XML-based Stand-off Representation and Exploitation of Multi-Level Linguistic Annotation

Stefanie Dipper
Institute of Linguistics
University of Potsdam
dipper@ling.uni-potsdam.de

Abstract: This paper deals with the representation of multi-level linguistic annotations. It proposes an XML-based, generic stand-off architecture and presents an example instantiation. Application scenarios that profit from this architecture are sketched out.[1]

In recent years, corpus linguistics has become more and more important to a broad community, including people working in theoretical, applied and computational linguistics. To many of them, speech and text corpora represent a rich source of data and phenomena, forming the basis of their research. Benefit from such data is even more important if the data is annotated by suitable information, allowing for fast and effective retrieval of relevant data. Whereas corpora of the first generation featured part-of-speech and syntactic annotations (e.g. PennTreebank [MSM93], TIGER corpus [BDE$^+$04]), the focus has now switched to properties beyond the (morpho-)syntactic level. Recent corpora are annotated by semantic information (PropBank [KP02], FrameNet [JPB$^+$03], SALSA [EKPP03]), pragmatic information (Penn Discourse TreeBank [MPJW04], RST Discourse Treebank [CMO03], Potsdam Commentary Corpus [Ste04]), and dialogue structure (Switchboard SWBD-DAMSL [JSB97]).

Annotations often have to be carried out manually — reliable (semi-)automatic tools exist only for the annotation of part of speech and syntax, and are restricted to well-researched languages like English or German. Moreover, hand-annotated training material is a prerequisite for the development of automatic tools. As a consequence, corpora and annotations ought to be reusable so that a large community can profit from the data.

To this end, various standardization efforts have been launched. Standardization of linguistic data concerns (see, e.g., [Sch05]):

(i) The physical data structure: here, XML has become the widely-recognized standard format.

(ii) The logical data structure: i.e., the data models that are used to model the phenomena and their properties (e.g. hierarchical structures like trees or graphs for syntax annotations

[1] The research reported in this paper was jointly financed by the German Research Foundation (DFG, SFB632) and the Federal Ministry of Education and Research (BMBF grant no. 03WKH22). Many thanks go to my colleagues, especially Michael Götze, for helpful discussions of the topics addressed in this paper.

vs. time-aligned tiers for speech and dialogue annotations). Examples of data models are annotation graphs [BL01] and the NITE Object Model [CKO$^+$03b].

(iii) Content: in several initiatives, XML applications for specific linguistic annotations have been developed. For instance, TEI[2] ("Text Encoding Initiative", [SB94]) defines highly-detailed DTDs for encoding all kinds of bibliographic and other information; XCES[3] ("XML-based Corpus Encoding Standard") provides DTDs for the annotation of chunks, alignment, etc.

More recently, however, it has been recognized that these standardized DTDs often do not meet application-specific needs. Hence, abstract, generic XML formats have been proposed that allow for the *formal* integration of application-specific annotations [IR01]. For the *conceptual* integration of specific annotations, so-called data category repositories as well as linguistic ontologies have been developed. They define reference categories, with precise semantics and examples, that specific annotation tags ought to be mapped to (see, e.g., DOLCE[4], "Descriptive Ontology for Linguistic and Cognitive Engineering").

This papers deals with the formal integration of specific annotations. It first addresses the subject of stand-off architecture (sec. 1). We then propose an XML-based representation of linguistic annotation and present an example application (instantiation) in some detail (sec. 2). We also sketch out some application scenarios that profit from such a flexible architecture (sec. 3) and address related approaches (sec. 4).

1 Stand-off Architecture

As early as in the mid-nineties, the topic of "stand-off annotation" has been discussed (see, e.g., [TM97]). This term describes the situation where primary data (e.g., the source text) and annotations of this data are stored in separate files. Stand-off annotation might seem problematic, because there is no immediate connection between the text and its annotation; hence, whenever the source text is modified, extra care has to be taken to synchronize its annotation. Similarly, human inspection of the data becomes cumbersome.

On the other hand, however, stand-off annotation has the great advantage of leaving the source text untouched. It thus allows for annotating text that cannot be modified for whatever reasons, e.g., because it is a text available on the Internet. Moreover, whereas XML as such does not easily account for overlapping segments and conflicting hierarchies,[5] they can be marked in a natural way in stand-off annotation: by distributing annotations over different files. That is, not only is the source text separated from its annotations, but individual annotations are separated from each other as well. This way, annotations at different levels can be created and modified independently of each other. Finally, competing, alternative annotations can even be represented, e.g. variants of part-of-speech annotations that are output of different tools.

[2] `http://www.tei-c.org/`

[3] `http://www.cs.vassar.edu/XCES/`

[4] `http://www.loa-cnr.it/DOLCE.html`

[5] Different methods have been proposed to accommodate conflicting markup into XML. We will come back to them below.

One of the first proposals for stand-off annotation of linguistic corpora is [DBD^{+}98]. An ISO working group is currently developing the stand-off based LAF[6] ("Linguistic Annotation Framework" [IRdlC03]). Some recent corpora like the ANC ("American National Corpus" [RI04]) are encoded in stand-off architecture. In our approach presented in this paper, we also subscribe to the principles of stand-off annotation.

2 A Generic XML Format

Our format defines generic XML elements like `<mark>` (markable), `<feat>` (feature), and `<struct>` (structure), which indicate which data type the annotation conforms to. We assume that primary data is stored in a file that optionally specifies a header, followed by a tag `<body>`, which contains the source text.

Annotations are stored in separate files; they may refer to the source text or to other annotations. These relations are encoded by means of XLinks and XPointers. We distinguish three different types of annotations: markables, structures, and features.

(i) Markables: `<mark>` tags specify text positions or spans of text (or spans of other markables) that can be annotated by linguistic information. For instance, `<mark>` tags might indicate tokens by specifying ranges of the source text, cf. fig. 1.

(ii) Structures: `<struct>` tags are special types of markables. Similar to `<mark>` tags, they specify objects that then can serve as anchors for annotations. Whereas `<mark>` tags define simple types of anchors (flat spans of text or markables), a `<struct>` tag represents a complex anchor involving relations between arbitrarily many markables (including `<struct>` elements). Relations (`<rel>`) can be further specified by an attribute `type`, e.g. as undirected or directed (= pointers). Put differently, a `<structList>` specifies a complete tree or graph, which consists of single tree fragments specified by the `<struct>` tags, cf. fig. 1.

(iii) Features: `<feat>` tags specify information annotated to markables or structures, which are referred to by `xlink` attributes. The type of information (e.g., "part of speech") is encoded by an attribute `type`, cf. fig. 2. For instance, the information encoded by the first `<feat>` in fig. 2 can be paraphrased as follows: Take the token that is defined by the tag `<mark>` with the ID attribute `id="tok_1"` and assign the part of speech "ART" (article) to that token.

We intend to adopt the idea of [CKO^{+}03a] by assuming that admissible feature values (such as "NN", normal/common noun, or "NE", named entity) may be complex types and are organized in a type hierarchy. For instance, "NN" and "NE" might be subtypes of the more general type "N", noun. `<feat>` tags then point to some type in the hierarchy (which is stored separately), thus specifying the value of the annotated property, cf. fig. 3.[7]

[6] ISO Technical TC37/SC4, `http://www.tc37sc4.org`

[7] Type hierarchies have to be defined by the user or they may be derived from annotation schemes that incorporate hierarchies, cf. the schemes used by the annotation tool MMAX. In case no hierarchy is defined, the features will be organized in a flat list. The stand-off architecture allows the user to experiment with different hierarchies.

Further examples of annotations are sketched out below. They illustrate that annotations may stem from different sources (see the attribute `source`) and encode various types of information.

Categorial annotation (anchored to constituents)

```
<header sfb_id="rabin1.const_cat" type="categories" source="TIGERcorpus"/>
<featList xml:base="rabin1.const.xml">
  <feat xlink:href="#syn_1" value="PN"/> <!--proper noun-->
  <feat xlink:href="#syn_2" value="PP"/> <!--prepos. phrase-->
  ...
```

Coreference annotation, marking coreferential expressions such as pronouns (referred to `xlink:href` attributes) and their antecedents (identified by `target` attributes)

```
<header sfb_id="rabin1.coref" type="coreference" source="MMAXcoref"/>
<featList>
  <feat xlink:href="rabin1.tok.xml#tok_19" (sein)
        target="rabin1.const.xml#syn_9" (Der Rabin-Attentäter Jigal Amir)
        value="identity"/>
  ...
```

Document structure: headers, paragraphs, lists, etc. (anchored to markables that refer to tokens)

```
<header sfb_id="rabin1.div" type="divisions"/>
  <markList xmlns:xlink="http://www.w3.org/1999/xlink"
        xml:base="rabin1.tok.xml">
   <mark id="div_1" xlink:href="#xpointer(id('tok_1')/range-to(id('tok_390')"/>
   <mark id="div_2" xlink:href="#xpointer(id('tok_1')/range-to(id('tok_89')"/>
   ...
```

```
<header sfb_id="rabin1.div_docstr" type="documentStructure"/>
  <featList xml:base="rabin1.div.xml">
   <feat xlink:href="#div_1" value="sec"> <!--section-->
   <feat xlink:href="#div_2" value="par"> <!--paragraph-->
   ...
```

Time alignment: temporal information, specifying starting point and duration (anchored to tokens)[8]

```
<header sfb_id="rabin1.tok_talign" type="timeAlignment" source="UNKNOWN"/>
<featList xml:base="rabin1.tok.xml">
  <feat xlink:href="#tok_1" value="time-range(0,0.2)"/><!--Der-->
  <feat xlink:href="#tok_2" value="time-range(0.2,0.9)"/><!--Rabin-Attent.-->
  ...
```

Annotation set: stand-off files that belong together and form one corpus are marked by `<struct>` elements. In the example, text, word-level and syntax annotations are grouped

[8] In canonical time-aligned annotation, the single annotations refer to time points and spans. In our example, it is the other way round: time-alignment is considered as some sort of annotation. However, our basic units, which are text positions in the examples presented above, may as well consist of points in time rather than points in text.

by individual `<struct>` elements. `<feat>` elements can be used to specify properties of these groups (such as "primary data", "syntax"). In a similar way, groups of annotation sets can be defined to form (sub)corpora.

```
<header sfb_id="rabin1.anno" type="annotations"/>
<structList xmlns:xlink="http://www.w3.org/1999/xlink">
    <struct id="anno_1">
      <rel id="rel_1" type="file" xlink:href="rabin1.text.xml"/>
    </struct>
    <struct id="anno_2">
      <rel id="rel_2" type="file" xlink:href="rabin1.tok.xml"/>
      <rel id="rel_3" type="file" xlink:href="rabin1.tok_pos.xml"/>
      <rel id="rel_4" type="file" xlink:href="rabin1.tok_morph.xml"/>
    </struct>
    <struct id="anno_3">
      <rel id="rel_5" type="file" xlink:href="rabin1.const.xml"/>
      <rel id="rel_6" type="file" xlink:href="rabin1.const_cat.xml"/>
    </struct>
    ...
```

3 Application Scenarios

As argued above, stand-off representation has many advantages. For further processing, however, such extensive use of xlinks can be considered problematic for performance. Similarly, our format is certainly not suitable for human inspection and debugging.

Inline Versions We therefore envisage the following scenario: Depending on the current application, an inline version is pre-computed, which consists of only those layers that are highly relevant to the application in question. For instance, token and sentence boundaries, word forms, and part-of-speech annotation offer enough information for many applications (and represent exactly the kind of data which traditional corpora used to comprise). Such a condensed, inline version of our above example is displayed below. The attribute `<source>` records the layers that the annotations have been taken from: token boundaries and word forms stem from the file with ID `rabin1.tok`; sentence boundaries are encoded by the file with ID `rabin1.const_cat`; finally, part-of-speech annotation is encoded in `rabin1.pos`.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE sfb632_inline SYSTEM "sfb632_inline.dtd">
<sfb632_inline version="1.0">
   <header sfb_id="rabin1_inline" type="inline"
      source="rabin1.tok,rabin1.pos,rabin1.const_cat"/>
   <inline_trad>
     <s id="s_1">
       <tok id="tok_1" pos="ART">Der</tok>
       <tok id="tok_2" pos="NN">Rabin-Attentäter</tok>
       <tok id="tok_3" pos="NE">Jigal</tok>
      ...
     </s>
     ...
</sfb632_inline>
```

Now, for sophisticated applications such as automatic text summarization, more layers are needed and, hence, added to the inline representation, e.g., document structure and sentence relevance,[9] as in the following example.

```
<sfb632_inline version="1.0">
  <header sfb_id="rabin1_inline" type="inline"
      source="rabin1_tok,rabin1_pos,rabin1.const_cat,rabin1.tok_docstr,
      rabin1_4grel,rabin1_wfrel,rabin1_porterrel"/>
  <inline_summar>
   <article>
    <p id="p_1">
     <s id="s_1" weight_4g="0.2184" weight_wf="0.1753" weight_porter="0.1861">
      <chunk id="ch_1" type="NP" ref_type="definite">
       <tok id="tok_1" pos="ART">Der</tok>
       <tok id="tok_2" pos="NN">Rabin-Attentäter</tok>
       <tok id="tok_3" pos="NE">Jigal</tok>
       <tok id="tok_4" pos="NE">Amir</tok>
      </chunk>
      <tok id="tok_5" pos="VAFIN">hat</tok>
      ...
</sfb632_inline>
```

A summarizing tool that operates on such input probably might also profit from the other annotations. For instance, sentences of the input that are recognized by the summarizer as being highly relevant will be included in the summary. If such a sentence contains a pronoun whose referent (antecedent) has not been extracted, the summarizer would start a fixing procedure: by making use of the stand-off coreference annotation, it would determine the pronoun's referent and replace the pronoun accordingly. That is, the summarizer takes just as much information into consideration as currently necessary.

Similar use cases are linguistic applications like the investigation of certain phenomena (e.g., information structure). Here, the relevant factors are often not known in advance and differ from phenomenon to phenomenon. Hence, it seems sensible to start with a restricted set of "canonical" information and then include more and more annotations in the investigation. This way, the impact of the individual linguistic features (i.e., annotation types) can be observed more directly and easily than by looking at complex annotations simultaneously.

Bringing Stand-off Annotations Together Obviously, the more (complex) annotation levels we include in the inline version, the more likely we are to induce conflicting hierarchies. An example of such a conflict involves overlapping syntactic and prosodic chunks (represented in ill-formed XML):

```
<chunk id="ch_1"> syntactic content ...
  <pros id="pros_1"> prosodic/syntactic content ...
</chunk>
  prosodic content ...  </pros>
```

[9] In the example: relevances computed on the base of 4grams, word forms, and porter stems, respectively. The summarizing tool might, e.g., compute the average value of these relevances, or else make use of the relevance types in different ways during processing.

Different strategies have been proposed to deal with such conflicts, namely: (cf. [SB94, ch.31], [BBG$^+$95])

Milestones: empty elements mark the start and end point of that nesting element which is considered less important

```
<chunk id="ch_1"> syntactic content  ...
  <pros_start id="pros_1a"/> prosodic/syntactic content ...
</chunk>
  prosodic content ...  <pros_end id="pros_1b"/>
```

Fragmentation: the less important nesting element is broken into smaller units
+ **Virtual joins**: the tag `<join>` is added to the fragmentation representation to explicitely mark elements that belong together

```
<chunk id="ch_1"> syntactic content  ...
  <pros id="pros_1a"> prosodic/syntactic content ...</pros>
</chunk>
  <pros id="pros_1b"> prosodic content ...  </pros>
<join targets="pros_1a pros_1b" result="pros"/>
```

Alternatively, `next` and `prev` attributes can be added to the fragments.

```
<chunk id="ch_1"> syntactic content  ...
  <pros id="pros_1a" next="pros_1b"> prosodic/syntactic content ...</pros>
</chunk>
  <pros id="pros_1b" prev="pros_1a"> prosodic content ...  </pros>
```

Redundant encoding: multiply-annotated text (*prosodic/syntactic content*) is duplicated, resulting in multiple files (each of which is inline). Obviously, this is not an option for an efficient exploitation of multiple annotations.

```
<chunk id="ch_1"> syntactic content  ...
prosodic/syntactic content ...
</chunk>
```

```
<pros id="pros_1"> prosodic/syntactic content
prosodic content ...
</pros>
```

Today, there is a very limited number of tools that support creating inline versions of stand-off annotations, e.g. LT XML[10]. However, LT XML does not allow for conflicting hierarchies. [WGSL05] present a Prolog-based tool of merging two conflicting XML hierarchies by replacing one of the annotations by milestones or fragments. They rely on redundant encoding as the input to their tool. In some preliminary experiments, we successfully applied this tool to a Prolog representation of our sample data, which we created by XLS stylesheets.

[10] `http://www.ltg.ed.ac.uk/software/xml/index.html`

4 Related Approaches

Most recent work in corpus annotation relies on XML and many projects now make use of stand-off annotations, e.g. ANC [RI04], FrameNet [JPB$^+$03], PropBank [KP02]. Most of these projects, however, focus on one or two types of annotation only, such as (morpho-) syntax, or syntax combined with semantics. Semantic annotations like predicate-argument relations typically result in overlapping hierarchies (see, e.g. [KP02], [EKPP03]).

Few of the projects address annotations at more than two levels. One such example is the MULI project, which used multi-level annotations for the investigation of information structure, comprising a syntactic, discourse and prosodic level [BBHS$^+$04]. Similarly to MULI, we deal with multi-level, heterogeneous annotation. In contrast to them, however, we use a generic XML format to represent the data.

Such generic formats have been proposed as interchange formats, e.g., in LAF (Linguistic Annotation Framework [IRdlC03]), AIF (ATLAS Interchange Format [LFGP02]) or TIGER/SALSA XML [EP04]. The exact form of LAF is still under discussion (on the way to becoming an ISO standard), AIF is available in a beta version[11]. TIGER/SALSA XML has been applied successfully in the SALSA project to encode frames (semantic roles) [EKPP03].

Whereas these formats might in principle host heterogeneous annotation, projects dealing with such data (like MULI) tend to develop task-specific formats. In a way, our work presents a "proof of concept" of such generic formats in the domain of multi-level, heterogeneous annotation. Our standard format currently integrates data annotated by part of speech, morphology and lemma, syntax, rhetorical relations, anaphoric relations, and information structure [Ste04]. Some data is also annotated by phonetic/phonological information (breaks, pitch-range, tones, etc.).[12]

5 Conclusion and Outlook

We presented a generic, stand-off XML representation that allows for flexible integration of various kinds of linguistic information. Annotations from different tools and formats can be mapped to our generic standard format. The stand-off architecture supports the representation of conflicting hierarchies and competing annotations.

Exploitation of the data can proceed in a similarly flexible way: in the first run, data to be considered is restricted to often-used, canonical information; additional data is only added upon request. This architecture supports the use and reuse of multiply-annotated data in

[11] http://www.nist.gov/speech/atlas/develop/aif.html

[12] The annotations are created by means of different tools: EXMARaLDA (http://www.rrz.uni-hamburg.de/exmaralda/), annotate (http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/annotate.html), MMAX (http://mmax.eml-research.de/), and RST Tool (http://www.wagsoft.com/RSTTool/). The export format of these tools is mapped to our standard format. For manual inspection of the data at multiple levels, our project has developed the tool ANNIS, which provides viewing and searching facilities [DGSW04] (http://www.sfb632.uni-potsdam.de/annis/).

many different applications, by offering inline versions of the data that are tailored to the application-specific needs.

As one of our next steps, we plan to design a representation of ambiguities and underspecification that fits into our general architecture. A quick solution would be to simply represent all possible interpretations by stand-off files. However, this solution is neither efficiently computable nor does it explicitely represent the actual facts: namely the fact that parts of the data is shared by all files while other parts of it diverge.

References

[BBG$^+$95] David Barnard, Lou Burnard, Jean-Pierre Gaspart, Lynne A. Price, C. M. Sperberg-McQueen, and Giovanni Batista Varile. Hierarchical Encoding of Text: Technical Problems and SGML Solutions. *Text Encoding Initiative: Background and Context. Special Issue of Computers and the Humanities*, 29(211–231), 1995.

[BBHS$^+$04] Stefan Baumann, Caren Brinckmann, Silvia Hansen-Schirra, Geert-Jan Kruijff, Ivana Kruijff-Korbayová, Stella Neumann, and Elke Teich. Multi-dimensional annotation of linguistic corpora for investigating information structure. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, Boston, Massachusetts, 2004.

[BDE$^+$04] Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. TIGER: Linguistic Interpretation of a German Corpus. *Research on Language and Computation*, 2(4):597–620, 2004.

[BL01] Steven Bird and Mark Liberman. A formal framework for linguistic annotation. *Speech Communication*, 33(1,2):23–60, 2001.

[CKO$^+$03a] Jean Carletta, Jonathan Kilgour, Tim O'Donnell, Stefan Evert, and Holger Voormann. The NITE Object Model Library for Handling Structured Linguistic Annotation on Multimodal Data Sets. In *Proceedings of the EACL Workshop on Language Technology and the Semantic Web (NLPXML-2003)*, 2003.

[CKO$^+$03b] Jean Carletta, Jonathan Kilgour, Timothy O'Donnell, Stefan Evert, and Holger Voormann. The NITE Object Model Library for Handling Structured Linguistic Annotation on Multimodal Data Sets. In *Proceedings of the EACL Workshop on Language Technology and the Semantic Web (3rd Workshop on NLP and XML, NLPXML-2003)*, 2003.

[CMO03] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory. In Jan van Kuppevelt and Ronnie Smith, editors, *Current and New Directions in Discourse and Dialogue*. Kluwer Academic Publishers, 2003.

[DBD$^+$98] Laila Dybkjær, Niels Ole Bernsen, Hans Dybkjær, David McKelvie, and Andreas Mengel. The MATE Markup Framework. MATE Deliverable D1.2, 1998.

[DGSW04] Stefanie Dipper, Michael Götze, Manfred Stede, and Tillmann Wegst. ANNIS: A Linguistic Database for Exploring Information Structure. In Shinichiro Ishihara, Michaela Schmitz, and Anne Schwarz, editors, *Interdisciplinary Studies on Information Structure (ISIS)*, volume 1, pages 245–279. Universitätsverlag Potsdam, Potsdam, Germany, 2004.

[EKPP03] Katrin Erk, Andrea Kowalski, Sebastian Padó, and Manfred Pinkal. Towards a Resource for Lexical Semantics: A Large German Corpus with Extensive Semantic Annotation. In *Proceedings of ACL 2003*, Sapporo, Japan, 2003.

[EP04] Kathrin Erk and Sebastian Padó. A powerful and versatile XML Format for representing role-semantic annotation. In *Proceedings of LREC 2004*, Lisbon, Portugal, 2004.

[IR01] Nancy Ide and Laurent Romary. Standards for Language Resources. In *Proceedings of the IRCS Workshop on Linguistic Databases*, pages 141–149, Philapdelphia, 2001.

[IRdlC03] Nancy Ide, Laurent Romary, and Eric de la Clergerie. International Standard for a Linguistic Annotation Framework. In *Proceedings of HLT-NAACL'03 Workshop on The Software Engineering and Architecture of Language Technology*, 2003.

[JPB$^+$03] Christopher R. Johnson, Miriam R. L. Petruck, Collin F. Baker, Michael Ellsworth, Josef Ruppenhofer, and Charles J. Fillmore. FrameNet: Theory and Practice, 2003. Version 1.1.

[JSB97] Daniel Jurafsky, Elizabeth Shriberg, and Debra Biasca. Switchboard SWBD-DAMSL shallow-discourse-function annotation, 1997. Coders manual, draft 13. Technical Report 97-02, University of Colorado.

[KP02] Paul Kingsbury and Martha Palmer. From TreeBank to PropBank. In *Proceedings of LREC 2002*, Las Palmas,Spain, 2002.

[LFGP02] Christophe Laprun, Jonathan G. Fiscus, John Garofolo, and Sylvain Pajot. A practical introduction to ATLAS. In *Proceedings of LREC 2002*, Las Palmas,Spain, 2002.

[MPJW04] Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. The Penn Discourse TreeBank. In *Proceedings of LREC 2004*, Lisbon, Portugal, 2004.

[MSM93] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[RI04] Randi Reppen and Nancy Ide. The American National Corpus: Overall Goals and the First Release. *Journal of English Linguistics*, 32:105–113, 2004.

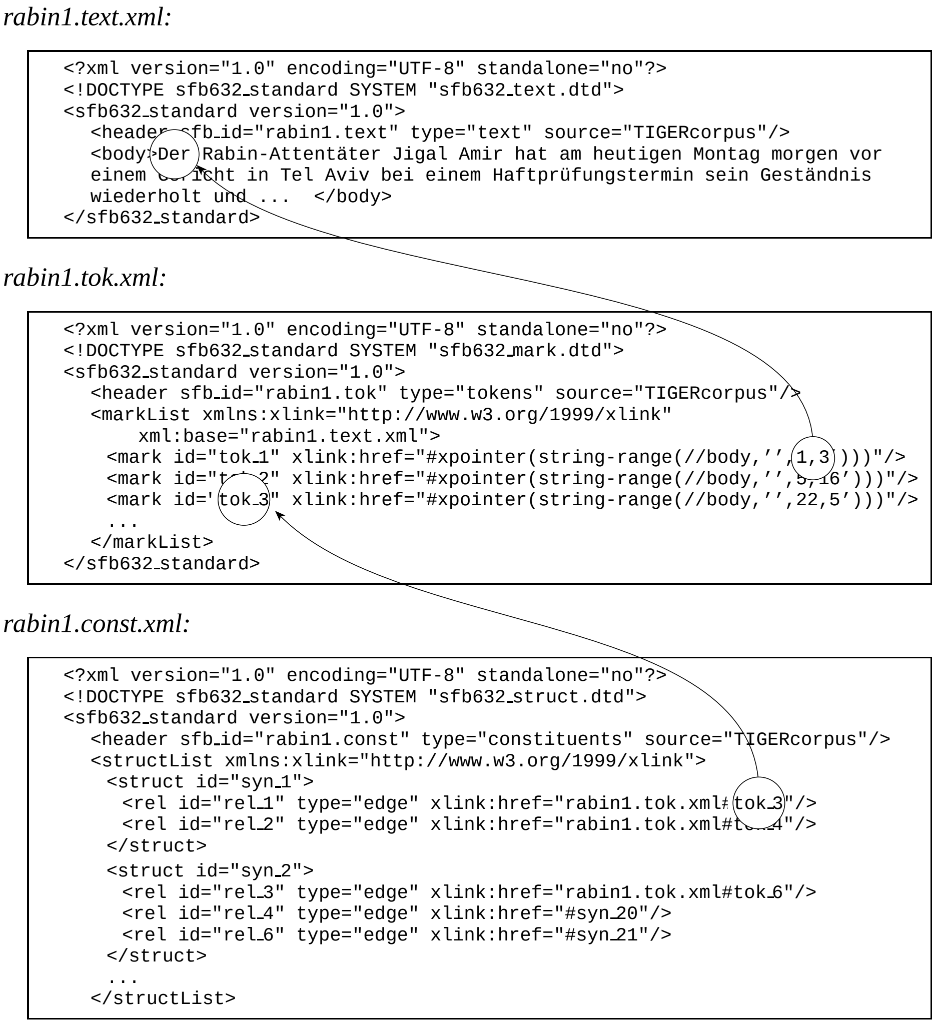
[SB94] C. M. Sperberg and Lou Bernard, editors. *Guidelines for Electronic Text Encoding and Interchange (TEI P3)*. Text Encoding Initiative, Chicago, Oxford, 1994.

[Sch05] Thomas Schmidt. EXMARaLDA und die Datenbank "Mehrsprachigkeit" — Konzepte und praktische Erfahrungen. In Stefanie Dipper, Michael Götze, and Manfred Stede, editors, *Heterogeneity in Focus: Creating and Using Linguistic Databases*, ISIS (Interdisciplinary Studies on Information Structure), Working Papers of the SFB 632, Universität Potsdam. Universitätsverlag Potsdam, 2005.

[Ste04] Manfred Stede. The Potsdam Commentary Corpus. In *Proceedings of the ACL Workshop on Discourse Annotation*, pages 96–102, Barcelona, 2004.

[TM97] Henry Thompson and David McKelvie. Hyperlink semantics for standoff markup of read-only documents. In *Proceedings of SGML Europe'97*, 1997. `http://www.ltg.ed.ac.uk/~ht/sgmleu97.html`.

[WGSL05] Andreas Witt, Daniela Goecke, Felix Sasaki, and Harald Lüngen. Unification of XML Documents with Concurrent Markup. *Literary and Linguistic Computing*, 20(1):103–116, 2005.



rabin1.text.xml:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE sfb632_standard SYSTEM "sfb632_text.dtd">
<sfb632_standard version="1.0">
  <header sfb_id="rabin1.text" type="text" source="TIGERcorpus"/>
  <body>Der Rabin-Attentäter Jigal Amir hat am heutigen Montag morgen vor
  einem Gericht in Tel Aviv bei einem Haftprüfungstermin sein Geständnis
  wiederholt und ...  </body>
</sfb632_standard>
```

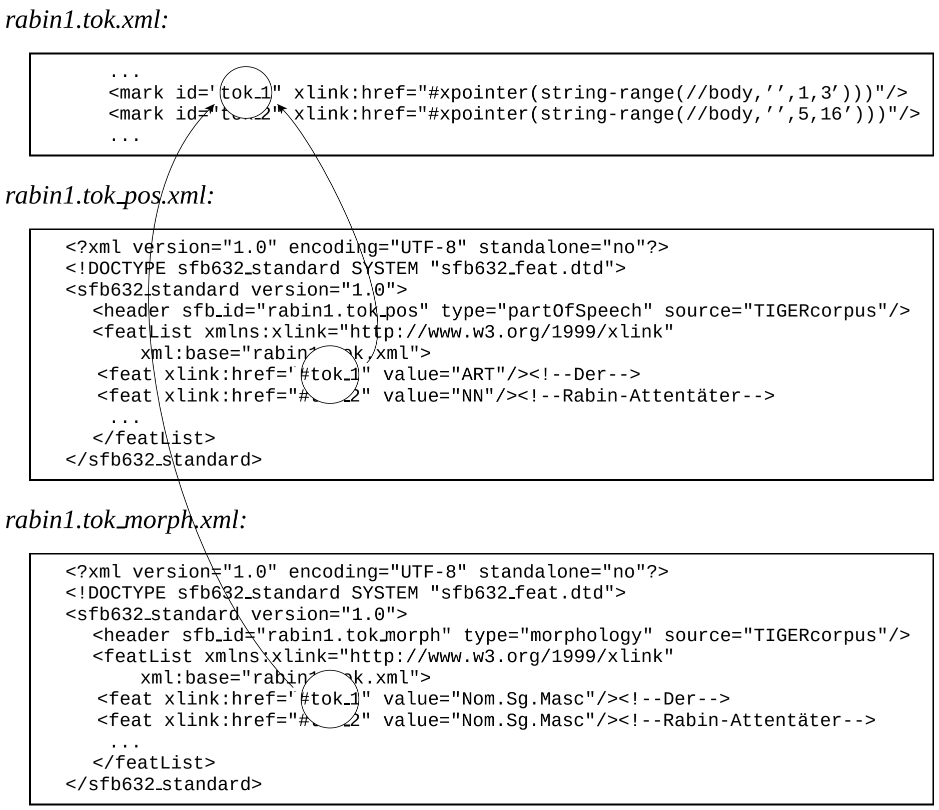
rabin1.tok.xml:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE sfb632_standard SYSTEM "sfb632_mark.dtd">
<sfb632_standard version="1.0">
  <header sfb_id="rabin1.tok" type="tokens" source="TIGERcorpus"/>
  <markList xmlns:xlink="http://www.w3.org/1999/xlink"
      xml:base="rabin1.text.xml">
   <mark id="tok_1" xlink:href="#xpointer(string-range(//body,’’,1,3’)))"/>
   <mark id="tok_2" xlink:href="#xpointer(string-range(//body,’’,5,16’)))"/>
   <mark id="tok_3" xlink:href="#xpointer(string-range(//body,’’,22,5’)))"/>
   ...
  </markList>
</sfb632_standard>
```

rabin1.const.xml:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE sfb632_standard SYSTEM "sfb632_struct.dtd">
<sfb632_standard version="1.0">
  <header sfb_id="rabin1.const" type="constituents" source="TIGERcorpus"/>
  <structList xmlns:xlink="http://www.w3.org/1999/xlink">
   <struct id="syn_1">
     <rel id="rel_1" type="edge" xlink:href="rabin1.tok.xml#tok_3"/>
     <rel id="rel_2" type="edge" xlink:href="rabin1.tok.xml#tok_4"/>
   </struct>
   <struct id="syn_2">
     <rel id="rel_3" type="edge" xlink:href="rabin1.tok.xml#tok_6"/>
     <rel id="rel_4" type="edge" xlink:href="#syn_20"/>
     <rel id="rel_6" type="edge" xlink:href="#syn_21"/>
   </struct>
   ...
  </structList>
```

Figure 1: Source text (*rabin1.text.xml*), `<mark>` tags specifying tokens (*rabin1.tok.xml*), and `<struct>` tags specifying constituents (*rabin1.const.xml*)



rabin1.tok.xml:

```
      ...
      <mark id="tok_1" xlink:href="#xpointer(string-range(//body,'',1,3')))"/>
      <mark id="tok_2" xlink:href="#xpointer(string-range(//body,'',5,16')))"/>
      ...
```

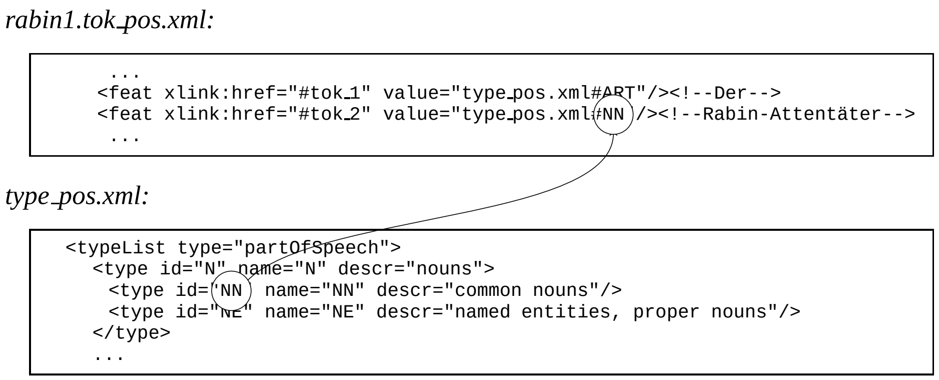
rabin1.tok_pos.xml:

```
  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
  <!DOCTYPE sfb632_standard SYSTEM "sfb632_feat.dtd">
  <sfb632_standard version="1.0">
     <header sfb_id="rabin1.tok_pos" type="partOfSpeech" source="TIGERcorpus"/>
     <featList xmlns:xlink="http://www.w3.org/1999/xlink"
          xml:base="rabin1.tok.xml">
      <feat xlink:href="#tok_1" value="ART"/><!--Der-->
      <feat xlink:href="#tok_2" value="NN"/><!--Rabin-Attentäter-->
       ...
     </featList>
  </sfb632_standard>
```

rabin1.tok_morph.xml:

```
  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
  <!DOCTYPE sfb632_standard SYSTEM "sfb632_feat.dtd">
  <sfb632_standard version="1.0">
     <header sfb_id="rabin1.tok_morph" type="morphology" source="TIGERcorpus"/>
     <featList xmlns:xlink="http://www.w3.org/1999/xlink"
          xml:base="rabin1.tok.xml">
      <feat xlink:href="#tok_1" value="Nom.Sg.Masc"/><!--Der-->
      <feat xlink:href="#tok_2" value="Nom.Sg.Masc"/><!--Rabin-Attentäter-->
       ...
     </featList>
  </sfb632_standard>
```

Figure 2: Tokens (*rabin1.tok.xml*) and `<feat>` tags specifying part of speech (*rabin1.tok_pos.xml*) and morphology (*rabin1.tok_morph.xml*)



rabin1.tok_pos.xml:

```
       ...
      <feat xlink:href="#tok_1" value="type_pos.xml#ART"/><!--Der-->
      <feat xlink:href="#tok_2" value="type_pos.xml#NN"/><!--Rabin-Attentäter-->
       ...
```

type_pos.xml:

```
  <typeList type="partOfSpeech">
     <type id="N" name="N" descr="nouns">
       <type id="NN" name="NN" descr="common nouns"/>
       <type id="NE" name="NE" descr="named entities, proper nouns"/>
     </type>
     ...
```

Figure 3: Annotation of part of speech: anchoring to token markables (*rabin1.tok.xml*) and definition in a type hierarchy (*type_pos.xml*)