NLP

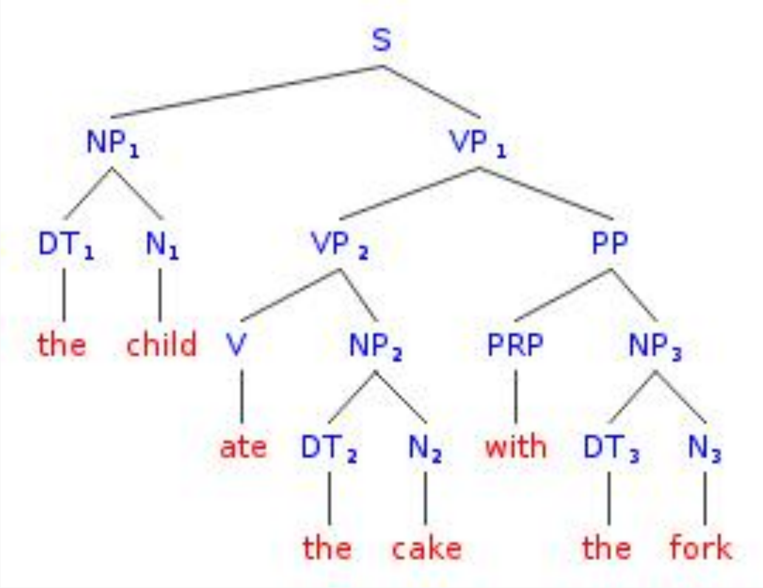
Introduction to NLP

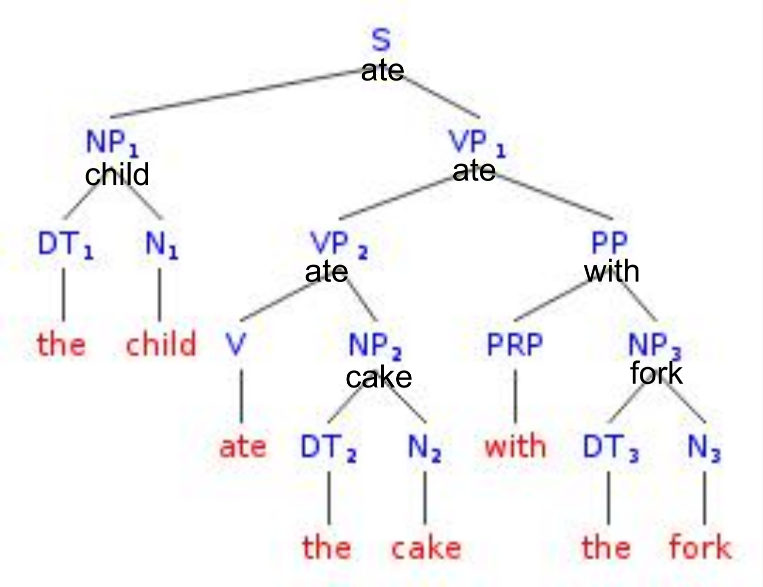
Lexicalized parsing

Limitations of PCFGs

- The probabilities don’t depend on the specific words
  - E.g., *give* someone something (2 arguments) vs. *see* something (1 argument)
- It is not possible to disambiguate sentences based on semantic information
  - E.g., eat pizza with *pepperoni* vs. eat pizza with *fork*
- Lexicalized grammars – idea
  - Use the head of a phrase as an additional source of information
  - VP[ate] -> V[ate]

Lexicalization Example





Collins Parser (1999) 1/2

- Generative, lexicalized model
- Types of rules
  - LHS → $L_n L_{n-1} \ldots L_1 H\ R_1 \ldots R_{m-1} R_m$
  - H gets generated first
  - L gets generated next
  - R gets generated last

Collins Parser (1999) 2/2

- Maximum likelihood estimates

P_{ML} (PPof-IN | VPthink-VB) =
Count (PPof-IN right of the head VPthink-VB) /
Count (symbols right of the head VPthink-VB)

- Smoothing

smoothedP (PPof-IN | VPthink-VB) = λ_1 P (PPof-IN | VPthink-VB) +
+ λ_2 P (PPof-IN | VP-VB) + (1-λ_1-λ_2) P (PPof-IN | VP))

Issues With Lexicalized Grammars

- Sparseness of training data
  - Many probabilities are difficult to estimate from the Penn Treebank
- Combinatorial explosion
  - Need for parameterization

Discriminative Reranking

- A parser may return many parses of a sentence, with small differences in probabilities
- The top returned parse may not necessarily be the best because the PCFG may be deficient
- Other considerations may need to be taken into account
  - parse tree depth
  - left attachment vs. right attachment
  - discourse structure
- Can you think of others features that may affect the reranking?

Answer

- Considerations that may affect the reranking
  - parse tree depth
  - left attachment vs. right attachment
  - discourse structure
- Can you think of others?
  - consistency across sentences
  - or other stages of the NLU pipeline

Discriminative Reranking

- n-best list
  - Get the parser to produce a list of n-best parses (where n can be in the thousands)
- reranking
  - Train a discriminative classifier to rerank these parses based on external information such as a bigram probability score or the amount of right branching in the tree

Sample Performances

- F1 (sentences <= 40 words)
  - Charniak (2000) – 90.1%
  - Charniak and Johnson (2005) – 92% (discriminative reranking)

NLP