

12 Datengestützte Syntaxmodelle

12.1 Induzierte PCFG-Modelle

- 12.1.1 *grammar induction* aus Treebank
- 12.1.2 Evaluation von PCFG-Modellen
- 12.1.3 Unabhängigkeitsannahmen
- 12.1.4 Lexikalisierte PCFGs (Kopfannotation)
- 12.1.5 *history-based* PCFGs (*Parent Annotation*)

12 Datengestützte Syntaxmodelle

12.1 Induzierte PCFG-Modelle

- **Grammatikentwicklung (*grammar writing*) ist aufwendig**
 - Grammatiken mit **von Experten geschriebenen Regeln** mit hoher *Abdeckung*
- **Alternative: Induktion von Grammatikregeln aus Korpora**
 - **empirisches Syntaxmodell**
 - Berücksichtigung **relativer Häufigkeiten der Regeln** ⇒ PCFG
 - als **statistisches Modell**: direkte Verwendung zur **Disambiguierung**

12.1.1 *grammar induction* aus Treebank

- **Treebank als implizite Grammatik**
 - jeder **Teilbaum** der Tiefe 1 als **implizite CFG-Regel**
 - **Expansion eines Nonterminals**
- **Extraktion von CFG-Regeln** aus den Ableitungen der Treebank
- Frequenzbestimmung der Regeln und Berechnung **Regelwahrscheinlichkeiten** über **relative Häufigkeiten** (\Rightarrow PCFG)
 - **Gewichtung** insbesondere **bei induzierter Grammatik notwendig**: viele Regeln \Rightarrow hohe Ambiguität
- Anwendung von **Smoothing und Normalisierung**

- **Form der induzierten Grammatik** hängt stark vom **Annotationschema** der dem Training des Modells zugrundeliegenden Treebank ab:
 - **flache Grammatik = viele Regel-types:**
 - Penn-Treebank: 1 Mill. *Worttokens*, 1 Mill. *nicht-lexikalische Regel-tokens*, 17.500 *Regel-types*
 - z. B. jedes PP-Adjunkt mit eigener Regel:
 $VP \rightarrow V PP, VP \rightarrow V PP PP, VP \rightarrow V PP PP PP$ usw.
 - **tieferer Bäume: mehr Nonterminale, weniger Regel-types:**
 - z. B. X-Bar:
 $VP \rightarrow V', V' \rightarrow V' PP, V' \rightarrow V$

Auflistung 1: Ableitungsbaum in Penn-Treebank

```
1 from nltk.corpus import treebank
2 t=nltk.corpus.treebank.parsed_sents('wsj_0003.mrg')[4]
3 print(t)
4 # (S
5 #   (NP-SBJ (DT A) (NNP Lorillard) (NN
6 #     spokewoman))
7 #   (VP
8 #     (VBD said)
9 #     (, ,)
10 #     (` `)
11 #     (S
12 #       (NP-SBJ (DT This))
13 #       (VP (VBZ is) (NP-PRD (DT an) (JJ old)
14 #         (NN story))))))
15 #   (. .))
```

Auflistung 2: NLTK: Extraktion von Grammatikregeln aus Treebank (*nltk.induce_pcfg*)

```
1  ##http://www.nltk.org/book/ch08-extras.html
2  ##http://www.nltk.org/_modules/nltk/grammar.html#indu
3
4  import nltk
5  from nltk.corpus import treebank
6
7  productions = []
8  S = nltk.Nonterminal('S')
9  for tree in
    nltk.corpus.treebank.parsed_sents('wsj_0003.mrg')[4
10     productions += tree.productions()
11
12  grammar = nltk.induce_pcfg(S, productions)
13  for production in grammar.productions():
14     print(production)
```

NP-SBJ → DT NNP NN [0.5]

DT → 'A' [0.333333]

NNP → 'Lorillard' [1.0]

NN → 'spokewoman' [0.5]

VP → VBD , `` S [0.5]

VBD → 'said' [1.0]

, → ',' [1.0]

`` → '``' [1.0]

S → NP-SBJ VP [1.0]

NP-SBJ → DT [0.5]

DT → 'This' [0.333333]

VP → VBZ NP-PRD [0.5]

VBZ → 'is' [1.0]

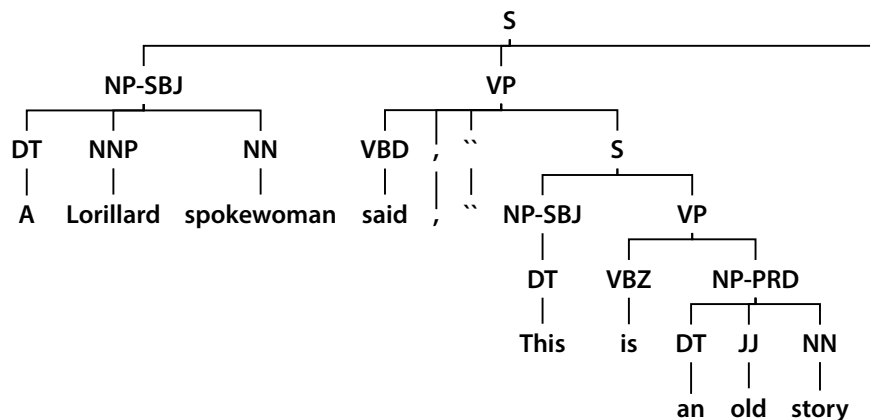
NP-PRD → DT JJ NN [1.0]

DT → 'an' [0.333333]

JJ → 'old' [1.0]

NN → 'story' [0.5]

. → '.' [1.0]



Stanford-PCFG-Parser:

- basiert auf **aus Treebanks extrahierten PCFG-Modellen**
→ <https://nlp.stanford.edu/software/lex-parser.shtml>
- Trainingskorpus des englischen Modells (`englishPCFG.ser.gz`):
Penn Treebank
- Trainingskorpus des deutschen Modells (`germanPCFG.ser.gz`):
NEGRA Korpus

12.1.2 Evaluation von PCFG-Modellen

- Messen der **Güte von Grammatikmodellen/Parsern** durch Parsen von Sätzen einer **Testmenge**
→ **Teilmenge einer hand-annotierten Treebank** = *gold-standard-*Ableitungen, z. B. von Penn-Treebank
- **PARSEVAL-Maße** (Black et al. 1991): Übereinstimmung von Konstituenten in den Ableitungen von **geparsten Daten** (**Ableitungshypothese H**) mit denen der **Test-Daten** (**Referenz-Ableitung R**)
→ Konstituente ist **korrekt** wenn Übereinstimmung in **Nichtterminal-Symbol** und **Spanne (gleicher Start- und Endpunkt)**

- **Recall** = $\frac{(\text{Anzahl von korrekten Konstituenten in Hypothese})}{(\text{Anzahl von Konstituenten in Referenz-Ableitung})}$
- **Precision** = $\frac{(\text{Anzahl von korrekten Konstituenten in Hypothese})}{(\text{Anzahl von allen Konstituenten in Hypothese})}$
 - Hypothese: (A) (B C D)
 - Referenz: (A) (B) (C) (D)
 - Recall = 1/4; Precision: 1/2
- **cross-brackets**: Anzahl an Konstituenten mit ((A B) C) in Ableitungshypothese aber (A (B C)) in Referenz-Ableitung
- **moderne Parser**: ca. **90% Precision und Recall**, ca. **1% cross-brackets**-Konstituenten (trainiert und getestet mit Penn-Treebank)

12.1.3 Unabhängigkeitsannahmen

2 Unabhängigkeitsannahmen von PCFGs

- **Annahme Unabhängigkeit von lexikalischem Material**
→ Wahrscheinlichkeiten von Teilbäumen sind unabhängig von Terminalen
- **Annahme Unabhängigkeit von Kontext**
→ Wahrscheinlichkeiten von Teilbäumen sind unabhängig von Elternknoten
- **Zurücknahme von Unabhängigkeitsannahmen:**
 - ⇒ **beschreibungsadäquatere Syntaxmodelle**
 - ⇒ **Berücksichtigung linguistischer Abhängigkeiten**

- Berücksichtigung **lexikalischer Abhängigkeiten**:
 - ⇒ ***lexikalisierte PCFGs***
 - ⇒ **Auflösung lexikalischer Ambiguität**
- Berücksichtigung **struktureller Abhängigkeiten zwischen Regeln**:
 - ⇒ ***history-based PCFGs***
 - ⇒ **Auflösung kontextabhängiger struktureller Ambiguität**

12.1.4 Lexikalisierte PCFGs (Kopfannotation)

Lexikalisierte PCFGs

- PCFGs basierend auf einfachen CFG-Regeln:
⇒ nur **strukturelle Disambiguierung**
- Probleme mit **lexikalisch determinierter Ambiguität**, z. B. bei **Subkategorisierung** oder **PP-Attachment**
- **statisches Modellierung lexikalischer Abhängigkeiten**
- bekannter lexikalisierte Parser: Collins Parser (Collins, 1999)

Vorgehen Lexikalisierung

- **bottom-up-Annotation** nichtterminaler Kategorien **mit lexikalischer Information** (Kopf-Perkolation): VP(kennt)
- auch Annotation mit **Part-of-Speech-Tag** möglich: NP(er, PRON)

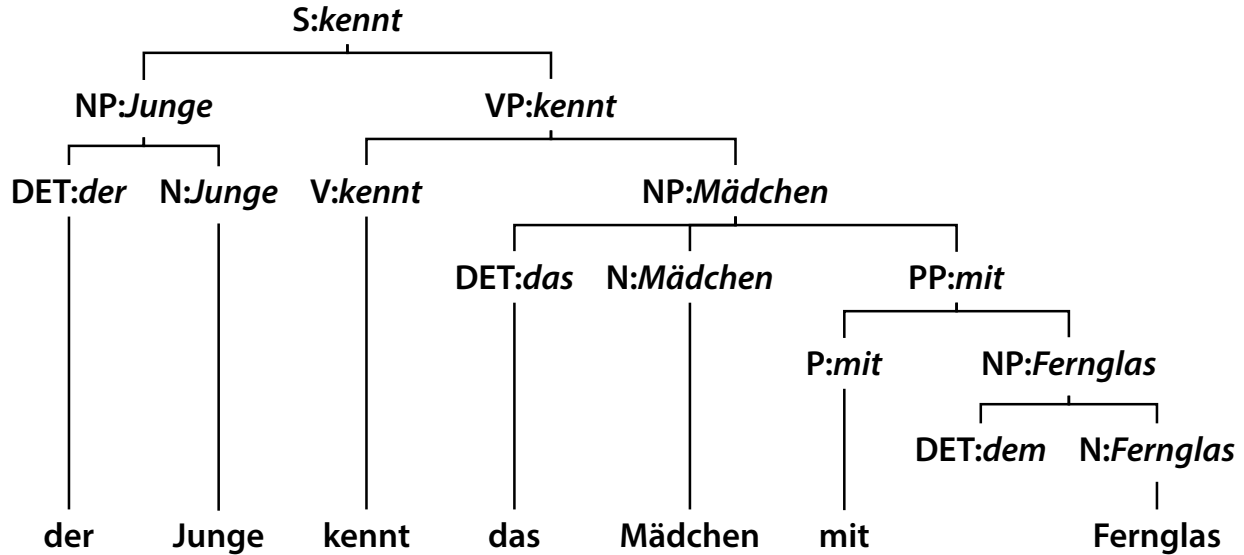
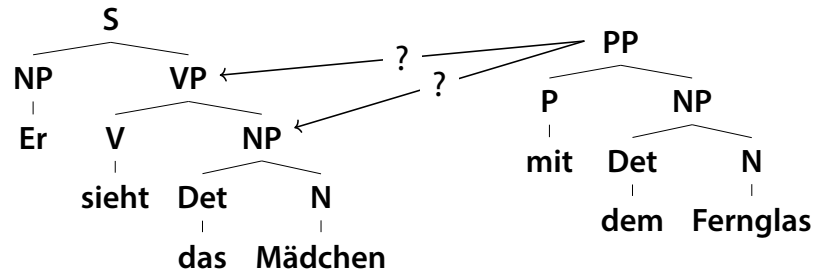


Abbildung 1: Beispiel für lexikalisierte Phrasenstruktur

PP-Attachment



- strukturelle Ambiguität:
NP- oder VP-Anbindung?
 ⇒ 2 strukturelle Lesarten:
 → (VP V (NP N PP))
 → (VP V (NP N) PP)
- **unlexikalisierte PCFG:** immer Entscheidung für eine Variante
 → z. B. **englisches Trainigskorpus: NP-Attachment-Frequenz etwas höher**

- häufig: Anbindung lexikalisch konditioniert (**lexikalische Abhängigkeit**):
 - **Bevorzugung von VP-Anbindung:** *Sie stellt die Blumen ins Wasser.*
→ engere Verbindung von *stellt* mit *ins* als zwischen *Blumen* und *ins*
 - **Bevorzugung von NP-Anbindung:** *Der Junge kennt das Mädchen mit dem Fernglas.*
→ engere Verbindung von *Mädchen* mit *mit* als zwischen *kennt* und *mit*

Subkategorisierung

- **statisches Modellierung Subkategorisierung** statt regelbasiert über Subkategorisierungsrahmen
- **transitive Verben:** hohe Wahrscheinlichkeit $P(VP \rightarrow V \ NP)$
 $\rightarrow P(V \ NP \mid VP, \text{sehen}) > P(V \mid VP, \text{sehen})$
- **intransitive Verben:** hohe Wahrscheinlichkeit $P(VP \rightarrow V)$
 $\rightarrow P(V \mid VP, \text{laufen}) > P(V \ NP \mid VP, \text{laufen})$

Probleme lexikalisierter PCFGs

- **Modell wird sehr groß**
 - Grund: **viel mehr Ereignisse** durch lexikalisierte Regeln
 - **Regelvervielfachung:**
 $VP(sieht) \rightarrow V(sieht) NP(Mädchen)$
 $VP(kennt) \rightarrow V(kennt) NP(Mädchen)$
- **umfangreiche Trainingsdaten** notwendig für Parameterabschätzung des Modells

- **neue Abschätzung** für Regelwahrscheinlichkeiten notwendig
→ **MLE-Abschätzung** über $P(\alpha \rightarrow \beta | \alpha) = \frac{\text{count}(\alpha \rightarrow \beta)}{\text{count}(\alpha)}$ ist **zu spezifisch**
→ **geht meistens gegen 0**, da **nur sehr wenige Instanzen** der lexikalisierten Regeln in Trainingskorpus vorhanden
- ***sparse data***-Problem aufgrund von in Trainingsdaten **ungesehenen Wörtern/Instanzen** (\Rightarrow keine Regel vorhanden)
→ Lösung: **Backoff** = **Verzicht auf Lexikalisierung** bei **unbekanntem lexikalischem Kopf**

- dazu notwendig: **Smoothing** (Glättung der Regelwahrscheinlichkeiten)
 - **Reservierung von Wahrscheinlichkeitsmasse** für Regeln bei Backoff bei ungesehenen Köpfen
 - Zuordnung von Wahrscheinlichkeit für Regel mit **ungesehenem Kopf**
 - z. B. **Laplace-Smoothing**: zu jeder Häufigkeit im Korpus: **Wert addieren** (1 = **Add-One-Smoothing**) \Rightarrow Backoff-Regel: $P > 0$
- **Backoff bei Collins Parser**: unbekannte Köpfe aus Testmenge und aus Trainingsmenge mit Frequenz < 6 werden mit UNKNOWN ersetzt

12.1.5 *history-based* PCFGs (*Parent Annotation*)

***history-based* PCFGs**

- Berücksichtigung **Abhängigkeit Expansion von Kontext**
 - **Regelauswahl abhängig von vorheriger Regelauswahl**
 - Wahrscheinlichkeit einer Expansion ist abhängig von der **Position im Strukturbaum**
- z. B. **unterschiedliche Expansionswahrscheinlichkeiten für NPs** in Subjekt- bzw. Objektposition
 - **Subjekt-NP** (S-dominiert) erweitert **wahrscheinlicher zu Pronomen als Objekt-NP** (VP-dominiert)
 - $P(\text{NP} \rightarrow \text{PRON} \mid \text{S}) > P(\text{NP} \rightarrow \text{PRON} \mid \text{VP})$
 - $P(\text{PRON} \mid \text{NP}, \text{S}) > P(\text{PRON} \mid \text{NP}, \text{VP})$

- Grund = **Informationsstruktur**

→ **Subjekt** typischerweise Topik = **bekannte Information**, die durch Pronomen ausgedrückt wird

	Pronomen	Nicht-Pronomen
Subjekt	91%	9%
Objekt	34%	66%

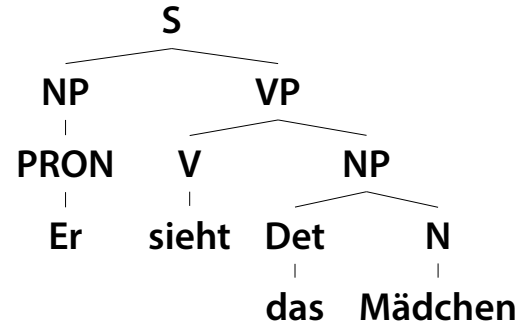


Abbildung 2: Verteilung der Form von Subjekt und Objekt in englischem Korpus (nach Francis et al., 1999, vgl. SLP2, 502)

- **erwünschte Regelgewichtung Subjekt (S-dominiert):**
NP → PRON **0.91**
NP → DET N **0.09**
- **erwünschte Regelgewichtung Objekt (VP-dominiert):**
NP → PRON **0.34**
NP → DET N **0.66**
- **normale PCFG** (keine Differenzierung, Daten aus Korpus):
NP → PRON **0.25**
NP → DET N **0.28**

- **Lösung: Splitting NP-Kategoriensymbol (*parent annotation*):**

$NP^S \rightarrow PRON \ 0.91$

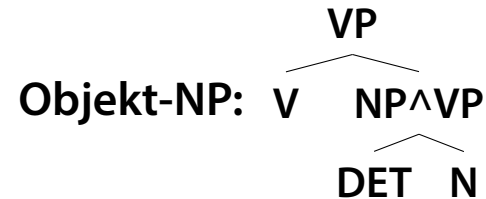
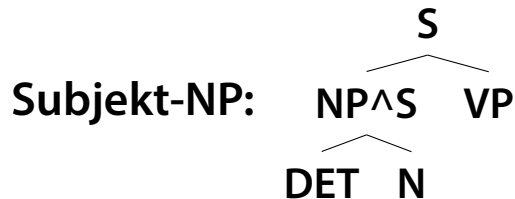
$NP^S \rightarrow DET \ N \ 0.09$

$NP^{VP} \rightarrow PRON \ 0.34$

$NP^{VP} \rightarrow DET \ N \ 0.66$

Vorgehen

- **Annotation** nichtterminaler Kategorien mit **Kategorie des Mutterknotens** (= *history*)
 ⇒ ***parent annotation***
 → Subjekt-NP: NP^{^S}
 → Objekt-NP: NP^{^VP}
 → "*Splitting* von Nicht-Terminalen"



Probleme von *history-based* PCFGs

- **ähnlich** wie bei **Lexikalisierung**, aber weniger stark ausgeprägte Regelvervielfachung durch *parent annotation*
→ *sparse data*: **unbekannte Vorgängerkategorie**
- kleinere Regelmenge durch **selektive *parent annotation***
→ **nur Splitten**, wenn ***accuracy* erhöht** wird