

# *Syntax natürlicher Sprachen*

*Tutorium*

PCFGs

Sarah Anna Uffelmann

12.01.2024

# PCFGs

= Probabilistic Context Free Grammar

= **kontextfreie Grammatik + Regelwahrscheinlichkeiten**

- **Ziel: syntaktische Disambiguierung** durch Auswahl des wahrscheinlichsten Parsebaumes (der wahrscheinlichsten Ableitung)
- Wahrscheinlichkeiten werden anhand von Korpusdaten (**Treebanks**) gelernt
- **besten Parsebaum = wahrscheinlichster Parsebaum** aufgrund der Korpusdaten
- Die Wahrscheinlichkeiten aller Regeln mit derselben linken Seite summieren zu 1 (d.h. sie ergeben eine **Wahrscheinlichkeitsverteilung**)
- Die **Wahrscheinlichkeit eines Parsebaumes** ist das Produkt seiner Regelwahrscheinlichkeiten

# PCFGs

„das Mädchen sieht das Huhn mit dem Fernglas.“

S	-> NP VP	Was ist die Regelwahrscheinlichkeit?
NP	-> Det N	(0,7)
NP	-> NP PP	Was ist die Regelwahrscheinlichkeit?
VP	-> V NP	
VP	-> VP PP	
PP	-> P NP	
Det	-> das	
Det	-> dem	
N	-> Mädchen	
N	-> Huhn	
N	-> Fernglas	
V	-> sieht	
P	-> mit	



# PCFGs

„das Mädchen sieht das Huhn mit dem Fernglas.“

S	-> NP VP	(1)
NP	-> Det N	(0,7)
NP	-> NP PP	(0,3)
VP	-> V NP	(0,6)
VP	-> VP PP	(0,4)
PP	-> P NP	(1)
Det	-> das	(0,7)
Det	-> dem	(0,3)
N	-> Mädchen	(0,4)
N	-> Huhn	(0,3)
N	-> Fernglas	(0,3)
V	-> sieht	(1)
P	-> mit	(1)

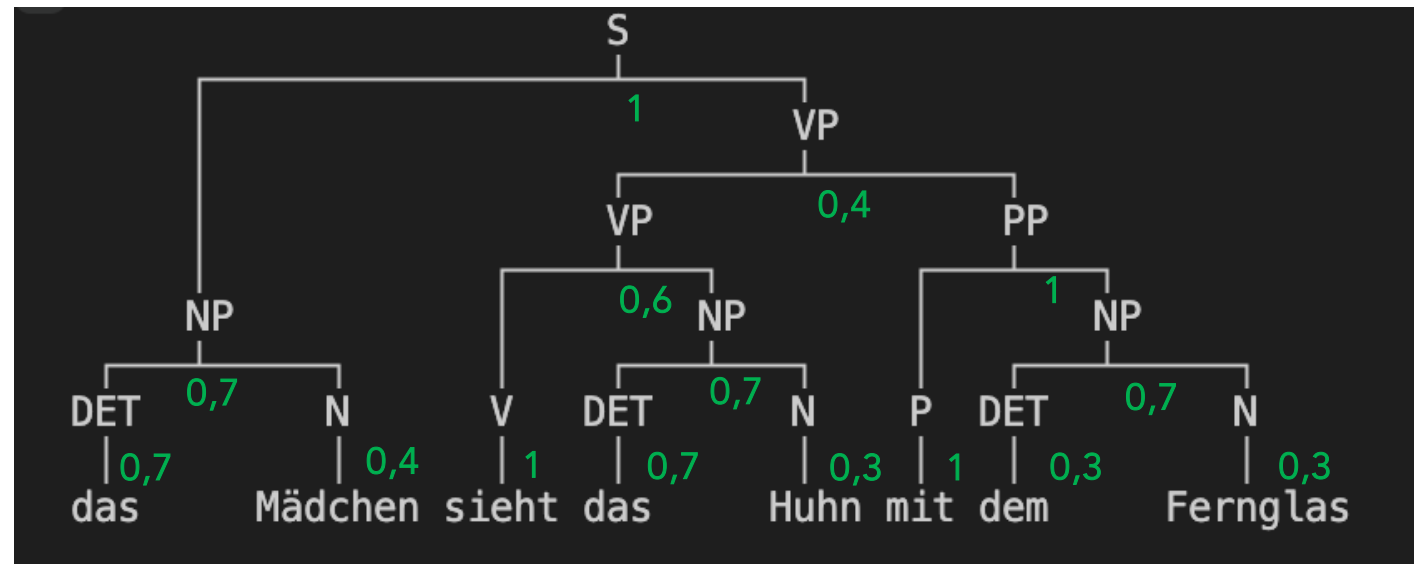
Die Wahrscheinlichkeiten aller Regeln mit derselben linken Seite summieren zu 1 (d.h. sie ergeben eine **Wahrscheinlichkeitsverteilung**)

# PCFGs

„das Mädchen sieht das Huhn mit dem Fernglas.“

S	-> NP VP	(1)
NP	-> Det N	(0,7)
NP	-> NP PP	(0,3)
VP	-> V NP	(0,6)
VP	-> VP PP	(0,4)
PP	-> P NP	(1)
Det	-> das	(0,7)
Det	-> dem	(0,3)
N	-> Mädchen	(0,4)
N	-> Huhn	(0,3)
N	-> Fernglas	(0,3)
V	-> sieht	(1)
P	-> mit	(1)

T1:



Die Wahrscheinlichkeit eines Parsebaumes ist das Produkt seiner Regelwahrscheinlichkeiten:

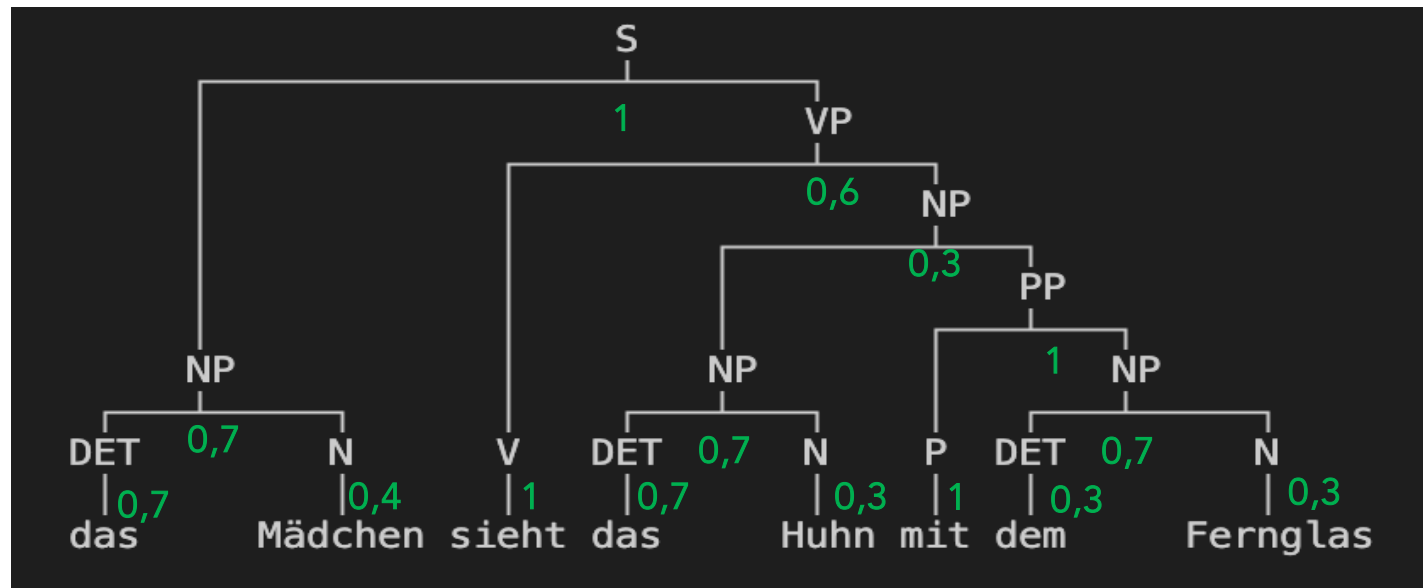
$$\begin{aligned} P(T1) &= 1 * 0,7 * 0,7 * 0,4 * 0,4 * 0,6 * 1 * 0,7 * 0,7 * 0,3 * 1 * 1 * 0,7 * 0,3 * 0,3 \\ &= 0,0004356374 \end{aligned}$$

# PCFGs

„das Mädchen sieht das Huhn mit dem Fernglas.“

S	-> NP VP	(1)
NP	-> Det N	(0,7)
NP	-> NP PP	(0,3)
VP	-> V NP	(0,6)
VP	-> VP PP	(0,4)
PP	-> P NP	(1)
Det	-> das	(0,7)
Det	-> dem	(0,3)
N	-> Mädchen	(0,4)
N	-> Huhn	(0,3)
N	-> Fernglas	(0,3)
V	-> sieht	(1)
P	-> mit	(1)

T2:



Die Wahrscheinlichkeit eines Parsebaumes ist das Produkt seiner Regelwahrscheinlichkeiten:

$$\begin{aligned} P(T2) &= 1 * 0,7 * 0,7 * 0,4 * 0,6 * 1 * 0,3 * 0,7 * 0,7 * 0,3 * 1 * 1 * 0,7 * 0,3 * 0,3 \\ &= 0,0003267281 \end{aligned}$$

# PCFGs

$$P(T1) = 0,0004356374$$

$$P(T2) = 0,0003267281$$

-> **T1 ist der wahrscheinlichere Parsebaum**

Die **Wahrscheinlichkeit des Satzes** ist die Summe der Wahrscheinlichkeiten aller seiner möglichen Ableitungen (Parsebäume).

In unserem Beispiel:  $P(S) = P(T1) + P(T2) = 0,0007623655$

**Woher erhalten wir die Regelwahrscheinlichkeiten?**

-> Sie werden aus Regelhäufigkeiten geschätzt.

Regelhäufigkeiten erhalten wir

(1) durch Zählen der Regeln in einer Treebank oder

(2) aus einem automatisch geparsten und disambiguierten Korpus

# PCFG Parsing

## Viterbi Parser

-> Gibt den **wahrscheinlichsten** Parsebaum (und **nur** diesen!) zurück

## Probabilistische Chart-Parser

- probabilistische Varianten von Chart-Parsing-Algorithmen (Earley, CYK)
- NLTK: InsideChartParser, LongestChartParser
- haben Zustandsmengen (**edge queue**), die nach unterschiedlichen Kriterien sortiert werden können:
- **lowest cost first** (bei InsideChartParser)  
Sortierung nach Wahrscheinlichkeit, findet also immer die wahrscheinlichste Ableitung
- **beam search** (bei Inside Chart Parser beam\_size definieren)  
wie lowest cost first, aber es werden nur die n wahrscheinlichsten Ableitungen behalten
- **best first search** (bei LongestChartParser)  
Sortierung nach Länge der Ableitung (hat zur Folge, dass evtl. nicht die wahrscheinlichste Ableitung an erster Stelle steht)