DEREKO Linguistic Markup

Tylman Ule

SfS Tübingen

ule@sfs.uni-tuebingen.de

February 27, 2002

Abstract

Linguistic annotation is added to the DEREKO corpus by identifying first the word form tokens, and consequently grouping them to describe higher levels of syntactic annotation, resulting in, e.g. chunks and topological fields. Linguistic annotation is added using a markup scheme originating in the *Corpus Encoding Standard* (CES), or rather, in its XML variant XCES. All extensions to the XCES with respect to linguistic annotation are discussed here.

1 Introduction

DEREKO corpora are annotated in two major phases. First, the texts are converted into a representation readable by computers, and meta-information of a text (concerning, e.g., author, date of publication, etc.) is encoded in the markup according to the *Corpus Encoding Standard* (CES) [1], including paragraph and sentence segmentation.[1] The first phase is the basis for the linguistic markup described below.

All annotation is added to the original document in-line. The original text can be reproduced when all linguistic markup is removed.[2] The CES is extended using in-line linguistic annotation. The original proposal of using stand-off annotation was not followed mainly for the following reasons:

- It is assumed that the DEREKO linguistic annotation will be useful for most applications. Processing links from the annotation to the original corpus (i.e. using stand-off annotation) may outweigh the benefit of having a smaller source, when the links have to be followed for the vast majority of applications.
- The original unannotated corpus can be reproduced from the annotated corpus, because e.g. all formatting information is preserved in the markup.

[1] The first phase is carried out by the IDS, resulting in CES *level 1 conformance*. The acronyms CES and XCES are used interchangeably in this document.

[2] All original information is recorded in the attribute `f` of tokens and surrounding whitespace.

- A separate version of the unannotated corpus is mainly advisable when several orthogonal annotations are added later. The annotation scheme used in the DEREKO project, however, is considered to be a starting point for further annotation itself.

- Many current XML processing tools support efficient processing of a stream of XML data, and standards for linking XML documents are not yet as widely supported.

All characters that encode format information are called *whitespace* characters, and they include tabs, line breaks, spaces between word forms, etc. All `#PCDATA` information in DEREKO documents contains only information about whitespace.

Markup is described in this text as follows:

`<element>` Description of element `element`

`attribute` Description of attribute `attribute` of `<element>`

`"value":` possible value of `attribute`. `"value"` does not necessarily specify the complete value. It may also refer to parts of the attribute value string, as long as its value is defined as `CDATA` .

Some symbols have been borrowed from regular expressions syntax to describe the occurrence of elements and attributes. An element occurs at a certain position (i.e. relative to its parent element) at least once when it is marked by `+`. Accordingly, a `*` means it may occur zero times or more, and a `?` that it may occur once, or not at all. `1` means it has to occur once and only once. Only `1` and `?` are applicable for attributes, because they may occur at most once per element.

A short example in the text is given a follows:

- `<t f='Beispiel'/>`

A longer example is given in a paragraph of its own as follows:

```
<t f='Etwas' i=''/> <t f='Mehr' i=''/>
<t f='Text' i=''/><t f='.' i='PUNCT'/>
```

All examples show correctly annotated text. Details not relevant in the current context, however, may be left out. The symbols `?`, `1`, `*`, `+`, on the other hand, specify what to expect minimally or maximally from the annotated corpus.

2 Corpus

The DTD for linguistic annotation is an extension to the *xcesDoc*-DTD. The xcesDoc-DTD has not been modified to accommodate linguistic annotation above the sentence level.[3]

[3]Please contact the IDS for changes to the xcesDoc-DTD at document level (B. Endres `<endres@ids-mannheim.de>`).

3 Sentence

<s> [+] groups word forms in a sentence. The tokens may be grouped inside <s> to build larger units using <ch>, <fd> or <cl> elements. Whitespace within <s> is recorded from the unannotated document.

4 Token

<t> [+] contains all information about a single word form token. Information is only given in the element's attributes and its sub-elements. Whitespace within the element or its sub-elements carries no meaning.

f [1] The word form

- <t f='forderte'/>

i [?] Verbose information concerning tokenisation. The attribute contains CDATA and may contain one or more of the values listed below. The attribute may also be empty, and then the word form is a *plain* word form made up only of the characters given in Fig. 1. The word form may also consist of "'" characters except for the first and last letter (but see *clitics* below).

a-zäöüßA-ZÄÖÜáàâæçéëèêíïìîñóòôúûùÁÀÂÃÇÉÈÊÍÏÌÎÑÓÒÔÚÙÛ

Figure 1: Characters in *plain* word forms

More complex word forms are only marked up in certain contexts. Therefore, some of the values given below may be combined, and not all occurrences of the following types of word forms are detected in the corpus.

"PUNCT": Punctuation

- <t i='PUNCT' f='.'/>

"ABBR": Abbreviation

- <t i='ABBR' f='Mio.'/>
- <t i='ABBR' f='z.B.'/>

"INIT": Initial

- <t i="" f='Von'/> <t i='INIT' f='C.'/> <t i="" f='Albrecht'/> <t i="" f='und'/>

"NUM": Number

- <t i='NUM' f='13.000'/>
- <t i='NUM' f='1984'/>

"NUMTOK": Token containing a number

- <t i='NUMTOK' f='R2D2'/>

"ORD": Ordinal number

- `<t i=" f='im'/> <t i='ORD' f='2.'/> <t i=" f='Quartal'/>`

"TEL": Telephone number

- `<t i='TEL' f='030/4609-212'/>`

"RATIO": Ratio

- `<t i='RATIO' f='3:2'/>`

"AREA": Area

- `<t i='AREA' f='6x4'/>`

"URL": Internet URL

- `<t i=" f='im'/> <t i='URL' f='http://www.bundestag.de/'/>`
- `<t i=" f='der'/> <t i=" f='Website'/> <t i='URL' f='www.napster.com'/>`

"MAIL": Internet mail address

- `<t i='MAIL' f='ule@sfs.uni-tuebingen.de'/>`

The word forms `ein` and `es` when used as *clitics* are recognised when followed by a space. In addition, the token preceding `es` has to consist of all lowercase letters.

"CLITIC": Clitic

- `<t i='CLITIC' f="'ne"/> <t i=" f='Mark'/>`
- `<t i=" f='gibt'/><t i='CLITIC' f="'s"/> <t i=" f='nicht'/>`

Parts of a date expression have the following attribute values:

"DAY": Day (number or weekday)

"MONTH": Month (number or name of month)

"YEAR": Year. A word form matching the following regular expression: `/(1[89]|20|')[0-9][0-9]/`

A name of a month is always annotated as such. A number referring to a month, a day, or a year (which is always expected to be a number) is only marked up when at least two of them occur in a sequence. Month numbers may be Arabic or Roman numbers.

- `<t i=" f='am'/> <t i='DAY' f='15.'/> <t i='MONTH' f='Juni'/>`
- `<t i=" f='ab'/> <t i='DAY' f='17.'/> <t i='MONTH' f='April'/> <t i='YEAR' f='1979'/>`

Amounts of a currency are recognised when they are followed or preceded by a currency unit. Currency units may either be symbols or given as text.

"CURNUM": Amount of currency

"CURTYP": Unit of currency

- `<t i='CURNUM' f='150,-'/> <t i='CURTYP' f='DM'/>`
- `<t i='CURNUM' f='25.000'/> <t i='CURTYP' f='Mark'/>`

More generally, *numbers with units of measurement* are recognised when the unit follows the number.

"MEASNUM": Number

"MEASTYP": Unit of measurement

- `<t i='MEASNUM' f='3'/><t i='MEASUNIT' f='kg'/>`

Hyphenated word forms (using either dash "–" or slash "/") are recognised when the parts of the token separated by hyphens are of one of the following types: date, telephone number, ratio, area, token containing a number, currency, measure, ordinal number, number, mail address, URL, abbreviation, initial, or plain word form. Word forms with a trailing dash ("–") are also marked up as hyphenated word forms. Quotation marks (double or single quotes) may surround the parts of a hyphenated word form,

"HY": records each hyphenated part of a word form (always followed by "–" or "/").

- `<t i='NUM HY- HY-' f='50-Prozent-Quotierung'/>`
- `<t i=' HY-' f="'Traumhochzeit'-Moderator"/>`

Amounts of a currency, dates (when lexicalised), abbreviations, and numbers with units of measurement rely on lexicons and are only recognised when there is information available for them.

<P> [+] groups all information related to a single part of speech (POS). There is a `<P>` child element to a `<t>` for each distinct POS of a word form. Information is recorded not only for a single best POS analysis, but also for other possible analyses.

t [1] The POS tag. One of the tags specified in the STTS tag set [4].

r [1] Each POS analysis is assigned a rank. The first rank ("1") is assigned to the POS judged to be the best analysis. All subsequent ranks are assigned to the respective next best POS. All POS assigned rank "0" are ignored. The rank is usually derived from the relation of certainties assigned to all POS. It may also be assigned by rule-based methods not assigning certainties, however.

c [1] Certainty assigned to a POS tag analysis. `c` may take any value from the interval $[0, 1]$. It may be zero, e.g., for a POS determined only by a rule-based correction of statistical POS taggers, when this attribute is used to

record the certainties assigned by the statistical taggers. The sum of all these values over all POS is 1.

There is not always only a single first rank. The mean of the certainty of all judges is computed to determine the rank of a POS. If the mean is equal for two or more POS, the standard deviation of the means is considered next, preferring lower standard deviations. The POS receive the same rank if the respective standard deviations are equal, too.

Example:

```
<t f='zusammengestrichen' i=''>
  <P t='ADJD' r='2' c='0.364375925'/>
  <P t='VVPP' r='1' c='0.635624075'/>
</t>
```

Example for two first ranks:

```
<t f='Coca-Cola' i=' HY-'>
  <P t='NE' r='1' c='0.5'>
    <j n='taggerA' c='1'/>
    <j n='taggerB' c='1'/>
  </P>
  <P t='NN' r='1' c='0.5'>
    <j n='taggerC' c='1'/>
    <j n='taggerD' c='1'/>
  </P>
</t>
```

<b> [*] contains the baseform of a word form and groups all morphological analyses for this baseform with respect to the parent POS element and grandparent word form token element. The morphological analyses therefore represent the morphological ambiguity class for the combination of baseform, POS, and word form. Not all word forms are assigned an analysis by the morphological analyser used presently for DEREKO (DMOR – Deutsche Morphologie [3]), so that the element <b> is not available for all word form tokens.

f [1] Baseform

Example:

```
<t f='sich' i=''>
  <P t='PRF' r='1' c='1'>
    <b f='Sie'>
      <j n='machineA' c='1'/>
```

```
      <m d='p3'/>
    </b>
    <b f='er'>
      <j n='machineA' c='1'/>
      <m d='s3'/>
    </b>
    <b f='es'>
      <j n='machineA' c='1'/>
      <m d='s3'/>
    </b>
    <b f='sie'>
      <j n='machineA' c='1'/>
      <m d='p3'/>
      <m d='s3'/>
    </b>
  </P>
</t>
```

<m> [*] encodes a possible morphological analysis of a full form. All <m> child elements of the common <b> parent element make up a morphological ambiguity class. <m> elements only occur within <b> elements, so that there are no <m> elements when the <b> element is not given.

d [1] Description of a morphological analysis. A combination of morphological features (column *Combination* in Table 1)[4] is defined for each POS tag together with possible feature values. Each feature is assigned a letter position in a certain order, resulting in a set of feature combinations (see Table 2), and each feature may take any of the values given in Table 3.

Example for a token with more than one POS, baseform, and morphological analysis:

```
<t f='der' i=''>
  <P t='PRELS' r='2' c='0.0007'>
    <j n='taggerA' c='0.001055903'/>
    <j n='taggerB' c='0.0006380229'/>
    <j n='taggerC' c='0.001174612'/>
    <b f='der'>
      <j n='machineA' c='1'/>
      <m d='nsm'/>
    </b>
    <b f='die'>
```

[4]For completeness' sake the value combinations of the original DMOR analysis corresponding to the POS in the column *STTS* are given in the column *DMOR*. They do not appear in the final DEREKO markup, however.

```
      <j n='machineA' c='1'/>
      <m d='dsf'/>
    </b>
  </P>
  <P t='ART' r='1' c='0.9993'>
    <j n='taggerA' c='0.9989441'/>
    <j n='taggerB' c='0.999362'/>
    <j n='taggerC' c='0.9988254'/>
    <j n='taggerD' c='1'/>
    <b f='der'>
      <j n='machineA' c='1'/>
      <m d='nsm'/>
    </b>
    <b f='die'>
      <j n='machineA' c='1'/>
      <m d='dsf'/>
      <m d='gp0'/>
      <m d='gsf'/>
    </b>
  </P>
</t>
```

Table 1: Feature combinations for STTS tags

STTS Tag	DMOR	Combination
$, $. $(	IP	not analysed
ADJA	ADJ	cngs
ADJA	ADJ.Invar	not analysed
ADJA ADJD ADV	ORD	cngs
ADJD	ADJ.Adv	not analysed
ADJD	ADJ.Pred	not analysed
ADV	ADV	not analysed
APPO APZR	POSTP	not analysed
APPR	PREP	c
APPRART	PREP/ART	cng
ART	ART	cng
CARD	CARD	not analysed
ITJ	INTJ	not analysed
KOKOM	KONJ.Vgl	not analysed
KON	KONJ.Kon	not analysed
KOUI	KONJ.Inf	not analysed
KOUS	KONJ.Sub	not analysed
NE	NE	cng

table continues on next page

STTS Tag	DMOR	Combination
NE	NE.Invar	not analysed
NE	NEGeo	cng
NN	NN	cngs
PAV	PROADV	not analysed
PDAT	DEM.attr	not analysed
PDS	DEM.pro	not analysed
PDS	DEM.subst	not analysed
PIDAT	INDEF	not analysed
PIDAT PIAT PIS	INDEF.pro	cng
PIDAT PIS	INDEF.attr	cng
PIS	INDEF.subst	cng
PPER	PPRO.pers	cngp
PPOSAT	POSS.attr	cng
PPOSAT	POSS.pro	cng
PPOSS	POSS.subst	cng
PRELAT	REL.attr	cng
PRELS	REL.subst	cng
PRF	PPRO.prfl	cngp
PRF	PPRO.refl	ngp
PRF	PPRO.rez	not analysed
PTKA	PTKL.Adj	not analysed
PTKANT	PTKL.Ant	not analysed
PTKNEG	PTKL.Neg	not analysed
PTKVZ	VPRE	not analysed
PTKZU	PTKL.zu	not analysed
PWAT	WPRO.pro	not analysed
PWAT	WPRO.attr	not analysed
PWAV	WADV	not analysed
PWAV	WADV	not analysed
PWS	WPRO.subst	not analysed
TRUNC	TRUNC	not analysed
VAFIN VMFIN VVFIN	V.PPres	not analysed
VAIMP VVIMP	V.Imp	not analysed
VAINF VMINF VVINF	V.Inf	not analysed
VAPP VMPP VVPP	V.PPast	not analysed
VVFIN VAFIN VMFIN	V	pn
VVIZU	V.Inf.zu	not analysed

<j> [+] A judge. Judges may be POS taggers assigning a certainty to a POS tag, or also morphological analysers determining the baseform and/or the morphological ambiguity class of a token. Judges are always child elements of the elements that they vote for.

Table 2: Feature combinations

Combination	Features
c	Case
cng	Case, Number, Gender
cngp	Case, Number, Gender, Person
cngs	Case, Number, Gender, Inflection Type
g	Gender
n	Number
ngp	Number, Gender, Person
pn	Person, Number
s	Inflection Type

Table 3: Feature values

Feature	Letter	Possible Values
Case	c	n (Nom) g (Gen) a (Akk) d (Dat)
Number	n	s (Sg) p (Pl)
Gender	g	f (Fem) m (Masc) n (Neut) 0 (NoGend)
Person	p	1 (1. Pers) 2 (2. Pers) 3 (3. Pers)
Inflection Type	s	m (Mix) t (St) T (St/Mix) w (Sw) W (Sw/Mix)

n [1] The name of the judge. This name is unique in a corpus and is connected to a single judge within a corpus.

c [1] The judge votes for the judge's parent element with certainty `c`. All `c` attributes of a judge do not necessarily sum up to 1. Votes of a single judge are, however, normalised by the sum of all their `c` values within the same `<t>` element when they are used to determine POS ranks.

Only a single morphological analyser (DMOR) is used at the moment to determine the baseform and the morphological ambiguity class. It does not weigh its analyses, so that all `<j>` children of `<b>` elements receive certainty 1.

Examples:

```
<t f='Tode' i=''>
  <P t='NN' r='1' c='1'>
    <j n='taggerA' c='1'/>
    <j n='taggerB' c='1'/>
  </P>
</t>
```

```
<t f='Seite' i=''>
  <P t='NE' r='2' c='0.041545375'>
    <j n='taggerD' c='0.1661815'/>
  </P>
  <P t='NN' r='1' c='0.958454625'>
    <j n='taggerA' c='1'/>
    <j n='taggerB' c='1'/>
    <j n='taggerC' c='1'/>
    <j n='taggerD' c='0.8338185'/>
    <b f='Seite'>
      <j n='machineA' c='1'/>
    </b>
  </P>
</t>
```

5 Chunk, Field, Clause

Only the element and attribute names used to encode chunks, fields, and clauses are listed here. A more detailed description of the syntactic structures annotated with these elements is given in [2], which also lists all attribute values used in the markup.

<ch> [*] Chunk

- **c** [1] The category of the chunk.

<fd> [*] Field

- **c** [1] The category of the field.

<cl> [*] Clause

- **c** [1] The category of the clause.

6 Quotation Marks

When linguistic annotation dominating one or more tokens is added, quotation marks (i.e. single "'" and double "") are treated as part of the following token. As a result, quotation marks may sometimes appear in unexpected places, e.g. inside of verb chunks. They should be considered not to be linguistically attached at all, despite of their position in the XML tree structure. The XML tree structure, which is used directly to encode linguistic structure in the DEREKO corpus, cannot handle unattached elements straightforwardly when the sequence of elements resembles the original word order. As a result, quotation marks may appear in any element dominating word form tokens.

7 Efficiency and Minimising XML

Both corpus size and processing speed become an issue when corpora of up to 1×10^9 word form tokens are annotated. Therefore, an experiment was carried out to examine the influence of different element and attribute names and other XML minimisation techniques on processing speed and on the size of the annotated corpus (see Table 4). The experiments were conducted with a fully annotated corpus of 1.5×10^6 tokens.

Table 4: Size and Processing Speed vs. Minimisation Strategy

S	Size	%	Size gz	%	xmlnorm	%	nsgmls	%
1	798.888.363	100	84.209.882	100	88.54	100	283.1	100
2	614.052.706	77	78.057.635	93	66.48	75	233.42	82
3	492.465.059	62	74.489.661	88	72.45	82	265.63	94
4	438.983.279	55	72.558.418	86	60.59	68	230.79	82
5	321.688.472	40	55.487.508	66	52.06	58	191.19	68

The columns "Size" and "Size gz" show the size (in bytes) of the uncompressed and compressed corpus file for each minimisation strategy "S".[5] The columns "xmlnorm" and "nsgmls" show the time (in seconds) that these tools need to validate the corpus.[6] Validation time is expected to show the influence of XML parsing on the overall time needed for linguistic annotation, or for processing the corpus in general.

Minimisation strategies 1 to 4 result in markup conveying the same information content. Strategy 5 drops the judge encoding the standard deviation. Strategies 1 to 4 only differ in what XML minimisation techniques are applied to the data.

1. No minimisation is performed, i.e. elements and attributes have verbose names, and there are explicit closing tags for empty elements. Example for the word "Zum":

```
<token form='Zum' info=''><pos tag='APPRART'
rank='1' cert='1'><judge name='news-100'
cert='1'></judge><judge name='novel-100'
cert='1'></judge><judge name='all-100'
cert='1'></judge><judge name='sd'
cert='1'></judge><baseform form='*zum'><judge
name='DMOR-MK1' cert='1'></judge><morph
desc='dsm'></morph><morph
desc='dsn'></morph></baseform></pos></token>
```

2. Some attribute values are replaced by DTD default values, and element and attribute names are still verbose. Empty elements are abbreviated using the XML empty element notation. Example:

[5] Using `gzip` with the default compression ration of 6.

[6] `xmlnorm` is part of the LT XML library (version 1.2.4beta; `http://www.ltg.ed.ac.uk/software/xml/`). `nsgmls` version 1.3.4 was used for the experiments (`http://www.jclark.com/sp/`).

```
<token form='Zum'><pos tag='APPRART'><judge
name='news-100'/><judge
name='novel-100'/><judge name='all-100'/><judge
name='sd'/><baseform form='*zum'><judge
name='DMOR-MK1'/><morph desc='dsm'/><morph
desc='dsn'/></baseform></pos></token>
```

3. Long element and attribute names are replaced by short names, but DTD default values are not used. Empty elements are encoded using XML empty element notation. Example:

```
<t f='Zum'><P c='1' r='1' t='APPRART'><j c='1'
n='news-100'/><j c='1' n='novel-100'/><j c='1'
n='all-100'/><j c='1' n='sd'/><b f='*zum'><j
c='1' n='DMOR-MK1'/><m d='dsm'/><m
d='dsn'/></b></P></t>
```

4. Combination of strategy 2 and strategy 3, i.e. short names, DTD default values, and XML empty element notation. Example:

```
<t f='Zum'><P t='APPRART'><j n='news-100'/><j
n='novel-100'/><j n='all-100'/><j n='sd'/><b
f='*zum'><j n='DMOR-MK1'/><m d='dsm'/><m
d='dsn'/></b></P></t>
```

5. Symbolic judge names are replaced by unique numbers, and the judge encoding standard deviation is dropped. Example:

```
<t f='Zum'><P t='APPRART'><j n='1'/><j
n='2'/><j n='3'/><b f='*zum'><j n='4'/><m
d='dsm'/><m d='dsn'/></b></P></t>
```

Processing speed is increased most strikingly by using DTD default values, while corpus size (esp. uncompressed) is reduced best using short element and attribute names. All of the above minimisation techniques are used to encode DEREKO markup, because strategy 5 results in a 34% reduction in compressed file size and speeds up processing by up to 42%. Scripts accompany the corpus converting the DEREKO XML format into, e.g., a bracketed vertical format, or into HTML, so that they compensate for reduced legibility of the XML source text caused by short element and attribute names.

References

[1] Nancy Ide, Patrice Bonhomme, and Laurent Romary. XCES: An XML-based encoding standard for linguistic corpora. In *2nd International Conference on Language Resources & Evaluation (LREC 2000), 31 May–2 June 2000*, Athens, Greece, 2000.

[2] Frank Henrik Müller. Shallow-parsing style book for German. Technical report, Seminar für Sprachwissenschaft, Universität Tübingen, 2002.

[3] Anne Schiller. DMOR: Benutzerhandbuch. Technical report, IMS, University of Stuttgart, 1995.

[4] Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. Vorläufige Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, Universität Stuttgart, Institut für maschinelle Sprachverarbeitung, and Seminar für Sprachwissenschaft, Universität Tübingen, 1995.