

7. Context-Free Parsing with CKY

Constituency

A constituent is a contiguous sequence of words that functions as coherent unit.

- Substitution: a constituent can be substituted by other phrases of the same type.

Example: Eleanor ate [the pad see ew]. $\rightarrow$ Eleanor ate [it].

- Clefting: a constituent can be moved from its normal position to the beginning of a cleft sentence, which takes the form: It is/was X that ..., where X is the sequence of words we are testing for constituency.

Example: John loves [the red car]. $\rightarrow$ It is [the red car] that John loves.

- Answer Ellipsis: a constituent can appear alone as the answer to a single wh-question.

Example: Papa eats caviar [with a spoon]. $\rightarrow$ How does papa eat caviar [with a spoon].

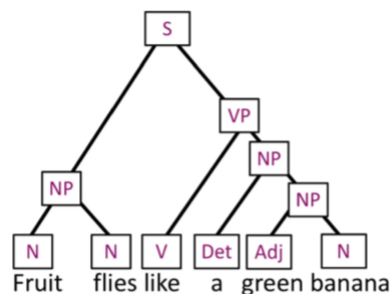
Context Free Grammar

A context-free grammar G is a quadruple (N, S, E, R) consisting of

- A finite set of non-terminal symbols N ; written in upper-case letters, eg. N_1, N_2, N_3
- A distinguished start non-terminal S
- An alphabet of terminal symbols Σ ; written as lower-case letters, eg. a_1, a_2, a_3 .
- A set of production rules R of the form $N \rightarrow \alpha$, where $N \in N$ and $\alpha \in (N \cup \Sigma)^*$

Example:

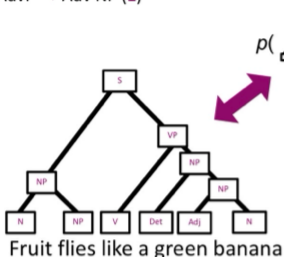
$S \rightarrow NP VP \mid N VP$
 $NP \rightarrow Det N \mid Det NP \mid N N \mid Adj N$
 $PP \rightarrow Prp NP$
 $VP \rightarrow V NP \mid V PP$
 $AdvP \rightarrow Adv NP$
 $N \rightarrow fruit \mid flies \mid banana$
 $V \rightarrow flies \mid like$
 $Det \rightarrow a$
 $Adj \rightarrow green$
 $Adv \rightarrow like$



Probabilistic CFGs

$S \rightarrow NP VP (.6) \mid N VP (.4)$
 $NP \rightarrow Det N (.4) \mid Det NP (.2) \mid N N (.1) \mid Adj N (.3)$
 $PP \rightarrow Prp NP (.1)$
 $VP \rightarrow V NP (.5) \mid V PP (.5)$
 $AdvP \rightarrow Adv NP (.1)$

$N \rightarrow fruit (.3) \mid flies (.3) \mid banana (.4)$
 $V \rightarrow flies (.4) \mid like (.6)$
 $Det \rightarrow a (.1)$
 $Adj \rightarrow green (.1)$
 $Adv \rightarrow like (.1)$



$$p(\text{tree}) =$$

$$\begin{aligned}
 & p(S \rightarrow NP VP) \times p(NP \rightarrow N NP) \times \\
 & p(N \rightarrow Fruit) \times p(NP \rightarrow flies) \times \\
 & p(VP \rightarrow V NP) \times p(V \rightarrow like) \times \\
 & p(NP \rightarrow Det NP) \times p(Det \rightarrow a) \times \\
 & p(NP \rightarrow Adj N) \times p(Adj \rightarrow green) \times \\
 & p(N \rightarrow banana)
 \end{aligned}$$

- PCFGs are locally normalized

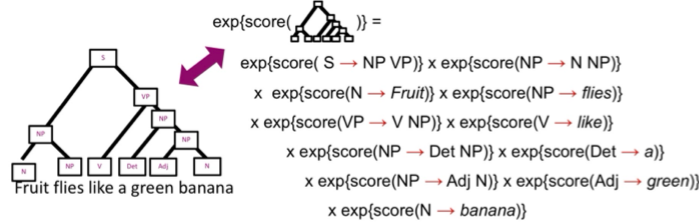
- For all rules with the same left-hand side N , $(N \rightarrow \alpha, \dots, N \rightarrow \alpha_k)$, their prob. must sum to 1: $\sum_{k=1}^k p(\alpha_k | N) = 1$.

$$0.3 + 0.3 + 0.4 = 1$$

Weighted CFGs

$S \rightarrow NP VP$ (2) | $N VP$ (1)
 $NP \rightarrow Det N$ (4) | $Det NP$ (2) | $N N$ (1) | $Adj N$ (3)
 $PP \rightarrow Prp NP$ (5)
 $VP \rightarrow V NP$ (2) | $V PP$ (2)
 $AdvP \rightarrow Adv NP$ (1)

$N \rightarrow fruit$ (2) | $flies$ (4) | $banana$ (6)
 $V \rightarrow flies$ (1) | $like$ (2)
 $Det \rightarrow a$ (1)
 $Adj \rightarrow green$ (3)
 $Adv \rightarrow like$ (4)



- NCFGs are globally normalized
- $P(t) = \frac{1}{Z} \prod_{r \in t} \exp\{\text{score}(r)\}$
- $Z = \sum_{t \in T} \prod_{r \in t} \exp\{\text{score}(r)\}$
- T is countably infinite

Chomsky Normal Form

A grammar is in Chomsky Normal Form (CNF) if all productions have the form:

- $N_1 \rightarrow N_2 N_3$, where N_1, N_2, N_3 are non-terminals
- $N \rightarrow \alpha$, where N is a non-terminal and α is a terminal
- $S \rightarrow \epsilon$, where S is the start symbol and ϵ is the empty string.

Parsing into CNF

$$P(t|s) = \frac{1}{Z(s)} \prod_{(x \rightarrow yz) \in t} \exp\{\text{score}(x \rightarrow yz)\} \cdot \prod_{(x \rightarrow x) \in t} \exp\{\text{score}(x \rightarrow x)\}$$

$$Z(s) = \sum_{t \in T(s)} \prod_{(x \rightarrow yz) \in t} \exp\{\text{score}(x \rightarrow yz)\} \cdot \prod_{(x \rightarrow x) \in t} \exp\{\text{score}(x \rightarrow x)\}$$

* Assume $S \neq \epsilon$

non-terminal production

terminal production

Cocke-Kasami-Younger (CKY) Algorithm

$O(N^3 |R|)$, N length of sentence $|R|$ size rule set

Algorithm 6.1

def WEIGHTEDCKY($s, \langle \mathcal{N}, S, \Sigma, \mathcal{R} \rangle, \text{score}$):

$N \leftarrow |s|$

$\text{chart} \leftarrow 0$

for $n = 1, \dots, N$:

for $X \rightarrow s_n \in \mathcal{R}$:

$\text{chart}[n, n+1, X] += \exp\{\text{score}(X \rightarrow s_n)\}$

▷ Handles single word tokens

end for

end for

for $\text{span} = 2, \dots, N$:

for $i = 1, \dots, N - \text{span} + 1$:

▷ i marks the beginning of the span

$k \leftarrow i + \text{span}$

▷ k marks the end of the span

for $j = i + 1, \dots, k - 1$:

▷ j marks the breaking point of the span

for $X \rightarrow YZ \in \mathcal{R}$:

$\text{chart}[i, k, X] += \exp\{\text{score}(X \rightarrow YZ)\} \times \text{chart}[i, j, Y] \times \text{chart}[j, k, Z]$

end for

end for

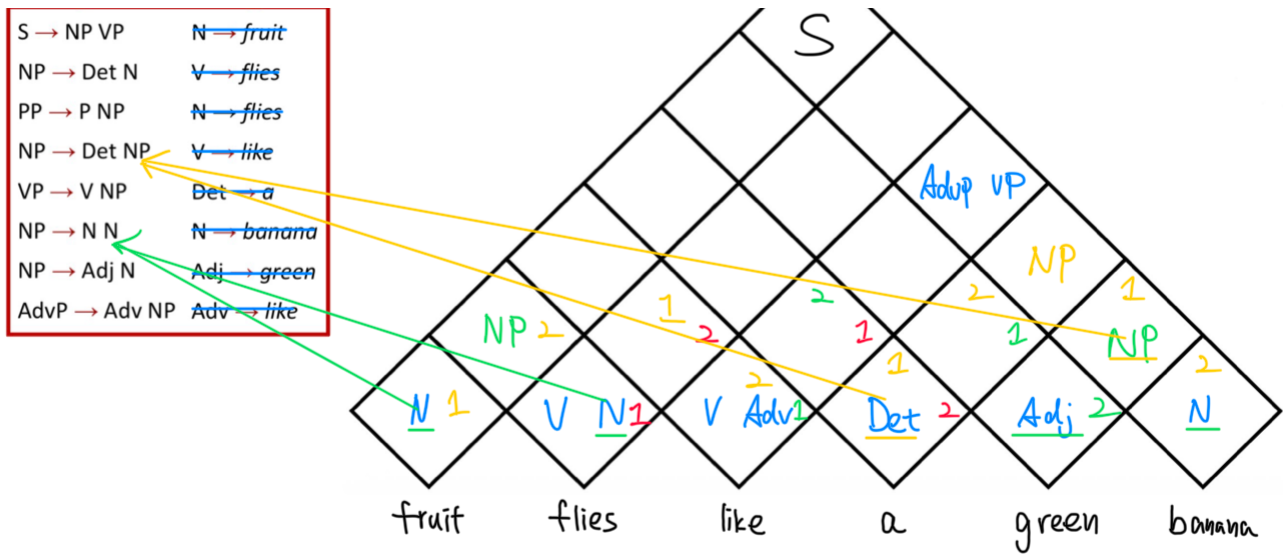
end for

end for

return $\text{chart}[1, N+1, S]$

Example:

Grammar



Generalize With Semirings

Algorithm 6.2

```

def SEMIRINGCKY( $s, \langle \mathcal{N}, S, \Sigma, \mathcal{R} \rangle, \text{score}$ ):
     $N \leftarrow |s|$ 
     $\text{chart} \leftarrow 0$ 
    for  $n = 1, \dots, N$ :
        for  $X \rightarrow s_n \in \mathcal{R}$ :
             $\text{chart}[n, n+1, X] \oplus= \exp\{\text{score}(X \rightarrow s_n)\}$   $\triangleright$  Handles the single word tokens
        end for
    end for
    for  $\text{span} = 2, \dots, N$ :
        for  $i = 1, \dots, N - \text{span} + 1$ :  $\triangleright i$  marks the beginning of the span
             $k \leftarrow i + \text{span}$   $\triangleright k$  marks the end of the span
            for  $j = i + 1, \dots, k - 1$ :  $\triangleright j$  marks the breaking point of the span
                for  $X \rightarrow YZ \in \mathcal{R}$ :
                     $\text{chart}[i, k, X] \oplus= \exp\{\text{score}(X \rightarrow YZ)\} \otimes \text{chart}[i, j, Y] \otimes \text{chart}[j, k, Z]$ 
                end for
            end for
        end for
    end for
    return  $\text{chart}[1, N+1, S]$ 

```

