

Probeklausur: Syntax natürlicher Sprachen

Axel Wisiosek
Centrum für Informations- und Sprachverarbeitung

21.01.2025

| | | |
|----|---|----|
| 1 | Konstituenten- und Adjunkttests | 2 |
| 2 | Konstituentengrammatik | 3 |
| 3 | X-Bar-Struktur | 4 |
| 4 | Parsingalgorithmen | 5 |
| 5 | Dependenzstruktur | 6 |
| 6 | Syntaktische Funktion | 7 |
| 7 | Komplexer Satz | 8 |
| 8 | Unifikation und Subsumption | 9 |
| 9 | Unifikationsgrammatiken | 10 |
| 10 | Statistisches Parsing | 11 |
| 11 | Datengestützte Syntaxanalyse | 13 |
| 12 | Chunk-Analysen | 14 |
| | <i>Angabe: Hilfsmittel</i> | 15 |

1 Konstituenten- und Adjunkttests

1. Permutations- und geschehens-Test

- (a) Geben Sie für folgenden Satz eine Permutation an, welche das finite Verb als Konstituente bestätigt.

Das ist ein langer Satz.

| |
|--|
| |
|--|

- (b) Gegeben sei folgender Satz, dessen drittes Satzglied den geschehens-Test besteht. Geben Sie (unter Erhalt der Wohlgeformtheit des Ausgangssatzes) ein alternatives drittes Satzglied an, so dass der geschehens-Test fehlschlägt.

Er wartet im Park.

| |
|--|
| |
|--|

2 Konstituentengrammatik

2. Temporale Ambiguität

Schreiben Sie zu dem folgenden Beispielsatz für temporale syntaktische Ambiguität ein minimale CFG, die die intendierte Struktur des Beispielsatzes erkennt.

the old (man the boat)VP (intendierte Struktur)

3 X-Bar-Struktur

3. Erweiterung einer CFG

Gegeben sei folgender Satz und eine entsprechende Grammatik.

der Hund jagt den Briefträger

| | | | | |
|---|--|-----|---|------------------------|
| 1 | | S | → | NP VP |
| 2 | | VP | → | V NP |
| 3 | | NP | → | DET N |
| 4 | | DET | → | "der" "den" |
| 5 | | N | → | "Hund" "Briefträger" |
| 6 | | V | → | "jagt" |

- (a) Erweitern Sie den Satz der Angabe um ein präpositionales Adverbial.

- (b) Erweitern Sie die Grammatik um entsprechende lexikalische und syntaktische Regeln für den erweiterten Satz. Verwenden Sie das X-Bar-Schema, so dass eine *rekursive Adjunktion* von Adverbialen an die VP ermöglicht wird.

4 Parsingalgorithmen

4. Laufzeit, indirekte Rekursion, Shift-Reduce

- (a) Geben Sie CFG-Regeln an, die bei einem Top-Down-Parser wegen Ambiguität zu längerer Laufzeit führen können.

- (b) Geben Sie CFG-Regeln an, die eine indirekte Rekursion ermöglichen.

- (c) Gegeben sei folgende Grammatik:

| | | | | | |
|---|--|-------|---|---------|----|
| 1 | | S | → | NP | VP |
| 2 | | NP | → | PROPN | |
| 3 | | NP | → | DET | N |
| 4 | | VP | → | V | NP |
| 5 | | VP | → | V | |
| 6 | | DET | → | "der" | |
| 7 | | N | → | "Hund" | |
| 8 | | PROPN | → | "Max" | |
| 9 | | V | → | "kennt" | |

Mit dieser soll ein Shift-Reduce-Parser den Satz *der Hund kennt Max* analysieren. Während des Analysevorgangs wurden bereits die ersten drei Wörter eingelesen; dies ergab den folgenden Stack-Zustand:

| |
|----|
| V |
| NP |

Der Stack-Zustand ist hier wie in der Vorlesung graphisch so dargestellt, dass neue Elemente immer oben auf den Stack kommen.

- i. Welche Parsing-Strategie verfolgt dieser Parser?

- ii. Welche Operation wird als nächstes ausgeführt?

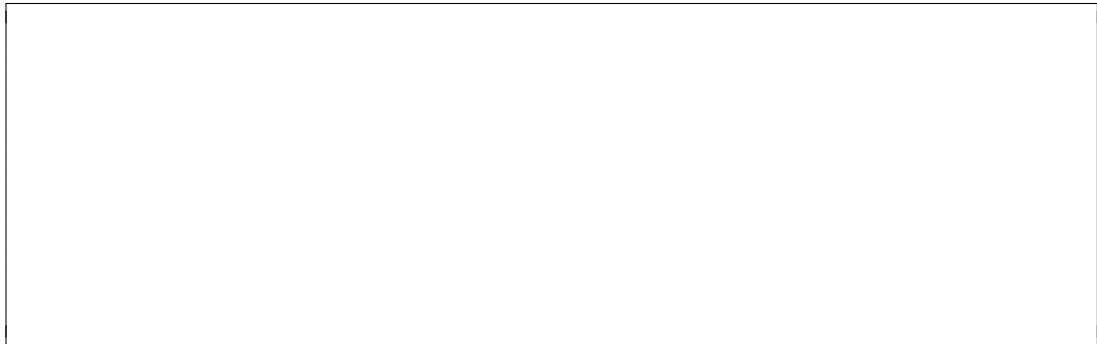
- iii. Wie sieht der Stack nach Ausführen der Operation aus?

5 Dependenzstruktur

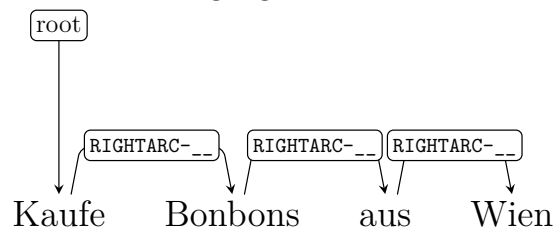
5. Dependenzgrammatik und Dependency-Parsing

- (a) Schreiben Sie zu dem folgenden Beispielsatz für syntaktische Ambiguität eine ungelabelte Dependenzgrammatik gemäß der UD-Dependenzregeln.

I shot an elephant in my pajamas



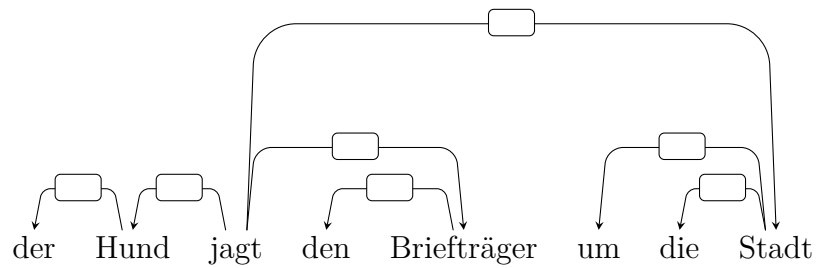
- (b) Geben Sie in folgender Dependenzanalyse die Reihenfolge der Durchführung der ‘REDUCE’-Übergänge mit einem Shift-Reduce-Dependency-Parser an.



6 Syntaktische Funktion

6. Dependenzanalyse

Vervollständigen Sie die Analyse der Dependenzbeziehungen des folgenden Satzes, indem Sie die entsprechenden UD-Label ergänzen.



7 Komplexer Satz

7. CFG- und UD-Analyse komplexer Satz

Gegeben sei folgender Satz:

der Briefträger schreibt dass der Hund den Briefträger jagt

- (a) Um welche Art komplexen Satz handelt es sich hier?

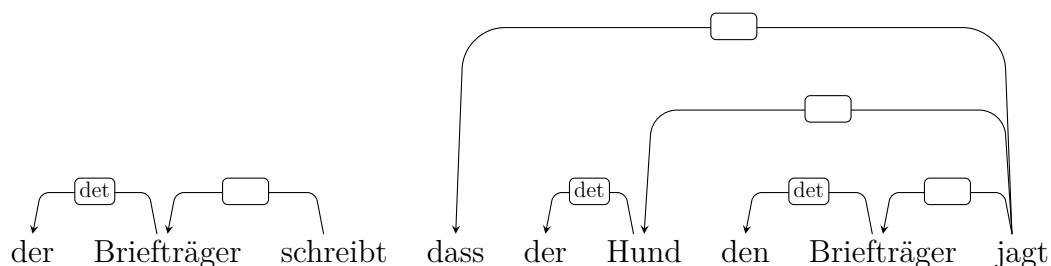
- (b) Ergänzen Sie folgende CFG um die entsprechenden lexikalischen und syntaktischen Regeln für die Modellierung des komplexen Satzes.

Verwenden Sie nur **SBAR** und **COMP** als neue Nonterminale.

Beachten Sie die invertierte Wortstellung von Verb und Objekt im Nebensatz.

| | | | |
|---|-----|---|------------------------|
| 1 | S | → | NP VP |
| 2 | VP | → | V NP |
| 3 | NP | → | DET N |
| 4 | DET | → | "der" "den" |
| 5 | N | → | "Hund" "Briefträger" |
| 6 | V | → | "jagt" "schreibt" |

- (c) Vervollständigen Sie die Analyse der Abhängigkeitsbeziehungen des komplexen Satzes, indem Sie die fehlenden UD-Label sowie Relation(en) ergänzen.



8 Unifikation und Subsumption

8. Unifikation von Merkmalsstrukturen und Typhierarchie

- (a) Gegeben sei folgende Merkmalsstruktur. Geben Sie eine nicht-leere Merkmalsstruktur an, die *nicht* mit der gegebenen unifiziert.

$$\left[\begin{array}{ll} \text{CASE} & \textit{nom} \\ \text{AGR} & \left[\begin{array}{ll} \text{GEN} & \textit{mask} \\ \text{PERS} & \textit{1} \end{array} \right] \end{array} \right]$$

- (b) Gegeben sei folgende Typhierarchie, die mit abgekürztem Typnamen durch ein **CASE**-Feature implementiert wird.

$\perp \sqsubseteq \text{Genitiv}$

$\perp \sqsubseteq \text{nicht-Genitiv}$

$\text{nicht-Genitiv} \sqsubseteq \text{Nominativ-Akkusativ}$

$\text{nicht-Genitiv} \sqsubseteq \text{Dativ}$

$\text{Nominativ-Akkusativ} \sqsubseteq \text{Nominativ}$

$\text{Nominativ-Akkusativ} \sqsubseteq \text{Akkusativ}$

- i. Zeichnen Sie die Beziehungen der Typhierarchie als Baumstruktur.
- ii. Geben Sie zu folgender Merkmalstruktur **f1** eine (nicht-leere) Merkmalstruktur **f2** an, sodass gilt: **f2** subsumiert **f1**.

$$\mathbf{f1} = \left[\begin{array}{ll} \text{*CASE*} & \textit{Nominativ-Akkusativ} \end{array} \right]$$

9 Unifikationsgrammatiken

9. Modellierung von Subkategorisierung, Rektion und Agreement

Betrachten Sie folgenden Ausschnitt aus einer merkmalsbasierten Grammatik für einen kleinen Ausschnitt des Englischen.

Beantworten Sie untenstehende Fragen und geben Sie jeweils die Zeilennummern an, auf die sich Ihre Antwort bezieht.

```
1 | S -> NP [NUM=?n] VP [NUM=?n]
2 |
3 | NP [NUM=?n] -> N [NUM=?n]
4 | NP [NUM=?n] -> PropN [NUM=?n]
5 | NP [NUM=?n] -> Det [NUM=?n] N [NUM=?n]
6 | NP [NUM=pl] -> N [NUM=pl]
7 |
8 | VP [TENSE=?t, NUM=?n] -> IV [TENSE=?t, NUM=?n]
9 | VP [TENSE=?t, NUM=?n] -> TV [TENSE=?t, NUM=?n] NP
```

(a) Wie wird hier Subkategorisierung modelliert?

(b) Nennen Sie eine alternative Modellierung von Subkategorisierung.

(c) Wie wird hier das Subjekt-Verb-Agreement modelliert?

(d) Geben Sie das entsprechende Constraint für das Subjekt-Verb-Agreement mittels einer Gleichung an (z.B. als Pfadgleichung der Form $CAT1@Feat = CAT2@Feat$).

10 Statistisches Parsing

10. PCFG und Mini-Treebank

- (a) Geben Sie zu folgendem PCFG-Parsing die Berechnung für die Ableitungswahrscheinlichkeit in Zeile 15 an (** = unkenntlich gemacht).

```

1 grammar = nltk.PCFG.fromstring("""
2     S    -> NP VP                [1.0]
3     VP   -> TV NP                [0.4]
4     VP   -> IV                   [**]
5     VP   -> DatV NP NP           [0.3]
6     TV   -> 'saw'                [1.0]
7     IV   -> 'ate'                [1.0]
8     DatV -> 'gave'                [1.0]
9     NP   -> 'telescopes'         [0.8]
10    NP   -> 'Jack'                [0.2]
11    """)
12 viterbi_parser = nltk.ViterbiParser(grammar)
13 for tree in viterbi_parser.parse(['Jack', 'saw', 'telescopes']):
14     print(tree)
15 (S (NP Jack) (VP (TV saw) (NP telescopes))) (p=0.064)

```

- (b) Welchen Wert muss in der PCFG aus der ersten Teilaufgabe das Gewicht für die Regel **VP** → **IV** haben?

- (c) Gegeben sei folgende Mini-Treebank mit PP-Attachment-Sätzen. Welches Attachment bevorzugt die daraus induzierte Grammatik? Begründen Sie.

```

1 (S
2     (NP Ich)
3     (VP (V gehe))
4     (PP auf dem Weg)
5 )
6
7 (S
8     (NP Ich)
9     (VP (V steige) (PP auf den Berg))
10 )
11

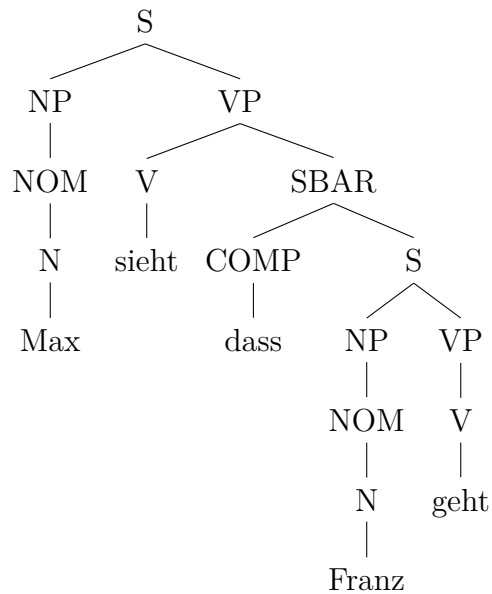
```

```
12 | (S
13 |     (NP Ich)
14 |     (VP (V klettere) (PP auf den Berg))
15 | )
16 |
17 | (S
18 |     (NP Ich)
19 |     (VP (V laufe))
20 |     (PP auf dem Weg)
21 | )
22 |
23 | (S
24 |     (NP Ich)
25 |     (VP (V renne))
26 |     (PP auf dem Weg)
27 | )
```

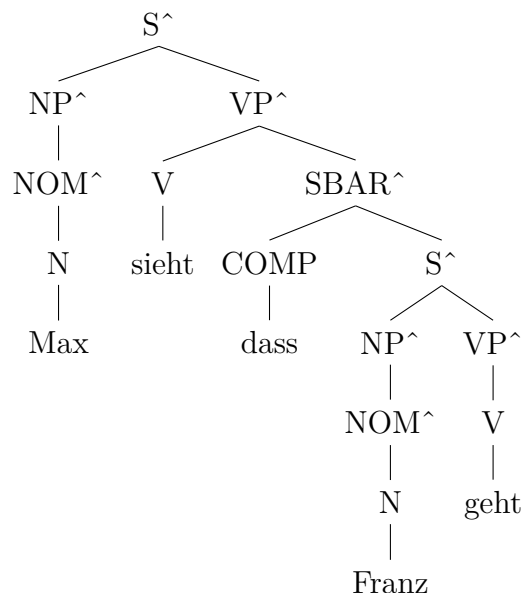
11 Datengestützte Syntaxanalyse

11. Annotation für history-based sowie lexikalisierte PCFG

Gegeben sei folgender Syntaxbaum:



- Führen Sie zunächst für die syntaktischen Regeln eine Parent-Annotation im Syntaxbaum durch (mit \wedge als Trennerzeichen).
- Nehmen Sie anschließend für die syntaktischen Regeln eine Kopfannotation im Syntaxbaum vor (mit $:$ als Trennerzeichen).



12 Chunk-Analysen

12. IOB-Tagging und Evaluationsmetriken

- (a) Markieren Sie alle Nominalphrasen (NPs), indem Sie den folgenden Satz nach dem IOB-Tagging-Schema annotieren; verwenden Sie nur folgende Label: B-NP, I-NP, O.

| Token | Der | kleine | Hund | bringt | ihm | einen | Knochen | . |
|-------|-----|--------|------|--------|-----|-------|---------|---|
| Tag | | | | | | | | |

- (b) Berechnen Sie Accuracy, Precision und Recall für folgende korrekte Annotationen (**truth**) und folgende Hypothesen (**predict**). Geben Sie bitte jeweils Brüche an.

| Sample | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------------|----|----|----|----|---|----|----|----|----|----|
| truth | PP | PP | PP | O | O | O | O | PP | O | PP |
| predict | PP | O | O | PP | O | PP | PP | O | PP | PP |

Accuracy: _____

Precision (für die Klasse PP): _____

Recall (für die Klasse PP): _____

Angabe: Hilfsmittel**Universal Dependency Relations**

| | Nominals | Clauses | Modifier words | Function Words |
|--|---|--|---|---|
| Core arguments | <u>nsubj</u> <u>obj</u> <u>iobj</u> | <u>csubj</u> <u>ccomp</u> <u>xcomp</u> | | |
| Non-core dependents | <u>obl</u> <u>vocative</u> <u>expl</u> <u>dislocated</u> | <u>advcl</u> | <u>advmod</u> * <u>discourse</u> | <u>aux</u> <u>cop</u> <u>mark</u> |
| Nominal dependents | <u>nmod</u> <u>appos</u> <u>nummod</u> | <u>acl</u> | <u>amod</u> | <u>det</u> <u>clf</u> <u>case</u> |
| Coordination | MWE | Loose | Special | Other |
| <u>conj</u> <u>cc</u> | <u>fixed</u> <u>flat</u> <u>compound</u> | <u>list</u> <u>parataxis</u> | <u>orphan</u> <u>goeswith</u> <u>reparandum</u> | <u>punct</u> <u>root</u> <u>dep</u> |

X-Bar Grundregeln

- 1 NP → NOM | DET NOM
 - 2 NOM → ADJP NOM | NOM PP
 - 3 NOM → N | N NP
-
- 1 VP → VERBAL | AUX VERBAL
 - 2 VERBAL → VERBAL PP | VERBAL ADJP
 - 3 VERBAL → V | V NP | V NP NP

SBAR Grundregeln

- 1 SBAR → COMP S
- 2 S → SBAR VP
- 3 VP → V SBAR
- 4 S → NP VP SBAR
- 5 NP → NP SBAR

Koordination Grundregeln

- 1 S → S CC S
- 2 NP → NP CC NP