Lecture 10 – Probabilistic Context Free Grammars (PCFG)

1. CFG, Ambiguity
2. PCFGs
3. Algorithms related to PCFG
    a. Parsing
    b. LM
4. Limitations of PCFGs

Why would we want to generate syntax trees?

Applications in sentiment analysis, machine translation, information structure…

Context Free Grammars

$$G =< N, \Sigma, R, S >$$

$N -$ Tags like N, NP, VP, …

$\Sigma -$ words, non-terminals

$R -$ rules

$S -$ start symbol

Example rule: $R: X \to Y_1, Y_2, X \in N, Y_i \in \{N \cup \Sigma\}$

This model cannot tell you which syntax tree is more likely.

We need a notion of probability for syntax trees.

Probabilistic Context Free Grammars

Assign probability to each rule in R

To determine the probability of a syntax tree, multiply the probability of the rules used in the syntax tree.

Algorithms

Given $S, T(S)$ (sentence and trees associated with the sentence)

Want to determine…

1. $T^* = \underset{T \in T(S)}{\operatorname{argmax}} p(S, T)$ (parsing task)
2. $p(S) = \sum_{T \in T(S)} p(S, T)$ (LM task)

Parameters of model: $p(\alpha \to \beta | \alpha) = \frac{count(\alpha \to \beta)}{count(\alpha)}$

CYK Algorithm

Chomsky-Normal Form: Any context free grammar can be translated a grammar with the following types of rules:

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

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

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

$\pi[i, j, k]$ = max probability of a tree that starts at position i, ends at position j, and is derived from N_k.

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

Base Case:

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

Recursive Case:

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