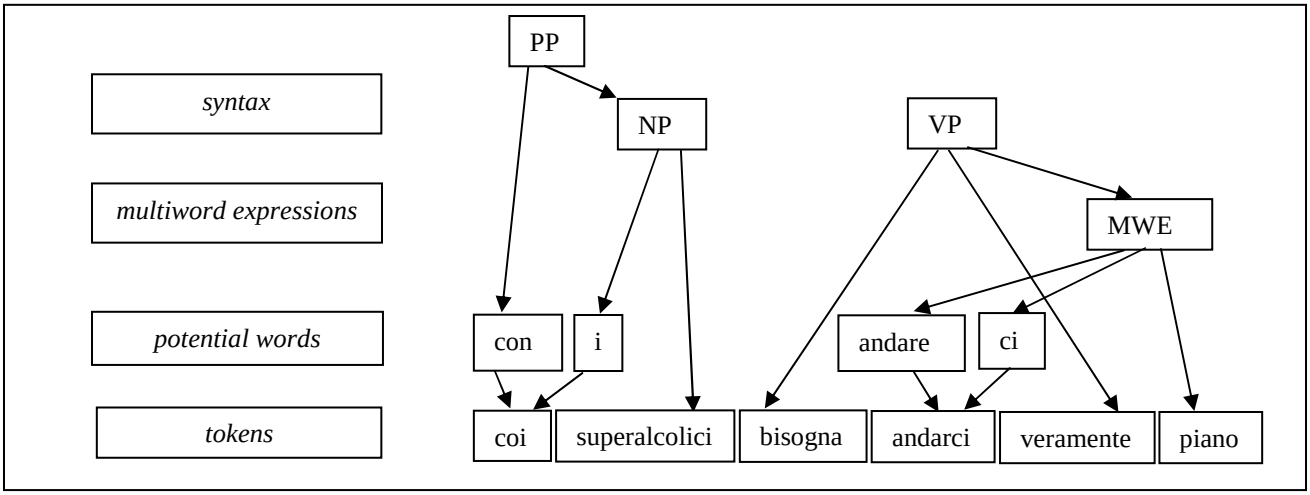


Figure 1. Interaction between lexical representation levels

On the other hand, the two graphical words "andarci" and "piano" correspond to one lexical unit with a unitary and non compositional meaning (*take it easy*). Also, note in the example that the two graphical words ("andarci", "piano") that compose the multiword, are non-contiguous, see Figure 1.

The example shows that we need to distinguish at least two other word-like units beyond graphical words. For sake of clarity, let us introduce the following notions and definitions:

a) **Token**: a graphical word (also called orthographical form), e.g.: *coi, superalcolici, bisogna, andarci, veramente, piano.*
b) **Potential word**: this notion was introduced by Pianta & Tovena (1999) to refer to an inflected word form before phonological and/or orthographic adjustment is applied to adjacent word forms, thus the notion makes sense from a generation point of view; note also that certain sequences of potential words may never occur in real texts because of obligatory adjustment rules. The potential words in our example are: *con, i, superalcolici, bisogna, andare, ci, veramente, piano.*
c) **Lexical unit**: one or more potential words carrying a unitary lexical meaning, e.g.: *con, i, superalcolici, bisogna, andare_ci_piano, veramente.*

The relations between these three levels can be complex. One token can correspond to more than one potential word, as in the examples below:

(Ita.) coi => con, i (preposition, article)
(Ita.) andarci => andare, ci (verb, clitic)
(Eng.) don't => do, not (verb, negation)
(Ger.) im => in dem (preposition, article)

On the other hand, more than one potential word can form a single lexical unit. This typically occurs with multiwords, as in "andare_ci_piano" and "take_it_easy".

As a further example consider the token "andarci" within the different sentences:

IT: Voglio *andarci* adesso
EN: I want to go there now

IT: Bisogna *andarci* piano
EN: People should take it easy

In the first sentence, the token "andarci" corresponds to two potential words (andare, ci) and to two lexical units (andare, ci). In the second sentence the token still corresponds to two potential words (andare, ci) but to only one lexical unit together with "piano".

The three levels illustrated above are conceptually distinct and, in principle, they correspond to three distinct levels of linguistic annotation.

Tokens are the basic representation level on which all the following ones are built. Generally speaking tokenization is not a trivial task. Many decisions need to be taken, and these decisions influence the analyses that are carried out at the following levels. To make some examples, tokenizing a text requires handling cases like the following ones:

- to distinguish a full stop that ends a sentence (separate token) from the full stop that ends an abbreviation (in-token character),
- to decide whether in "20%" the percentage sign is part of the preceding number or a separate token,
- to recognize that in "citta'" the "'" character is a representation of the accent on the "a" and not a quote surrounding the word, etc.

The representation of potential words and lexical units is also crucial for other representation levels. For instance, recognizing *potential words* is crucial for a correct syntactic annotation and also for word level alignment of parallel texts. If we do not distinguish the two potential words that compose the token "coi" we will not be able to annotate the prepositional phrase "coi superalcolici" with the correct syntactic structure: [*PP* con [*NP* i superalcolici]] (see Figure 1). We need potential words also to properly carry out word alignment of parallel texts:

andare [*align with*: go], ci [*align with*: there]

On the other hand, recognizing *lexical units* is crucial for lexical semantic annotation (e.g. with WordNet word senses), and for syntactic annotation as well.

In the current practice the first representation level is handled by the so-called *orthographic annotation*, which describes the actual tokens as they are found in the text. As for potential words and lexical units, they are usually represented together in one single annotation level, which is currently referred to as *morphosyntactic annotation*. Note that in this annotation approach, not only are the two levels based on an in-line annotation approach, but also no distinction is made between potential words and lexical units. This practice is due to the fact that in most cases lexical units and potential words coincide. When this is not the case, some problematic issues arise. Multiwords are the typical exception to the one-to-one correspondence between potential words and lexical units. The only proposal for representing multiword expressions that we could find in the literature is due to Ide & Romary (2002). However, this proposal has some limitations that we will examine in the next section.

4 Annotation of multiwords

The study on the annotation of multiwords that is presented in this section has been carried out in the framework of the MEANING project, more specifically in the context of the development of the Italian MEANING Corpus, a multi-level linguistically annotated corpus, having domain representativeness as main text selection criterion (Bentivogli et al., 2003). In designing the annotation scheme of the corpus we adhered as much as possible to the proposals for the new ISO/TC 37/SC 4 standard for linguistic resources (Ide and Romary, 2002), which are based on *annotation structures* (nestable <struct> elements) and *data categories* (<feat> tags). Different representation levels are contained in separate documents. Also, we use the XLink and XPointer syntax to represent relations between elements in different XML documents, and IDREFs attributes for relations within the same document.

```
<!-- morphosyntactic level -->
<!- - CONTINUOUS MULTIWORDS-->
<!- - w-level within mwd-level (lexical units coincide with pot. words) -->

<!-- bisogna (Eng. (people) should) -->
<struct type="w-level" id="w_4" xlink:href="#xpointer(id('t_3'))">
   <feat type="lemma">bisognare</feat>
   <feat type="pos">v</feat>
        ...
</struct>

<!-- andarci piano (Eng. take it easy) -->
<struct type="mwd-level" id="mwd_1">
   <feat type="lemma">andarci_piano</feat>
   <feat type="pos">v</feat>
        ...

   <!-- andare (Eng. take) -->
   <struct type="w-level" id="w_5" xlink:href="#xpointer(id('t_4'))">
      <feat type="lemma">andare</feat>
      <feat type="pos">v</feat>
      <feat type="mwd-function">head</feat>
           ...
   </struct>

   <!-- ci (Eng. it) -->
   <struct type="w-level" id="w_6" xlink:href="#xpointer(id('t_4'))">
      <feat type="lemma">ci</feat>
      <feat type="pos">clitic</feat>
      <feat type="mwd-function">satellite</feat>
           ...
    </struct>

   <!-- piano (Eng. easy) -->
   <struct type="w-level" id="w_7" xlink:href="#xpointer(id('t_5'))">
      <feat type="lemma">piano</feat>
      <feat type="pos">adv</feat>
      <feat type="mwd-function">satellite</feat>
           ...
   </struct>

</struct>
```

Figure 2. Annotation Scheme A, for continuous multiword expressions

In the actual annotation phase of the Italian MEANING Corpus, we faced the task of annotating multiwords, and we realized that the current annotation schemes available in the literature do not always allow to distinguish between potential words and lexical units, and do not provide satisfactory solutions for the annotation of discontinuous multiwords.

4.1 Continuous multiwords

If all the elements of each multiword were adjacent, we could still easily represent both the potential word and lexical unit levels through in-line annotation, following the proposal by Ide and Romary (2002). For instance, we can annotate the sentence "bisogna andarci piano" (Eng. "people should take it easy") as shown in Figure 2 above.

In Annotation Scheme A, simple lexical units (where potential words and lexical units coincide) are annotated with w-level structures, whereas complex lexical units are annotated in-line with mwd-level structures. Each mwd-level structure encompasses the w-level structures describing the single potential words which constitute the multiword. Note that in the example above, even if the annotation of multiword expressions is in-line with respect to potential words, the annotation of potential words at morphosyntactic level is stand-off with respect to the token level. This is in fact the only way to specify that the two potential words *andare* and *ci* correspond to the one token *andarci.*

This annotation scheme is slightly different from the original proposal by Ide and Romary, in which both simple and complex lexical units are annotated with w-level structures. We think that the w-level and the mwd-level are to be kept distinct, because certain pieces of information only pertain to the mwd-level. For instance, the lemma and the PoS of the multiword can only be annotated at the mwd-level. This is an important point that should be kept in mind to understand some of the proposals that will follow.

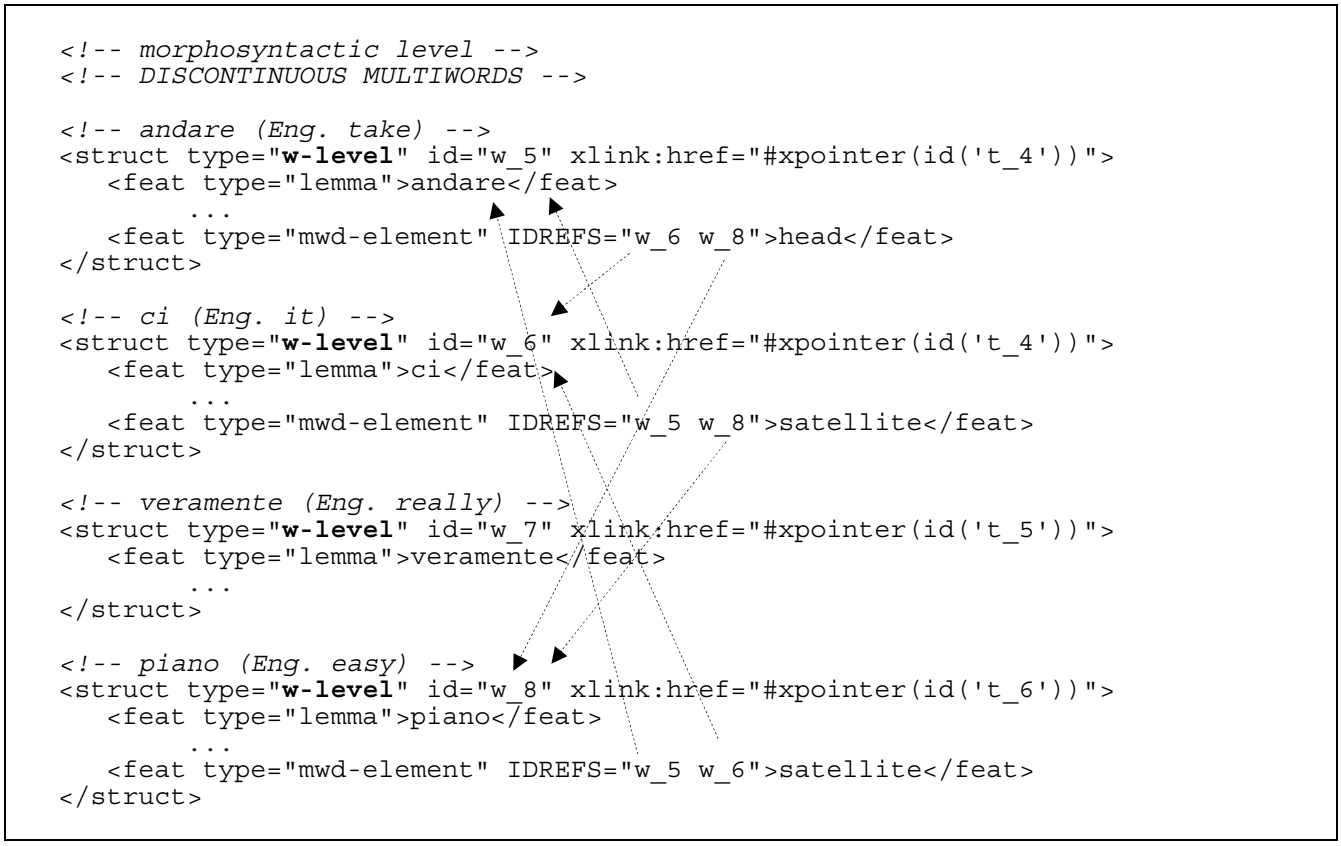


Figure 3. Annotation scheme B: w-level structures with IDREFs

Also, we explicitly mark the head and the satellites of the multiword (see the feature mwd-function), assuming that at least some features of the head (for instance agreement features) are passed over to the all multiword. Note also that the two potential words "andare" and "ci" point to the same token "andarci" in the orthographic file through XLink and XPointer links.

4.2 Discontinuous multiwords

Unfortunately, multiwords can be discontinuous, as is shown in the sentence of our case study "Coi superalcolici bisogna *andarci* veramente *piano*" (Eng. "People should *take it* really *easy* with liquors"). The adverb "veramente" (really) can be inserted within the multiword, but is by no means part of the multiword. Annotation Scheme A seems not to be suitable to represent this case. More specifically there seems not to be any way to represent both the fact that "andare", "ci", and "piano" compose a single lexical unit, and the fact that the adverb "veramente" occurs between the potential words "ci" and "piano", but is a distinct lexical unit.

In the rest of this section we will illustrate two alternative solutions based on in-line annotation (Annotation Schemes B and C), and another solution which requires a stand-off annotation (Annotation Scheme D).

The first solution is given in Annotation Scheme B (see Figure 3 above). All potential words are represented by w-level structures, and we do not use an explicit mwd-level. However we represent the fact that a potential word is part of a multiword through the feature tags in the w-level structure. When a potential word is part of a multiword, its w-level structure contains a `<feat>` tag like the following:

```
<feat type="mwd-element"
      IDREFS="w_6 w_8"> head </feat>
```

The advantage of this solution is its structural simplicity. We don't need to introduce a new type of structure to represent multiwords: all we need are pointers inter-connecting the various parts of each multiword, and discontinuity is not an issue. The disadvantages of this solution are on one side the proliferation of pointers, on the other side the lack of a specific structure to represent information that pertains to the multiword as a unit and not to its components, e.g. the lemma and the PoS. The lack of a specific multiword level structure is a problem also for higher level linguistic annotations. For instance, at the syntax level we would like be able to refer to a multiword as a unit (see the pointer that links the VP node to the multiword verb in Figure 1). It is hard to see how this could be done within Annotation scheme B.

On the other hand Annotation Scheme C (Figure 4) resorts to the explicit representation of the mwd-level. However, the strategy here is the opposite of the one used in Annotation Scheme A: instead of representing simple structures within complex ones, i.e. w-level structures within mwd-level structures, we represent information about complex structures within the simple ones.

Figure 4. Annotation Scheme C: mwd-level within w-level structures

In Annotation Scheme C, we include the mwd-level structure, containing the information pertaining to the multiword, within the w-level structure representing the *head* of the multiword ("andare"). This mwd-level structure contains also the pointers to the possibly discontinuous satellites of the multiword (through the IDREF attribute). The w-level structures describing the *satellites* of the multiword include a mwd-level structure each, containing a pointer to the head of the multiword.

Also this annotation scheme has some drawbacks. First, it may be incorrect or at least inelegant to nest conceptually complex structures within simple ones. Second, the description of the function of each element of the multiword (head vs. satellites) has been put at the mwd-level, even if it logically pertains to the w-level. Finally, selecting information about multiwords is somehow awkward, as it is contained within simple words.

There is a further solution (Annotation Scheme D represented in Figure 5) which solves these drawbacks resorting to stand-off annotation.

In Annotation Scheme D, the potential word level and the multiword level are represented in two different sections. The first section represents potential words through w-level structures and their ordering in the text. Information about multiwords is easily accessible in the second section, where each mwd-structure contains the relevant multiword information and pointers to the multiword constituents in the first section. The status of a word as element of a multiword is marked explicitly in the potential word section, whereas the information pertaining to the multiword level can be retrieved starting from the first section, by following the ID-IDREF link backward with an XPATH expression. On the other hand the stand-off syntactic annotation can point to unitary multiword level structures in the multiword section of the annotation.

Figure 5. Annotation Scheme D: w-level and mwd-level structures in two files (or sections of file)

It is worthwhile to note that if the potential word level and the multiword level are to be represented in two different files instead of the same file, annotation scheme D can still be applied substituting the IDREFs with XLinks and XPointers.

In annotation scheme D, as well as in the previous ones, when simple lexical units coincide with potential words, they are represented with plain w-level structures.

We think that the stand-off approach illustrated by Annotation Scheme D can be considered the best compromise to represent discontinuous multiwords, in terms of structural clarity, expressive power and conciseness, so this solution will be applied to the annotation of multiwords in the Meaning Italian Corpus.

5 Conclusions

In this paper we analyzed the problem of linguistically annotating discontinuous elements with XML-based annotation schemes. The difficulty of this task seems to have the same grounds as the difficulty to include multiple or alternative linguistic annotations in the same XML document, that is the fact that an XML document cannot represent multiple branch-crossing trees.

Whereas stand-off annotation is the standard solution proposed to solve the multiple (alternative) annotation problem, less attention has been paid in the literature to the issue of representing discontinuous elements. We analyzed this issue by taking as case study the representation of discontinuous multiwords.

To this extent, first we pointed out the opportunity of conceptually distinguishing between tokens (graphical words), potential words (words before phonological adjustment) and lexical units (lexical semantic units), by showing that the objects of these three levels do not always correspond in a one-to-one way. Second, we showed that annotation schemes available in the literature do not allow to represent discontinuous multiwords. Finally, we proposed four different annotation schemes in XML for representing the three linguistic levels introduced above, by taking into account the most recent proposals for linguistic annotation standards, and by making explicit the distinction between potential words and lexical units whenever they do not correspond in a one-to-one way. Three of the proposed annotation schemes allow to represent discontinuous multiwords. However we got to the conclusion that stand-off annotation is the most suitable approach to represent discontinuous multiwords, and, more generally, to represent the complex relationships that hold between tokens, potential words, and multiwords.

References

Bateman, J., Henschel, R. & Delin, J. (2002). A brief introduction to GeM annotation schema for complex document layout. In Proceedings of the 2nd Wordshop on NLP and XML (pp. 13--20) Taipei, Taiwan.

Bentivogli, L., Girardi, G. & Pianta, E. (2003). The MEANING Italian Corpus. In Proceedings of Corpus Linguistics 2003(pp. 103--112) Lancaster, UK.

Clark, J. (1997). Comparison of SGML and XML. World Wide Web Consortium Note 15 December 1997. http://www.w3.org/TR/NOTE-sgml-xml.html

Cunningham, H., Maynard, D., Bontcheva, K. & Tablan, V. (2002). GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, USA.

Harold, H. R. (2001). *XML Bible (second edition)*.

Ide, N. & Romary, L. (2002). Standards for Language Resources. In Proceedings of LREC 2002 (pp. 59--65) Las Palmas, Canary Islands, Spain.

Mann W. C. & Thomson S. A. (1987). Rhetorical Structure Theory: a Theory for Text Organisation. Technical Report RS-87.190, USC/Information Science Institute.

Murata, M. (1995). File format for documents containing both logical structures and layout structures. In Electronic Publishing, 8(4), 295--317.

Pianta, E. & Tovena, L. M. (1999). Mixing representation levels: The hybrid approach to automatic text generation. In Proceedings of the AISB'99 Workshop on Reference Architectures and Data Standards for NLP (pp. 8--13) Edinburgh, UK.

Sperberg-McQueen, C. M. & Burnard, L. (Eds) (2001). TEI P4: Guidelines for Electronic Text Encoding and Interchange: XML-compatible edition. The TEI Consortium. http://www.tei-c.org/P4X/.

Sperberg-McQueen C. M. & Huitfeldt, C. (1999). Concurrent Document Hierarchies in MECS and SGML. In Literary and Linguistic Computing 14, 29--42.

XCES: Corpus Encoding Standard for XML. http://www.cs.vassar.edu/XCES/

Enabling xComForTable Mapping to the Linguistic Annotation Framework

Marion Freese

IMS, Universität Stuttgart
Azenbergstr. 12, D-70174 Stuttgart, Germany
marion@ims.uni-stuttgart.de

Abstract

In order to support reusability of linguistic resources, the sub-committee 4 of ISO/TC 37 currently develops a general framework for linguistic annotation. In this paper we propose the integration of xComForT into this framework for a convenient resource integration. xComForT is an XML-based common format for text based language resources, that is designed to support easy mapping. Its data architecture allows for rich and flexible annotation as well as for tayloring the encoding scheme to the individual needs of the resource. We first present xComForT's main properties and use for richly annotated corpora, and then discuss its potential of adding value to the Linguistic Annotation Framework.

1. Introduction

The increasing use of data driven approaches in natural language processing (NLP) work has led to a need for large size text resources, such as newspaper text, HTML text, multimedia and multimodal corpora etc. The formats of the resources as well as the in- and output formats for NLP tools are highly heterogeneous. Therefore a standardized access to texts is essential for creation and exploitation of (richly annotated) corpora, in order to reduce the effort of converting text resources for NLP tool compatibility.The encoding for standardized access should provide for reusability and extensibility, which are the requirements for "good annotated corpora" as stated in (Ide and Brew, 2000).

In order to address these issues, an e**x**tensible **Com**mon **For**mat for **T**ext (xComForT, (Freese et al., 2003)) was defined, enhancing the Corpus Encoding Standard (CES, (Ide and Priest-Dorman, 1996; Ide, 1998)). Like CES, xComForT is flexible with regard to linguistic annotation schemes. In addition, it allows for modular definition extensions that can be taylored to the specific needs of the particular resource (e.g., newspaper text, multimodal or multimedia resources,...). Several tools have been created in the scope of this work, e.g. to support resource conversion to xComForT and to enable the integration of annotation/analysis tools.

As the stand-off annotation is strictly applied to xComForT, it enables creation, storage, manipulation and exploitation of richly annotated corpora. Due to its flexibility towards annotation schemes it could easily be integrated into the linguistic annotation framework (LAF) under development by ISO/TC 37/SC4[1] (Ide et al., 2003).

This paper first outlines the xComForT design and tool support in section 2., and then concentrates on those of its features that provide for easy integration into annotation frameworks (sections 3.,4.). In section 5., after showing that the xComForT framework meets all requirements of the LAF (as listed in (Ide et al., 2003)), we propose two integration possibilities to facilitate resource mapping to the LAF data model.

[1]ISO sub-committee (SC4) under Technical Committee 37 (TC 37, Terminology and Other Language Resources)

2. xComForT Overview

By enhancing XCES[2], the XML instantiation of the CES definition, an extensible Common Format for Text (xComForT) was developed in order to approach the goal to suffice the in (Ide and Brew, 2000) stated criteria for "good corpora", namely reusability and extensibility: Both the format specification as well as the texts encoded in xComForT are "reusable, i.e., potentially usable in more than one research project and by more than one research team, and extensible, i.e., capable of futher enhancement" (Ide and Brew, 2000).

xComForT basically consists of a definition for text structure encoding, based on de-facto standards, namely CES, TEI (Text Encoding Initiative, (Sperberg-McQueen and Burnard, 2002)) and XML. The three main features and its benefits are:

- standards-based ⇒
  common tools available and usable;
- stand-off annotation ⇒ easy plugging-in of arbitrary linguistic annotation scheme;
- easily extensible markup of primary document
  ⇒ easy adaptation to arbitrary resource.

The logical level data model of xComForT distinguishes primary data (i.e., raw data without annotations), metadata, markup of text structure (divisions, paragraphs, bylines etc.) and several levels of linguistic annotation (see figure 1). The text resource (i.e., the primary data) is stored in a "read-only" *base document*, enriched with metadata and text structure markup. The core definition for the base document is defined as TEI customization, instantiated using the TEI.2 DTD and TEI extension files. Its major part is adopted from XCES. However, the main feature of xComForT is the possibility to customize an invariant core definition with resource-specific extensions. Thus, it enables tayloring to the individual corpus models.

The physical level of representation is adopted from XCES, too, making use of XML, XML-related standards and stand-off annotation. The latter, however, is realized

[2]http://www.xml-ces.org/

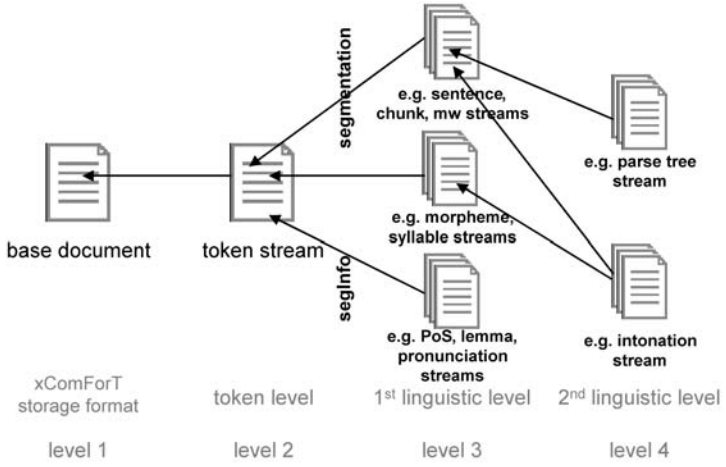


Figure 1: xComForT – annotation levels in stand-off architecture (inter-document linkage represented by arrows)

more consequently on the logical level by allowing exclusively structural markup in the base document. In addition, stand-off annotation enables a flexible integration and combination of linguistic annotation schemes.

Even though the xComForT scheme applies primarily to text resources, other language resources, media types and modalities can easily be inserted and interconnected due to xComForT's extensibility and flexibility in matters of the annotation document representation.

2.1. Data Architecture

In xComForT the stand-off strategy is adopted by separating the linguistic annotations from the original text that is enriched with structural markup and stored in a base document. Although xComForT does not include encoding definitions for linguistic annotation, we propose separate storage of each annotation type (e.g., sentence, part-of-speech), grouped into different levels of abstraction:

The annotation targets (tokens, chunks, etc.) implicitly group the annotation streams into several levels of abstraction[3] (see figure 1). The following levels cover the majority of annotation types in NLP work:

level 1 base document: text structure markup

level 2 basic segmentation: token annotation

level 3 linguistic level:

- segmentation
  (e.g., sentences, chunks[4],...)
- information on tokens
  (e.g., part-of-speech, lemma, ...)

level 4 higher linguistic level (based on level 3):

- hierarchies of segments
  (e.g., parse trees)
- information on level 3 segments
  (e.g intonation)

Annotation documents targeting level 4 documents will create a 5th level and so forth. Inter- and intra-level linkage is accomplished directly using the XML-related standards XLink (DeRose et al., 2001), XPath (Clark and DeRose, 1999) and XPointer (DeRose et al., 2002). For example, the token stream consists of one XML element per token with an XLink attribute indicating the token's start and end point in the base document. The separate storage of each single annotation type enables adding new streams and modifying existing ones without having to take into account the streams that are not relevant for the new stream.

2.2. Relation to the CES

The invariant xComForT core definition for the base document is based on the XCES DTD for the encoding of primary data[5]. The XCES definition has been modified in order to realize the separation of structural markup and linguistic annotation. A more detailed comparison is given in (Freese et al., 2003) and (Freese, 2002).

The most important modification at the logical level is the restriction to structural markup in the base document, serving as a common starting point for linguistic annotation[6].

The most important technical enhancement is the extensibility of the primary data encoding scheme via an extension mechanism, while still retaining TEI-conformance. The mechanism can be easily applied for user-defined customizations for a broad range of text formats as well as for integration of other resource types (e.g. including various modalities), as shown in section 4. Thus, the xComForT base definition provides more flexibility compared to XCES.

2.3. Annotation and Exploitation Support for Richly Annotated Corpora

Since mostly corpora of large size are required in NLP work, automatic conversion into the corpus representation

[3] Abstraction in terms of link distance (number of steps) to the base document, i.e. abstraction from the character sequence.

[4] Non-recursive chunks in the sense of Abney.

[5] http://www.cs.vassar.edu/XCES/dtd/xcesDoc.dtd

[6] As conversion from resource to the structural markup is source format dependent, whereas linguistic annotation (including abbreviations, dates, etc.) is not.

is desired, as well as automatic annotation and exploitation. The xComForT framework provides several tools to support these tasks.

2.3.1. Legacy Data Conversion

Two generic tools were developed to support conversion from proprietary formats: character set normalization[7] and generation of the gross xComForT structure (i.e., markup of divisions, e.g., articles).

The biggest portion of the conversion is admittedly the resource specific transformation. Therefore, a classification tool has to be developed or trained for the translation of specific text parts into the appropriate xComForT elements. However, since xComForT provides for resource reusability, only one conversion tool for each resource format has to be created instead of one tool for each combination of source and target format.

Exemplarily, a classification tool has been written for the format of the German newspaper Süddeutsche Zeitung. Furthermore, the integration of other text resources (e.g., linguistic corpora) was discussed within the scope of this work.

2.3.2. Integration of Annotation Tools

The NLP community has produced many tools for automatic linguistic annotation. xComForT provides the possibility to apply existing tools for creation of annotation documents. The integration of such external tools is supported by converting the tool output to annotation documents and XLinks to the related annotation target. For this purpose we defined a common internal representation of the tool input/output formats, enriched with the information of the original position of the input data in the xComForT document. Therefore, for each different input and output format of the annotation tools, a tool for the transformation from xComForT into the internal representation has to be created (e.g., an XSLT stylesheet).

Currently, we provide XSLT stylesheets enabling the annotation with the IMS tokenizer (Schmid, 2000), as well as with the IMS TreeTagger[8] (Schmid, 1994), and other tools that require the same input/output formats.

Instead of a specific tool for the annotation tool integration, an annotation tool framework that provides the adaptation of import/export filters can be used, e.g., the NITE XML toolkit[9].

2.3.3. Exploitation

The XML framework provides for a wide range of transformation possibilities due to the transformation language XSLT (Clark, 1999), that can be used to convert XML documents into other documents in any form (e.g., XML, plain text, HTML). XSLT supports document manipulation like selection of elements or portions of elements, transformation of extracted data and addition of information in the target document.

Thus, corpora stored in xComForT can be manipulated by extracting the appropriate information and transformed into the required format for any application.

Exemplary XSLT stylesheets have been created in the context of this work, showing the transformation of newspaper text – stored in xComForT with structural markup and annotation of tokens, sentence boundaries, POS and lemma information – in several target formats (plain text and XML)[10].

3. Annotation Flexibility

Within the scope of a standardized framework xComForT allows for flexible annotation of both text structure markup and linguistic information. It does not provide for its own linguistic annotation schemes, but consists of an extensible XML core definition for the markup of the text structure that applies to all text-based formats. In order to provide for maximum flexibility, xComForT has been designed to fulfill the requirements described in the following paragraphs.

3.1. Extensibility

With regard to the linguistic annotation, any XML-based annotation scheme can be plugged in due to the stand-off annotation architecture.

Furthermore, xComForT features a simple integration of user defined extensions for resource specific customization. Together, the core definition and the resource specific extension form the document type defintition (DTD) for the base storage format, constituting a TEI-extension. Thus, documents in base storage format are in *TEI local processing format* (Sperberg-McQueen and Burnard, 2002, chapter 28.1.2) as well. The combination of extension sets permits a modular generation of the base DTD, as will be shown in section 4. By means of separate DTDs the base storage format can be precisely taylored to different corpus types, type internal variations etc., which XCES does not provide: The XCES encoding definition for primary data lacks this modularity, since no extensions are foreseen. In order to adapt XCES to resource features that are not covered by the encoding standard, one would have to copy and edit the definition. Moreover, the XCES definition for primary data encoding consists of a single file. Thus, extended definitions intended for different resource types will be mixed up.

3.2. Incrementality

Due to the stand-off architecture, the linguistic annotation documents can be created incrementally, covering also ambiguities, alternatives, etc. Relations between annotations can be represented using XLink and XPointer, thus enabling comparison. Partial results can be represented and merged as well.

[7] All characters that do not belong to the target character set (passed to the tool as a parameter) are translated to the XML-Unicode representation.

[8] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

[9] http://nite.nis.sdu.dk/links/NXT/; see also (Bernsen et al., 2002).

[10] In order to handle the current lack of XLink/XPointer support, the XLink resolution is implemented manually, i.e., the target elements are addressed explicitly.

3.3. Uniformity

In order to provide for a uniform representation, the representations of the resource structures are based on the common core definition. For building definition extensions the extension mechanism (cf. section 4.) has to be applied. The stand-off annotation provides for uniform methods for combining primary data and linguistic annotations.

3.4. Consistency

In order to provide the possibility for consistent linguistic annotations, no linguistic information is encoded in the primary data document, but it is strictly separated from structural encoding. Thus, a consistent linguistic annotation scheme can be attached. For the structural features we designed a consistent encoding definition. Therefore the XCES definition was modified, e.g. in order to allow for a consistent use of the `<head>` element for headlines, which in XCES can be applied only division-initial, but does not apply for cross headings on the same level.

3.5. Potential for Richly Annotated Corpora

The strict realization of the stand-off annotation enables the creation, storage, manipulation and exploitation of richly annotated corpora. In addition to a faster processing due to smaller annotation documents, the separate storage of each annotation type allows for creating annotation documents that are independent of annotation documents at the same level or at higher levels[11]. This results in the following advantages:

- representation of alternative/concurrent/ambiguous annotations;
- flexible merging and comparison of annotation types;
- representation of partial and under-specified results;
- modifications (e.g. by changing the annotation tool) involve only the related annotations;
- inclusion of various annotation types, e.g. different modalities, meta data, links to media objects etc., without modifying the format's core definition.

Furthermore, the common format for encoding enables the application of common tools for corpus creation, annotation, manipulation and exploitation.

4. Extension Mechanism

Due to its extensible design, xComForT applies to a wide range of text formats. The invariant core markup definition covers all major text structural features (e.g. article, paragraph, byline, lists, placeholder for picture). In order to reasonably encode resources that comprise additional features, the core markup definition has to be extended with an extension definition tailored to the underlying resource characteristics. For xComForT this can be accomplished without modifying the original standardized core definition document, and - provided that a documentation will go along with the user-defined extension - TEI conformance will be retained (cf section 3.1.. This feature is one of the main advantages of xComForT compared to XCES.

In order to build an extended storage format definition, the user simply has to list the new definitions, compliant to a small set of extension guidelines (Freese, 2002). The guidelines basically prohibit restrictive modifications in order to guarantee that the core element definitions are a subset of the extension definitions. The new storage format will then be generated automatically, as illustrated in figure 2: The user should create separate documents with the following contents in DTD syntax:

- definitions for new markup elements;
- definitions for context extensions of existing element definitions for integration of the new elements;
- documentation of the extensions in terms of comments.

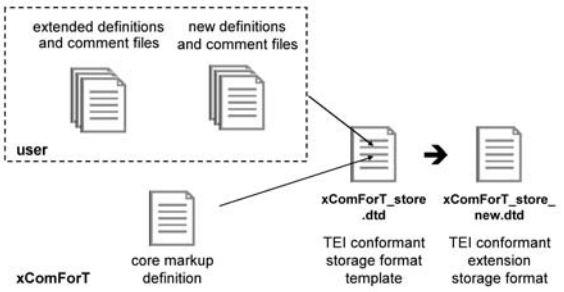


Figure 2: Creation of a user-defined extended definition in xComForT by applying the Extension Mechanism

These documents as well as the core markup definition will then be automatically inserted into an extension integration template, thus resulting in the new TEI conformant extension storage DTD (Document Type Definition).

The technical realization of the extension mechanism can be explained with the following example: In the scope of this work an extension DTD was created for the text structure of the Süddeutsche Zeitung, where e.g. the interviewer is frequently stated in the byline of an article. In the xComForT core definition each disjunctive content model definition[12] references an XML parameter entity related to the markup element. For the above example the relevant core definition lines are

```
<!ENTITY % x.byline ''>
<!ELEMENT byline (#PCDATA
                  | author %x.byline;)>[13]
```

The default content of these entities is empty. When the user redefines the entity content, this content will be added to the related element's content model definition, and thus

[11] Pointers to documents of the same or a higher level are not prohibited. However, they can possibly override the advantages of independency.

[12] An XML element's content model is the definition of the allowed structure of the element's content, i.e. of its embedded elements. A disjunctive content model has the form `(a|b|...)*`.

[13] For better understanding the content models of the examples are slightly simplified with respect to the full xComForT definition.

the core definition will be extended. In order to add the element <interviewer> as subelement of <byline>, the content model of <byline> has to be extended via the related extension entity x.byline:

```
<!ENTITY % x.byline '| interviewer'>
```

The parameter entity will be resolved DTD-internally by means of simple string replacement, resulting in

```
<!ELEMENT byline (#PCDATA
                  | author | interviewer)>
```

The element definition itself can be given independently of the core definition, e.g.

```
<!ELEMENT interviewer (#PCDATA)
<!ATTLIST interviewer
     type (short|full) #IMPLIED>
```

[14]

This extension mechanism does not allow for restrictive content model modifications. Thus the user is not able to violate the xComForT extension guidelines as regards the content model extensions and has only to attend to the compliance to the guidelines in matters of the attribute modifications, since these cannot be controlled by the mechanism due to the given W3C XML 1.1 recommendation.

5. Connection to the Standard for a Linguistic Annotation Framework

In this section the integration of the xComForT into the linguistic annotation framework (LAF) under development by ISO 37/SC4 will be proposed. We regard the integration as an effortless way of facilitating the mapping between heterogenous resource formats and the LAF data model. Beforehand we want to point out that xComForT meets the general requirements for a LAF presented in (Ide et al., 2003).

5.1. Requirements for Integration

xComForT was designed to provide the features outlined in section 3. These features are also part of the general requirements for a LAF. LAF's further requirements are fulfilled by the xComForT framework as well:

- *Expressive/Semantic adequacy, Openness, Consistency*
  Since there is no restriction regarding the choice of the linguistic annotation schemes (unless XML-conformity), these requirements are not violated. It will be up to the LAF data model to support representations for all varieties of linguistic information with formal semantics using consistent mechanisms.

- *Media independence*
  By means of the annotation streams also resources of other type than text can be integrated by making use of the XLink mechanims. Thus, relationships can be encoded between primary/annotation data and media streams (audio, video etc.), modalities (e.g., gestures), lexica, ontologies (e.g., using RDF/OWL), data categories (e.g., the MILE[15] Lexical Data Category Registry (Lenci and Ide, 2002)) etc.

- *Human readability*
  Due to an intuitive naming convention xComForT documents are human readable as far as possible within the limits of XML. Readability can be further improved by using commonly available XML display or editing tools.

- *Processability (explicitness)*
  The XML structure together with xComForT's intuitive naming convention determine an unambiguous encoding interpretation.

5.2. Two Proposals for Integration

The current LAF architecture is shown in figure 3. Its core component is the data model. The mapping between the user-defined document format on the left and the data model on the right is accomplished via a rigid XML-based "dump" format.

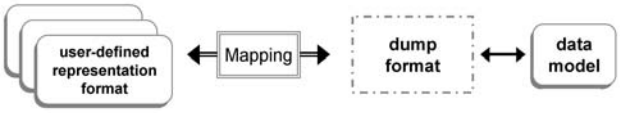


Figure 3: LAF architecture (in (Ide and Romary, 2003))

xComForT provides a core definition for primary data structure encoding that can easily be costumized to arbitrary resources. Furthermore, it is open for integration of any XML-based linguistic annotation scheme.

The following paragraphs describe two possible approaches to xComForT's integration into the LAF architecture that will lead to an easier mapping between proprietary resource formats and the annotation data model as well as to added value regarding resource reusability.

5.2.1. Intermediate Format as Common Document Form

The first proposal is to place xComForT as an intermediate format at user side (see figure 4). Since the "document form is largely under user control" in the LAF (Ide et al., 2003), a standardized intermediate format allows for a tightly defined and targeted mapping to the dump format. Thus, a common mapping tool, e.g. an XSLT stylesheet, can be provided by the LAF.
One example for a potential LAF dump format is given in (Ide and Romary, 2001), cf. figure 5. A possible encoding for the first part of the sentence ("Jones followed him into the front room .") that applies to the xComForT guidelines consists of the following documents[16](level 3 and 4 correspond to the encoding in figure 5):

level 1 Primary document with structural encoding (PTBraw.xml);

```
<xcomfortDoc type="text" extension="PTBraw"
   version="v0.6" TEIform="TEI.2">
 <cesHeader type="text" status="new" version="v0.1"
            TEIform="teiHeader">
   <!-- ... -->
 </cesHeader>
```

[14]The attribute type encodes the information whether the interviewer's name is given in short or in full form.

[15]Multilingual ISLE Lexical Entry

[16]xComForT uses the current XPointer syntax (http://www.w3.org/TR/2002/WD-xptr-20020816/).

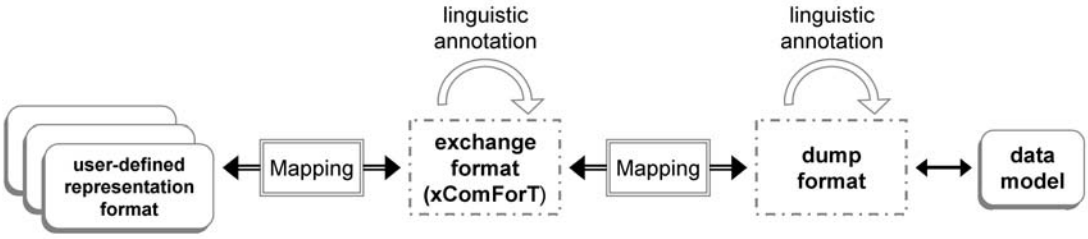


Figure 4: xComForT applied as additional exchange format in the LAF

```
<struct id="s0" type="S">
    <struct id="s1" type="NP"
            xlink:href="xptr(substring(p/s[1]/text(),1,5))" rel="SBJ"/>
    <struct id="s2" type="VP"
            xlink:href="xptr(substring(p/s[1]/text(),7,8))"/>
    <struct id="s3" type="NP"
            xlink:href="xptr(substring(p/s[1]/text(),16,3))"/>
    <struct id="s4" type="PP"
            xlink:href="xptr(substring(p/s[1]/text(),20,4))" rel="DIR">
        <struct id="s5" type="NP"
                xlink:href="xptr(substring(p/s[1]/text(),25,14))"/>
        </struct>
    <struct id="s6" type="S" rel="ADV">
      <!-- ... -->
</struct>
```

Figure 5: Penn Treebank example encoded according to the structural skeleton in (Ide and Romary, 2001)

```
 <text xml:lang="en">
  <!-- ... -->
  <div type="doc" id="div1">
   <p id="div1.p1">
Jones followed him into the front room .
<!-- next sentence -->
    </p>
</div>
</xcomfortDoc>
```

level 2 Token annotation document (token.xml);

```
<segments level="token" type="token"
    xml:base="PTBraw.xml">
  <tok id="div1.p1.tok1"
    xlink:href="#substring(id('div1.p1'),1,5)"/>
  <tok id="div1.p1.tok2"
    xlink:href="#substring(id('div1.p1'),7,8)"/>
  <!-- ... -->
</segments>
```

level 3 Sentence annotation document (sentence.xml):

```
<segments level="ling1" type="sentence"
    xml:base="/tmp/token.xml">
  <s id="div1.p1.s1"
     xlink:href="#xpointer(id('div1.p1.tok1')/
                 range-to(id('div1.p1.tok8'))"/>
</segments>
```

level 3 Chunk annotation document (chunk.xml):

```
<segments level="ling1" type="chunk"
    xlink:type="simple" xml:base="token.xml">
  <chunk id="div1.p1.chunk1" type="NP"
    xlink:href="#div1.p1.tok1"/>
  <chunk id="div1.p1.chunk2" type="VP"
    xlink:href="#div1.p1.tok2"/>
  <chunk id="div1.p1.chunk3" type="NP"
    xlink:href="#div1.p1.tok3"/>
  <chunk id="div1.p1.chunk4" type="PP"
    xlink:href="#xpointer(id('div1.p1.tok4')/
                range-to(id('div1.p1.tok7'))"/>
  <chunk id="div1.p1.chunk5" type="NP"
    xlink:href="#xpointer(id('div1.p1.tok5')/
                range-to(id('div1.p1.tok7'))"/>
</segments>
```

level 4 Chunk relation annotation document:

```
<segInfo level="ling2" type="rel"
    xlink:type="simple" xml:base="chunk.xml">
  <rel id="div1.p1.chunk1.rel"
      xlink:href="#div1.p1.chunk1>SBJ</rel>
  <rel id="div1.p1.chunk4.rel"
      xlink:href="#div1.p1.chunk4>DIR</rel>
</segInfo>
```

The mapping from xComForT to the LAF encoding proposal for the dump format in figure 5 can be accomplished by developing an XSLT stylesheet. Starting with the highest level documents as input, all XLinks can be resolved and the structure can be converted into the dump format.

The responsibility of converting to xComForT would still be on the producer of the resource. But we already developed a couple of supporting tools, e.g for creating extended format definitions (cf. section 4.) and a conversion tool to the xComForT base document which creates a basic structural markup down to division elements (`<div>`, e.g, articles) (cf. section 2.3.1.). Moreover, the user needs no detailed knowledge of XML or XML Schema to provide the mapping.

Furthermore, the resource with its annotations (possibly added incrementally via the dump format or by integrating annotation tools into the xComForT framework itself) will be available to each proprietary annotation format for which a mapping to the intermediate format exists. This results in an added value of reusability: Having a resource once mapped to the LAF data model, its linguistic annotations as well as the text/media structural markup will be fully available to the NLP community in a standardized format.

xComForT is flexible with respect to the linguistic annotation schemes. In addition, it is also flexible regarding legacy data encoding by means of enabling convenient standardized extensions of the structural markup definition, thus allowing adaptation to a wide range of resource-structural features (e.g. insertion and interconnection of other modalities, media types etc.), which XCES does not provide. Hence it allows for maximum flexibiliy for annotators as well as for easy adaptation to pre-existing annotations.

To sum up, xComForT as intermediate format offers at least flexibility in terms of both legacy data and new annotations, broad reusability and the possibility to support overall mapping to the data model. The dump format in the LAF can then provide for processing efficiency independently from the exchange format. Thus, changes to the dump format can be made without influencing the mapping between the resources and the intermediate/exchange format.

5.2.2. Dump Format

As xComForT allows for simple plugging-in of XML-based annotation schemes, the schemes[17] developed in the LAF as well as data categories, meta data documents etc. can directly be integrated. Therefore xComForT could also serve directly as the dump format (see figure 6), still providing its advantages as an intermediate format. In addition, the second mapping to the processing format would not be necessary. Since integration of external annotation tools is possible (as shown in (Freese, 2002)), the framework also allows for incremental annotation.

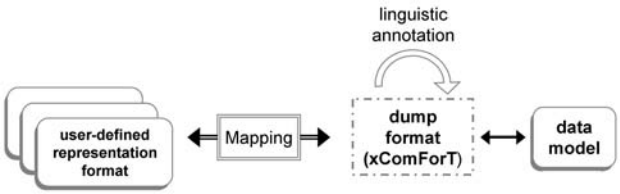


Figure 6: xComForT applied as dump format in the LAF

The LAF dump format is supposed to be a rigid format, whereas xComForT's main feature is flexibility. With its being flexible, however, it can be conveniently adapted to the LAF data model and afterwards frozen to a rigid format.

However, a rigid processing format should not serve as exchange format and vice versa. Firstly, because a rigid format cannot be adapted to the needs of the individual resource features. Secondly, because it must be possible to change the data structure of the processing format (e.g., for better processing efficiency) without influencing the mapping between the exchange format and the users' resource formats.

6. Conclusion

In this paper the potential of xComForT (**ex**tensible **Com**mon **For**mat for **T**ext) for richly annotated corpora has been shown (cf. section 3.5.). The basic cause of this potential is the format's flexibility, based on de-facto standards.

Both the definition of xComForT (and thus the encoding of text structure) as well as the linguistic annotation schemes can be customized to specific needs in the different fields of computational linguistics. For the customization of the text structure encoding an extensible core definition is provided, combined with an extension mechanism with which the definition can be convenienty taylored to the specific needs of the particular resource (e.g., newspaper text, multimodal or multimedia resources,...).

The stand-off markup provides for a flexible use of linguistic annotation schemes. Thus, it is possible to select the appropriate scheme for the individual needs, as well as to be open for emerging standards.

The second part of the paper proposed two possible ways of integration into the Standard for a Linguistic Annotation Framework developed under ISO/TC 37/SC 4. We showed that xComForT meets all the requirements for integration into the LAF, and then how the mapping between user-defined resources and the LAF data model can be facilitated.

We find the integration as intermediate/exchange format between the proprietary resource formats and the LAF dump format the most appropriate approach (cf. section 5.2.1.). It allows for a convenient mapping support to the LAF model, as well as for an independent processing format. Thus, it enables both maximum annotation flexibility and maximum processing efficiency.

7. Acknowledgements

The work described here was conducted in the framework of a diploma thesis at Sony Corporate Laboratories Europe (SCLE, Sony International (GmbH) Europe), and partially funded by the German Ministry for Research and Technology as part of the SmartKom project.

We want to thank the supervisors of this work, Martin Emele and Ulrich Heid, for their support and their fruitful comments.

8. References

Bernsen, Niels Ole, Laila Dybkjær, and Mykola Kolodnytsky, 2002. The NITE Workbench. A Tool for Annotation of Natural Interactivity and Multimodal Data. In *LREC Proceedings 2002*. Las Palmas, Spain.

Clark, James (ed.), 1999. *XSL Transformations (XSLT) Version 1.0. W3C Recommendation, 16 November 1999*. `http://www.w3.org/TR/xslt`.

Clark, James and Steve DeRose (eds.), 1999. *XML Path Language (XPath) Version 1.0. W3C Recommendation 16 November 1999*. `http://www.w3.org/TR/xpath`.

DeRose, Steve, Ron Daniel Jr., Paul Grosso, Eve Maler, Jonathan Marsh, and Norman Walsh (eds.), 2002. *XML Pointer Language (XPointer). W3C Working Draft 16 August 2002*. `http://www.w3.org/TR/2002/WD-xptr-20020816/`.

DeRose, Steve, Eve Maler, and David Orchard (eds.), 2001. *XML Linking Language (XLink) Version 1.0. W3C Recommendation 27 June 2001*. `http://www.w3.org/TR/xlink`.

Freese, Marion, 2002. *Einheitliches Format für strukturelle und linguistische Annotation heterogener Textressourcen - Definition und Werkzeugunterstützung*. Master's thesis, Uni Stuttgart.

[17] e.g., the current LAF encoding proposal for the dump format in (Ide and Romary, 2001), shown in figure 5.

Freese, Marion, Ulrich Heid, and Martin Emele, 2003. Enhancing XCES to xComForT. An Extensible Modular Architecture for the Annotation and Manipulation of Text Resources. In *NLPXML-2003 Workshop on Language Technology and the Semantic Web*, EACL. Budapest, Hungary.

Ide, Nancy, 1998. Corpus Encoding Standard: SGML Guidelines for Encoding Linguistic Corpora. In *Proceedings of the First International Language Resources and Evaluation Conference*, LREC. Granada, Spain. http://www.cs.vassar.edu/faculty/ide/pubs.html.

Ide, Nancy and Chris Brew, 2000. Requirements, Tools and Architectures for Annotated Corpora. In *Proceedings of Data Architectures and Software Support for Large Corpora*. Paris, France: ELRA. http://www.cs.vassar.edu/~ide/pubs.html.

Ide, Nancy and Greg Priest-Dorman, 1996. Corpus Encoding Standard - Document CES 1. http://www.cs.vassar.edu/CES.

Ide, Nancy and Laurent Romary, 2001. A Common Framework for Syntactic Annotation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, ACL'01. Toulouse, France. http://www.cs.vassar.edu/faculty/ide/pubs.html.

Ide, Nancy and Laurent Romary, 2003. Outline of the International Standards Linguistic Annotation Framework. In *Proceedings of ACL'03 Workshop on Linguistic Annotation: Getting the Model Right*, ACL'03. Sapporo, Japan. http://www.cs.vassar.edu/faculty/ide/pubs.html.

Ide, Nancy, Laurent Romary, and Eric de la Clergerie, 2003. International Standards for a Linguistic Annotation Framework. In *Proceedings of HLT-NAACL'03 Workshop on The Software Engineering and Architecture of Language Technology*, HLT-NAACL. Edmonton, Canada. http://www.cs.vassar.edu/faculty/ide/pubs.html.

Lenci, Alessandro and Nancy Ide, 2002. The mile lexical model: Linguistic and formal architecture. Presentation, ISLE/EAGLES Workshop "MILE (Multilingual ISLE Lexical Entry) as a Step towards Sharable Multilingual Resources". http://www.ilc.pi.cnr.it/EAGLES96/isle.

Schmid, Helmut, 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of NeMLaP 1994*. http://www.ims.uni-stuttgart.de/ftp/pub/corpora/tree-tagger1.ps.gz.

Schmid, Helmut, 2000. Unsupervised learning of period disambiguation for tokenisation. Internal Report, IMS, University of Stuttgart. http://www.ims.uni-stuttgart.de/~schmid.

Sperberg-McQueen, C.M. and L. Burnard (eds.), 2002. *TEI P4: Guidelines for Electronic Text Encoding and Interchange*. Text Encoding Initiative Consortium. XML Version: Oxford, Providence, Charlottesville, Bergen. http://www.tei-c.org/P4X/.

EULIA: a graphical web interface for creating, browsing and editing linguistically annotated corpora

X. Artola, A. Díaz de Ilarraza, N. Ezeiza, K. Gojenola*, A. Sologaistoa and A. Soroa

Faculty of Computer Science, Donostia / *School of Engineering, Bilbo
University of the Basque Country (UPV/EHU)
The Basque Country
jipdisaa@si.ehu.es

Abstract

In this paper we present EULIA, a tool which has been designed for dealing with the linguistic annotated corpora generated by a set of different linguistic processing tools. The objective of EULIA is to provide a flexible and extensible environment for creating, consulting, visualizing, and modifying documents generated by existing linguistic tools. The documents used as input and output of the different tools contain TEI-conformant feature structures (FS) coded in XML. The tools integrated until now are a lexical database, a tokenizer, a wide-coverage morphosyntactic analyzer, a general purpose tagger/lemmatizer, and a shallow syntactic analyzer.

1. Introduction

In this paper we present EULIA, a tool which has been designed for dealing with the linguistic annotated corpora generated by a set of different linguistic processing tools[1](Artola *et al.*, 2000). The objective of EULIA is to provide a flexible and extensible environment for consulting, visualizing, and modifying the documents generated by existing linguistic tools, which follow a coherent and general annotation scheme (Artola *et al.*, 2002).

The interface is based on a general document annotation scheme based on XML. XML provides us with a well-formalized basis for the exchange of linguistic information among the different text analysis tools. TEI-P4 conformant (http://www.tei-c.org/P4X/DTD/) feature structures constitute the representation schema for the different documents that convey the information from one linguistic tool to the next one in the analysis chain. So, XML-coded documents are used as input and output of the integrated tools.

XML is a well-defined standard for representing structured documents. Its value is due to the fact that it closes off the option of a proliferation of ad-hoc notations and the associated software needed to read and write them. The most important reason for using XML to encode the I/O streams between programs is that it forces us to formally describe the mark-up used, and that there exists more and more software available to deal with it.

The rest of the paper is organized as follows. Section 2 will be dedicated to explain the representation we have chosen for the linguistic information obtained from the different tools. In section 3 we present the information flow among the different linguistic processors. Section 4 describes the graphical interface with its main design features. Finally, section 5 presents conclusions and future work.

2. The annotation framework

A key issue in software development in NLP processes is the definition of a framework for linguistic knowledge representation. Such a framework has to satisfy needs entailed by the different tools and has to be general enough (Basili *et al.*, 1998). It is not trivial to adopt a formalism to represent this information. Different approaches have been considered for this task. For example, ALEP (Advanced Language Engineering Platform) (Simkins , 1994), can be considered the first integrating environment for NLP design. All the components (linguistic information, processing modules and resources) are homogeneously described using the ALEP User Language (AUL) based on a DAG formalism. Others, like GATE (Cunningham *et al.*, 1996), represent textual information by using the notion of textual annotation firstly introduced in the TIPSTER project.

There is a general trend for establishing standards for effective language resource management (ISO/TC 37/TC 4 (Ide *et al.*, 2003)). The main objective is to provide a framework for language resource development and use.

In our case, within a framework of stand-off linguistic annotation, the output of each of the analysis tools may be seen as composed of several XML documents: *the annotation web*. Figure 1 shows the currently implemented document model including tokenization, segmentation, morphosyntactic analysis, multiword recognition and lemmatization/disambiguation. This model fulfils the general requirements proposed in the standards (Ide *et al.*, 2003), as in (Bird *et al.*, 2000; Schäffer , 2003):

- It provides a way to represent different types of linguistic information, ranging from the very general to the very fine-grained one.
- It uses feature structures as a general data model, thus providing a formal semantics and a well known logical operation set over the linguistic information represented by them.
- Partial results and ambiguities can be easily represented.
- A general abstract model has been identified over the particular linguistic processors. Therefore, NLP applications are able to import/export the information they need in a unified way.
- The representation is not dependent on any linguistic theory nor any particular processing software.

[1]URL: http://ixa.si.ehu.es

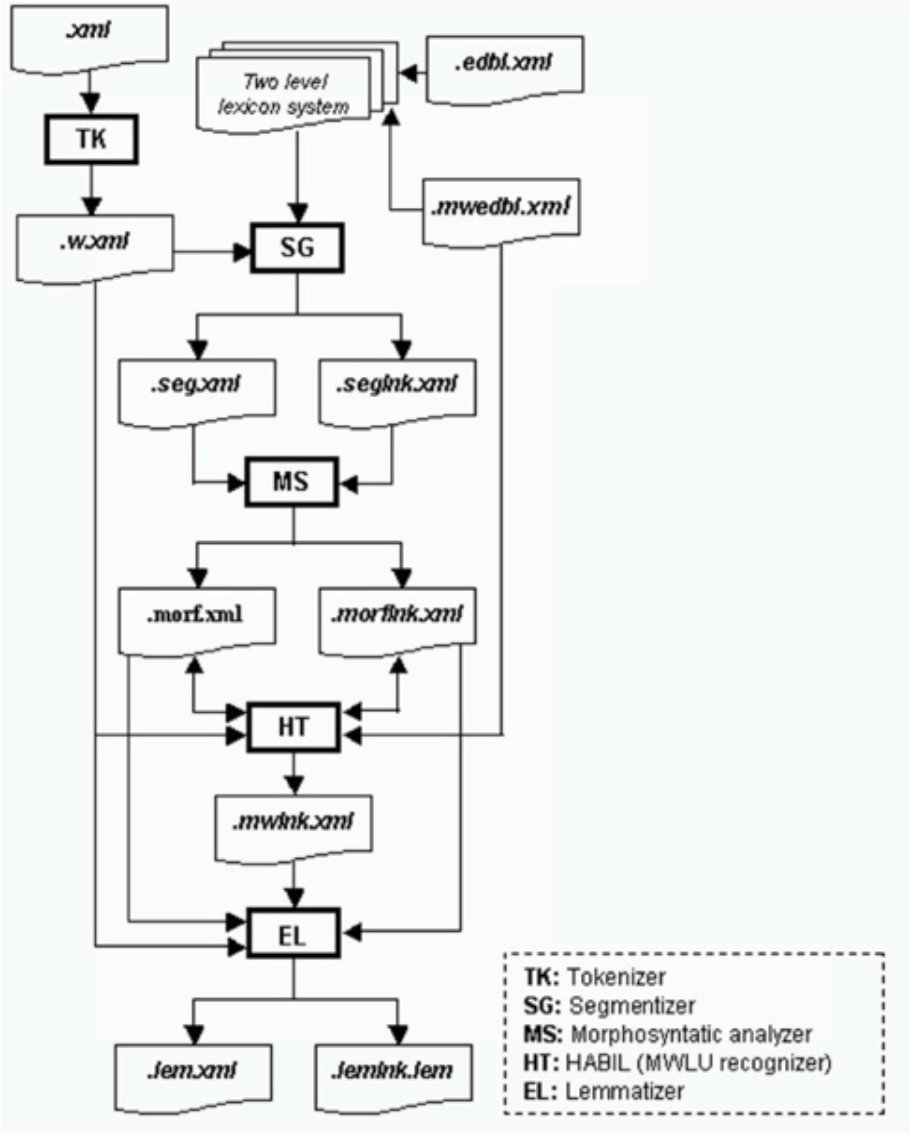


Figure 1: The multi-document annotation web

- As said before, our model relies in the XML mark-up language. XML is a well-defined standard for the representation of structured texts that provides a formal framework for the internal processing. As more and more pieces of software are available for checking the syntactic correctness of the documents, information retrieval, modifications, filtering, and so on, it makes it easy to generate the information in different formats (for processing, printing, screen-displaying, publishing in the web, or translating into other languages).
- Our model guarantees that no different mechanism is used to indicate the same type of information.

We identified the consistent underlying data model which captures the structure and relations contained in the information to be manipulated. These data models are represented by classes which are encapsulated in several library modules. These modules offer the necessary operations the different tools need to perform their task when recognizing the input and producing their output. These functions allow:

- Getting the necessary information from an XML document containing tokens, links, multiword structure links and FSs.
- Producing with ease the corresponding output according to a well-defined XML description.

We have identified different groups and types of documents:

- Text anchors: text elements found in the input.
  - Single-word tokens issued from the tokenizer. They are tagged with the XML <w> element, and represented by the W class.
  - multiword lexical units: the collection of "multiword tokens" identified in the input. The MWSTRUCT class represents the constituents of a multiword unit that are tagged by means of <link> elements. MWSTRUCTL represents lists of MWSTRUCT objects.

  - The structure of the syntactic chunks recognized in the text: the collection of "spans" identified in the input. The SPANSTRUCT class represents the constituents of a chunk that are also tagged by means of <link> elements.

- Analysis collections: collections of linguistic analyses obtained by the different tools. Due to the complexity of the information to be represented we decided to use feature structures as a general data structure. The use of feature structures quickly spread to other domains within linguistics since Jacobson (1949) first used them for the representation of phonemes. Feature structures serve as a general-purpose linguistic metalanguage; this reason led us to use them as the basis of our encoding. The feature structures in the integrated system are coded following the TEI's DTD for FSs, and they fulfil the Feature Structure Declarations (FSD) that have been thoroughly described for all the inputs/outputs in the tool pipeline. Following the object oriented paradigm, the following classes have been defined in order to deal with feature structures: FS (feature structure class), FL (list of features of a feature structure), F (feature class), FVL (the list of values of a feature), FVALUE (the value of a feature), and so on. The list of <fs> elements is represented by the class FSL. We distinguish two kinds of collections:
  - Libraries containing the analyses (FSs) corresponding to the text anchors set in the processed texts through the different analysis phases: *seglib, morflib, lemlib, sflib* and *deplib*. They are tagged by means of <fslib> elements.
  - Text-specific documents. Syntactic annotations associated to a particular input text.

- Links between anchors and their corresponding analyses, tagged by means of <link> elements. They are represented by the LINK and LINKL (list of LINK instances) classes.

- Documents: collections of text anchors —single tokens, multiword tokens and spans—, analyses, and links. Several classes to deal with the different kinds of XML documents participating in the annotation web have been defined: list of text elements (WXMLDOC), list of analyses (AXMLDOC), list of links (LNKXMLDOC), list of multiword units (MWXMLDOC), etc.

The multi-document annotation web gives, as pointed out in (Ide and Véronis , 1995; Ide *et al.*, 2003), more independence and flexibility to the different processes, and greater facilities for their integration.

3. The I/O stream between programs

These are the linguistic tools integrated so far:

1. EDBL, a lexical database for Basque, which at the moment contains more than 85,000 entries (Aduriz *et al.*, 1998a)
2. A tokenizer that identifies tokens and sentences from the input text.
3. *Morpheus*, a wide-coverage morphosyntactic analyzer for Basque (Alegria *et al.*, 1996). It attaches to each input word form all its possible interpretations. The result is a set of possible morphosyntactic readings of a word in which each morpheme is associated with its corresponding features in the lexicon: category, subcategory, declension case, number, and definiteness, as well as its syntactic function (Karlsson *et al.*, 1995) and some semantic features. It is composed of several modules such as:
   - A segmentizer, which splits up a word into its constituent morphemes.
   - A morphosyntactic analyzer (Aduriz *et al.*, 2000), whose goal is to group the morphological information associated with each morpheme obtaining the morphosyntactic information of the word form considered as a unit. This is an important step in our analysis process due to the agglutinative character of Basque.
   - A recognizer of multiword lexical units (MWLUs), which performs the morphosyntactic analysis of multiword units present in the text (Aduriz *et al.*, 1996).
4. *EusLem*, a general-purpose tagger/lemmatizer (Ezeiza *et al.*, 1998).

In the future we plan to integrate other tools currently under development, such as a shallow syntactic analyzer based on Constraint Grammar (Karlsson *et al.*, 1995; Aduriz *et al.*, 1998b),

Figure 1 illustrates the integration of the lexical database, the tokenizer, the morphological segmentation, morphosyntactic treatment, treatment of MWLUs, and *EusLem* (lemmatization) emphasizing that the communication among the different processes is made by means of XML documents. Thick line-border rectangles are used to represent processes, which will be described in sequence:

1. Having an XML-tagged input text file, the tokenizer takes this file and creates, as output, a $w.xml$ file, which contains the list of the tokens recognized in the input text. The tokenized text is of great importance in the rest of the analysis process, in the sense that it intervenes as input for different processes.
2. After the tokenization process, the segmentizer takes as input the tokenized text and the general lexicon issued from the lexical database, and updates the segmentation analyses library (FSs describing the different morphemic segments found in each word token) producing as well a document (*seg.xml*) containing the links between the tokens in the $w.xml$ file and their corresponding analyses (one or more) in the library. We want to point out that, because of the stand-off strategy followed in annotating the documents, different analyses may be easily attached to one token, thus allowing us to represent ambiguous interpretations.

3. After that, the morphosyntactic treatment module takes as input the output of the segmentation process and updates the library of morphosyntactic analyses. It processes the *seg.xml* document issued in the previous phase producing a *morflnk.xml* document containing the links between the tokens in the $w.xml$ file and their corresponding analyses (one or more) in $morf.xml$. This document will be later enriched by the MWLUs' treatment module. This module performs the processing of multiword lexical units producing an *mw* document that describes, by means of a collection of <link> elements, the structure of the MWLUs identified in the text. This module has obviously access to the morphosyntactic analyses and the *morflnk.xml* document, into which it will add the links between the *mwlnk.xml* document and the library.

4. The morphosyntactic analyses and the output of the tokenizer constitute the input of the lemmatizer. The lemmatizer updates the library of lemmatizations producing two link documents: on the one hand, a *lemlnk.xml* document that contains the links between the tokens and MWLUs, and their corresponding lemmatization analyses. The lemmatizer is also capable of updating the *mwlnk.xml* document if, due to the disambiguation performed, it has to remove some of the incorrect links previously included in it. Figure 2 shows a part of the document collection corresponding to the output of the lemmatizer.

4. EULIA: An application for creation, browsing and disambiguation on the annotation web

In this chapter we describe an extensible, component-based software architecture to integrate natural language engineering applications and to exploit the data created by these applications. The strategy we have explained for the integration of NLP tools is complex, as the linguistic information of different levels is distributed in many documents that must be processed. For any linguistic task it is necessary to coordinate different tools and data sources, and when we add new tools to the production chain, coordination will become more difficult. Therefore, in order to carry out the mentioned strategy, we have defined and implemented EULIA, a web-based interface.

4.1. Main functionalities

EULIA is an environment to coordinate NLP tools and to exploit the data generated by this tools. The NLP tools explained before are integrated in EULIA and new tools are currently being integrated. EULIA has two main goals:

- User-oriented linguistic data manager, with an intuitive and easy-to-use GUI.

- A system to integrate, coordinate and access NLP tools. This task is possible by means of a coordination module and the cooperation of this module with the user interface.

The GUI is a web-based interface which works with XML documents created by the integrated NLP tools. Its main functions are the following ones:

- consultation and browsing of the linguistic annotation attached to texts

- manual disambiguation of analysis results

- manual annotation facilities and suitable codification for new linguistic information

- simple text editor to create new texts

- submit a text to be analyzed in the coordination module

- search, queries and results analysis

- users control and personalization

4.2. Architecture and implementation

EULIA's implementation is based on a client-server architecture where the client is a Java Applet accessible by any Java-enabled web browser and the server is a combination of different modules implemented in Java, C++ and Perl (see Figure 4). All modules are designed using an object oriented methodology. As a consequence, EULIA presents a robust design which is easy to extend. The client's goal is to be the intermediary between users and NLP tools. It fulfils users' control and user requests' management. The interface provides different facilities which can be grouped in three main tasks:

- Data browsing: it visualizes the answers of the requests that users make to EULIA. Usually, these requests involve a complex procedure and need the information available in the server to resolve it; that is why the requests are processed by the server. In case, it is necessary to submit an answer to the user, this will be a XML document and will be visualized according to the suitable stylesheet (XSL document). These stylesheets could be changed dynamically depending on both the users' choice and the type of answer.

- Manual disambiguation: because of the integration strategy, disambiguation is an easy task. It consists of eliminating or marking the wrong links among analyses and units (token, multiword, dependencies, etc.) in link documents. EULIA presents a specific interface for this task which is generic for all link documents coded according to TEI guidelines.

- Manual annotation: depending on annotation type, a different kind of information is needed. In order to get these data, EULIA's GUI generates a suitable form, based on the XMLSchema, which defines the document's format for that annotation type. These forms are a HTML document and are generated using XSL documents. Communication between the GUI Applet and the server is established by means of Java Remote Method (RMI), which allows incremental construction of the communication protocol and a natural way to

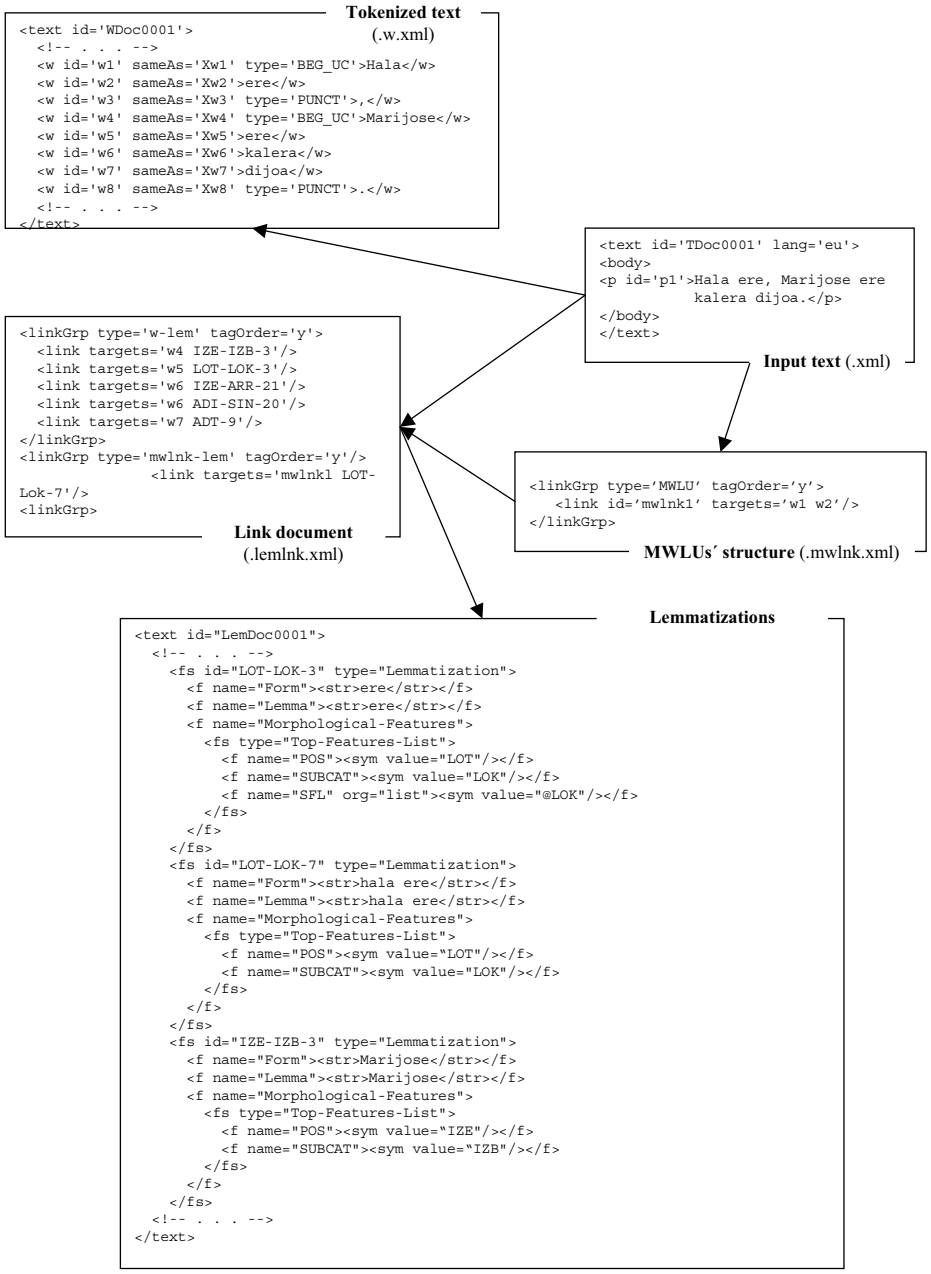


Figure 2: Output of the lemmatizer: a sample of the multi-document annotation web

relate client and server objects. While the client side of the EULIA system consists of an Applet, the server part contains a set of three modules. The first module gives service to clients and it coordinates the integrated NLP tools and stored linguistic analyses.

The second module is a layer between the coordination module and NLP tools. It carries out a generalization of the tools and the analyses.

Finally, the last component is not a module but a set of integrated tools and their outputs.

- Coordination: It coordinates clients' request process and submits the answer in a XML document. In order to answer clients' requests, sometimes it is necessary to generate new linguistic information by the use of integrated tools. Other times, it is enough to search the answer in an existing annotation web. In case it is necessary to generate new information, the system sends the request to the abstraction layer. On the contrary, if the request can be answered from the stored information, we use LibiXaML library to interpret the annotation web and to recover the documents from the abstraction layer. The coordination module has responsibility of managing the set of integrated NLP tools. The final objective of this module is twofold: a) To be the GUI's server and to answer GUI's requests. To solve the requests, this module distributes the tasks among the integrated tools. b) To create a workbench which facilitates the integration of NLP tools and the cooperation among them.

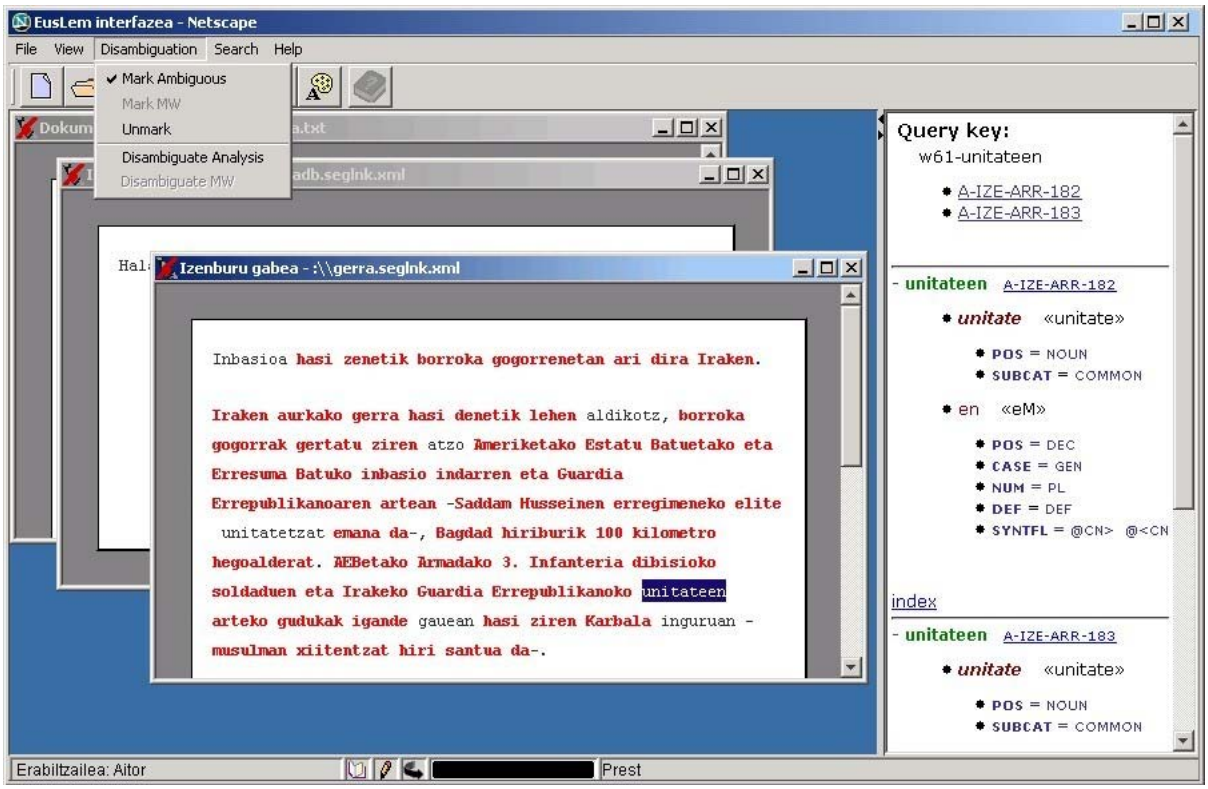


Figure 3: Application GUI.

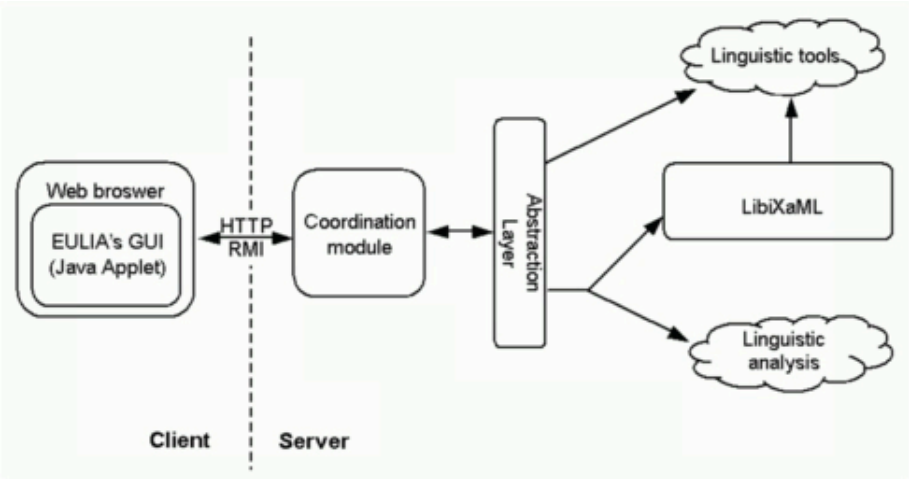


Figure 4: General architecture of the tool.

- Abstraction layer: the main goal of this module is to keep separate the coordination module of integrated tools, the analyses and their location. In order to archive this goal, this layer implements an interface for the coordination module. In this layer the relation between analysis type and tools and the way to recover stored information is defined, and it facilitates the definition of different computing paradigms to determinate the interaction among the linguistic tools. For the moment, a simple serial model has been implemented.

- Set of tools: this set is composed of integrated tools and their outputs. These tools' input and output are coded according to the integration strategy explained before.

In order to integrate a new NLP tool in EULIA system, the input and output of the mentioned tool has to be coded according to integration strategy presented before. Moreover, for a complete integration, it is necessary to define the relation between the new analysis type and tools and the stylesheets used to visualize this analysis. EULIA is a powerful system but it is not complex thanks to the integration strategy. In this strategy, all linguistic information is coded in a similar model, so the treatment of different data is similar. Moreover, EULIA is a generic system and offers many possibilities to be extended to different applications. EULIA is, without a doubt, a useful basis for different areas

of linguistic engineering.

4.3. Example

The interface has been designed to be easy-to-use and intuitive. The main window is divided into two parts (see Figure 3): a left MDI panel where the analyzed text is shown to the user, and the right part where linguistic information is shown in an understandable way. The interface provides hypertextual facilities, showing on the right hand-side linguistic information associated to items selected on the left part. The environment is designed as a tool for general users and linguists. The system gives the information the user has asked for about ambiguous units in a lemmatized text. It is important to notice that the item selected can be, in the example, a single word or a multiword expression, since currently the application has been tuned to deal with lemmatization results (actually, the selectability of text chunks depends on the underlying tool the interface is dealing with).

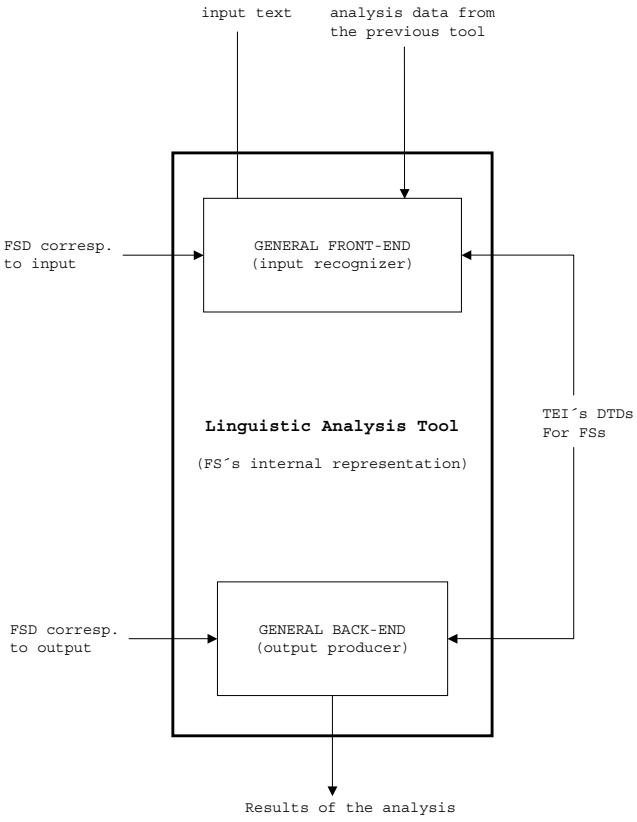


Figure 5: Schematic view of a linguistic analysis tool with its general front- and back-ends.

5. Conclusion and future work

We have presented a general environment for linguistic processing. The environment is oriented to be used by general users and has been designed to be informative, easy-to-use, and intuitive. It is coupled to a methodology of integration of linguistic tools based on a common annotation framework, general and extensible to similar systems.

For the near future, we are considering the feasibility of building general front- and back-end modules for the analysis tools, which will take as input the specific FSDs for each input/output. A schematic view of the integration of these general modules with a particular tool can be seen in Figure 5. This will facilitate the future integration of new tools into the analysis chain. Indeed, the work done so far confirms the scalability of our approach.

Acknowledgements

This research was partially funded by the Basque Government under the HIZKING21 program (project EJ-ETORTEK2002HIZKING21).

6. References

Aduriz I., Aldezabal J.M., Artola X., Ezeiza N.,Urizar R. 1996. Multiword Lexical Units in EUSLEM: a lemmatiser-tagger for Basque. In *Proc. in Computational Lexicography (Complex'96)*, 1-8. Linguistics Institute, Hungarian Academy of Sciences. Budapest (Hungary).

Aduriz I., Agirre E., Aldezabal I., Alegria I., Ansa O., Arregi X., Arriola J.M., Artola X., Díaz de Ilarraza A., Ezeiza N., Gojenola K., Maritxalar A., Maritxalar M., Oronoz M., Sarasola K., Soroa A., Urizar R., Urkia M. 1998. A Framework for the Automatic Processing of Basque. In *Proc. of the First Int. Conf. on Language Resources and Evaluation*. Granada (Spain).

Aduriz I., Aldezabal I., Ansa O., Artola X., Díaz de Ilarraza A., Insausti J. M. 1998. EDBL: a Multi-Purposed Lexical Support for the Treatment of Basque. In *Proc. of the First Int. Conf. on Language Resources and Evaluation*, vol II, 821-826. Granada (Spain).

Aduriz I., Agirre E., Aldezabal I., Arregi X., Arriola J.M., Artola X., Gojenola K., Maritxalar A., Sarasola K., Urkia M. 2000. A Word-Level Morphosyntactic Grammar For Basque. In *Proc. of the Second Int. Conf. on Language Resources and Evaluation*. Athens (Greece).

Aduriz I., Aldezabal I., Aranzabe M., Arrieta B., Arriola J., Atutxa A., Díaz de Ilarraza A., Gojenola K., Oronoz M., Sarasola K. 2002 Construcción de un corpus etiquetado sintácticamente para el euskera. In *Actas del XVIII Congreso de la SEPLN*. Valladolid (Spain).

Artola X., Díaz de Ilarraza A., Ezeiza N., Gojenola K., Maritxalar M., Soroa A. 2000 A proposal for The Integration of NLP Tools using SGML-Tagged documents. *Second Int. Conf. on Language Resources and Evaluation*. Athens (Greece). May.

Artola X., Díaz de Ilarraza A., Ezeiza N., Gojenola K., Hernández G., Soroa A. 2002. A Class Library for the Integration of NLP Tools: Definition and implementation of an Abstract Data Type Collection for the manipulation of SGML documents in a context of stand-off linguistic annotation. In *Third Int. Conf. on Language Resources and Evaluation*. Las Palmas. Spain.

Alegria I., Artola X., Sarasola K., Urkia M. 1996. Automatic morphological analysis of Basque. *Literary & Linguistic Computing*, 11, no. 4, 193-203.

Basili, R., Di Nanni, M., Pazienza, M.T. 1998. "Engineering of IE Systems: An Object-oriented approach". *Information Extraction: Towards scalable, Adaptable Systems*. M.T. Pazienza (Ed.). Springer Verlag.

Bird, S., Day, D., Garofolo, J., Henderson J., Laprun G., Liberman M. 2000. ATLAS: a Flexible and Extensible Architecture for Linguistic Annotation. In *Second Int. Conf. on Language Resources and Evaluation*. 1699-1706. Athens (Greece).

Cunningham H., Gaizauskas R.J. and Wilks Y. 1996. A General Architecture for Language Engineering (GATE) - a new approach to Language Engineering R&D. In *Proceedings of COLING'96*. Copenhagen.

Ezeiza N., Aduriz I., Alegria I., Arriola J.M., Urizar R. 1998. Combining Stochastic and Rule-Based Methods for Disambiguation in Agglutinative Languages. In *Proc. COLING-ACL'98*, 10-14. Montreal (Canada).

Goldfarb, C.F. 1999. *The XML Handbook*. Prentice Hall Iberia. SRL, Madrid.

Ide N., Véronis J. (eds.), 1995. *Text Encoding Initiative. Background and Context.* Kluwer Academic Pub.

Ide N., Romary L., 2003. A Common Framework for Syntactic Annotation . *Proc. of ACL'2001*, pp 298-305. Toulouse (France).

Ide N., Romary L., Clergerie E. de la, 2003. International Standard for a Linguistic Annotation Framework. *Proc. HLT-NAACL 2003 Workshop: Software Engineering and Architecture of Language Technology Systems*, pp 25-30. Edmonton (Canada).

Jacobson R. 1949. The Identification of Phonemic Entities. *Travaux du Cercle Linguistique de Copenhague*, 5, 205-213.

Karlsson F., Voutilainen A., Heikkilä J., Anttila A. 1995. *Constraint Grammar: A Language-independent System for Parsing Unrestricted Text*. Mouton de Gruyter.

Schäffer U. 2003. WHAT: An XSLT-based Infraestructure for the Integration of Natural Language Processing Components. *Proc. HLT-NAACL 2003 Workshop: Software Engineering and Architecture of Language Technology Systems*, pp 9-16. Edmonton (Canada).

Simkins N. K. 1994. An Open Architecture for Language Engineering. In *First CEC Language Engineering Convention*. Paris.

Thompson H.S., Tobin R., Mckelvie D. and Brew C. 1997. LT XML Software API and toolkit for XML processing. http://www.ltg.ed.ac.uk/software/xml/index.html

Simple Annotation Tools for Complex Annotation Tasks: an Evaluation

Stefanie Dipper, Michael Götze, Manfred Stede

University of Potsdam
Dept. of Linguistics, Computational Linguistics
D-14415 Potsdam, Germany
{dipper,goetze,stede}@ling.uni-potsdam.de

Abstract

This paper presents a comparative evaluation of ready-to-use, XML-based tools for annotating linguistic data. We start by describing our research project that deals with the creation and annotation of empirical data related to information structure. Based on the requirements of this project and the data, we develop a set of evaluation criteria and apply them in the evaluation of five selected annotation tools.

1. Introduction

Linguistic research based on real-life data has become more and more prominent during the last years. As a consequence, the need for corpora that are (i) large and (ii) richly annotated has grown as well. First, corpora that consist of real-life data such as newspaper texts or recorded dialogues must be large enough to offer enough instances of the phenomena under study. Second, many linguistic phenomena involve factors of different linguistic domains; for instance, word order in German is supposed to depend, among other things, on grammatical functions (syntax), thematic roles (semantic), information structure, and intonation (phonetics). Investigations of such phenomena require corpora that are annotated with detailed information at various linguistic levels.

The creation of large and richly annotated corpora is a time-consuming and expensive task. Whereas morphological and syntactic annotation may be supported, if not taken over, by trained taggers and parsers, the situation is different for the annotation of, e.g., semantic or discourse-related properties. Here, informed linguists have to perform all (or large parts) of the annotation task. Hence, people tend to restrict the data they are going to annotate to *relevant* data, i.e., data featuring the phenomenon in question. The resulting corpora are rather small but may be richly annotated. In such scenarios, the creation of a corpus is a side issue, which should not take up much time or effort. Hence, easy-to-use annotation tools that support manual annotation in a suitable way are desirable. Since the development of such tools is extremely time-consuming and expensive, reuse of already existing tools is to be preferred.

This paper grew out of our work in the Sonderforschungsbereich (SFB, collaborative research center) on information structure at the University of Potsdam[1]. In the context of this SFB, a lot of data of diverse languages will be collected and annotated on various annotation levels. In order to maximize the benefit of this data, we make use of an XML-based encoding standard to facilitate data exchange and reuse. The XML representation will be fed into a database that offers visualization and search facilities.[2]

This paper presents a survey and evaluation of selected, XML-based tools that can be applied in manual annotation. For the evaluation, we developed a set of criteria, based on the SFB requirements. We believe, however, that these criteria are relevant not only to the SFB but to many projects that deal with complex, multi-level annotation. We then applied these criteria to selected annotation tools.

Based on our user-oriented criteria, we believe that, at least in the short run, ready-to-use tools (i.e., tools which are easy to get used to, especially by users without programming skills) serve the annotator better than complex tool kits, which require adaptations by the user (as also argued by Orăsan, 2003).

The paper is organized as follows. First, we describe the context and requirements of the SFB. We then turn to the presentation of the criteria we have developed. Finally, we present the results of the evaluation and give a summary.

2. Requirements

In this section, we present our research project and describe the annotation scenario. Based on this, we formulate requirements for annotation tools, which we believe to be of relevance for similar annotation efforts.

2.1. The Project Context

The SFB "Information structure: the linguistic means for structuring utterances, sentences and texts" consists of 12 individual research projects from disciplines such as theoretical linguistics, psycholinguistics, first and second language acquisition, typology, and historical linguistics. The overarching objective of these projects is the investigation of information structure (IS). This is an area well-known to be prone to terminological or even conceptual confusion—many different theories of how to partition utterances into IS-relevant segments compete with each other, and, furthermore, there is little agreement on what level(s) of utterance representation IS should be located. In a situation like this, the availability of annotated data, which allows for comparing, sharing, and further developing the underlying ideas, is very important. The collection and distribution of empirical data is thus an important objective in the SFB. This concerns in particular the following projects:

[1] `http://www.ling.uni-potsdam.de/sfb/`

[2] The XML encoding standard and the database are under development. Our current work focuses on the annotation task, including the choice of annotation tools and the development of annotation standards.

Semantic annotation The project "A2: Quantification and information structure" examines the relation of quantifier scope and IS and will annotate semantic features such as quantifier scope, identifiability, and definiteness.

Discourse annotation "A3: Rhetorical structure in spoken language: modeling of global prosodic parameters" investigates the correlation between rhetorical and prosodic structure of spoken discourse. Data consist of radio news and newspaper commentaries.

Focus annotation in African languages The projects "B1: Focus in Gur and Kwa languages" and "B2: Focussing in African Tchadic languages" examine the phenomenon of focus in Western African languages. Both projects carry out field studies.

Diachronic data The project "B3: The role of information structure in the development of word order regularities in Germanic" investigates the evolution of the verb-second phenomenon, which occurred in certain Germanic languages only (e.g., in Modern German as opposed to Modern English). Based on language data of Old High German and Old English, the role of IS in this evolution will be studied.

Typology of information structure "D2: Typology of information structure" focuses on the development of a typology of the means for expressing IS. In close cooperation with the other projects, a questionnaire will be developed, which will serve as a basis to collect language data relevant for IS from typologically diverse languages.

One of the main objectives of the SFB is to determine the factors that play a role in IS. Hence, it is highly desirable that each project can profit from the data collected and annotated by the other projects. This presupposes compliance to certain standards, (i) an annotation standard and (ii) an encoding standard. First, the annotated data must be understandable and comparable. Therefore, SFB-wide working groups are defining an SFB Annotation Standard with tagsets and annotation guidelines for morphosyntax, prosody, semantics/pragmatics, and information structure. Second, we are developing an SFB Encoding Standard, an XML-based stand-off representation of the data, which will serve as the common exchange format within the SFB and thus support the standardization process.

Figure 1 gives an overview of the data flow in the project: a number of different projects will collect and annotate data according to the common SFB Annotation Standard, using a small set of annotation tools. The annotated data will be mapped to the SFB Encoding Standard, which serves as the common basis for further processing. This includes a web-based linguistic database, which provides visualization and retrieval of the SFB data, both by the members of the SFB and the research community.

The circumstances of the annotation differ: parts of the annotation will be done under conditions of fieldwork (as in the projects B1 and B2). Some tagsets to be applied are available, others will have to be created or developed further. Common to all of the projects are the limited resources available for the annotation task: annotation represents only one aspect of the project work and is usually not the main

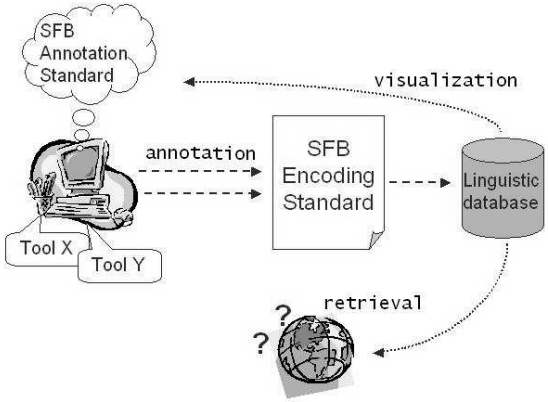


Figure 1: Data flow in the SFB

focus. Furthermore, some projects have no or little experience with annotating data at the levels mentioned.

2.2. Requirements for Annotation Tools

The described scenario is typical for cross-language research based on empirical data and focusing on the investigation of phenomena that require annotation on multiple linguistic levels. Based on the analysis of the needs of the SFB, we define the following list of requirements for annotation tools.

Diversity of data Language data to be annotated differs with respect to modality (written vs. spoken language, monologue vs. dialogue) and basic unit (sentence vs. discourse). In addition, special character sets (e.g., for Kwa languages) must be supported.

Multi-level annotation A very central requirement is support of annotation on multiple levels, each level representing one type of information, e.g. morphemic transcription, grammatical functions, pitch accents, etc.

Diversity of annotation Data types of the annotated information range from attribute-value pairs to set relations (e.g., for annotating co-reference), directed relations/pointers (e.g., for annotating anaphoric relations), trees, and graphs. Furthermore, it might be desirable to allow for annotations relating different levels ("cross-level annotation").

Simplicity The annotation tools must be simple tools, for several reasons. Users of annotation tools in the described scenario usually have little or no prior knowledge about annotation tools. Moreover, for less-studied languages, the researcher has to collect the data during field studies, which means that often there will be no technical support available. Finally, annotating data forms only a small part of the researcher's tasks; hence, using annotation tools should be as simple and intuitive as possible.

Customizability Usually, the development of suitable tagsets (including annotation guidelines) and the actual annotation are not independent tasks but affect each other. Suitability of tagsets and guidelines has to be proven in practice, i.e., by successful, consistent annotation. This

means that in the beginning phase, tagset definitions may change quite often. Tools should therefore allow for easy customization.

Quality assurance The annotated linguistic data is a central resource for the SFB research on IS. Hence, high-quality annotation is an important issue. Quality concerns consistency and completeness of the annotation as well as compliance to the SFB Annotation and Encoding Standards. The annotation tools should support these aspects in annotation.

Convertibility Support of data conversion is important for several reasons: First of all, this facilitates reuse of existing linguistic resources (e.g., treebanks or speech corpora). In addition, it supports standardization, since data usually have to be transformed into common standard formats (in our project: the SFB Encoding Standard). Finally, data convertibility is a prerequisite for applying specialized tools to the same data: tool X for transcription of the data, tool Y for annotation on multiple levels, and tool Z for for posing complex cross-level queries.

Tools may support convertibility in two ways: (i) by providing a standardized input and output format, which allows the user to easily convert the data; (ii) by providing ready converters from/to other tools.

It is important to note that the individual requirements might be of different relevance to different annotation projects. Their relative importance might also change over time: for instance, users could gather experience and would like to use more elaborate tools; the need for customizability might decrease; etc.

At the current stage of the SFB, the requirements of Simplicity, Quality Assurance, and Convertibility represent the most crucial needs. Later, issues like the support of more complex annotation (such as cross-level annotation or the annotation of ambiguous phenomena) will pose further challenges.

We now move to the tool evaluation criteria, which we derive from the requirements presented in this section.

3. Criteria

In this section, we first present the criteria we applied in choosing candidate tools ("selection criteria"). We then define the criteria for evaluating the individual tools, in the form of a feature checklist.

In line with the suggestions of standardization groups working on software evaluation (ISO, 2001; EAGLES, 1996), our evaluation starts from the user's needs. That is, both the choice of tools to be evaluated as well as the choice of evaluation criteria are guided by the user requirements that have been described above.

Note that our criteria do not test for highly detailed tool features (compared to, e.g., the feature checklist for translation memory by EAGLES (EAGLES, 1996)). This is because the tools we are comparing have been developed for different purposes and therefore exhibit many differing features, which we compare at a quite abstract level only.

3.1. Selection Criteria

We regard the following criteria as highly relevant for the SFB's annotation scenario and thus use them to restrict the set of tool candidates that we evaluate.

XML-based The tools must provide for an XML-based export and import format. This eases the data transfer between the annotation tools and the SFB-internal Encoding Standard.

Maintenance Maintenance of the tools should be guaranteed, hence we focus on tools that are being actively supported.

Ready and easy to use At the present stage, we consider tools that are ready and easy to use, i.e., installation and use of the tool must not require advanced programming skills. The end user (the annotator) should be able to apply the tool with little or no support.

Linguistic tools For similar reasons, we restrict the evaluation to tools developed and tailored specifically for linguistic purposes. That is, we exclude general-purpose tools such as XML editors.

Portability The tools must run on any platform and must be easy to install.

Cost The tools must be available free of charge for research purposes.

3.2. Evaluation Criteria

We developed a checklist of features to evaluate the tools one by one. These features can be classified according to the quality characteristics proposed by the ISO 9126-1 standard (ISO, 2001). Our features exemplify the ISO characteristics of "Functionality" and "Usability".

Functionality The aspect of Functionality concerns the presence or absence of functions that are relevant for a specified task. Roughly speaking, Functionality concerns the relation tool–task.

In our context, the ISO subcharacteristics of "Suitability" and "Interoperability" (which belong to the more general aspect of Functionality) are relevant.

- Suitability indicates whether a tool provides appropriate functions for the specified task.
- Interoperability concerns the capability of the tool to interact with other systems.

Usability In contrast to Functionality, Usability takes user aspects into consideration by evaluating the effort needed for use; i.e., it concerns the relation tool–user. In the SFB context, the following ISO subcharacteristics of Usability are important:

- "Learnability": Is the tool easy to learn?
- "Attractiveness": Does the user enjoy using the tool?
- "Documentation" measures the availability and quality of documentation.
- "Compliance": Does the tool adhere to standards/conventions relating to usability? For instance, for tasks such as text editing: does the tool provide features known from common text editors?
- "Operability": Is the tool easy to operate?

In the following paragraphs, we relate the ISO characteristics to the SFB requirements presented in 2.2. and define concrete criteria that instantiate these characteristics.

3.2.1. Functionality (I): Suitability

This aspect concerns the presence/absence of appropriate functions. Referring to the SFB requirements, Suitability indicates the tool's appropriateness with regard to the requirements of Diversity of data, Multi-level annotation, and Diversity of annotation.

The concrete criteria that we define to measure suitability of the tools concern source data and annotated data:

Primary/source data This criterion covers properties of the source data (i.e., the data that are input to the tool).

(1) Modality: Which input formats does the tool allow for?
   (a) discourse (sequence of sentences)
   (b) speech/audio
   (c) video
   (d) monologue
   (e) dialogue

(2) Preprocessing: Does the primary data need any preprocessing before annotation can start (e.g., is tokenization necessary)?

(3) Unicode: Does the tool support Unicode for the representation of special characters?

Secondary data This criterion concerns properties of the annotations.

(4) Markables (segments): The basic units referenced by the annotation are defined by inclusion/embedding (e.g., <markable>...</markable>) vs. specifying a start and end point (e.g., <markable span="id_2..id_4"/>).[3]

(5) Data structure: Secondary data consist of:
   (a) atomic features of a markable (e.g., part-of-speech tags)
   (b) relations between markables: (undirected) relations, pointers
   (c) dominance relations: bracketing, trees/graphs
   (d) conflicting hierarchies (e.g., overlapping markables or trees can be defined)

(6) Metadata: Can meta-information be annotated?
   (a) header: meta-information relating to the entire document (e.g., header data such as the author of an input text)
   (b) comments: referring to specific basic units or annotations

(7) Unicode: Does the tool support Unicode in the secondary data?

[3] Only tools that specify markables by their start and end point may represent conflicting hierarchies, see below.

3.2.2. Functionality (II): Interoperability

The aspect of Interoperability relates to interface properties, including the interaction with other tools. (All selected tools provide for an XML-based export and import format.) This feature covers the SFB requirement of Convertibility. We define the following criteria:

(8) Export and import
   (a) is stand-off representation supported?
   (b) can annotation schemes (see below) be imported/exported and if yes, in which format?

(9) Converters: Are converters from/to other tool formats provided?[4]

(10) Plug-ins: Is it possible to attach other tools?

3.2.3. Usability (I): Learnability/Attractiveness

The aspects of Learnability and tool Attractiveness—people should as much as possible enjoy annotation—relate to the SFB requirement of Simplicity. Since they are of central importance in our context, we performed a separate study of these issues, see Section 4.3.

3.2.4. Usability (II): Operability

We consider the aspect of Operability ("Is the tool easy to operate?") to cover the SFB requirements Simplicity, Customizability[5], and Quality assurance.

The criteria we define to measure Operability cover tool features that are tailored to the actual task of annotation. They concern features related to annotation schemes and the annotation process.

Specifying annotation schemes This criterion concerns tool features that allow the user to restrict the format and/or content of the annotation data (secondary data); it covers important aspects of Customizability.

(1) Annotation levels: Can levels be defined as obligatory, optional?

(2) Annotation tagsets: Can tagsets (i.e., admissible tag values) be specified? If yes, can the tagsets be structured, i.e., is it possible to define interdependencies between tag specifications? (For instance, the user is prompted to annotate the type of anaphoric reference only if the markable in question is marked as being anaphoric.)

(3) Specification: Are annotation levels or tagsets defined by external files or within the tool?

[4] Some tools provide APIs for further processing of the data, including conversion to other formats. However, the use of APIs requires programming skills, which we do not expect the user to have. Hence, we do not take API support into account.

[5] The requirement of Customizability could just as well be considered as reflecting the ISO characteristic of "Maintainability" (King, 2001).

Annotation process This criterion concerns properties of the annotation process.

(4) Automatic annotation: Does the tool support some kind of automatic annotation? (For instance, based on previously annotated data, the tool makes suggestions the annotator can accept or reject.)

(5) Selection-based: Does the tool support selection-based annotation? (For instance, only tags and tag values that are defined by annotation schemes are presented to the user.)

(6) Visualization: How is the annotated information presented?

  (a) scope: the annotated information is visible for all markables vs. only for the currently active markable (= the markable "in focus")

  (b) style: how is the annotated information displayed? (annotation as, e.g., text, XML source, or menu/radio button)

  (c) additional highlighting: does the tool provide further means to visualize the annotated information? (e.g., by coloring, font size/type, brackets, etc.)[6]

  (d) reference units of additional highlighting: do the additional highlightings in (c) refer to features or feature values? (e.g., all markables that are annotated for the feature "case" are highlighted vs. only markables with a specific case feature, e.g., "case = ergative", are highlighted)

  (e) user adaptation: can the visualization be changed dynamically by the user (e.g., by temporarily hiding certain annotation levels, by modifying coloring, font size, etc.)?

  (f) user definition: can the visualization be defined by the user?

(7) Search: Does the tool integrate a simple search facility (for primary and/or secondary data)?

3.2.5. Usability (III): Documentation

The aspect of Documentation relates to the SFB requirement of Simplicity. It refers to the availability and quality of:

(8) general documentation

(9) help (problem-specific documentation)

(10) example files, which can be loaded and modified

(11) tutorial (detailed walk-through)

3.2.6. Usability (IV): Compliance

Compliance ("Does the tool adhere to standards?") again relates to the requirement of Simplicity. We define criteria that concern features known from common document processing tools:

(12) Mouse vs. keyboard: Are there shortkeys for all (important) actions?

(13) Editing etc.: Does the tool provide undo/redo/autosave/...

(14) Unicode: Is there any input support for Unicode?

4. Evaluation

This section presents the results of the evaluation. First, however, the tools selected for evaluation are shortly described. Then the results of the feature checklist are given. In addition, we present results of a questionnaire focusing on tool usability. Finally, we present implications that our evaluation might have for the choice of annotation tools.

4.1. The Evaluated Tools

Given the selection criteria outlined above, we found the following tools to be suitable candidates.

TASX Annotator[7] 'Time Aligned Signal data eXchange Format' (Milde and Gut, 2002). The TASX Annotator allows transcription and annotation of speech and video data on multiple levels.

EXMARaLDA[8] 'EXtensible MARkup Language for Discourse Annotation' (Schmidt, 2001). EXMARaLDA aims at the multimodal transcription and analysis of discourse.

Since the TASX Annotator and EXMARaLDA specialize for speech annotation, the annotated information is represented by tiers and refers to segments ("events") that are defined with respect to a common timeline.

MMAX[9] 'Multi-Modal Annotation in XML' (Müller and Strube, 2001). MMAX is a tool for annotation of text and dialogue, following a strongly relation-based annotation paradigm.

PALinkA[10] 'Perspicuous and Adjustable Links Annotator' (Orăsan, 2003). PALinkA is an annotation tool that has been employed in several discourse-related tasks.

Systemic Coder[11] The Systemic Coder was initially developed in the context of a discourse analysis project.

The ready-to-use criterion excludes multi-purpose tool kits such as the Annotation Graph Toolkit[12] (AGTK), the NITE XML Toolkit[13], and CLaRK[14]. The issue of customizing a powerful toolkit to the needs of the SFB projects might be reconsidered at a later stage, when standards, formats, annotation and retrieval procedures in the SFB have matured.

[6] For the focus-based tools MMAX and PALinkA, additional highlighting concerns the annotation of all markables, not just the markable "in focus"—in contrast to (b).

[7] http://tasxforce.lili.uni-bielefeld.de/

[8] http://www.rrz.uni-hamburg.de/exmaralda/index.html

[9] http://www.eml-research.de/english/Research/NLP/Publications

[10] http://clg.wlv.ac.uk/projects/PALinkA/

[11] http://www.wagsoft.com/Coder/

[12] http://sourceforge.net/projects/agtk/

[13] http://sourceforge.net/projects/nite/

[14] http://www.bultreebank.org/clark/

4.2. Results of the Feature Checklist

The detailed results of the feature checklist evaluation are presented by the tables in Figures 3 and 4 . Figure 3 presents criteria measuring Functionality, Figure 4 lists criteria measuring Usability. The criteria are numbered according to Section 3.2.; '+' means: "feature (as defined in Section 3.2.) is available", '–' means "feature is not available".

These are the prominent findings (focusing on the SFB-relevant criteria):

Simplicity

- 'Ready-to-use': Here, the TASX Annotator and EXMaRaLDA perform best: They do not require any data preprocessing; no tagsets must (and can) be defined. The copy-and-paste function (see footnote [1] in Figure 3) allows for a quick start.

  With the Coder, the user has to specify annotation tagsets before annotation can start; however, the Coder supports tool-internal defining of tagsets.

  Finally, with MMAX and PALinkA, the user must preprocess the input text and define tagsets externally. Both requires an understanding of the XML format that underlies the data and tagset representation, respectively.
- EXMARaLDA offers a tutorial (in German), which allows even unexperienced users to get access to the tools on their own.

Quality assurance

- Predefined tagsets (MMAX, PALinkA, Coder) improve the quality of annotation (at the cost of simplicity), by defining admissible features and/or feature values; this improves consistency of annotation. Moreover, it improves completeness of annotation, by prompting the user to annotate the predefined tagsets. Finally, structured tagsets (MMAX, Coder) can be used for modeling decision trees, which guide the user through the annotation task.
- Good visualization is important. The tier-based tools (TASX Annotator, EXMARaLDA) display the annotated information in a straightforward way. The primary data and annotation layers are presented by horizontal tiers. That is, a sequence of adjacent markables and the associated annotations can be inspected simultaneously. However, only a small part of primary data can be viewed at the same time, which is a disadvantage for the annotation of phenomena that involve larger spans of discourse (such as discourse or anaphoric relations).
  In contrast, the focus-based tools (MMAX, PALinkA, and Coder) allow for concurrent visualization of a large amount of primary data, while annotated information is displayed for only one markable in turn. This drawback is partly compensated by annotation-dependent coloring of the primary data. The search facility provided by MMAX even allows for highlighting markables with feature combinations on different annotation levels (e.g. direct objects marked as topic).

Convertibility

- All of the selected tools offer XML-based import and export formats. Hence, all support convertibility in this aspect.
- In addition, some of the evaluated tools offer good opportunities for working with the same data in several 'special-purpose' tools (tools for annotation, visualization, querying). As the evaluation table shows (see footnotes [5]+[6] in Figure 3), the tier-based tools (TASX Annotator, EXMARaLDA) offer a lot of transformation opportunities.

Multi-level annotation/Diversity of annotation and data

- All tools support multi-level annotation. However, they differ with regard to the data structures of the annotated information. PALinkA and MMAX are the only tools that allow for structural annotation (by pointers, brackets).
- Only the TASX Annotator allows for direct annotation of audio and video data. With the other tools, this kind of data has to be annotated via an intermediate textual representation.

4.3. Results of the Usability Questionnaire

Since Usability is an important aspect for our annotation scenario, we decided to conduct an additional study with the future annotators of the SFB. We therefore provided a one-day tutorial about the annotation tools. After the tutorial, we asked the participants to fill in a questionnaire, reporting about their subjective impressions, covering aspects of Usability such as Attractiveness, Learnability and Operability. Due to time limitations, we considered only three of the tools: EXMARaLDA, MMAX and PALinkA.

A further goal of the tutorial was to get the annotators acquainted with a set of annotation tools and to enable them to work with the tools on their own. We therefore first introduced the basic functionality of each tool by demonstrating and practising segmentation and tag assignment, focusing on a simple annotation task on sentence level. After that, we addressed the process of preparing the primary data (preprocessing, tokenization) and the customization of tagsets (for MMAX and PALinkA only).

The most noteworthy results of the questionnaire are:

- The participants were most satisfied with the visualization in EXMARaLDA, where the annotation of sequences of markables can be inspected simultaneously. The XML-like visualization in PALinkA was criticized because of its poor readability. Apparently, additional means of visualizing annotated information (such as coloring, brackets) did not offer sufficient support. This means that visualization plays a highly important role in the annotation process.
- In the tutorial, we provided scripts for external preprocessing and tokenizing. Nevertheless, the preparation of the primary data remained difficult for the participants.

	TASX	EXMARaLDA	MMAX	PALinkA	Coder
Immediate Annotation	+	+	–	–	–
Consistent Annotation	0	0	+	+	+
Guided Annotation	–	–	+	0	+

Figure 2: Suitability according to the annotation scenario

- Customization of tagsets, which has to be performed tool-externally, was considered to be too complex by most of the participants. Understanding and modifying tagset specification formats requires more than can be expected from many users of annotation tools.

4.4. Implications

How can the findings of this section help the users to decide which annotation tool fits their requirements best? Viewed from the perspective of the purpose of an annotation task, we can distinguish three types of annotation scenarios:

Immediate annotation Immediate annotation implies that the tool allows the user to start the annotation without preparatory work. This requirement may be typical of preliminary, experimental annotations of a small amount of selected data.

Consistent annotation This requirement is important for the creation of high-quality corpora with complex (multilevel) annotation.

Guided annotation The annotation of certain phenomena require detailed and complex annotation guidelines, consisting of decision trees and lists of annotation criteria that the annotator has to check for. In such a scenario, guided annotation may model (parts of) the annotation guidelines.

The table in Figure 2 estimates the suitability of the evaluated tools with regard to these requirements ('+' means 'well suited for the annotation scenario', '–' means 'not suited', '0' is neutral).

5. Conclusion

In this paper, we presented selected XML-based tools that can be applied in manual annotation of language data. Due to the requirements of the SFB, we decided to focus on ready-to-use tools, which would not require programming skills.

On the base of a list of requirements, we developed a set of evaluation criteria for these tools, covering aspects of functionality and usability. Inspecting the results of this evaluation, we can state that these tools fulfill many of the criteria and offer a lot of support for the annotator. That is, the use of a small set of ready-to-use tools can be seen a worthwhile alternative to the application of complex toolkits, even for the multilevel and complex annotations the SFB is aiming at.

However, practice showed that the tools still require considerable effort for many users. The central drawbacks of the evaluated tools concern the visualization of the annotation, preprocessing of primary data, and tagset customization.

Our conclusions are therefore:

- Suitable visualization of the annotated information is highly important.
- A tool-internal preprocessing facility would render the tools more 'ready to use'.[15]
- A tool-internal interface for the specification of own tagsets would be an important step forward.

There is, of course, little value in seeking a "final ranking" for such a comparative evaluation of tools. Instead, it is clear that the annotation scenario determines which tools are suitable and which are not. We have suggested three such scenarios and provided a comparison of the tools along those lines (Figure 2). However, the potential users are encouraged to define their own, specific annotation scenario in terms of the fine-grained features we provided, and then peruse the information in Figures 3 and 4.

6. References

EAGLES, 1996. Evaluation of natural language processing systems. Final report. EAGLES DOCUMENT EAG-EWG-PR.2. Version of October 1996; `http://issco-www.unige.ch/projects/ewg96/ewg96.html`.

ISO, 2001. ISO/IEC 9126-1:2001: Software engineering – product quality – part 1: Quality model. `http://www.iso.org`.

King, Maghi, 2001. Standards work related to evaluation. MTEval Workshop Geneva, Hand-out; `http://www.issco.unige.ch/projects/isle/mteval-april01/maghi-isonew.html`.

Milde, Jan-Torsten and Ulrike Gut, 2002. The TASX-environment: an XML-based toolset for time aligned speech corpora. In *Proceedings of the third international conference on language resources and evaluation (LREC 2002)*. Gran Canaria, Spain.

Müller, Christoph and Michael Strube, 2001. MMAX: A tool for the annotation of multi-modal corpora. In *Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*. Seattle, WA.

Orăsan, Constantin, 2003. PALinkA: A highly customisable tool for discourse annotation. In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue*. Sapporo, Japan.

Schmidt, Thomas, 2001. The transcription system EXMARaLDA: An application of the annotation graph formalism as the basis of a database of multilingual spoken discourse. In *Proceedings of the IRCS Workshop on Linguistic Databases*. Philadelphia, PA.

[15] Such a facility is provided by the Coder: It enables the import of plain text and its segmentation into sentences, for instance.

Criterion	TASX	EXMARaLDA	MMAX	PALinkA	Coder
(I) Suitability					
Primary data:					
(1) Modality					
(a) Discourse	+	+	+	+	+
(b) Audio	+	–	–	–	–
(c) Video	+	–	–	–	–
(d) Monologue	+	+	+	+	+
(e) Dialogue	+	+	+	–	–
(2) Preprocessing	optional [1]	optional [1]	obligatory	obligatory	optional [1]
(3) Unicode	+	+	+	+	+
Secondary data:					
(4) Markables	start/end	start/end	start/end	inclusion	inclusion
(5) Data structure					
(a) Atomic features	+	+	+	+	+
(b) Relations	–	–	undirected rel., pointer	pointer	–
(c) Dominance rel.	–	–	–	bracketing	–
(d) Conflicting hier.	–	–	+	–	–
(6) Metadata					
(a) Header	+	+	–	+	–
(b) Comments	– [2]	– [2]	– [2]	– [2]	+
(7) Unicode	+	+	–	–	–
(II) Interoperability					
(8) Export/Import					
(a) Stand-off	–	–	+	–	–
(b) Annot. schemes	[3]	[3]	+, XML	+, text	+, text
(9) Converters [4]					
(a) Import	+ [5]	+ [6]	–	–	–
(b) Export	+ [5]	+ [6]	–	–	–
(10) Plug-ins [4]	+ [5]	–	–	–	–

[1] Primary data may be imported both in tokenized or untokenized format. TASX/EXMARaLDA: If untokenized data (= plain text) is to be imported, the data must be imported via copy and paste. Coder: Plain text files can be imported.

[2] These tools do not provide extra means for encoding comments. However, comments can easily be encoded as an ordinary annotation.

[3] TASX/EXMARaLDA: These tools do not allow for specification of annotation schemes, hence export/import of annotation schemes is not an issue.

[4] The given lists of converters and plug-ins are taken from the TASX and EXMARaLDA documentation. We did not check their functionality.

[5] The TASX Annotator provides import and export converters for Annotation Graphs, EXMARaLDA, Praat-label, ESPS-label, ESPS-freq. In addition, it provides import converters for Anvil, SyncWriter, Transcriber (STM), and export converters for NITE, HTML. Finally, it comes with plug-ins for Praat (Spectrogram, Pitch), sox.

[6] EXMARaLDA provides import and export converters for TASX, Praat TextGrid, ELAN Annotation File. In addition, it provides import converters for HIAT-DOS, ExSync Data, and export converters for AIF (Atlas Interchange Format), HTML partitur, RTF partitur.

Figure 3: Functionality evaluation

Criterion	TASX	EXMARaLDA	MMAX	PALinkA	Coder
(II) Operability					
Specifying annotation schemes:					
(1) Annotation levels	–	– [1]	–	–	–
(2) Annot. tagsets	–	–	+, structured	+	+, structured
(3) Specification	[1]	[1]	external	external	internal, extern.
.....	.....	.....	.....	.....	.....
Annotation process:					
(4) Automatic annot.	+ [2]	–	–	+ [3]	–
(5) Selection-based	–	–	+	+	+
(6) Visualization					
(a) Scope	all	all	focus	focus	focus
(b) Style	text	text	choice menu	XML	text
(c) Additional Highlighting	(+) [4]	coloring, font type, font size	(+) [4]	coloring, brackets [5]	coloring
(d) Reference unit	(feat, value) [4]	feat	(feat, value) [4]	feat	value
(e) User adaptation	tier hiding	tier hiding	–	+	–
(f) User definition	(+) [4]	–	(+) [4]	+	–
(7) Search	prim., second.	prim., second.	secondary	primary	primary
(III) Documentation					
(8) Documentation	+	+	+	(+) [6]	+
(9) Help	+	[7]	[7]	[7]	[7]
(10) Example files	–	+	+	+	+
(11) Tutorial	–	+	–	–	–
(IV) Compliance					
(12) Shortkeys	+	+	–	–	–
(13) Editing etc.					
(a) Undo/redo	undo, redo	–	undo (once)	undo, redo	–
(b) Cut/copy/paste	+	+	(+) [8]	–	(+) [8]
(c) Search/replace	+	+	–	–	–
(d) Autosave	+	–	–	+	–
(14) Unicode	virtual keyb.	virtual keyb.	–	–	–

[1] TASX/EXMARaLDA: These tools do not allow for explicit specification of annotation schemes. Within EXMARaLDA, however, XML elements specifying annotation levels may be added to the input data, thus simulating the definition of annotation levels.
[2] The TASX Annotator provides a completion function for the annotated information, by suggesting word completions (which can be accepted or rejected).
[3] PALinkA provides suggestions for annotation by taking previously annotated data into account.
[4] TASX and MMAX: Feature and value-depending coloring and fonts can be defined by the user. However, the definitions must be done tool-externally, by XSLT stylesheets. MMAX offers a query function that can be used to mark values.
[5] PALinkA allows the insertion of arbitrary, user-defined material to mark encodings visually.
[6] PALinkA's documentation is (at the time of writing) incomplete.
[7] Documentation and help files are identical.
[8] The features are not fully functional.

Figure 4: Usability evaluation

XML-Based Language Archiving

Peter Wittenburg, Hennie Brugman, Daan Broeder, Albert Russel

Max-Planck-Institute for Psycholinguistics,
Wundtlaan 1, 6525 XD Nijmegen, The Netherlands
peter.wittenburg@mpi.nl

Abstract

XML technology was one of the keys to build up a well-organized and accessible online collection of multimedia resources with complex multimodal annotations to which many different researchers and projects contributed. The metadata infrastructure is based on a linked and distributed domain of XML-based metadata descriptions created according to the IMDI standard. Multimodal annotations are created as XML-based texts structured according to the EAF standard linked with media streams. In both cases the nucleus are XML-structured files which are archival formats and can be accessed by any user without a special shell. Relational databases for example are only used to create optimized representations for special purposes such as indexes for speeding up searching. For presentation purposes the XML-files can easily be transformed to formats such as HTML.

1. Introduction

At the MPI for Psycholinguistics the multimedia/multimodal archive now comprises close to 30.000 sessions that can be seen as linguistically meaningful units of analysis. Most of these sessions have a multimedia basis in so far that the primary data is either based on sound or on video recordings – in total more than 5000 hours. A large fraction of these recordings are associated with annotations of various types. The archive also contains other linguistic data types such as lexicons, sketch grammars, field notes and others. Various projects contributed to this digital collection such as

- field workers from the MPI studying language behavior of different cultures and language acquisition processes by children and adults with the help of longitudinal observations;
- researchers of the MPI studying multimodal interactions in various circumstances and from various cultural backgrounds;
- researchers of the MPI and within the ECHO project studying sign languages from different countries teams documenting endangered languages from all areas of the world;
- Dutch and Belgium researchers building the Dutch National Spoken Corpus;
- researchers from 5 European countries studying the language use of immigrants.

Forming and maintaining this large archive that comprises various corpora from different researcher groups so that it is visible as one coherent collection and that it can be exploited with a limited set of tools has been a major effort during the last 4 years and was only feasible by relying on XML-based technologies. However, it has also to be made clear that proper data modeling is the step that has to be made first.

In the following we will outline how proper data modeling for different aspects of corpus creation, management and exploitation together with XML-based instantiation of these models helped us to cope with the challenges.

In this paper we shall use the following terminology: An **archive** denotes the full and organized collection of resources that has to be administered and offered to the user community. A **corpus** is a sub-part of this collection that was created by a person or a project. In the general sense **metadata** can be any data associated with other data, i.e., metadata can be annotations of video streams or of annotations, lexicons the entries of which refer to tokens appearing in a corpus, ontological entries that refer to concepts of the real world, keyword type descriptions of the resources in a collection or many others. For reasons of simplicity we will refer to metadata in this paper as the keyword type of description of resources that is useful for discovery and exploitation purposes.

2. Archive Organization and Management

When we took the decision to digitize all material and provide access to multimedia recordings not any longer via traditional audio/video technology but via computers, it was clear that we would be faced with the problem of how to organize the resulting large and ever increasing collection, how to give users access to the resources it includes and how to allow managers to maintain and extend the collections without ending in chaos.

A number of fundamental and far going conclusions guided us during the design and development phase. (1) From the beginning we assumed that our archive has to be seen as just one building block in a world-wide domain of online archives of language resources that are brought together by the Internet. Users and in particular agents would need an interoperable domain of language resources. As a consequence our organization solutions should be open for easy integration and exploitation. (2) We assumed that many of our typical users would work often without connection to the Internet at least temporarily. So our tools should be able to work with local collections without the need of connecting to a central site. (3) We understood that even within a discipline such as linguistics very different types of users would like to make use of the emerging domain of archives, i.e. shells addressing the specialists and those for the computer illiterates should be available. (4) Knowing that the underlying physical structure (storage architecture) would change regularly it was clear to us that

we would have to enable and convince people that they should discover and access useful resources via a virtual layer and not by using physical access paths. (5) Also from the beginning it was evident that the wishes of users and user groups in describing their data would be different so that flexibility was necessary. For more details about this work we refer to Wittenburg et al (2002a) and Wittenburg et al (2002b).

2.1 Metadata Model

The basis of all our archiving work was the design of the IMDI[1] metadata model in collaboration with others mainly within the ISLE[2], INTERA[3] and DOBES[4] projects. It should be used by the archive managers to organize the material in a way independent of the actual physical location and to carry out typical management tasks as far going as possible. It should be used by the users to discover and access the resources. Only such a system would give the system managers the freedom to take appropriate decisions at relevant moments without affecting the usage of the resources.

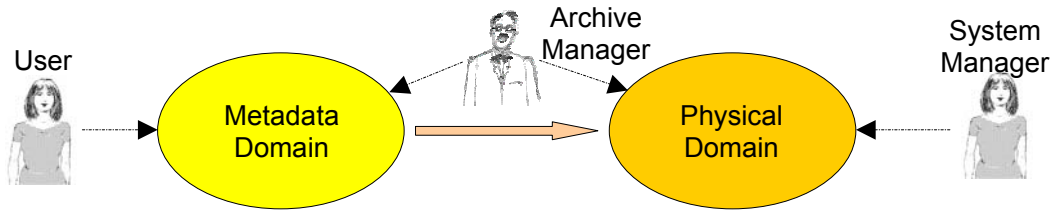


Figure 1 shows the distinction between the physical and the virtual metadata domain with which users are confronted.

When we wanted to uncouple users from the physical organization of the resources we had to understand first the way how users organize their data. Therefore the metadata model had to preserve the most relevant elements of resource organization. The major conclusions can be summarized as:

- Resources are organized in bundles of data at various levels. Recordings and their different levels of annotations are tightly coupled. They share the same time axis or the same notion of sequential order. Therefore the term "session" (later bundle) was introduced to denote the smallest addressable unit in an archive structure and to ask users to create metadata descriptions at this level, since the individual resources belonging to one session share most of the information necessary for discovery and retrieval.
- Resources are organized according to various criteria such as field trip dates, languages, age groups and others into manageable sub-corpora. Users also wanted to have some flexibility to regroup resources at this level dependent on their actual research interests. It was decided that this level of bundling could best be described by abstraction nodes representing some concept that the sub-sequent sessions share.
- Other structural relations between resources are for example that a lexicon is created for a specific language and at least partially derived from some of the resources in the archive. The creation of an abstract "language" node could document this relation and the lexicon would typically be associated with such a node. There are many different types of these relations.

These considerations on the one hand and the need to offer users a full-fledged alternative to the physical structuring methods led us, amongst other considerations, to a metadata model that is different from what was suggested for example by the Dublin Core (DC)[5] and OLAC[6] initiatives. Here metadata was introduced just for resource discovery via search engines. In our case we also decided to cover the organizational and management aspects. Only the possibility of for example grouping based on abstractions would allow us to treat them as units of management (copying, associating access rights, associating unique identifiers, etc).

2.2 Metadata Set

Another essential pillar of the IMDI metadata model was the definition of a metadata set. It had to mimic the bundling structure of sessions and be flexible enough to meet the individual needs of different projects. In addition to the mentioned criteria it was understood that only a structured set would meet the user's needs. While a flat set such as DC would not allow to relate attributes such as "sex" and "education" to participants except by introducing refinements, but then modifying the semantic scope of the concept itself, IMDI introduced structure into the set, i.e. various elements in the IMDI set can be associated with attributes that are part of the IMDI standard.

Flexibility was achieved defining a core set of IMDI elements on the one hand, but on the other hand data providers can add own element-value pairs at various places. Such extensions could theoretically lead to a proliferation of element categories reducing the success rate in resource discovery. First, the creation of metadata is a painful effort for data providers and the experience showed that they are in general satisfied with the semantics already offered by the core elements. Second, the introduction of "profiles" for specific user groups such

1 IMDI=ISLE Metadata Initiative, http://www.mpi.nl/IMDI
2 ISLE=International Standards for Language Engineering
3 INTERA=Integrated European language Resource Area
4 DOBES= Dokumentation Bedrohter Sprachen; http://www.mpi.nl/DOBES
5 DC = Dublin Core Initiative; http://www.dublincore.org
6 OLAC = Open Language Archives Community; http://www.language-archives.org

as Sign Language researchers or projects such as the Dutch Spoken Corpus project that used a number of additional elements taken from the TEI[7] give controlled extensions a quasi official status. So the current IMDI set has three layers with respect to the semantics supported: (1) the Core IMDI set, (2) special project or sub-discipline oriented profiles and (3) user specific extensions.

Further, the IMDI set needed to be accompanied by definitions of controlled vocabularies that define the range of values certain elements can take. It was decided in the metadata model to not make controlled vocabularies part of the schema controlling the IMDI set itself, since it was foreseen that they will change frequently to meet the needs of the users and to not force users to define their own elements due to a missing value. For specific elements such as "language code" it must also be possible to allow the inclusion of different vocabularies such as the ISO norms and the Ethnologue list.

2.3 IMDI Infrastructure

Based on this elaborated IMDI metadata model we were able to design the basics of the concrete IMDI infrastructure. In contrast to other approaches that start with a relational database implementation as the core, we decided to use a linked domain of XML-files as the core data structures of the IMDI infrastructures. All other data structures would be derived from these XML representations.

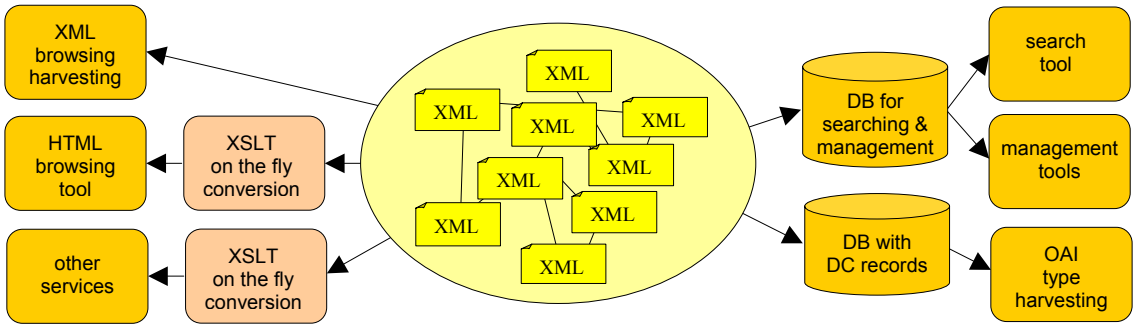


Figure 2 shows the elements of the metadata infrastructure with the IMDI files as the primary and various others as secondary formats.

This approach has shown to have many advantages for us. (1) It immediately allows us to connect various emerging IMDI sub-domains by simply adding links, i.e. IMDI can operate in a distributed domain of resource providers without any additional efforts. (2) It allows everyone to crawl in this domain and create any service that can be useful to harvest and exploit the rich IMDI metadata domain. (3) It allowed us to build an XML browser exploiting the special features of the IMDI model such as the bundle concept, but also allowed us to offer HTML versions such that users can exploit the IMDI domain with normal HTML browsers. (4) From a structural perspective it is easy to generate DC records so that OAI type of harvesting is supported. (5) It is easily possible to generate all kinds of specialist databases to efficiently support searching and management. (6) Extractions of sub-corpora from the pool of IMDI metadata descriptions are easily made and copied for example to a notebook including all relevant structure information.

With respect to management a whole set of integrated analysis programs are available to check the correctness and state of the archive's structure. Further, an access management framework was developed that allows archive managers and responsible researchers to define access rights. Here a canonical metadata hierarchy is essential since the manager can select an arbitrary node and set rights for all resources that occur below the selected node.

The creation of HTML representations of the metadata descriptions can easily be achieved by style-sheet-based XSLT transformations that are carried out on the fly. However, the indication of the position in the linked metadata structure as a navigation help requires additional program execution at the server side.

Therefore central in many ways is also the availability of well-documented XML-schemas and concept definitions. The XML schemas describe the structure of the metadata descriptions and of the controlled vocabularies. Aspects such as support for multilingualism had to be considered since IMDI is used in various countries. All XML-Schemas defining the IMDI set are openly available in the Web (www.mpi.nl/IMDI).

2.4 Metadata Interoperability and Future

Metadata interoperability will become one of the essential pillars of the Semantic Web. At the encoding level interoperability is achieved by using UNICODE and at the syntactic level by using XML and by having validated the created IMDI files. We know from practical experience in the ECHO project where we created an interoperable metadata domain of 5 different humanities disciplines how important the encoding and syntactical interoperability is. We were confronted with non-validated XML repositories generated from databases of various types also including different character encodings. It was and is a very time-consuming effort to transform these repositories into useful representations. In a dynamic environment where such repositories change continuously this is not feasible. For many holdings we are still far away from the ideal state that the OAI MHP protocol is used where well-checked data is offered by the data providers.

It is even more problematic to achieve interoperability at the semantic level. Currently, mapping relations between

7 TEI = Text Encoding Initiative; http://www.tei-c.org

elements are hardwired into a wrapper to realize for example the IMDI-to-OLAC mapping. This is an unsatisfying approach since no one can influence the mapping. Within ISO TC37/SC4 we work on a framework where all concepts used are described in an XML-based and ISO compliant (ISO 11179, ISO 12620) Data Category Registry (DCR). In this way semantics are defined in a machine readable form and individual schemas will refer to entries in the DCR. While equality relations between two metadata sets can easily be implemented by referring to the same DCR entry, more complex relations can be implemented as RDF assertions referring to two different entries. The emerging ISO framework will naturally improve the semantic interoperability and open the possibility for projects to define their own metadata sets by re-using concepts that are already defined in the open DCR.

3. Archive Exploitation

The archive contains a number of different linguistic data types such as lexicons, field notes and others that we do not want to discuss in this paper. We would like to focus on the complex problems associated with annotated multimedia recordings and texts and the important role of XML in this context. We will also not discuss the encoding aspects in detail, but focus on structural aspects in such annotated multimedia recordings.

Also in this respect we were guided by a number of fundamental decisions: (1) We are faced with incrementally added and updated annotations within a tier, but also on newly created annotation tiers. (2) In multimodal interaction studies one can easily have a large amount of annotation tiers (>50). (3) Annotations will exhibit all possible time relations since the multimodal streams such as speech, gesture, eye movements and others have to be seen as independent from each other. (4) Annotations will refer to periods in media time, but also to sequences in other annotations. References to points in time are seen as periods of unity length. (5) Some annotations will exhibit hierarchical relations within tiers (syntax trees) or across tiers as in the case of interlinearized morphological glossing. (6) Different types of cross-referencing have to be supported that allow to refer to many objects on different tiers in the extreme case.

Similar as in the case of metadata it was the intention that the annotated recordings should be available in well-documented formats to users directly so that they can carry out their own types of processing on them. On the other hand we wanted to provide (multimedia) tools that allow to create and exploit complex annotations. For those who have problems to even download and install tools, access options via normal web-browsers were planned.

3.1 Abstract Corpus Model

Guided by the above mentioned criteria and our experience with complex multimodal annotations for more than 6 years we were able to design and further develop an Abstract Corpus Model that was seen as the blueprint for writing programs like ELAN[8]. This ACM has the power to express the complexity mentioned above and it should be possible to import many of the well-known formats such as CHAT[9], SHOEBOX[10] and others. Therefore ACM can be seen as an attempt to define a general model for complex annotations. Our experience showed, however, that we had to refine the model several times to make it powerful enough to handle the continuously increasing demands of the scientists.

Further, the model had to account for annotation tier types and constraints that are specified for annotation types. The modeling was carried out in UML. Figure 3 indicates the core part of ACM.

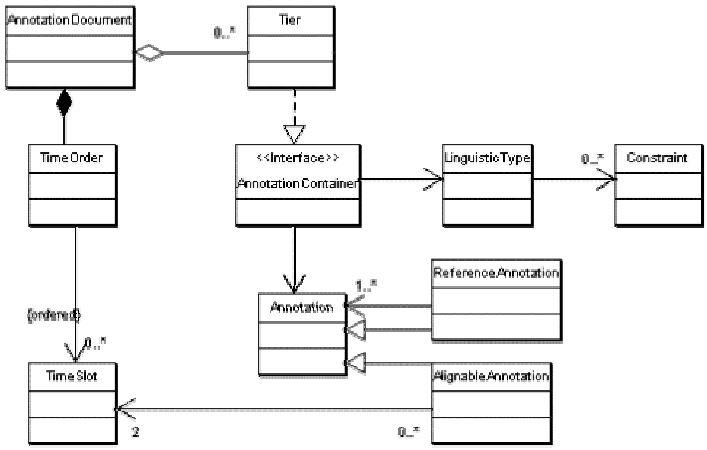


Figure 3 shows the core part of the UML chart that explains the Abstract Corpus Model. Tiers are seen as containers for annotations of certain linguistic type that share the same set of constraints. Annotations are split between those who are referring to periods in time, i.e. every annotation has a begin and end time and annotations can share such slots, and those that refer to one or several other annotations.

Many concrete annotation configurations were tested to see whether the model is powerful enough to handle all phenomena mentioned above. The following figure gives such a configuration, more elaborations can be found in Brugman (2001), Brugman (2003) and Bird (2001)

8 ELAN = EUDICO Linguistic Annotator; http://www.mpi.nl/tools

9 CHAT = Format used in the CHILDES project; http://childes.psy.cmu.edu

10 SHOEBOX is a program frequently used by field linguists; http://www.sil.org/computing/shoebox

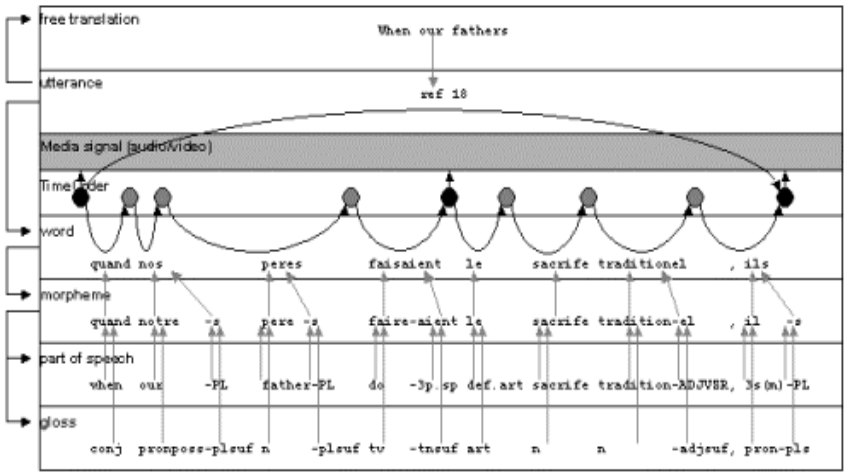


Figure 4 shows a typical complex annotation configuration where utterances and words are linked to time slots. All other annotations refer to ordered annotations. The annotations on the word, morpheme, POS and gloss tiers are part of an hierarchical system.

3.2 ELAN Annotation Format (EAF)

Given the concepts of the ACM and the ways they are associated, it was straightforward to design a format to deal with persistence and implement it using XML. The EAF is based on an open XML-schema which defines that each annotation is treated as an atomic entry referring to another annotation or to time slots. This way of formulation guarantees that parallel independent and therefore only partially overlapping streams can be represented without problems and that changes can be integrated easily.

The format is open to whether the annotations belonging to different annotations tiers are represented in one or several files. We assume that whenever a set of annotation tiers is created by one researcher or a closely collaborating team it makes sense to represent them in one file. However, as soon as several persons work independently from each other a stand-off annotation is the natural choice. It is essential that the full complexity of annotation structures as described above can be represented in the case of a stand-off as well. The extensive use of ID/IDREFs should solve this aspect. The general requirements of stand-off annotations can be found in Thompson (1997).

Several different types of research tasks were carried out using ACM and EAF as the basis. In the DOBES (Documentation of Endangered Languages) project the linguistic analysis work based on sound and video material is represented in EAF. Its obliged tiers are an orthographic or phonemic transcription and a translation into a major language. For some material a deep linguistic analysis has to be added allowing later generations to reconstruct the language. The most complex system is Advanced Glossing Drude (2002) defining 24 tiers to describe morphology and syntax. Similar annotation work is carried out by many researchers of the MPI.

Sign language studies are characterized by very complex multi-tier annotations since the movements of all articulators have to be described and analyzed in terms of their contribution to the standard linguistic layers. Such studies have been carried out within the ECHO project by 3 European SL communities for a comparative study of sign languages (Brugman et al 2004) and by a new research group bringing together signers from very different cultures.

Much work has been done in the area of gesture analysis and the relation of gestures to speech utterances. Several cross-cultural studies were carried out including annotations from the articulator to the interpretation level. Also here complex annotations covering many tiers, various types of time overlapping and cross-references can be found (Brugman et al 2002).

3.3 Access and Interoperability

As mentioned above we have to understand that user groups differ in the way how they work with such richly annotated corpora. XML is an excellent basis to serve to generate the different usage types. For those who prefer using standard web-browsers to exploit annotated media we are currently testing different ways for web-presentations. One option is to generate SMIL formatted representations that produce media streams with synchronized sub-titles, another way is to generate HTML versions where annotations can be clicked on to invoke the appropriate media fragments. Starting from the XML representation it is not difficult to generate other representational forms with the help of style sheets. The difficulty is mostly to find an appropriate general layout for presenting complex annotations with the help of HTML. Because of the extensive use of ID/IDREFs it is less trivial to do style sheet based transformations that maintain the full structural complexity of annotation documents.

Since XML is now a widely agreed and powerful enough basis for structuring and tagging complex text documents, it is perfect to transform all other formats to XML. Import modules and converters were created even for structured WORD documents that are still frequently used by field linguists. However, often the differences in the underlying data models create problems for the transformation step. TRANSCRIBER[11] for example has the notion of events

11 TRANSCRIBER is a program used for audio transcription;
http://www.etca.fr/CTA/gip/Projets/Transcriber/

and its annotations are marked by just the begin time which is also the end time of the previous annotation. An interpretation step beyond XML is necessary before the TRANSCRIBER created XML file can be transformed into for example the EAF format. The emergence of XML has reduced the number of different formats and obviously it helped to stimulate a world-wide discussion about and convergence of suitable annotation formats, which will result in a unification. Currently, efforts are taken within ISO TC37/SC4 in this direction.

4. Conclusions

Many institutions still prefer to take a relational database instantiation as the core for their holding. We have described the reasons for choosing XML files as basis in both cases where we are confronted with more complex documents that are increasingly often to be seen as objects in a distributed Internet scenario. We also indicated that we primarily see advantages in the fact that experienced users can exploit the files directly. This is of particular relevance in the Semantic Web era where we assume that intelligent agents will find their way through a domain of related and complex structured documents which ideally will be associated with schemas that refer to data category registries and ontological repositories. Using relational databases as core would always mean to introduce a web-service that exports the database contents. Therefore, we see relational databases as special containers that include optimized representations for specific purposes such for implementing fast searching.

We have shown that XML plays a fundamental role in our archiving work. It is at the center of our representations of complex information about the archive organization and its content and it helps to easily generate different types of presentation formats. We expect that XML will foster the international unification and help us to increase interoperability. However, we have also shown that (1) it is important to design a proper data model before designing an XML-based representation format and that (2) it is a good container for defining the structural elements, but that (3) it does not solve the semantic and interpretation problems. Here data category repositories also applying XML and RDF-based repositories with relational information will emerge.

References

P. Wittenburg, W. Peters, D. Broeder (2002a), Metadata Proposals for Corpora and Lexica. LREC 2002 Conference. Las Palma, Mai

P. Wittenburg, D. Broeder (2002b), Management of Language Resources with Metadata. Workshop on International Standards of Terminology and Language Resources Management. Las Palmas, Mai.

H. Brugman, P. Wittenburg (2001), The application of annotation models for the construction of databases and tools. IRCS Workshop on Linguistic Databases, University of Pennsylvania.

H. Brugman (2003), Annotated Recordings and Texts in the DOBES project. EMELD Workshop, East Michigan University.

S. Bird, M. Liberman (2001), A formal framework for linguistic annotation. Speech Communication 33 (1,2), pp 23-60

H.S. Thompson (1997), Towards a Base Architecture for Spoken Language Transcript{s,tion}; www.ltg.ed.ac.uk/~ht/rhodes.html

S. Drude (2002), Advanced Glossing – a language documentation format and its implementation with Shoebox. Int. Workshop on Resources and Tools in Field Linguistics, LREC 2002

H. Brugman, O. Crasborn, A. Russel (to appear), Collaborative Annotation of Sign Language Data with Peer-to-Peer Technology, submitted for LREC2004

H. Brugman, P. Wittenburg, St. Levinson, S. Kita (2002), Multimodal Annotations in Gesture and Sign Language Studies. LREC 2002 Conference. Las Palma, Mai

Transcribing and annotating spoken language with EXMARaLDA

Thomas Schmidt

Sonderforschungsbereich 538 'Mehrsprachigkeit'
University of Hamburg, Max Brauer-Allee 60, D-22765 Hamburg
thomas.schmidt@uni-hamburg.de

Abstract

This paper describes EXMARaLDA, an XML-based framework for the construction, dissemination and analysis of corpora of spoken language transcriptions. Departing from a prototypical example of a "partitur" (musical score) transcription, the EXMARaLDA "single timeline, multiple tiers" data model and format is presented alongside with the EXMARaLDA Partitur-Editor, a tool for inputting and visualizing such data. This is followed by a discussion of the interaction of EXMARaLDA with other frameworks and tools that work with similar data models. Finally, this paper presents an extension of the "single timeline, multiple tiers" data model and describes its application within the EXMARaLDA system.

Background

The EXtensible MARkup Language for Discourse Annotation (EXMARaLDA) is being developed at the 'SFB Mehrsprachigkeit' (Research Centre on Multilingualism) in Hamburg as the core architectural component of a database of multilingual spoken discourse. This database is intended as a platform for exchanging, archiving and analyzing the transcription data that the different SFB projects work with. The theoretical backgrounds and research goals of the projects differ greatly: they range from phonetic analyses of child language over studies of the acquisition of syntax in a generative framework to discourse analyses in a functional-pragmatic context. As a result of this diversity in research interests, the transcription systems, data formats and tools currently in use are also very dissimilar: for instance, one project works with a relational database of phonetically transcribed utterances whereas others use the syncWriter software (for a brief overview, see Bernsen et al., 2002) for creating orthographic multi-modal transcriptions in partitur notation.

This theoretical and technical diversity being an obvious obstacle in data exchange, the main challenge in the development of EXMARaLDA lies in the construction of a modeling framework that enables linguists to express their different models of spoken language on a common structural basis. Departing from such a data model, it should become possible to develop a set of interoperable data formats and tools that make the construction and exchange of richly annotated spoken language corpora easier.

A simple data model for multi-layered transcriptions

Partitur Transcriptions

Four of the fourteen projects at the SFB transcribe multiparty discourse according to the HIAT conventions (Ehlich, 1992). HIAT uses the so called partitur (musical score) notation in order to visualize temporal sequence and simultaneity between the utterances of different speakers, between different modalities (verbal and nonverbal behavior) and between segmental and nonsegmental (prosodic) phenomena. As the following figure shows, further analytic information – like an utterance-based transcription and a phonetic annotation – can also be integrated into the partitur:

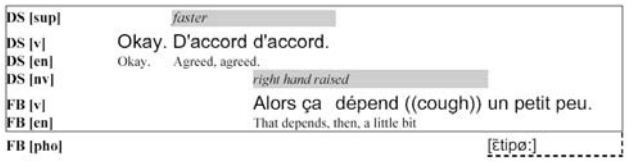


Figure 1: A partitur transcription

Structural relations in a partitur

As figure 1 illustrates, partiturs can be used to visualize a number of structural relations between entities of spoken language: The subdivision of a partitur into several tiers reflects an assignment of entities to different *speakers* (*DS* and *FB* in the example) and to different annotation *categories*. The categories in turn can be grouped into three different *types*:

- The actual *transcription* of verbal behavior (*v*-tiers above) which is used as the temporal point of reference for all other entities, i.e. every other entity is related to the verbal material by aligning the corresponding symbolic descriptions with an appropriate position in the transcription tiers. This is only possible because every symbolic description in a transcription tier is segmentable into smaller units, and because the sequence of these units corresponds to a temporal ordering of the entities (words, word fragments, phonemes, etc.) they describe.
- Like the transcriptions, *descriptions* of non-verbal behavior (the *nv*-tier above) relate to events that are independent of events in other tiers. In contrast to transcriptions, however, descriptions are atomic units that cannot further subdivided.
- *Annotations* (the *sup*-, *en*- and *pho*-tiers above) describe additional features (prosody, translations etc.) of verbal behavior that are not captured in the transcription tiers. As they are thus always related to verbal material, annotations, unlike transcriptions and descriptions, are *not* independent entities.

Lastly, the relation of entities in different tiers of the partitur can be thought of as the reference to *a common timeline*: simultaneous events or entity/feature pairs are placed at the same horizontal position, and the left-to-right direction within a tier or across tiers corresponds to temporal sequence.

The following figure, which represents the structure of the example above, sketches the "single timeline, multiple tiers" data model that results from these considerations:

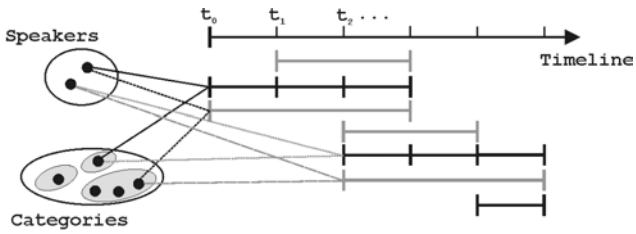


Figure 2: "Single timeline, multiple tiers" data model

In the EXMARaLDA system, data of this kind can be represented in an XML file that conforms to the *basic-transcription* document type definition:[1]

```
<!ELEMENT basic-transcription (head, basic-body)>
<!ELEMENT head (speakertable)>
<!ELEMENT speakertable (speaker*)>
<!ELEMENT speaker EMPTY>
<!ATTLIST speaker
    id ID #REQUIRED>
<!ELEMENT basic-body (common-timeline, tier*)>
<!ELEMENT common-timeline (tli*)>
<!ELEMENT tli EMPTY>
<!ATTLIST tli
    id ID #REQUIRED
    time CDATA #IMPLIED>
<!ELEMENT tier (event*)>
<!ATTLIST tier
    speaker IDREF #IMPLIED
    category CDATA #REQUIRED
    type (t | d | a ) #REQUIRED>
<!ELEMENT event (#PCDATA)>
<!ATTLIST event
    start IDREF #REQUIRED
    end IDREF #REQUIRED>
```

Figure 3: EXMARaLDA basic-transcription DTD

According to this DTD, the smallest entities of a partitur transcription are represented as *events*. Events refer to the items (*tli*) on a *common-timeline* via the *start* and *end* attributes and are grouped into *tiers* where each tier is assigned a *speaker*, a *category* and one of the three *types* described above. Additionally, each item of the timeline can be assigned an absolute time value by means of an optional *time* attribute and thus point to a position in the transcribed audio or video recording.

An editor for partitur transcriptions

For creating and editing *basic-transcriptions*, EXMARaLDA provides the Partitur-Editor[2], a tool written in Java that visualizes the data as a partitur and allows interactive editing of tiers (adding, deleting, reordering), events (adding, deleting, splitting, merging and a number of other specialized functions), the timeline and the speaker table. In contrast to most other transcription tools currently under development, the Partitur-Editor offers extensive support for the use of different font types, styles and sizes and thus enables the user to typographically distinguish different types of information:

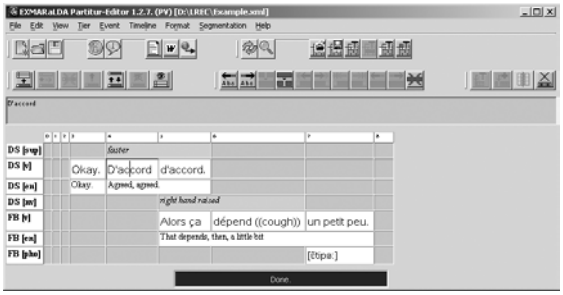


Figure 4: EXMARaLDA Partitur-Editor

Beside these essential editing functionalities, the Partitur-Editor also provides some basic support for audio playback given that the timeline items of the basic-transcriptions have been assigned absolute time values (see above). Furthermore, as a truly Unicode-enabled tool, the editor comprises a customizable virtual keyboard for input of symbols that are not available via the system keyboard:

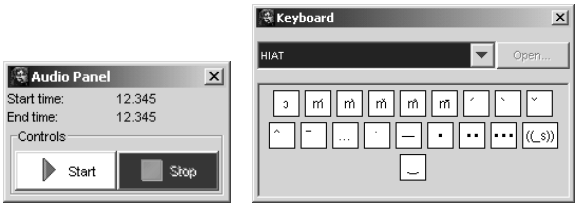


Figure 5: Audio playback panel and virtual keyboard

Especially for the analysis of multi-modal behavior, it is often desirable to link parts of the transcribed material to portions of the underlying recording or to image data. To this end, the Partitur-Editor contains a link panel in which single events can be associated with external audio, video or image files.

Rendering partiturs on screen and paper

The kind of discourse analysis that uses HIAT as its transcription system (and, in fact, a great number of other linguistic methodologies) relies heavily on a qualitative interpretation of written transcripts. While being able to display a partitur on the screen may be sufficient for some purposes, the possibility of having a readable *printout* of the transcription remains a vital requirement from these users' point of view. Paradoxically, many transcription tools currently under development attach very little or no importance to that aspect, either because their focus is entirely on computer-based (and hence "screen-centered") analysis methods, or because of the alleged ease with

1 For the sake of simplicity, some details of the DTD have been left out.

2 The Partitur-Editor is freeware and can be downloaded from http://www.exmaralda.org.

which XML-encoded data can be transformed into presentation formats via XSL-Stylesheets.
However, the non-hierarchical nature of the "single timeline, multiple tiers" data model makes the use of stylesheet transformations a non-trivial matter, and the interlinear structures in a partitur are a notoriously difficult area for common rendering software like browsers and word processors, see (Bow et al., 2003)[3].
The EXMARaLDA system therefore provides an extensive functionality for transforming a *basic-transcription* into a printable form. It allows the user to parameterize the formatting properties (font types and styles, borders, numbering etc.) of a partitur and to specify page formats (size and margins) and, based on these parameters, calculates a line-wrapped version of a partitur that can then be output directly to a printer or imported into a word-processor as an RTF file or into a browser as an HTML file:

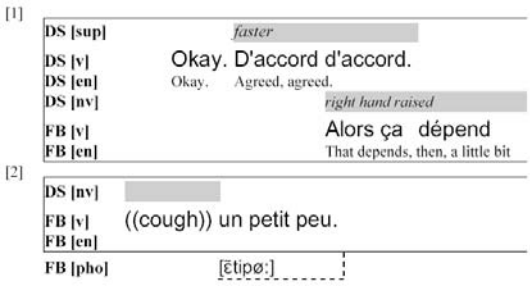


Figure 6: A wrapped partitur

Exchange with other tools and formats

The "single timeline, multiple tiers" data model is not unique to the EXMARaLDA system. Among other tools or systems that work with a comparable data model are:

- the TASX-Annotator developed at the University of Bielefeld (see the contribution from Milde to this workshop),
- the Praat software developed by Paul Boersma (http://www.fon.hum.uva.nl/praat/),
- the EUDICO Linguistic Annotator (ELAN), developed at the Max-Planck-Institute for Psycholinguistics in Nijmegen (Brugmann, 2003).

Although the structures of these data models are not a hundred percent identical to that of an EXMARaLDA *basic-transcription*, they are sufficiently similar to make a fully automatic conversion in both directions possible. Such import and export filters are an integral part of the EXMARaLDA system, and they have proven especially valuable because the EXMARaLDA Partitur-Editor on the one hand and the TASX-Annotator, Praat and ELAN on the other hand address partly complementary needs: whereas the Partitur-Editor is superior to the other tools with respect to parameterizability of the visualization and output functionalities, it offers only minimal support for the interaction of digitized recordings with the transcription process. The TASX-Annotator, Praat and ELAN, on the other hand, provide precisely that kind of support, and an interoperability between the tools therefore has a great synergetic value from the users' point of view.
Thus, one project at the SFB uses Praat for a rough segmentation and transcription of the digitized audio recordings and then imports these data into the EXMARaLDA Partitur-Editor for a refinement of the transcription, an addition of analytical annotations and the print-out of transcripts (see Schmidt, 2003b). Similarly, other users make their primary transcriptions of video recordings in TASX or ELAN and then transfer these data to EXMARaLDA for further processing and output.

Legacy data and other data

The different SFB projects have large amounts[4] of legacy data which, in their original form, have very limited potential for exchange and reuse. One major part of the database project therefore consists in the conversion of these legacy data into the EXMARaLDA format.
On the one hand, this pertains to partitur transcriptions created with the software tools HIAT-DOS and syncWriter. As the data models of these tools are geared towards visual display rather than logical structure, a fully automated conversion is not possible. The corresponding conversion methods therefore map parts of the data structure to an EXMARaLDA *basic-transcription* and thus reduce the cost of manual post-editing as far as possible.
Many legacy data, on the other hand, have a much simpler structure than a partitur transcription: they have been created with simple text editors or as RDB-tables and follow the concept of a simple line-for-line transcription where each line contains exactly one utterance and temporal overlaps are marked with an appropriate bracketing:

DS: Okay.
DS: D'accord <d'accord.>1>
FB: <Alors >1> ça depend ((cough)) un petit peu.

Figure 7: A line-for-line transcription

These kind of data can be imported into EXMARaLDA via the "Simple EXMARaLDA" interface, an import filter that operates on plain text files and maps the structure of a line-for-line transcription onto the "single timeline, multiple tiers" data model. Conversion in this case is fully automatic, i.e. it requires no manual post-editing.

Beyond the single timeline

The "single timeline, multiple tiers" data model has proven to be useful because it is powerful enough to express a lot of structural relations in spoken language while at the same time being sufficiently simple and intuitive to form the basis of user-friendly and efficient implementations.
However, it is beyond doubt that the transcription and annotation of spoken language can lead to data structures that are not covered by this simple data model. Again, EXMARaLDA is not unique in acknowledging this limitation and recognizing the need for more powerful mechanisms: The approach taken by TASX is the so called "TASX level 2" data model where events can either refer to the common timeline or to events in other tiers thus

3 What (Bow et al., 2003) discuss under the notion of "interlinear text" is conceptionally slightly different from my notion of a "partitur" (cf. Schmidt, 2003a). The difficulties in rendering, however, are very similar for both concepts.

4 More than 1000 hours of transcribed spoken language, or over 2500 single transcriptions, as a rough estimate.

allowing the construction of hierarchical annotation structures (Milde/Gut, 2003). The EUDICO Abstract Corpus Model (Brugman, 2003) also goes beyond strictly time-based structures by allowing symbolic subdivisions and symbolic associations of entities in different tiers.
The EXMARaLDA approach is different from these approaches because it does not abandon the timeline metaphor altogether, but instead extends it to a more complex construction: a *segmented-transcription*. In contrast to an *basic-transcription*, the timeline of an EXMARaLDA *segmented-transcription* can have bifurcations. This is a need that arises as soon as a *temporally* structured transcription is segmented into *linguistic* units. For instance, in the above example, a segmentation of the verbal tiers into words will lead to a data structure in which the temporal relation between some words of different speakers cannot be determined:

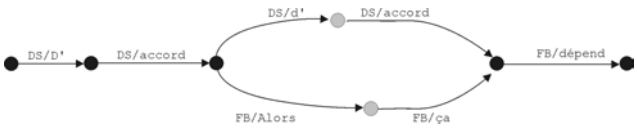


Figure 8: A bifurcated timeline

The segmentation of transcribed material into linguistic units being the most important prerequisite for most analytical processes (like additional annotation or search), the EXMARaLDA segmented-transcription thus provides an extension of the "single timeline, multiple tiers" data model that is crucial in obtaining truly computer-suitable representations of spoken language.

Segmenting with finite state machines

EXMARaLDA does not provide a tool for inputting and editing *segmented-transcriptions* directly. Instead, *segmented-transcriptions* are automatically generated from *basic-transcriptions* on the basis of the punctuation in the transcription tiers. This punctuation is interpreted as an implicit markup, i.e. as symbols marking the beginning and the end of linguistic units, and transformed into explicit XML-markup by means of a finite state machine (FSM):

Basic-Transcription

Event	**Event**	**Event**
Okay.	D'accord	d'accord.

FSM for HIAT

Segmented-Transcription

Segment Chain									
Utterance		**Utterance**							
W	**P**	**W**	**P**	**W**	**P**	**W**	**P**	**W**	**P**
Okay	.	D	'	accord		d	'	accord	.

Figure 9: Segmentation

It is important to note that this process of segmenting transcriptions of spoken language is different from what a sentencizer or tokenizer does for texts of written language: as punctuation use in transcriptions is always done according to a specific transcription convention, the types and positions of punctuation symbols are totally predictable – for instance, there will be no ambiguity about whether a particular punctuation mark must be interpreted as an utterance terminator or as a word terminator.
As segmentation is thus dependent on the transcription system used, the algorithm must be parameterizable. This is achieved by using different finite state machines for different transcription systems. At the time of writing, three different FSMs – one for HIAT, one for DIDA (Klein/Schütte, 2001) and one for CHAT (MacWhinney, 2000) – are integrated into the EXMARaLDA system. As the FSMs are also formulated as XML files, this mechanism can be easily adapted or modified to meet the conventions of other transcription systems. Furthermore, encoding the segmentation algorithm as an XML file also ensures that it is largely independent of the rest of the software and could thus be readily integrated into other environments.
Besides being the basis for the transformation of a *basic-transcription* into a *segmented-transcription*, the finite state machines can also serve as a means for controlling the validity of transcriptions. A failure of the segmentation algorithm will tell the transcriber that somewhere in the transcription a certain symbol does not conform to the underlying conventions. In order to be able to easily identify such errors, the EXMARaLDA Partitur-Editor provides a segmentation panel that allows the user to go through the transcription step-by-step and find places where the segmentation algorithm runs into an problem:

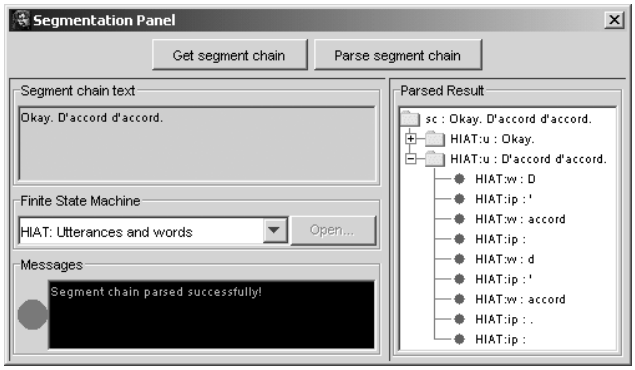


Figure 10: Segmentation Panel

Making use of segmentation

The *basic-transcription* data model is used for input and visualization of transcriptions in partitur notation. After a *basic-transcription* has been successfully transformed into a *segmented-transcription*, further processing methods become possible:
Building on the segmentation into utterances (or equivalent units), a visualization in a line-for-line notation as in figure 7 can be calculated. The same transformation can also be used as the basis for a conversion of time-based EXMARaLDA data into formats that follow a more hierarchically structured data model (e.g. the TEI format for the transcription of speech).
Similarly, the segmentation can be used for a calculation of alphabetic word lists. A much used feature of the Partitur-Editor is the option to output such word lists in HTML and link them to a HTML output of a partitur transcript thus enabling a quick word search in context:

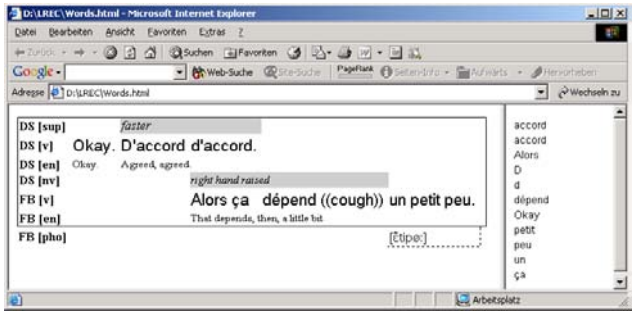


Figure 11: Word list and linked transcript

Last but not least, a segmentation of transcriptions will be the prerequisite for any elaborate analysis method like detailed annotation, querying etc. EXMARaLDA does not yet provide a generic tool for performing such analyses, but first tests with a small corpus of *segmented-transcriptions* and a standard RDB-system (Schmidt, 2003b) have shown that the potential of EXMARaLDA data clearly exceeds the possibilities of older tools and formats in that respect.

Time-based data models and XML technology

The "single timeline, multiple tiers" data model is arguably the most simple and intuitive one for describing the kind of transcription data that discourse analysts and many other linguists work with. As shown above, its conceptual structure can straightforwardly be represented physically in an XML file, and the resulting corpora can thus profit from many of the benefits that XML as a wide-spread standard offers – the data become exchangeable between different tools and platforms, the full use of Unicode becomes a matter of course (an aspect that is of obvious relevance especially to multilingual data), and XML also lends itself to making some processing methods for transcription data (e.g. the segmentation by FSMs, see above) parameterizable in a software-independent way.

By and large, however, the role of XML in the EXMARaLDA system remains limited to that of a standardized storage format, and the full potential of XML technology can thus not be exploited. The reason for that is that most of XML technology is closely tied to a primarily hierarchical data model, whereas we do not see hierarchies as the primary structural relation in our kind of data. For the time-based data model(s) that result from this consideration, XML technology therefore does not always constitute the optimal framework, for instance:

- As DTDs and Schemata primarily serve the purpose of checking the well-formedness and validity of tree-structures, they will not be sufficient to describe and verify XML-encoded instances of the "single timeline, multiple tiers" data model. For instance, the DTD in figure 3 will not check whether the start and end points of a given event follow one another in the timeline or whether two events in one tier do not overlap.
- As XSL is mainly a language for transforming source trees into result trees, it is not well suited to calculate visualizations whose primary structure is not hierarchical. Partitur transcriptions are a case in point for such visualizations.
- As query languages like XQuery are also designed around a hierarchical data model – they are efficient in navigating and querying tree structures – their use for querying multi-layered data like those presented here is also questionable.

Two extreme conclusions could be drawn from this dilemma: One would be that time-based data models, since they cannot tap the full potential of XML technology, are not the most useful approach to the goal of constructing richly annotated language corpora. It is this view that underlies (Carletta et al., 2000)'s criticism of the (time-based) annotation graph formalism.[5] The other would be that XML, since its associated technologies do not adequately support the intuitive time-based data model, should not be considered a relevant factor in the construction of such corpora.

EXMARaLDA follows an approach that lies in-between these two extremes. On the one hand, it relies strongly on XML as a standardized storage format and, insofar as it structures time-aligned entities into a system of tiers, also partly accommodates the prototypical hierarchical XML data model.[6] On the other hand, it does not view XML technology as the paramount criterion that decides on the choice of data structures and processing methods for a spoken language corpus – because spoken language is very rich in non-hierarchical structures (at least according to the models that many transcription systems work with), prioritizing hierarchical relations over other relations would mean an artificial restriction hindering an efficient processing of transcription data rather than facilitating it.

A drawback resulting from the latter point is the lack of an industry-supported framework or API that would help developers in the construction of tools for input and analysis of time-based data in the same way that XML technology aids the processing of hierarchically structured data. In that respect, interoperability between existing tools and formats for time-based data becomes a very important requirement. The possibilities of data exchange between TASX, EXMARaLDA, Praat and ELAN, as described above, are already a major step in this direction. Further harmonizing the respective formats and, in particular, a common approach to an extension of the "single timeline, multiple tiers" data model would seem like a good next step.

Outlook

By the time of writing, EXMARaLDA can be said to have left the stage of a prototype system. The tools and formats are used in the every-day-work of linguists both inside and outside the SFB for research and teaching.[7] Beside maintenance and improvement of the existing tools, further work will focus on corpus management and corpus analysis. Two tools addressing these aspects are currently under development: One is the EXMARaLDA Corpus manager (Wörner, forthcoming), a tool which supports the creation

5 "We propose that since most XML use privileges element hierarchies by making hierarchical structures easy and fast to navigate, element hierarchies should be used to represent the most important relations in an XML data set." (Carletta et al., 2000)

6 In that respect, it is a less powerful but also an easier-to-handle data model than the more general annotation graph data model.

7 Judging by download figures for the Partitur-Editor, the total number of EXMARaLDA users should be somewhere between 500 and 1000.

and management of corpus meta-data and the linking of this information to the actual transcriptions. The other is a concordance tool designed to help with the search and analysis of transcribed and annotated phenomena in an EXMARaLDA corpus.

The ongoing conversion of legacy data into the EXMARaLDA format and the use of EXMARaLDA tools for the creation of new data should meanwhile lead to a number of "real-life" sized multilingual corpora of spoken language that will allow an insight into possibilities for further development and optimization of the framework.

References

Bernsen, N. / Dybkjaer, L. / Kolodnytsky, M. (2002). An Interface for Annotating Natural Interactivity. In Kuppevelt, J. v. / R. W. Smith (eds.). Current and New Directions in Discourse and Dialogue. Dordrecht: Kluwer.

Bow, C./Hughes, B./Bird, S. (2003). Towards a General Model of Interlinear Text. In Proceedings of the E-Meld Workshop on Digitizing and Annotating Texts and Field Recordings. Lensing: LSA Institute, Michigan State University.

Brugman, Hennie (2003). Annotated Recordings and Texts in the DoBeS Project. In Proceedings of the E-Meld Workshop on Digitizing and Annotating Texts and Field Recordings. Lensing: LSA Institute, Michigan State University.

Carletta, J./Isard, A./McKelvie, D. (2000): Linguistic Data Processing For Everyman. In Proceedings of the Workshop on Web-Based Language Documentation and Description. Philadelphia: Institute for Research in Cognitive Science, University of Pennsylvania.

Ehlich, K. (1992). HIAT - a Transcription System for Discourse Data. In Edwards, J. / Lampert, M. (eds.). Talking Data – Transcription and Coding in Discourse Research. Hillsdale: Erlbaum, 123-148.

Klein, W. / Schütte, W. (2001): Transkriptionsrichtlinien für die Eingabe in DIDA. Mannheim: Institut für Deutsche Sprache (IDS).

MacWhinney, Brian (2000): The CHILDES project: tools for analyzing talk. Mahwah, NJ u.a. : Lawrence Erlbaum.

Milde, J.T./Gut, U. (2003): Multimodale bilinguale Korpora gesprochener Sprache: Korpuserstellung, -analyse und -dissemination in der TASX-Umgebung. In In Seewald-Heeg (ed.). Sprachtechnologie für die multilinguale Kommunikation - Textproduktion, Recherche, Übersetzung, Lokalisierung (Beiträge der GLDV-Frühjahrstagung 2003). Sankt Augustin: gardez!, 406-420.

Schmidt, T. (2003a). Visualising Linguistic Annotation as Interlinear Text. In Working Papers in Multilingualism, Series B (46). Hamburg.

Schmidt, T. (2003b). Korpus „Skandinavische Semikommunikation" - ein mehrsprachiges Diskurskorpus auf XML-Basis. In Seewald-Heeg (ed.). Sprachtechnologie für die multilinguale Kommunikation - Textproduktion, Recherche, Übersetzung, Lokalisierung (Beiträge der GLDV-Frühjahrstagung 2003). Sankt Augustin: gardez!, 421-427.

Wörner, K. (forthcoming): CoMa – A corpus manager for EXMARaLDA data. To appear in Working Papers in Multilingualism, Series B. Hamburg.