Special Topics in Natural Language Processing

CS698O

Ashutosh Modi
CSE Department, IIT Kanpur

Lecture 16: Parsing 2
Feb 9, 2020

Context Free Grammar (CFG)

- Constituent: Group of words behaving as a single unit. E.g. phrases

  NP: Harry the Horse

- Constituent Structure is modeled using ***Context Free Grammar (CFG)***
- CFG contains set of ***rules (or productions)*** and a ***lexicon***
  - Rules express the ways that symbols of the language can be grouped and ordered together

$$NP \rightarrow Det \;\; Nominal \qquad\qquad Det \rightarrow a$$

$$NP \rightarrow ProperNoun \qquad\qquad Det \rightarrow the$$

$$Nominal \rightarrow Noun \mid NominalNoun \qquad\qquad Noun \rightarrow flight$$

  - Lexicon contains words and ***symbols*** in the language

Context Free Grammar (CFG)

Symbols

Terminal

Words in the language
(introduced via rules in the lexicon)

Non-Terminal

Clustering or generalization of terminals
(e.g. NP, POS, etc.)

$$\textbf{Non-Terminal} \rightarrow \textbf{Terminal} \mid \textbf{Non-Terminal}$$

$$NP \rightarrow Det \;\; Nominal$$
$$Noun \rightarrow flight$$

CFG: DEFINITION

A context free grammar (CFG) is a 4-tuple $G = (N, \Sigma, R, S)$

- N is a finite set of non-terminal symbols.
- Σ is a finite set of terminal symbols.
- R is a finite set of rules of the form $X \rightarrow Y_1, \ldots, Y_n$,
  where $X \in N,\ n \geq 0$, and $Y_i \in (N \cup \Sigma)\ for\ i = 1, 2, \ldots, n$
- $S \in N$ is a distinguished start symbol.

CFG: EXAMPLE

$S = \texttt{S}$

$N = \{\texttt{S}, \texttt{NP}, \texttt{VP}, \texttt{PP}, \texttt{DT}, \texttt{Vi}, \texttt{Vt}, \texttt{NN}, \texttt{IN}\}$

$\Sigma = \{\texttt{sleeps, saw, woman, telescope, dog man, the, with, in}\}$

$R =$

S → NP VP
VP → Vi VP → Vt NP VP → VP PP
NP → DT NN NP → NP PP
PP → IN NP

Vi → sleeps Vt → saw
NN → man NN → woman NN → telescope NN → dog
DT → the
IN → with IN → in

LEFT MOST DERIVATION

Given a context free grammar G,

a left most derivation is a sequence of strings $s_1, s_2, \ldots, s_n$

where,

- $s_1 = \mathtt{S}$ i.e., s_1 consists of a single element, the start symbol
- $s_n = \Sigma^*,$ i.e. s_n is made up of terminal symbols only
- Each s_i for $i = 2...n$ is derived from $s_i - 1$

  by picking the left-most non-terminal X in $s_i - 1$

  and replacing it by some β where

  $X \rightarrow \beta$ is a rule in R

CFG LANGUAGE

Each left most derivation of CFG G results in $s_n \in \Sigma^*$,
s_n is called *Yield* of the derivation

A string $s \in \Sigma^*$ is said to be in the *language* defined by the CFG, if there is at least one derivation whose yield is s

PROBABILISTIC CFGs (PCFGs)

A PCFG consists of:

1. A context free grammar $G = (N, \Sigma, R, S)$
2. A parameter $q(\alpha \rightarrow \beta)$ for each rule $\alpha \rightarrow \beta \in R$

q can be interpreted as the conditional probability of choosing rule $\alpha \rightarrow \beta$ in a left most derivation, given that the non-terminal being expanded is α

$$\sum_{\substack{\alpha \rightarrow \beta \in R \\ \alpha = X \in N}} q(\alpha \rightarrow \beta) = 1$$

$$q(\alpha \rightarrow \beta) \geq 0 \ \forall \ \alpha \rightarrow \beta \in R$$

PROBABILISTIC CFGs (PCFGs)

A PCFG consists of:

1. A context free grammar $G = (N, \Sigma, R, S)$
2. A parameter $q(\alpha \to \beta)$ for each rule $\alpha \to \beta \in R$

Given a parse tree $t \in \mathcal{T}_G$ containing rules $\alpha_1 \to \beta_1, \alpha_2 \to \beta_2, \ldots, \alpha_n \to \beta_n$

$$p_{\text{PCFG}}(t) = \prod_{i=1}^{n} q(\alpha_i \to \beta_i)$$

PCFG

$$G = (N, \Sigma, R, S, q)$$

$$\sum_{\substack{\alpha \to \beta \in R \\ \alpha = X \in N}} q(\alpha \to \beta) = 1 \qquad q(\alpha \to \beta) \geq 0 \ \forall \ \alpha \to \beta \in R$$

Two Questions:

1. ***Learning Problem*** : How do we learn the parameters (probabilities)?
2. ***Decoding Problem***: Given a sentence *s* how do we find the most likely tree?

$$\underset{t \in \mathcal{T}_G(s)}{\operatorname{argmax}} \ p(t)$$

TREEBANKS

- Linguists have annotated sentences in a large text corpus with syntactic parse
- Each sentence in the corpus has a parse tree
- This corpus of sentences and parse trees is referred as **Treebank**

TREEBANKS

- Linguists have annotated sentences in a large text corpus with syntactic parse
- Each sentence in the corpus has a parse tree
- This corpus of sentences and parse trees is referred as **Treebank**
- One of the most famous treebank used for English is **Penn Treebank**.
- Created from Wall Street Journal, Brown, Switchboard and ATIS corpora.

PENN TREEBANK EXAMPLE

That cold, empty sky was full of fire and light.

PENN TREEBANK EXAMPLE

That cold, empty sky was full of fire and light.

$$
\begin{array}{l}
((S \\
\quad (NP\text{-}SBJ\ (DT\ That\ ) \\
\quad\quad (JJ\ cold\ )(,\ ,\ ) \\
\quad\quad (JJ\ empty\ )(NP\ sky\ )) \\
\quad (VP\ (VBD\ was\ ) \\
\quad\quad (ADJP\text{-}PRD\ (JJ\ full\ ) \\
\quad\quad\quad (PP\ (IN\ of\ ) \\
\quad\quad\quad\quad (NP\ (NN\ fire\ ) \\
\quad\quad\quad\quad\quad (CC\ and\ ) \\
\quad\quad\quad\quad\quad (NN\ light\ )\ )\ )\ )\ )\ ) \\
\quad (.\ .\ )\quad)\ )
\end{array}
$$

PENN TREEBANK EXAMPLE

$((S$

$(NP\text{-}SBJ\ (DT\ That\ )$

$(JJ\ cold\ )(,\ ,\ )$

$(JJ\ empty\ )(NP\ sky\ ))$

$(VP\ (VBD\ was\ )$

$(ADJP\text{-}PRD\ (JJ\ full\ )$

$(PP\ (IN\ of\ )$

$(NP\ (NN\ fire\ )$

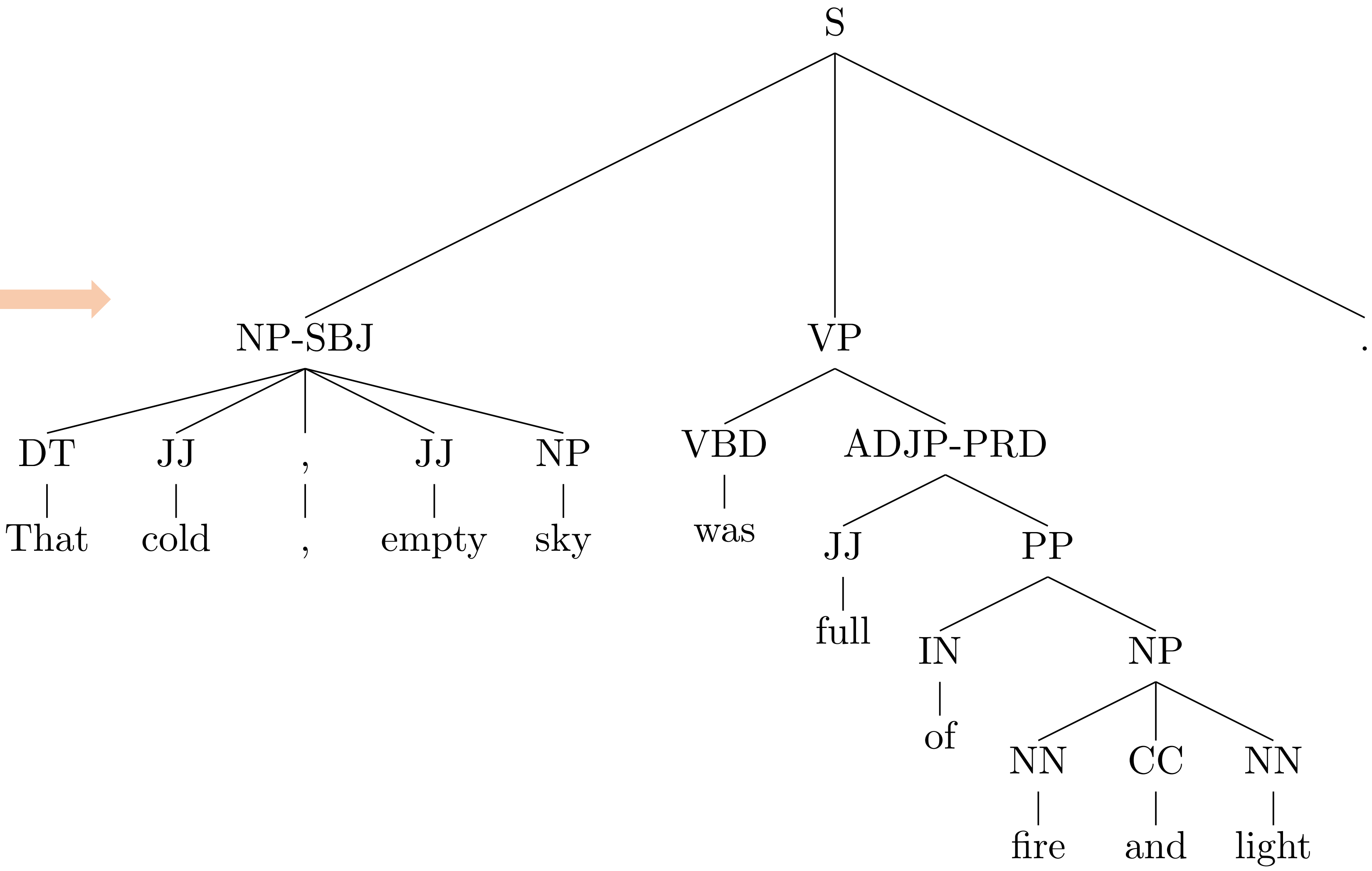
$(CC\ and\ )$

$(NN\ light\ )\ )\ )\ )\ )$

$(.\ .\ )\quad)\ )$

PENN TREEBANK EXAMPLE

```
((S
    (NP-SBJ (DT That )
      (JJ cold )(, , )
      (JJ empty )(NP sky ))
    (VP (VBD was )
      (ADJP-PRD (JJ full )
         (PP (IN of )
            (NP (NN fire )
               (CC and )
               (NN light ) ) ) ) )
    (. . )   ) )
```



PCFG

$$G = (N, \Sigma, R, S, q)$$

$$\sum_{\substack{\alpha \rightarrow \beta \in R \\ \alpha = X \in N}} q(\alpha \rightarrow \beta) = 1 \qquad\qquad q(\alpha \rightarrow \beta) \geq 0 \;\forall\; \alpha \rightarrow \beta \in R$$

Two Questions:

1. ***Learning Problem*** : How do we learn the parameters (probabilities)?
2. ***Decoding Problem***: Given a sentence *s* how do we find the most likely tree?

$$\underset{t \in \mathcal{T}_G(s)}{\operatorname{argmax}} \; p(t)$$

LEARNING IN PCFGs

- We are given a treebank

$$\{s_i, t_i\}_{i=1}^{m}$$

LEARNING IN PCFGs

- We are given a treebank

$$\{s_i, t_i\}_{i=1}^{m}$$

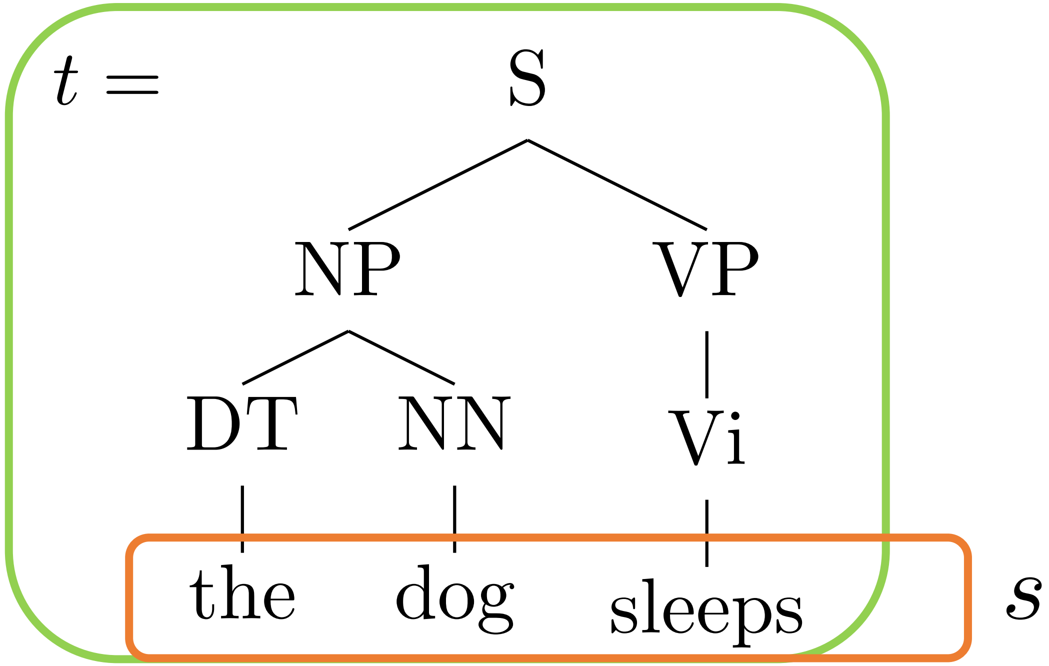
$$s_i = \mathbf{Yield}(t_i)$$

LEARNING IN PCFGs

- We are given a treebank

$$\{s_i, t_i\}_{i=1}^{m}$$

$$s_i = \textbf{Yield}(t_i)$$



LEARNING IN PCFGs

$$G = (N, \Sigma, R, S, q)$$

$$\sum_{\substack{\alpha \to \beta \in R \\ \alpha = X \in N}} q(\alpha \to \beta) = 1 \qquad\qquad q(\alpha \to \beta) \geq 0 \ \forall \ \alpha \to \beta \in R$$

- Learning problem is about how to get the values of q

LEARNING IN PCFGs

$$G = (N, \Sigma, R, S, q)$$

$$\sum_{\substack{\alpha \to \beta \in R \\ \alpha = X \in N}} q(\alpha \to \beta) = 1 \qquad\qquad q(\alpha \to \beta) \geq 0 \ \forall \ \alpha \to \beta \in R$$

- Learning problem is about how to get the values of q
- We are given training corpus (Treebank), we can do MLE

LEARNING IN PCFGs

$$G = (N, \Sigma, R, S, q)$$

$$\sum_{\substack{\alpha \to \beta \in R \\ \alpha = X \in N}} q(\alpha \to \beta) = 1 \qquad q(\alpha \to \beta) \geq 0 \; \forall \; \alpha \to \beta \in R$$

$$\boxed{q(\alpha \to \beta)_{ML} = \frac{C(\alpha \to \beta)}{C(\alpha)}}$$

PCFG

$$G = (N, \Sigma, R, S, q)$$

$$\sum_{\substack{\alpha \to \beta \in R \\ \alpha = X \in N}} q(\alpha \to \beta) = 1 \qquad q(\alpha \to \beta) \geq 0 \; \forall \; \alpha \to \beta \in R$$

Two Questions:

1. ***Learning Problem*** : How do we learn the parameters (probabilities)?
2. ***Decoding Problem***: Given a sentence *s* how do we find the most likely tree?

$$\underset{t \in \mathcal{T}_G(s)}{\text{argmax}} \; p(t)$$

CHOMSKY NORMAL FORM (CNF)

A CFG $G = (N, \Sigma, R, S)$ is in Chomsky Normal Form (CNF) if each rule $\alpha \rightarrow \beta \in R$ takes one of the two following forms:

$$X \rightarrow Y_1, Y_2 \text{ where, } X \in N,\ Y_1 \in N,\ Y_2 \in N$$

OR

$$X \rightarrow Y \text{ where, } X \in N,\ Y \in \Sigma$$

CHOMSKY NORMAL FORM (CNF)

$S = \texttt{S}$

$N = \{\texttt{S}, \texttt{NP}, \texttt{VP}, \texttt{PP}, \texttt{DT}, \texttt{Vi}, \texttt{Vt}, \texttt{NN}, \texttt{IN}\}$

$\Sigma = \{\texttt{sleeps, saw, woman, telescope, dog man, the, with, in}\}$

$R, q =$

S → NP VP	1.0
VP → Vt NP	0.8
VP → VP PP	0.2
NP → DT NN	0.8
NP → NP PP	0.2
PP → IN NP	1.0

Vi → sleeps	1.0
Vt → saw	1.0
NN → man	0.1
NN → woman	0.1
NN → telescope	0.3
NN → dog	0.5
DT → the	1.0
IN → with	0.6
IN → in	0.4

PCFG: DECODING

Given, $PCFG\ (G = (N, \Sigma, R, S, q))$ in CNF form,

a sentence $s = x_1, \ldots, x_n$, where, x_i is the i^{th} word in the sentence,

we are interested in

$$\underset{t \in \mathcal{T}_G(s)}{\text{argmax}}\ p(t)$$

PCFG: DECODING

Given, $PCFG\ (G = (N, \Sigma, R, S, q))$ in CNF form,

a sentence $s = x_1, \ldots, x_n$, where, x_i is the i^{th} word in the sentence,

we are interested in

$$\underset{t \in \mathcal{T}_G(s)}{\operatorname{argmax}}\ p(t)$$

Decoding can be done using CKY (Cocke-Kasami-Younger) algorithm

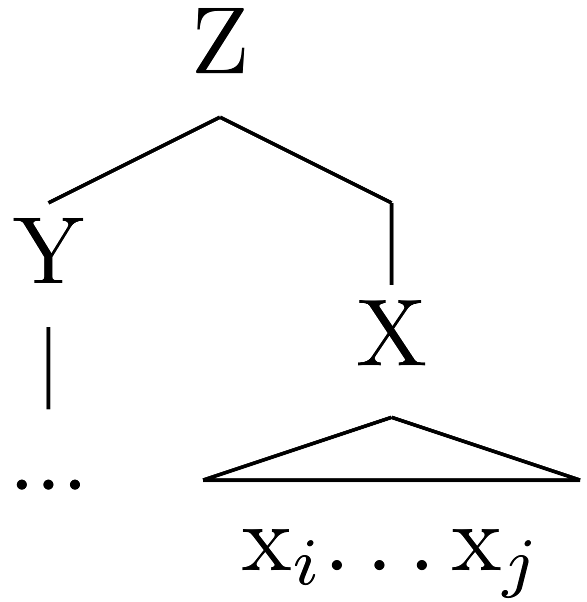
It is a Dynamic Programming algorithm

PCFG: DECODING

$$\mathcal{T}(i, j, X) = \text{Set of all parse trees for words } x_i \ldots x_j \ (\forall\, 1 \le i \le j \le n)$$
$$\text{with } X \in N \text{ as the root}$$

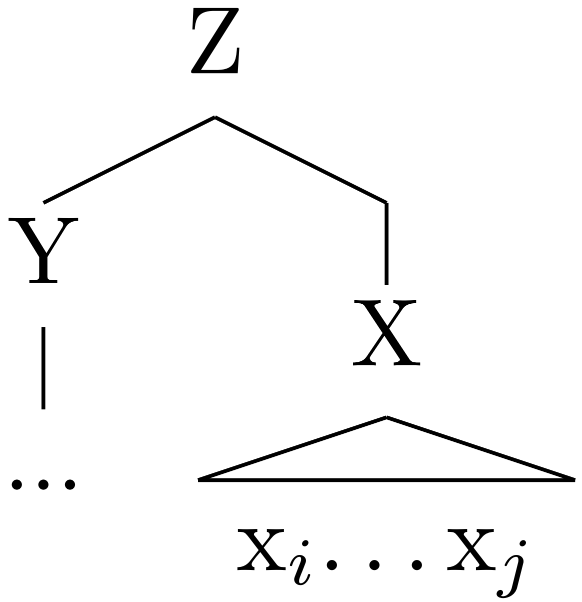
PCFG: DECODING

$$\mathcal{T}(i, j, X) = \text{Set of all parse trees for words } x_i \ldots x_j \ (\forall\ 1 \leq i \leq j \leq n)$$
$$\text{with } X \in N \text{ as the root}$$



PCFG: DECODING

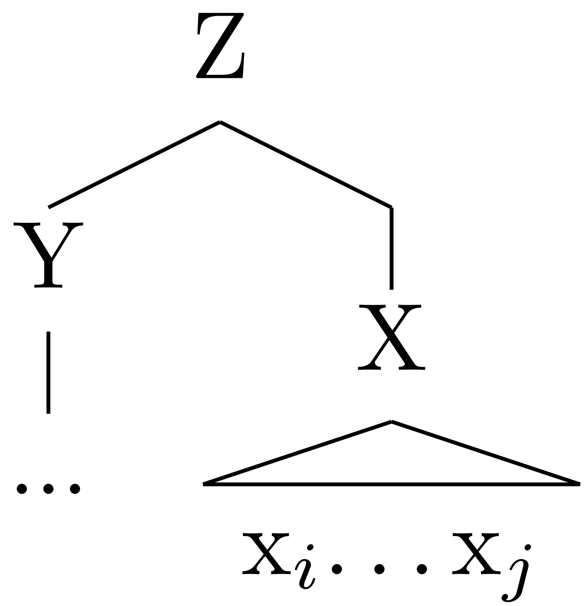
$$\mathcal{T}(i, j, X) = \text{Set of all parse trees for words } x_i \ldots x_j \; (\forall\, 1 \leq i \leq j \leq n)$$
$$\text{with } X \in N \text{ as the root}$$



$$\pi(i, j, X) := \max_{t \in \mathcal{T}(i,j,X)} p(t)$$

PCFG: DECODING

$$\mathcal{T}(i, j, X) = \text{Set of all parse trees for words } x_i \ldots x_j \ (\forall\ 1 \leq i \leq j \leq n)$$
$$\text{with } X \in N \text{ as the root}$$

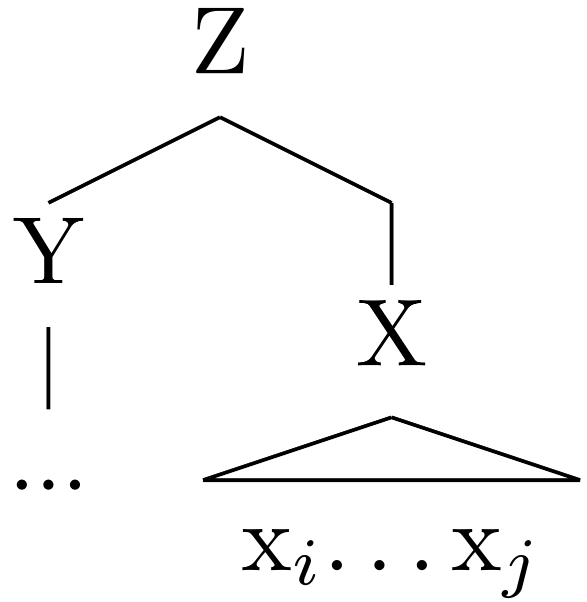


$$\boxed{\pi(i, j, X) := \max_{t \in \mathcal{T}(i,j,X)} p(t)}$$

$$p(t) = \prod_{k=1}^{r} q(\alpha_k \to \beta_k)$$

PCFG: DECODING

$$\mathcal{T}(i, j, X) = \text{Set of all parse trees for words } x_i \ldots x_j \ (\forall\ 1 \leq i \leq j \leq n)$$
$$\text{with } X \in N \text{ as the root}$$

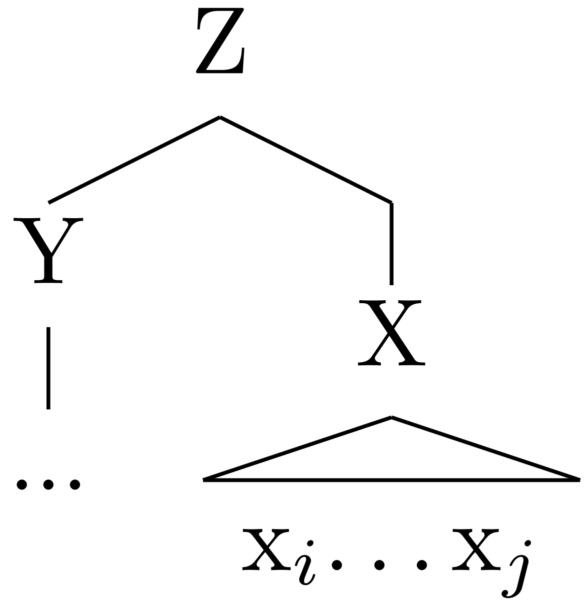


$$\boxed{\pi(i, j, X) := \max_{t \in \mathcal{T}(i,j,X)} p(t)}$$

$$\pi(1, n, S) = \, ?$$

PCFG: DECODING

$$\mathcal{T}(i,j,X) = \text{Set of all parse trees for words } x_i \ldots x_j \; (\forall\, 1 \le i \le j \le n)$$
$$\text{with } X \in N \text{ as the root}$$



$$\boxed{\pi(i,j,X) := \max_{t \in \mathcal{T}(i,j,X)} p(t)}$$

$$\pi(1,n,S) = \max_{t \in \mathcal{T}_G(s)} p(t)$$

PCFG DECODING: CYK ALGORITHM

Dynamic Programming based algorithm

PCFG DECODING: CYK ALGORITHM

Base case, $\forall\ 1 \leq i \leq n$ and $\ X \in N$

$$\pi(i, i, X) = \begin{cases} q(X \rightarrow x_i) & \text{If } X \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

PCFG DECODING: CYK ALGORITHM

Base case, $\forall\ 1 \leq i \leq n$ and $X \in N$

$$\pi(i,i,X) = \begin{cases} q(X \rightarrow x_i) & \text{If } X \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Recursion, $\forall\ 1 \leq i,j \leq n$ and $X \in N$

$$\pi(i,j,X) = \max_{\substack{X \rightarrow YZ \in R, \\ s \in (i \ldots (j-1))}} \left(q(X \rightarrow YZ) \times \pi(i,s,Y) \times \pi(s+1,j,Z)\right)$$

PCFG DECODING: CYK ALGORITHM

Base case, $\forall\ 1 \leq i \leq n$ and $X \in N$

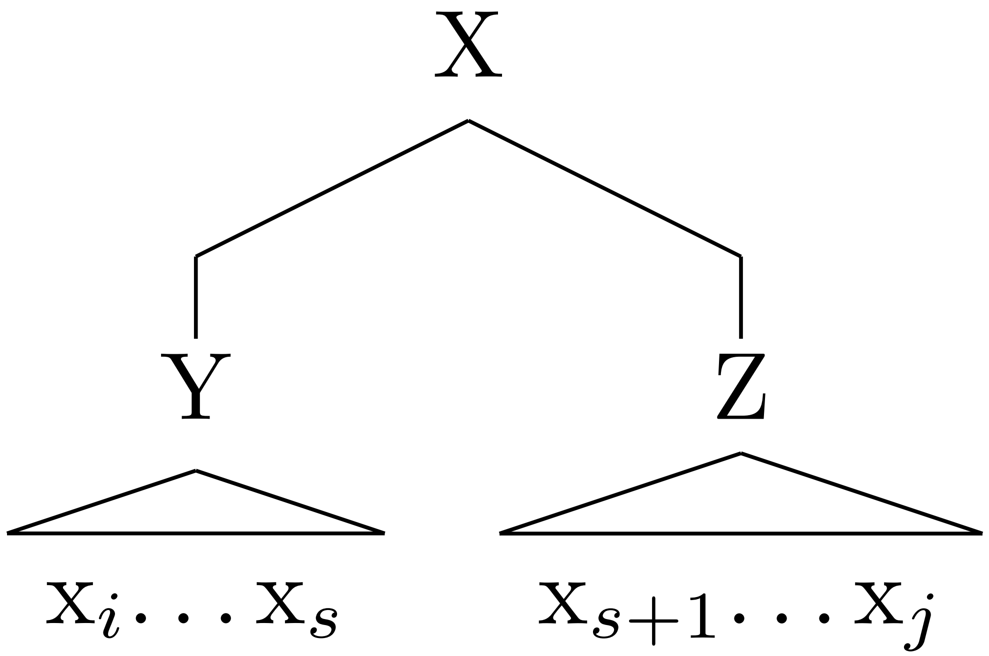
$$\boxed{\pi(i,i,X) = \begin{cases} q(X \rightarrow x_i) & \text{If } X \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}}$$

$$\begin{array}{ccccc} \mathrm{X} & & \mathrm{Y} & & \mathrm{Z} \\ \pi(1,1,X)\ \Big| & & \pi(2,2,Y)\ \Big| & \cdots\cdots & \Big|\ \pi(n,n,Z) \\ \mathrm{x}_1 & & \mathrm{x}_2 & & \mathrm{x}_n \end{array}$$

PCFG DECODING: CYK ALGORITHM

Recursion, $\forall\ 1 \leq i, j \leq n$ and $X \in N$

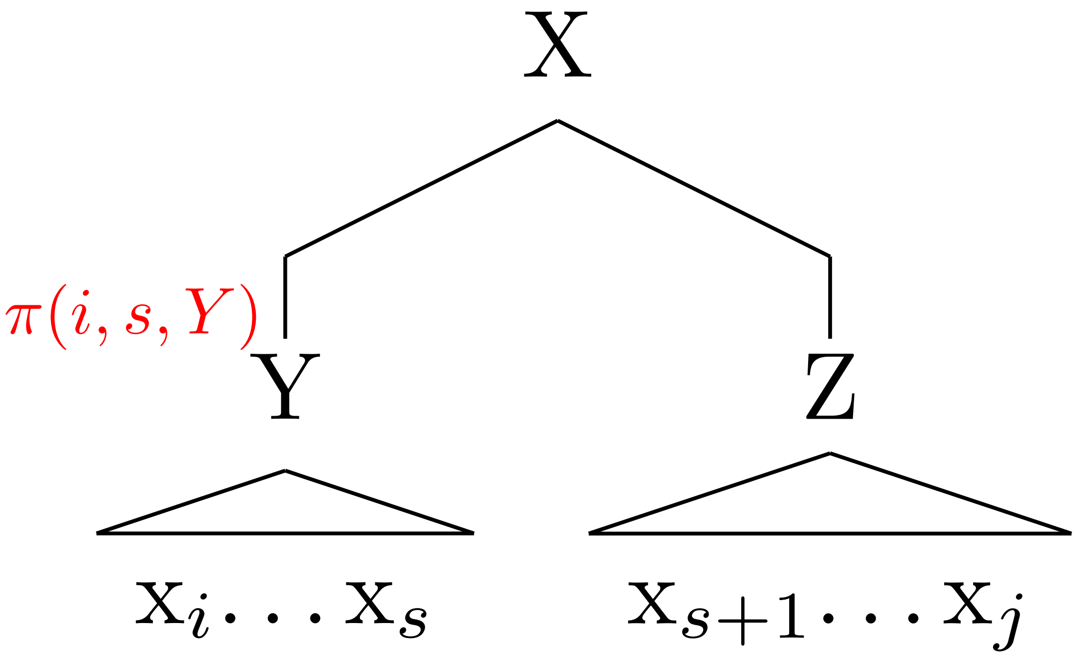
$$\boxed{\pi(i,j,X) = \max_{\substack{X \to YZ \in R,\\ s \in (i \ldots (j-1))}} \left(q(X \to YZ) \times \pi(i,s,Y) \times \pi(s+1,j,Z)\right)}$$



PCFG DECODING: CYK ALGORITHM

Recursion, $\forall\ 1 \le i, j \le n$ and $X \in N$

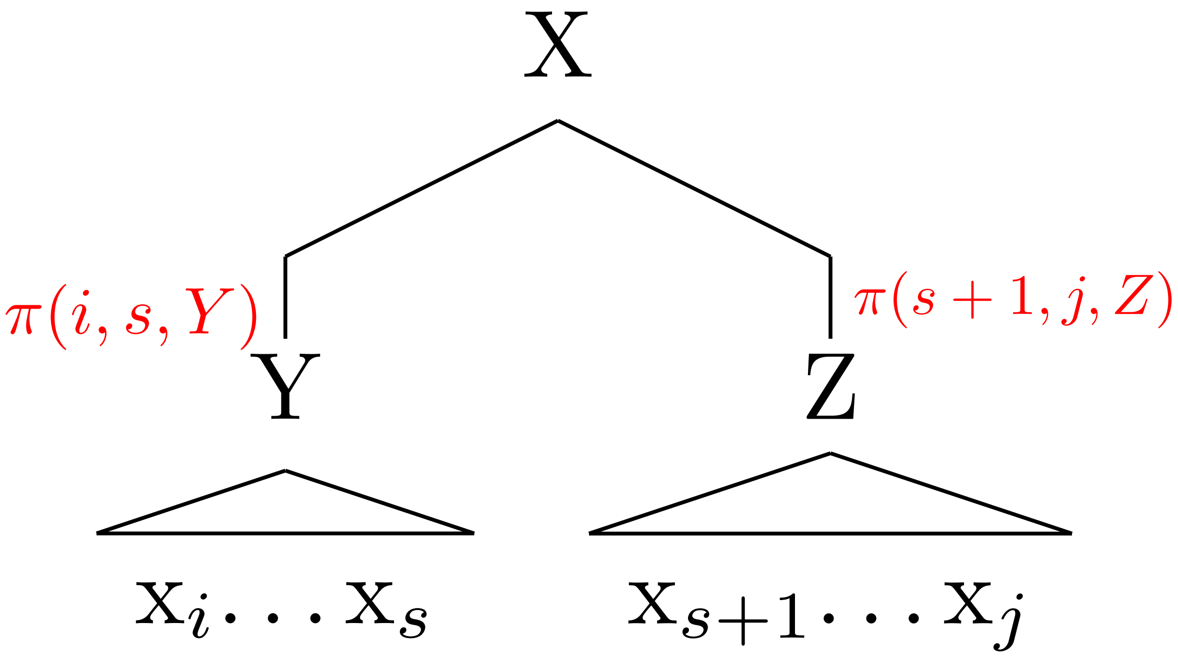
$$\boxed{\pi(i,j,X) = \max_{\substack{X \to YZ \in R,\\ s \in (i \ldots (j-1))}} \left(q(X \to YZ) \times \pi(i,s,Y) \times \pi(s+1,j,Z) \right)}$$



PCFG DECODING: CYK ALGORITHM

Recursion, $\forall\ 1 \leq i, j \leq n$ and $X \in N$

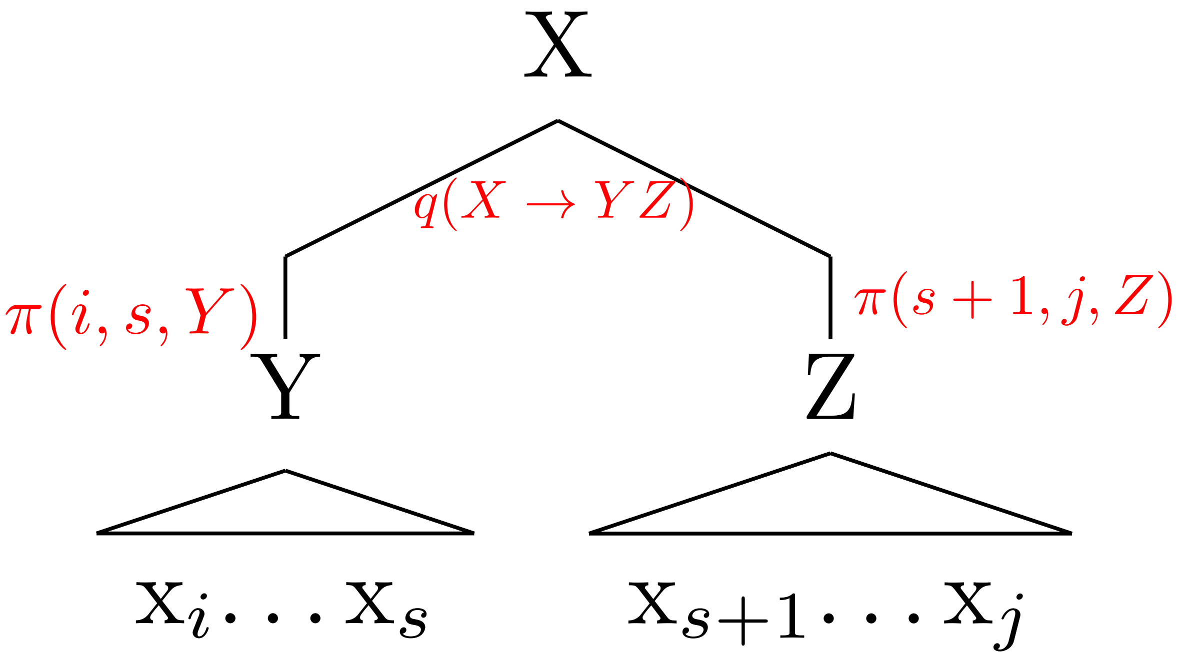
$$\boxed{\pi(i, j, X) = \max_{\substack{X \to Y Z \in R, \\ s \in (i \ldots (j-1))}} \left(q(X \to Y Z) \times \pi(i, s, Y) \times \pi(s+1, j, Z) \right)}$$



PCFG DECODING: CYK ALGORITHM

Recursion, $\forall\ 1 \le i, j \le n$ and $X \in N$

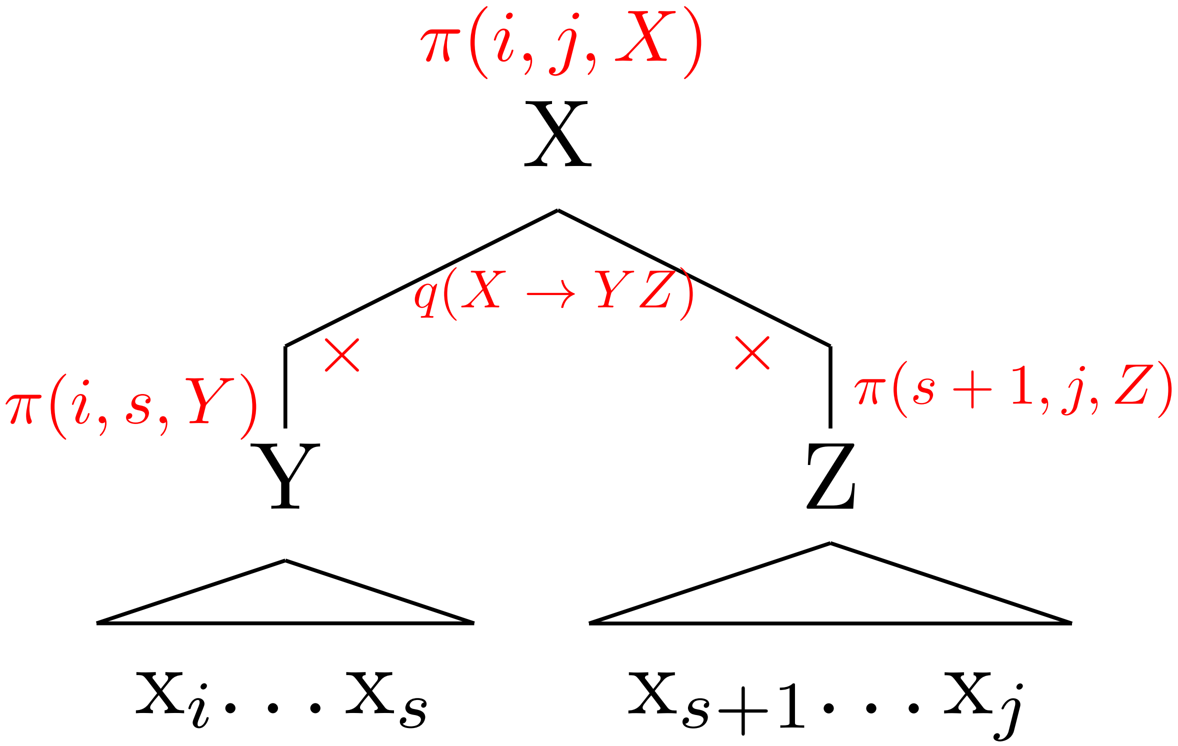
$$\boxed{\pi(i,j,X) = \max_{\substack{X \to Y Z \in R, \\ s \in (i \ldots (j-1))}} \left(q(X \to Y Z) \times \pi(i,s,Y) \times \pi(s+1,j,Z) \right)}$$



PCFG DECODING: CYK ALGORITHM

Recursion, $\forall\ 1 \leq i, j \leq n$ and $X \in N$

$$\boxed{\pi(i,j,X) = \max_{\substack{X \to Y\,Z \in R,\\ s \in (i \ldots (j-1))}} \left(q(X \to Y\,Z) \times \pi(i,s,Y) \times \pi(s+1,j,Z) \right)}$$



CYK ALGORITHM

Input: sentence $s = x_1 \ldots x_n$ and $PCFG$ $G = (N, \Sigma, R, S, q)$ in CNF form

CYK ALGORITHM

Input: sentence $s = x_1 \dots x_n$ and $PCFG\ G = (N, \Sigma, R, S, q)$ in CNF form

Initialization: $\forall\ 1 \leq i \leq n$ and $\ X \in N$

$$\pi(i, i, X) = \begin{cases} q(X \to x_i) & \text{If } X \to x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

CYK ALGORITHM

Input: sentence $s = x_1 \dots x_n$ and $PCFG$ $G = (N, \Sigma, R, S, q)$ in CNF form

Initialization: $\forall\ 1 \leq i \leq n$ and $X \in N$

$$\pi(i,i,X) = \begin{cases} q(X \rightarrow x_i) & \text{If } X \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

CYK ALGORITHM

Input: sentence $s = x_1 \dots x_n$ and $PCFG$ $G = (N, \Sigma, R, S, q)$ in CNF form

Initialization: $\forall\ 1 \leq i \leq n$ and $\ X \in N$

$$\pi(i, i, X) = \begin{cases} q(X \to x_i) & \text{If } X \to x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

For $k = 1, \dots, (n-1)$

CYK ALGORITHM

Input: sentence $s = x_1 \ldots x_n$ and $PCFG\ G = (N, \Sigma, R, S, q)$ in CNF form

Initialization: $\forall\ 1 \leq i \leq n$ and $\ X \in N$

$$\pi(i,i,X) = \begin{cases} q(X \to x_i) & \text{If } X \to x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

For $k = 1, \ldots, (n-1)$

For $i = 1, \ldots, (n-k)$

CYK ALGORITHM

Input: sentence $s = x_1 \dots x_n$ and $PCFG\ G = (N, \Sigma, R, S, q)$ in CNF form

Initialization: $\forall\ 1 \leq i \leq n$ and $\ X \in N$

$$\pi(i, i, X) = \begin{cases} q(X \rightarrow x_i) & \text{If } X \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

For $k = 1, \dots, (n-1)$

For $i = 1, \dots, (n-k)$

$j = i + k$

CYK ALGORITHM

Input: sentence $s = x_1 \dots x_n$ and $PCFG\ G = (N, \Sigma, R, S, q)$ in CNF form

Initialization: $\forall\ 1 \leq i \leq n$ and $\ X \in N$

$$\pi(i,i,X) = \begin{cases} q(X \rightarrow x_i) & \text{If } X \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

For $k = 1, \dots, (n-1)$

For $i = 1, \dots, (n-k)$

$j = i + k$

$\forall X \in N,$ calculate

$$\pi(i,j,X) = \max_{\substack{X \rightarrow YZ \in R, \\ s \in (i \dots (j-1))}} \left(q(X \rightarrow YZ) \times \pi(i,s,Y) \times \pi(s+1,j,Z)\right)$$

CYK ALGORITHM

Input: sentence $s = x_1 \ldots x_n$ and $PCFG\ G = (N, \Sigma, R, S, q)$ in CNF form

Initialization: $\forall\ 1 \leq i \leq n$ and $X \in N$

$$\pi(i,i,X) = \begin{cases} q(X \rightarrow x_i) & \text{If } X \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

For $k = 1, \ldots, (n-1)$

For $i = 1, \ldots, (n-k)$

$j = i + k$

$\forall X \in N$, calculate

$$\pi(i,j,X) = \max_{\substack{X \rightarrow YZ \in R, \\ s \in (i \ldots (j-1))}} \left(q(X \rightarrow YZ) \times \pi(i,s,Y) \times \pi(s+1,j,Z)\right)$$

$$bp(i,j,X) = \operatorname*{argmax}_{\substack{X \rightarrow YZ \in R, \\ s \in (i \ldots (j-1))}} \left(q(X \rightarrow YZ) \times \pi(i,s,Y) \times \pi(s+1,j,Z)\right)$$

end for

end for

CYK ALGORITHM

Input: sentence $s = x_1 \ldots x_n$ and $PCFG\ G = (N, \Sigma, R, S, q)$ in CNF form

Initialization: $\forall\ 1 \leq i \leq n$ and $X \in N$

$$\pi(i, i, X) = \begin{cases} q(X \rightarrow x_i) & \text{If } X \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

For $k = 1, \ldots, (n-1)$

For $i = 1, \ldots, (n-k)$

$j = i + k$

$\forall X \in N,$ calculate

$$\pi(i, j, X) = \max_{\substack{X \rightarrow YZ \in R, \\ s \in (i \ldots (j-1))}} (q(X \rightarrow YZ) \times \pi(i, s, Y) \times \pi(s+1, j, Z))$$

$$bp(i, j, X) = \operatorname*{argmax}_{\substack{X \rightarrow YZ \in R, \\ s \in (i \ldots (j-1))}} (q(X \rightarrow YZ) \times \pi(i, s, Y) \times \pi(s+1, j, Z))$$

end for

end for

Output: Return $\pi(1, n, S) = \max_{t \in \mathcal{T}_G(s)} p(t)$ and $bp(1, n, S) = \operatorname*{argmax}_{t \in \mathcal{T}_G(s)} p(t)$

CYK ALGORITHM: Example

Initialization: $\forall\ 1 \leq i \leq n$ and $\ X \in N$

$$\pi(i,i,X) = \begin{cases} q(X \rightarrow x_i) & \text{If } X \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

For $k = 1, \ldots, (n-1)$

For $i = 1, \ldots, (n-k)$

$j = i + k$

$\forall X \in N$, calculate

$$\pi(i,j,X) = \max_{\substack{X \rightarrow YZ \in R, \\ s \in (i \ldots (j-1))}} (q(X \rightarrow YZ) \times \pi(i,s,Y) \times \pi(s+1,j,Z))$$

$$bp(i,j,X) = \operatorname*{argmax}_{\substack{X \rightarrow YZ \in R, \\ s \in (i \ldots (j-1))}} (q(X \rightarrow YZ) \times \pi(i,s,Y) \times \pi(s+1,j,Z))$$

end for

end for

Output: Return $\pi(1,n,S) = \max_{t \in \mathcal{T}_G(s)} p(t)$ and $bp(1,n,S) = \operatorname*{argmax}_{t \in \mathcal{T}_G(s)} p(t)$

CYK ALGORITHM: Example

S → NP VP	1.0
VP → Vt NP	0.8
VP → VP PP	0.2
NP → DT NN	0.8
NP → NP PP	0.2
PP → IN NP	1.0

Vi → sleeps	1.0
Vt → saw	1.0
NN → man	0.1
NN → woman	0.1
NN → telescope	0.3
NN → dog	0.5
DT → the	1.0
IN → with	0.6
IN → in	0.4

Initialization: $\forall\ 1 \le i \le n$ and $X \in N$

$$\pi(i,i,X) = \begin{cases} q(X \to x_i) & \text{If } X \to x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

For $k = 1, \ldots, (n-1)$
　For $i = 1, \ldots, (n-k)$
　　$j = i + k$
　　$\forall X \in N$, calculate

$$\pi(i,j,X) = \max_{\substack{X \to YZ \in R,\\ s \in (i \ldots (j-1))}} (q(X \to YZ) \times \pi(i,s,Y) \times \pi(s+1,j,Z))$$

$$bp(i,j,X) = \operatorname*{argmax}_{\substack{X \to YZ \in R,\\ s \in (i \ldots (j-1))}} (q(X \to YZ) \times \pi(i,s,Y) \times \pi(s+1,j,Z))$$

　end for
end for

Output: Return $\pi(1,n,S) = \max_{t \in \mathcal{T}_G(s)} p(t)$ and $bp(1,n,S) = \operatorname*{argmax}_{t \in \mathcal{T}_G(s)} p(t)$

CYK ALGORITHM: Example

S → NP VP	1.0
VP → Vt NP	0.8
VP → VP PP	0.2
NP → DT NN	0.8
NP → NP PP	0.2
PP → IN NP	1.0

Vi → sleeps	1.0
Vt → saw	1.0
NN → man	0.1
NN → woman	0.1
NN → telescope	0.3
NN → dog	0.5
DT → the	1.0
IN → with	0.6
IN → in	0.4

Initialization: $\forall\ 1 \le i \le n$ and $X \in N$

$$\pi(i,i,X) = \begin{cases} q(X \to x_i) & \text{If } X \to x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

For $k = 1, \ldots, (n-1)$
  For $i = 1, \ldots, (n-k)$
    $j = i + k$
    $\forall X \in N$, calculate

$$\pi(i,j,X) = \max_{\substack{X \to Y Z \in R, \\ s \in (i \ldots (j-1))}} \left(q(X \to Y Z) \times \pi(i,s,Y) \times \pi(s+1,j,Z) \right)$$

$$bp(i,j,X) = \operatorname*{argmax}_{\substack{X \to Y Z \in R, \\ s \in (i \ldots (j-1))}} \left(q(X \to Y Z) \times \pi(i,s,Y) \times \pi(s+1,j,Z) \right)$$

  end for
end for

Output: Return $\pi(1,n,S) = \max_{t \in \mathcal{T}_G(s)} p(t)$ and $bp(1,n,S) = \operatorname*{argmax}_{t \in \mathcal{T}_G(s)} p(t)$

$$s = \texttt{the man saw the dog}$$

CYK ALGORITHM: Example

S → NP VP	1.0
VP → Vt NP	0.8
VP → VP PP	0.2
NP → DT NN	0.8
NP → NP PP	0.2
PP → IN NP	1.0

Vi → sleeps	1.0
Vt → saw	1.0
NN → man	0.1
NN → woman	0.1
NN → telescope	0.3
NN → dog	0.5
DT → the	1.0
IN → with	0.6
IN → in	0.4

Initialization: $\forall\ 1 \leq i \leq n$ and $X \in N$

$$\pi(i,i,X) = \begin{cases} q(X \rightarrow x_i) & \text{If } X \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

For $k = 1, \ldots, (n-1)$

For $i = 1, \ldots, (n-k)$

$j = i + k$

$\forall X \in N$, calculate

$$\pi(i,j,X) = \max_{\substack{X \rightarrow Y Z \in R, \\ s \in (i \ldots (j-1))}} \left(q(X \rightarrow Y Z) \times \pi(i,s,Y) \times \pi(s+1,j,Z) \right)$$

$$bp(i,j,X) = \operatorname*{argmax}_{\substack{X \rightarrow Y Z \in R, \\ s \in (i \ldots (j-1))}} \left(q(X \rightarrow Y Z) \times \pi(i,s,Y) \times \pi(s+1,j,Z) \right)$$

end for

end for

Output: Return $\pi(1,n,S) = \max_{t \in \mathcal{T}_G(s)} p(t)$ and $bp(1,n,S) = \operatorname*{argmax}_{t \in \mathcal{T}_G(s)} p(t)$

$$s = \texttt{the man saw the dog}$$

$$t^* = \operatorname*{argmax}_{t \in \mathcal{T}_G(s)} p(t)$$

PROBABILITY OF A SENTENCE

- Given a PCFG what is the probability of a sentence *s* under the PCFG?

PROBABILITY OF A SENTENCE

- Given a PCFG what is the probability of a sentence *s* under the PCFG?

$$\text{Given } PCFG\ G = (N, \Sigma, R, S, q) \text{ in } CNF \text{ form}$$

$$\text{A sentence } s = x_1 \dots x_n$$

PROBABILITY OF A SENTENCE

- Given a PCFG what is the probability of a sentence *s* under the PCFG?

$$\text{Given } PCFG\ G = (N, \Sigma, R, S, q) \text{ in } CNF \text{ form}$$

$$\text{A sentence } s = x_1 \ldots x_n$$

$$p(s) = ?$$

PROBABILITY OF A SENTENCE

- Given a PCFG what is the probability of a sentence *s* under the PCFG?

$$\text{Given } PCFG\ G = (N, \Sigma, R, S, q) \text{ in } CNF \text{ form}$$

$$\text{A sentence } s = x_1 \ldots x_n$$

$$p(s) = \sum_{t \in \mathcal{T}_G(s)} p(t)$$

INSIDE ALGORITHM

Input: sentence $s = x_1 \ldots x_n$ and $PCFG$ $G = (N, \Sigma, R, S, q)$ in CNF form

Initialization: $\forall\ 1 \leq i \leq n$ and $X \in N$

$$\pi(i, i, X) = \begin{cases} q(X \rightarrow x_i) & \text{If } X \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

For $k = 1, \ldots, (n-1)$

For $i = 1, \ldots, (n-k)$

$j = i + k$

$\forall X \in N$, calculate

$$\pi(i, j, X) = \sum_{\substack{X \rightarrow YZ \in R, \\ s \in (i \ldots (j-1))}} \left(q(X \rightarrow YZ) \times \pi(i, s, Y) \times \pi(s+1, j, Z)\right)$$

end for

end for

Output: Return $\pi(1, n, S) = \sum_{t \in \mathcal{T}(s)} p(t)$

Summary

- PCFG: $G = (N, \Sigma, R, S, q)$
- Linguists have created large corpus of sentences and corresponding trees (Treebank)
- Parameters of a PCFG can be learned using MLE on Treebank
- Most probable tree can be found using CKY algorithm
- Probability of a sentence under the PCFG can be found using Inside Algorithm

References

1. Michael Collin’s NLP Lecture Notes: http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/pcfgs.pdf

2. Chapter 12, Speech and Language Processing, Dan Jurafsky and James Martin