Introduction to Parsing

Overview

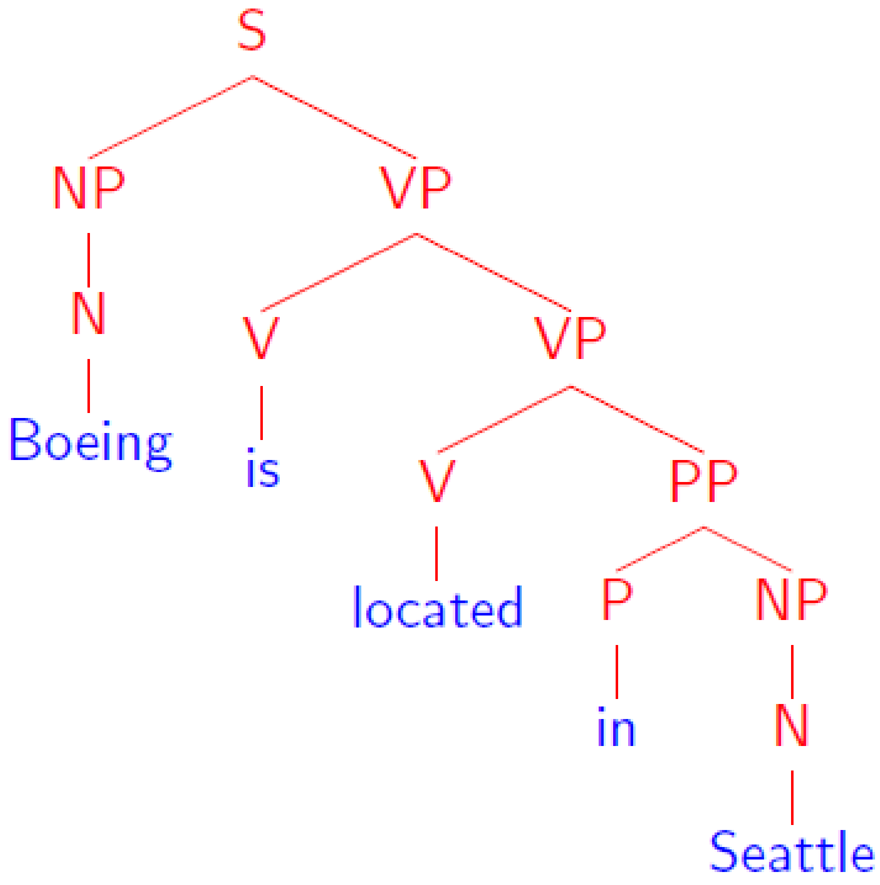
- An introduction to the parsing problem
- Context free grammars
- A brief(!) sketch of the syntax of English
- Examples of ambiguous structures

Parsing (Syntactic Structure)

INPUT:

Boeing is located in Seattle.

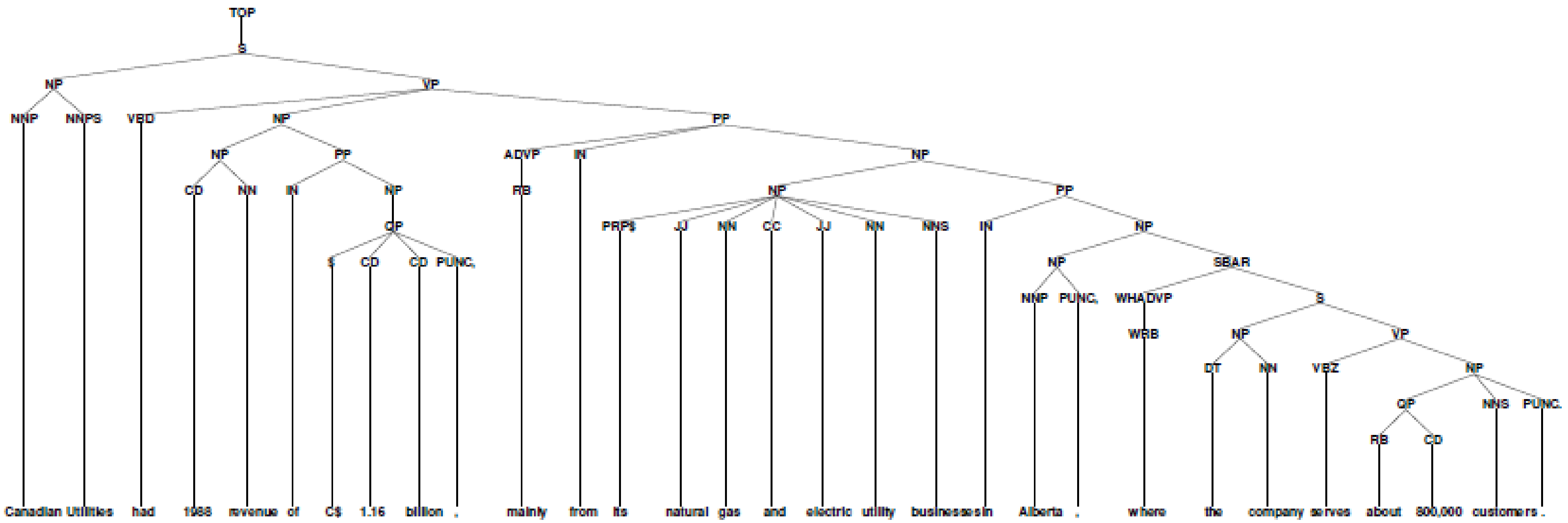
OUTPUT:



Data for Parsing Experiments

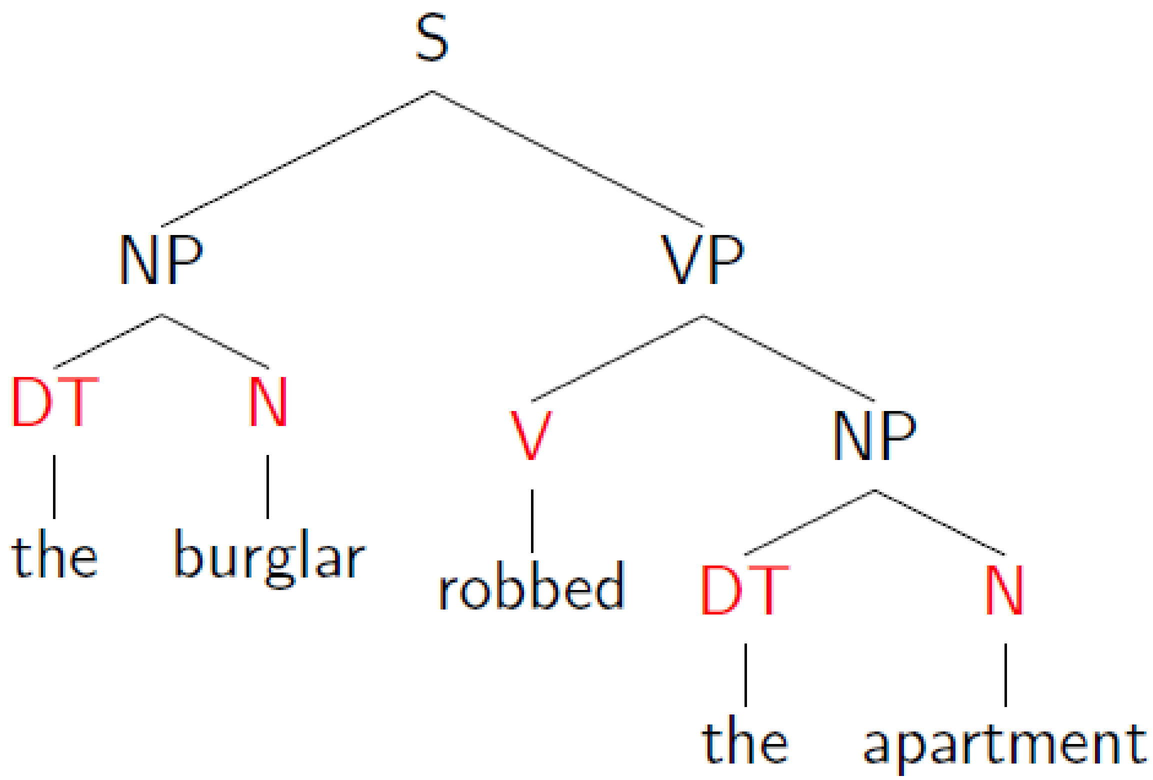
- Penn WSJ Treebank = 50,000 sentences with associated trees
- Usual set-up: 40,000 training sentences, 2400 test sentences

An example tree:



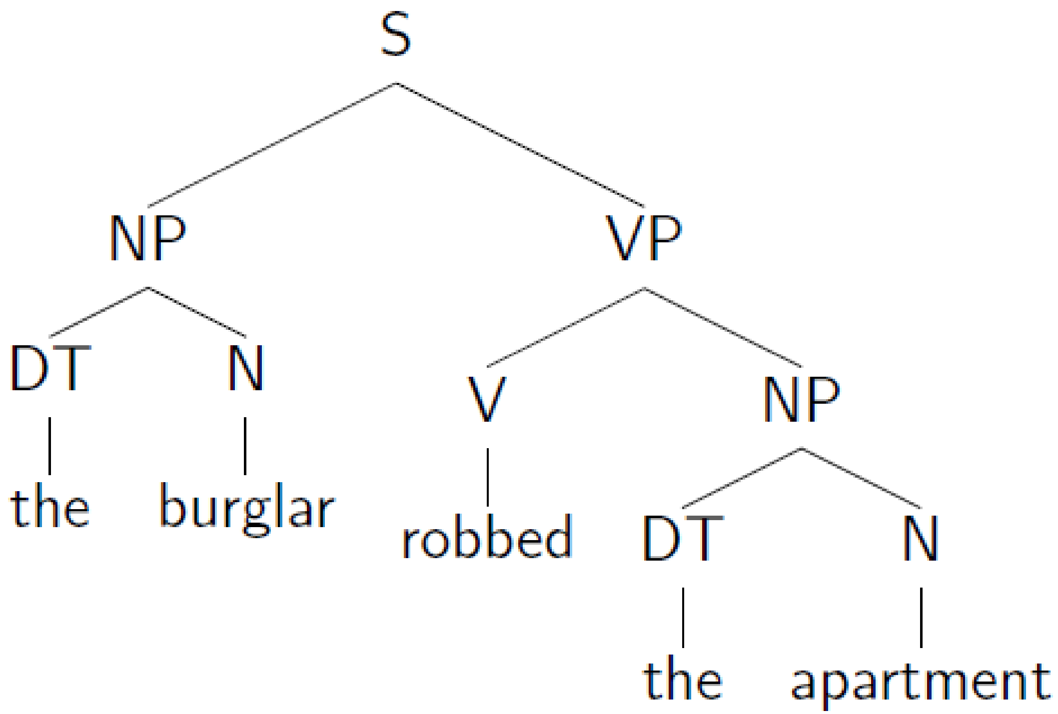
The Information Conveyed by Parse Trees

(1) Part of speech for each word
(N = noun, V = verb, DT = determiner)



The Information Conveyed by Parse Trees (continued)

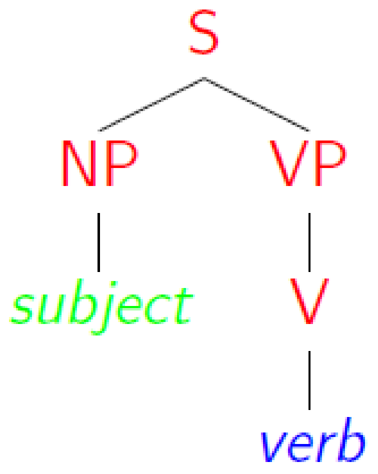
(2) Phrases

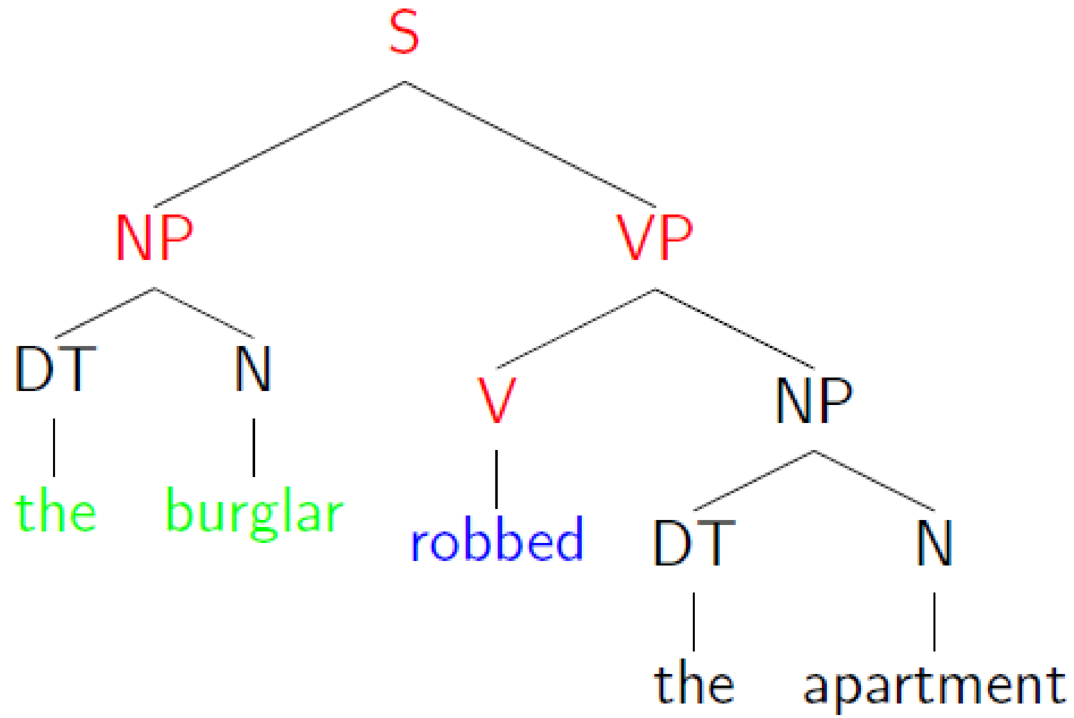


Noun Phrases (NP): “the burglar”, “the apartment”
Verb Phrases (VP): “robbed the apartment”
Sentences (S): “the burglar robbed the apartment”

The Information Conveyed by Parse Trees (continued)

(3) Useful Relationships





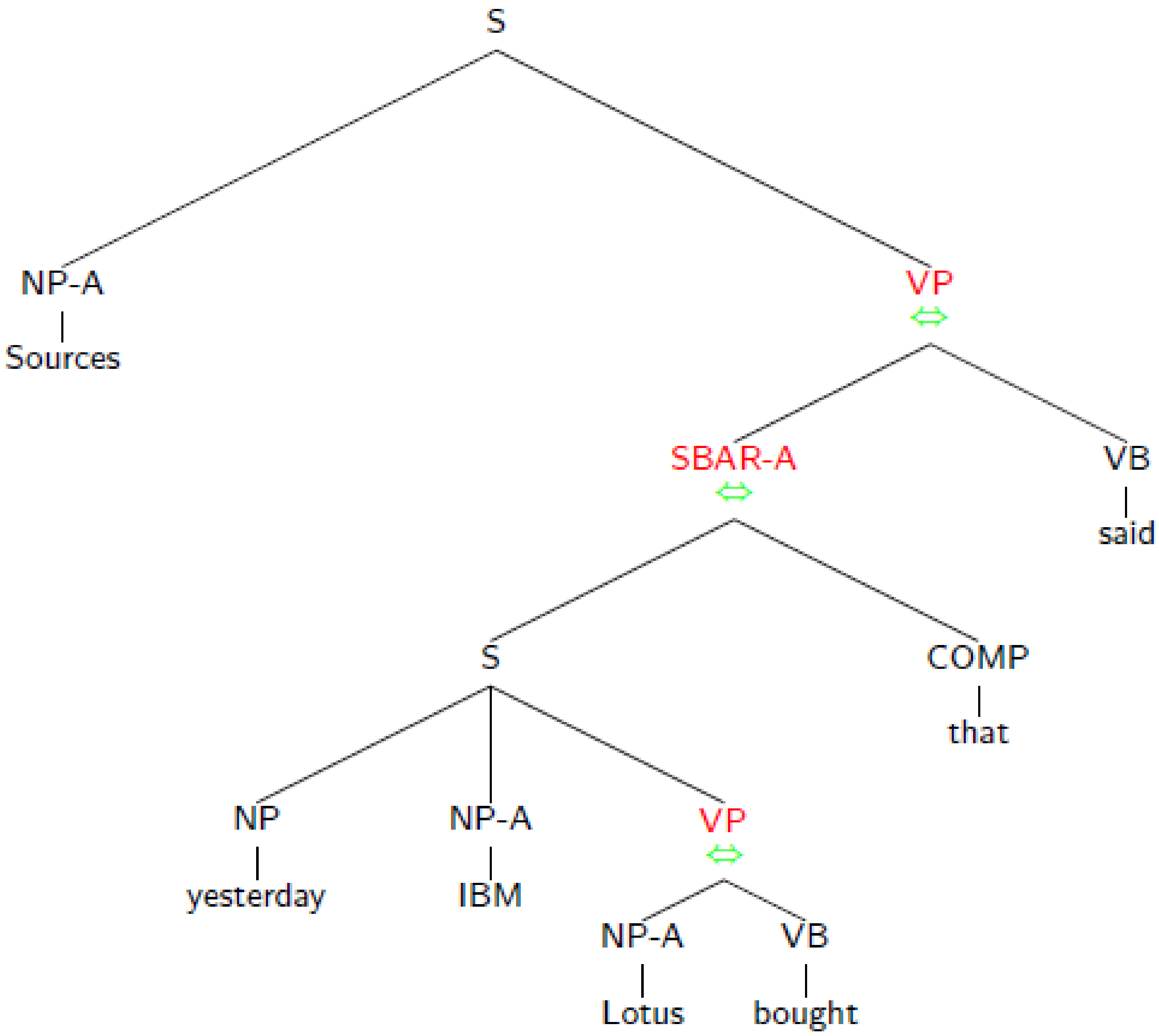
⇒ "the burglar" is the subject of "robbed"

An Example Application: Machine Translation

- English word order is *subject – verb – object*
- Japanese word order is *subject – object – verb*

English:	IBM bought Lotus
Japanese:	*IBM Lotus bought*

English:	Sources said that IBM bought Lotus yesterday
Japanese:	*Sources yesterday IBM Lotus bought that said*


S
NP-A
Sources
VP
SBAR-A
VB
said
S
COMP
that
NP
yesterday
NP-A
IBM
VP
NP-A
Lotus
VB
bought

Context-Free Grammars

Hopcroft and Ullman, 1979

A context free grammar $G = (N, \Sigma, R, S)$ where:

- N is a set of non-terminal symbols
- Σ is a set of terminal symbols
- R is a set of rules of the form $X \rightarrow Y_1 Y_2 \ldots Y_n$ for $n \geq 0$, $X \in N$, $Y_i \in (N \cup \Sigma)$
- $S \in N$ is a distinguished start symbol

A Context-Free Grammar for English

$N = \{\text{S, NP, VP, PP, DT, Vi, Vt, NN, IN}\}$
$S = \text{S}$
$\Sigma = \{\text{sleeps, saw, man, woman, telescope, the, with, in}\}$

$R =$

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
DT	$\rightarrow$	the
IN	$\rightarrow$	with
IN	$\rightarrow$	in

Note: S=sentence, VP=verb phrase, NP=noun phrase, PP=prepositional phrase, DT=determiner, Vi=intransitive verb, Vt=transitive verb, NN=noun, IN=preposition

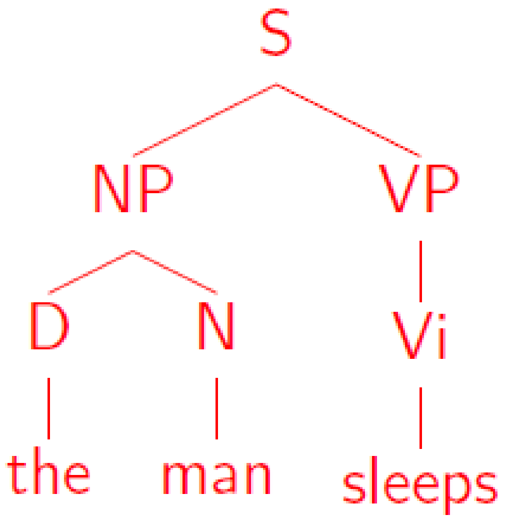
Left-Most Derivations

A left-most derivation is a sequence of strings $s_1 \dots s_n$, where

- $s_1 = S$, the start symbol
- $s_n \in \Sigma^*$, i.e. s_n is made up of terminal symbols only
- Each s_i for $i = 2 \dots n$ 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

For example: [S], [NP VP], [D N VP], [the N VP], [the man VP], [the man Vi], [the man sleeps]

Representation of a derivation as a tree:



An Example

DERIVATION

S

NP VP

DT N VP

the N VP

the dog VP

the dog VB

the dog laughs

RULES USED

S $\rightarrow$ NP VP

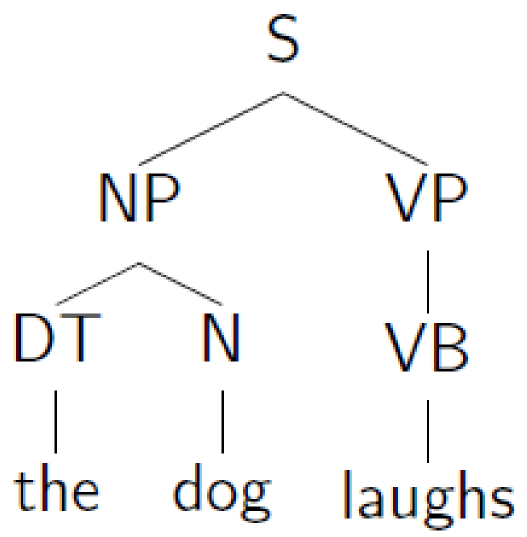
NP $\rightarrow$ DT N

DT $\rightarrow$ the

N $\rightarrow$ dog

VP $\rightarrow$ VB

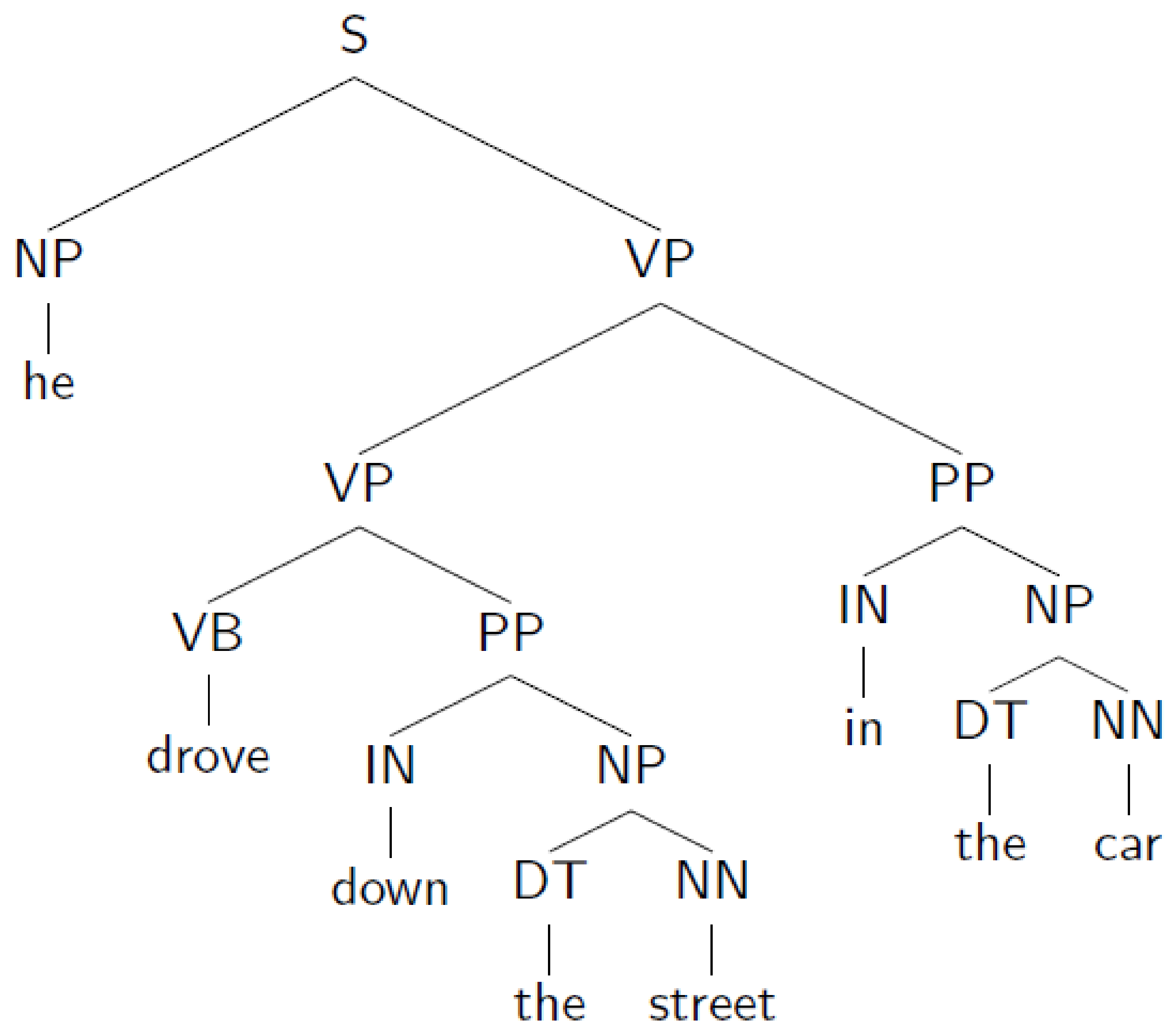
VB $\rightarrow$ laughs



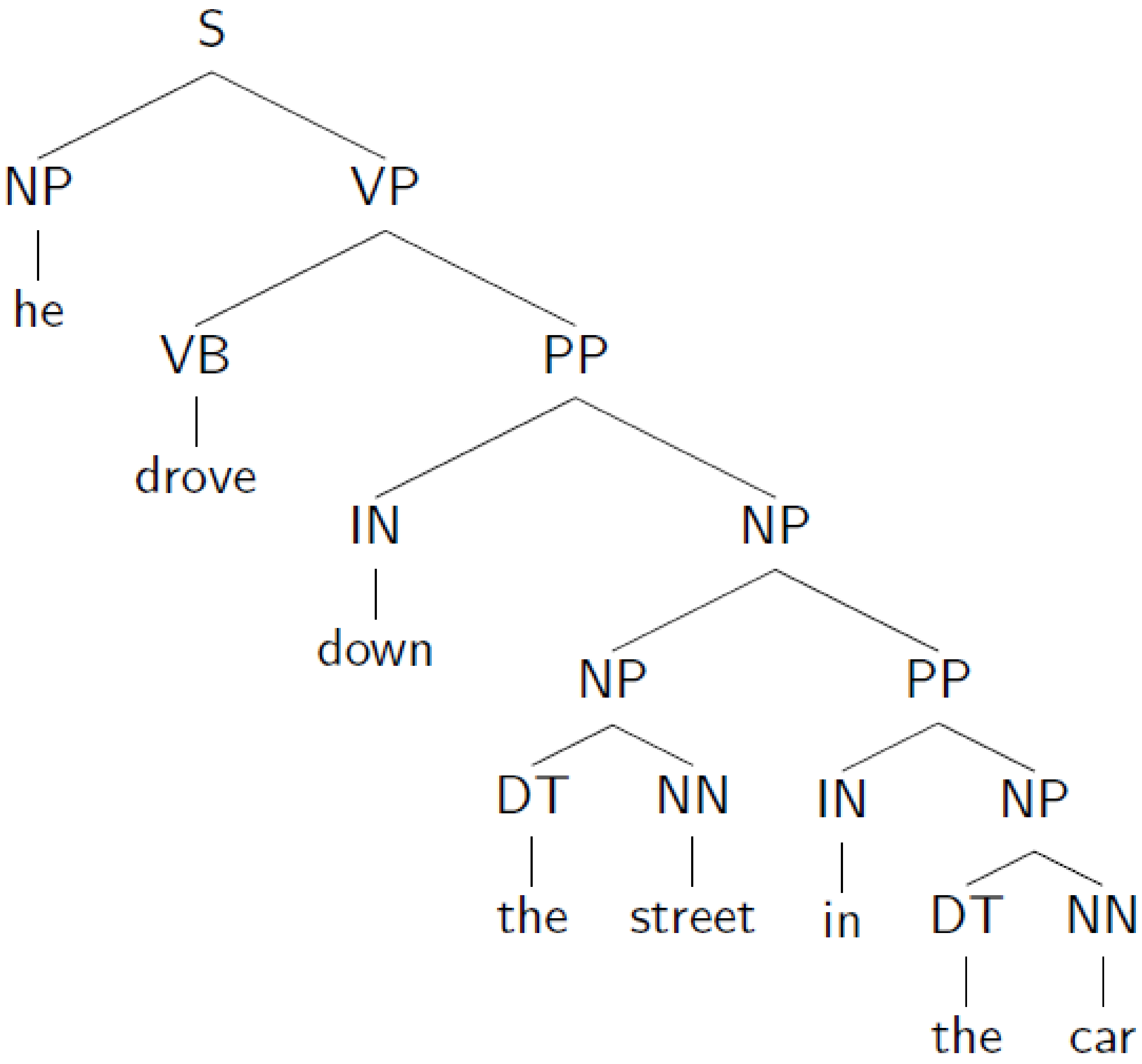
Properties of CFGs

- A CFG defines a set of possible derivations
- A string $s \in \Sigma^*$ is in the *language* defined by the CFG if there is at least one derivation that yields s
- Each string in the language generated by the CFG may have more than one derivation ("ambiguity")

An Example of Ambiguity



An Example of Ambiguity (continued)



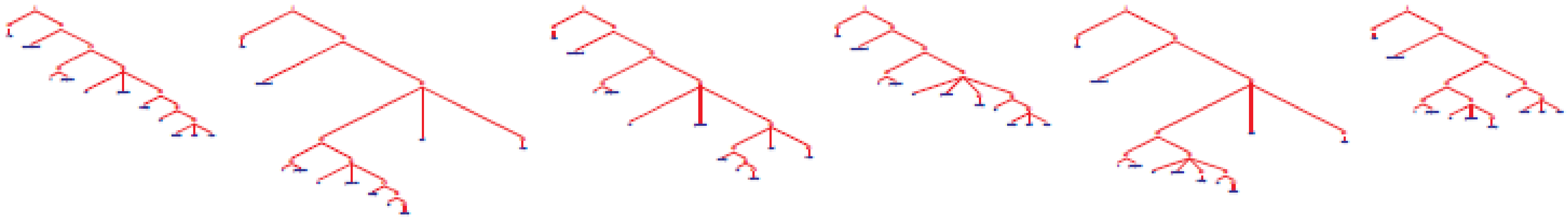
The Problem with Parsing: Ambiguity

INPUT:

She announced a program to promote safety in trucks and vans

$\Downarrow$

POSSIBLE OUTPUTS:



And there are more...

A Brief Overview of English Syntax

Parts of Speech (tags from the Brown corpus):

- Nouns
  NN = singular noun e.g., man, dog, park
  NNS = plural noun e.g., telescopes, houses, buildings
  NNP = proper noun e.g., Smith, Gates, IBM
- Determiners
  DT = determiner e.g., the, a, some, every
- Adjectives
  JJ = adjective e.g., red, green, large, idealistic

A Fragment of a Noun Phrase Grammar

$\bar{N}$	$\Rightarrow$	NN	
$\bar{N}$	$\Rightarrow$	NN	$\bar{N}$
$\bar{N}$	$\Rightarrow$	JJ	$\bar{N}$
$\bar{N}$	$\Rightarrow$	$\bar{N}$	$\bar{N}$
NP	$\Rightarrow$	DT	$\bar{N}$

NN	$\Rightarrow$	box
NN	$\Rightarrow$	car
NN	$\Rightarrow$	mechanic
NN	$\Rightarrow$	pigeon
DT	$\Rightarrow$	the
DT	$\Rightarrow$	a

JJ	$\Rightarrow$	fast
JJ	$\Rightarrow$	metal
JJ	$\Rightarrow$	idealistic
JJ	$\Rightarrow$	clay

Prepositions, and Prepositional Phrases

- Prepositions
  IN = preposition   e.g., of, in, out, beside, as

An Extended Grammar

$\bar{N}$	$\Rightarrow$	NN	
$\bar{N}$	$\Rightarrow$	NN	$\bar{N}$
$\bar{N}$	$\Rightarrow$	JJ	$\bar{N}$
$\bar{N}$	$\Rightarrow$	$\bar{N}$	$\bar{N}$
NP	$\Rightarrow$	DT	$\bar{N}$
PP	$\Rightarrow$	IN	NP
$\bar{N}$	$\Rightarrow$	$\bar{N}$	PP

NN	$\Rightarrow$	box
NN	$\Rightarrow$	car
NN	$\Rightarrow$	mechanic
NN	$\Rightarrow$	pigeon
DT	$\Rightarrow$	the
DT	$\Rightarrow$	a

JJ	$\Rightarrow$	fast
JJ	$\Rightarrow$	metal
JJ	$\Rightarrow$	idealistic
JJ	$\Rightarrow$	clay
IN	$\Rightarrow$	in
IN	$\Rightarrow$	under
IN	$\Rightarrow$	of
IN	$\Rightarrow$	on
IN	$\Rightarrow$	with
IN	$\Rightarrow$	as

Generates:
in a box, under the box, the fast car mechanic under the pigeon in the box, . . .

Verbs, Verb Phrases, and Sentences

- Basic Verb Types

Vi = Intransitive verb	e.g., sleeps, walks, laughs
Vt = Transitive verb	e.g., sees, saw, likes
Vd = Ditransitive verb	e.g., gave

- Basic VP Rules

$$\mathrm{VP} \rightarrow \mathrm{Vi}$$
$$\mathrm{VP} \rightarrow \mathrm{Vt} \quad \mathrm{NP}$$
$$\mathrm{VP} \rightarrow \mathrm{Vd} \quad \mathrm{NP} \quad \mathrm{NP}$$

- Basic S Rule

$$\mathrm{S} \rightarrow \mathrm{NP} \quad \mathrm{VP}$$

Examples of VP:

sleeps, walks, likes the mechanic, gave the mechanic the fast car

Examples of S:

the man sleeps, the dog walks, the dog gave the mechanic the fast car

PPs Modifying Verb Phrases

A new rule: $\text{VP} \rightarrow \text{VP} \quad \text{PP}$

New examples of VP:
sleeps in the car, walks like the mechanic, gave the mechanic the fast car on Tuesday, ...

Complementizers, and SBARs

- Complementizers
  COMP = complementizer   e.g., that
- SBAR
  SBAR $\rightarrow$ COMP S

Examples:

that the man sleeps, that the mechanic saw the dog ...

More Verbs

- New Verb Types

V[5]	e.g., said, reported
V[6]	e.g., told, informed
V[7]	e.g., bet

- New VP Rules

VP	$\rightarrow$	V[5]	SBAR		
VP	$\rightarrow$	V[6]	NP	SBAR	
VP	$\rightarrow$	V[7]	NP	NP	SBAR

Examples of New VPs:

said that the man sleeps

told the dog that the mechanic likes the pigeon

bet the pigeon $50 that the mechanic owns a fast car

Coordination

- A New Part-of-Speech:
  CC = Coordinator   e.g., and, or, but
- New Rules

NP	$\rightarrow$	NP	CC	NP
$\bar{\text{N}}$	$\rightarrow$	$\bar{\text{N}}$	CC	$\bar{\text{N}}$
VP	$\rightarrow$	VP	CC	VP
S	$\rightarrow$	S	CC	S
SBAR	$\rightarrow$	SBAR	CC	SBAR

We've Only Scratched the Surface...

- Agreement

  The dogs laugh *vs.* The dog laughs

- Wh-movement

  The dog that the cat liked ___

- Active vs. passive

  The dog saw the cat *vs.*
  The cat was seen by the dog

- If you're interested in reading more:

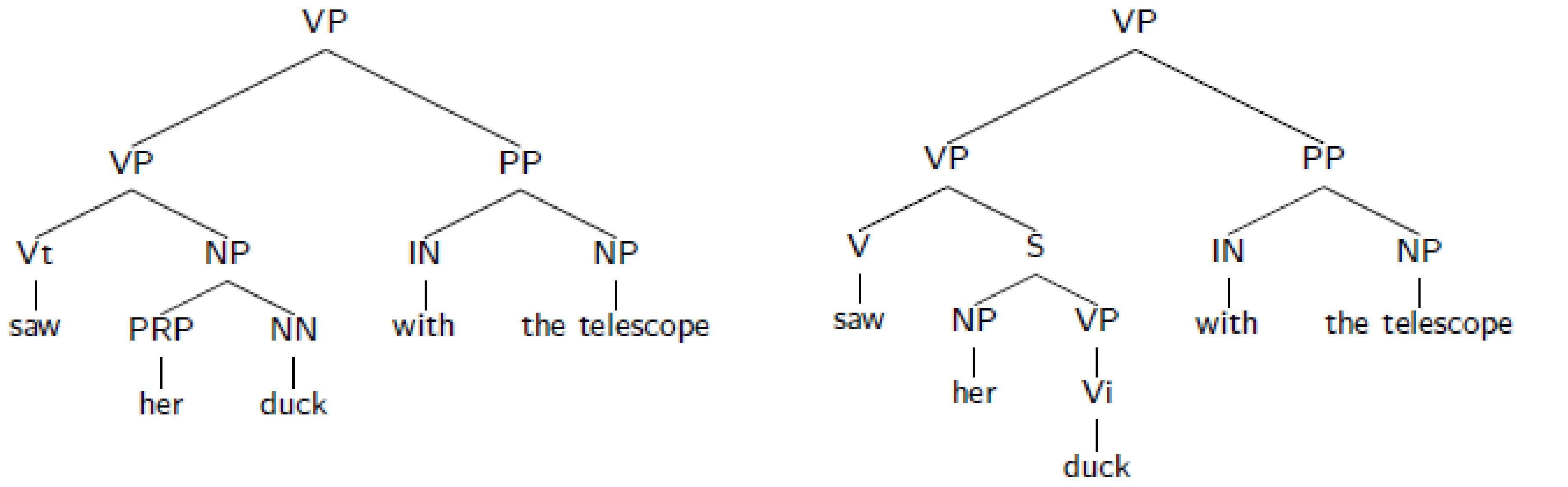
  Syntactic Theory: A Formal Introduction, 2nd Edition. Ivan A. Sag, Thomas Wasow, and Emily M. Bender.

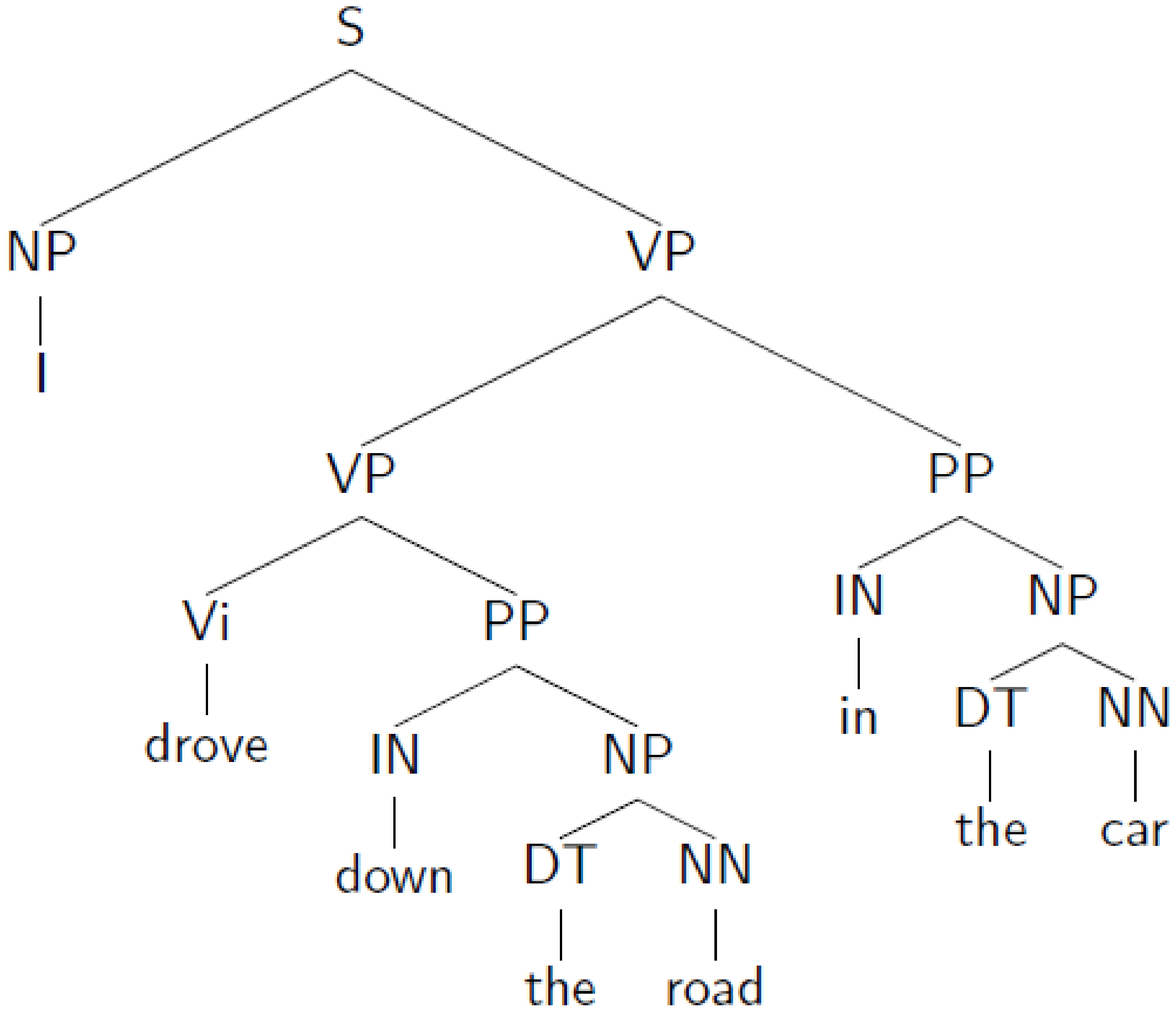
Sources of Ambiguity

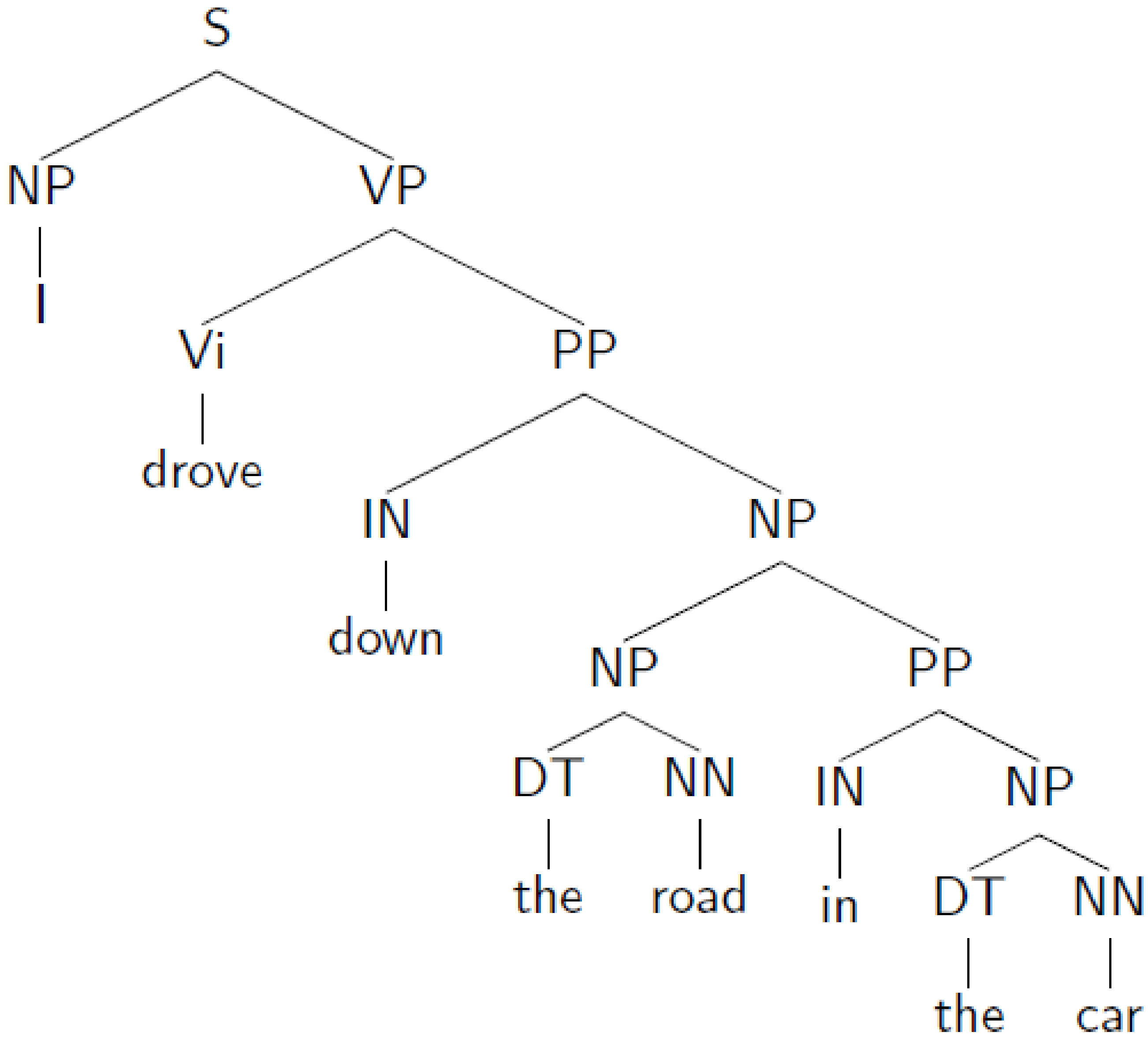
- Part-of-Speech ambiguity

  NN → duck

  Vi → duck



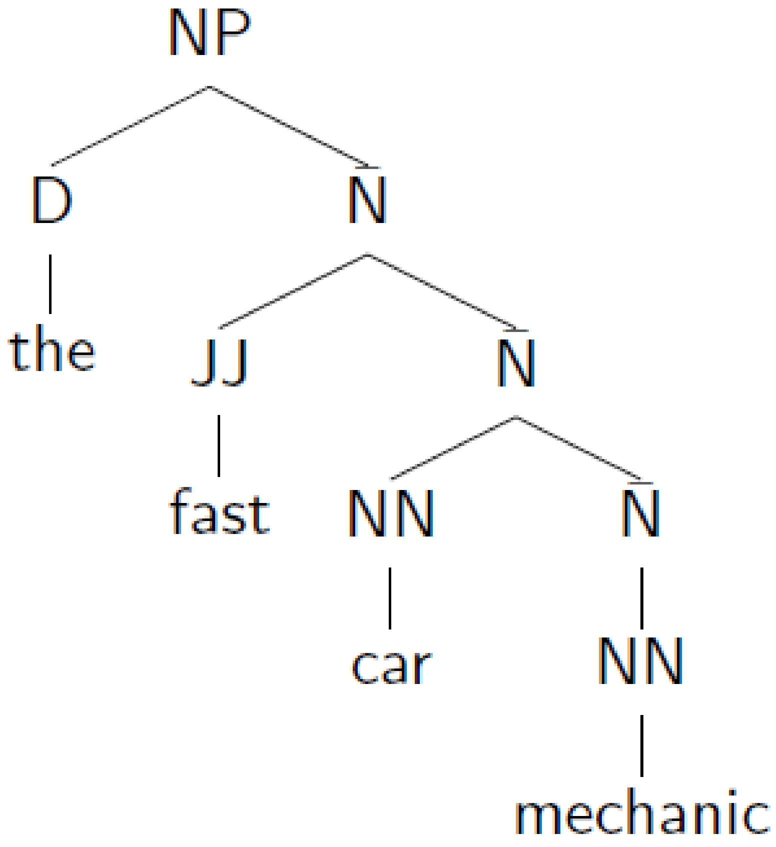

S
NP
I
VP
VP
PP
Vi
drove
PP
IN
down
NP
DT
the
NN
road
IN
in
NP
DT
the
NN
car

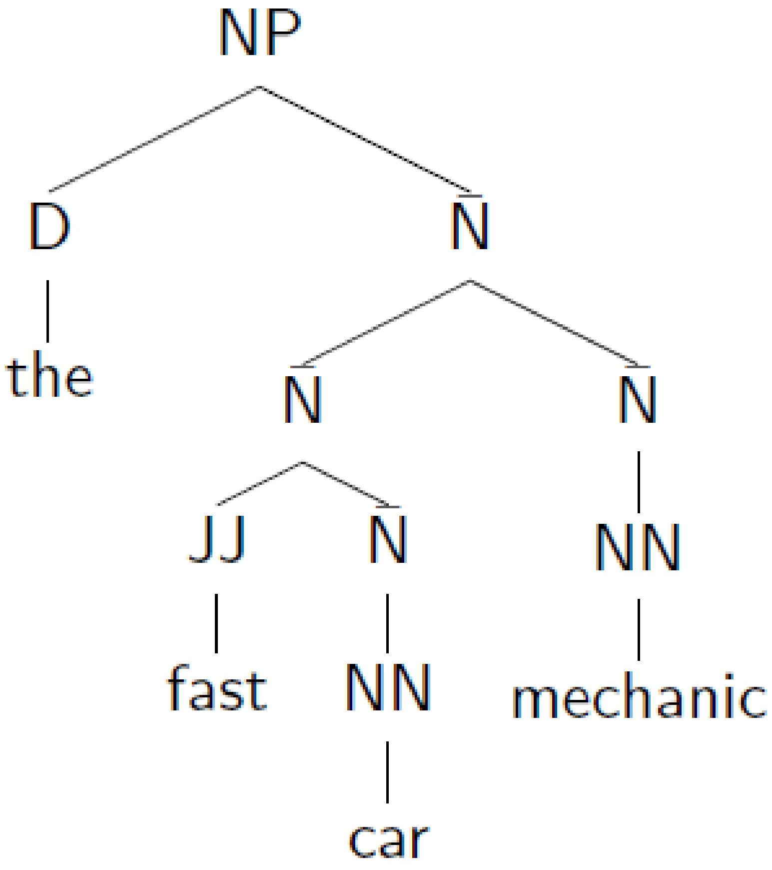

S
NP
VP
I
Vi
PP
drove
IN
NP
down
NP
PP
DT
NN
IN
NP
the
road
in
DT
NN
the
car

Two analyses for: John was believed to have been shot by Bill

Sources of Ambiguity: Noun Premodifiers

- Noun premodifiers:





Overview

- Probabilistic Context-Free Grammars (PCFGs)
- The CKY Algorithm for parsing with PCFGs

A Probabilistic Context-Free Grammar (PCFG)

S	$\Rightarrow$	NP	VP	1.0
VP	$\Rightarrow$	Vi		0.4
VP	$\Rightarrow$	Vt	NP	0.4
VP	$\Rightarrow$	VP	PP	0.2
NP	$\Rightarrow$	DT	NN	0.3
NP	$\Rightarrow$	NP	PP	0.7
PP	$\Rightarrow$	P	NP	1.0

Vi	$\Rightarrow$	sleeps	1.0
Vt	$\Rightarrow$	saw	1.0
NN	$\Rightarrow$	man	0.7
NN	$\Rightarrow$	woman	0.2
NN	$\Rightarrow$	telescope	0.1
DT	$\Rightarrow$	the	1.0
IN	$\Rightarrow$	with	0.5
IN	$\Rightarrow$	in	0.5

- Probability of a tree t with rules

$$\alpha_1 \rightarrow \beta_1, \alpha_2 \rightarrow \beta_2, \ldots, \alpha_n \rightarrow \beta_n$$

is $p(t) = \prod_{i=1}^{n} q(\alpha_i \rightarrow \beta_i)$ where $q(\alpha \rightarrow \beta)$ is the probability for rule $\alpha \rightarrow \beta$.

DERIVATION	RULES USED	PROBABILITY
S		1.0
	$S \rightarrow NP\ VP$	
NP VP		0.3
	$NP \rightarrow DT\ NN$	
DT NN VP		1.0
	$DT \rightarrow$ the	
the NN VP		0.1
	$NN \rightarrow$ dog	
the dog VP		0.4
	$VP \rightarrow Vi$	
the dog Vi		0.5
	$Vi \rightarrow$ laughs	
the dog laughs		

Properties of PCFGs

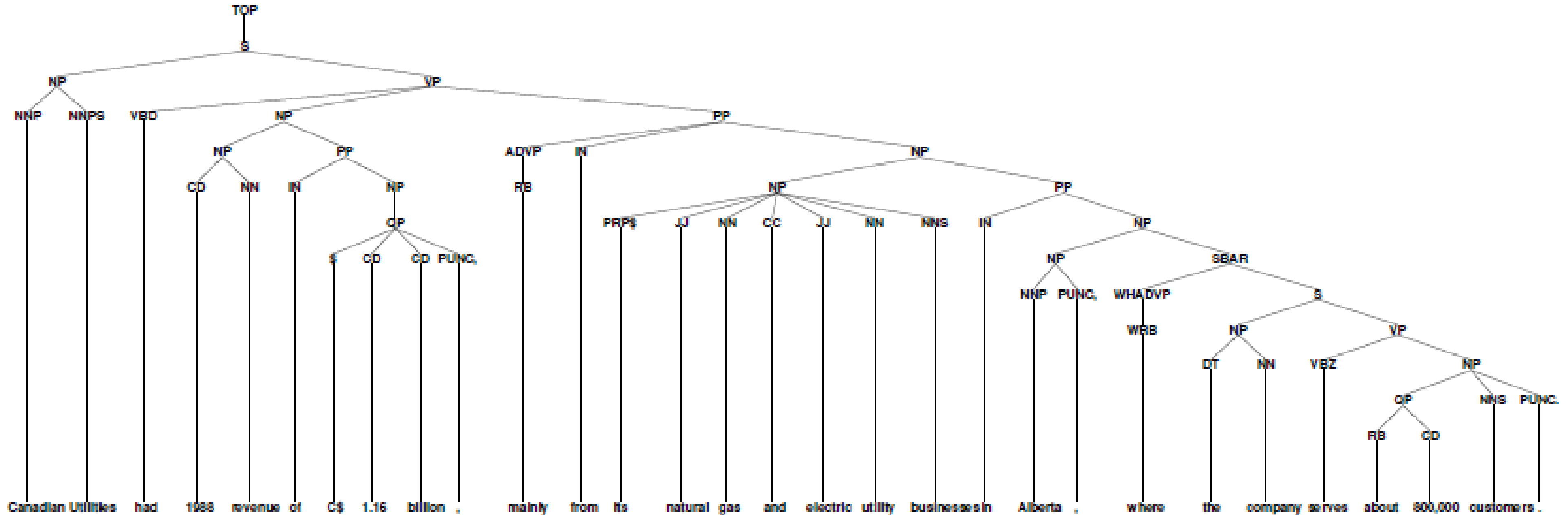
- Assigns a probability to each *left-most derivation*, or parse-tree, allowed by the underlying CFG
- Say we have a sentence s, set of derivations for that sentence is $\mathcal{T}(s)$. Then a PCFG assigns a probability $p(t)$ to each member of $\mathcal{T}(s)$. i.e., *we now have a ranking in order of probability.*
- The most likely parse tree for a sentence s is

$$\arg\max_{t \in \mathcal{T}(s)} p(t)$$

Data for Parsing Experiments: Treebanks

- Penn WSJ Treebank = 50,000 sentences with associated trees
- Usual set-up: 40,000 training sentences, 2400 test sentences

An example tree:



Deriving a PCFG from a Treebank

- Given a set of example trees (a treebank), the underlying CFG can simply be **all rules seen in the corpus**

- Maximum Likelihood estimates:

$$q_{ML}(\alpha \to \beta) = \frac{\text{Count}(\alpha \to \beta)}{\text{Count}(\alpha)}$$

  where the counts are taken from a training set of example trees.

- **If the training data is generated by a PCFG**, then as the training data size goes to infinity, the maximum-likelihood PCFG will converge to the same distribution as the "true" PCFG.

Parsing with a PCFG

- Given a PCFG and a sentence s, define $\mathcal{T}(s)$ to be the set of trees with s as the yield.
- Given a PCFG and a sentence s, how do we find

$$\arg\max_{t \in \mathcal{T}(s)} p(t)$$

Chomsky Normal Form

A context free grammar $G = (N, \Sigma, R, S)$ in Chomsky Normal Form is as follows

- N is a set of non-terminal symbols
- Σ is a set of terminal symbols
- R is a set of rules which take one of two forms:
  - $X \to Y_1 Y_2$ for $X \in N$, and $Y_1, Y_2 \in N$
  - $X \to Y$ for $X \in N$, and $Y \in \Sigma$
- $S \in N$ is a distinguished start symbol

How to covert rule in CNF

A → B C D

<u>CNF</u>

- A → X D
- X → B C

A Dynamic Programming Algorithm

- Given a PCFG and a sentence s, how do we find

$$\max_{t \in \mathcal{T}(s)} p(t)$$

- Notation:

$$n = \text{number of words in the sentence}$$
$$w_i = i\text{'th word in the sentence}$$
$$N = \text{the set of non-terminals in the grammar}$$
$$S = \text{the start symbol in the grammar}$$

- Define a dynamic programming table

$$\pi[i, j, X] = \text{maximum probability of a constituent with non-terminal } X \text{ spanning words } i \ldots j \text{ inclusive}$$

- Our goal is to calculate $\max_{t \in \mathcal{T}(s)} p(t) = \pi[1, n, S]$

An Example

the dog saw the man with the telescope

The dog saw the man with the telescope

Iteration 0, i = j, span length is 1 word

S	$\Rightarrow$	NP	VP	1.0
VP	$\Rightarrow$	Vi		0.4
VP	$\Rightarrow$	Vt	NP	0.4
VP	$\Rightarrow$	VP	PP	0.2
NP	$\Rightarrow$	DT	NN	0.3
NP	$\Rightarrow$	NP	PP	0.7
PP	$\Rightarrow$	P	NP	1.0

Vi	$\Rightarrow$	sleeps	1.0
Vt	$\Rightarrow$	saw	1.0
NN	$\Rightarrow$	man	0.7
NN	$\Rightarrow$	woman	0.2
NN	$\Rightarrow$	telescope	0.1
DT	$\Rightarrow$	the	1.0
IN	$\Rightarrow$	with	0.5
IN	$\Rightarrow$	in	0.5

The dog saw the man with the telescope

Iteration 1, j = i+1, span length is 2 words

S	⇒	NP	VP	1.0
VP	⇒	Vi		0.4
VP	⇒	Vt	NP	0.4
VP	⇒	VP	PP	0.2
NP	⇒	DT	NN	0.3
NP	⇒	NP	PP	0.7
PP	⇒	P	NP	1.0

Vi	⇒	sleeps	1.0
Vt	⇒	saw	1.0
NN	⇒	man	0.7
NN	⇒	woman	0.2
NN	⇒	telescope	0.1
DT	⇒	the	1.0
IN	⇒	with	0.5
IN	⇒	in	0.5

The dog saw the man with the telescope

Iteration 2, j = i+2, span length is 3 words

S	⇒	NP	VP	1.0
VP	⇒	Vi		0.4
VP	⇒	Vt	NP	0.4
VP	⇒	VP	PP	0.2
NP	⇒	DT	NN	0.3
NP	⇒	NP	PP	0.7
PP	⇒	P	NP	1.0

Vi	⇒	sleeps	1.0
Vt	⇒	saw	1.0
NN	⇒	man	0.7
NN	⇒	woman	0.2
NN	⇒	telescope	0.1
DT	⇒	the	1.0
IN	⇒	with	0.5
IN	⇒	in	0.5

The dog saw the man with the telescope

Iteration 3, j = i+3, span length is 4 words

S	$\Rightarrow$	NP	VP	1.0
VP	$\Rightarrow$	Vi		0.4
VP	$\Rightarrow$	Vt	NP	0.4
VP	$\Rightarrow$	VP	PP	0.2
NP	$\Rightarrow$	DT	NN	0.3
NP	$\Rightarrow$	NP	PP	0.7
PP	$\Rightarrow$	P	NP	1.0

Vi	$\Rightarrow$	sleeps	1.0
Vt	$\Rightarrow$	saw	1.0
NN	$\Rightarrow$	man	0.7
NN	$\Rightarrow$	woman	0.2
NN	$\Rightarrow$	telescope	0.1
DT	$\Rightarrow$	the	1.0
IN	$\Rightarrow$	with	0.5
IN	$\Rightarrow$	in	0.5

The dog saw the man with the telescope

Iteration 4, j = i+4, span length is 5 words

S	⇒	NP	VP	1.0
VP	⇒	Vi		0.4
VP	⇒	Vt	NP	0.4
VP	⇒	VP	PP	0.2
NP	⇒	DT	NN	0.3
NP	⇒	NP	PP	0.7
PP	⇒	P	NP	1.0

Vi	⇒	sleeps	1.0
Vt	⇒	saw	1.0
NN	⇒	man	0.7
NN	⇒	woman	0.2
NN	⇒	telescope	0.1
DT	⇒	the	1.0
IN	⇒	with	0.5
IN	⇒	in	0.5

The dog saw the man with the telescope

Iteration 5, j = i+5, span length is 6 words

S	⇒	NP	VP	1.0
VP	⇒	Vi		0.4
VP	⇒	Vt	NP	0.4
VP	⇒	VP	PP	0.2
NP	⇒	DT	NN	0.3
NP	⇒	NP	PP	0.7
PP	⇒	P	NP	1.0

Vi	⇒	sleeps	1.0
Vt	⇒	saw	1.0
NN	⇒	man	0.7
NN	⇒	woman	0.2
NN	⇒	telescope	0.1
DT	⇒	the	1.0
IN	⇒	with	0.5
IN	⇒	in	0.5

The dog saw the man with the telescope

Iteration 6, j = i+6, span length is 7 words

S	$\Rightarrow$	NP	VP	1.0
VP	$\Rightarrow$	Vi		0.4
VP	$\Rightarrow$	Vt	NP	0.4
VP	$\Rightarrow$	VP	PP	0.2
NP	$\Rightarrow$	DT	NN	0.3
NP	$\Rightarrow$	NP	PP	0.7
PP	$\Rightarrow$	P	NP	1.0

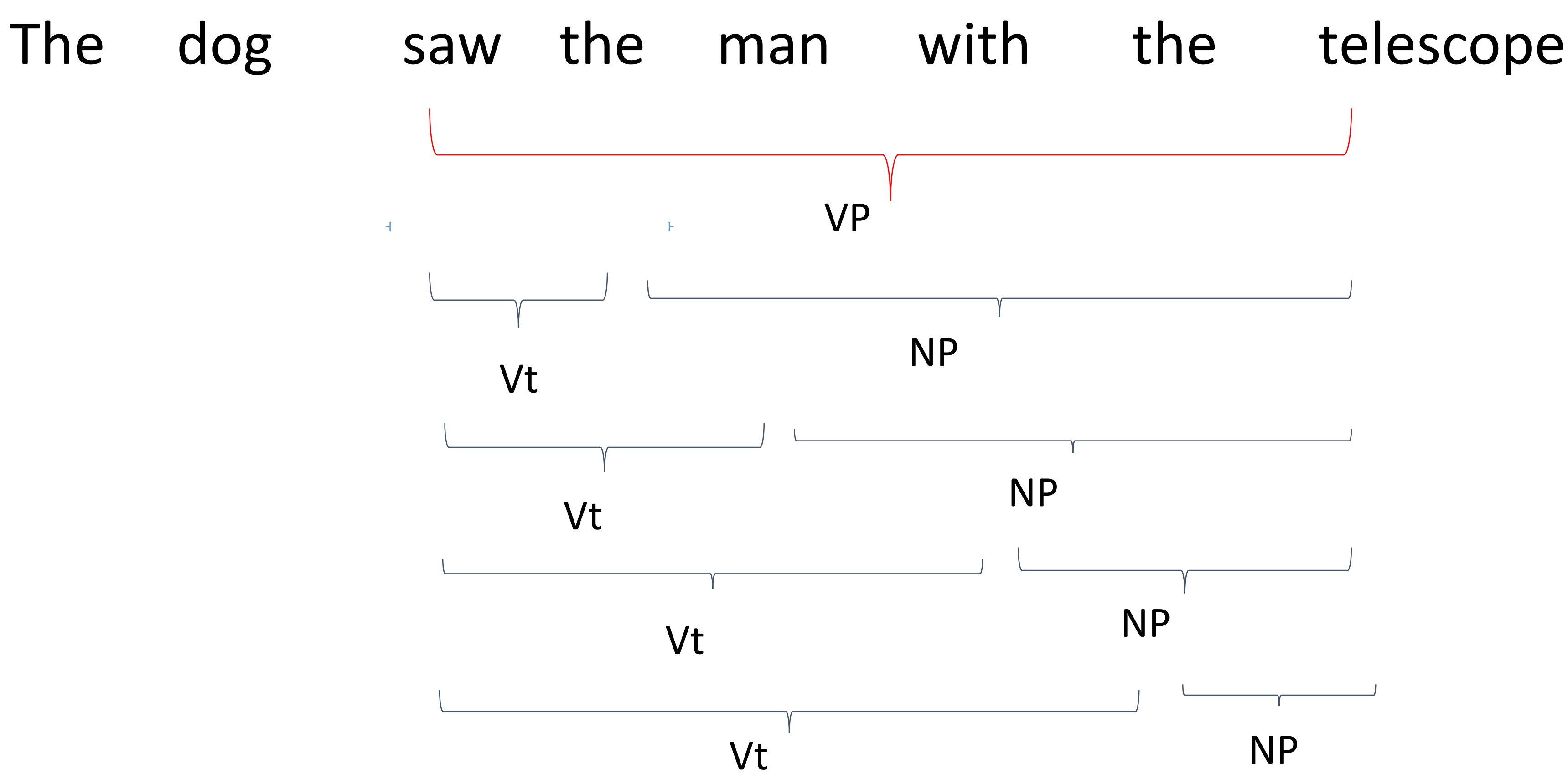
Vi	$\Rightarrow$	sleeps	1.0
Vt	$\Rightarrow$	saw	1.0
NN	$\Rightarrow$	man	0.7
NN	$\Rightarrow$	woman	0.2
NN	$\Rightarrow$	telescope	0.1
DT	$\Rightarrow$	the	1.0
IN	$\Rightarrow$	with	0.5
IN	$\Rightarrow$	in	0.5

The dog saw the man with the telescope

Iteration 7, j = i+7, span length is 8 words

S	⇒	NP	VP	1.0
VP	⇒	Vi		0.4
VP	⇒	Vt	NP	0.4
VP	⇒	VP	PP	0.2
NP	⇒	DT	NN	0.3
NP	⇒	NP	PP	0.7
PP	⇒	P	NP	1.0

Vi	⇒	sleeps	1.0
Vt	⇒	saw	1.0
NN	⇒	man	0.7
NN	⇒	woman	0.2
NN	⇒	telescope	0.1
DT	⇒	the	1.0
IN	⇒	with	0.5
IN	⇒	in	0.5



VP → Vt NP

Iteration 4, j = i+4, span length is 5 words

The boundary for the 2 right hand side non terminals has various options

A Dynamic Programming Algorithm for the Sum

- Given a PCFG and a sentence s, how do we find

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

- Notation:

$$n = \text{number of words in the sentence}$$
$$w_i = i\text{'th word in the sentence}$$
$$N = \text{the set of non-terminals in the grammar}$$
$$S = \text{the start symbol in the grammar}$$

- Define a dynamic programming table

$$\pi[i, j, X] \quad = \quad \text{sum of probabilities for constituent with non-terminal } X \text{ spanning words } i \ldots j \text{ inclusive}$$

- Our goal is to calculate $\sum_{t \in \mathcal{T}(s)} p(t) = \pi[1, n, S]$

A Dynamic Programming Algorithm

- Base case definition: for all $i = 1 \ldots n$, for $X \in N$

  $$\pi[i, i, X] = q(X \rightarrow w_i)$$

  (note: define $q(X \rightarrow w_i) = 0$ if $X \rightarrow w_i$ is not in the grammar)

- Recursive definition: for all $i = 1 \ldots n$, $j = (i+1) \ldots n$, $X \in N$,

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

An Example

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

the dog saw the man with the telescope

The Full Dynamic Programming Algorithm

Input: a sentence $s = x_1 \ldots x_n$, a PCFG $G = (N, \Sigma, S, R, q)$.
Initialization:
For all $i \in \{1 \ldots n\}$, for all $X \in N$,

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

Algorithm:

- For $l = 1 \ldots (n-1)$
  - For $i = 1 \ldots (n-l)$
    - Set $j = i + l$
    - For all $X \in N$, calculate

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

      and

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

Summary

- PCFGs augments CFGs by including a probability for each rule in the grammar.
- The probability for a parse tree is the product of probabilities for the rules in the tree
- To build a PCFG-parsed parser:
  1. Learn a PCFG from a treebank
  2. Given a test data sentence, use the CKY algorithm to compute the highest probability tree for the sentence under the PCFG

CYK Example

Input String:

b a a b a

- $S \rightarrow AB$ 0.3
- $S \rightarrow BC$ 0.7
- $A \rightarrow BA$ 0.4
- $A \rightarrow a$ 0.6

$B \rightarrow CC$ 0.4

$B \rightarrow b$ 0.6

$C \rightarrow AB$ 0.5

$C \rightarrow a$ 0.5

Initialization:

π [1,1,S] = 0, π [1,1,A] = 0, π [1,1,B] = 0.6, π [1,1,C] = 0

π [2,2,S] = 0, π [2,2,A] = 0.6, π [2,2,B] = 0, π [2,2,C] = 0.5

π [3,3,S] = 0, π [3,3,A] = 0.6, π [3,3,B] = 0, π [3,3,C] = 0.5

π [4,4,S] = 0, π [4,4,A] = 0, π [4,4,B] = 0.6, π [4,4,C] = 0

π [5,5,S] = 0, π [5,5,A] = 0.6, π [5,5,B] = 0, π [5,5,C] = 0.5

CYK Example

Input String:

b a a b a

- $S \rightarrow AB$ 0.3
- $S \rightarrow BC$ 0.7
- $A \rightarrow BA$ 0.4
- $A \rightarrow a$ 0.6

$B \rightarrow CC$ 0.4

$B \rightarrow b$ 0.6

$C \rightarrow AB$ 0.5

$C \rightarrow a$ 0.5

First Iteration **(l = 1, i = 1):**

$$\pi [1,2,S] = \text{Max} \begin{cases} q(S \rightarrow AB) * \pi[1,1,A] * \pi[2,2,B] = 0.3*0*0 = 0 \\ q(S \rightarrow BC) * \pi[1,1,B] * \pi[2,2,C] = 0.7*0.6*0.5 = 0.21 \end{cases}$$

$$\pi [1,2,A] = q(A \rightarrow BA) * \pi[1,1,B] * \pi[2,2,A] = 0.4*0.6*0.6 = 0.144$$

$$\pi [1,2,B] = q(B \rightarrow CC) * \pi[1,1,C] * \pi[2,2,C] = 0.4*0*0.5 = 0$$

$$\pi [1,2,C] = q(C \rightarrow AB) * \pi[1,1,A] * \pi[2,2,B] = 0.5*0*0 = 0$$

CYK Example

Input String:

b a a b a

- $S \rightarrow AB$ 0.3
- $S \rightarrow BC$ 0.7
- $A \rightarrow BA$ 0.4
- $A \rightarrow a$ 0.6

$B \rightarrow CC$ 0.4

$B \rightarrow b$ 0.6

$C \rightarrow AB$ 0.5

$C \rightarrow a$ 0.5

First Iteration **(l = 1, i = 2):**

$$\pi\,[2,3,S] = \text{Max}\begin{cases} q\,(S \rightarrow AB) * \pi\,[2,2,A] * \pi\,[3,3,B] = 0.3*0.6*0 = 0 \\ q\,(S \rightarrow BC) * \pi\,[2,2,B] * \pi\,[3,3,C] = 0.7*0*0.5 = 0 \end{cases}$$

$$\pi\,[2,3,A] = q\,(A \rightarrow BA) * \pi\,[2,2,B] * \pi\,[3,3,A] = 0.4*0*0.6 = 0$$

$$\pi\,[2,3,B] = q\,(B \rightarrow CC) * \pi\,[2,2,C] * \pi\,[3,3,C] = 0.4*0.5*0.5 = 0.1$$

$$\pi\,[2,3,C] = q\,(C \rightarrow AB) * \pi\,[2,2,A] * \pi\,[3,3,B] = 0.5*0.6*0 = 0$$

CYK Example

Input String:

b a a b a

- $S \rightarrow AB$ 0.3
- $S \rightarrow BC$ 0.7
- $A \rightarrow BA$ 0.4
- $A \rightarrow a$ 0.6

$B \rightarrow CC$ 0.4

$B \rightarrow b$ 0.6

$C \rightarrow AB$ 0.5

$C \rightarrow a$ 0.5

First Iteration **(l = 1, i = 3):**

$$\pi[3,4,S] = \text{Max}\begin{cases} q(S \rightarrow AB) * \pi[3,3,A] * \pi[4,4,B] = 0.3*0.6*0 = 0 \\ q(S \rightarrow BC) * \pi[3,3,B] * \pi[4,4,C] = 0.7*0*0.5 = 0 \end{cases}$$

$$\pi[3,4,A] = q(A \rightarrow BA) * \pi[3,3,B] * \pi[4,4,A] = 0.4*0*0.4 = 0$$

$$\pi[3,4,B] = q(B \rightarrow CC) * \pi[3,3,C] * \pi[4,4,C] = 0.4*0.5*0.5 = 0.1$$

$$\pi[3,4,C] = q(C \rightarrow AB) * \pi[3,3,A] * \pi[4,4,B] = 0.5*0.6*0 = 0$$

CYK Example

Input String:

b a a b a

- S → AB 0.3
- S → BC 0.7
- A → BA 0.4
- A → a 0.6

B → CC 0.4

B → b 0.6

C → AB 0.5

C → a 0.5

First Iteration **(l = 1, i = 4):**

$$\pi\,[4,5,S] = \text{Max}\begin{cases} q\,(S \to AB) * \pi\,[4,4,A] * \pi\,[5,5,B] = 0.3*0.4*0 = 0 \\ q\,(S \to BC) * \pi\,[4,4,B] * \pi\,[5,5,C] = 0.7*0*0.5 = 0 \end{cases}$$

$$\pi\,[4,5,A] = q\,(A \to BA) * \pi\,[4,4,B] * \pi\,[5,5,A] = 0.4*0.6*0.6 = 0.144$$

$$\pi\,[4,5,B] = q\,(B \to CC) * \pi\,[4,4,C] * \pi\,[5,5,C] = 0.4*0*0.5 = 0$$

$$\pi\,[4,5,C] = q\,(C \to AB) * \pi\,[4,4,A] * \pi\,[5,5,B] = 0.5*0*0 = 0$$

Input String:

b a a b a

- $S \rightarrow AB$ 0.3
- $S \rightarrow BC$ 0.7
- $A \rightarrow BA$ 0.4
- $A \rightarrow a$ 0.6

$B \rightarrow CC$ 0.4

$B \rightarrow b$ 0.6

$C \rightarrow AB$ 0.5

$C \rightarrow a$ 0.5

Second Iteration **(l = 2, i = 1):**

$$\pi[1,3,S] = \text{Max}\begin{cases} q(S \rightarrow AB) * \pi[1,1,A] * \pi[2,3,B] = \\ q(S \rightarrow AB) * \pi[1,2,A] * \pi[3,3,B] = \\ q(S \rightarrow BC) * \pi[1,1,B] * \pi[2,3,C] = \\ q(S \rightarrow BC) * \pi[1,2,B] * \pi[3,3,C] = \end{cases}$$

$$\pi[1,3,A] = \text{Max}\begin{cases} q(A \rightarrow BA) * \pi[1,1,B] * \pi[2,3,A] = \\ q(A \rightarrow BA) * \pi[1,2,B] * \pi[3,3,A] = \end{cases}$$

$$\pi[1,3,B] = \text{Max}\begin{cases} q(B \rightarrow CC) * \pi[1,1,C] * \pi[2,3,C] = \\ q(B \rightarrow CC) * \pi[1,2,C] * \pi[3,3,C] = \end{cases}$$

$$\pi[1,3,C] = \text{Max}\begin{cases} q(C \rightarrow AB) * \pi[1,1,A] * \pi[2,3,B] = \\ q(C \rightarrow AB) * \pi[1,2,A] * \pi[3,3,B] = \end{cases}$$

Input String:

b a a b a

- $S \rightarrow AB$ 0.3
- $S \rightarrow BC$ 0.7
- $A \rightarrow BA$ 0.4
- $A \rightarrow a$ 0.6

$B \rightarrow CC$ 0.4

$B \rightarrow b$ 0.6

$C \rightarrow AB$ 0.5

$C \rightarrow a$ 0.5

Second Iteration **(l = 2, i = 2):**

$$\pi[2,4,S] = \text{Max}\begin{cases} q(S \rightarrow AB) * \pi[2,2,A] * \pi[3,4,B] = \\ q(S \rightarrow AB) * \pi[2,3,A] * \pi[4,4,B] = \\ q(S \rightarrow BC) * \pi[2,2,B] * \pi[3,4,C] = \\ q(S \rightarrow BC) * \pi[2,3,B] * \pi[4,4,C] = \end{cases}$$

$$\pi[2,4,A] = \text{Max}\begin{cases} q(A \rightarrow BA) * \pi[2,2,B] * \pi[3,4,A] = \\ q(A \rightarrow BA) * \pi[2,3,B] * \pi[4,4,A] = \end{cases}$$

$$\pi[2,4,B] = \text{Max}\begin{cases} q(B \rightarrow CC) * \pi[2,2,C] * \pi[3,4,C] = \\ q(B \rightarrow CC) * \pi[2,3,C] * \pi[4,4,C] = \end{cases}$$

$$\pi[2,4,C] = \text{Max}\begin{cases} q(C \rightarrow AB) * \pi[2,2,A] * \pi[3,4,B] = \\ q(C \rightarrow AB) * \pi[2,3,A] * \pi[4,4,B] = \end{cases}$$

Input String:

b a a b a

- $S \rightarrow AB$ 0.3
- $S \rightarrow BC$ 0.7
- $A \rightarrow BA$ 0.4
- $A \rightarrow a$ 0.6

$B \rightarrow CC$ 0.4

$B \rightarrow b$ 0.6

$C \rightarrow AB$ 0.5

$C \rightarrow a$ 0.5

Second Iteration **(l = 2, i = 3):**

$$\pi[3,5,S] = \text{Max}\begin{cases} q(S \rightarrow AB) * \pi[3,3,A] * \pi[4,5,B] = \\ q(S \rightarrow AB) * \pi[3,4,A] * \pi[5,5,B] = \\ q(S \rightarrow BC) * \pi[3,3,B] * \pi[4,5,C] = \\ q(S \rightarrow BC) * \pi[3,4,B] * \pi[5,5,C] = \end{cases}$$

$$\pi[3,5,A] = \text{Max}\begin{cases} q(A \rightarrow BA) * \pi[3,3,B] * \pi[4,5,A] = \\ q(A \rightarrow BA) * \pi[3,4,B] * \pi[5,5,A] = \end{cases}$$

$$\pi[3,5,B] = \text{Max}\begin{cases} q(B \rightarrow CC) * \pi[3,3,C] * \pi[4,5,C] = \\ q(B \rightarrow CC) * \pi[3,4,C] * \pi[5,5,C] = \end{cases}$$

$$\pi[3,5,C] = \text{Max}\begin{cases} q(C \rightarrow AB) * \pi[3,3,A] * \pi[4,5,B] = \\ q(C \rightarrow AB) * \pi[3,4,A] * \pi[5,5,B] = \end{cases}$$

Input String:

b a a b a

- S → AB 0.3
- S → BC 0.7
- A → BA 0.4
- A → a 0.6

B → CC 0.4

B → b 0.6

C → AB 0.5

C → a 0.5

Third Iteration **(l = 3, i = 1):**

$$\pi[1,4,S] = \text{Max}\begin{cases} q(S \to AB) * \pi[1,1,A] * \pi[2,4,B] \\ q(S \to AB) * \pi[1,2,A] * \pi[3,4,B] \\ q(S \to AB) * \pi[1,3,A] * \pi[4,4,B] \\ q(S \to BC) * \pi[1,1,B] * \pi[2,4,C] \\ q(S \to BC) * \pi[1,2,B] * \pi[3,4,C] \\ q(S \to BC) * \pi[1,3,B] * \pi[4,4,C] \end{cases}$$

Input String:

b a a b a

- $S \rightarrow AB$ 0.3
- $S \rightarrow BC$ 0.7
- $A \rightarrow BA$ 0.4
- $A \rightarrow a$ 0.6

$B \rightarrow CC$ 0.4

$B \rightarrow b$ 0.6

$C \rightarrow AB$ 0.5

$C \rightarrow a$ 0.5

Third Iteration **(l = 3, i = 1):**

$$\pi[1,4,A] = \text{Max} \begin{cases} q(A \rightarrow BA) * \pi[1,1,B] * \pi[2,4,A] \\ q(A \rightarrow BA) * \pi[1,2,B] * \pi[3,4,A] \\ q(A \rightarrow BA) * \pi[1,3,B] * \pi[4,4,A] \end{cases}$$

$$\pi[1,4,B] = \text{Max} \begin{cases} q(B \rightarrow CC) * \pi[1,1,C] * \pi[2,4,C] \\ q(B \rightarrow CC) * \pi[1,2,C] * \pi[3,4,C] \\ q(B \rightarrow CC) * \pi[1,3,C] * \pi[4,4,C] \end{cases}$$

$$\pi[1,4,C] = \text{Max} \begin{cases} q(C \rightarrow AB) * \pi[1,1,A] * \pi[2,4,B] \\ q(C \rightarrow AB) * \pi[1,2,A] * \pi[3,4,B] \\ q(C \rightarrow AB) * \pi[1,3,A] * \pi[4,4,B] \end{cases}$$

Input String:

b a a b a

- S → AB 0.3
- S → BC 0.7
- A → BA 0.4
- A → a 0.6

B → CC 0.4

B → b 0.6

C → AB 0.5

C → a 0.5

Third Iteration **(l = 3, i = 2):**

$$\pi[2,5,S] = \text{Max}\begin{cases} q(S \rightarrow AB) * \pi[2,2,A] * \pi[3,5,B] \\ q(S \rightarrow AB) * \pi[2,3,A] * \pi[4,5,B] \\ q(S \rightarrow AB) * \pi[2,4,A] * \pi[5,5,B] \\ q(S \rightarrow BC) * \pi[2,2,B] * \pi[3,5,C] \\ q(S \rightarrow BC) * \pi[2,3,B] * \pi[4,5,C] \\ q(S \rightarrow BC) * \pi[2,4,B] * \pi[5,5,C] \end{cases}$$

Input String:

b a a b a

- $S \rightarrow AB$ 0.3
- $S \rightarrow BC$ 0.7
- $A \rightarrow BA$ 0.4
- $A \rightarrow a$ 0.6

$B \rightarrow CC$ 0.4

$B \rightarrow b$ 0.6

$C \rightarrow AB$ 0.5

$C \rightarrow a$ 0.5

Third Iteration **(l = 3, i = 2):**

$$\pi[2,5,A] = \text{Max} \begin{cases} q(A \rightarrow BA) * \pi[2,2,B] * \pi[3,5,A] = \\ q(A \rightarrow BA) * \pi[2,3,B] * \pi[4,5,A] = \\ q(A \rightarrow BA) * \pi[2,4,B] * \pi[5,5,A] = \end{cases}$$

$$\pi[2,5,B] = \text{Max} \begin{cases} q(B \rightarrow CC) * \pi[2,2,C] * \pi[3,5,C] = \\ q(B \rightarrow CC) * \pi[2,3,C] * \pi[4,5,C] = \\ q(B \rightarrow CC) * \pi[2,4,C] * \pi[5,5,C] = \end{cases}$$

$$\pi[2,5,C] = \text{Max} \begin{cases} q(C \rightarrow AB) * \pi[2,2,A] * \pi[3,5,B] = \\ q(C \rightarrow AB) * \pi[2,3,A] * \pi[4,5,B] = \\ q(C \rightarrow AB) * \pi[2,4,A] * \pi[5,5,B] = \end{cases}$$

Input String:

b a a b a

- $S \rightarrow AB$ 0.3
- $S \rightarrow BC$ 0.7
- $A \rightarrow BA$ 0.4
- $A \rightarrow a$ 0.6

$B \rightarrow CC$ 0.4

$B \rightarrow b$ 0.6

$C \rightarrow AB$ 0.5

$C \rightarrow a$ 0.5

Fourth Iteration **(l = 4, i = 1):**

$$\pi[1,5,S] = \text{Max}\begin{cases} q(S \rightarrow AB) * \pi[1,1,A] * \pi[2,5,B] = \\ q(S \rightarrow AB) * \pi[1,2,A] * \pi[3,5,B] = \\ q(S \rightarrow AB) * \pi[1,3,A] * \pi[4,5,B] = \\ q(S \rightarrow AB) * \pi[1,4,A] * \pi[5,5,B] = \\ q(S \rightarrow BC) * \pi[1,1,B] * \pi[2,5,C] = \\ q(S \rightarrow BC) * \pi[1,2,B] * \pi[3,5,C] = \\ q(S \rightarrow BC) * \pi[1,3,B] * \pi[4,5,C] = \\ q(S \rightarrow BC) * \pi[1,4,B] * \pi[5,5,C] = \end{cases}$$

Reading

- Chapter 13 and 14 for Speech and Language Processing Third Edition
- https://web.stanford.edu/~jurafsky/slp3/13.pdf
- https://web.stanford.edu/~jurafsky/slp3/14.pdf