

Formale Syntax Tutorium

Besprechung der Aufgabe 06

Danny Rehl

Institut für Computerlinguistik
Ruprecht-Karls-Universität Heidelberg

16.06.2011





letztes Tutorium

- Besprechung der **Aufgabe 04**
- **Lexical uncertainty**
- **Equi-** & **Raising**verben

heute

- Besprechung der **Aufgabe 06**

Aufgabe 06 - Teilaufgabe 1



zu modellierende Sätze

- 1 er singt das Lied.
- 2 er singt es.
- 3 die Lieder gefallen ihnen.
- 4 * ihm singt ihn.
- 5 * der Lied singt ihn.



erste Analyse

```
% analyze-string {er}
```

```
{er "+Token" | sie "+PPRO" ".Pers" ".3" ".Sg" ".Masc" ".Nom" ".None"}
```

Probleme:

- 1 Tag +PPRO fehlt
- 2 Sublexikalische Regel fehlt
- 3 Morphosyntaktisches Feature .Pers fehlt



Lösungen:

- 1 +PPRO PRON-T XLE .
- 2 PRON →
PRON-S_BASE
PRON-T_BASE
Type-F_BASE “.Pers”
Pers-F_BASE “.3”
Num-F_BASE “.Sg”
Gend-F_BASE “.Masc”
Case-F_BASE “.Nom”
Decl-F_BASE “.None”
- 3 .PERS Type-F XLE .



jetzt funktioniert

- parse “PRON: er”
- parse “er singt das Lied.” (Satz 1)
- parse “er singt es.” (Satz2)



erste Analyse

```
% analyze-string {Lieder}  
{ {Lieder|lieder} "+Token" | Lied "+NN" ".Neut" ".NGA" ".PI" }
```

Problem:

- 1 Morphosyntaktisches Feature .NGA fehlt

Lösung:

- 1 .NGA Case-F XLE
 { @(CASE nom)
 | @(CASE gen)
 | @(CASE acc) }.



jetzt funktioniert

- parse “N: Lieder”
- parse “NP: die Lieder”
- parse “die Lieder gefallen ihnen.” (Satz3)

parst ordnungsgemäß nicht

- parse “ihm singt ihn.” (Satz 4)
- parse “der Lied singt ihn.” (Satz 4)



zu modellierende Sätze

- 1 er singt dieses Lied.
- 2 der Student singt diese.
- 3 der Student singt diesen.
- 4 * er singt diesen Lied.
- 5 * der Student singt diesem.



erste Analyse

```
% analyze-string {dieses}
```

```
{ dieses "+Token" | diese "+DEM" ".Pro" {".Neut" ".NA"|".MN" ".Gen"} ".Sg" ".St"}
```

Probleme:

- 1 Tag +DEM fehlt
- 2 Sublexikalische Regel fehlt
- 3 Morphosyntaktisches Feature .Pro fehlt
- 4 Lunkown-Behandlung fehlt
- 5 In der C-Struktur-Regelmenge fehlt DEM



Lösung:

- 1 +DEM DEM-T XLE .
- 2 DEM →
DEM-S_BASE
DEM-T_BASE "+DEM"
Type-F_BASE ".Pro"
Gend-F_BASE ".Neut"
Case-F_BASE ".NA"
Num-F_BASE ".Sg"
Decl-F_BASE ".St"
- 3 .Pro Type-F XLE .
- 4 DEM-S XLE { (↑SPEC)=%stem | (↑PRED)='pro' }
- 5 NP → { (DEM) N | DEM: (↑PRED) = 'pro' }



jetzt funktioniert

- parse “DEM: dieses”
- parse “DEM: diesen”
- parse “DEM: diese”
- parse “DEM: diesem”
- parse “er singt dieses Lied.” (Satz 1)
- parse “der Student singt diese.” (Satz 2)
- parse “der Student singt diesen.” (Satz 3)

parst ordnungsgemäß nicht

- parse “er singt diesen Lied.” (Satz 4)
- parse “der Student singt diesem.” (Satz 4)



“des” taucht nicht als Vollform im Lexikon auf. Deshalb nimmt der Parser das Wort als Lunknown auf und probiert alle ihm gegebenen Möglichkeiten durch. Von Lunknown können Verben (V-S), Adverbien (ADV-S), Präpositionen (PREP-S), Pronomen (PRON-S) oder Artikel (DET-S) potenziell aufgenommen werden. Der Parser macht auch nichts anderes als die Möglichkeiten durchzugehen und zu schauen, ob die Part-of-Speech tags mit ihren morphologisch möglichen features eine sinnvolle Analyse erlauben.

So ist beispielsweise bei analyze-string {morgen} ersichtlich, dass +ADV das einzige PoS-Tag für das Lemma morgen ist. Für analyze-string {des} ist die Ausgabe des Transducers der PoS-Tag +ART mit seinen morphologischen Kodierungen.

In den sublexikalischen Regeln wird darauf wieder Bezug genommen. So ist dort z.B. definiert, dass ein Adverb über ADV-T_BASE das PoS-Tag +ADV haben darf, aber von einem DET-T, was zu +DET führen würde, ist dort nichts erwähnt. So wird die falsche Interpretation von parse {ADV: des} ausgeschlossen.



erste Analyse

```
% analyze-string {an}
```

```
an {"+VPRE" | "+CIRCP" | "+Token" | "+PREP" ".DA"}
```

Probleme:

- 1 Tag +PREP fehlt
- 2 Morphosyntaktisches Feature .DA fehlt
- 3 Sublexikalische Regeln für P fehlen.



Lösungen:

- 1 +PREP PRON-T XLE .
- 2 .DA Case-F XLE { @(CASE dat) | @(CASE acc) }.
- 3 P →
PREP-S_BASE
PREP-T_BASE "+PREP"
Case-F_BASE ".DA"



jetzt funktioniert

- parse “PREP: an”
- parse “PREP: bei”
- parse “Maria denkt an den Baum.” (Satz 1)
- parse “Maria schläft bei dem Gewitter” (Satz2)

parst ordnungsgemäß nicht

- parse “Maria schläft an dem Baum.” (Satz 1)
- parse “Maria schläft bei das Gewitter” (Satz 2)

Aufgabe 06 - Teilaufgabe 2

(leicht abweichend von der Aufgabenstellung)



Subject-to-Subject Raising

können V-S XLE

(↑PRED)='%stem<(↑XCOMP)>(↑SUBJ)'
@(SUBJ-NOM)
(↑SUBJ) = (↑XCOMP SUBJ)
(↑XCOMP ZU-INF).

Subject-to-Object Raising

sehen V-S XLE

(↑PRED)='%stem<(↑SUBJ)(↑XCOMP)>(↑OBJ)'
@(SUBJ-NOM)
(↑OBJ) = (↑XCOMP SUBJ)
(↑XCOMP ZU-INF).



Subjekt-Kontrolle

versprechen V-S XLE

(↑PRED)='%stem<(↑SUBJ)(↑COMP)>'

@(SUBJ-NOM)

(↑COMP SUBJ PRED) = 'pro'

(↑COMP ZU-INF) =c + .

Objekt-Kontrolle

überreden V-S XLE

(↑PRED)='%stem<(↑SUBJ)(↑COMP)>(↑OBJ)'

@(SUBJ-NOM)

(↑COMP OBJ PRED) = 'pro'

(↑COMP ZU-INF) =c + .

Vielen Dank für Eure Aufmerksamkeit!

