An XML-based representation format for syntactically annotated corpora

Andreas Mengel, Wolfgang Lezius

IMS, University of Stuttgart
Azenbergstr.12, D-70174 Stuttgart
{mengel,lezius}@ims.uni-stuttgart.de

Abstract

This paper discusses a general approach to the description and encoding of linguistic corpora annotated with hierarchically structured syntactic information. A general format can be motivated by the variety and incompatibility of existing annotation formats. By using XML as a representation format the theoretical and technical problems encountered can be overcome.

Introduction

As there are various formats for the representation and storage of linguistic corpora, there are also a number of formats for the representation of syntactically annotated corpora or treebanks: Tipster (Grishman, 1998), Penn Treebank (Marcus et al., 1993), Susanne (Sampson, 1995), NeGra (Skut et al., 1998) and several formats for chunked corpora. This variety of formats complicates the access to syntactic data and thus contradicts the aim of creating standard resources only once and to enable easy exchange of data.

In this paper we propose an XML-based, theory-independent exchange format for syntactically annotated corpora (section 1). We show how to encode trees and graphs representing syntactic annotation (section 2), and we discuss the advantages of our approach (section 3).

XML-based exchange format

There are two aspects of current treebank formats which complicate the distribution of data:

1. Representation format: Instead of reusing existing representation formats more and more new formats have been developed. Yet, any new format requires the creation of special tools that support the maintenance of and access to the data (Mengel, 1999a).
2. Underlying theory: Encoding of information implies a model of the entities. In the case of linguistic corpora many resources are designed to fit the actual theory used for the description of the data. Thus, the format does not support the encoding of other theoretical descriptions.

The format proposed in this paper tries to overcome these restrictions by developing a theory-independent exchange format that works as an interface between the existing formats (cf. figure 1). The interface consists of two levels of interchange:

First of all, the existing formats should be exportable to the exchange format. As this just means to transform the syntactic structure of the data, this step is easy-going and straightforward. By using XML as the underlying format, the generated corpora can be displayed, queried etc. by several XML tools.

The more difficult aspect of the exchange is the import of the corpora encoded in the XML format. On the one hand, external tools and the existing formats can concentrate on the import of only one format and profit on a variety of XML parsing libraries. But on the other hand, one should keep in mind that a linguistic re-interpretation is still necessary due to the different underlying theories:

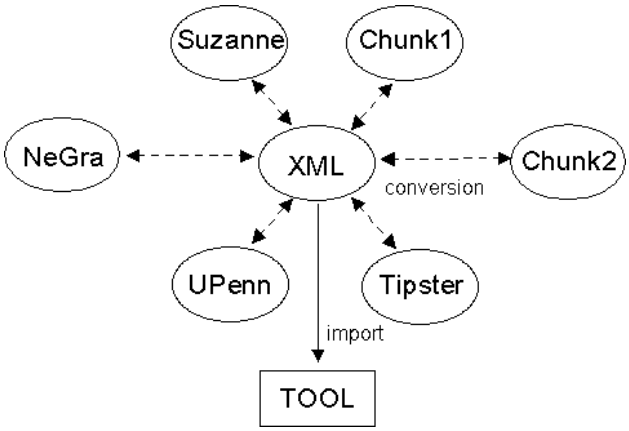


Figure 1: XML-based exchange format

For example, long-distance dependencies are encoded by traces in the Penn treebank, but by crossing edges in the NeGra annotation scheme. However, this problem is not relevant for external tools which do not depend on the underlying theory (e.g. Tiger search engine; König & Lezius, 2000).

XML-based syntax format

When designing an encoding format for syntactic annotations, two decisions have to be made: First of all, the representation format has to be chosen (annotation format). We decided to use XML which has been designed to allow maximum portability. Second, one has to choose the underlying theory (annotation scheme). Since we intend to support a broad range of existing formats, we have to generalize as much as necessary and to find a common denominator for the encoding of both trees and graphs representing syntactic annotation.

XML

XML (Extensible Markup Language, W3C, 1997a) is a descendant of SGML (Standardized General Markup Language) which has recently attracted much attention. The difference between SGML and XML is that XML has a stricter grammar and is thus easier to parse. The difference between XML and the well known HTML (Hypertext Markup Language, W3C, 1997b) is that XML is not restricted to a fixed number of tags and that XML allows the users to define their own structure and any number of tags necessary for the encoding task envisaged.

The growing number of applications that support XML for representation, manipulation, and display of data make information encoded in this data format more accessible and enhance the establishment of supported formats like the one proposed here. As XML does not make any assumptions about the entities to encode and their properties, XML can be considered theory-independent.

Encoding trees

Encoding syntactic information in XML might at a first look seem either trivial or like reinventing the wheel. In fact, SGML and the TEI guidelines (Sperber-McQueen & Burnard, 1994) explicitly encode structural aspects of texts. Consider the sentence "*The boy likes the girl*". In XML this could be encoded as:

```
<s>
  <np>
    <w word="The"/>
    <w word="boy"/>
  </np>
  <vp>
    <w word="likes"/>
    <np>
      <w word="the"/>
      <w word="girl"/>
    </np>
  </vp>
</s>
```

Figure 2: Simple encoding scheme

Yet, there are two restrictions in the use of XML or SGML that make the simple embedding structure of the example above inappropriate for the encoding of syntactic information.

First, the use of embedding structures for syntactic information restricts the description variety to one relation, the part-whole relation. Thus, by such a structure neither syntactic relations are encoded nor are trees represented, but a hierarchically arranged sequence of embedded segments only. Therefore, in this structure any higher order element (e.g. *sentence*) must embrace a chain of continuous sub-elements (e.g. *phrases* or *words*). Discontinuous constituents cannot be represented.

Second, as there is only one relation, with this kind of annotation, no means exists to label the relationship between higher order elements and their constituents. The default relation - though not made explicit - is the part-whole relation.

Consequently, we propose to use special elements for edges in syntactic annotation trees. This allows the explicit representation of both edges and their labels. In appendix A an example sentence is presented which has been annotated according to the NeGra scheme and converted to XML.

In our approach there are basically four XML elements which describe this type of graph: sentence elements <s>, non-terminal elements <n>, terminal elements <w>, and edge elements <edge>, used to link nodes (cf. figure 3). The sentence element indicates the beginning of a new sentence and its syntactic annotation. The inner nodes of a syntactic tree are represented by non-terminal elements, the outer nodes by terminal elements. The node features are realized as element attributes (e.g. syntactic category or POS). Additionally, we have added some attributes to encode a unique identifier value. For each edge element, an extra attribute is provided which has the ID value of the node linked to by the edge. The labelling of edges is realized by an attribute *label* of the edge elements.

```
<s id="s1" href="#id(n1_500)"/>

<n id="n1_500" cat="S">
  <edge id="edge1_1" href="#id(n1_501)"/>
  <edge id="edge1_2" href="#id(n1_502)"/>
</n>
<n id="n1_501" cat="NP">
  <edge id="edge1_3" href="#id(w1_0)"/>
  <edge id="edge1_4" href="#id(w1_1)"/>
</n>
<n id="n1_502" cat="VP">
  <edge id="edge1_5" href="#id(w1_2)"/>
  <edge id="edge1_6" href="#id(n1_503)"/>
</n>
<n id="n1_503" cat="NP">
  <edge id="edge1_7" href="#id(w1_3)"/>
  <edge id="edge1_8" href="#id(w1_4)"/>
</n>

<w id="w1_0" word="The"/>
<w id="w1_1" word="boy"/>
<w id="w1_2" word="likes"/>
<w id="w1_3" word="the"/>
<w id="w1_4" word="girl"/>
```

Figure 3: The proposed encoding scheme

Encoding graphs

For the encoding of syntactic phenomena trees are sometimes inappropriate. For example, different linguistic descriptions have been proposed for the encoding of long-distance dependencies (e.g. extraposed relative clauses): Whereas the Penn Treebank makes use of traces, the NeGra annotation scheme allows crossing edges (cf. NeGra annotation scheme, Skut et al., 1998). In this case, sentence structure is represented by means of *directed acyclic graphs* (DAGs). In the format proposed in this paper, crossing edges are actually encoded by edge elements. In appendix B, there is an example sentence containing crossing edges.

In the NeGra annotation format, there are also so-called *secondary edges* representing semantic information (e.g. coreference). This can only be represented by structure-sharing (two nodes link to the same third node). In this case, it is useful to distinguish between these two types of edges which can be realized by introducing an additional edge attribute (cf. Appendix C).

Since the proposed format is based on the encoding of DAGs, even dependency graphs (cf. Hajic, 1999) can be encoded. This means to allow structure-sharing and links from terminal nodes.

Discussion of the XML-based format

The encoding in this format is as general as possible, i.e., existing formats can easily be represented and the encoding proposal can be further expanded by additional structure types and feature annotations. And, as it is independent from linguistic theory, it is applicable as a general representation format.

Compared with the tree-encoding recommendations of the TEI (Sperberg-McQueen & Burnard, 1994), this format additionally allows labelled edges which is an important improvement for expressing modifier relations (cf. NeGra, Skut et al., 1998) and dependency graphs (Hajic, 1999).

Another important advantage of this proposal is the use of XML as the encoding formalism. XML markup is highly expandable which means that completely different annotation levels can easily be combined (e.g. speech and syntax, Mengel et al., 2000). In contrast to SGML, different levels of description can also be distributed over different files.

Many XML tools are already available, and new ones are being implemented; this makes access to this formalism easier: Validating parsers allow to control the input and the output of XML conversion routines. There are also visualization tools which provide an overview of the sentence structure. The XML-support of browsers in combination with style sheets will help browsing XML-annotated corpora. Finally, XML search engines and search engines optimized to the format proposed here (Lezius, 1999) are being developed which enable the search on hierarchically structured documents.

Conclusions

We have presented a general representation format for encoding syntactically annotated corpora. It is based on XML which guarantees maximum portability and expandability. By using DAGs for encoding the syntactic structure, many existing annotation formats are supported. Thus, the proposed format is applicable as a theory independent format. An online conversion routine for some formats and more details about the proposed format can be found on the Web at http://www.ims.uni-stuttgart.de/projekte/TIGER/xml.

Acknowledgements

The work described in this paper has been supported by the "Deutsche Forschungsgemeinschaft e.V." (Project TIGER, http://www.coli.uni-sb.de/cl/projects/tiger) and EC Telematics Project LE4-8370 (MATE, http://mate.nis.sdu.dk). We would like to thank Stefanie Dipper and Arne Fitschen (IMS) for comments on earlier versions of this paper.

References

Goldfarb, C.F. (1990). The SGML Handbook. Clarendon Press.

Grishman, R. (1998). TIPSTER Text Architecture Design. Internal Report. New York University.

Hajic, J. (1998). Building a Syntacticly Annotated Corpus: The Prague Dependency Treebank In: Issues of Valency and Meaning, pp. 106-132. Karolinum, Prague.

König, E. & Lezius, W. (2000). A Description Language for Syntactically Annotated Corpora. Submitted for publication.

Lezius, W. (1999). The TIGER project. Online-information. http://www.ims.uni-stuttgart.de/projekte/TIGER/.

Marcus, M.P., Santorini, B. & Marcinkiewicz, M.A. (1993). Building a large annotated corpus of English: the Penn Treebank. Computational Linguistics, vol. 19.

Mengel, A., Dybkjaer, L., Garrido, J.M., Heid, U., Klein, M., Pirrelli, V., Poesio, M., Quazza, S., Schiffrin, A. & Soria, C. (2000). MATE Dialogue Annotation Guidelines (M-DAG). MATE Deliverable D2.1. http://www.ims.uni-stuttgart.de/projekte/mate/mdag.

Mengel, A. (1999a). Die integrierte Repräsentation linguistischer Daten. In: Gippert, J. (ed.): Multilinguale Corpora - Codierung, Strukturierung und Analyse. Enigma corporation, Prague.

Mengel, A. & Heid, U. (1999b). Query Language for Access to Speech Corpora. Forum Acusticum, Berlin. (ASA, EAA, DEGA).

Sampson, G. (1995). English for the Computer - The SUSANNE Corpus and Analytic Scheme. Clarendon Press.

Skut, W., Brants, T., Krenn, B. & Uszkoreit, H. (1998). A Linguistically Interpreted Corpus of German Newspaper Text. ESSLI-1998, Workshop on Recent Advances in Corpus Annotation.

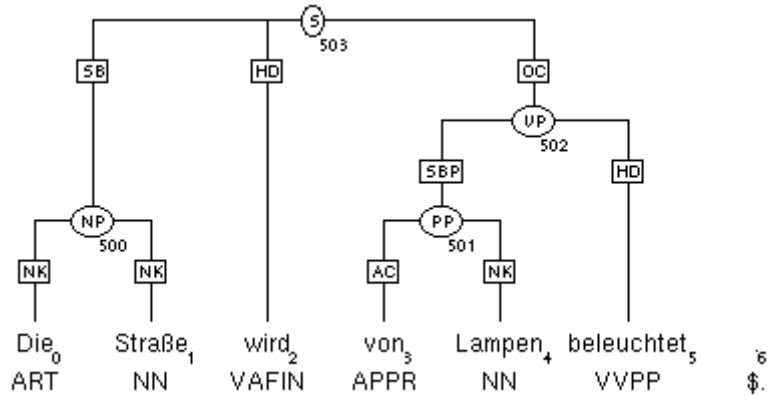
Sperberg-McQueen, C.M. & Burnard, L. (1994). TEI Guidelines for Electronic Text Encoding and Interchange (P3). Chicago and Oxford: ACH/ACL/ALLC Text Encoding Initiative.

W3C (1997a). Extensible Markup Language. http://www.w3.org/XML.

W3C (1997b). HyperText Markup Language. http://www.w3.org/MarkUp.

Appendix A

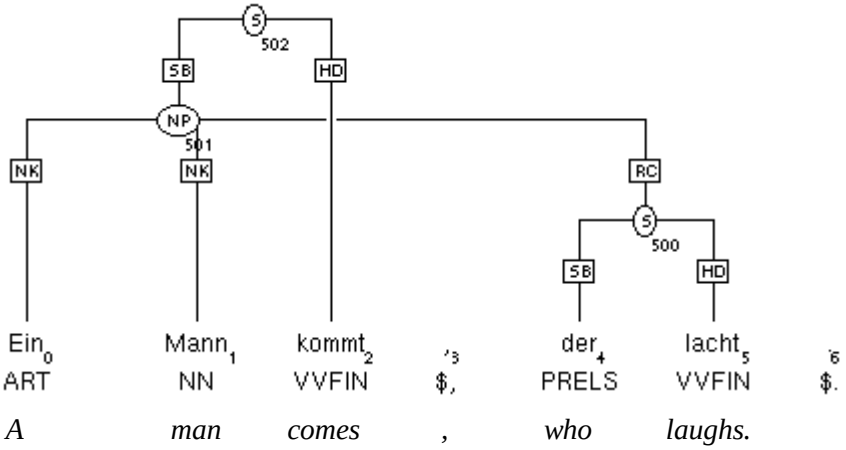
The following sentences have been annotated according to the NeGra annotation scheme (Skut et al., 1998) and automatically converted into XML.
Note that there are labelled edges in the NeGra annotation scheme (printed bold type in the XML encoding of the present example sentence).
Abbreviations of the edge labels in the example sentence: AC = adpositional case marker, HD = head, NK = noun kernel, OC = clausal object, SB = subject, SBP = passivised subject (PP). In the NeGra annotation scheme auxiliaries usually embed the non-finite verb and its arguments as a clausal object (OC).




```
<s id="s1" href="#id(n1_503)"/>
<n id="n1_500" cat="NP">
  <edge id="edge1_1" label="NK" href="#id(w1_0)"/>
  <edge id="edge1_2" label="NK" href="#id(w1_1)"/>
</n>
<n id="n1_501" cat="PP">
  <edge id="edge1_3" label="AC" href="#id(w1_3)"/>
  <edge id="edge1_4" label="NK" href="#id(w1_4)"/>
</n>
<n id="n1_502" cat="VP">
  <edge id="edge1_5" label="HD" href="#id(w1_5)"/>
  <edge id="edge1_6" label="SBP" href="#id(n1_501)"/>
</n>
<n id="n1_503" cat="S">
  <edge id="edge1_7" label="HD" href="#id(w1_2)"/>
  <edge id="edge1_8" label="SB" href="#id(n1_500)"/>
  <edge id="edge1_9" label="OC" href="#id(n1_502)"/>
</n>
<w id="w1_0" word="Die" pos="ART"/>
<w id="w1_1" word="Stra&#xdf;e" pos="NN"/>
<w id="w1_2" word="wird" pos="VAFIN"/>
<w id="w1_3" word="von" pos="APPR"/>
<w id="w1_4" word="Lampen" pos="NN"/>
<w id="w1_5" word="beleuchtet" pos="VVPP"/>
<w id="w1_6" word="." pos="$."/>
```

Appendix B

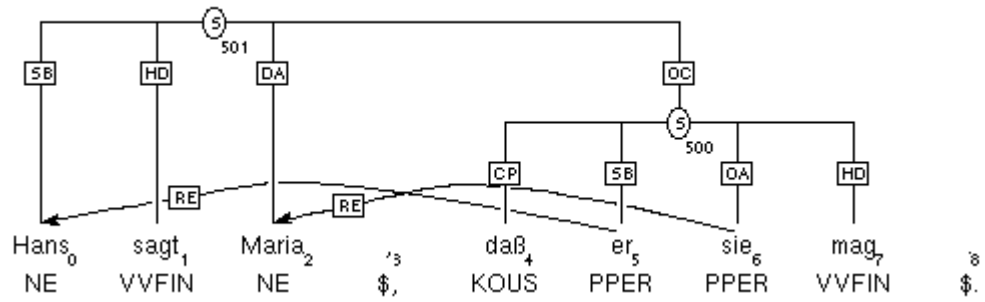
The following example illustrates the phenomenon of an extraposed relative clause which is expressed by crossing edges (the crossing edges are printed bold type in the XML encoding).




```
<s id="s2" href="#id(n1_502)"/>
<n id="n1_500" cat="S">
  <edge id="edge1_1" label="SB" href="#id(w1_4)"/>
  <edge id="edge1_2" label="HD" href="#id(w1_5)"/>
</n>
<n id="n1_501" cat="NP">
  <edge id="edge1_3" label="NK" href="#id(w1_0)"/>
  <edge id="edge1_4" label="NK" href="#id(w1_1)"/>
  <edge id="edge1_5" label="RC" href="#id(n1_500)"/>
</n>
<n id="n1_502" cat="S">
  <edge id="edge1_6" label="HD" href="#id(w1_2)"/>
  <edge id="edge1_7" label="SB" href="#id(n1_501)"/>
</n>
<w id="w1_0" word="Ein" pos="ART"/>
<w id="w1_1" word="Mann" pos="NN"/>
<w id="w1_2" word="kommt" pos="VVFIN"/>
<w id="w1_3" word="," pos="$,">
<w id="w1_4" word="der" pos="PRELS"/>
<w id="w1_5" word="lacht" pos="VVFIN"/>
<w id="w1_6" word="." pos="$."/>
```

Appendix C

The following sentence comprises a coreference information (abbreviation: RE = repeated element, corefering anapher). The edges representing semantic information are printed bold type in the XML encoding.



Hans tells Maria , that he her likes.

```
<s id="s3" href="#id(n1_501)"/>
<n id="n1_500" cat="S">
  <edge id="edge1_1" label="CP" href="#id(w1_4)"/>
  <edge id="edge1_2" label="SB" href="#id(w1_5)"/>
  <edge id="edge1_3" label="OA" href="#id(w1_6)"/>
  <edge id="edge1_4" label="HD" href="#id(w1_7)"/>
</n>
<n id="n1_501" cat="S">
  <edge id="edge1_5" label="SB" href="#id(w1_0)"/>
  <edge id="edge1_6" label="HD" href="#id(w1_1)"/>
  <edge id="edge1_7" label="DA" href="#id(w1_2)"/>
  <edge id="edge1_8" label="OC" href="#id(n1_500)"/>
</n>
<w id="w1_0" word="Hans" pos="NE"/>
<w id="w1_1" word="sagt" pos="VVFIN"/>
<w id="w1_2" word="Maria" pos="NE"/>
<w id="w1_3" word="," pos="$,"/>
<w id="w1_4" word="da&#xdf;" pos="KOUS"/>
<w id="w1_5" word="er" pos="PPER">
  <edge id="edge1_9" label="RE" type="semantic" href="#id(w1_0)"/>
</w>
<w id="w1_6" word="sie" pos="PPER">
  <edge id="edge1_10" label="RE" type="semantic" href="#id(w1_2)"/>
</w>
<w id="w1_7" word="mag" pos="VVFIN"/>
<w id="w1_8" word="." pos="$."/>
```