CSCI 544: Applied Natural Language Processing

Constituency Parsing

Xuezhe Ma (Max)

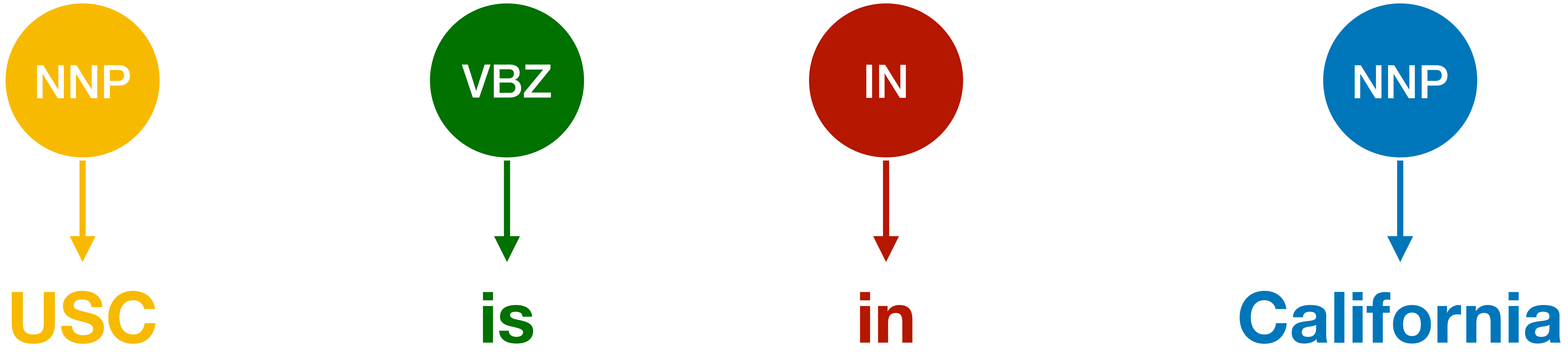
USC University of Southern California

Recap: Sequence Labeling

A type of structured prediction tasks

$Y = < y_i, y_2, \ldots, y_n >$

$X = < x_i, x_2, \ldots, x_n >$



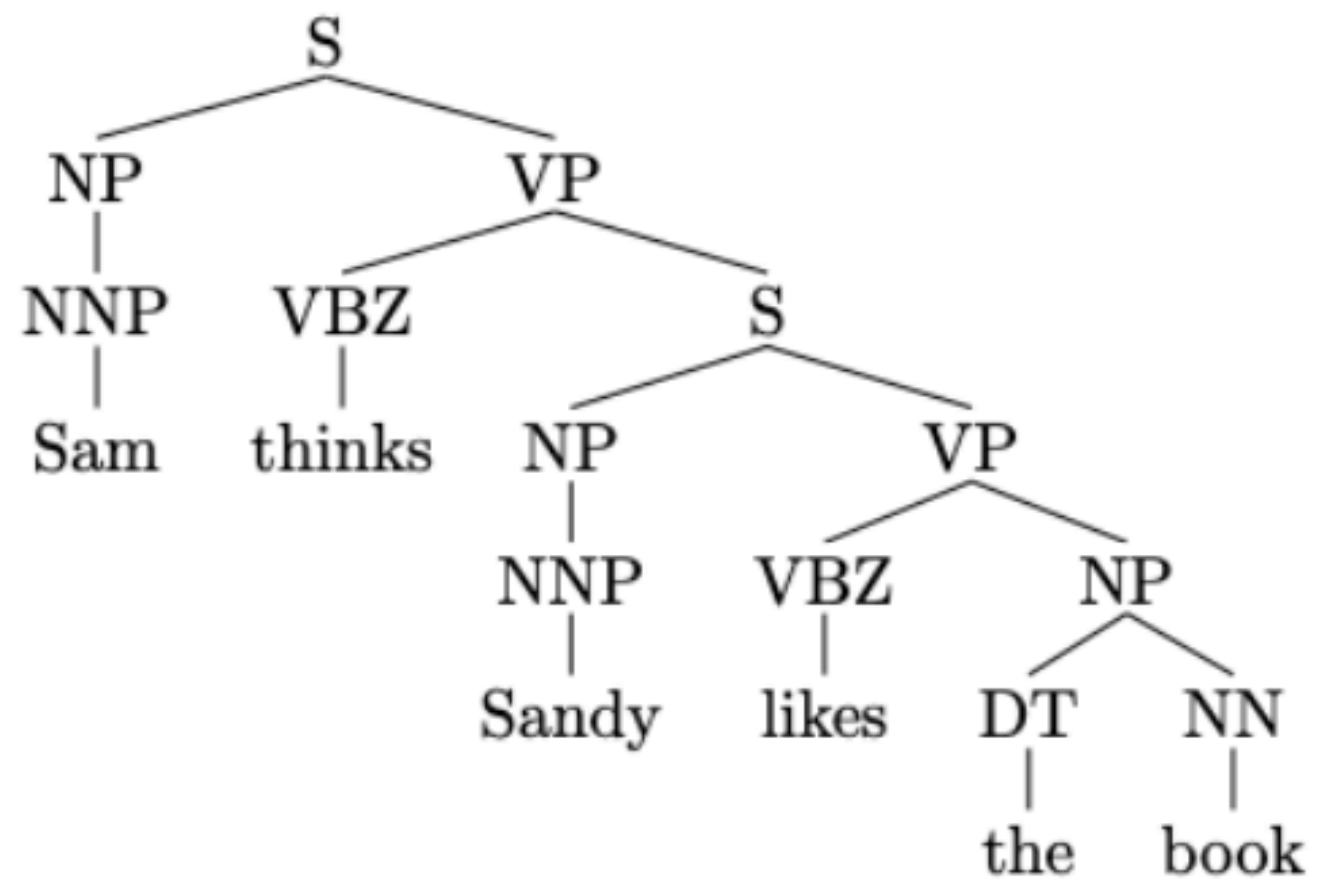
Assigning each token of X, e.g. x_i a corresponding label y_i

Syntactic Structure: Constituency vs. Dependency

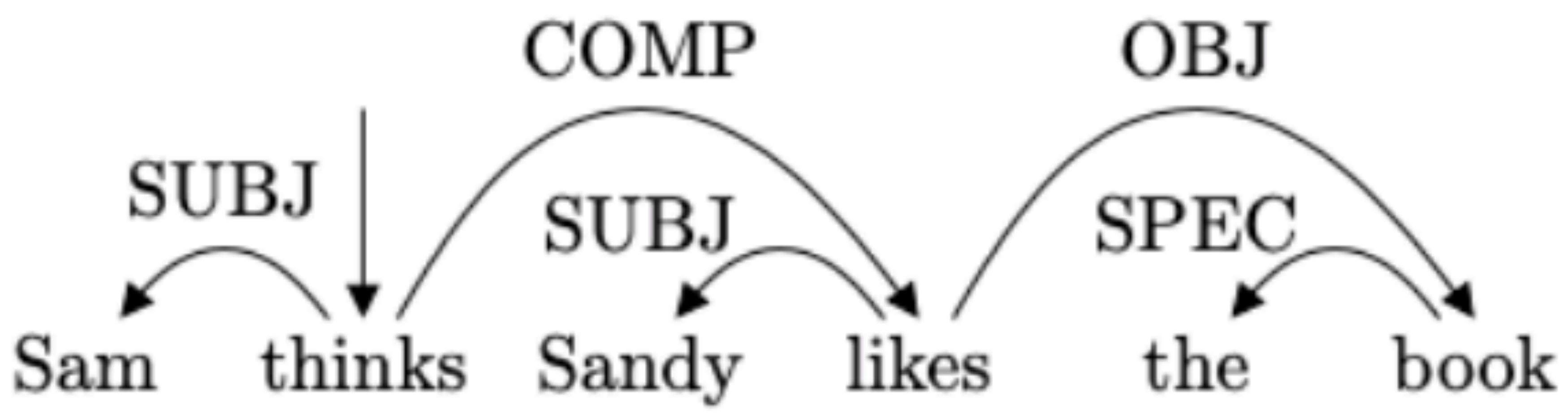
Theme: How to represent the structure of sentences using (syntax) trees?

Two views of linguistic structures

- **Constituency** (this lecture)
  - = phrase structure grammar
  - Based on context-free grammars (CFGs)



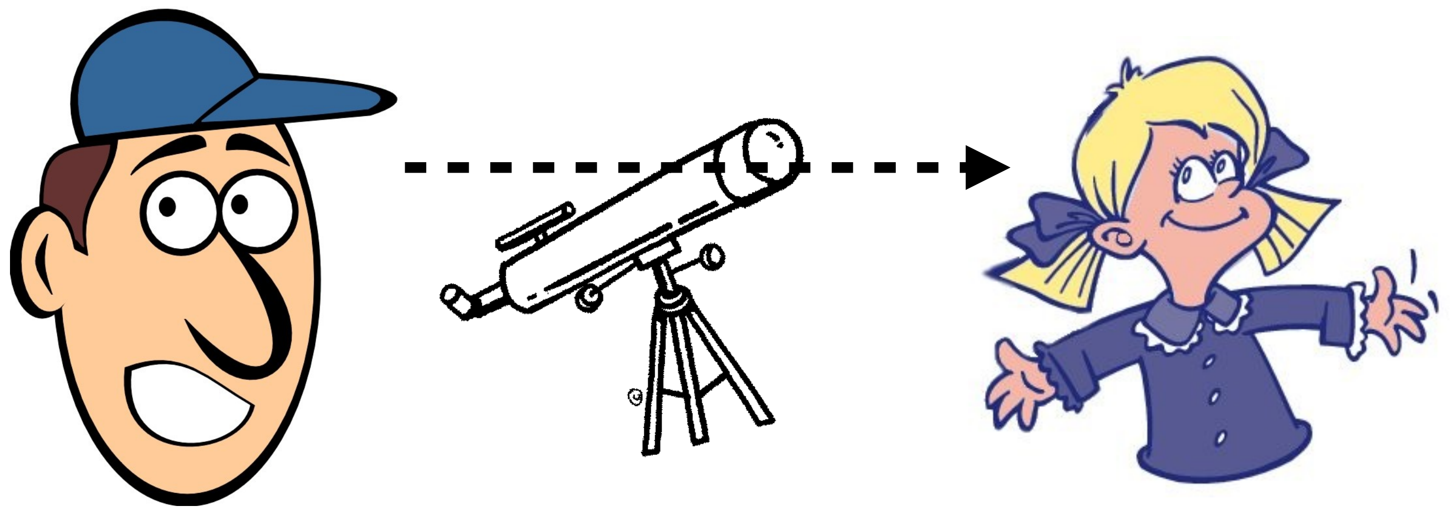
- **Dependency** (next lecture)

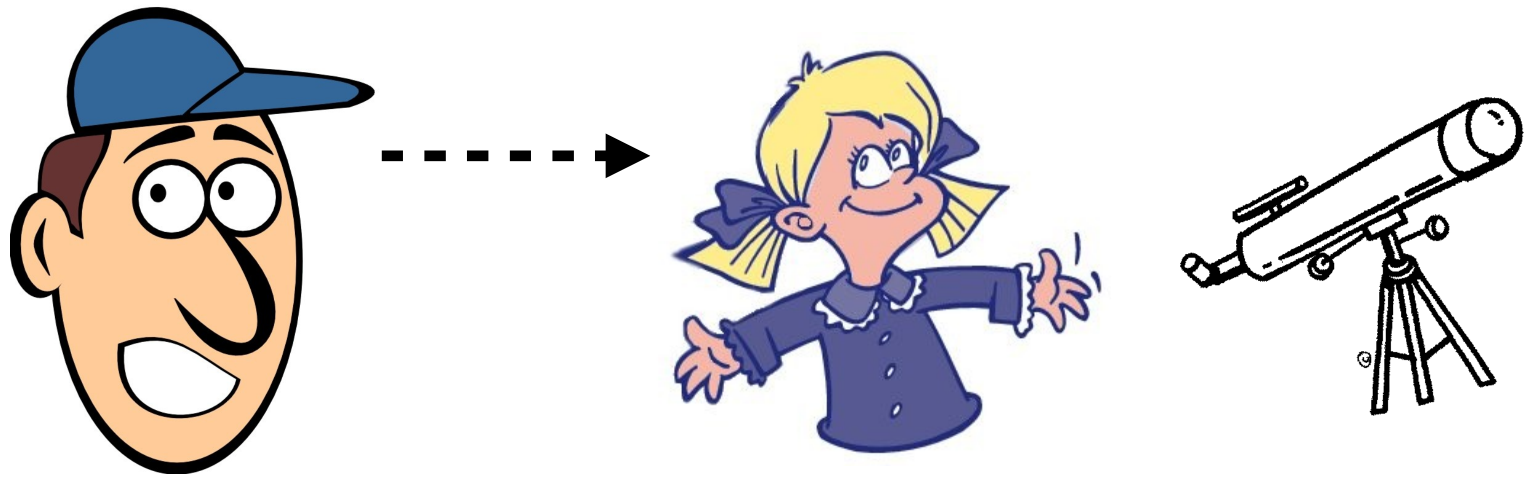


Slide credit: Danqi Chen COS 484/584

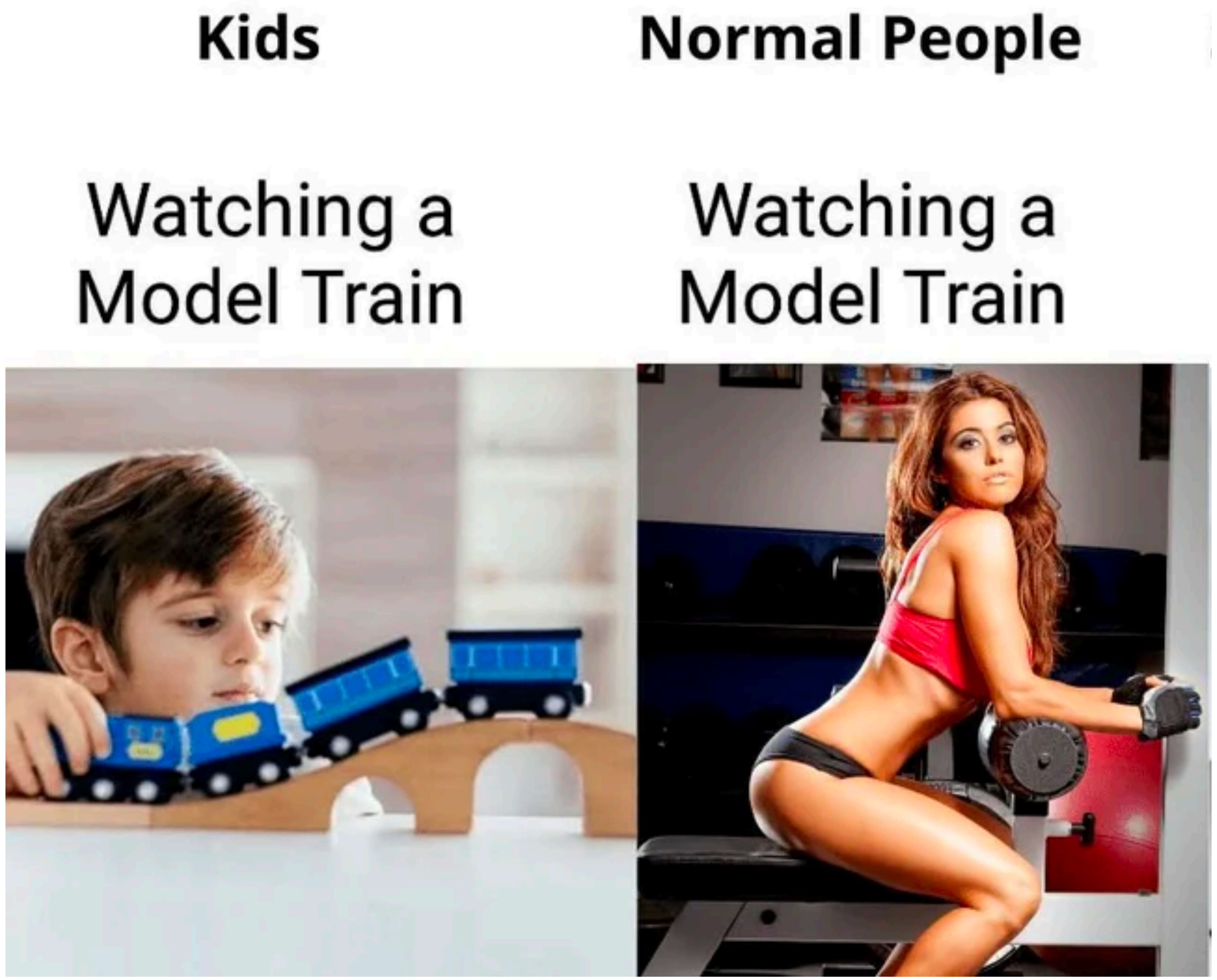
Why Syntactic Structures?

I saw a girl with a telescope

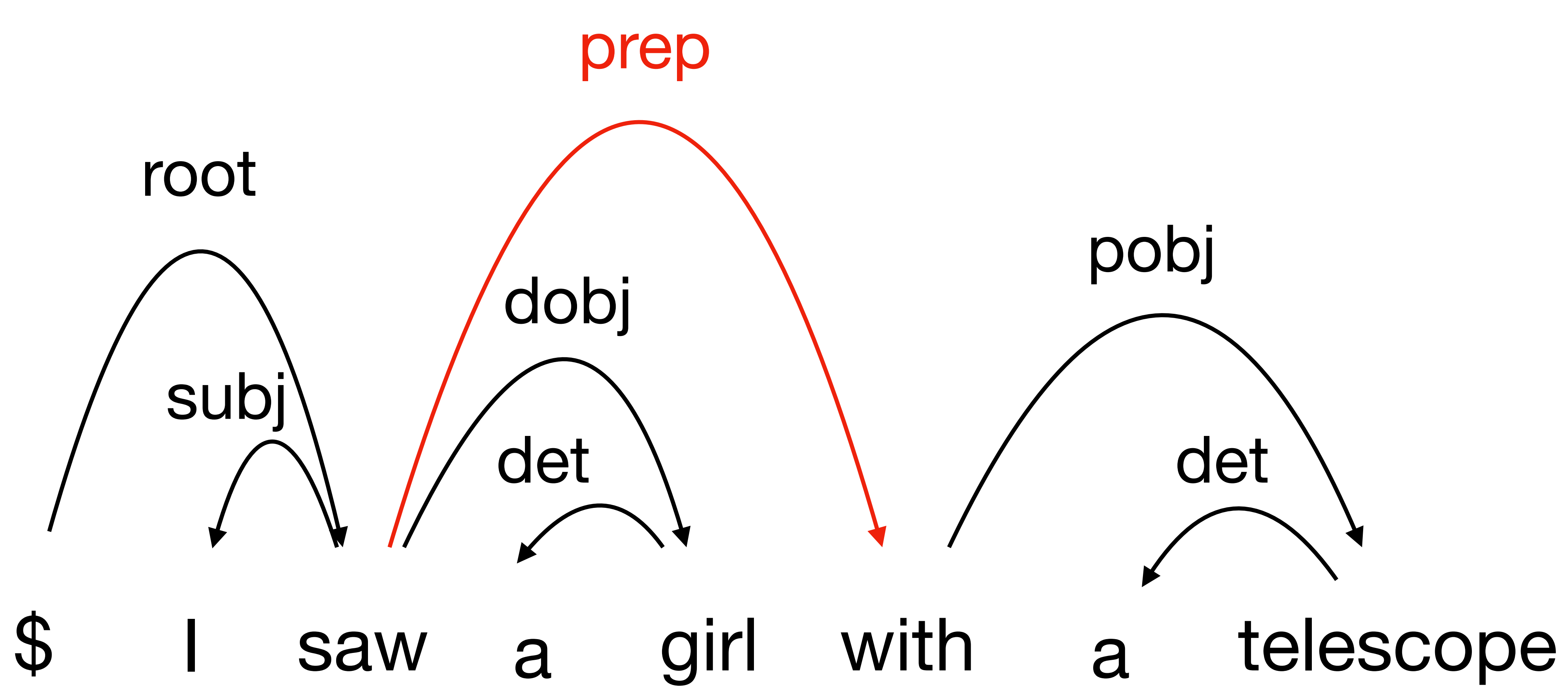


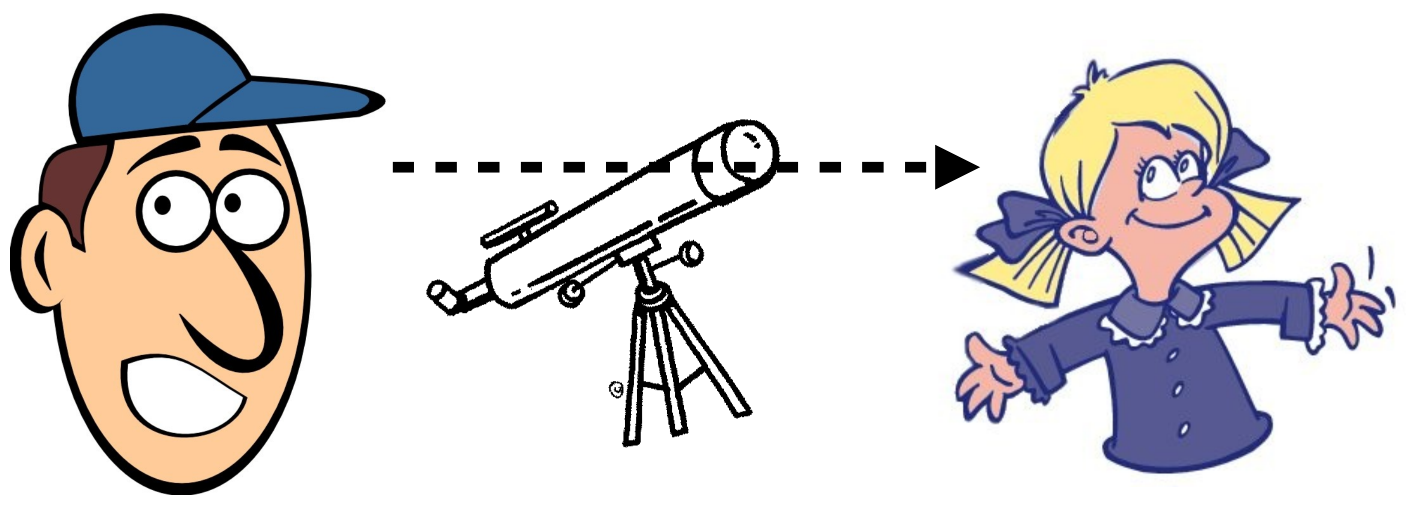


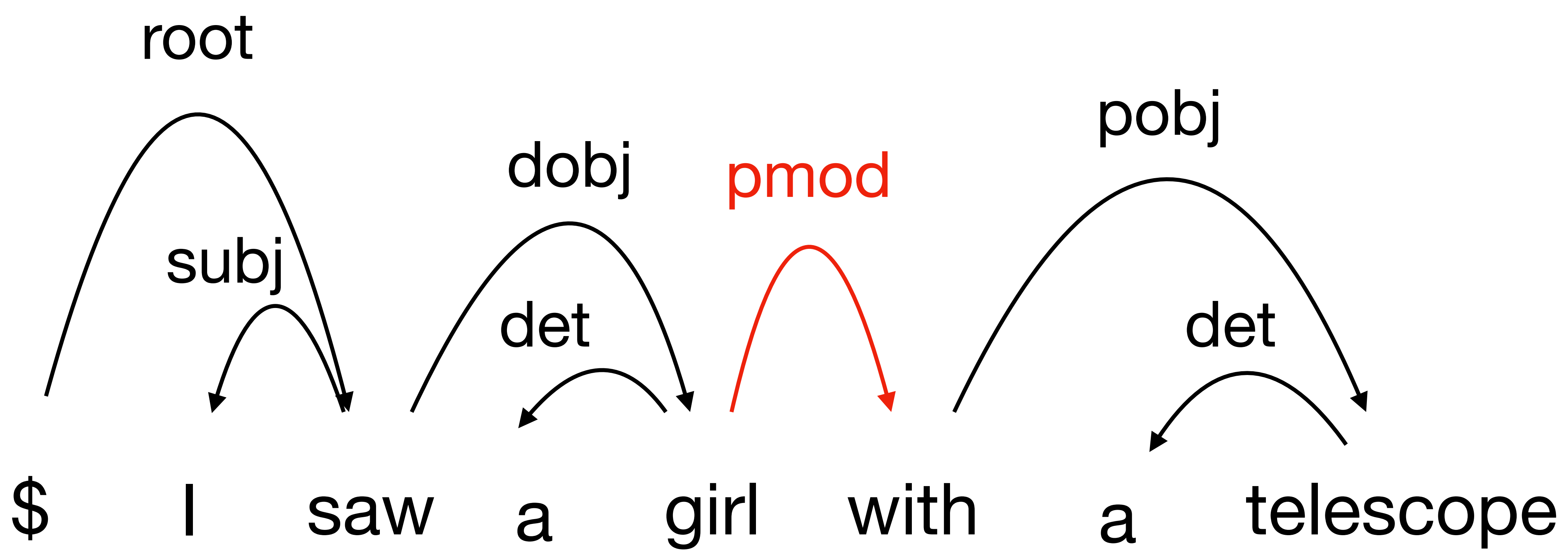
Why Syntactic Structures?

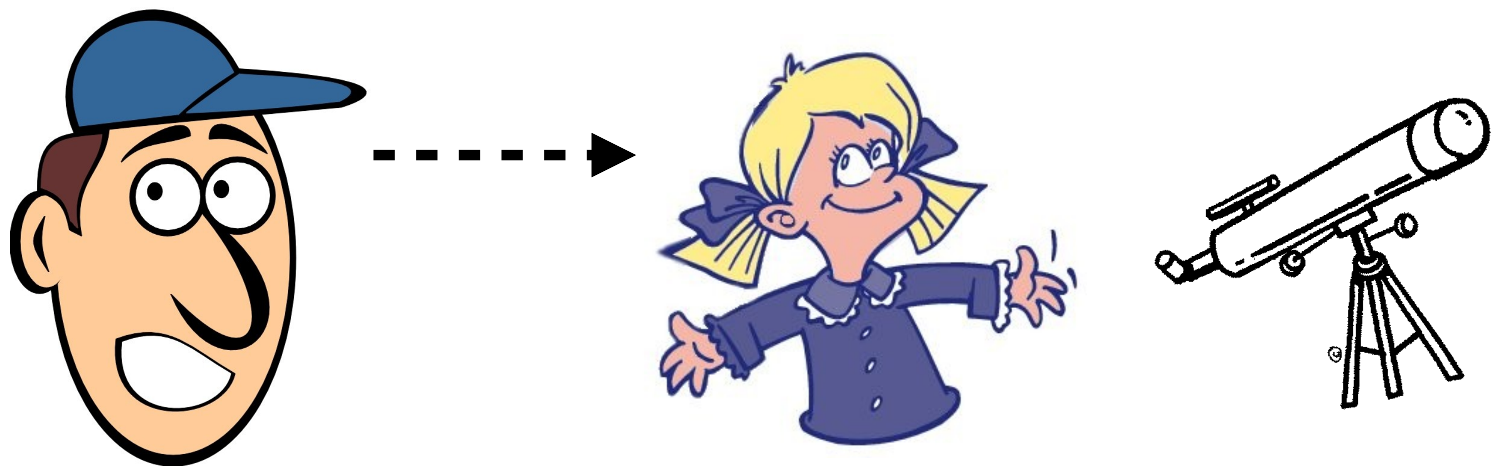


Syntactic Structures Resolve Ambiguity



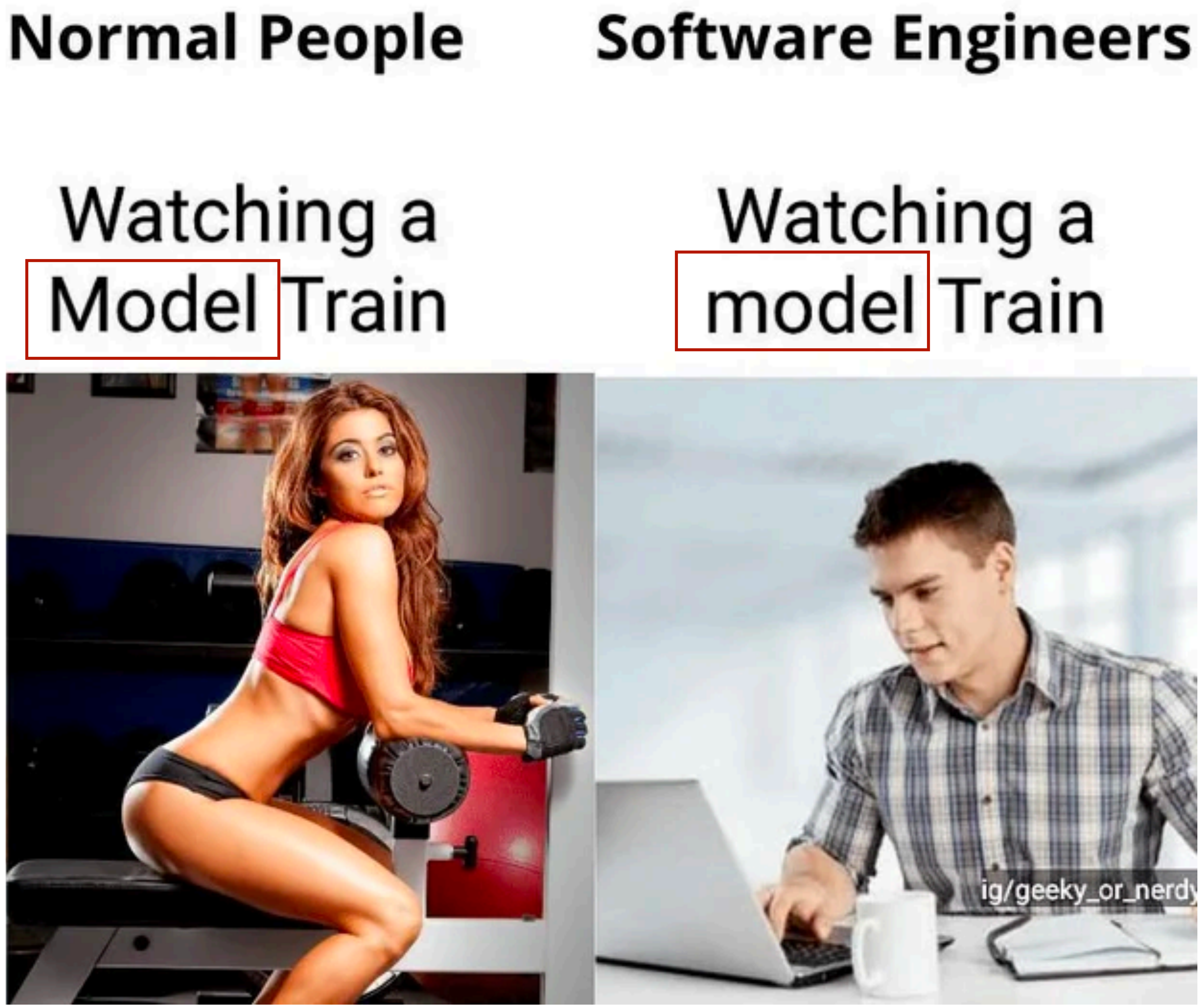






Limitation of Syntactic Structures

Syntax structures cannot resolve semantic ambiguities



The same syntactic structures
Different semantic meaning of "*model*"

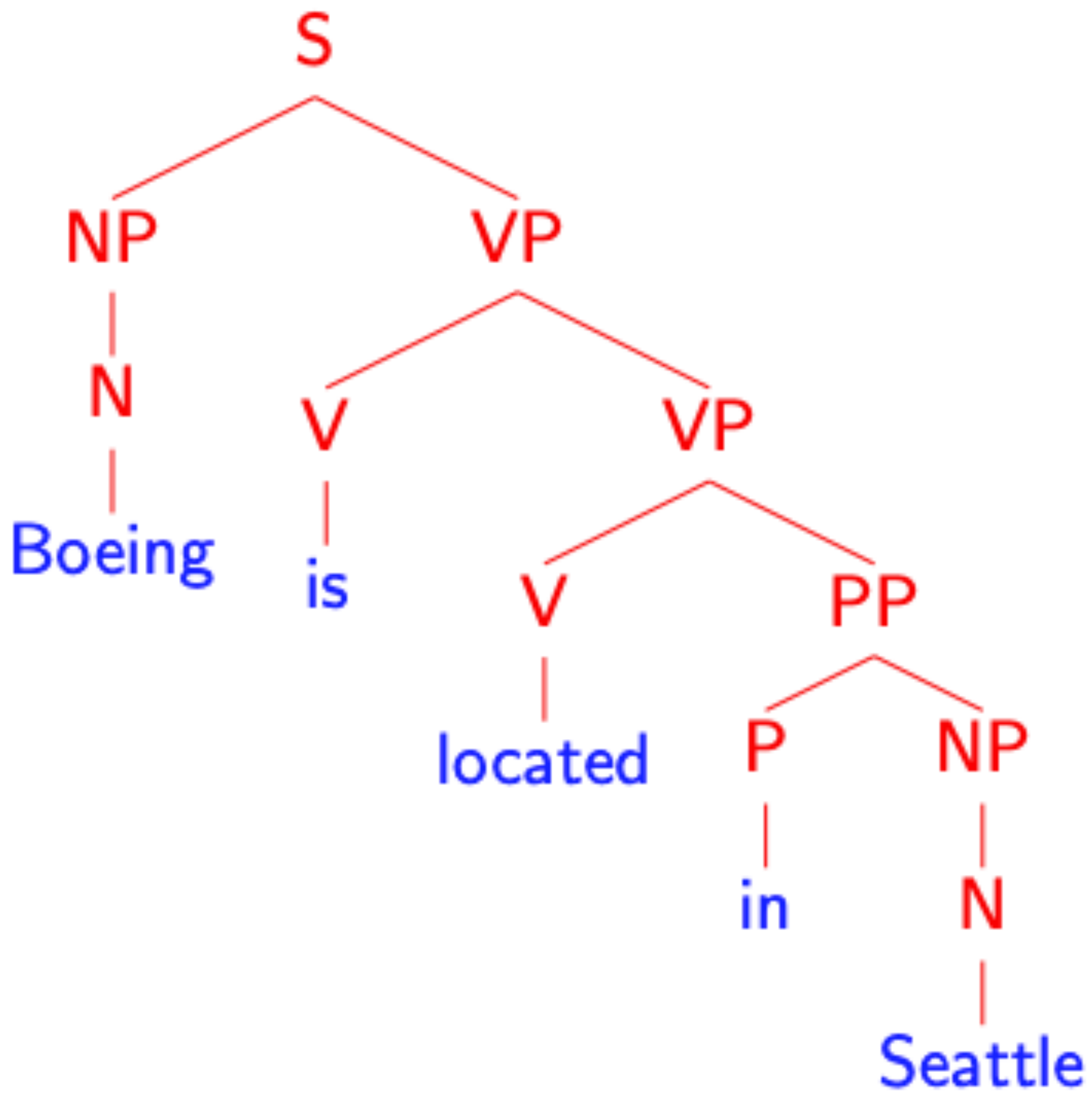
Syntactic Tree Parsing

A type of structured prediction tasks

Input X: a sentence

Boeing is located in Seattle.

Output Y: a parsing tree



Slide credit: Michael Collins COMS W4705

Syntactic Formalisms

- **Work in formal syntax goes back to Chomsky's PhD thesis in the 1950s**
- **Examples of current formalisms:**
  - Minimalism
  - Lexical Function Grammar (LFG)
  - Head-driven Phrase-structure Grammar (HPSG)
  - Tree Adjoining Grammar (TAG)
  - Categorial Grammars
  - … …

Slide credit: Michael Collins COMS W4705

Syntactic Parsing: Application

- **Grammar Checking**
  - If a sentence cannot be parsed, it may have grammatical errors (or at least hard to read)

- **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*

Slide credit: Michael Collins COMS W4705

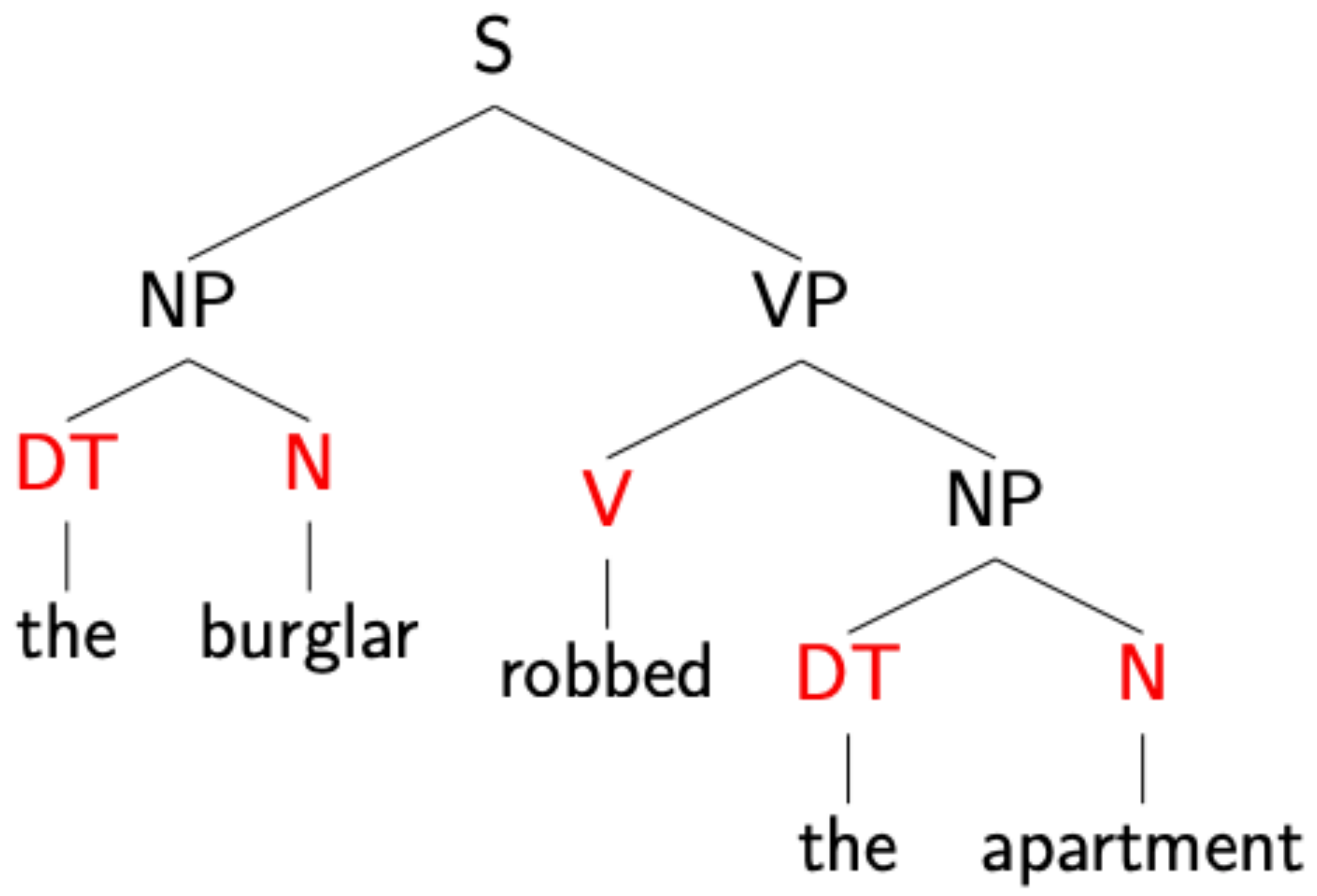
Overview

- **Constituency Parsing**
  - Constituency Structure
  - Context-free Grammar (CFG) & Probabilistic Context-free Grammar (PCFG)
  - The CKY algorithm
  - Lexicalized PCFGs
- **Dependency Parsing**
  - Dependency Structure
  - Graph-based Dependency Parsing
  - Transition-based Dependency Parsing

Constituency Parsing

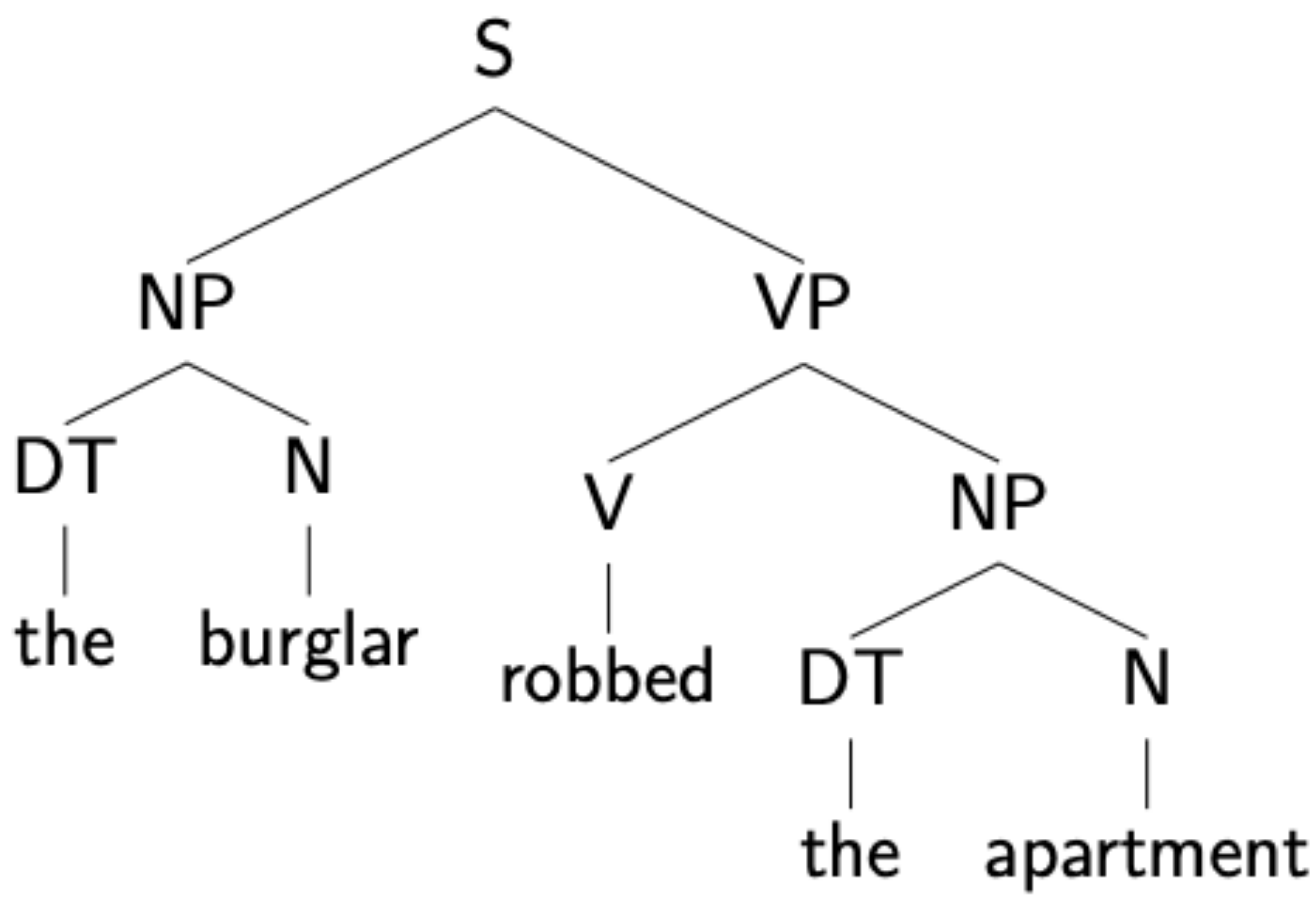
Constituency Structure

- **Starting units:** words are given a category: part-of-speech tags
  - N = noun, V = verb, DT = determiner



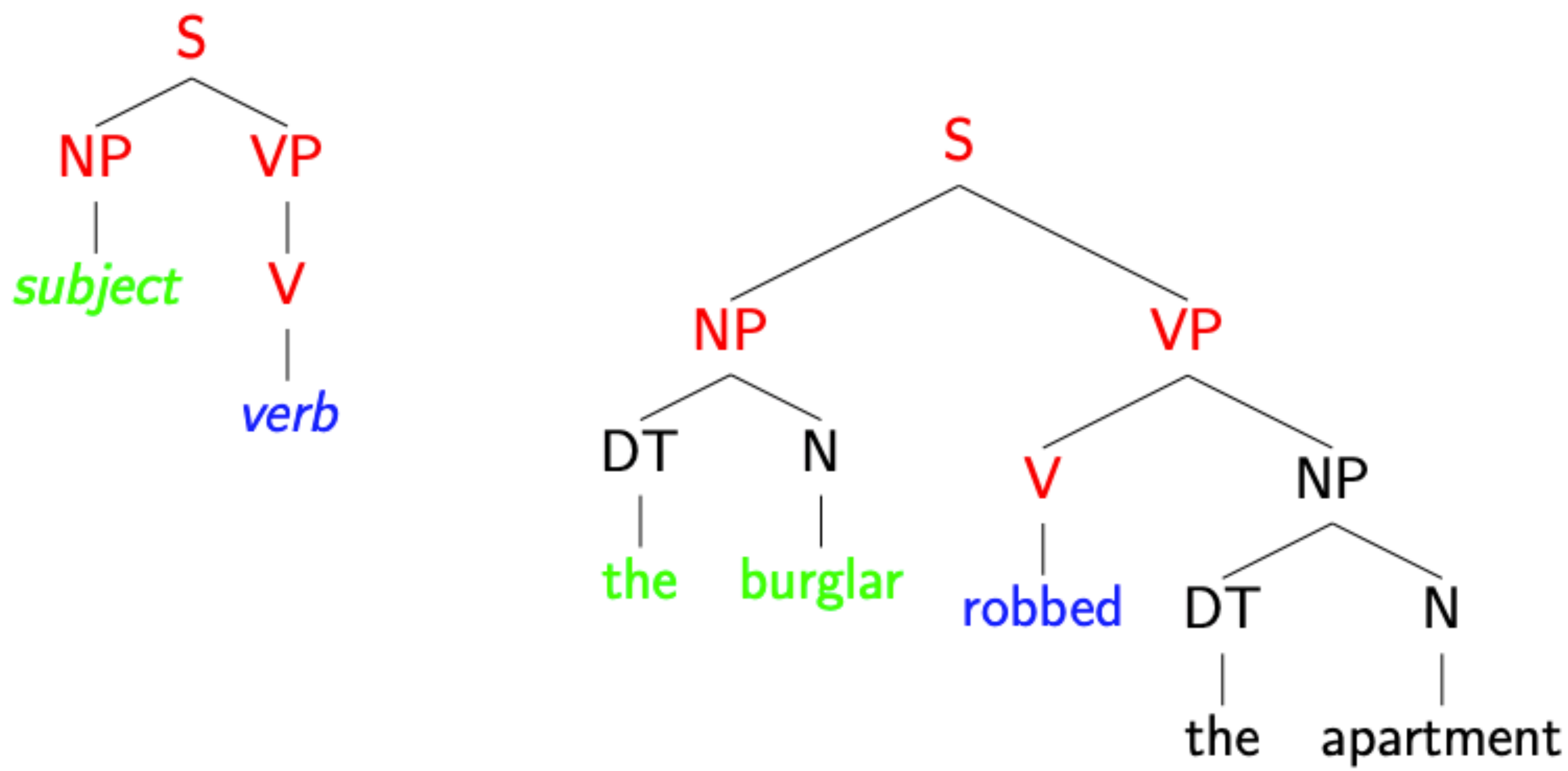
Constituency Structure

- **Starting units:** words are given a category: part-of-speech tags
  - N = noun, V = verb, DT = determiner
- **Phrases:** words combine into phrases with categories
  - NP = noun phrase, VP = verb phrase, S = sentence
  - Phrases can combine into bigger phrases recursively



Constituency Structure

- **Useful Relationships**

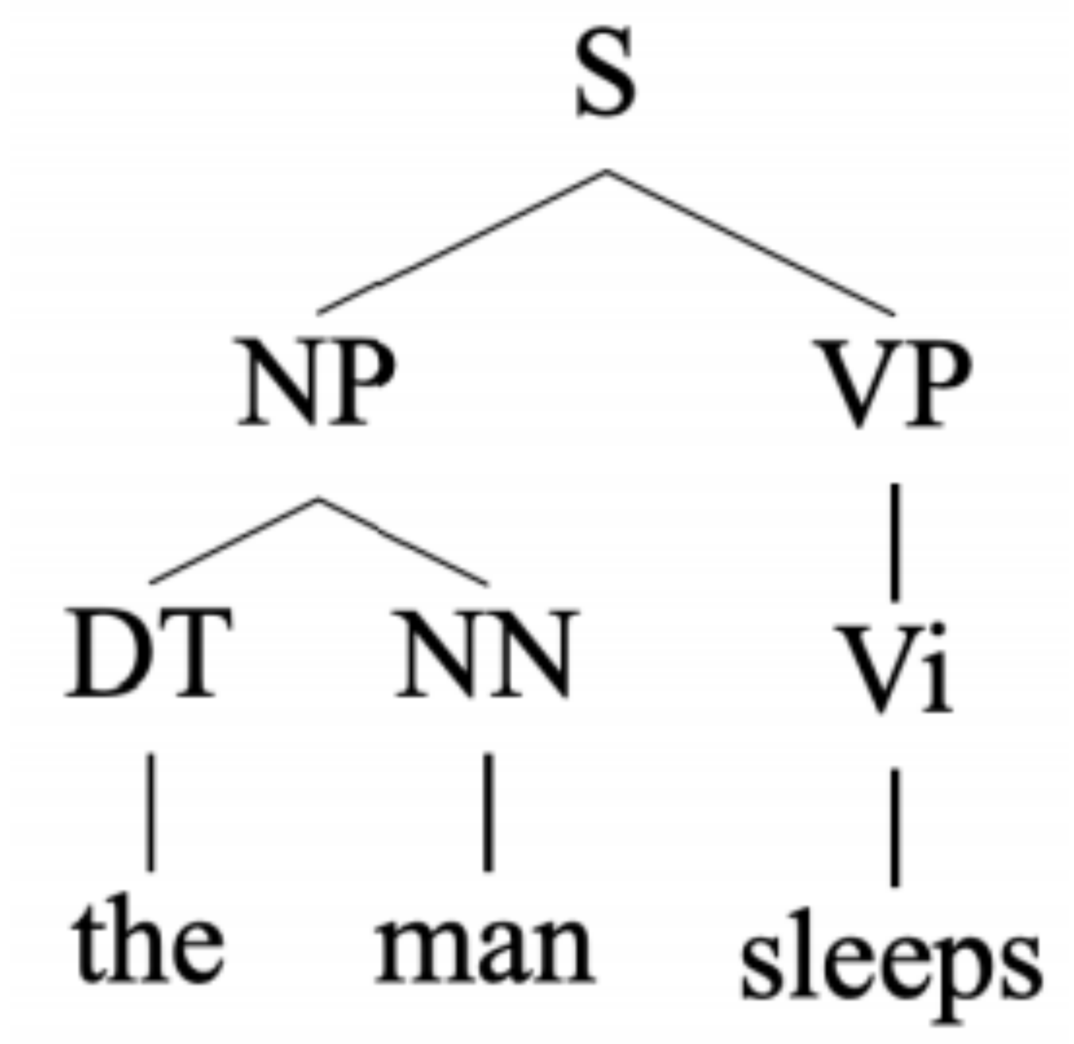


$\Rightarrow$ "the burglar" is the subject of "robbed"

Context-free Grammar (CFG)

Context-free Grammars (CFGs)

- **The most widely used formal system for modeling constituency structure in English and other natural languages**
  - Hopcroft and Ullman, 1979
- **A context free grammar (CFG) $G = (N, \Sigma, R, S)$ where:**
  - N is a set of non-terminal symbols
    - ✦ Phrasal categories: S, NP, VP, …
    - ✦ Part-of-speech: DT, NN, Vi, … (pre-terminals)
  - Σ is a set of terminal symbols: the, man, sleeps, …
  - R is a set of rules of the form $X \rightarrow Y_1 Y_2 \ldots Y_n$, for $n \geq 0$, $X \in N, Y_i \in (N \cup \Sigma)$
    - ✦ Examples: S -> NP VP, NP -> DT NN, NN -> man
  - $S \in N$ 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

Grammar

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

Lexicon

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

Slide credit: Michael Collins COMS W4705

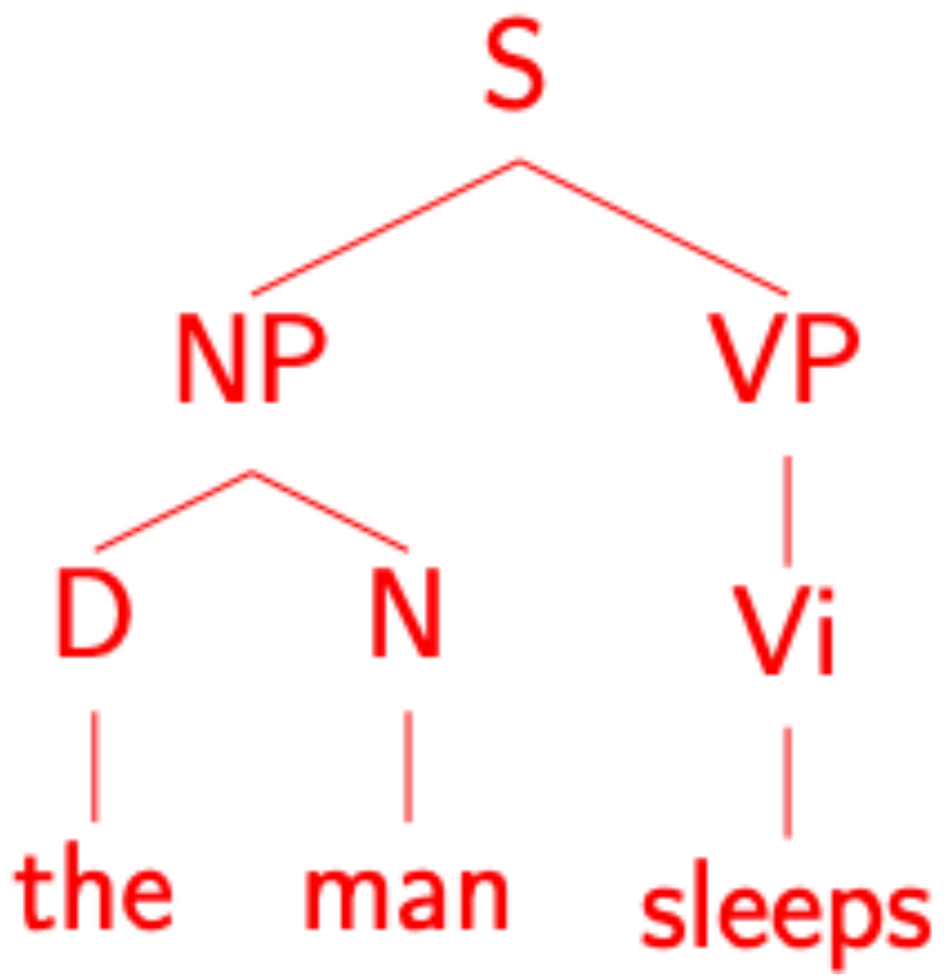
CFG: Left-Most Derivations

A left-most derivation is a sequence of strings $s_1 \ldots s_n$, 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 $i = 2 \ldots n$ 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

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:



Slide credit: Michael Collins COMS W4705

CFG: An Example

DERIVATION	RULES USED
S	

CFG: An Example

DERIVATION	RULES USED
S	
	S $\rightarrow$ NP VP
NP VP	

CFG: An Example

DERIVATION	RULES USED
S	
	S $\rightarrow$ NP VP
NP VP	
	NP $\rightarrow$ DT N
DT N VP	

CFG: An Example

DERIVATION	RULES USED
S	
	S $\rightarrow$ NP VP
NP VP	
	NP $\rightarrow$ DT N
DT N VP	
	DT $\rightarrow$ the
the N VP	

CFG: An Example

DERIVATION	RULES USED
S	
	S $\rightarrow$ NP VP
NP VP	
	NP $\rightarrow$ DT N
DT N VP	
	DT $\rightarrow$ the
the N VP	
	N $\rightarrow$ dog
the dog VP	

CFG: An Example

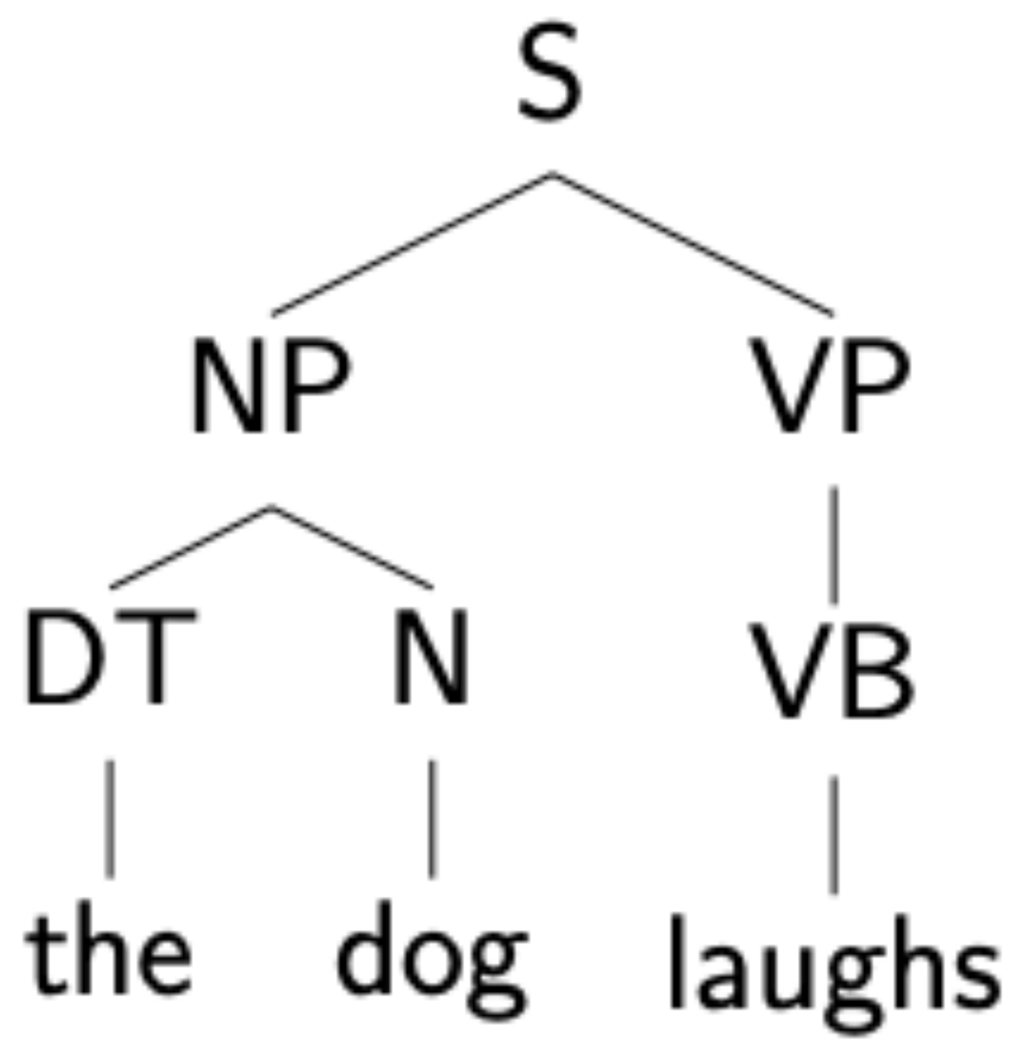
DERIVATION	RULES USED
S	
	S $\rightarrow$ NP VP
NP VP	
	NP $\rightarrow$ DT N
DT N VP	
	DT $\rightarrow$ the
the N VP	
	N $\rightarrow$ dog
the dog VP	
	VP $\rightarrow$ VB
the dog VB	

CFG: An Example

DERIVATION	RULES USED
S	$S \rightarrow NP\ VP$
NP VP	$NP \rightarrow DT\ N$
DT N VP	$DT \rightarrow$ the
the N VP	$N \rightarrow$ dog
the dog VP	$VP \rightarrow VB$
the dog VB	$VB \rightarrow$ laughs
the dog laughs	

CFG: An Example

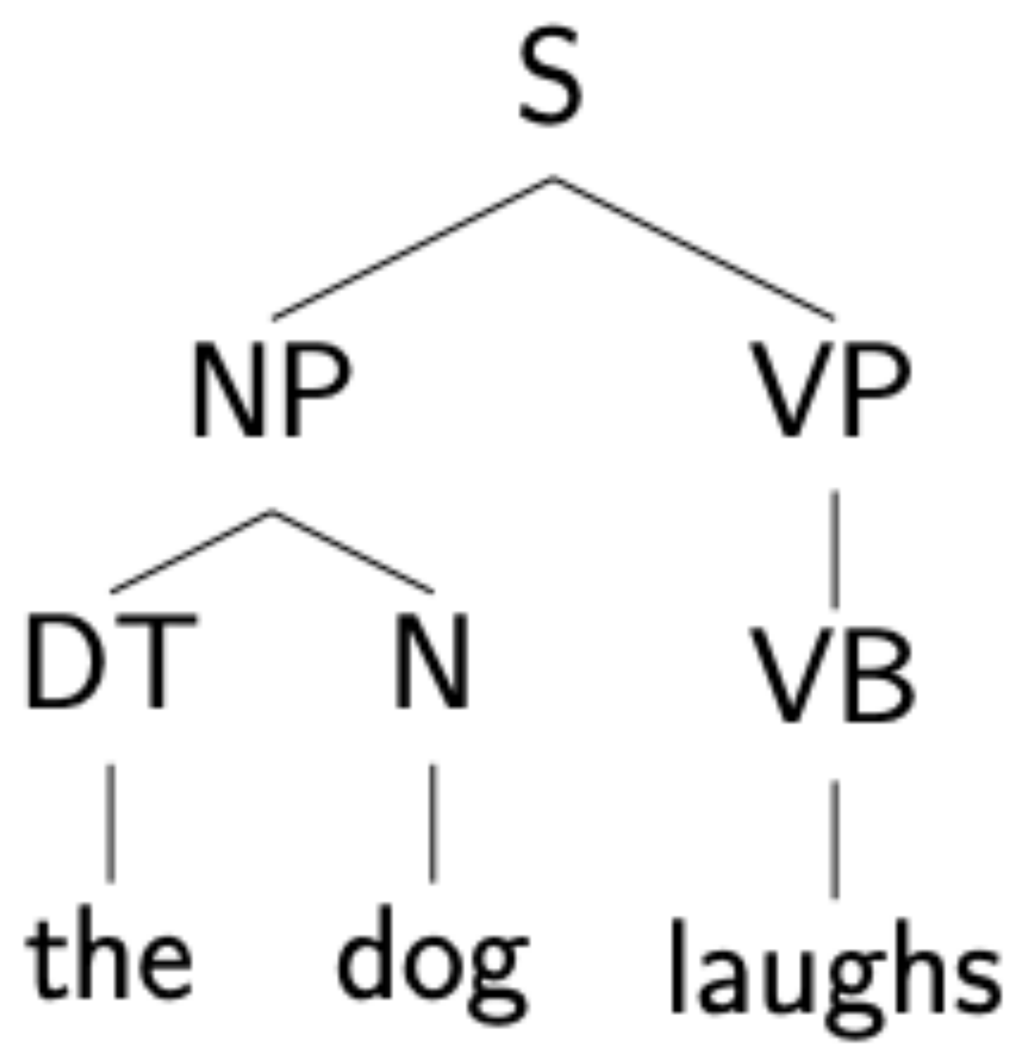
DERIVATION	RULES USED
S	S $\rightarrow$ NP VP
NP VP	NP $\rightarrow$ DT N
DT N VP	DT $\rightarrow$ the
the N VP	N $\rightarrow$ dog
the dog VP	VP $\rightarrow$ VB
the dog VB	VB $\rightarrow$ laughs
the dog laughs	



A derivation can be represented as a constituency tree

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")



A derivation can be represented as a constituency tree

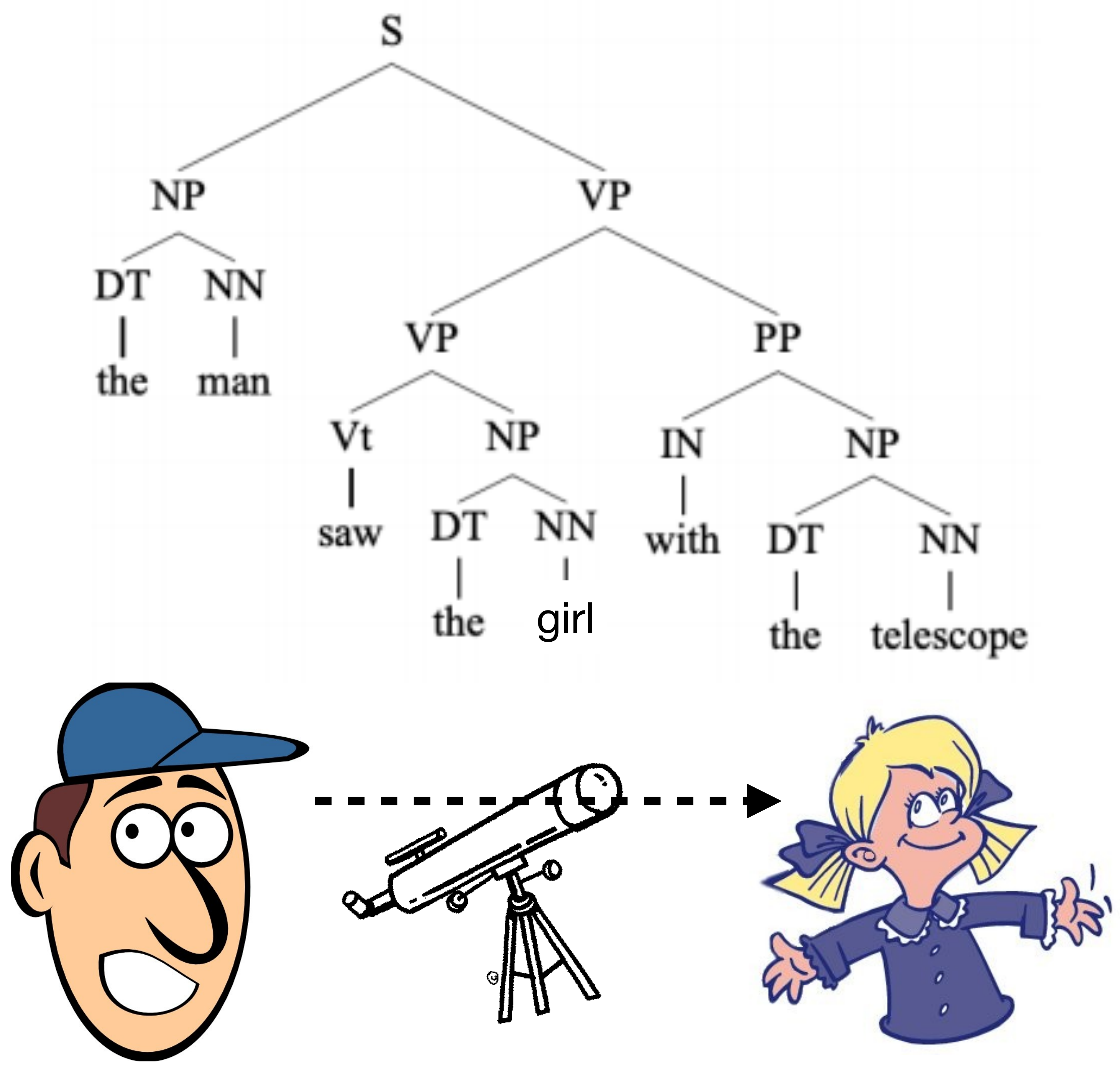
A derivation -> a sentence

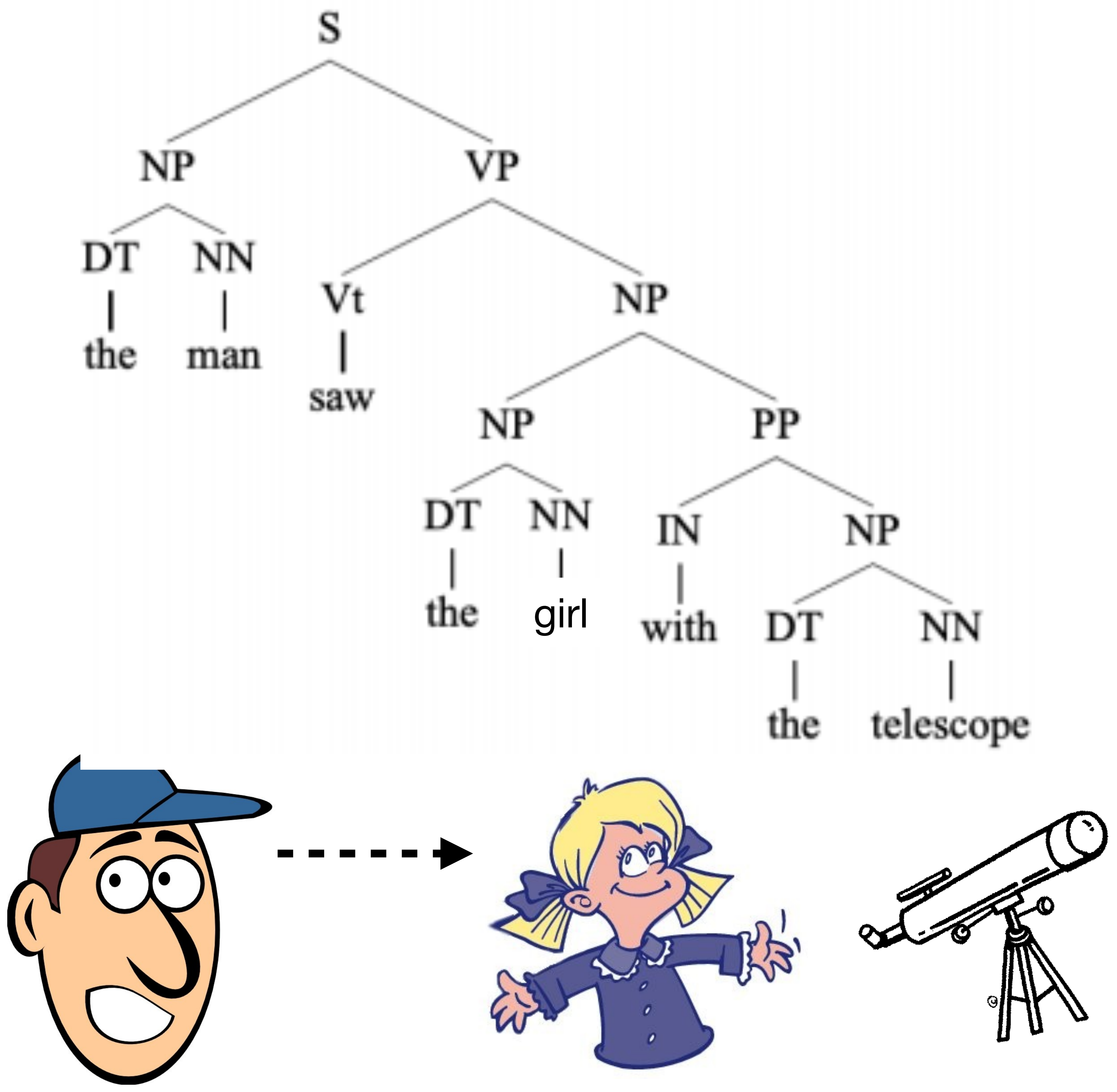
A sentence -> multiple derivations

Slide credit: Michael Collins COMS W4705

The Problem with Parsing: Ambiguity

The man saw the girl with the telescope

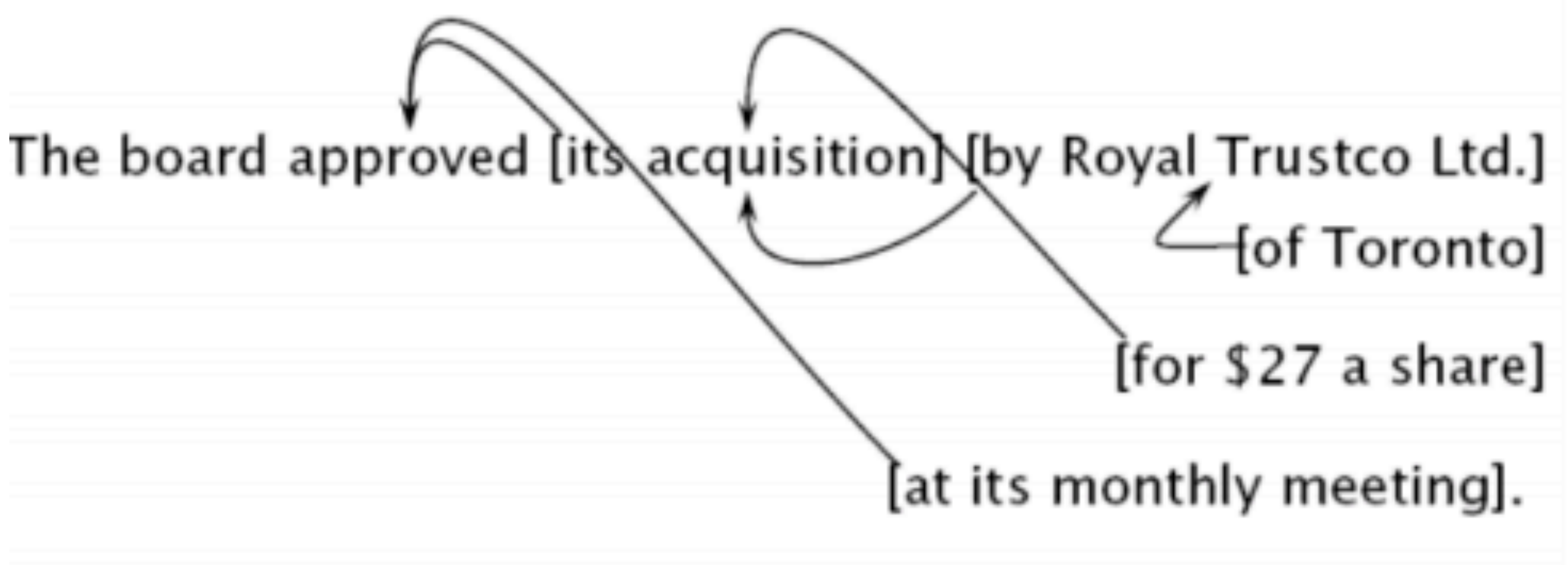




The Problem with Parsing: Ambiguity

- In fact, a sentences can have a very large number of possible parses

The board approved [its acquisition] [by Royal Trustco Ltd.] [of Toronto] [for $27 a share] [at its monthly meeting].



((ab)c)d   (a(bc))d   (ab)(cd)   a((bc)d)   a(b(cd))

Catalan number: $C_n = \frac{1}{n+1}\binom{2n}{n}$

- There is no way to choose the right one!
- Constructing a grammar is difficult — a less constrained grammar can parse more sentences but result in more parses for even simple sentences

Since it is hard to avoid ambiguity in grammars, is it possible to model ambiguity?

Slide credit: Danqi Chen COS 484/584

Probabilistic Context-free Grammars (PCFGs)

Probabilistic Context-free Grammars (PCFGs)

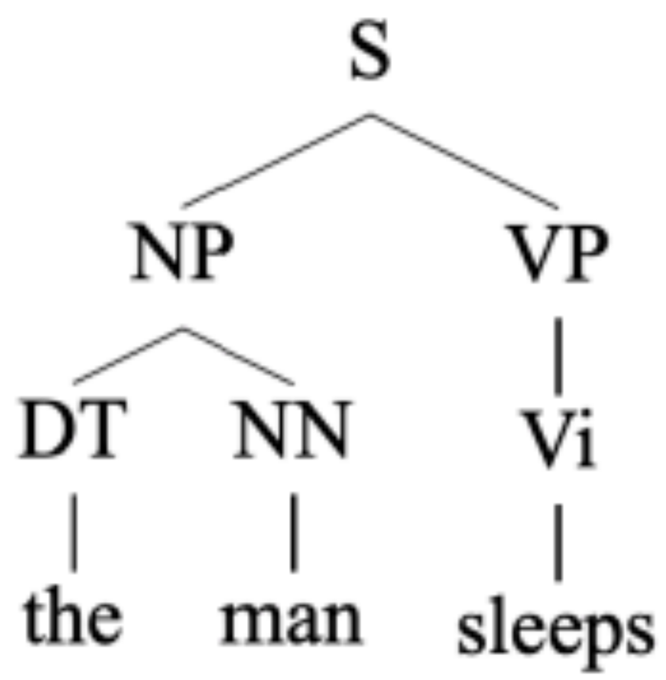
- **A PCFG consists of**
  - A context free grammar (CFG) $G = (N, \Sigma, R, S)$
  - For each rule $X \to Y$, there is a parameter $q(X \to Y) \geq 0$:

$$\sum_{Y \in N \cup \Sigma} q(X \to Y) = 1$$

  - For any derivation (parse tree) containing rules: $X_1 \to Y_1, \ldots, X_n \to Y_n$

$$P(t) = \prod_{i=1}^{n} q(X_i \to Y_i)$$

PCFG: An Example



$R, q =$

S	$\rightarrow$	NP	VP	1.0
VP	$\rightarrow$	Vi		0.3
VP	$\rightarrow$	Vt	NP	0.5
VP	$\rightarrow$	VP	PP	0.2
NP	$\rightarrow$	DT	NN	0.8
NP	$\rightarrow$	NP	PP	0.2
PP	$\rightarrow$	IN	NP	1.0

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

$$P(t) = q(\text{S} \rightarrow \text{NP VP}) \times q(\text{NP} \rightarrow \text{DT NN}) \times q(\text{DT} \rightarrow \text{the})$$
$$\times q(\text{NN} \rightarrow \text{man}) \times q(\text{VP} \rightarrow \text{Vi}) \times q(\text{Vi} \rightarrow \text{sleeps})$$

$$= 1.0 \times 0.8 \times 1.0 \times 0.1 \times 0.3 \times 1.0 = 0.024$$

Q: Why do we want $\sum_{Y \in N \cup \Sigma} q(X \rightarrow Y) = 1$?

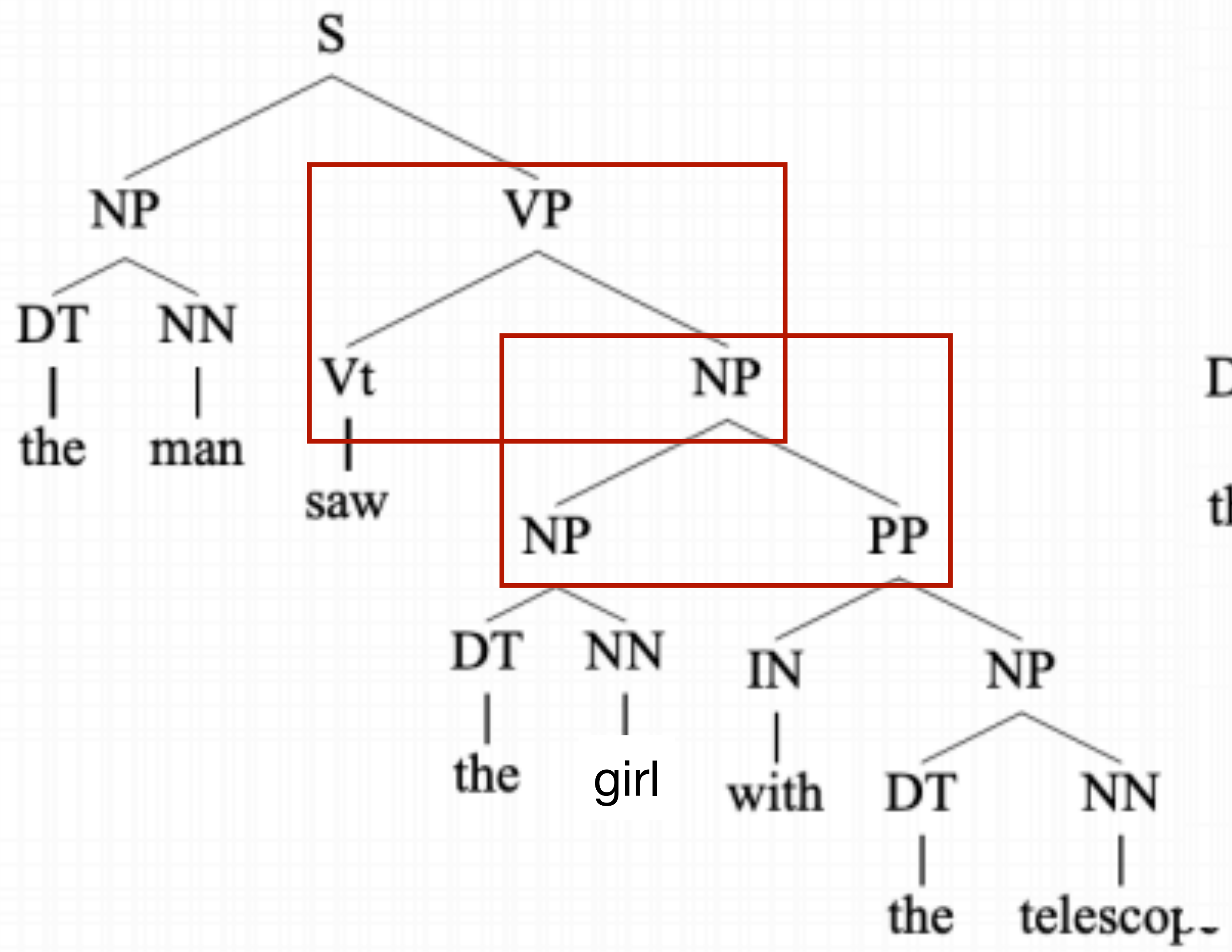
Slide credit: Danqi Chen COS 484/584

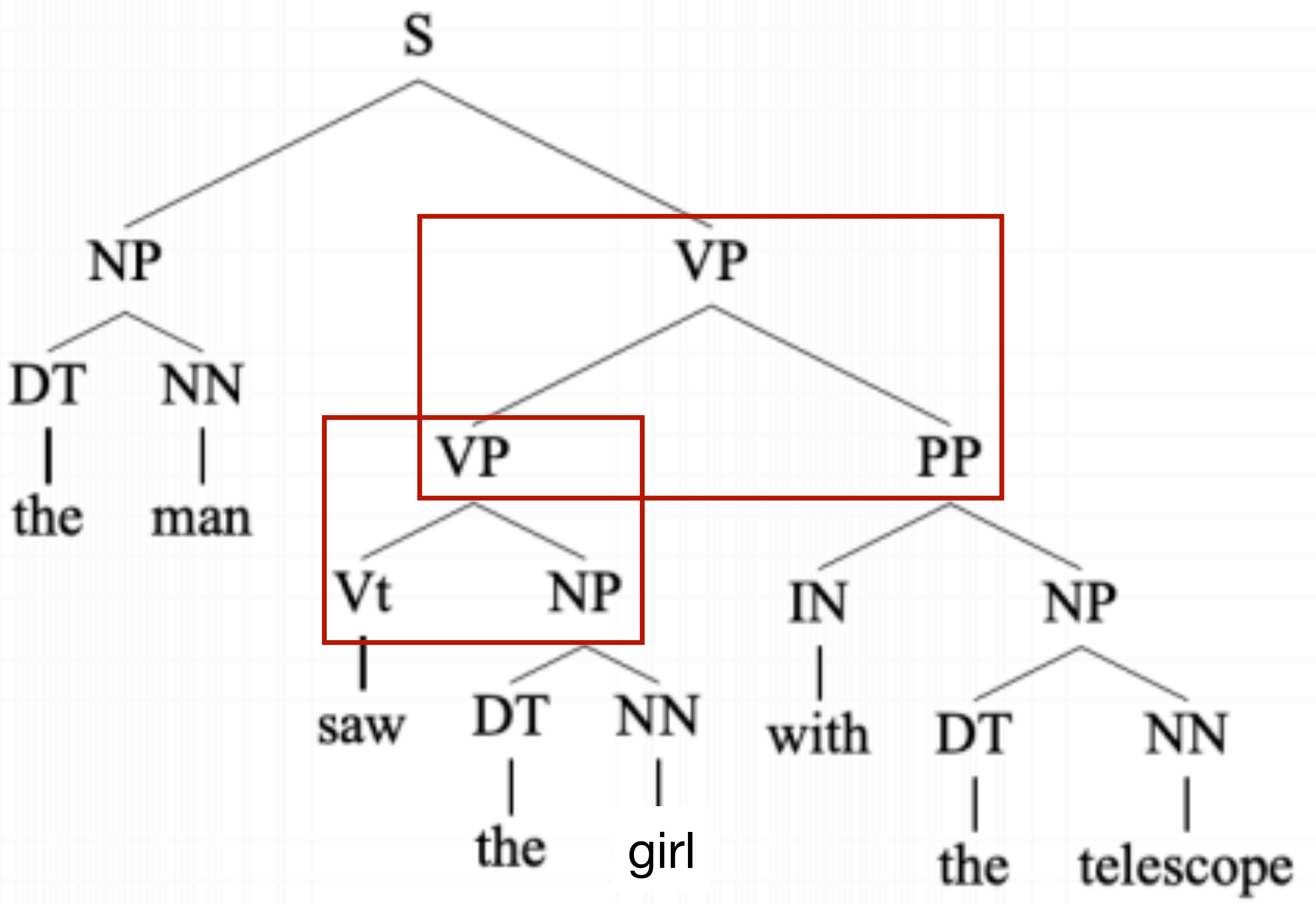
Handling Ambiguity with Probability

$R, q =$

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	→	girl	0.5
DT	→	the	1.0
IN	→	with	0.6
IN	→	in	0.4





$$q(\text{VP} \rightarrow \text{Vt NP}) \times q(\text{NP} \rightarrow \text{NP PP}) = 0.5 \times 0.2 = 0.1$$

$$q(\text{VP} \rightarrow \text{VP PP}) \times q(\text{VP} \rightarrow \text{Vt NP}) = 0.2 \times 0.5 = 0.1$$

Slide credit: Danqi Chen COS 484/584

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

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

The Rise of Annotated Data

- **Learning from data**: treebanks
- **Adding probabilities to the rules**: probabilistic CFGs

Treebanks: a collection of sentences paired with their annotated parse trees

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

(a)

```
((S
   (NP-SBJ The/DT flight/NN )
   (VP should/MD
     (VP arrive/VB
       (PP-TMP at/IN
         (NP eleven/CD a.m/RB ))
       (NP-TMP tomorrow/NN )))))
```

(b)

The Penn Treebank Project (Marcus et al, 1993)

Slide credit: Danqi Chen COS 484/584

Penn Treebank

Standard setup

- 40,000 sentences for training
- 1,700 for development
- 2,400 for testing

Phrasal categories

ADJP	Adjective phrase
ADVP	Adverb phrase
NP	Noun phrase
PP	Prepositional phrase
S	Simple declarative clause
SBAR	Subordinate clause
SBARQ	Direct question introduced by *wh*-element
SINV	Declarative sentence with subject-aux inversion
SQ	Yes/no questions and subconstituent of SBARQ excluding *wh*-element
VP	Verb phrase
WHADVP	Wh-adverb phrase
WHNP	Wh-noun phrase
WHPP	Wh-prepositional phrase
X	Constituent of unknown or uncertain category
*	"Understood" subject of infinitive or imperative
0	Zero variant of *that* in subordinate clauses
T	Trace of wh-Constituent

Slide credit: Danqi Chen COS 484/584

Penn Treebank

Part-of-speech tagset

CC	Coordinating conj.	TO	infinitival *to*
CD	Cardinal number	UH	Interjection
DT	Determiner	VB	Verb, base form
EX	Existential there	VBD	Verb, past tense
FW	Foreign word	VBG	Verb, gerund/present pple
IN	Preposition	VBN	Verb, past participle
JJ	Adjective	VBP	Verb, non-3rd ps. sg. present
JJR	Adjective, comparative	VBZ	Verb, 3rd ps. sg. present
JJS	Adjective, superlative	WDT	Wh-determiner
LS	List item marker	WP	Wh-pronoun
MD	Modal	WP$	Possessive *wh*-pronoun
NN	Noun, singular or mass	WRB	Wh-adverb
NNS	Noun, plural	#	Pound sign
NNP	Proper noun, singular	$	Dollar sign
NNPS	Proper noun, plural	.	Sentence-final punctuation
PDT	Predeterminer	,	Comma
POS	Possessive ending	:	Colon, semi-colon
PRP	Personal pronoun	(	Left bracket character
PP$	Possessive pronoun	)	Right bracket character
RB	Adverb	"	Straight double quote
RBR	Adverb, comparative	‘	Left open single quote
RBS	Adverb, superlative	“	Left open double quote
RP	Particle	’	Right close single quote
SYM	Symbol	”	Right close double quote

Slide credit: Danqi Chen COS 484/584

Learning 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 we have seen the rule VP $\rightarrow$ Vt NP 105 times, and the the non-terminal VP 1000 times, $q(\text{VP} \rightarrow \text{Vt NP}) = 0.105$

Slide credit: Michael Collins COMS W4705

Learning a PCFG: An Example

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

(a)

```
((S
  (NP-SBJ The/DT flight/NN )
  (VP should/MD
    (VP arrive/VB
      (PP-TMP at/IN
        (NP eleven/CD a.m/RB ))
      (NP-TMP tomorrow/NN )))))
```

(b)

```
( (S (`` ``)
    (S-TPC-2
      (NP-SBJ-1 (PRP We) )
      (VP (MD would)
        (VP (VB have)
          (S
            (NP-SBJ (-NONE- *-1) )
            (VP (TO to)
              (VP (VB wait)
                (SBAR-TMP (IN until)
                  (S
                    (NP-SBJ (PRP we) )
                    (VP (VBP have)
                      (VP (VBN collected)
                        (PP-CLR (IN on)
                          (NP (DT those)(NNS assets)))))))))))))
    (, ,) ('' '')
    (NP-SBJ (PRP he) )
    (VP (VBD said)
      (S (-NONE- *T*-2) ))
    (. .) ))
```

Slide credit: Danqi Chen COS 484/584

Learning a PCFG: An Example

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

(a)

```
((S
   (NP-SBJ The/DT flight/NN )
   (VP should/MD
     (VP arrive/VB
       (PP-TMP at/IN
         (NP eleven/CD a.m/RB ))
       (NP-TMP tomorrow/NN )))))
```

(b)

```
( (S (`` ``)
    (S-TPC-2
      (NP-SBJ-1 (PRP We) )
      (VP (MD would)
        (VP (VB have)
          (S
            (NP-SBJ (-NONE- *-1) )
```

Grammar	Lexicon
$S \rightarrow NP\ VP\ .$	$PRP \rightarrow we \mid he$
$S \rightarrow NP\ VP$	$DT \rightarrow the \mid that \mid those$
$S \rightarrow$ " S " , $NP\ VP\ .$	$JJ \rightarrow cold \mid empty \mid full$
$S \rightarrow$ -NONE-	$NN \rightarrow sky \mid fire \mid light \mid flight \mid tomorrow$
$NP \rightarrow DT\ NN$	$NNS \rightarrow assets$
$NP \rightarrow DT\ NNS$	$CC \rightarrow and$
$NP \rightarrow NN\ CC\ NN$	$IN \rightarrow of \mid at \mid until \mid on$
$NP \rightarrow CD\ RB$	$CD \rightarrow eleven$
$NP \rightarrow DT\ JJ\ ,\ JJ\ NN$	$RB \rightarrow a.m.$
$NP \rightarrow PRP$	$VB \rightarrow arrive \mid have \mid wait$
$NP \rightarrow$ -NONE-	$VBD \rightarrow was \mid said$
$VP \rightarrow MD\ VP$	$VBP \rightarrow have$
$VP \rightarrow VBD\ ADJP$	$VBN \rightarrow collected$
$VP \rightarrow VBD\ S$	$MD \rightarrow should \mid would$
$VP \rightarrow VBN\ PP$	$TO \rightarrow to$
$VP \rightarrow VB\ S$	
$VP \rightarrow VB\ SBAR$	
$VP \rightarrow VBP\ VP$	
$VP \rightarrow VBN\ PP$	
$VP \rightarrow TO\ VP$	
$SBAR \rightarrow IN\ S$	
$ADJP \rightarrow JJ\ PP$	
$PP \rightarrow IN\ NP$	

Parsing with PCFGs: CKY Algorithm

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

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

Slide credit: Michael Collins COMS W4705

Chomsky Normal Form (CNF)

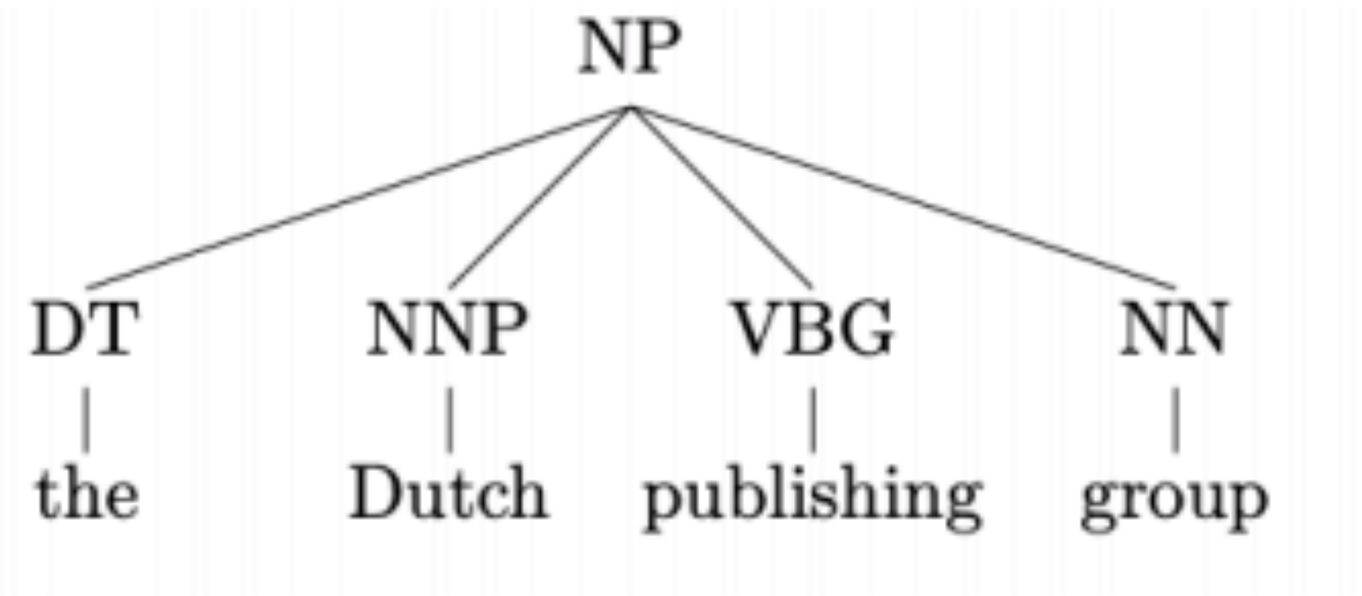
A context free grammar $G = (N, \Sigma, R, S)$ in Chomsky Normal Form is as follows

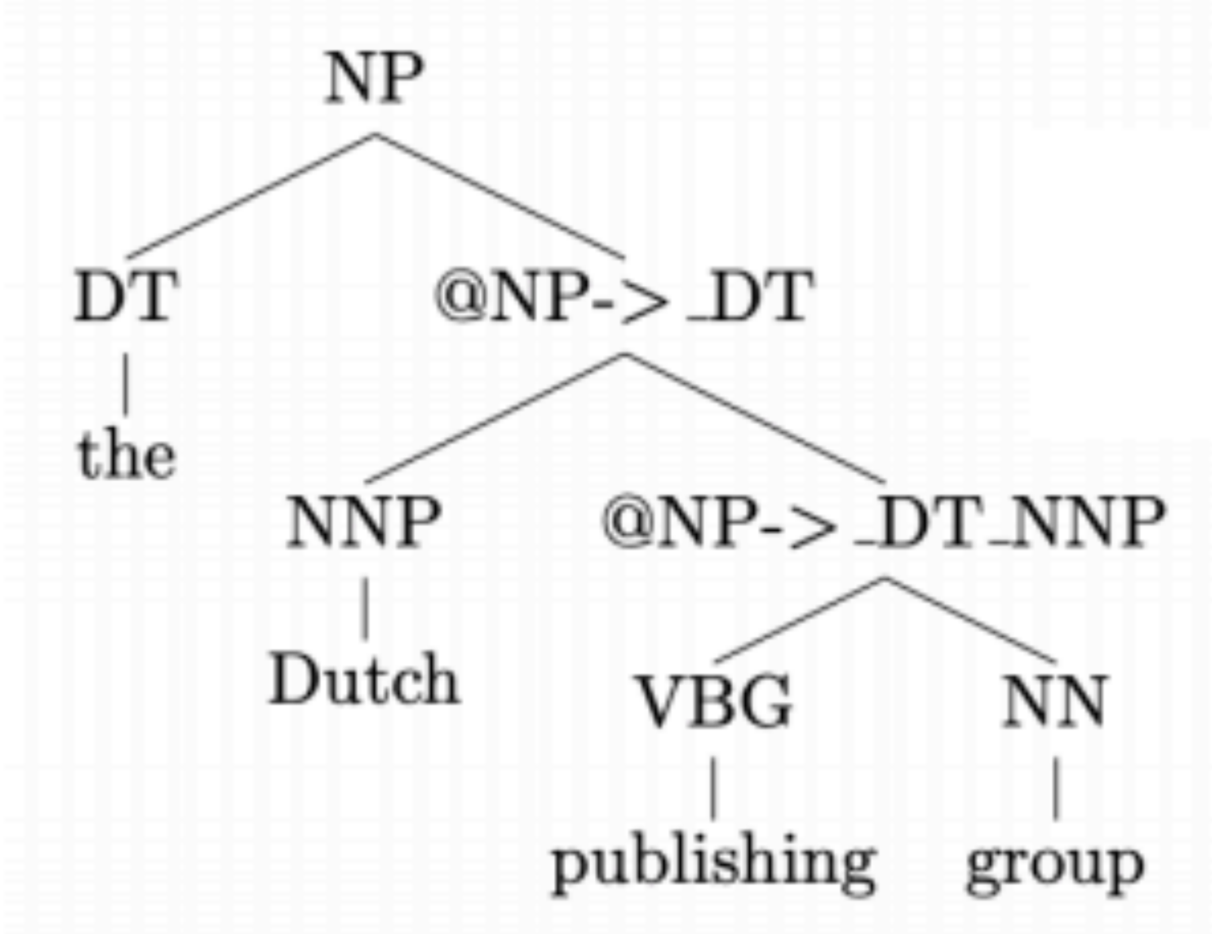
- N is a set of non-terminal symbols
- Σ is a set of terminal symbols
- R is a set of rules which take one of two forms:
  - $X \rightarrow Y_1 Y_2$ for $X \in N$, and $Y_1, Y_2 \in N$
  - $X \rightarrow Y$ for $X \in N$, and $Y \in \Sigma$
- $S \in N$ is a distinguished start symbol

Slide credit: Michael Collins COMS W4705

Converting PCFGs into CNFs

- n-ary rules ($n > 2$): NP $\rightarrow$ DT NNP VBG NN





The CKY Algorithm

- Notation:

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

- Define a dynamic programming table

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

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

The CKY Algorithm

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

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

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

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

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

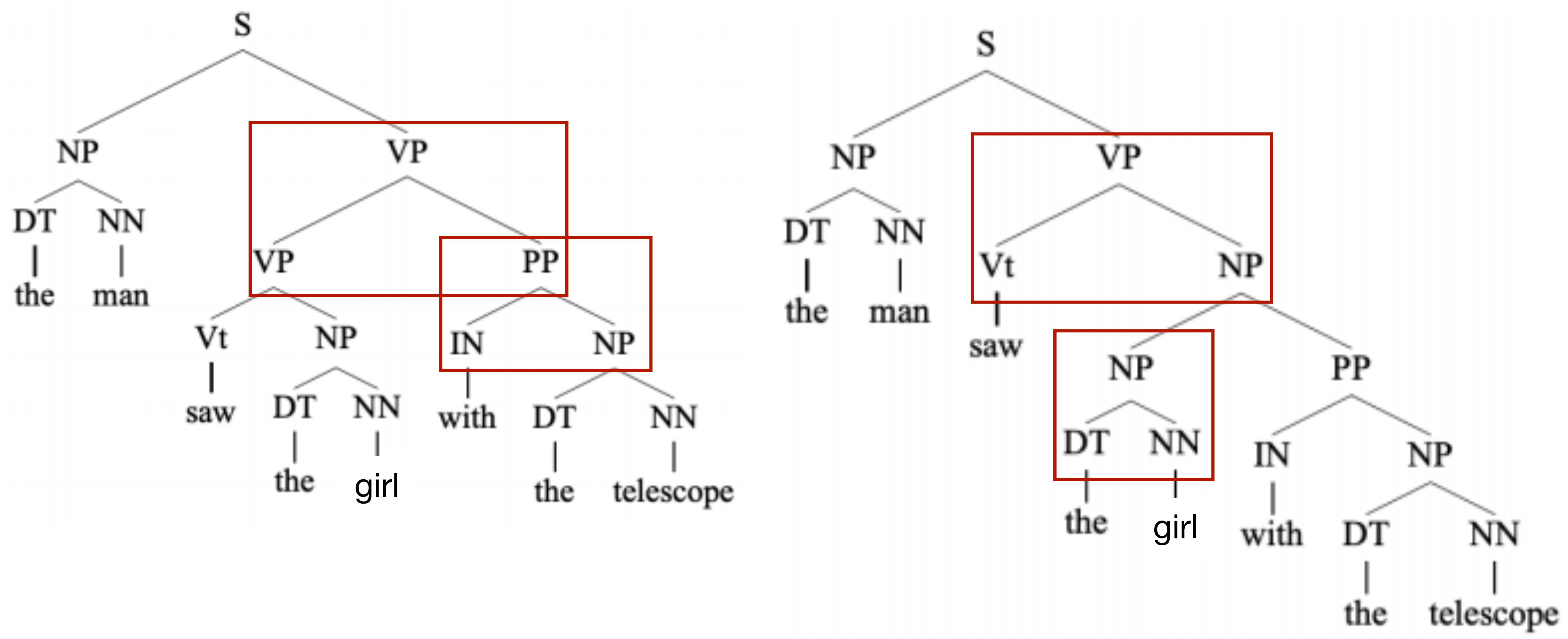
Q: Running time?

$O(n^3|R|)$

Weaknesses of PCFGs

- **Lack of sensitivity to lexical information (words)**

The man saw the girl with the telescope



The only difference between these two parses:

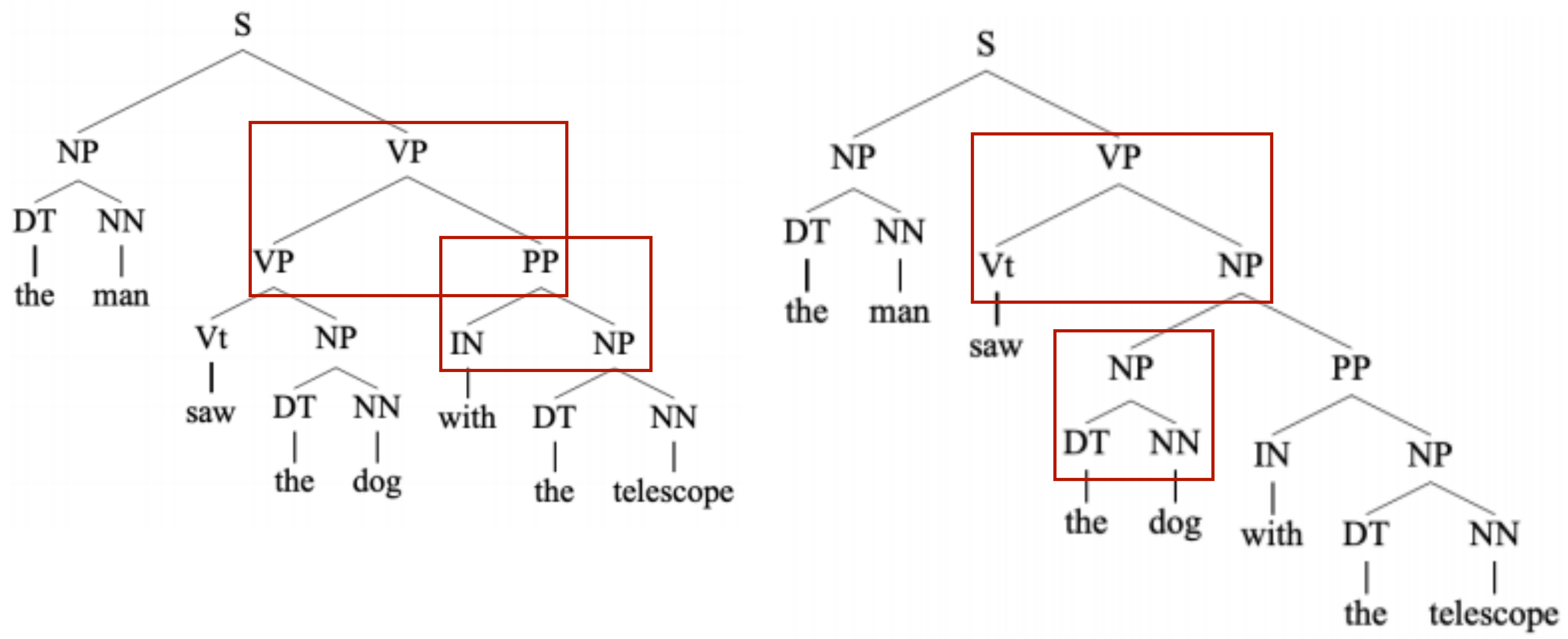
$q(\text{VP} \rightarrow \text{VP PP})$ vs $q(\text{NP} \rightarrow \text{NP PP})$

… without looking at the words!

Weaknesses of PCFGs

- **Lack of sensitivity to lexical information (words)**

The man saw the dog with the telescope



The only difference between these two parses:

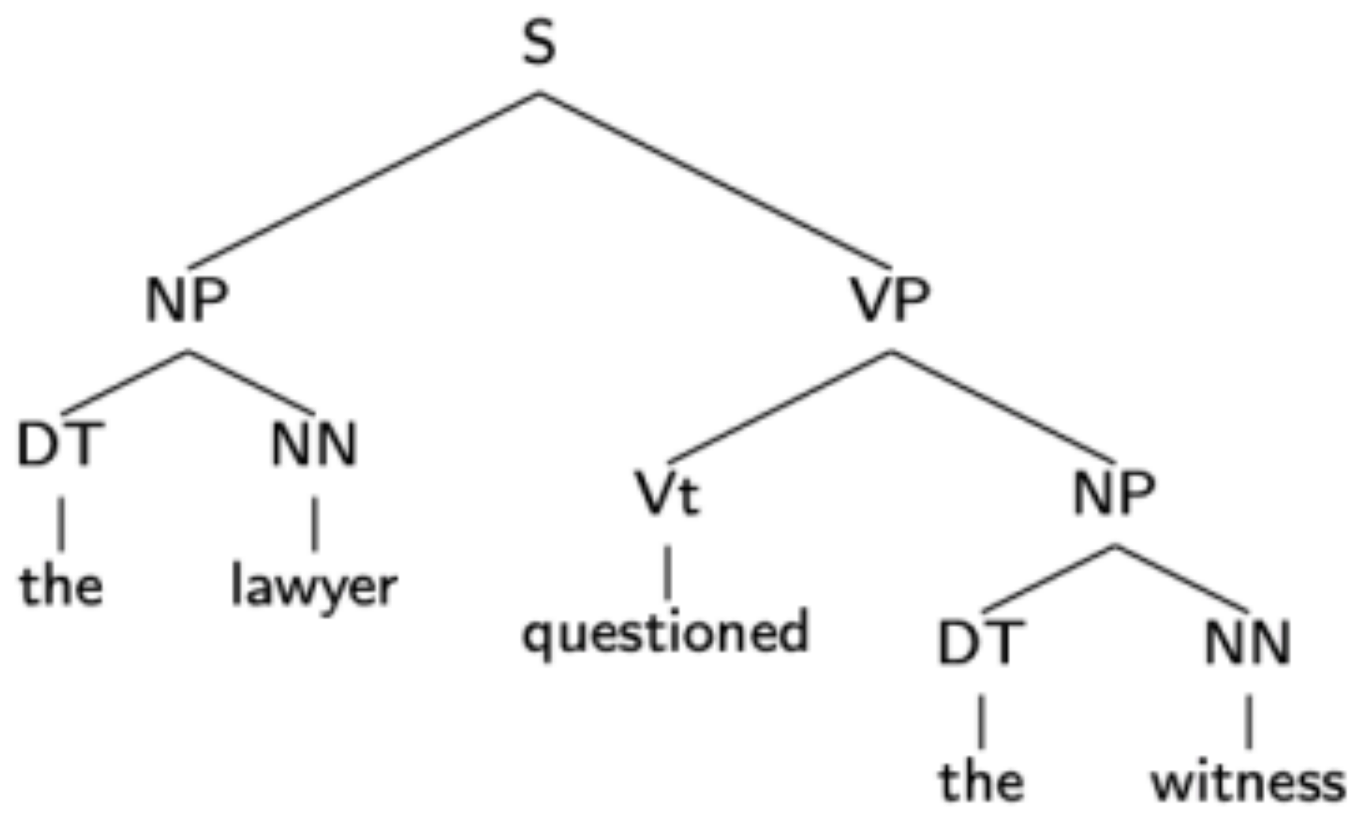
$q(\mathrm{VP} \rightarrow \mathrm{VP}\ \ \mathrm{PP})$ vs $q(\mathrm{NP} \rightarrow \mathrm{NP}\ \ \mathrm{PP})$

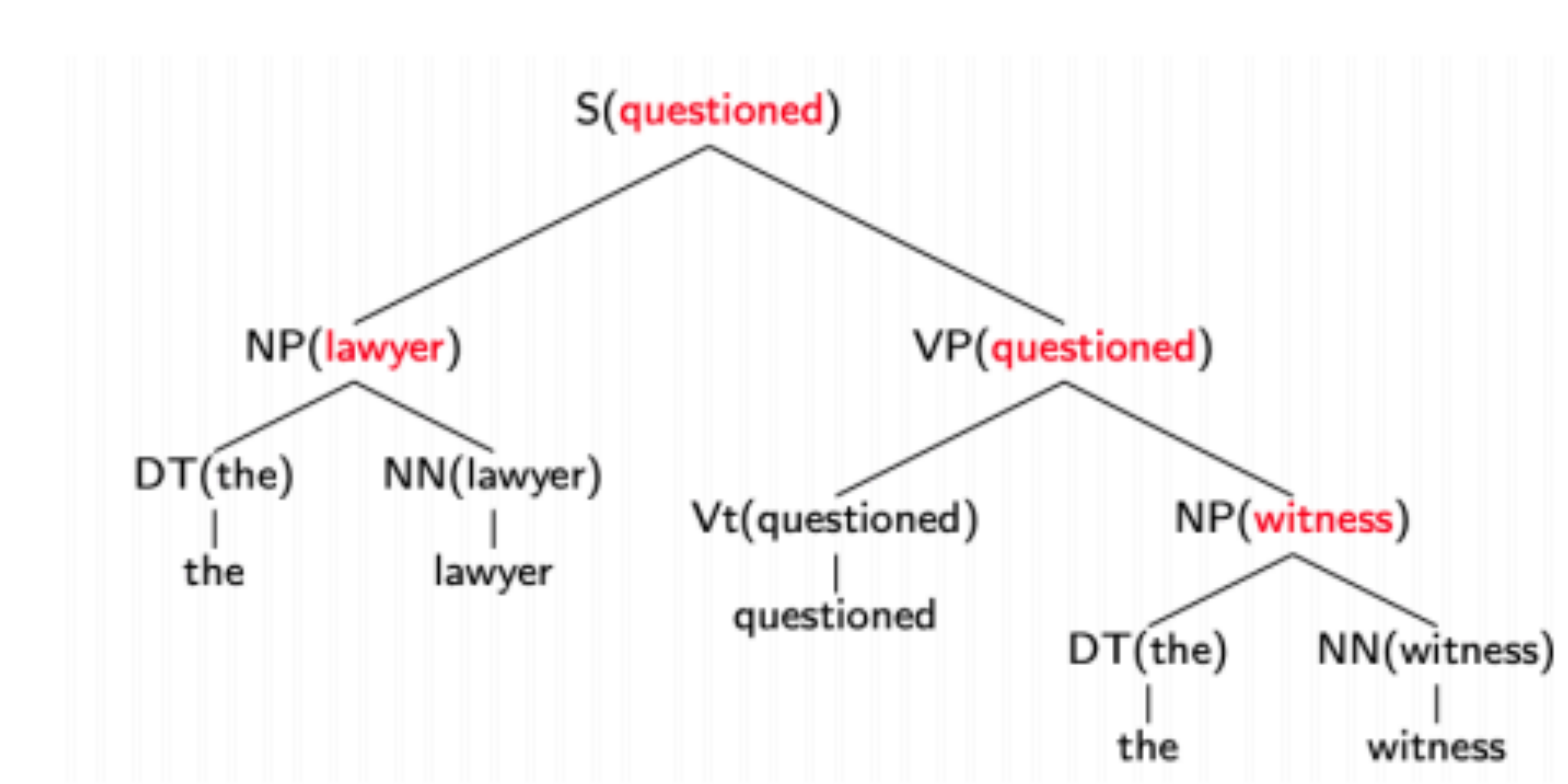
… without looking at the words!

Lexicalized PCFGs

Lexicalized PCFGs

- Key idea: add headwords to trees





- Each context-free rule has one special child that is the head of the rule

S	$\Rightarrow$	NP	VP		(VP is the head)
VP	$\Rightarrow$	Vt	NP		(Vt is the head)
NP	$\Rightarrow$	DT	NN	NN	(NN is the head)

Lexicalized PCFGs

The heads are decided by rules:

If the rule contains NN, NNS, or NNP:
Choose the rightmost NN, NNS, or NNP

Else If the rule contains an NP: Choose the leftmost NP

Else If the rule contains a JJ: Choose the rightmost JJ

Else If the rule contains a CD: Choose the rightmost CD

Else Choose the rightmost child

If the rule contains Vi or Vt: Choose the leftmost Vi or Vt

Else If the rule contains a VP: Choose the leftmost VP

Else Choose the leftmost child

Lexicalized PCFGs

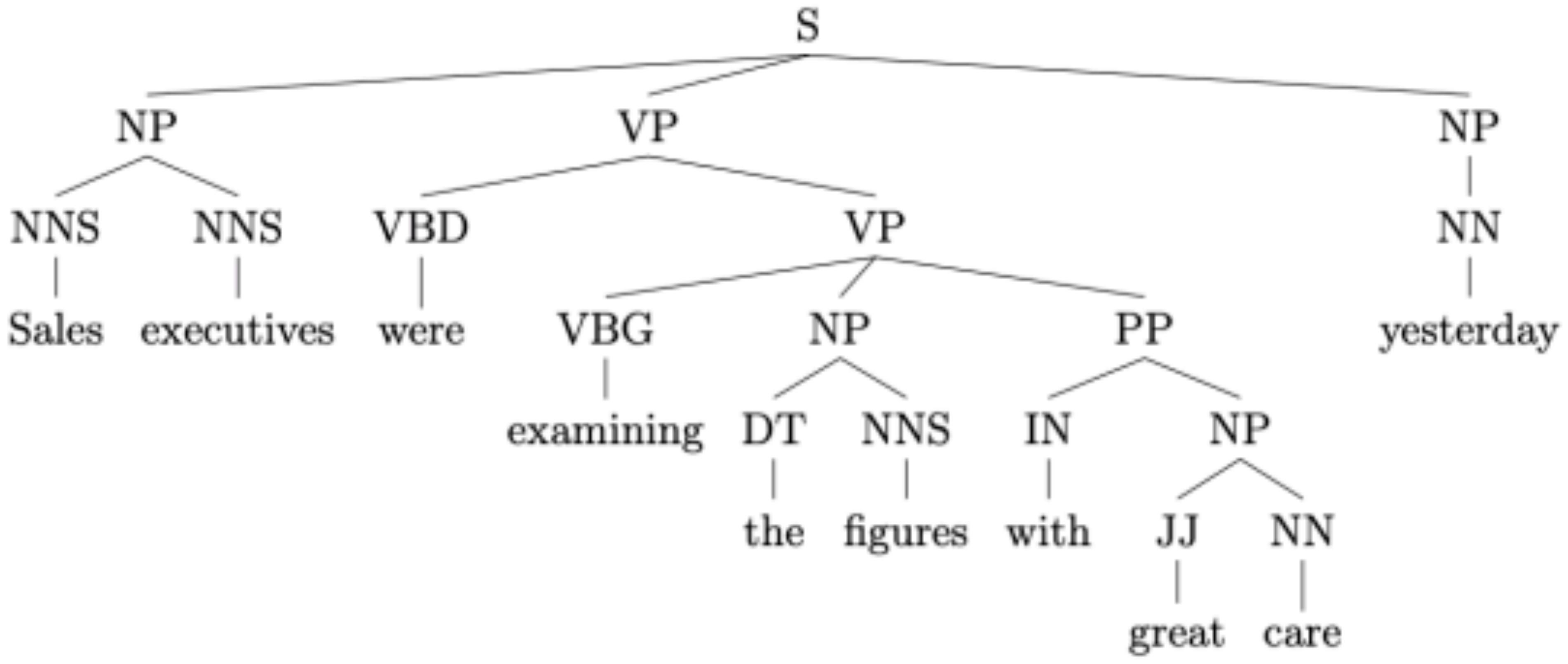
S(saw)	$\rightarrow_2$	NP(man)	VP(saw)
VP(saw)	$\rightarrow_1$	Vt(saw)	NP(dog)
NP(man)	$\rightarrow_2$	DT(the)	NN(man)
NP(dog)	$\rightarrow_2$	DT(the)	NN(dog)
Vt(saw)	$\rightarrow$	saw	
DT(the)	$\rightarrow$	the	
NN(man)	$\rightarrow$	man	
NN(dog)	$\rightarrow$	dog	

- Further reading: *Michael Collins. 2003. Head-Driven Statistical Models for Natural Language Parsing.*
- Results for a PCFG: 70.6% recall, 74.8% precision
- Results for a lexicalized PCFG: 88.1% recall, 88.3% precision

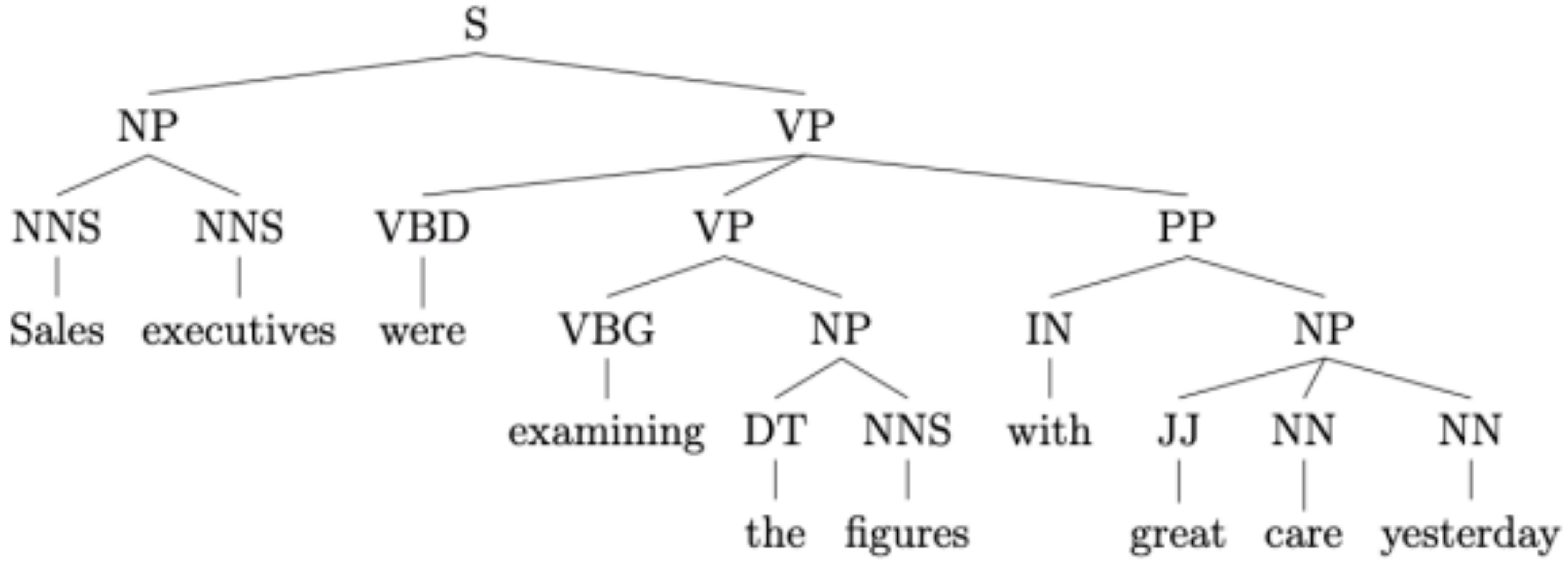
Evaluating Constituency Parsing

- Comparing the predicted tree against the gold-stand one

Gold: **(1, 10, S), (1, 2, NP)**, (3, 9, VP), (4, 9, VP), **(5, 6, NP)**, (7, 9, PP), (8, 9, NP), (10, 10, NP)



Predicted: **(1, 10, S), (1, 2, NP)**, (3, 10, VP), (4, 6, VP), **(5, 6, NP)**, (7, 10, PP), (8, 10, NP)



Evaluating Constituency Parsing

- Recall: (# correct constituents in candidate) / (# constituents in gold tree)
- Precision: (# correct constituents in candidate) / (# constituents in candidate)
- Labeled precision/recall require getting the non-terminal label correct
- F1 is the harmonic mean of precision and recall = (2 * precision * recall) / (precision + recall)
- Part-of-speech tagging accuracy is evaluated separately

Reading Materials

- **Notes from Michael Collins:**
  - Probabilistic Context-free Grammars (PCFGs)
  - Lexicalized PCFGs