Lecture 11

1. Generative grammars, PCFG
    a. CYK algorithm
2. Limitations of PCFG
    a. Lexicalized PCFG
    b. Other grammars
3. Dependency parsing: Graph-based parsing
    a. Scoring
    b. Inference
    c. Learning

Two inference problems for PCFG

1. $T^* = \underset{T \in T(S)}{\text{argmax}}\, p(S,T)$; parsing
2. $p(S) = \sum_{T \in T(S)} p(S,T)$; language modeling

Recall that any PCFG can be written in Chomsky Normal Form:

$X \to Y_1 Y_2, X, Y_1, Y_2 \in N$

$X \to Y, X \in N, Y \in \Sigma$

Advantage of PCFG: Efficient calculation via dynamic programming

Limitation: Can only capture short term dependencies

CYK Algorithm

$\pi[i,j,k] =$ max probability of a tree that starts at position i and ends at position j from non-terminal k

Assume non-terminals are numbered: $N = \{N_1, \dots, N_K\}$, $N_1 = S$

Want to find $\pi[1,n,S]$.

Base Case:

$$\pi[i,i,k] = p(N_k \to w_i | N_k)$$

Recursive Case:

$$\pi[i,j,k] = \max_{l,m,s} \pi[i,s,l]\pi[s,j,m]\, p(N_l, N_m | N_k)$$

Complexity: $O(n^3 k^3)$, can be made $O(n^3 |G|)$

# of subproblems: $n^2 K$

Time per subproblem: nK^2

How can we modify the algorithm for language modeling? Take sum instead of max in recursion.

Limitations

She ate pasta with a fork.

She ate pasta with butter.

The grammars do not care about the actual words in the sentence – it only says which tree is more likely. So in the ambiguity above, it will always parse one of the sentences above incorrectly.

Lexicalization

Recursively propagate head information

How to compute? Compute as normal, but do some linear interpolation as well.