

# Programmierkurs Python II

Michaela Regneri

FR 4.7 Allgemeine Linguistik (Computerlinguistik)

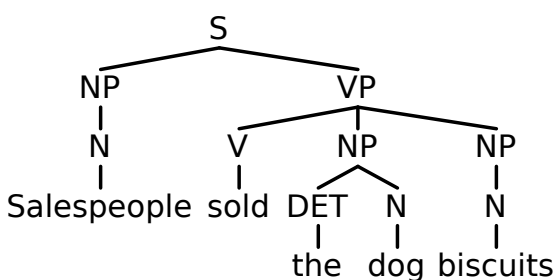
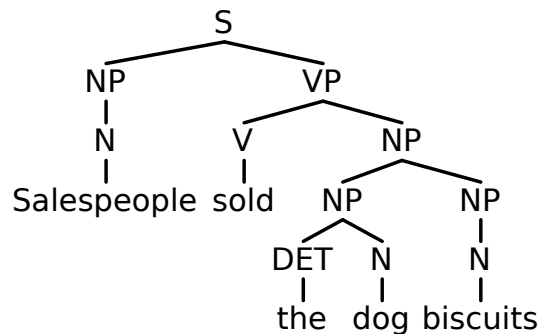
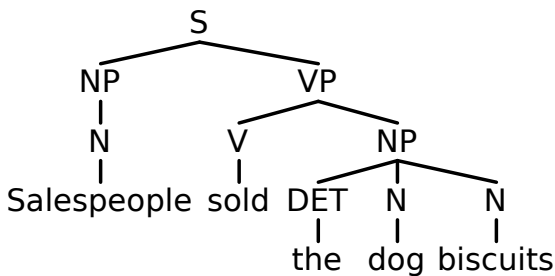
Universität des Saarlandes

Sommersemester 2013



(Charniak, 1997)

## *Salespeople sold the dog biscuits*



$S \rightarrow NP VP$	$NP \rightarrow NP NP$
$VP \rightarrow V NP$	$NP \rightarrow N$
$VP \rightarrow V NP NP$	$DET \rightarrow the$
$NP \rightarrow DET N$	$N \rightarrow dog$
$NP \rightarrow DET N N$	...

# Ambiguität & Disambiguierung

- **Probabilistische Disambiguierung**  
wenn mehrere Ableitungsbäume für eine gegebene Eingabe möglich sind ⇒ wähle den wahrscheinlichsten
- **Wir brauchen:**
  - Probabilistisches Grammatikmodell
  - Methoden zur Ermittlung von Wahrscheinlichkeiten

3

## (Weitere) Motivation

- **Natürliche Sprache ist hochgradig ambig**  
⇒ Disambiguierung
- **Grammatikentwicklung**  
⇒ Automatische Induktion von Grammatiken
- **Effiziente Suche**  
⇒ wahrscheinlichste Hypothese(n) zuerst
- **Robustheit**

4

# Probabilistische kontextfreie Grammatiken (PCFG)

- **Probabilistische kontextfreie Grammatik (PCFG)**
  - eine kontextfreie Grammatik  $\langle V, \Sigma, R, S \rangle$
  - zusammen mit einer Funktion  $P$ , die jeder Regel einen Wert  $w \in [0, 1]$  zuweist, so dass  $\sum_{\beta \in V^*} P(A \rightarrow \beta) = 1$
- $P(A \rightarrow \beta)$  ist die bedingte Wahrscheinlichkeit, dass in einer Ableitung das Symbol  $A$  zu  $\beta$  expandiert wird.
  - Alternative Notationen:  $P(\beta \mid A)$ ,  $P(A \rightarrow \beta \mid A)$ ,  $A \rightarrow \beta [p]$

5

# Ableitungsbaum (Wdh.)

- **Ableitungsbäume** einer kontextfreien Grammatik  $G$  repräsentieren Ableitungen  $S \Rightarrow^* w$ :
  - Die Wurzel ist mit dem Startsymbol beschriftet
  - Blattknoten sind mit Terminalsymbolen oder  $\epsilon$  beschriftet
  - Innere Knoten repräsentieren zusammen mit ihren Töchterknoten die bei der Ableitung verwendeten Regeln
- **Parsing:**  
Ermittlung eines oder mehrerer Ableitungsbäume
- **Probabilistisches Parsing:**  
Ermittlung des wahrscheinlichsten Ableitungsbaums

6

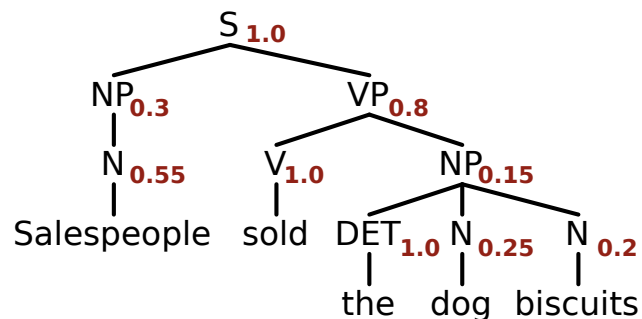
# Probabilistische kontextfreie Grammatiken (PCFG)

- Eine PCFG  $G$  weist einem Satz  $w$  einen Ableitungsbaum  $T$  mit einer bestimmten Wahrscheinlichkeit zu.
- Die **Wahrscheinlichkeit eines Ableitungsbaums  $T$**  ist das Produkt der Wahrscheinlichkeit aller Regeln, die zum Aufbau des Ableitungsbaums  $T$  verwendet wurden:
  - $P(T, w) = P(T) = \prod_{n \in T} P(R(n))$ 
    - $R(n)$  ist die Regel, mit der Knoten  $n$  expandiert wurde
    - Beachte:  $P(T, w) = P(T) P(w | T) = P(T)$ , da  $P(w | T) = 1$
- Die **Wahrscheinlichkeit eines Wortes (Satz)** ist die Summe der Wahrscheinlichkeiten seiner Ableitungen:
  - Für  $w \in L(G)$  ist  $P(w) = \sum_T P(w, T)$

7

## *Salespeople sold the dog biscuits*

$S \rightarrow NP VP$	[1.0]
$VP \rightarrow V NP$	[0.8]
$VP \rightarrow V NP NP$	[0.2]
$NP \rightarrow DET N$	[0.5]
$NP \rightarrow N$	[0.3]
$NP \rightarrow DET N N$	[0.15]
$NP \rightarrow NP NP$	[0.05]
$DET \rightarrow the$	[1.0]
$N \rightarrow Salespeople$	[0.55]
$N \rightarrow dog$	[0.25]
$N \rightarrow biscuits$	[0.2]
$V \rightarrow sold$	[1.0]

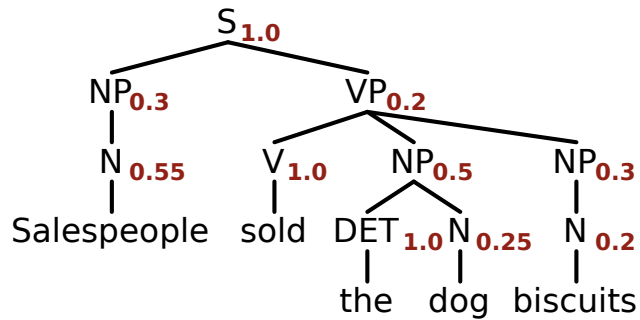


$$\begin{aligned}
 P(t) &= 1.0 \times 0.3 \times 0.55 \times \\
 &\quad 0.8 \times 1.0 \times 0.15 \times \\
 &\quad 1.0 \times 0.25 \times 0.2 \\
 &= 9.9 \times 10^{-4}
 \end{aligned}$$

8

# Salespeople sold the dog biscuits

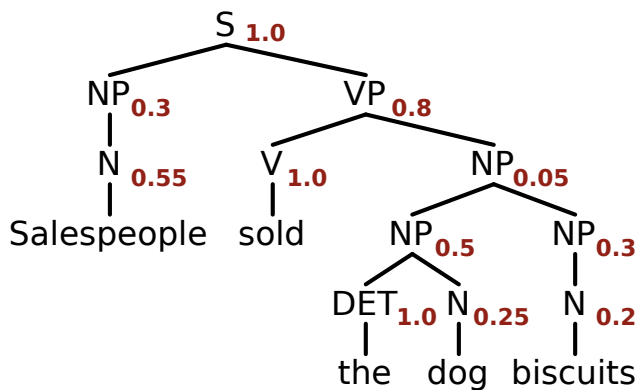
S → NP VP	[1.0]
VP → V NP	[0.8]
VP → V NP NP	[0.2]
NP → DET N	[0.5]
NP → N	[0.3]
NP → DET N N	[0.15]
NP → NP NP	[0.05]
DET → <i>the</i>	[1.0]
N → <i>Salespeople</i>	[0.55]
N → <i>dog</i>	[0.25]
N → <i>biscuits</i>	[0.2]
V → <i>sold</i>	[1.0]



$$\begin{aligned}
 P(t) &= 1.0 \times 0.3 \times 0.55 \times \\
 &\quad 0.2 \times 1.0 \times 0.5 \times \\
 &\quad 1.0 \times 0.25 \times 0.3 \times 0.2 \\
 &= 2.475 \times 10^{-4}
 \end{aligned}$$

# Salespeople sold the dog biscuits

S → NP VP	[1.0]
VP → V NP	[0.8]
VP → V NP NP	[0.2]
NP → DET N	[0.5]
NP → N	[0.3]
NP → DET N N	[0.15]
NP → NP NP	[0.05]
DET → <i>the</i>	[1.0]
N → <i>Salespeople</i>	[0.55]
N → <i>dog</i>	[0.25]
N → <i>biscuits</i>	[0.2]
V → <i>sold</i>	[1.0]



$$\begin{aligned}
 P(t) &= 1.0 \times 0.3 \times 0.55 \times 0.8 \times \\
 &\quad 1.0 \times 0.05 \times 0.5 \times 1.0 \times \\
 &\quad 0.25 \times 0.3 \times 0.2 \\
 &= 4.95 \times 10^{-5}
 \end{aligned}$$

# Probabilistische kontextfreie Grammatiken (PCFG)

- Die Wahrscheinlichkeit eines Wortes (Satz) ist die Summe der Wahrscheinlichkeiten seiner Ableitungen:
  - Für  $w \in L(G)$  ist  $P(w) = \sum_T P(w, T)$
- Eine PCFG ist **konsistent**, wenn  $\sum_{w \in L(G)} P(w) = 1$
- Rekursion kann zu inkonsistenten Grammatiken führen:
  - $S \rightarrow S S$  [0.6]
  - $S \rightarrow a$  [0.4]

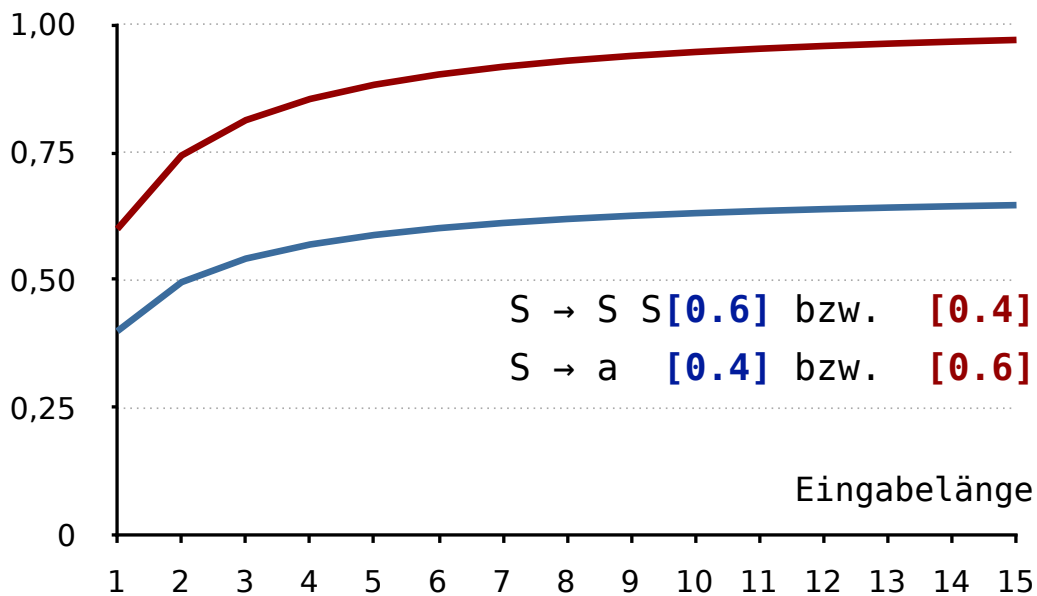
11

## Eine Inkonsistente PCFG

- $S \rightarrow S S$  [0.6] bzw. [0.4]
- $S \rightarrow a$  [0.4] bzw. [0.6]
  
- $P(a^i) = \#B\ddot{a}ume(a^i) \times 0.6^{i-1} \times 0.4^i = 0.4$ 
  - $P(a) = 0.4, P(aa) = 0.096, P(aaa) = 0.0461, \dots$
- $P(a^i) = \#B\ddot{a}ume(a^i) \times 0.4^{i-1} \times 0.6^i = 0.4$ 
  - $P(a) = 0.6, P(aa) = 0.144, P(aaa) = 0.06912, \dots$
- Anzahl Ableitungsb\ddot{a}ume f\ddot{u}r  $a^{i+1} = i$ -te Catalananzahl

12

## Eine Inkonsistente PCFG



13

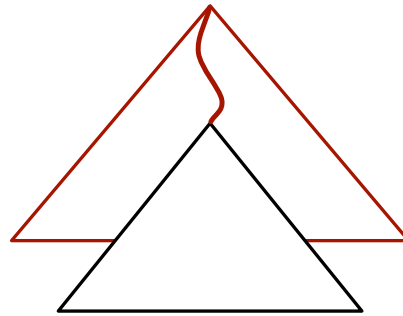
## Probabilistisches Parsing

- **Sprachmodelle („inside probabilities“)**  
ermittle zu einer gegebenen Eingabekette  $w \in L(G)$  die Wahrscheinlichkeit, dass  $S \Rightarrow^* w$ 
  - $P(w) = \sum_T P(w, T)$
- **Probabilistisches Parsing („viterbi scores“)**  
ermittle zu einer gegebenen Eingabekette  $w \in L(G)$  den wahrscheinlichsten Ableitungsbaums  $T(w)$ .
  - $T(w) = \arg \max_T P(T | w)$   
 $= \arg \max_T \frac{P(T, w)}{P(w)}$   
 $= \arg \max_T P(T)$

14

## Eigenschaften von PCFGen

- Die Wahrscheinlichkeit eines Teilbaums ist unabhängig vom Kontext in dem er vorkommt
- Die Wahrscheinlichkeit eines Teilbaums ist unabhängig von den Knoten, die ihn dominieren



15

## Probabilistisches CYK-Parsing

- Bringe die Grammatik in Chomsky-Normalform
- Erweitere den CYK-Algorithmus:
  - Speichere zusätzlich Wahrscheinlichkeiten  $T[i, j, A] \in [0, 1]$
- **Wahrscheinlichkeit einer Kette**
  - $T[i, j, A] =$  Summe der Wahrscheinlichkeiten aller Ableitungen  $A \Rightarrow^* w_{i+1} \dots w_j$
- **Wahrscheinlichkeit eines Ableitungsbaums**
  - $T[i, j, A] =$  die Wahrscheinlichkeit der wahrscheinlichsten Ableitung  $A \Rightarrow^* w_{i+1} \dots w_j$
  - $B[i, j, A]$  repräsentiert den Ableitungsbaum („Backpointer“)

16



## CYK (ohne Wahrscheinlichkeiten)

```
CYK(w = w1 ... wn):
  initialisiere T[i, j] = ∅ für alle 0 ≤ i < j ≤ n
  for i in 1 ... n:
    T[i-1, i] = { A | A → wi }
  for j in 2 ... k:
    for i in j - 2 ... 0:
      for k in i + 1 ... j - 1:
        for all A → B C:
          if B ∈ T[i, k] und C ∈ T[k, j] then
            T[i, j] = T[i, j] ∪ {A}
  return S ∈ T[0, n]
```

17

## CYK mit Wahrscheinlichkeiten (Erkenner)

```
PCYK(w = w1 ... wn):
  initialisiere π[i, j, A] = 0 für alle 0 ≤ i < j ≤ n, A ∈ N
  for i in 1 ... n:
    π[i-1, i, A] = P(A → wi)
  for j in 2 ... k:
    for i in j - 2 ... 0:
      for k in i + 1 ... j - 1:
        for all A → B C:
          if π[i, k, B] > 0 und π[k, j, C] > 0 then
            π[i, j, A] = π[i, j, A] +
              P(A → B C) * π[i, k, B] * π[k, j, C]
  return π[0, n, S]
```

Beachte: Wahrscheinlichkeit  
eines Wortes = Summe der  
Wahrscheinlichkeiten seiner  
Ableitungen

18

## CYK mit Wahrscheinlichkeiten (Parser)

PCYK( $w = w_1 \dots w_n$ ):

...

for j in 2 ... k:

for i in j - 2 ... 0:

for k in i + 1 ... j - 1:

for all  $A \rightarrow B C$ :

$pr = \pi[i, k, B] * \pi[k, j, C] * P(A \rightarrow B C)$

if  $pr > \pi[i, j, A]$  then

$\pi[i, j, A] = pr$

$B[i, j, A] = \langle \text{Backpointer aktualisieren} \rangle$

return  $\langle \text{besten Baum} + \text{Wahrscheinlichkeit } \pi[0, n, S] \rangle$

19

## Regelwahrscheinlichkeiten

- Möglichkeiten zur Ermittlung („Abschätzung“) der Regelwahrscheinlichkeiten:
  - über **relative Häufigkeiten** aus mit syntaktisch annotierten Korpora („Baumbanken“)
  - ohne annotierte Daten durch Anwendung des **Inside-Outside-Algorithmus**

20

# Regelwahrscheinlichkeiten

- **Ausgangspunkt:**  
syntaktisch annotiertes Korpus = Menge von Bäumen
- **Extraktion einer Grammatik:**  
Regeln = Menge der „Teilbäume“ der Tiefe 1
- **Abschätzung der Regelwahrscheinlichkeiten:**
  - $P(A \rightarrow \alpha) = \frac{\text{count}(A \rightarrow \alpha)}{\sum_{\beta} \text{count}(A \rightarrow \beta)}$
  - count(A → α) gibt an, wie Häufig eine Regel in allen Bäumen verwendet wurde.

21

# Regelwahrscheinlichkeiten

- Eine einfache Baumbank:
  - S1: [S [NP grass] [VP grows]]
  - S2: [S [NP grass] [VP grows] [AP fast]]
  - S3: [S [NP grass] [VP grows] [AP slowly]]
  - S4: [S [NP bananas] [VP grow]]
- Regeln & Regelwahrscheinlichkeiten:
  - S → NP VP      2/4
  - S → NP VP AP    2/4
  - NP → grass      3/4
  - ...
  -

22

## Regelwahrscheinlichkeiten

Nr.	Regel	$P(A \rightarrow \alpha)$
r <sub>1</sub>	S → NP VP	2/4
r <sub>2</sub>	S → NP VP AP	2/4
r <sub>3</sub>	NP → grass	3/4
r <sub>4</sub>	NP → bananas	1/4
r <sub>5</sub>	VP → grows	3/4
r <sub>6</sub>	VP → grow	1/4
r <sub>7</sub>	AP → fast	1/2
r <sub>8</sub>	AP → slowly	1/2

23

## Regelwahrscheinlichkeiten

- Wahrscheinlichkeiten der Beispielsätze:
  - $P(S1) = P(r_1) \cdot P(r_3) \cdot P(r_5) = 2/4 \cdot 3/4 \cdot 3/4 = 0.28125$
  - $P(S2) = P(r_2) \cdot P(r_3) \cdot P(r_5) \cdot P(r_7) = 0.140625$
  - $P(S3) = P(r_2) \cdot P(r_3) \cdot P(r_5) \cdot P(r_7) = 0.140625$
  - $P(S4) = P(r_1) \cdot P(r_4) \cdot P(r_6) = 0.03125$

24

# Evaluation

- **Coverage (Abdeckung):** Wieviele Sätze aus dem Testkorpus werden von der Grammatik als wohlgeformt erkannt?
  - Annahme: die Sätze des Korpus sind wohlgeformt
- **Accuracy (Akkuratheit):** Wieviele der laut Grammatik wahrscheinlichsten Bäume sind laut Testkorpus korrekt?
  - wird als „relative Korrektheit“ bzgl. Kategorie, Anfangs- und Endposition aller Konstituenten gemessen
  - **Labelled precision:** Anzahl korrekter Konstituenten im Ableitungsbaum relativ zu allen Konstituenten des Baums
  - **Labelled recall:** Anzahl korrekter Konstituenten im Ableitungsbaum relativ zu Referenzkorpus

25

# Evaluation

- **Labelled Precision:**
$$\frac{\# \text{ korrekte Konstituenten im Ableitungsbaum für Eingabe}}{\# \text{ Konstituenten im Ableitungsbaum für Eingabe}}$$
- **Labelled Recall:**
$$\frac{\# \text{ korrekte Konstituenten im Ableitungsbaum für Eingabe}}{\# \text{ Konstituenten im Referenzkorpus für Eingabe}}$$
- **Korrekt:** die Konstituente hat die richtige Kategorie und überspannt den richtigen Teil der Eingabe.
  - Satzzeichen etc. werden üblicherweise ignoriert

26

# Chomsky-Normalform

- Wenn bereits eine Grammatik vorliegt:
  - ersetze Regel  $A \rightarrow A_1 A_2 A_3 \dots A_k [p]$  durch
  - $\langle A_1, \dots, A_{k-1} \rangle \rightarrow A_1 \dots A_{k-1} [1.0]$
  - $A \rightarrow \langle A_1, \dots, A_{k-1} \rangle A_k [p]$
- Wenn die Grammatik aus einer Baumbank extrahiert wird, kann man alternativ zuerst die Bäume der Baumbank binarisieren.

27

# Literatur

- Jurafsky & Martin (2009) Speech and Language Processing Kapitel 14.
- Manning & Schütze (1999). Foundations of Statistical Natural Language Processing. Kapitel 11 & 12.
- Eugene Charniak (1993). Statistical Language Learning. Kapitel 5.

28