

Probabilistische kontextfreie Grammatiken und Parsing

Sebastian Pado

1

Robustes Parsing

- ▶ Ziel: Syntaktische Analyse von freiem Text
- ▶ Anwendungen:
 - ▶ Freier Dialog
 - ▶ Genaue Extraktion von Information aus großen Textmengen (Internet)
- ▶ Nötig: Grammatik

▶ 2

Erzeugung einer großen Grammatik

- ▶ Von Hand schreiben?



G

Peter

redet

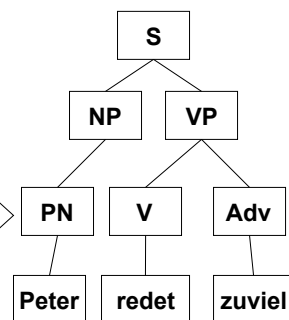
zuviel

▶ 3

Erzeugung einer großen Grammatik

- ▶ Von Hand schreiben?

G



▶ 4

Herausforderungen für die Grammatik

“and i think she or he is located just a second uh at the end of the highway five ninety so what i'm doing is to send a helicopter at the end of two sixty at the very beginning and after that oh i also have do you know where i have buses”

“Similar thing happened here a while back, the kids at one of the international schools had class photos taken.”

- ▶ Unterschied zwischen Grammatik und Gebrauch
 - ▶ Die meisten gesprochenen Sätze sind „ungrammatisch“
 - ▶ Viele informelle geschriebene Sätze sind „ungrammatisch“
- ▶ Wir möchten trotzdem eine Analyse

▶ 5

Herausforderungen für die Grammatik

“We would need a lot of gear.”

REC: we would need a lot of your.

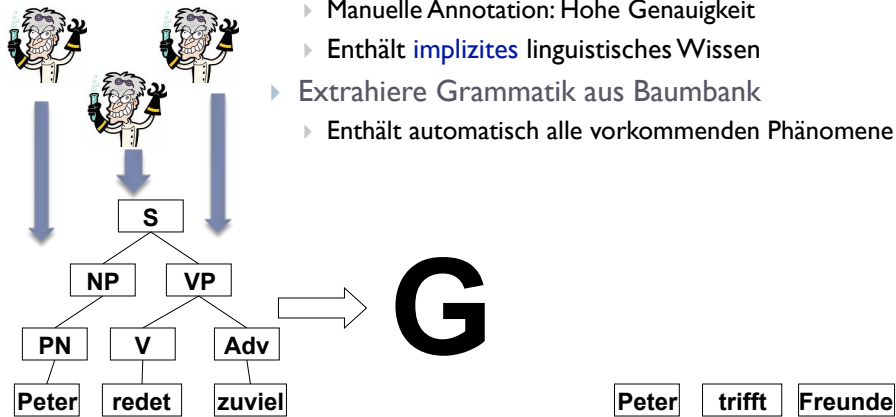
- ▶ Fehler durch falsche Spracherkennung
 - ▶ Sätze tatsächlich ungrammatisch
 - ▶ Aber wir möchten so viel Struktur wie möglich „retten“

▶ 6

Erzeugung einer großen Grammatik

▶ Alternative: Erzeuge **Baumbank**

- ▶ Korpus von Sätzen mit syntaktischer Analyse
 - ▶ Manuelle Annotation: Hohe Genauigkeit
 - ▶ Enthält **implizites** linguistisches Wissen
- ▶ Extrahiere Grammatik aus Baumbank
 - ▶ Enthält automatisch alle vorkommenden Phänomene

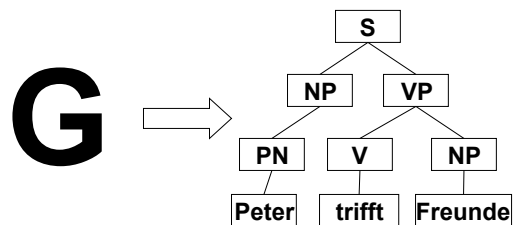


▶ 7

Erzeugung einer großen Grammatik

▶ Alternative: Erzeuge **Baumbank**

- ▶ Korpus von Sätzen mit syntaktischer Analyse
 - ▶ Manuelle Annotation: Hohe Genauigkeit
 - ▶ Enthält **implizites** linguistisches Wissen
- ▶ Extrahiere Grammatik aus Baumbank
 - ▶ Enthält automatisch alle vorkommenden Phänomene



▶

Manuell geschriebene vs. automatisch gelernte Grammatiken

- ▶ **Vorteile automatisch gelernter Grammatiken**
 - ▶ Deckt alle Phänomene im Korpus ab (auch “grenzwertig grammatische”)
 - ▶ Kann Grammatikalität von Sätzen quantifizieren
 - ▶ (siehe Rest dieser Vorlesung)
 - ▶ Linguistisches Wissen bleibt implizit
 - ▶ Verbesserungen in der statistischen Modellierung von Korpora führen automatisch zu besseren Parsern
 - ▶ Arbeit kann von Studenten ausgeführt werden
 - ▶ Aber: verlangt gute Annotationsrichtlinien!
- ▶ **Nachteile**
 - ▶ Redundanz: Häufige Phänomene werden vielfach annotiert
 - ▶ Generalisierung
 - ▶ Seltene Phänomene, die nicht im Korpus vorkommen, fehlen in der Grammatik
 - ▶ Unklar, ob Gemeinsamkeit zwischen ähnlichen Phänomenen erfaßt werden
 - ▶ Beispiele folgen!

▶ 9

Extraktion einer Baumbank-Grammatik

Baum = Menge von Regeln

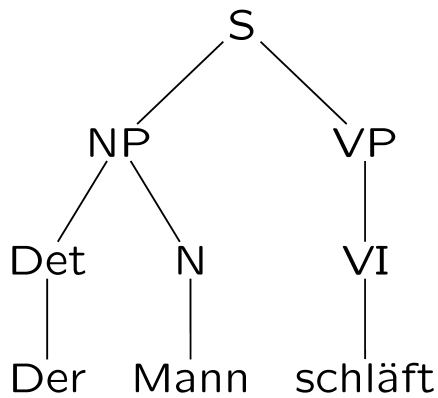
- ▶ **Naiver** Extraktionsalgorithmus einer kontextfreien Baumbank-Grammatik:
 - ▶ Initialisierung: „Leere Grammatik“ ohne Regeln
 - ▶ Eine (kontextfreie) Baumbank nehmen
 - ▶ Alle Syntaxbäume in Einzelregeln aufteilen und zur Grammatik hinzufügen

▶ 10

Naive Baumbank-Grammatiken

Parsebaum

Abgeleitete Regeln



S → NPVP
NP → Det N
VP → VI

Det → Der
N → Mann
VI → schläft

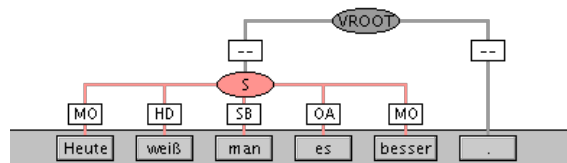
▶ 11

Probleme von Baumbank-Grammatiken

▶ 12

Form der Baumbank-Regeln

Baumbank-Bäume sind typischerweise sehr flach



- ▶ **Tiefere Bäume**
 - ▶ Mehr Regeln im Baum: mehr Annotationsentscheidungen
- ▶ **Flachere Bäume**
 - ▶ weniger Regeln im Baum: weniger Annotationsentscheidungen
 - ▶ einfacher zu annotieren

▶ 13

Baumbank-Grammatiken

- Bei flachen Bäumen gibt es dafür **mehr Regeln in der Grammatik!**
 - Beispiel: Regeln für NPs mit und ohne AdjP

Traditionelle Grammatik
(mit AdjP)

NP → AdjP N
NP → Det AdjP N

AdjP → ε
AdjP → Adj AdjP

Baumbank-Grammatik
(ohne AdjP)

NP → N
NP → Adj N
NP → Adj Adj N

NP → Det N
NP → Det Adj N
NP → Det Adj Adj N

▶ 14

Dt. Baumbank NEGRA

...
 NP -> ADJA \$*LRB* \$,ADJA NN
 NP -> ADJA \$*LRB* NN
 NP -> ADJA \$*LRB* NN \$*LRB* PP
 NP -> ADJA \$*LRB* NN PP
 NP -> ADJA \$,ADJA NN
 NP -> ADJA \$,ADJA NN PP
 NP -> ADJA \$,AP NN
 NP -> ADJA ADJA ADJA NN
 NP -> ADJA ADJA CARD
 NP -> ADJA ADJA CNP
 NP -> ADJA ADJA NN \$*LRB* NE
 NP -> ADJA ADJA NN \$*LRB* NP
 NP -> ADJA ADJA NN NP
 NP -> ADJA ADJA NN
 ...

\$.: Komma
 ADJA: Attributives Adjektiv
 CARD: Kardinalzahl
 CNP: Koordinierte NP
 \$*LRB*: Klammer
 NE: Eigennamen
 NN: Normales Nomen
 NP: Nominalphrase
 PP: Präpositionalphrase

Grammatik wird
 sehr groß:
 19.000 Regeln für 18.000
 Sätze NEGRA-Korpus

▶ 15

Flache Regeln generalisieren nicht gut

▶ AdjP-Beispiel

Grammatik 1:

NP → AdjP N
 NP → Det AdjP N

AdjP → ε
 AdjP → Adj AdjP

Grammatik 2:

NP → N
 NP → Adj N
 NP → Adj Adj N

NP → Det N
 NP → Det Adj N
 NP → Det Adj Adj N

▶ Syntaktische Ähnlichkeit von „alte grüne Häuser“ und „grüne Häuser“?

▶ 16

Aber sie generalisieren **zu** gut

- ▶ Große Grammatiken enthalten sehr viele seltene Regeln
 - ▶ „Zipfsche Verteilung“
- ▶ Wieso ist das problematisch?
 - ▶ Es gibt Regeln für fast jedes Muster, wie selten es auch ist
 - ▶ „Übgenerierung“
- ▶ Beispiel:
 - ▶ Kleine Gruppe von Verben nimmt mehrere PP-Objekte
 - ▶ Bewegungsverben
 - ▶ Die Aktien **steigen** [von 50 Euro] [um 10 Euro] [auf 60 Euro]
 - ▶ Grammatik enthält Regel VP → V PP PP PP

▶ 17

Mehrdeutigkeit

- ▶ Auch auf Sätze, die gar nicht so analysiert werden können, wird diese Regeln angewendet:

Peter **telefoniert** [mit dem Handy] [von Siemens] [in der Hand]
 Ich **stehe** [mit dem Schal] [auf der Brücke] [von Köln]
 VP → V PP PP PP ?????

- ▶ Resultat: Eigentlich eindeutige Sätze werden mehrdeutig
 - ▶ Sätze mit 20-30 Worten erhalten z.T. > 1 Mio. Analysen
 - ▶ [Verarbeitungsproblem](#)
 - ▶ [Interpretationsproblem](#)

▶ 18

Baumbanken: Linguistik vs. NLP

- ▶ Baumbank-Grammatiken enthalten viele Regeln und generalisieren daher schlecht
- ▶ Linguistische Evaluation von Grammatiken
 - ▶ Kleiner ist besser: Occams Razor
 - ▶ Kleine Grammatik generalisiert besser auf neue Sätze
- ▶ Fazit: Baumbank-Grammatiken sind keine linguistisch befriedigenden Grammatiken
 - ▶ Aber sie sind „robust“ und damit anwendungsgeeignet

▶ 19

Grammatikalität

- ▶ **Naive** Baumbank-Grammatiken haben kein Konzept von **gradueller** Grammatikalität
 - ▶ Keine Möglichkeit, von plausibleren / unplausibleren Analyse zu sprechen
 - ▶ Besonders schlimm wegen hoher Mehrdeutigkeit
 - ▶ Was nutzen 1 Mio. Analysen für einen Satz?

▶ 20

Verbesserung: Probabilistische kontextfreie Grammatiken (PCFGs)

▶ 21

Probabilistische Grammatiken (PCFGs)

- ▶ Jede Regel erhält eine Wahrscheinlichkeit
 - ▶ $S \rightarrow NPVP$ (0.6)
 - ▶ $S \rightarrow NPVP PP$ (0.4)
 - ▶ „60% aller Konstituenten des Typs S (Sätze) haben die Form NPVP, 40% die Form NPVP PP“

- ▶ Wahrscheinlichkeiten für Parsebäume
 - ▶ **Berechnung aus Regelwahrscheinlichkeiten**

- ▶ Vorteile
 - ▶ Grammatikseite: Graduelle Grammatikalität
 - ▶ Beste Analyse = wahrscheinlichster Parsebaum
 - ▶ Verarbeitungsseite: Einschränkung des Suchraumes
 - ▶ Verfolge vor allem „wahrscheinliche“ Parsebäume

▶ 22

Parsen mit PCFGs

1. **Lerne Wahrscheinlichkeiten für Regeln**
2. Berechne Wahrscheinlichkeiten für Bäume aus Regelwahrscheinlichkeiten
3. (Finde effizientes Verfahren, um wahrscheinlichste Bäume zu konstruieren)

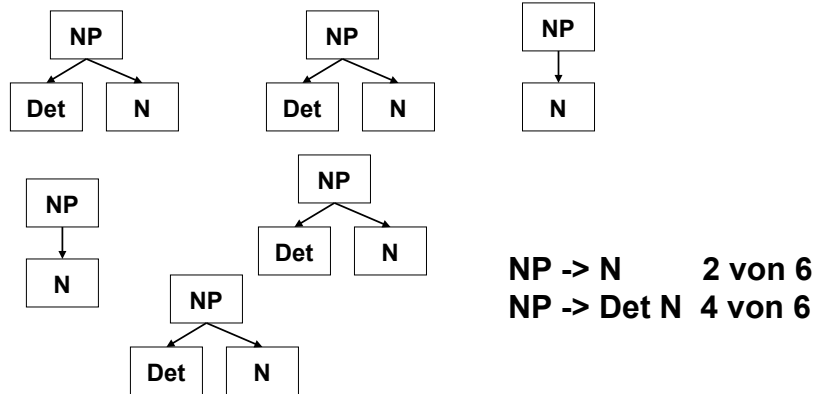
▶ 23

Regelwahrscheinlichkeiten lernen

- ▶ **Nicht naive Baumbank-Grammatik:**
 1. Baumbank nehmen
 2. Leere Grammatik initialisieren
 3. Alle Parsebäume in „lokale Bäume“ aufteilen
 4. Jeden neuen lokalen Baum als Regel hinzufügen
 - ▶ Zähle (relative) Häufigkeit von Regeln
- ▶ **Terminologie:**
 - ▶ Linke Seite (LS): Nonterminal
 - ▶ Rechte Seite (RS): Expansion

▶ 24

Beispiel



▶ 25

Regelwahrscheinlichkeiten

- ▶ Gleiche Idee wie bei statistischer Klassifikation (letzte Vorlesung)
 - ▶ Dort: Für gegebenes Ereignis, sammle Frequenzen der Klassen
 - ▶ Hier: Für gegebenes Nonterminal, sammle Frequenzen der Expansionen:

Von allen Auftreten des Nonterminals,
wie oft kommt eine Expansion vor?

- ▶ Das ist nichts anderes als eine bedingte Wahrscheinlichkeit:

$$P(\text{Expansion} \mid \text{Nonterminal}) = f(\text{Expansion, Nonterminal}) / f(\text{Nonterminal})$$

- ▶ Beispiel von letzter Seite:
 - ▶ $P(N \mid NP) = 2/6$
 - ▶ $P(\text{Det N} \mid NP) = 4/6$

▶ 26

„Echte“ Regelfrequenzen in NEGRA

...	
1 NP ADJA \$*LRB* \$, ADJA NN	
12 NP ADJA \$*LRB* NN	
1 NP ADJA \$*LRB* NN \$*LRB* PP	$P(\text{ADJA LRB NN} \mid \text{NP}) = 12/89$
2 NP ADJA \$*LRB* NN PP	
8 NP ADJA \$, ADJA NN	$P(\text{ADJA ADJA NN} \mid \text{NP}) = 51/89$
1 NP ADJA \$, ADJA NN PP	
3 NP ADJA \$, AP NN	$P(\text{ADJA ADJA CARD} \mid \text{NP}) = 1/89$
1 NP ADJA ADJA ADJA NN	
1 NP ADJA ADJA CARD	
2 NP ADJA ADJA CNP	
1 NP ADJA ADJA NN \$*LRB* NE	
1 NP ADJA ADJA NN \$*LRB* NP	
4 NP ADJA ADJA NN NP	
51 NP ADJA ADJA NN	
...	

▶ 27

Parsing mit PCFGs

1. Lerne Regeln mit Wahrscheinlichkeiten
2. **Berechne Wahrscheinlichkeiten für Bäume aus Regelwahrscheinlichkeiten**
3. (Finde effizientes Verfahren, um wahrscheinlichste Bäume zu konstruieren)

▶ 28

Baumwahrscheinlichkeiten berechnen

- ▶ Baum ist Menge an Regeln
 - ▶ Z.B. (R_1, R_2, R_3)
- ▶ Bekannt: Wahrscheinlichkeit der Regeln
 - ▶ $P(R_1), P(R_2), P(R_3)$
- ▶ Gesucht: Wahrscheinlichkeit des Baumes
 - ▶ Idee: **Produkt** der Regelwahrscheinlichkeiten
 $P(R_1, R_2, R_3) = \prod_i P(R_i) = P(R_1) * P(R_2) * P(R_3)$

▶ 29

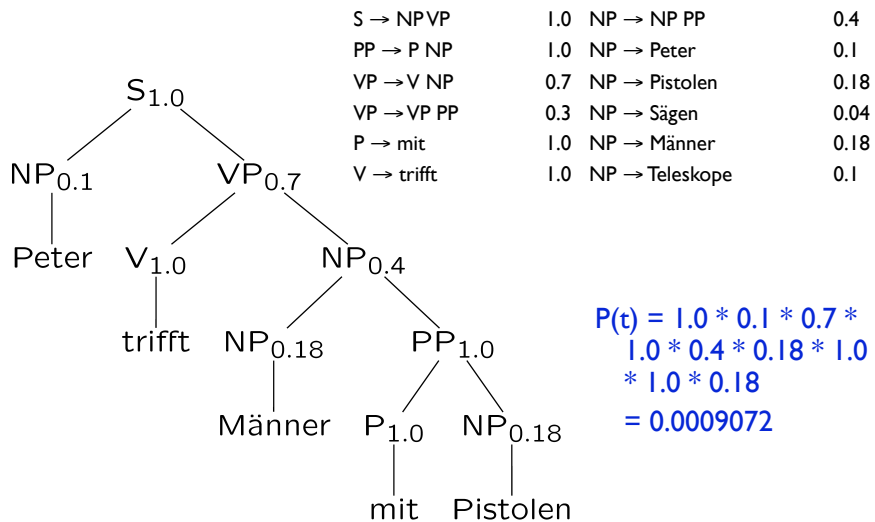
Beispiel

S → NPVP	1.0	NP → NP PP	0.4
PP → P NP	1.0	NP → Peter	0.1
VP → V NP	0.7	NP → Pistolen	0.18
VP → VP PP	0.3	NP → Sägen	0.04
P → mit	1.0	NP → Männer	0.18
V → trifft	1.0	NP → Teleskope	0.1

Satz: „Peter trifft Männer mit Pistolen.“

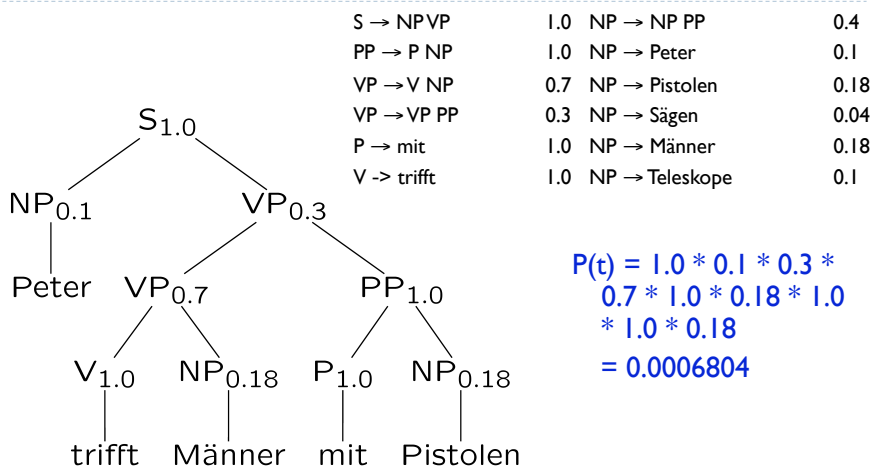
▶ 30

Analyse 1



▶ 31

Analyse 2



▶ 32

Zwischenstand

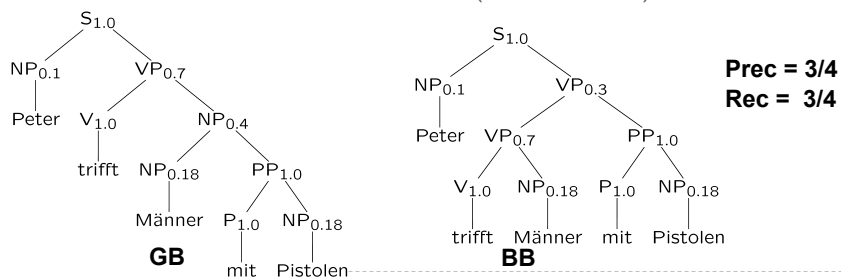
- ▶ PCFG (probab. kontextfreie Grammatik)
 - ▶ Regeln der Form $X \rightarrow A B \dots$
 - ▶ $P(\text{Regel}) = P(\text{RS} \mid \text{LS}) = P(\text{Expansion} \mid \text{Nonterminal})$
 - ▶ $P(\text{Baum}) = \text{Produkt aller } P(\text{Regeln})$
- ▶ Vorteile
 - ▶ PCFG kann aus Baumbank abgelesen werden
 - ▶ $P(\text{Baum})$ kann als Maß für Grammatikalität interpretiert werden
 - ▶ Gute Analysen: hohe Wahrscheinlichkeit
 - ▶ Schlechte Analysen: niedrige Wahrscheinlichkeit

Aber stimmt das wirklich?

▶ 33

Evaluation von Baumbank-PCFGs

- ▶ Vergleich des wahrscheinlichsten Baumes der gelernten Grammatik (GB) mit dem Baumbank-Baum (BB)
 - ▶ Precision: Anteil der GB-Konstituenten, die auch im BB vorkommen
 - ▶ Recall: Anteil der Konstituenten im BB, die auch im GB vorkommen
 - ▶ Konstituenten zählen dann als gleich, wenn sie den gleichen Typ haben und die gleichen Terminale umspannen
 - ▶ Konstituenten, aus nur einem Wort („Präterminale“) zählen nicht



▶ 34

Evaluation von Baumbank-PCFGs

Englisch (Penn Treebank):

Satzlänge	Ø Länge	Precision	Recall
2-16 Test	11.4	85.0	87.7
2-25 Test	16.3	82.0	84.0
2-40 Test	21.9	78.8	80.4

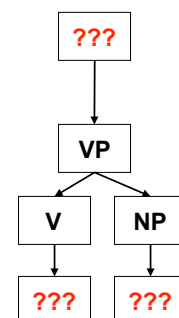
- ▶ Wahrscheinlichkeit von Baumbank-PCFGs modelliert Grammatikalität nicht optimal
 - ▶ 1 von 5 Regeln des BB (E) fehlt
 - ▶ 1 von 5 Regeln des GB (E) ist dazuerfunden

Woher kommen die Fehler?

▶ 35

Unabhängigkeit

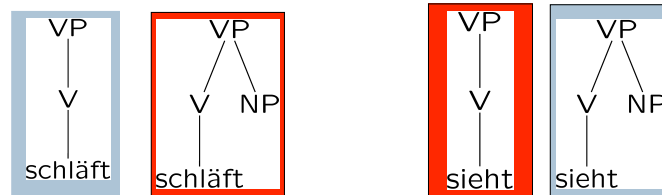
- ▶ Baumwahrscheinlichkeit: $P(R_1, R_2, R_3) = P(R_1) * P(R_2) * P(R_3)$
- ▶ Was bedeutet es, die $P(R)$ zu multiplizieren?
 - ▶ Alle Regeln ist **unabhängig** voneinander
 - ▶ Ob R_1 vorkommt, hat keinen Einfluss auf R_2
- ▶ Beispiel: $P(V \ NP \ | \ VP) =$ wie oft expandiert VP zu V NP?
 - ▶ Verb, Objekt, ... spielen keine Rolle
 - ▶ Linguistisch unplausibel



▶ 36

Beispiel 1: Einfluss der Kinder bei VPs

- ▶ Transitive Vs: $P(V \mathbf{NP} | VP)$ soll sehr hoch sein, $P(V | VP)$ klein
- ▶ Intransitive Vs: $P(V | VP)$ soll sehr hoch sein, $P(V \mathbf{NP} | VP)$ klein



- ▶ Aber: Standardmodell kann zwischen den beiden Fällen nicht unterscheiden: es sieht die Verben nicht!
- ▶ Abhilfe: Modell mit neuen **Features** erweitern

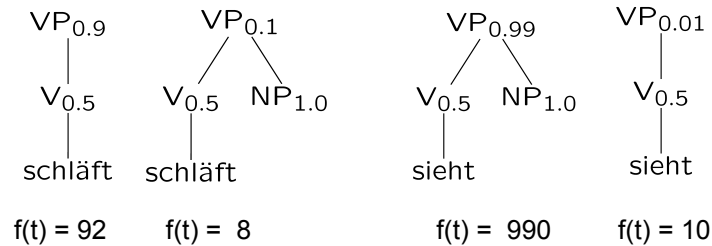
▶ 37

Lexikalisierung

- ▶ **Lexikalischen Kopf** als Feature verwenden
 - ▶ Form des Kopfes hängt von Phrasentyp ab
 - ▶ VPs, Ss sehen ihr (Haupt-) verb
 - ▶ NPs sehen ihr Nomen
 - ▶ APs sehen ihr Adjektiv
 - ▶ Neue Wahrscheinlichkeiten haben die Form: $P(V | VP, schlafen)$
- ▶ Jetzt ist möglich, daß $P(V | VP, schlafen) > P(V \mathbf{NP} | VP, schlafen)$
 - ▶ „Die Wahrscheinlichkeit, dass eine VP, deren Kopf „schlafen“ ist, nur aus einem V besteht (intransitiv ist), ist größer als die Wahrscheinlichkeit, dass sie aus einem V und einer NP besteht“

▶ 38

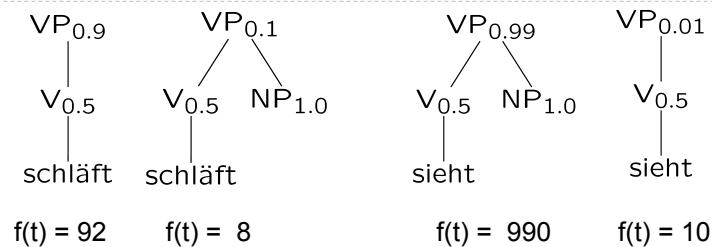
Lexikalisierung (Beispiel)



- ▶ Beim Auszählen muß jetzt das Kopfwort berücksichtigt werden
- ▶ $P(V | VP, schlafen) = f(V, VP, schlafen) / f(VP, schlafen)$
 - ▶ „Von allen VPs mit dem Kopf *schlafen*, wieviele bestehen nur aus einem Verb?“
- ▶ Hier: $P(V | VP, schlafen)$ $P(V NP | VP, schlafen)$
 $P(V | VP, sehen)$ $P(V NP | VP, sehen)$

▶ 39

Lexikalisierung (Beispiel)



- ▶ Beim Auszählen muß jetzt das Kopfwort berücksichtigt werden
- ▶ $P(V | VP, schlafen) = f(V, VP, schlafen) / f(VP, schlafen)$
 - ▶ „Von allen VPs mit dem Kopf *schlafen*, wieviele bestehen nur aus einem Verb?“
- ▶ Hier: $P(V | VP, schlafen) = 92/100$ $P(V NP | VP, schlafen) = 8/100$
 $P(V | VP, sehen) = 10/1000$ $P(V NP | VP, sehen) = 990/1000$

▶ 40

Korrektheit und Grammatikgröße

- ▶ Lexikalisierung modelliert linguistischen Zusammenhang
 - ▶ Erhöht im Prinzip **Korrektheit**
 - ▶ Sinnvolle Parsebäume erhalten höhere Wahrscheinlichkeit

▶ 41

Korrektheit und Grammatikgröße

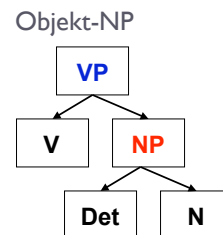
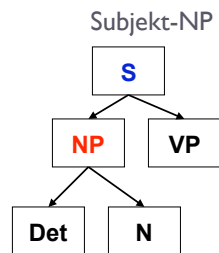
- ▶ Aber: Modell wird sehr groß
 - ▶ Je eine Regel für Kombination (Regel, Kopfwort)
 - ▶ An Stelle von $P(V | VP)$ haben wir
 - $P(V | VP, abfahren)$,
 - $P(V | VP, anrufen)$,
 - $P(V | VP, angeln)$,
 - $P(V | VP, ausrufen)$
 - Etc.
 - ▶ Resultat: Sehr großer Ereignisraum
 - ▶ **Sparse Data-Problem**
 - Gesehene Frequenzen verteilen sich auf mehr Ereignisse (bis zu 1000mal mehr Ereignisse)
 - Was tun für ungesehene Verben?

Konkrete Modelle können auch schlechter werden

▶ 42

Problem 2: Einfluß der Eltern bei NPs

- ▶ Regeln wissen nicht, was „über ihnen“ passiert
 - ▶ Sogenannte **Großvaterkategorie**



- ▶ NP-Regel NP -> Det N ist identisch
 - ▶ $P(\text{Det N} \mid \text{NP})$ kann nicht zwischen Subj und Obj unterscheiden

43

Eltern von NPs

- ▶ Wahrscheinlichkeiten für NP-Regeln unterscheiden sich aber in Wirklichkeit deutlich zwischen Subj und Obj:

Regel	% als Subjekt	% als Objekt
NP → PRP	13.7%	2.1%
NP → NP PP	5.6%	14.1%
NP → NP S	0.5%	2.6%
NP → DET N	5.6%	4.6%

- ▶ **Lösung:** Großvaterkategorie als weiteres Feature in die Wahrscheinlichkeitsberechnung aufnehmen
 - ▶ $P(\text{Det N} \mid \text{NP}, \mathbf{S})$ = „NP -> Det N als Subjekt“
 - ▶ $P(\text{Det N} \mid \text{NP}, \mathbf{VP})$ = „NP -> Det N als Objekt“

44

Beispiel für Regeln mit Geschichte

Regel	% als Subjekt	% als Objekt
NP → PRP	13.7%	2.1%
NP → NP PP	5.6%	14.1%
NP → NP S	0.5%	2.6%
NP → DET N	5.6%	4.6%

$$\begin{aligned}
 P(\text{PRP} \mid \text{NP}, \mathbf{S}) &= 0.137 & P(\text{NP S} \mid \text{NP}, \mathbf{S}) &= 0.005 \\
 P(\text{PRP} \mid \text{NP}, \mathbf{VP}) &= 0.021 & P(\text{NP S} \mid \text{NP}, \mathbf{VP}) &= 0.026 \\
 \\
 P(\text{NP PP} \mid \text{NP}, \mathbf{S}) &= 0.056 & P(\text{DET N} \mid \text{NP}, \mathbf{S}) &= 0.056 \\
 P(\text{NP PP} \mid \text{NP}, \mathbf{VP}) &= 0.141 & P(\text{DET N} \mid \text{NP}, \mathbf{VP}) &= 0.046
 \end{aligned}$$

45

Regeln mit Geschichte

- ▶ Geschichte modelliert wichtigen linguistischen Zusammenhang
 - ▶ Hilft auch bei anderen Unterscheidungen:
 - ▶ Akkusativ- und Dativ-Objekt (Pronominalisierung)
 - ▶ Hauptsatz und Nebensatz (D:Verbzweitsatz vs. Verbletztsatz)
 - ▶ Höhere Wahrscheinlichkeiten für korrekte Konstruktionen
- ▶ Wieder mehr Regeln
 - ▶ $P(\text{RS} \mid \text{LS}, \text{GV})$: Kombination aus Regel und Großvaterkategorie
 - ▶ Anzahl der Großvaterkategorien beschränkt
 - ▶ Nicht so schlimm wie bei Lexikalisierung
 - ▶ Beispiel NP: Regelzahl verdoppelt sich (Subjekt, Objekt)
 - ▶ Sparse data-Problem existiert, aber weniger dramatisch

▶ 46

Evaluation: State of the art

- ▶ **Heutige Parser verwenden Geschichte und Lexikalisierung**
 - ▶ Versuche zuerst, Wahrscheinlichkeit für Regeln mit Geschichte und Lexikalisierung zu finden
 - ▶ Wenn lexikalischer Kopf unbekannt, verzichte auf Lexikalisierung
 - ▶ Wenn Grossvaterkategorie unbekannt, verzichte auf Geschichte
- ▶ **Englisch (Penn Treebank): Recall 90%, Precision 90%**
 - ▶ (Naives Modell Recall 79%, Precision 80%)
- ▶ **Deutsch (NEGRA): Recall 81%, Precision 77%**
 - ▶ (Naives Modell Recall 73%, Precision 70%)

▶ 47

Bewertung

- ▶ **Was heißen diese Zahlen konkret?**
 - ▶ Typische Satzlänge: 20 Wörter = Baumgröße von 20 Regeln
 - ▶ Chance für vollständig richtigen Baum:
 - ▶ Bei 90% Akkuratheit (Englisch) grob etwa 12%
 - ▶ Bei 80% Akkuratheit (Deutsch) grob etwa 1%
- ▶ **Warum ist Deutsch schwieriger?**
 - ▶ Flexible Wortstellung
 - ▶ Flachere Annotation der Bankbanken
 - ▶ Kleinere Baumbank

▶ 48

Parsing mit PCFGs

1. Lerne Regeln mit Wahrscheinlichkeiten
2. Berechne Wahrscheinlichkeiten für Bäume aus Regelwahrscheinlichkeiten
3. **Finde effizientes Verfahren, um wahrscheinlichste Bäume zu konstruieren**

▶ 49

Herkömmliches Parsing

- ▶ Exhaustives (erschöpfendes) Parsing (wie in den vergangenen Vorlesungen:
 - ▶ Baue komplette Chart auf
 - ▶ Top-Down: Fange an mit S, und wende alle anwendbaren Regeln an
 - ▶ Bottom-Up: Fange an mit Worten, kombiniere mit allen anwendbaren Regeln
- ▶ Problem bei Baumbank-Grammatik: Anzahl der Regeln
 - ▶ Oft mehrere 10.000 Regeln
 - ▶ Exhaustives Parsing nicht beherrschbar für Sätze mit mehr als 10 Wörtern

▶ 50

Parsing mit Baumbank-Grammatiken

- ▶ **Hoffnung: ungleiche Wahrscheinlichkeitsverteilung**
 - ▶ Es gibt wenig „vernünftige“ Parsebäume, viel Müll
 - ▶ „Vernünftige“ Parsebäume sind viel wahrscheinlicher als (fast) alle anderen
- ▶ **Aufgabe: Konstruiere die kleine Anzahl sehr wahrscheinlicher Parsebäume**

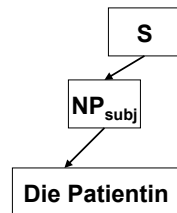
▶ 51

Determinisches Parsing

- ▶ **Extremansatz: verfolge eine einzige Hypothese**
 - ▶ Beginne irgendwo
 - ▶ Finde den **wahrscheinlichsten Teil-Parsebaum**
 - ▶ Finde **wahrscheinlichste** Erweiterung
 - ▶ Wiederhole
- ▶ **Sehr effizient (lineare Laufzeit)**
- ▶ **Erhält man so immer den wahrscheinlichsten vollständigen Parsebaum? Nein!**

▶ 52

Bottom-Up deterministisches Parsing



Wahrscheinlichste Analyse eines Eigennamens am
Satzanfang: Als Subjekt des Satzes

▶ 53

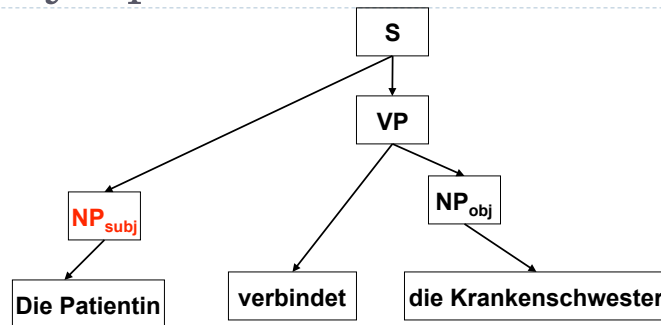
Analyseproblem



Auch wenn der Rest des Satzes nicht gut dazu passt,
muß deterministisches Parsing an der Original-Analyse
festhalten

▶ 54

Analyseproblem



Auch wenn der Rest des Satzes nicht gut dazu passt, muß deterministisches Parsing an der Original-Analyse festhalten: Unwahrscheinliche Analyse!

▶ 55

Beam Search

- ▶ Verfolge die n besten Analysen
- ▶ Datenstruktur: Liste von halbfertigen Parsebäumen (Hypothesen) mit Wahrscheinlichkeiten
 - ▶ Fange am Anfang des Satzes an
 - ▶ Lies das nächste Wort
 - ▶ Finde alle Regeln, die die aktuellen Parsebäume um dieses Wort erweitern können
 - ▶ Rechne die Wahrscheinlichkeiten für die erweiterten Hypothesen aus
 - ▶ Behalte die besten n Hypothesen
 - ▶ Gehe zum nächsten Wort

▶ 56

Beam Search

- ▶ $n = 1$: verfolge nur die besten Analysen
- ▶ $n = \infty$: exhaustives Parsing

- ▶ Gefahr: für kleines n kann der global beste Parsebaum „aus dem Strahl fallen“
 - ▶ Wie groß muß n sein?

▶ 57

Beam Search beim Parsen

n	Zeit (s)	Precision	Recall
20	2.07	87.9	87.1
15	1.58	87.7	86.9
10	1.07	87.7	86.9
7	0.76	87.4	86.6
5	0.56	87.3	86.8
2	0.35	85.1	86.1
1	0.14	82.4	83.4

▶ 58

Beam Search in der Praxis

- ▶ Optimales n muß empirisch ermittelt werden
- ▶ Funktioniert ziemlich gut
 - ▶ Bester Parsebaum (fast) immer in der vorne dabei
 - ▶ Bestätigt Hypothese über ungleiche Verteilung:
Grammatische Sätze haben im Allgemeinen hohe Wahrscheinlichkeiten

▶ 59

Zusammenfassung

- ▶ Kontextfreie Grammatiken
 - ▶ Können aus Baumbanken gelernt werden
 - ▶ Viele Regeln
 - ▶ Hohe Ambiguität
- ▶ Probabilistische kontextfreie Grammatiken
 - ▶ Möglichkeit, über „besten Baum“ zu sprechen
 - ▶ Regelwahrscheinlichkeiten ablesen
 - ▶ Einfache Regeln: kein gute Modellierung von Grammatikalität
 - ▶ Erweiterung mit Lexikalisierung und Geschichte
 - ▶ Suche nach besten Analysen mit Beam Search

▶ 60