Intro NLP Tools

Caroline Sporleder & Ines Rehbein

WS 09/10

Probabilistic Context-Free Grammar (PCFG) - Definition

- A PCFG is a CFG, where each rule is assigned a probability: $< N, \sum, R, S >$
  - N is the set of non-terminal symbols
  - $\sum$ is the set of terminal symbols
  - R is the set of rules $A \rightarrow \beta\ [p]$,
    where $A \epsilon N$ and $\beta \epsilon (N \cup \sum)*$,
    and p **is a number between 0 and 1**
  - S is the start symbol

- The sum of all probabilities for all RHS of a particular LHS adds up to 1

PCFG - Disambiguation

- A PCFG assigns each parse tree T (each possible derivation of a rule) a probability
- The parser choses the parse tree with the highest probability

- PCFGs can disambiguate between a number of possible derivations
- PCFGs allow to rank possible derivations according to their probability
- But: where do these probabilities come from?

Extracting a PCFG from a Treebank

- A treebank as a set of rules   e.g.   S → NP VP
- A PCFG assigns to each context-free rule LHS → RHS a conditional probability: $P_r(RHS|LHS)$
- Read all the rules off the treebank and add probabilities to the rules

$$P_r(RHS|LHS) = \frac{Freq(LHS \rightarrow RHS)}{Freq(LHS)}$$

(Maximum Likelihood Estimation)

Maximum Likelihood Estimation (MLE)

- Compute the probability of class x, based on its *relative frequency* in the training data: $P(x) = \frac{Freq(x)}{N}$
- $Freq(x)$ = Frequency of x in the training data
- N = Number of training instances
- Problems with MLE:
  - under-estimates the probability of unseen events
  - over-estimates the probability of rare events

Problems with PCFGs

- **(Independence assumption)**: the expansion of each node in the tree is dependent on the category of the node only
  (Markov assumption: the probability of an event is dependent on the previous *n* events only)

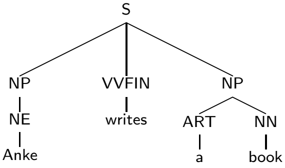
Disadvantage of PCFGs: not sensitive for lexical information or structural context, on the other hand: PCFGs can be computed in an efficient way

Lexicalisation

- Head of a phrase contains important information about structure and meaning (subcategorisiation frame, PP attachment, ...)
- Include information in the parsing model
- Magerman (1995), Charniak (1997), Collins (1997)

Lexicalisation (Charniak, 1997)

- **Step 1**: Mark the head in each rule



Rules:

S	→	NP VVFIN NP
NP	→	NE
NP	→	ART NN

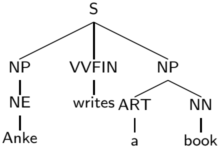
head marking ⇒

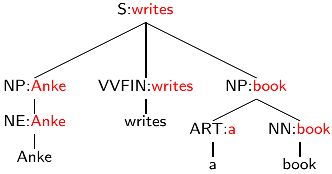
head-marked rules:

S	→	NP VVFIN' NP
NP	→	NE'
NP	→	ART NN'

Lexicalisation (Charniak, 1997)

- **Step 2**: Transform the original tree
  - Start with the leaf nodes, mark each mother node with the lexical head of the node (head percolation)
  - Continue until you reach the root node





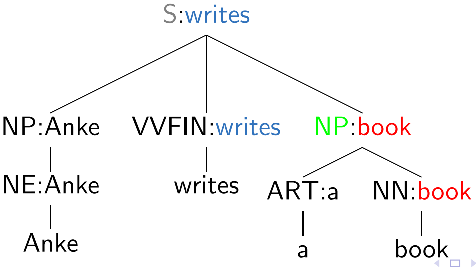
Charniaks Probability Model

- Probability of the whole tree = product of all rules in the tree
- e.g. probability fo NP(a book):
  - determine the probability of the head of the NP
  - determine the probability of the form of the NP, given the head
  - determine (recursively) the probability for all sub-constituents

Charniaks Probability Model - Dependencies

- h is the head of a constituent
- c is the category of the constituent
- pc is the category of the mother node
- ph is the head of the mother node

- Example: **Head probability** *d* (dependency)
  only depends on ph, c and pc $\Rightarrow$ $p(h|ph, c, pc)$
  z.B. $p(book|writes, NP, S)$



Lexicalised Parsing - Summary

- Syntactic structure of a constituent is determined according to its lexical head
- weakens the independence assumption of PCFGs
- makes PCFGs more sensible to differences between subcategorisation frames (selectional preferences)
- Sparse Data
- Other approaches to improving PCFGs:
  - Treebank Transformation (Parent-Encoding, Johnson 1999)
  - Treebank Refinement /Split & Merge

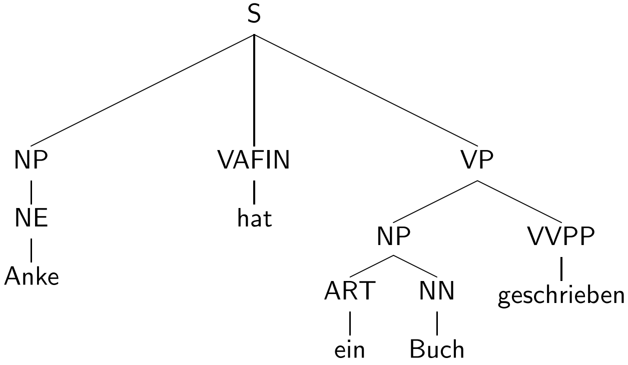
Lexicalised Parsing - Summary

- Syntactic structure of a constituent is determined according to its lexical head
- weakens the independence assumption of PCFGs
- makes PCFGs more sensible to differences between subcategorisation frames (selectional preferences)
- Sparse Data

- Other approaches to improving PCFGs:
  - Treebank Transformation (Parent-Encoding, Johnson 1999)
  - Treebank Refinement /Split & Merge

Treebank Transformation (Johnson, 1999)

- Add local context to the rules (Parent transformation)

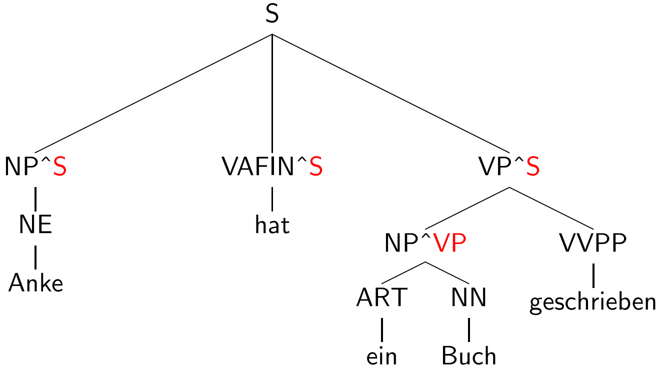


- Splits syntactic categories according to more fine-grained criteria:

NP^S	$\longrightarrow$	subject
NP^VP	$\longrightarrow$	object

Treebank Transformation (Johnson, 1999)

- Add local context to the rules (Parent transformation)



- Splits syntactic categories according to more fine-grained criteria:

NP^S $\rightarrow$ subject
NP^VP $\rightarrow$ object

Treebank Transformation, Split & Merge

- Parsers based on this idea:
  - Stanford Parser (Klein & Manning, 2003)
    - hand-written rules
  - Berkeley Parser (Petrov et al., 2006)
    - Split-and-Merge Algorithmus
    - automatically searchs for optimal splits
    - starts with a simple X-bar grammar, automatically performs splits and merges
    - Goal: maximise the probability (Likelihood) of the training data
    - State-of-the-art results on various languages

Treebank Transformation - Problems

- Dramatically increases the number of rules in the grammar $\Rightarrow$ can cause data sparseness
- can result in *overfitting* (very good performance on training data, poor performance on "real" test data)