Special Topics in Natural Language Processing

CS698O

Ashutosh Modi
CSE Department, IIT Kanpur

Lecture 15: Parsing 1
Feb 7, 2020

Context Free Grammar (CFG)

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

Context Free Grammar (CFG)

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

NP: Harry the Horse

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)***

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

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

$$NP \rightarrow Det \;\; Nominal$$

$$NP \rightarrow ProperNoun$$

$$Nominal \rightarrow Noun \mid Nominal Noun$$

$$Det \rightarrow a$$

$$Det \rightarrow the$$

$$Noun \rightarrow flight$$

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.)

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.)

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

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

Why CFG?

- CFG can be used for generating sentences
- CFG can be used for assigning structure to a sentence

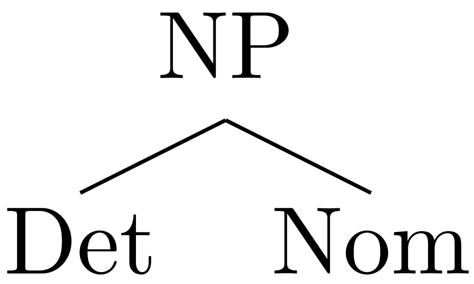
GENERATION USING CFG

GENERATION USING CFG

NP

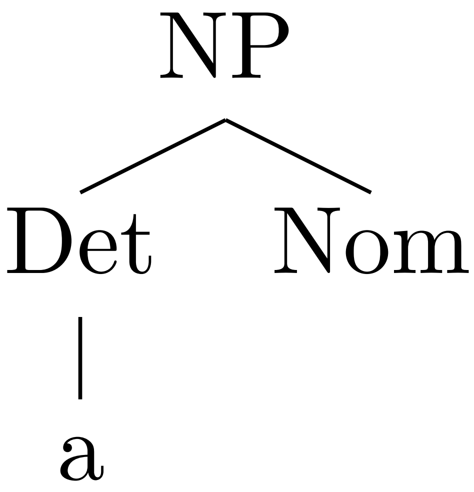
1. Start with a NP

GENERATION USING CFG



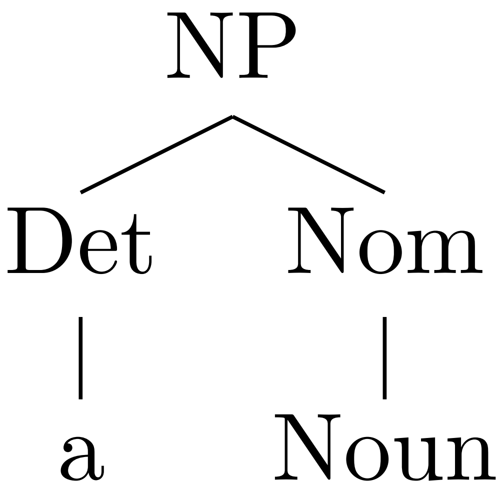
1. Start with a NP
2. Use rule 1 to rewrite NP as NP → Det Nominal

GENERATION USING CFG



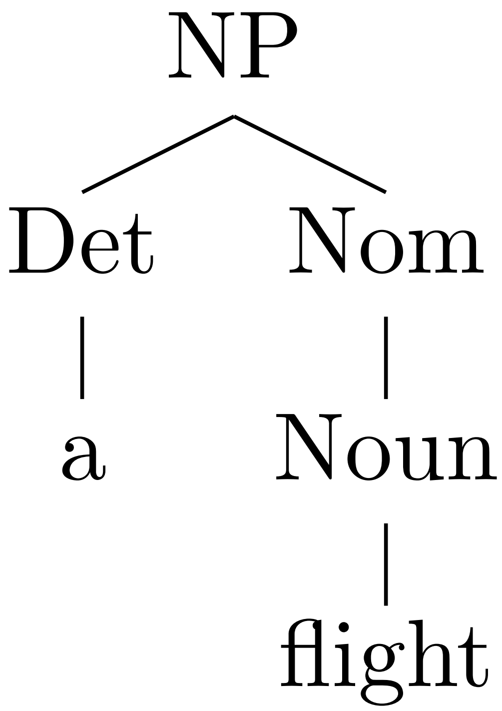
1. Start with a NP
2. Use rule 1 to rewrite NP as NP → Det Nominal
3. Use rule 4: Det → *a*

GENERATION USING CFG



1. Start with a NP
2. Use rule 1 to rewrite NP as NP → Det Nominal
3. Use rule 4: Det → *a*
4. Use rule 3: Nom → Noun

GENERATION USING CFG



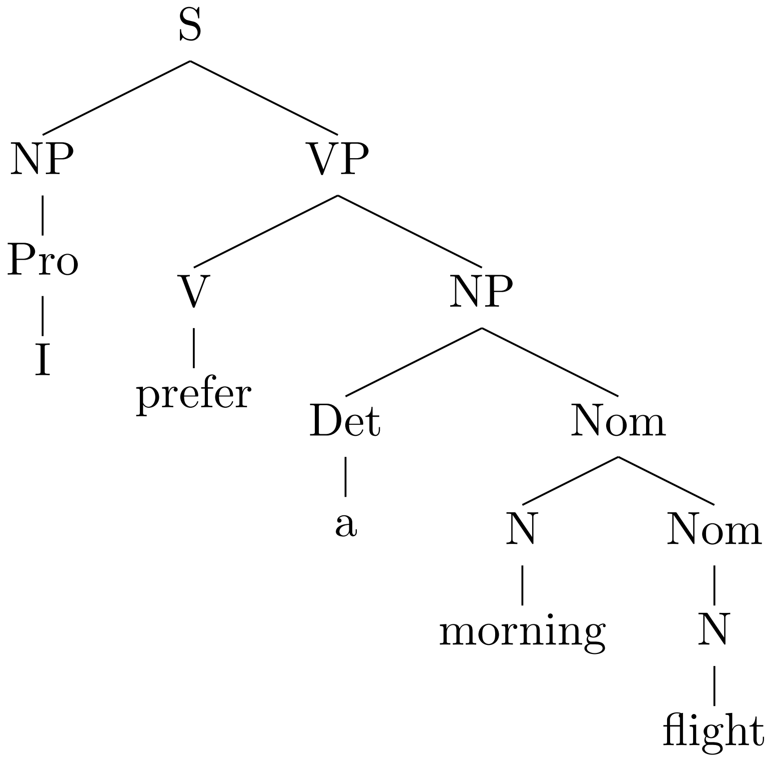
1. Start with a NP
2. Use rule 1 to rewrite NP as NP → Det Nominal
3. Use rule 4: Det → *a*
4. Use rule 3: Nom → Noun
5. Use rule 6: Noun → *flight*

CFG: BRACKETTED NOTATION

$[_{S}\ [_{NP}\ [_{Pro}\ \text{I}]\ ]\ [_{VP}\ [_{V}\ \text{prefer}]\ [_{NP}\ [_{Det}\ \text{a}]\ [_{Nom}\ [_{N}\ \text{morning}]\ [_{Nom}\ [_{N}\ \text{flight}]]]]]]$

CFG: BRACKETTED NOTATION

$[_{S}\ [_{NP}\ [_{Pro}\ \text{I}]\ ]\ [_{VP}\ [_{V}\ \text{prefer}]\ [_{NP}\ [_{Det}\ \text{a}]\ [_{Nom}\ [_{N}\ \text{morning}]\ [_{Nom}\ [_{N}\ \text{flight}]]]]]]$



CFG: DEFINITION

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

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.

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.

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 $Y_i \in (N \cup \Sigma)\ for\ i = 1, 2, \ldots, n$

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 $X \to Y_1, \ldots, Y_n$,

  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

$S = \texttt{S}$

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

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

$R =$

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

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

LEFT MOST DERIVATION

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

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

$$s_1 = \mathrm{S}$$

LEFT MOST DERIVATION

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

$$s_1 = \mathrm{S}$$
$$s_2 = \mathrm{NP}\ \ \mathrm{VP}$$

LEFT MOST DERIVATION

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

$$s_1 = \mathrm{S}$$
$$s_2 = \mathrm{NP}\ \ \mathrm{VP}$$
$$s_3 = \mathrm{DT}\ \ \mathrm{NN}\ \ \mathrm{VP}$$

LEFT MOST DERIVATION

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

$s_1 = \text{S}$

$s_2 = \text{NP VP}$

$s_3 = \text{DT NN VP}$

$s_4 = \text{the NN VP}$

LEFT MOST DERIVATION

S $\rightarrow$ NP VP
VP $\rightarrow$ Vi VP $\rightarrow$ Vt NP VP $\rightarrow$ VP PP
NP $\rightarrow$ DT NN NP $\rightarrow$ NP PP
PP $\rightarrow$ IN NP

Vi $\rightarrow$ sleeps Vt $\rightarrow$ saw
NN $\rightarrow$ man NN $\rightarrow$ woman NN $\rightarrow$ telescope NN $\rightarrow$ dog
DT $\rightarrow$ the
IN $\rightarrow$ with IN $\rightarrow$ in

$$s_1 = \text{S}$$
$$s_2 = \text{NP VP}$$
$$s_3 = \text{DT NN VP}$$
$$s_4 = \text{the NN VP}$$
$$s_5 = \text{the man VP}$$

LEFT MOST DERIVATION

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

$$s_1 = \texttt{S}$$
$$s_2 = \texttt{NP VP}$$
$$s_3 = \texttt{DT NN VP}$$
$$s_4 = \texttt{the NN VP}$$
$$s_5 = \texttt{the man VP}$$
$$s_6 = \texttt{the man Vi}$$

LEFT MOST DERIVATION

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

$s_1 = \text{S}$

$s_2 = \text{NP VP}$

$s_3 = \text{DT NN VP}$

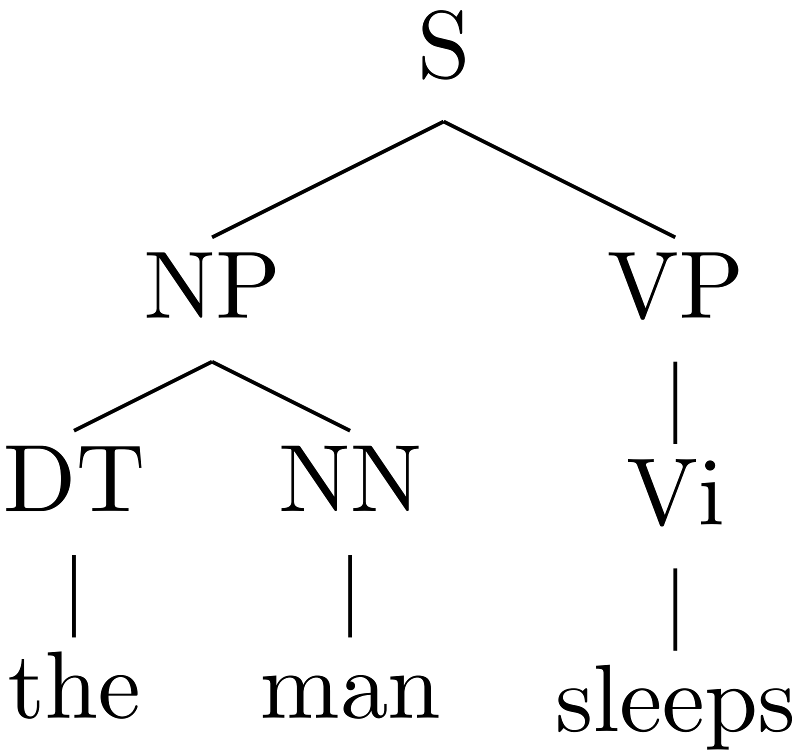
$s_4 = \text{the NN VP}$

$s_5 = \text{the man VP}$

$s_6 = \text{the man Vi}$

$s_7 = \text{the man sleeps}$

LEFT MOST DERIVATION



$$s_1 = \texttt{S}$$
$$s_2 = \texttt{NP VP}$$
$$s_3 = \texttt{DT NN VP}$$
$$s_4 = \texttt{the NN VP}$$
$$s_5 = \texttt{the man VP}$$
$$s_6 = \texttt{the man Vi}$$
$$s_7 = \texttt{the man sleeps}$$

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

AMBIGUITY

A string *s* is *ambiguous* under the CFG if there is more than one derivation with *s* as the yield

AMBIGUITY

A string *s* is *ambiguous* under the CFG if there is more than one derivation with *s* as the yield

PP Attachment Ambiguity

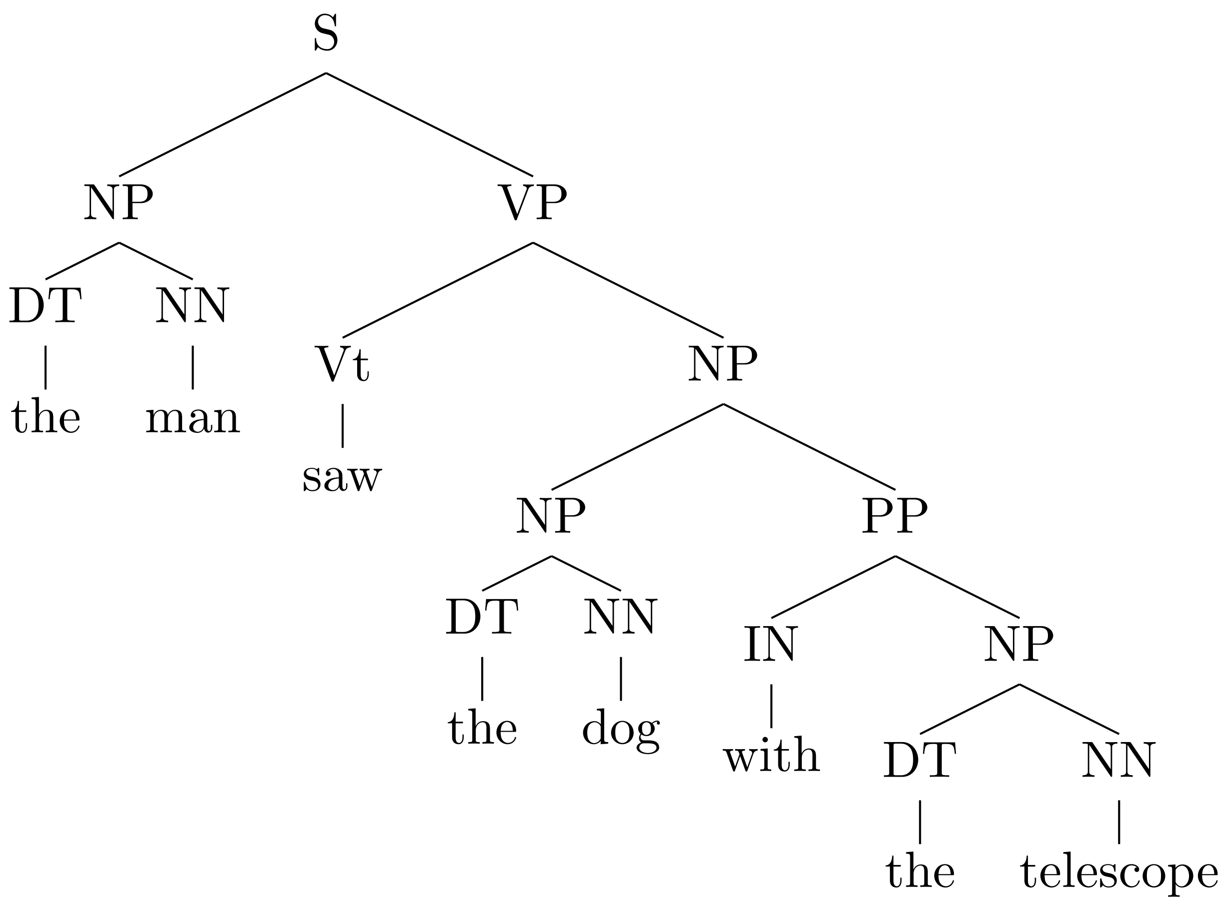
AMBIGUITY

A string *s* is *ambiguous* under the CFG if there is more than one derivation with *s* as the yield

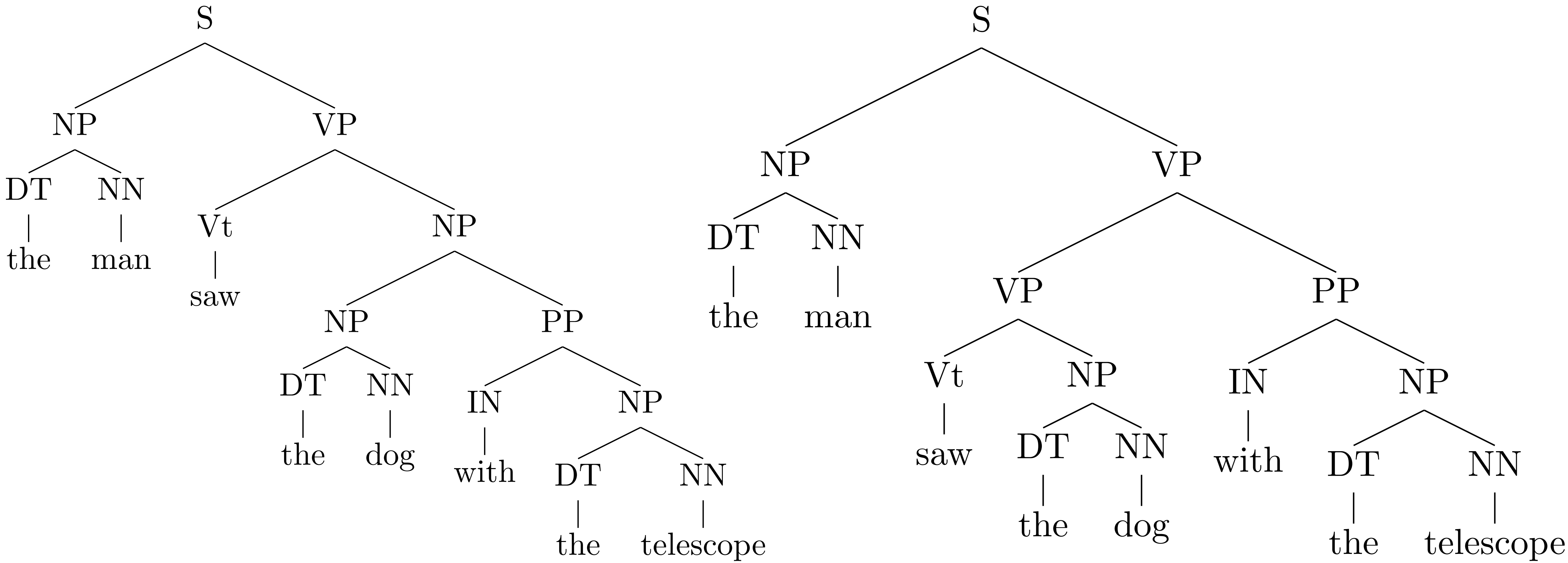
PP Attachment Ambiguity

the man saw the dog with the telescope

AMBIGUITY


S
NP
VP
DT
NN
the
man
Vt
saw
NP
NP
PP
DT
NN
the
dog
IN
with
NP
DT
NN
the
telescope

AMBIGUITY


S
NP VP
DT NN Vt NP
the man saw
NP PP
DT NN IN NP
the dog with
DT NN
the telescope
S
NP VP
DT NN VP PP
the man
Vt NP IN NP
saw with
DT NN DT NN
the dog the telescope

AMBIGUITY

How many parse trees for the sentence?

She announced a program to promote safety in trucks and vans

DEFINITIONS

$\mathcal{T}_G =$ is set of all possible left-most derivations (parse trees) under the grammar G

DEFINITIONS

$$\mathcal{T}_G = \text{is set of all possible left-most derivations (parse trees)}$$
$$\text{under the grammar } G$$

For any $t \in \mathcal{T}_G$, $\textbf{yield}(t) = s \in \Sigma^*$ denotes a string s that is yield of t

DEFINITIONS

$$\mathcal{T}_G = \text{is set of all possible left-most derivations (parse trees)}$$
$$\text{under the grammar } G$$

For any $t \in \mathcal{T}_G$, $\textbf{yield}(t) = s \in \Sigma^*$ denotes a string s that is yield of t

$$\text{Set of possible parse trees } = \{t : t \in \mathcal{T}_G, \textbf{yield}(t) = s \in \Sigma^*\}$$

DEFINITIONS

$\mathcal{T}_G =$ is set of all possible left-most derivations (parse trees) under the grammar G

For any $t \in \mathcal{T}_G$, $\textbf{yield}(t) = s \in \Sigma^*$ denotes a string s that is yield of t

Set of possible parse trees $= \{t : t \in \mathcal{T}_G, \textbf{yield}(t) = s \in \Sigma^*\}$

A sentence is ambiguous if $|\mathcal{T}_G| > 1$

DEFINITIONS

$\mathcal{T}_G =$ is set of all possible left-most derivations (parse trees) under the grammar G

For any $t \in \mathcal{T}_G$, $\textbf{yield}(t) = s \in \Sigma^*$ denotes a string s that is yield of t

Set of possible parse trees $= \{t : t \in \mathcal{T}_G, \textbf{yield}(t) = s \in \Sigma^*\}$

A sentence is ambiguous if $|\mathcal{T}_G| > 1$

A sentence is grammatical if $|\mathcal{T}_G| > 0$

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 $\alpha \to \beta \in R$

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

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 $q(\alpha \rightarrow \beta)$ 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 $q(\alpha \rightarrow \beta)$ for each rule $\alpha \rightarrow \beta \in R$

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

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

PCFG EXAMPLE

$S = \mathtt{S}$

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

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

S → NP VP	1.0
VP → Vi	0.3
VP → Vt NP	0.5
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 EXAMPLE

S → NP VP	1.0
VP → Vi	0.3
VP → Vt NP	0.5
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

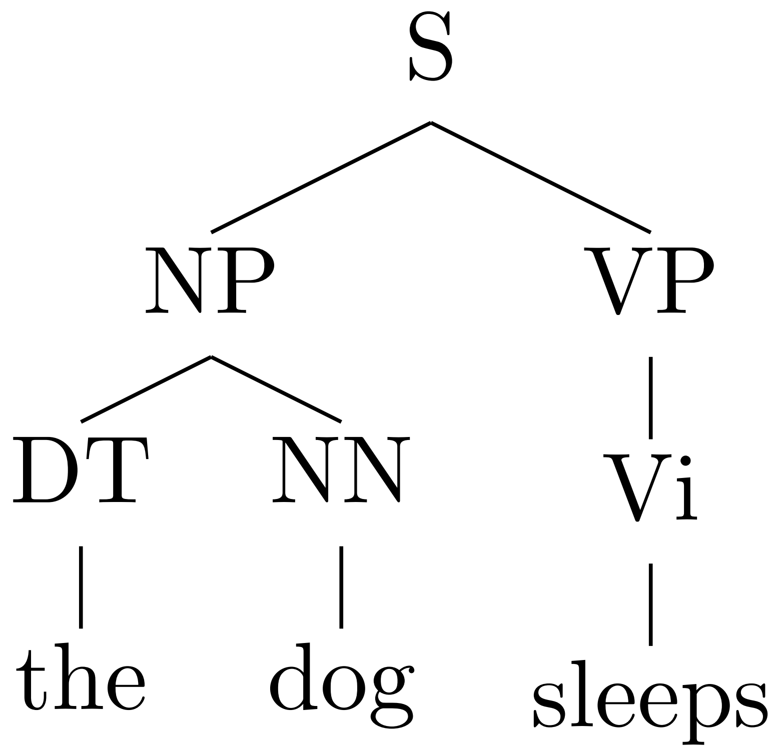
$$\sum_{\substack{\alpha \to \beta \in R: \\ \alpha = VP}} q(\alpha \to \beta) = ?$$

PCFG EXAMPLE

S → NP VP	1.0
VP → Vi	0.3
VP → Vt NP	0.5
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

$t =$

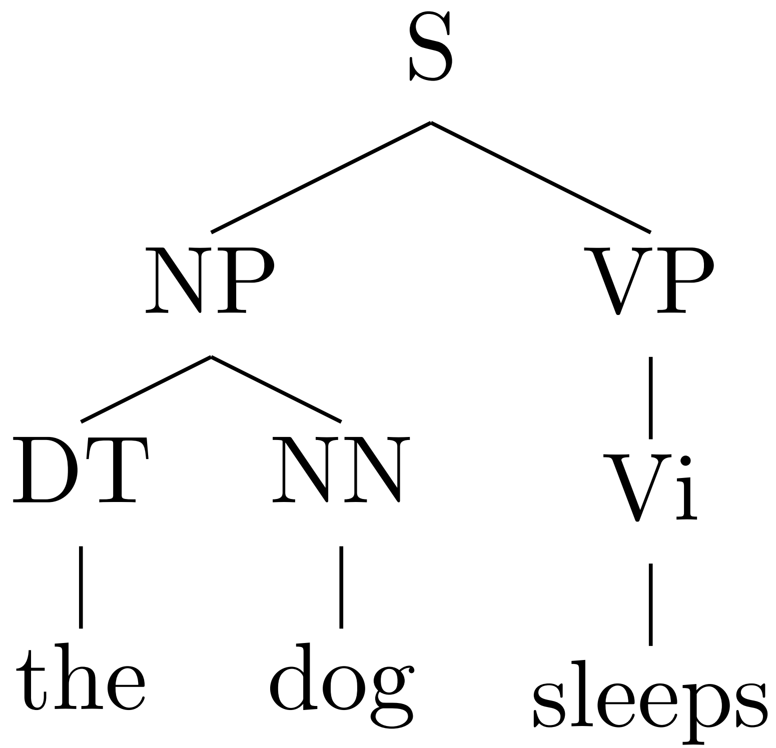


PCFG EXAMPLE

S → NP VP	1.0
VP → Vi	0.3
VP → Vt NP	0.5
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

$t =$



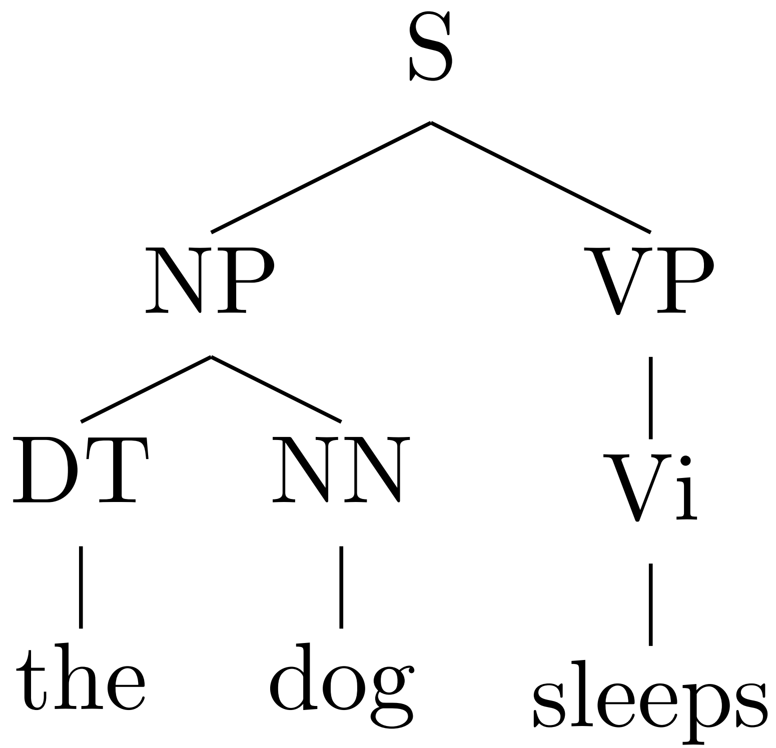
$$p(t) = ?$$

PCFG EXAMPLE

S → NP VP	1.0
VP → Vi	0.3
VP → Vt NP	0.5
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

$t =$



$$p(t) = q(\texttt{S} \to \texttt{NP VP}) \times q(\texttt{NP} \to \texttt{DT NN}) \times$$
$$q(\texttt{DT} \to \texttt{the}) \times q(\texttt{NN} \to \texttt{dog}) \times$$
$$q(\texttt{VP} \to \texttt{Vi}) \times q(\texttt{Vi} \to \texttt{sleeps})$$

GENERATION USING PCFG

GENERATION USING PCFG

Set $i = 1$ and define $s_i = \texttt{S}$,

GENERATION USING PCFG

Set $i = 1$ and define $s_i = \mathtt{S}$,

While s_i contains at least one non-terminal :

GENERATION USING PCFG

Set $i = 1$ and define $s_i = \mathtt{S}$,

While s_i contains at least one non-terminal :

$$\text{Let, } s_i = X\ Y \text{ where, } Y \in (N \cup \Sigma)$$

GENERATION USING PCFG

Set $i = 1$ and define $s_i = \mathtt{S}$,

While s_i contains at least one non-terminal :

$$\text{Let, } s_i = X\ Y \text{ where, } Y \in (N \cup \Sigma)$$

$$(X \to \beta) \sim q(X \to \beta)$$

GENERATION USING PCFG

Set $i = 1$ and define $s_i = \mathtt{S}$,

While s_i contains at least one non-terminal :

$$\text{Let, } s_i = X\ Y \text{ where, } Y \in (N \cup \Sigma)$$

$$(X \to \beta) \sim q(X \to \beta)$$

$$\text{Set } s_{i+1} = \beta\ Y$$

GENERATION USING PCFG

Set $i = 1$ and define $s_i = \texttt{S}$,

While s_i contains at least one non-terminal :

$$\text{Let, } s_i = X\ Y \text{ where, } Y \in (N \cup \Sigma)$$

$$(X \to \beta) \sim q(X \to \beta)$$

$$\text{Set } s_{i+1} = \beta\ Y$$

$$i = i + 1$$

PCFG

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

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

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?

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

Summary

1. Languages can be modeled using grammar
2. CFG models the language in terms of constituents making use of rules and lexicon
3. A CFG can be modeled using $G = (N,\Sigma,R,S)$
4. Left most derivation using G *yields* strings in the language
5. CFG can help in resolving ambiguity
6. Probabilistic version of CFG (PCFG) helps in computationally and automatically parsing a language

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