Der TUSNELDA Annotationsstandard

Laura Kallmeyer, Andreas Wagner
Projekt C1

Kolloquium des SFB 441, 8. Mai 2000

1 Einleitung

TUSNELDA = Tübinger Sammlung nutzbarer empirischer linguistischer Datenstrukturen

Die im SFB gesammelten Korpora sollen in TUSNELDA zusammengefasst werden und, soweit dies möglich ist, auch über den SFB hinaus verfügbar gemacht werden.

Übersicht über die (bisher vorgesehenen) Korpora in TUSNELDA:

- Projekt A1:
  deutsche Zeitungskorpora
  Aufbereitung: POS-Tagging, Chunking, später syntaktische Annotation
- Projekt B1:
  russische Korpora
  Aufbereitung: POS-Tagging
- Projekt B3:
  deutsche historische Korpora
  Aufbereitung: Annotation bis auf Satzebene
- Projekt B8:
  Texte und Bilder (Comics)
  Aufbereitung: Verbindung von Texten und Bildern, Kennzeichnung von Deiktika, Klassifizierung von Zeigegesten
- Projekt B9:
  spanische Korpora gesprochener Sprache, Fernsehübertragung eines Fußballspiels
  Aufbereitung: Segmentierung nach Geschehenseinheiten, Verknüpfung mit Beschreibungen von Geschehenseinheiten und mit einzelnen digitalisierten Bildern

Damit die gesammelten Daten über die konkrete Anwendung im SFB hinaus wiederverwendbar sind und mit standardisierten Tools bearbeitet werden können, müssen sie in einer einheitlichen und allgemeinen Standards (SGML, XML) entsprechenden Weise aufbereitet werden. Daraus ergibt sich die Notwendigkeit eines für den SFB verbindlichen Annotationsstandards für TUSNELDA.

Der TUSNELDA Annotationsstandard wurde im wesentlichen in einem vom Projekt C1 veranstalteten Workshop erarbeitet, an dem Mitarbeiter des SFB und des SfS teilnahmen.

2 Kurzeinführung in SGML

SGML (= Standard Generalized Markup Language): internationaler Standard zur expliziten Kodierung impliziter Eigenschaften von Dokumenten, insbesondere

- die interne Struktur von Texten (z. B. Abschnitt- und Satzgrenzen)
- klassifikatorische Merkmale von Texten (z. B. Entstehungszeit)

2.1 Strukturierungselemente in SGML: Marken (Tags)

Beispiel (Schwäbisches Tagblatt):

```
<s id=457>Mit sieben Bier in einer guten Stunde w&auml;re der Hundehalter nach
eigener Aussage <q>leicht angetrunken gewesen</q></s>
```

- markierter Text-Bestandteil (Element) wird umrahmt von Start-Tag <*tagname*> und End-Tag </*tagname*>
- *tagname* bezeichnet die Art des markierten Elements
- Attribute im Start-Tag kodieren weitere Eigenschaften bzw. Referenzpunkte

2.2 Festlegung der Markup-Struktur: Document Type Definitions (DTDs)

Eine Document Type Definition (DTD) definiert die Struktur eines Dokuments. Eine DTD enthält für jedes Element eine Regel, die Folgendes festlegt:

- seine Unterelemente und deren Anordnung
- seine Attribute

Beispiel:

```
<!ELEMENT sp          - -   (speaker*, (spokenPar | stage)+)   >
<!ATTLIST sp                who        NAME        #IMPLIED  >

<!ELEMENT spokenPar   - -   (#PCDATA | stage)*                 >

<!ELEMENT speaker     - -   (#PCDATA)                          >

<!ELEMENT stage       - -   (#PCDATA)                          >
```

```
<sp><speaker>Lord Windermere.</speaker>
    <spokenPar><stage>In a low voice.</stage>
        It was terribly rash of you to come!</spokenPar>
</sp>
```

Man kann automatisch überprüfen, ob ein Text gemäß seiner DTD korrekt strukturiert ist.

2.3 XML

XML (= eXtensible Markup Language): neuer Standard für die Kodierung von Internet-Dokumenten

- weitreichende Ähnlichkeit mit SGML
- erlaubt die Definition von Dokument-Strukturen mit DTDs (⇒ Erweiterung von HTML)
- Verarbeitungssoftware ist in der Entwicklung

Die Korpora für TUSNELDA werden mit SGML aufgebaut und später automatisch in XML konvertiert und in dieser Form elektronisch publiziert.

3 Bestehende Standards: TEI und CES

TEI (= Text Encoding Initiative): Standard zur Strukturierung von Texten mit SGML

Die TEI definiert DTDs für verschiedenste Textsorten.

Ziel: Ermöglichung standardisierter Verarbeitungsmechanismen für Dokumente; "Austauschbarkeit" von Dokumenten zwischen verschiedenen Institutionen

Nachteile von TEI aus korpuslinguistischer Sicht: Die TEI ist für eine weite Spanne von Anwendungsgebieten konzipiert (Linguistik, andere Geistes- und Sozialwissenschaften, Bibliothekswesen, Verlagswesen, etc.) Daher enthalten die TEI-Guidelines Definitionen, die für korpuslinguistische Zwecke

- nicht relevant
- zu generell
- unvollständig

sind.

Deshalb wurde von EAGLES (= Expert Advisory Group on Language Engineering Standards) der CES (= Corpus Encoding Standard) entwickelt, der auf TEI basiert, aber spezielle korpuslinguistische Bedürfnisse berücksichtigt.

CES bildet die Grundlage für den TUSNELDA-Annotstionsstandard.

4 Allgemeine Struktur eines annotierten Korpus

4.1 Das Element `tusneldaCorpus`

`<tusneldaCorpus>` ist das oberste Element in der DTD, es enthält ein gesamtes Korpus.

Ein Element `<tusneldaCorpus>` besteht aus einem Korpusheader und entweder einem oder mehr Subkorpora oder einem oder mehr Elementen `<tusneldaDoc>`. Ein Dokument `<tusneldaDoc>` wiederum besteht aus einem Textheader und einem oder mehr Texten.

Beispiel:

```
<tusneldaCorpus>
  <tusneldaHeader type='corpus'> ... </tusneldaHeader>
  <tusneldaCorpus>
    <tusneldaHeader type='corpus'> ... </tusneldaHeader>
    <tusneldaDoc>
      <tusneldaHeader type='text'> ... </tusneldaHeader>
      <text> ... </text>
      <text> ... </text>
      ...
    </tusneldaDoc>
    <tusneldaDoc> ... </tusneldaDoc>
    ...
  </tusneldaCorpus>
  <tusneldaCorpus> ... </tusneldaCorpus>
  ...
</tusneldaCorpus>
```

Dieser Aufbau ist von CES übernommen.

4.2 Aufbau eines Headers

Ein Element `<tusneldaHeader>` sowohl für Korpus- als auch Textheader.

Der Header ist eine Art "elektronische Titelseite" des Korpus oder Textes. Er enthält Informationen über Inhalt, Herkunft, Aufbereitung, Verfügbarkeit usw.

`<tusneldaHeader>` besteht aus folgenden Teilelementen:

1. `<fileDesc>`, enthält eine bibiographische Beschreibung des Korpus oder Textes und bezieht sich auf die (elektronische) Datei, die (im Falle eines Korpusheaders) das gesamte Korpus oder (im Falle eines Textheaders) einen individuellen Text enthält.

2. `<encodingDesc>`, beschreibt die Beziehung zwischen dem elektronischen Text und seiner Quelle, z.B. editorische Prinzipien, Vorgehen beim Sampling, verwendete Tags usw.

3. `<profileDesc>`, beschreibt weitere Aspekte des Textes, z.B. im Text verwendete Sprachen, Zeit und Ort der Entstehung des Ursprungstextes, evtl. vorhandene Übersetzungen usw.

4. `<revisionDesc>`, hält die Revisionsgeschichte des Korpus (der Datei) fest, d.h. enthält Informationen über alle vorgenommenen Änderungen

4.3 Aufbau eines Textes

Die Aufteilung eines Textes in Überschrift, Autoren, Schlüsselwörter, und größere Abschnitte `<div>` ist in CES bzw. TUSNELDA relativ unproblematisch und einfach nachzuvollziehen.

Beispiel:

```
<text><body>
 <div><head>Effi Briest</head>
      <byline><docAuthor>Theodor Fontane</docAuthor></byline>
      <div><head>Erstes Kapitel</head>
           <p>In Front des schon seit Kurf&uuml;rst Georg Wilhelm von der
              Familie von Briest bewohnten Herrenhauses zu Hohen-Cremmen fiel
              heller Sonnenschein auf die mittagsstille Dorfstra&szlig;e, ...
           </p>
      </div>
      ...
 </div>
</body></text>
```

5 Technisch motivierte Unterschiede zwischen TUSNELDA und CES

5.1 Angaben über die Größe eines Korpus/Textes

Das CES-Element `<extent>`, ein Teilelement der File Description im Header, gibt Auskunft über die Größe des elektronischen Textes.

Inhalt von `<extent>` in CES:

- `<byteCount>`, enthält eine Angabe über die Text/Korpusgröße in Bytes
- `<wordCount>`, enthält eine Angabe über die Anzahl Wörter in einem Text/Korpus

Problem: `<wordCount>` kann Satzzeichen mit einschließen, muss dies aber nicht. Dies könnte um einen bei einem Größenvergleich verschiedener Texte anhand des `<wordCount>` ein Problem sein. Außerdem möchte man im Idealfall Informationen über beides, die Anzahl Wörter einschließlich Satzzeichen und die Anzahl Wörter ohne Satzzeichen haben.

Lösung in TUSNELDA:

Neben `<wordCount>` weiteres Element `<tokenCount>`. `<wordCount>` zählt Wörter ohne Satzzeichen, während `<tokenCount>` Korpuspositionen, also Wörter einschließlich Satzzeichen zählt.

Weiteres Problem: Für manche (z.B. agglutinierende) Sprachen ist die Anzahl von Buchstaben in einem Text/Korpus ebenso informativ oder vielleicht sogar informativer als die Anzahl der Wörter.

Lösung in TUSNELDA:

Neues Element `<characterCount>`, das die Anzahl Buchstaben bzw. Zeichen im Text ohne Markup (Tags) angibt, wobei auch Satzzeichen gezählt werden.

5.2 Angaben zur Segmentierung

In der encoding description, dem zweiten Element des Headers, gibt es ein Teilelement `<segmentation>`, das angibt, nach welchen Prinzipien der Text in Taginhalte segmentiert wurde.

Z.B. Erkennen von Wort- und Satzgrenzen, Erkennen von Eigennamen, Erkennen von syntaktischen Chunks usw.

Problem: Das CES Element `<segmentation>` ist nicht weiter strukturiert. D.h. wenn man z.B. feststellen möchte, mit welcher Methode und mit welchem Tool ein Korpus in Sätze segmentiert wurde, muss man sich den gesamten Inhalt von `<segmentation>` ansehen und darin nach der gewünschten Information suchen.

Lösung in TUSNELDA:

Weitere Strukturierung von <segmentation>, um die Vergleichbarkeit von Korpora zu erhöhen:

<segmentation> definiert als Menge von Paaren von jeweils einem Element <tag> und einem Element <segmMethod>. Abschließend kann <segmentation> noch eine Segmentation Note <segmNote> enthalten.

Beispiel:

```
<segmentation>
 <tag>s</tag>
 <segmMethod>Automatisches Tagging mit dem Satzgrenzenerkenner
  'separate' von Helmut Schmid, IMS Stuttgart.
  separate geht wie folgt vorgeht: Wird ein Frage-
  oder Ausrufungszeichen erkannt, so ist ein Satzende erreicht.
  Wird ein Punkt erkannt, so wird &uuml;berpr&uuml;ft, ob es sich
  um einen zu einer Abk&uuml;rzung oder Ordnungszahl geh&ouml;renden
  Punkt handelt. Ist dies nicht der Fall, so ist eine Satzgrenze
  erkannt. Das Programm greift auf eine Liste von Abk&uuml;rzungen
  zu. Wenn die Zeichenkette zwischen der letzten Leerstelle und dem
  erkannten Punkt in der Abk&uuml;rzungsliste steht, wird abgenommen,
  dass es sich um eine Abk&uuml;rzung und kein Satzende handelt.
  Steht unmittelbar vor dem Punkt eine Ziffer und geht dieser eine
  Leerstelle voraus, so wird angenommen, dass es sich um eine
  Ordnungszahl und kein Satzende handelt.
 </segmMethod>
</segmentation>
```

5.3 Annotation von Dialogen, Drama etc.

In CES gibt es ein Element <sp> (speech), das ein Unterelement <stage> für Regieanweisungen enthält. <stage> kann beliebig tief in <sp> eingebettet sein, also als direktes Unterelement oder als Unterelement eines Unterelementes von <sp> usw. auftreten.

Problem: Diese Einbettung beliebiger Tiefe kann man zwar mit SGML, nicht aber mit XML ausdrücken.

Um diesem Problem zu entgehen, ist in XCES (der XML-Anpassung von CES) <stage> nur als Element auf Paragraphenebene vorgesehen, also als direktes Unterelement von <sp> (neben den anderen Unterelementen <speaker> und <p> für Paragraph).

Probleme: Die Zusammenfassung von einem Sprecher und seinem Text in <sp> (wie in CES) ist nicht mehr gegeben. (<sp> erlaubt beliebig viele Elemente <speaker>, <p> und <stage> in beliebiger Reihenfolge.)

Außerdem lassen sich Auftreten von Regieanweisungen innerhalb von Paragraphen nicht adequat annotieren.

Lösung in TUSNELDA:

Definition eines neuen Elementes `<spokenPar>`, das wie `<p>` aufgebaut ist, aber zusätzlich `<stage>` Elemente enthalten kann.

`<sp>` neu definiert: `<sp>` enthält beliebig viele Elemente `<speaker>` gefolgt von beliebig vielen Elementen `<stage>` oder `<spokenPar>`.

(Definition siehe Bsp. in Abschnitt 2.2.)

Dies garantiert, dass mit jeder neuen Sprecherangabe (Menge von Elementen `<speaker>`) ein neues Element `<sp>` beginnt. Außerdem kann `<stage>` sowohl auf Paragraphenebene (als Unterelement von `<sp>`) als auch als Teil eines Paragraphen (als Unterelement von `<spokenPar>`) auftreten.

Beispiel:

Wedekind: Frühlings Erwachen

```
<sp>
   <speaker>Wendla</speaker>
   <spokenPar>Nicht, liebe Mutter; nicht traurig sein!</spokenPar>
</sp>
<sp>
   <speaker>Frau Bergmann</speaker>
   <spokenPar><stage>sie k&uuml;ssend</stage> Mein einziges
                        Herzblatt!</spokenPar>
</sp>
```

Oscar Wilde: Lady Windermere's fan

```
<sp who="Lady Windermere">
    <speaker>Lady Windermere.</speaker>
    <spokenPar>That will do!</spokenPar>
</sp>
<sp><stage>Exit Parker C.</stage></sp>
<sp who="Lady Windermere">
    <spokenPar><stage>Speaking to Lord Windermere</stage>
       Arthur, if that woman comes here - I warn you -
    </spokenPar>
</sp>
```

6 Anpassung an projektspezifische Anforderungen

6.1 Geschachtelte Sätze

Innerhalb von Sätzen können wiederum (zitierte) Sätze auftreten. Die Untersuchung dieser Sätze kann linguistisch interessant sein.

Problem: Um geschachtelte Sätze automatisch extrahieren zu können, müssen sie explizit markiert werden.

Lösung in TUSNELDA:

Einführung eines Attributs `nested` für `<s>`

mögliche Werte: `yes` (für ein Element `<s>`, das ein anderes Element `<s>` enthält) und `no` (Default)

Beispiel:

```
<s nested="yes">
   <q id=q1 broken="yes" next=q2>
     <s id=s1 broken="yes" next=s2 nested="no">Drau&szlig;en</s>
   </q>, sprach er,
   <q id=q2 broken="yes" prev=q1>
     <s id=s2 broken="yes" prev=s1 nested="no">ist es kalt.</s>
   </q>
</s>
```

6.2 Sprachen in TUSNELDA

Im Header von TUSNELDA, seiner Teilkorpora und der einzelnen Texte werden, wie in CES, mit dem Element `<langUsage>` die Sprachen angegeben, die in den jeweiligen Texten bzw. Korpora vorkommen. Hierfür wird in CES u. a. der Standard ISO 639 bzw. ISO 639-2 verwendet, der Kürzel für Sprachen definiert (z. B. en, de, fr,...). Diese Kürzel werden als Wert des `<language>`-Attributs `iso639` kodiert.

Problem: Der ISO-Standard umfasst möglicherweise nicht alle Sprachen und Dialekte, die in TUSNELDA vorkommen werden.

Lösung in TUSNELDA:

Zusätzlich kann ein weiterer Standard verwendet werden, der wesentlich mehr Sprachen und Dialekte erfasst: Ethnologue. Ethnologue definiert Kürzel für 6703 Sprachen und Dialekte (ISO 639-2 für 460 Sprachen).

Einführung eines optionalen Attributs `ethnologue` für `<language>`; Beibehaltung des (obligatorischen) Attributs `iso639` wegen der großen Verbreitung des ISO-Standards

Beispiel:

```
<langUsage>
    <language id="fr" iso639="fr">French</language>
    <language id="en" iso639="en">English</language>
    <language id="la" iso639="la">Latin</language>
    <language id="swg" iso639="de" ethnologue="swg">Swabian</language>
</langUsage>
```

6.3 Entstehungszeit und -ort diachroner Texte

Für diachrone Texte ist es wichtig, auf Ort und Zeitraum der Entstehung zugreifen zu können. Diese Angaben werden in der *profile description* mit dem Element `<creation>` kodiert.

Problem: Oft ist es nicht möglich, das exakte Entstehungsdatum historischer Texte zu bestimmen, sondern man kann es nur zeitlich eingrenzen. In CES hat `<creation>` jedoch nur das Attribut `date` zur Angabe der Entstehungszeit; damit lassen sich der früheste und der späteste mögliche Entstehungszeitpunkt nicht explizit spezifizieren. Für den Entstehungsort sieht CES gar kein Attribut vor.

Lösung in TUSNELDA:

Ersetzen des Attributs `date` durch die Attribute `earliest` und `latest` für den frühesten bzw. spätesten möglichen Entstehungszeitpunkt; Einführung des Attributs `place` für den Entstehungsort

Beispiel:

```
<creation earliest=1066 latest=1127 place=England>...</creation>
```

6.4 Korrekturen der Quellen

Die Texte, die Korpora zugrunde liegen, können Fehler und layout-spezifische Gegebenheiten enthalten (z. B. Rechtschreibfehler, Wort-Trennung am Zeilenende), die für korpuslinguistische Zwecke (Suche nach Wortforman) von Nachteil sind. Deshalb sollen diese Fehler in TUSNELDA korrigiert werden. Diese Korrekturen sollen jedoch explizit markiert werden, und die Originalform soll in einem Attribut angegeben werden, damit keine Information verloren geht.

Beispiel:

```
...werden die <corr sic="miesten">meisten</corr> Rechtschreibfehler...
```

Problem: Bei historischen Texten kann es vorkommen, dass bestimmte Text-Teile durch Verunreinigung oder Verwitterung nicht oder nicht eindeutig entziffert werden können. In solchen Fällen ist eine sichere Korrektur nicht möglich. Das Gleiche gilt bei Transkriptionen von gesprochener Sprache mit Nebengeräuschen.

Lösung in TUSNELDA:

Einführung des in TEI, aber nicht in CES verwendeten Elements `<unclear>`, das in entsprechendenen Fällen an Stelle von `<corr>` verwendet wird

Beispiele:

```
<unclear cause="ink blot" resp=LB>The</unclear> sea between
yet hence his pray'r prevail'd

and then <unclear cause='passing truck'>marbled queen</unclear>
```

7 Internetadressen

TEI Guidelines for Electronic Text Encoding and Interchange (P3):
`http://etext.virginia.edu/TEI.html`

Corpus Encoding Standard (CES) Guidelines und DTD:
`http://www.cs.vassar.edu/CES/`

XCES (XML Version von CES):
`http://www.cs.vassar.edu/XCES/`

Ergebnisbericht des Workshops "Annotationsstandard" im November 1999:
`http://www.sfb441.uni-tuebingen.de/c1/workshop-tusnelda.html`

Übersicht über die anschließend noch vorgenommenen Modifikationen im Diskussionsforum des SFB:
`http://gros2.sfb.uni-tuebingen.de/intern/messages/28.html`

DTDs für TUSNELDA (einschließlich der Änderungen, die erst im Anschluss an den Workshop vorgenommen wurden):
`http://www.sfb441.uni-tuebingen.de/c1/tusneldaHeader.elt`
und
`http://www.sfb441.uni-tuebingen.de/c1/tusneldaDoc.dtd`

Ethnologue: Languages of the World
`http://www.sil.org/ethnologue`