Markup Manual for the Tübingen Partially Parsed Corpus of Written German (TüPP-D/Z)

Tylman Ule

SfS Tübingen

ule@sfs.uni-tuebingen.de

January 15, 2004

Abstract

This document describes the document markup of TüPP-D/Z, the *Tübingen Partially Parsed Corpus of Written German.* It describes the encoding of segmentation (into paragraphs, sentences, and words), of part-of-speech tags, and of morphological information. It also briefly shows how the levels of clauses, chunks, and topological fields are encoded, and complements Müller (2004), which describes the encoded phenomena in more detail.

Contents

1 Introduction

This document serves as a guide to the markup of TüPP-D/Z, the *Tübingen Partially Parsed Corpus of Written German* (Tübinger partiell geparstes Korpus – Deutsch / Zeitung). Markup in TüPP-D/Z is encoded using the extensible markup language (XML; W3C, 2000), and this document describes all XML elements and attributes used to encode linguistic annotation, and also all other markup used in the corpus, e.g. general textual markup encoding bibliographical information. Tools and the methods employed in the actual annotation process are not exhaustively described here – please refer to Ule and Müller (2004) for a more detailed description of the annotation process.

TüPP-D/Z consists of texts from the German daily newspaper *die tageszeitung (taz)*. Texts for the current release are taken from the 1999 HTML distribution of the *taz*, which includes newspaper articles from September 2, 1986, up to May 7, 1999 (more than 200 million words). Part of the document markup was originally encoded as HTML comments and has been retained in TüPP-D/Z (see Sec. 2).[1]

All linguistic annotation is added to the original document in-line. The original text can be reproduced when all linguistic markup is removed.[2] 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 TüPP-D/Z 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 list of all possible values. It may also refer to parts of the attribute value string when its value is not a predefined set but `CDATA`.

Some symbols have been borrowed from regular expression 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.

Longer examples of XML encoded text appear in `monospaced` font in a separate paragraph. Shorter examples are additionally surrounded by a box:

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

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.

Please consult the TüPP-D/Z homepage for up-to-date information:

`http://www.sfs.uni-tuebingen.de/tupp`

[1] Newer releases of the *taz* are natively provided as XML and will therefore differ in markup.

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

2 General Textual Markup

TüPP-D/Z consists of a sequence of days which form a collection. Each day contains a sequence of articles, which in turn contain the text which is linguistically annotated.[3]

<COLLECTION> [1] The root element. A collection consists of one or more days. It is meant to provide a container for a custom corpus consisting of all days, or only a subset of them.

<DAY> [+] A day holds all articles available for one day of *taz* data.

<ART> [+] An article groups the text of an article with bibliographical information.

<B> **removed** Bold face text in paragraphs, or inside italics text was encoded with this element. All occurrences have been removed.

<I> **removed** Italics text in paragraphs, or inside bold face text was encoded with this element. All occurrences have been removed.

**
** [?] Empty element that encodes HTML-style forced line breaks. Paragraphs do not span this element.

The following elements contain primary data, i.e. only here linguistic annotation is present in TüPP-D/Z. Whitespace within these element is recorded from the unannotated document.

<TI> [1] Main title, which may contain minor titles (<H2>, <H3>).

<H2> [*] Minor title

<H3> [*] Minor title. The element <H3> has been removed whenever it occurred inside of paragraphs in the original data.

<TX> [1] Text body

The remaining elements used in TüPP-D/Z are only briefly mentioned here, because they continue to store the same information as the corresponding markup in the original *taz* distribution. No linguistic annotation has been added inside of these elements. According to the DTD, at least one of them appears once or more ([+]) at the beginning of an <ART>, preceding <TI> and <TX>[4].

<AR> Text type

<AU> Author

<DT> Publishing date

<KT> Short title

[3] Markup described in this section is likely to change with new releases of TüPP-D/Z.

[4] Only weak restrictions have been imposed on these elements by the TüPP-D/Z DTD. It has not been tested whether stronger restrictions apply.

<QU> Source of article

<RE> Subject area

<SE> Page number

<TP> Unique ID of article

<ZE> Number of lines

3 Linguistic Annotation

Linguistic annotation in TüPP-D/Z can be seen as proceeding from plain text to full annotation in two-steps. First, the text is segmented into paragraphs, sentences, and words. Second, the words are assembled into chunks, topological fields, and clauses. The following description of the markup that encodes linguistic annotation proceeds in this order.

3.1 Paragraph

<p> [+] groups sentences. It never contains anything but sentences or whitespace, which is recorded from the original document. Paragraphs end when the surrounding elements end, when a
 occurs, or when two subsequent newlines are encountered. All primary non-whitespace data is enclosed in paragraph elements. Whitespace within <p> is recorded from the unannotated document.

3.2 Sentence

<s> [+] groups word forms into sentences. 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.

3.3 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-ZÄÖÜáàâæåæçéëèêíïìîñóòôøúûùÿýÁÀÂÃÅÆÇÉËÈÊÍÏÌÎÑÓÒÔØÚÙÛÝ
```

Figure 1: Characters in *plain* word forms

More complex word forms are only marked up in certain contexts, and recognition relies on hand-crafted lexicons. Therefore, some of the values given below only occur in combination, 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": Electronic mail address

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

"CLITIC": Clitic

```
<t i='CLITIC' f="'ne"/> <t i=" f='Mark'/>
```

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

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.

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, email 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 they are listed.

<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 (Schiller et al., 1995).
- **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.[5]

[5]The term *certainty* has been chosen instead of *probability*, because it may well reflect preferences without probabilistic grounding.

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 necessarily receive morphological annotation, because words may be unmarked for morphology, or the analyser may be unable to assign an analysis, 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 is defined for each POS tag (column *Combination* in Table 1).[6] Each feature is assigned a letter position in a certain order, resulting in a set of feature-value pairs (see Table 2 for a description of the feature names, and Table 3 for the values they can take).

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'>
      <j n='machineA' c='1'/>
      <m d='dsf'/>
    </b>
```

[6]For completeness' sake the value combinations of the original DMOR analysis (Schiller, 1995) corresponding to the POS in the column *STTS* are given in the column *DMOR*. Only instantiations of the *Combination* column appear in the final markup. $\emptyset$ means no analysis is available.

```
  </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	∅
ADJA	ADJ	cngs
ADJA	ADJ.Invar	∅
ADJA ADJD ADV	ORD	cngs
ADJD	ADJ.Adv	∅
ADJD	ADJ.Pred	∅
ADV	ADV	∅
APPO APZR	POSTP	∅
APPR	PREP	c
APPRART	PREP/ART	cng
ART	ART	cng
CARD	CARD	∅
ITJ	INTJ	∅
KOKOM	KONJ.Vgl	∅
KON	KONJ.Kon	∅
KOUI	KONJ.Inf	∅
KOUS	KONJ.Sub	∅
NE	NE	cng
NE	NE.Invar	∅
NE	NEGeo	cng
NN	NN	cngs

table continues on next page

STTS Tag	DMOR	Combination
PAV	PROADV	∅
PDAT	DEM.attr	∅
PDS	DEM.pro	∅
PDS	DEM.subst	∅
PIDAT	INDEF	∅
PIDAT PIAT PIS	INDEF.pro	cng
PIDAT PIS	INDEF.attr	cng
PIS	INDEF.subst	cng
PPER	PPRO.pers	cngp
PPER PRF	PPRO.prfl	cngp
PPOSAT	POSS.attr	cng
PPOSAT	POSS.pro	cng
PPOSS	POSS.subst	cng
PRELAT	REL.attr	cng
PRELS	REL.subst	cng
PRF	PPRO.refl	ngp
PRF	PPRO.rez	∅
PTKA	PTKL.Adj	∅
PTKANT	PTKL.Ant	∅
PTKNEG	PTKL.Neg	∅
PTKVZ	VPRE	∅
PTKZU	PTKL.zu	∅
PWAT	WPRO.pro	∅
PWAT	WPRO.attr	∅
PWAV	WADV	∅
PWAV	WADV	∅
PWS	WPRO.subst	∅
TRUNC	TRUNC	∅
VAFIN VMFIN VVFIN	V.PPres	∅
VAIMP VVIMP	V.Imp	∅
VAINF VMINF VVINF	V.Inf	∅
VAPP VMPP VVPP	V.PPast	∅
VVFIN VAFIN VMFIN	V	pn
VVIZU	V.Inf.zu	∅

<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.

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

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)

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 is currently used 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>
```

3.4 Chunk, Field, and Clause

Chunks, fields, and clauses only occur inside of sentences. Only the element and attribute names used to encode them are listed here. A more detailed description of the syntactic structures annotated with these elements is given in Müller (2004). Please refer to the DTD for a more detailed list of restrictions on the dominance relations.

<ch> [*] Chunk

- **c** [1] Category of the chunk. See table 1 in Müller (2004) for the list of values.

<fd> [*] Topological Field – only occurs in clauses and field coordinations

- **c** [1] Category of the field
  - **"CF":** Complementizer Field
  - **"KOORDF":** Coordination Field
  - **"PARORDF":** Coordination Field
  - **"LV":** Resumptive construction (*Linksversetzung*)
  - **"VF":** Initial field (*Vorfeld*)
  - **"MF":** Middle field (*Mittelfeld*)
  - **"NF":** Final field (*Nachfeld*)

<fdc> [*] Coordination of fields – only occurs in clauses

- **c** [1] Category of the field coordination
  - **"MFVCC":** Coordinations of *Mittelfeld* and right part of the verbal bracket, which is the only field coordination currently annotated.

<cl> [*] Clause

c [1] Category of the clause

"V1": Verb-first clause

"V2": Verb-second clause

"REL": Relative clause (a subtype of a verb-last clause)

"INF": Infinitive clause (a subtype of a verb-last clause)

"SUB": General subordinate clause (a subtype of a verb-last clause)

4 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 TüPP-D/Z, 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.

5 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".[7] The columns "xmlnorm"

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

and “nsgmls” show the time (in seconds) that these tools need to validate the corpus.[8] 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:

```
<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:

[8] `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/`).

```
<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 in TüPP-D/Z 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 TüPP-D/Z 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.

Acknowledgments

The annotation of TüPP-D/Z has taken great advantage of resources and tools originally set up in the DEREKO project (`http://www.sfs.uni-tuebingen.de/dereko`). The tools have been updated with support from the DFG project *Sonderforschungsbereich 441: Linguistische Datenstrukturen*.

TüPP-D/Z has been annotated using KaRoPars, which integrates a number of tools into a cascaded annotation system (Ule and Müller, 2004). We are grateful to the authors of these tools, which include

- `fsgmatch` – a general-purpose transducer operating on XML (Mikheev et al., 1999)
- `tnt` – a part-of-speech tagger (Brants, 2000)
- `xmlperl` – an XML processing/translating language (McKelvie, 1999)
- DMOR – Deutsche Morphologie (Schiller, 1995)

References

Thorsten Brants. TnT – a statistical part-of-speech tagger. In *Proceedings of ANLP-2000, April*, Seattle, WA, 2000.

David McKelvie. *XMLPERL 1.0.4*. Language Technology Group, University of Edinburgh, Edinburgh, 1999. URL `http://www.cogsci.ed.ac.uk/~dmck/xmlperl`.

Andrei Mikheev, Claire Grover, and Marc Moens. XML tools and architecture for named entity recognition. *Markup Languages*, 1(3):89–113, 1999.

Frank H. Müller. *Stylebook for the Tübingen Partially Parsed Corpus of Written German (TüPP-D/Z)*. Seminar für Sprachwissenschaft, Universität Tübingen, Tübingen, January 15 2004. URL `http://www.sfs.uni-tuebingen.de/tupp/dz/stylebook.ps`.

Anne Schiller. DMOR Benutzerhandbuch. Technical report, IMS, Universität Stuttgart, 1995.

Anne Schiller, Simone Teufel, Christine Thielen, and Christine Stöckert. *Guidelines für das Taggen deutscher Textcorpora mit STTS*. IMS Stuttgart und SfS Tübingen, Stuttgart und Tübingen, 1995. URL `http://www.sfs.uni-tuebingen.de/Elwis/stts/stts-guide.ps.gz`.

Tylman Ule and Frank Henrik Müller. KaRoPars: Ein System zur linguistischen Annotation großer Text-Korpora des Deutschen. In Alexander Mehler and Henning Lobin, editors, *Automatische Textanalyse. Systeme und Methoden zur Annotation und Analyse natürlichsprachlicher Texte*, Opladen, 2004. Westdeutscher Verlag.

Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation. W3C, 2000. URL `http://www.w3.org/TR/2000/REC-xml-20001006`.