

CS639 Deep Learning for NLP

Syntactic Parsing

Junjie Hu



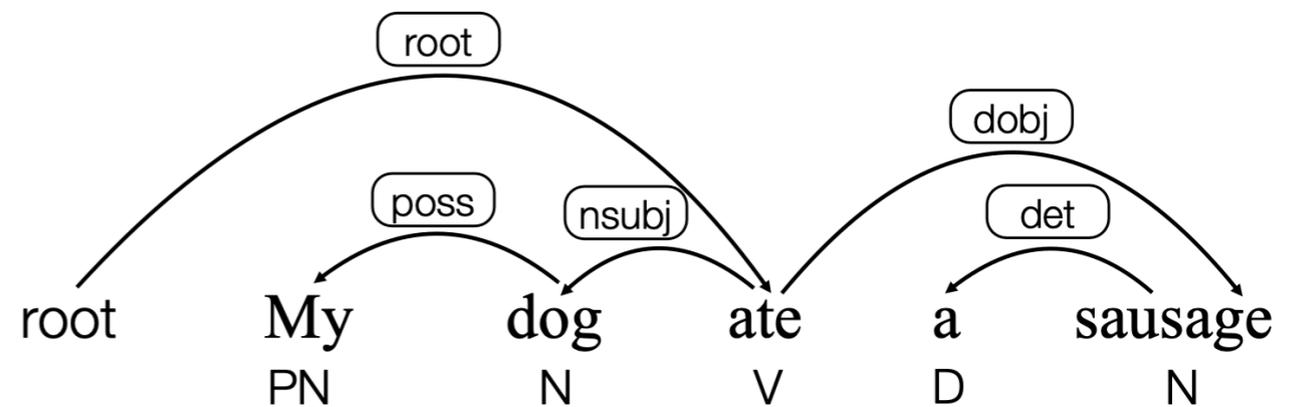
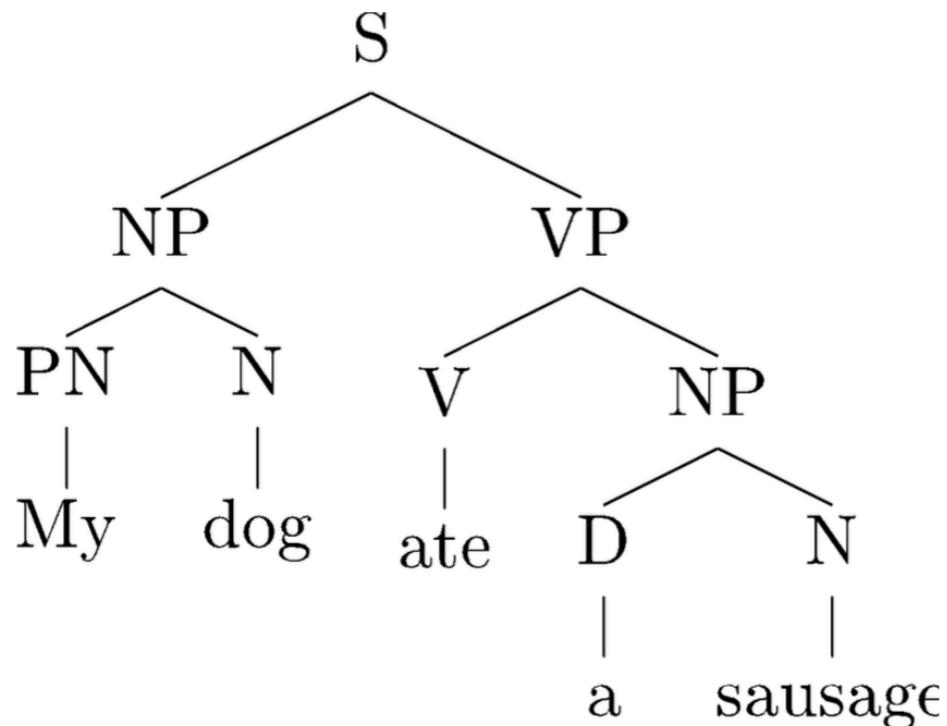
Slides adapted from Bob, Hao, Dan
<https://junjiehu.github.io/cs639-spring26>

Goals for Today

- Syntactic Parsing
- Probabilistic Context-Free Grammar (PCFG)
- **Supervised PCFG (Generative)**
- **CYK Decoding Algorithm**
- **Supervised Span-based Neural Models (Discriminative)**

Syntactic Parsing

- The process of predicting **syntactic representations**
- Two types of linguistic structures:



Constituency (aka phrase structure) tree:

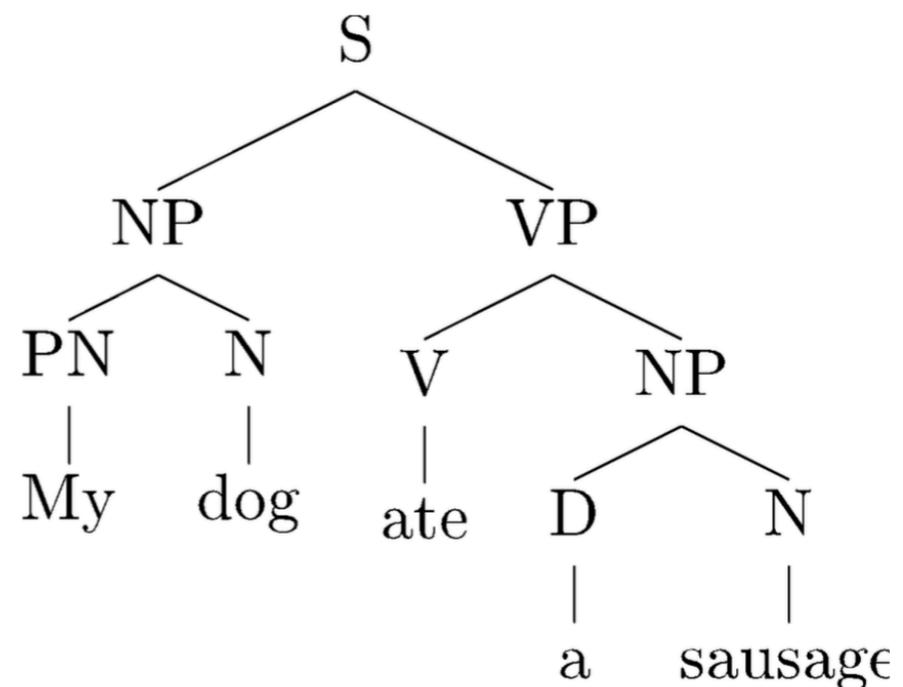
Focus on the structure of the sentence

Dependency tree:

Focus on relations between words

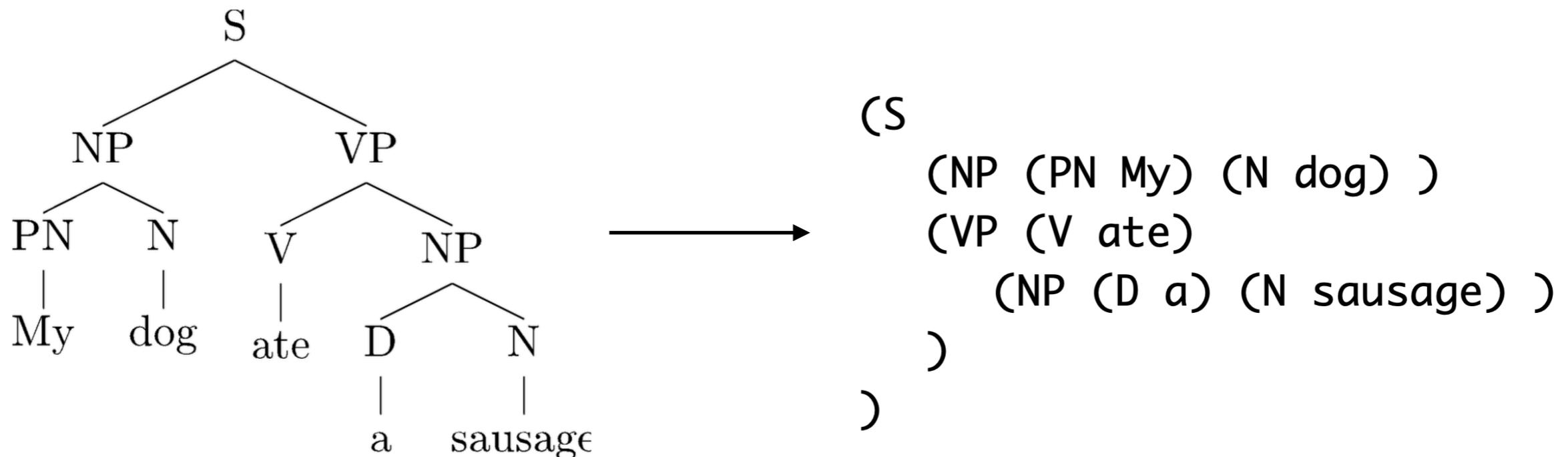
Constituency Trees

- Internal nodes (or non-terminals) correspond to phrases
 - S: a sentence
 - NP (noun phrase): My dog, a sandwich, ...
 - VP (verb phrase): ate a sausage, ...
 - PP (prepositional phrases): with a friend, in a car, ...
- Nodes immediately above words are part-of-speech tags (or preterminals).
 - PN: pronoun
 - D: determiner
 - V: verb
 - N: noun
 - P: preposition



Bracketing notation

- Often convenient to represent a tree as **a bracketed sequence**:
- In principle, constituency tree can be an n-nary tree, however, it is easy to convert it to a binary tree (by adding a null non-terminal \emptyset). By convention, we often just represent the structure as a binary tree.



Context Free Grammar (CFG) & Probabilistic CFG

Context-free grammars (CFG)

- **Context-free grammars (CFG)**: a formalism for parsing.

Grammar (CFG)

ROOT \rightarrow S

S \rightarrow NP VP

NP \rightarrow DT NN

NP \rightarrow NN NNS

NP \rightarrow NP PP

VP \rightarrow VBP NP

VP \rightarrow VBP NP PP

PP \rightarrow IN NP

Lexicon

NN \rightarrow interest

NNS \rightarrow raises

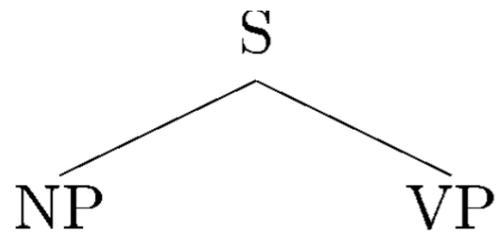
VBP \rightarrow interest

VBP \rightarrow raises

...

- Other **(non-CF)** grammar formalism: LFG, HPSG, TAG, CCG, ...

CFG for Syntactic Parsing



Grammar (CFG)

$S \rightarrow NP VP$

$VP \rightarrow V$

$VP \rightarrow V NP$

$VP \rightarrow VP PP$

$NP \rightarrow NP PP$

$NP \rightarrow D N$

$NP \rightarrow PN$

$PP \rightarrow P NP$

Lexicon

$N \rightarrow \text{girl}$

$N \rightarrow \text{telescope}$

$N \rightarrow \text{sandwich}$

$PN \rightarrow I$

$V \rightarrow \text{saw}$

$V \rightarrow \text{ate}$

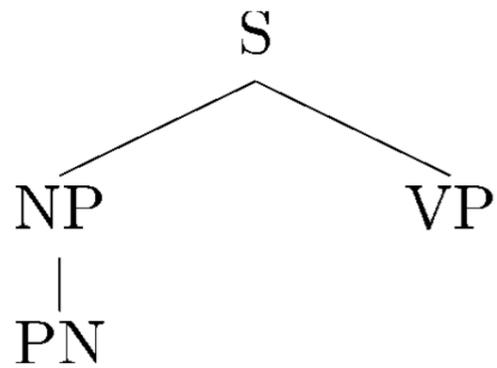
$P \rightarrow \text{with}$

$P \rightarrow \text{in}$

$D \rightarrow \text{a}$

$D \rightarrow \text{the}$

CFG for Syntactic Parsing



Grammar (CFG)

$S \rightarrow NP VP$

$VP \rightarrow V$

$VP \rightarrow V NP$

$VP \rightarrow VP PP$

$NP \rightarrow NP PP$

$NP \rightarrow D N$

$NP \rightarrow PN$

$PP \rightarrow P NP$

Lexicon

$N \rightarrow \text{girl}$

$N \rightarrow \text{telescope}$

$N \rightarrow \text{sandwich}$

$PN \rightarrow I$

$V \rightarrow \text{saw}$

$V \rightarrow \text{ate}$

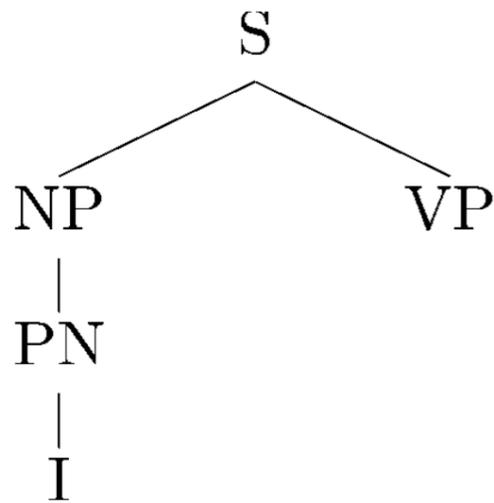
$P \rightarrow \text{with}$

$P \rightarrow \text{in}$

$D \rightarrow \text{a}$

$D \rightarrow \text{the}$

CFG for Syntactic Parsing



Grammar (CFG)

$S \rightarrow NP VP$

$VP \rightarrow V$

$VP \rightarrow V NP$

$VP \rightarrow VP PP$

$NP \rightarrow NP PP$

$NP \rightarrow D N$

$NP \rightarrow PN$

$PP \rightarrow P NP$

Lexicon

$N \rightarrow \text{girl}$

$N \rightarrow \text{telescope}$

$N \rightarrow \text{sandwich}$

$PN \rightarrow I$

$V \rightarrow \text{saw}$

$V \rightarrow \text{ate}$

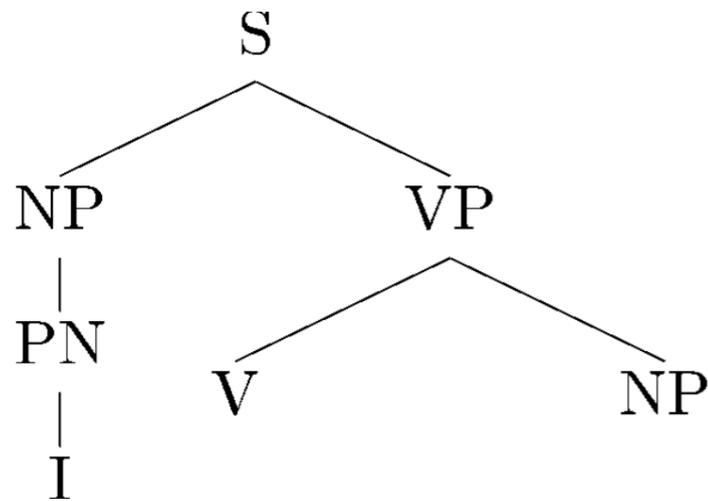
$P \rightarrow \text{with}$

$P \rightarrow \text{in}$

$D \rightarrow \text{a}$

$D \rightarrow \text{the}$

CFG for Syntactic Parsing



Grammar (CFG)

$S \rightarrow NP VP$

$VP \rightarrow V$

$VP \rightarrow V NP$

$VP \rightarrow VP PP$

$NP \rightarrow NP PP$

$NP \rightarrow D N$

$NP \rightarrow PN$

$PP \rightarrow P NP$

Lexicon

$N \rightarrow \text{girl}$

$N \rightarrow \text{telescope}$

$N \rightarrow \text{sandwich}$

$PN \rightarrow I$

$V \rightarrow \text{saw}$

$V \rightarrow \text{ate}$

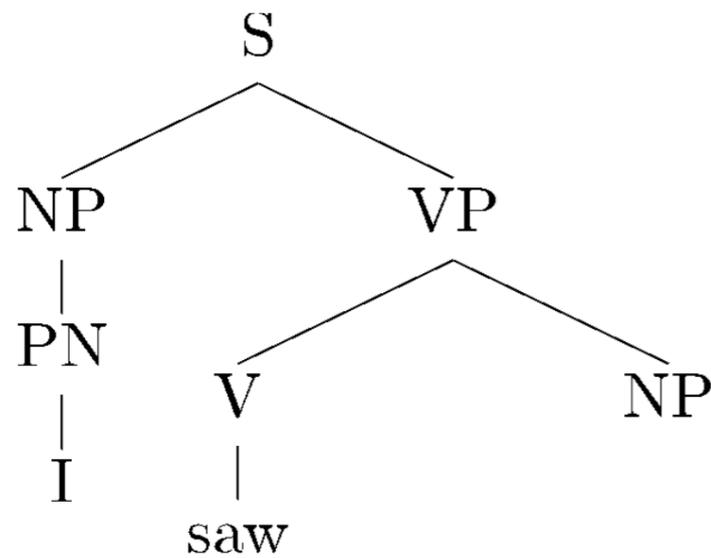
$P \rightarrow \text{with}$

$P \rightarrow \text{in}$

$D \rightarrow a$

$D \rightarrow \text{the}$

CFG for Syntactic Parsing



Grammar (CFG)

$S \rightarrow NP VP$

$VP \rightarrow V$

$VP \rightarrow V NP$

$VP \rightarrow VP PP$

$NP \rightarrow NP PP$

$NP \rightarrow D N$

$NP \rightarrow PN$

$PP \rightarrow P NP$

Lexicon

$N \rightarrow \text{girl}$

$N \rightarrow \text{telescope}$

$N \rightarrow \text{sandwich}$

$PN \rightarrow I$

$V \rightarrow \text{saw}$

$V \rightarrow \text{ate}$

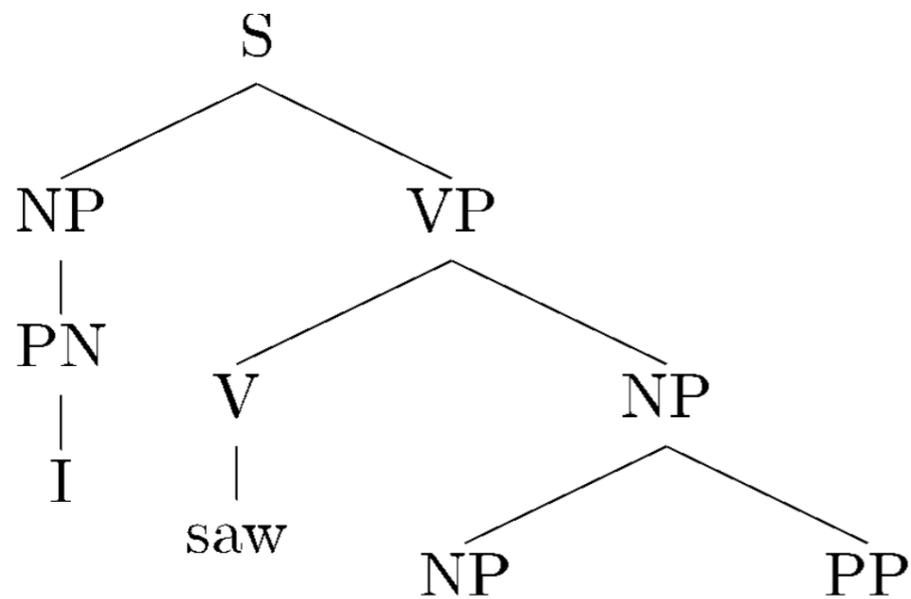
$P \rightarrow \text{with}$

$P \rightarrow \text{in}$

$D \rightarrow a$

$D \rightarrow \text{the}$

CFG for Syntactic Parsing



Grammar (CFG)

$S \rightarrow NP VP$

$VP \rightarrow V$

$VP \rightarrow V NP$

$VP \rightarrow VP PP$

$NP \rightarrow NP PP$

$NP \rightarrow D N$

$NP \rightarrow PN$

$PP \rightarrow P NP$

Lexicon

$N \rightarrow \text{girl}$

$N \rightarrow \text{telescope}$

$N \rightarrow \text{sandwich}$

$PN \rightarrow I$

$V \rightarrow \text{saw}$

$V \rightarrow \text{ate}$

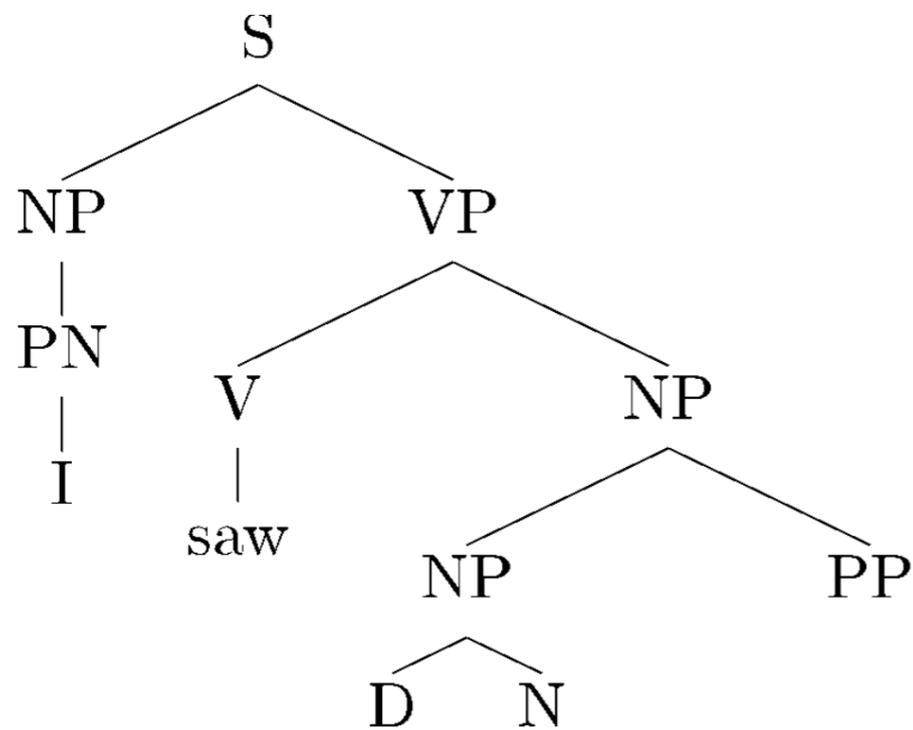
$P \rightarrow \text{with}$

$P \rightarrow \text{in}$

$D \rightarrow a$

$D \rightarrow \text{the}$

CFG for Syntactic Parsing



Grammar (CFG)

$S \rightarrow NP VP$

$VP \rightarrow V$

$VP \rightarrow V NP$

$VP \rightarrow VP PP$

$NP \rightarrow NP PP$

$NP \rightarrow D N$

$NP \rightarrow PN$

$PP \rightarrow P NP$

Lexicon

$N \rightarrow \text{girl}$

$N \rightarrow \text{telescope}$

$N \rightarrow \text{sandwich}$

$PN \rightarrow I$

$V \rightarrow \text{saw}$

$V \rightarrow \text{ate}$

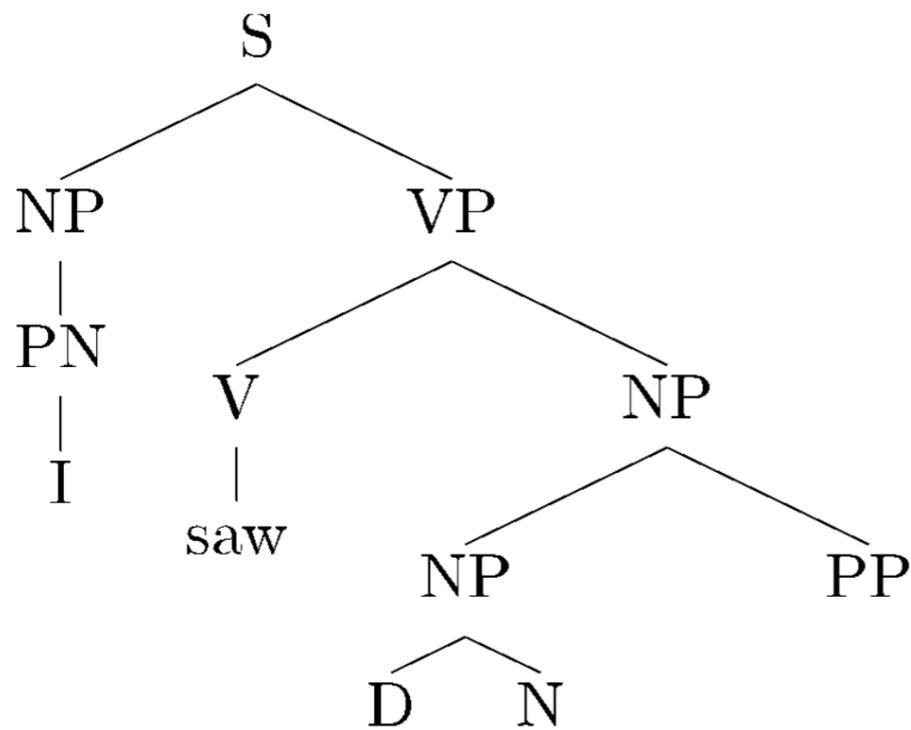
$P \rightarrow \text{with}$

$P \rightarrow \text{in}$

$D \rightarrow a$

$D \rightarrow \text{the}$

CFG for Syntactic Parsing



Grammar (CFG)

$S \rightarrow NP VP$

$VP \rightarrow V$

$VP \rightarrow V NP$

$VP \rightarrow VP PP$

$NP \rightarrow NP PP$

$NP \rightarrow D N$

$NP \rightarrow PN$

$PP \rightarrow P NP$

Lexicon

$N \rightarrow \text{girl}$

$N \rightarrow \text{telescope}$

$N \rightarrow \text{sandwich}$

$PN \rightarrow I$

$V \rightarrow \text{saw}$

$V \rightarrow \text{ate}$

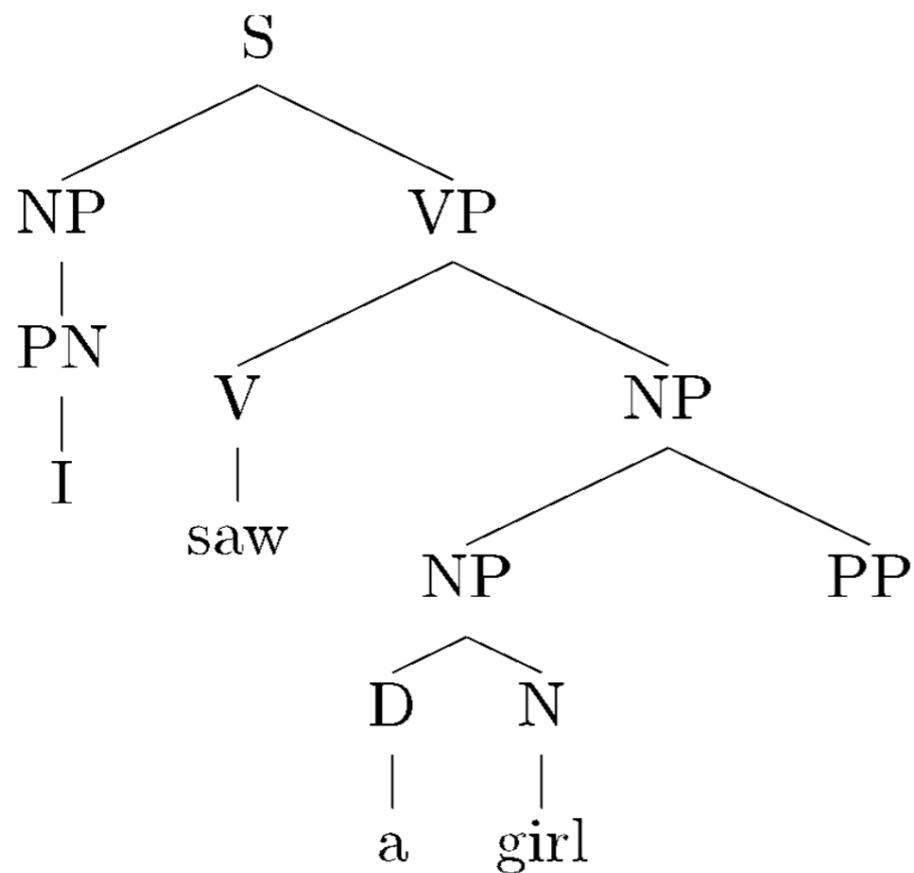
$P \rightarrow \text{with}$

$P \rightarrow \text{in}$

$D \rightarrow a$

$D \rightarrow \text{the}$

CFG for Syntactic Parsing



Grammar (CFG)

$S \rightarrow NP VP$

$VP \rightarrow V$

$VP \rightarrow V NP$

$VP \rightarrow VP PP$

$NP \rightarrow NP PP$

$NP \rightarrow D N$

$NP \rightarrow PN$

$PP \rightarrow P NP$

Lexicon

$N \rightarrow \text{girl}$

$N \rightarrow \text{telescope}$

$N \rightarrow \text{sandwich}$

$PN \rightarrow I$

$V \rightarrow \text{saw}$

$V \rightarrow \text{ate}$

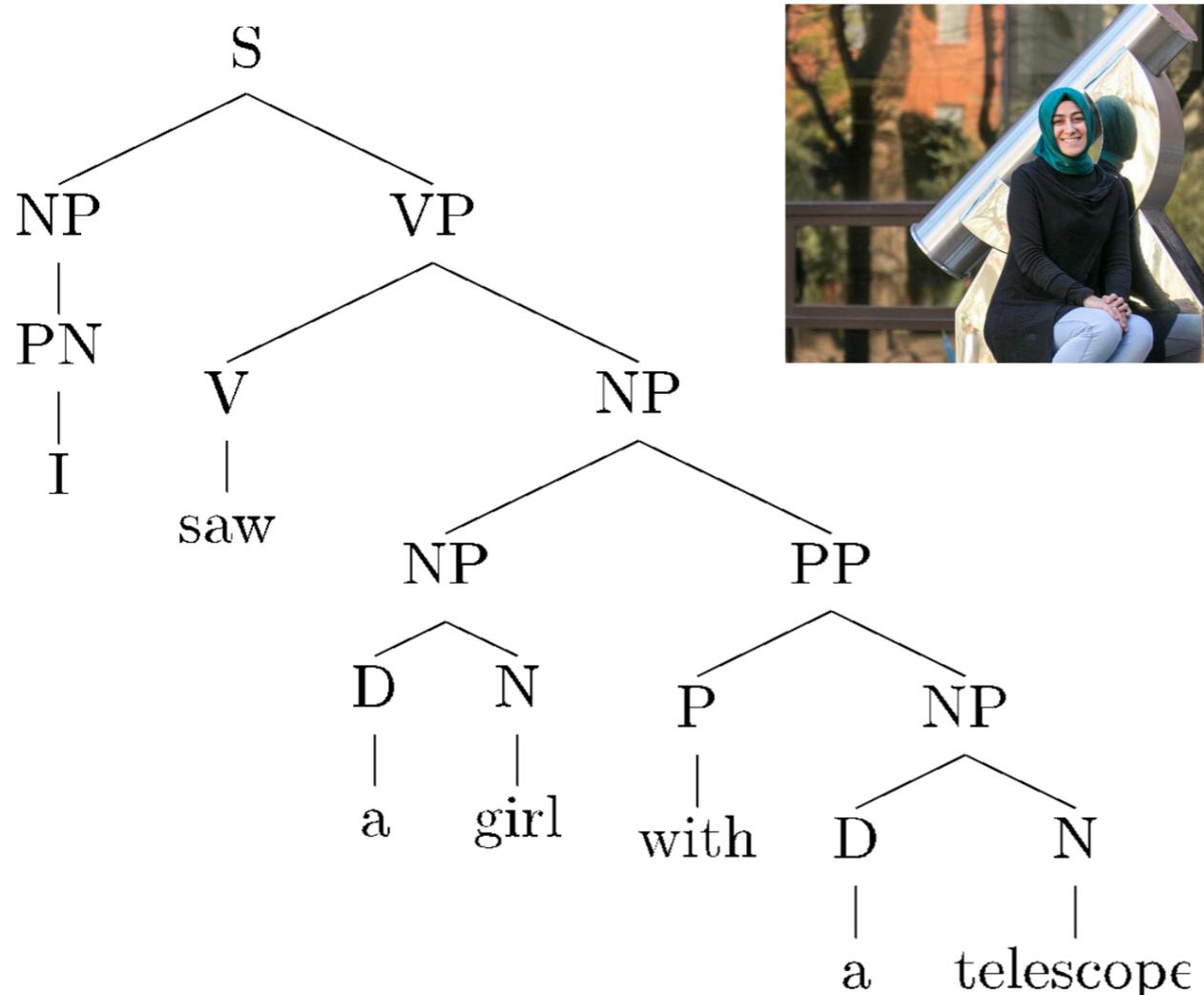
$P \rightarrow \text{with}$

$P \rightarrow \text{in}$

$D \rightarrow a$

$D \rightarrow \text{the}$

CFG for Syntactic Parsing



Grammar (CFG)

$S \rightarrow NP VP$

$VP \rightarrow V$

$VP \rightarrow V NP$

$VP \rightarrow VP PP$

$NP \rightarrow NP PP$

$NP \rightarrow D N$

$NP \rightarrow PN$

$PP \rightarrow P NP$

Lexicon

$N \rightarrow \text{girl}$

$N \rightarrow \text{telescope}$

$N \rightarrow \text{sandwich}$

$PN \rightarrow I$

$V \rightarrow \text{saw}$

$V \rightarrow \text{ate}$

$P \rightarrow \text{with}$

$P \rightarrow \text{in}$

$D \rightarrow a$

$D \rightarrow \text{the}$

Probabilistic context-free grammars (PCFG)

- **CFG**: A 4-tuple (N, Σ, R, S) :
 - N : a set of non-terminal symbols
 - Σ : a set of terminal symbols (disjoint from N)
 - S : a designated start symbol and a member of N
 - R : a set of rules, each of the form $A \rightarrow s$, where A is a non-terminal, s is a string of symbols, $A \in N, s \in (\Sigma \cup N)^*$

$$\begin{array}{l} S \rightarrow A, \quad A \in N \\ A \rightarrow BC, \quad A \in N, B, C \in N \cup \Sigma \\ A \rightarrow \alpha, \quad \alpha \in \Sigma \end{array}$$

Without loss of generality, only consider binary branching; Chomsky Normal Form

- **PCFG** adds a top-down production probability per rule.
 - Model the probability of each rule: $P(A \rightarrow s)$

$$\forall A \rightarrow s \in R : 0 \leq P(A \rightarrow s) \leq 1$$

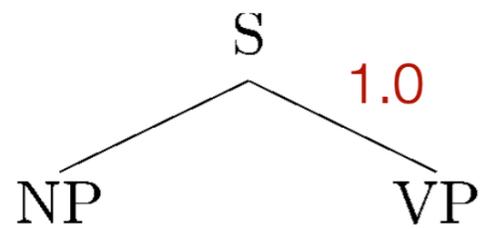
$$\forall A \in N : \sum_{s \text{ where } A \rightarrow s \in R} P(A \rightarrow s) = 1$$

PCFG (Example)

S → NP VP	1.0	(NP a girl) (VP ate a sandwich)	N → <i>girl</i>	0.2
VP → V	0.2		N → <i>telescope</i>	0.7
VP → V NP	0.4	(V ate) (NP a sandwich)	N → <i>sandwich</i>	0.1
VP → VP PP	0.4	(VP saw a girl) (PP with a telescope)	PN → <i>I</i>	1.0
NP → NP PP	0.3	(NP a girl) (PP with a sandwich)	V → <i>saw</i>	0.5
NP → D N	0.5	(D a) (N sandwich)	V → <i>ate</i>	0.5
NP → PN	0.2		P → <i>with</i>	0.6
PP → P NP	1.0	(P with) (NP a sandwich)	P → <i>in</i>	0.4
			D → <i>a</i>	0.3
			D → <i>the</i>	0.7

Now we can score a tree as a product of probabilities corresponding to the used rules!

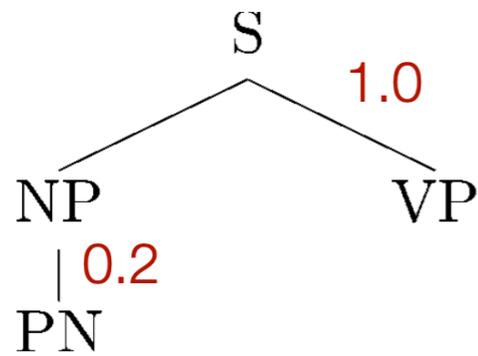
PCFG (Example)



$S \rightarrow NP VP$	1.0	$N \rightarrow girl$	0.2
		$N \rightarrow telescope$	0.7
$VP \rightarrow V$	0.2	$N \rightarrow sandwich$	0.1
$VP \rightarrow V NP$	0.4	$PN \rightarrow I$	1.0
$VP \rightarrow VP PP$	0.4	$V \rightarrow saw$	0.5
		$V \rightarrow ate$	0.5
$NP \rightarrow NP PP$	0.3	$P \rightarrow with$	0.6
$NP \rightarrow D N$	0.5	$P \rightarrow in$	0.4
$NP \rightarrow PN$	0.2	$D \rightarrow a$	0.3
		$D \rightarrow the$	0.7
$PP \rightarrow P NP$	1.0		

$$P(T, X) = 1.0^*$$

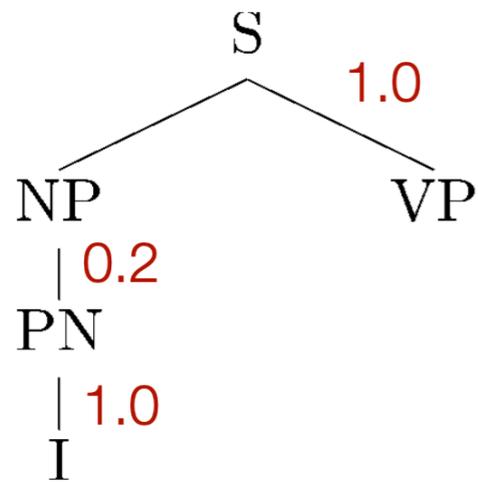
PCFG (Example)



S → NP VP	1.0	N → <i>girl</i>	0.2
		N → <i>telescope</i>	0.7
VP → V	0.2	N → <i>sandwich</i>	0.1
VP → V NP	0.4	PN → <i>I</i>	1.0
VP → VP PP	0.4	V → <i>saw</i>	0.5
		V → <i>ate</i>	0.5
NP → NP PP	0.3	P → <i>with</i>	0.6
NP → D N	0.5	P → <i>in</i>	0.4
NP → PN	0.2	D → <i>a</i>	0.3
		D → <i>the</i>	0.7
PP → P NP	1.0		

$$P(T, X) = 1.0 * 0.2 *$$

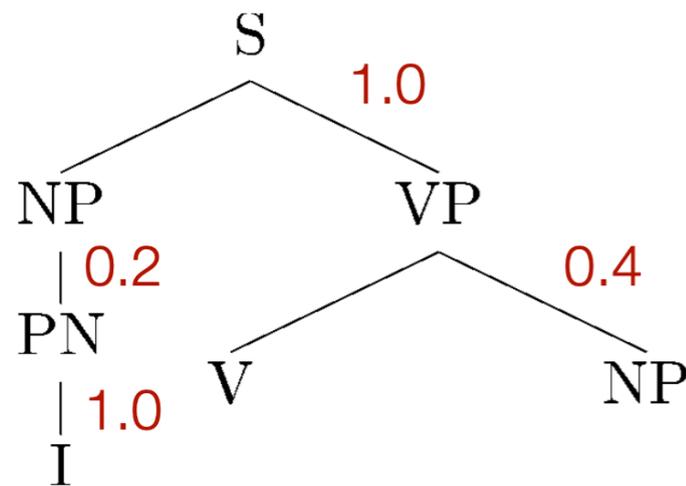
PCFG (Example)



S → NP VP	1.0	N → <i>girl</i>	0.2
		N → <i>telescope</i>	0.7
VP → V	0.2	N → <i>sandwich</i>	0.1
VP → V NP	0.4	PN → <i>I</i>	1.0
VP → VP PP	0.4	V → <i>saw</i>	0.5
		V → <i>ate</i>	0.5
NP → NP PP	0.3	P → <i>with</i>	0.6
NP → D N	0.5	P → <i>in</i>	0.4
NP → PN	0.2	D → <i>a</i>	0.3
		D → <i>the</i>	0.7
PP → P NP	1.0		

$$P(T, X) = 1.0 * 0.2 * 1.0 *$$

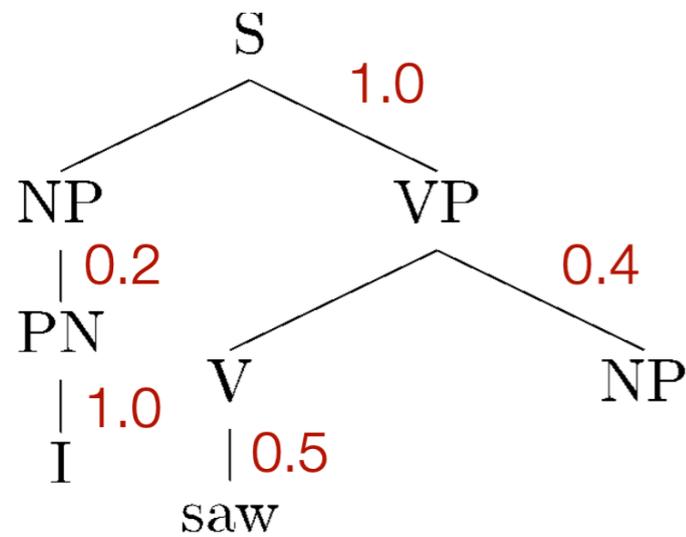
PCFG (Example)



S → NP VP	1.0	N → <i>girl</i>	0.2
		N → <i>telescope</i>	0.7
VP → V	0.2	N → <i>sandwich</i>	0.1
VP → V NP	0.4	PN → <i>I</i>	1.0
VP → VP PP	0.4	V → <i>saw</i>	0.5
		V → <i>ate</i>	0.5
NP → NP PP	0.3	P → <i>with</i>	0.6
NP → D N	0.5	P → <i>in</i>	0.4
NP → PN	0.2	D → <i>a</i>	0.3
		D → <i>the</i>	0.7
PP → P NP	1.0		

$$P(T, X) = 1.0 * 0.2 * 1.0 * 0.4 *$$

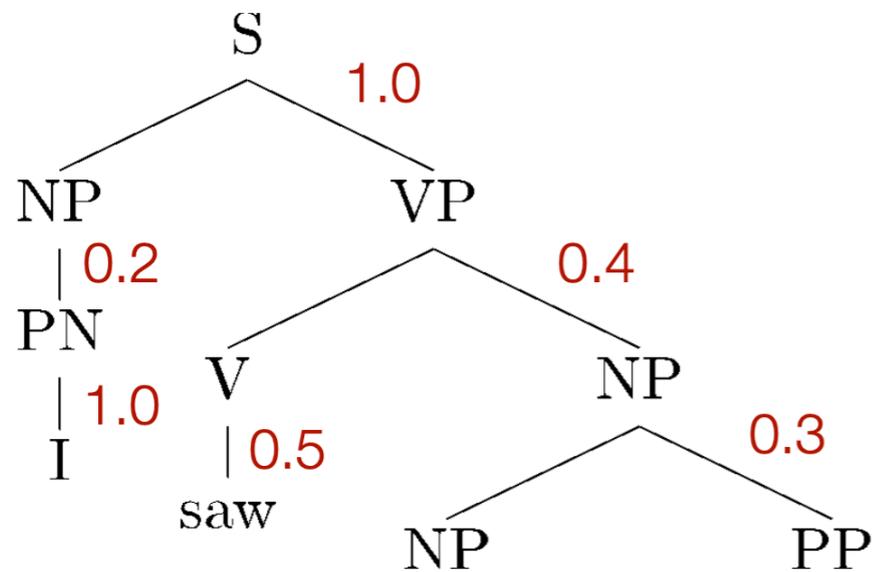
PCFG (Example)



S → NP VP	1.0	N → <i>girl</i>	0.2
		N → <i>telescope</i>	0.7
VP → V	0.2	N → <i>sandwich</i>	0.1
VP → V NP	0.4	PN → <i>I</i>	1.0
VP → VP PP	0.4	V → <i>saw</i>	0.5
		V → <i>ate</i>	0.5
NP → NP PP	0.3	P → <i>with</i>	0.6
NP → D N	0.5	P → <i>in</i>	0.4
NP → PN	0.2	D → <i>a</i>	0.3
		D → <i>the</i>	0.7
PP → P NP	1.0		

$$P(T, X) = 1.0 * 0.2 * 1.0 * 0.4 * 0.5 *$$

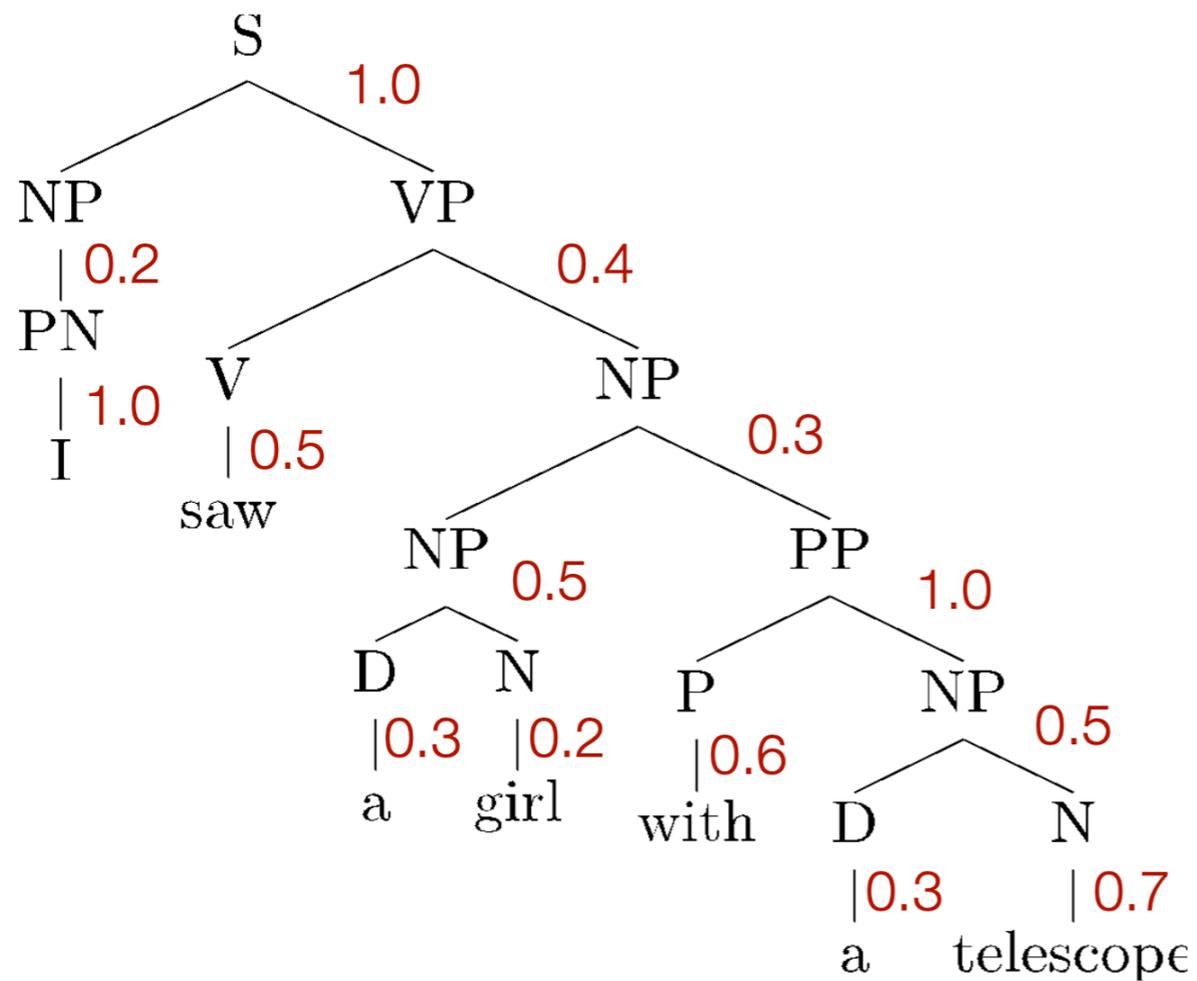
PCFG (Example)



S → NP VP	1.0	N → <i>girl</i>	0.2
		N → <i>telescope</i>	0.7
VP → V	0.2	N → <i>sandwich</i>	0.1
VP → V NP	0.4	PN → <i>I</i>	1.0
VP → VP PP	0.4	V → <i>saw</i>	0.5
		V → <i>ate</i>	0.5
NP → NP PP	0.3	P → <i>with</i>	0.6
NP → D N	0.5	P → <i>in</i>	0.4
NP → PN	0.2	D → <i>a</i>	0.3
		D → <i>the</i>	0.7
PP → P NP	1.0		

$$P(T, X) = 1.0 * 0.2 * 1.0 * 0.4 * 0.5 * 0.3$$

PCFG (Example)



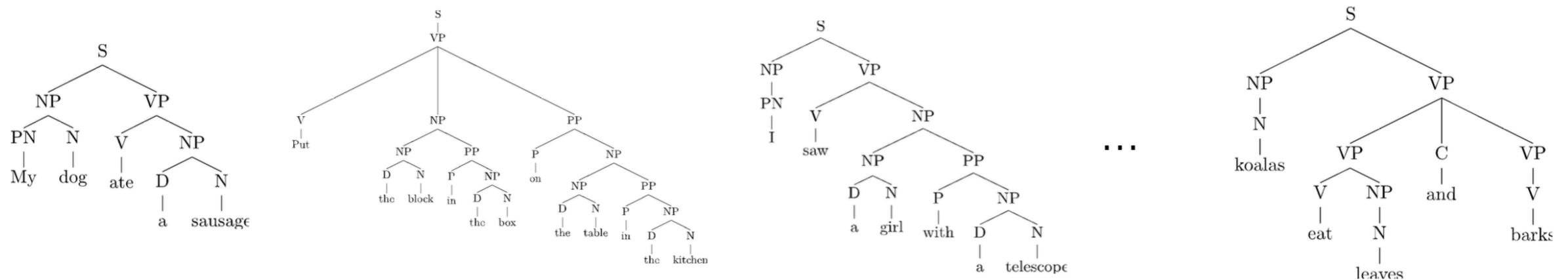
S → NP VP	1.0	N → <i>girl</i>	0.2
		N → <i>telescope</i>	0.7
VP → V	0.2	N → <i>sandwich</i>	0.1
VP → V NP	0.4	PN → <i>I</i>	1.0
VP → VP PP	0.4	V → <i>saw</i>	0.5
		V → <i>ate</i>	0.5
NP → NP PP	0.3	P → <i>with</i>	0.6
NP → D N	0.5	P → <i>in</i>	0.4
NP → PN	0.2	D → <i>a</i>	0.3
		D → <i>the</i>	0.7
PP → P NP	1.0		

$$P(T, X) = 1.0 * 0.2 * 1.0 * 0.4 * 0.5 * 0.3 * 0.2 * 1.0 * 0.6 * 0.5 * 0.3 * 0.7 = 2.26e-5$$

PCFG Supervised Learning & Decoding

PCFG Supervised Learning

- A treebank: a collection of sentences annotated with constituency trees
 - Penn Treebank: (X, T) pairs



- PCFG: a generative model, maximizing the joint probability of a sentence given a tree.
 - If we constraint the search space to be all valid trees that can generate the sentence, this becomes:

$$\max P(X, T) = \max P(X|T)P(T) = \max_{T \in \text{GEN}(X)} P(X|T)P(T)$$

PCFG Supervised Learning

- Estimate probability of each rule by **maximum likelihood estimation (MLE)**:

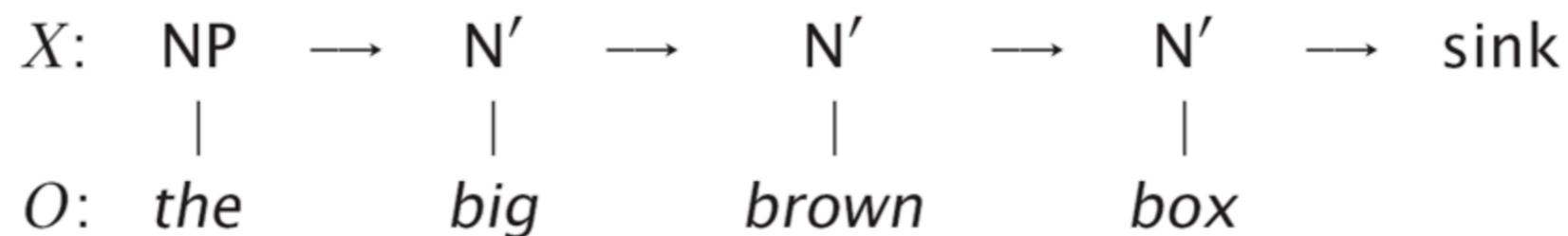
$$P(T, X) = \sum_{A \rightarrow s \in R} P(A \rightarrow s), \quad T \in \text{GEN}(X)$$

$$P(A \rightarrow s) = \frac{\text{Count}(A \rightarrow s)}{\text{Count}(A)} \quad \begin{array}{l} \# \text{ times the rule was used in the data} \\ \# \text{ times the nonterminal was used in the data} \end{array}$$

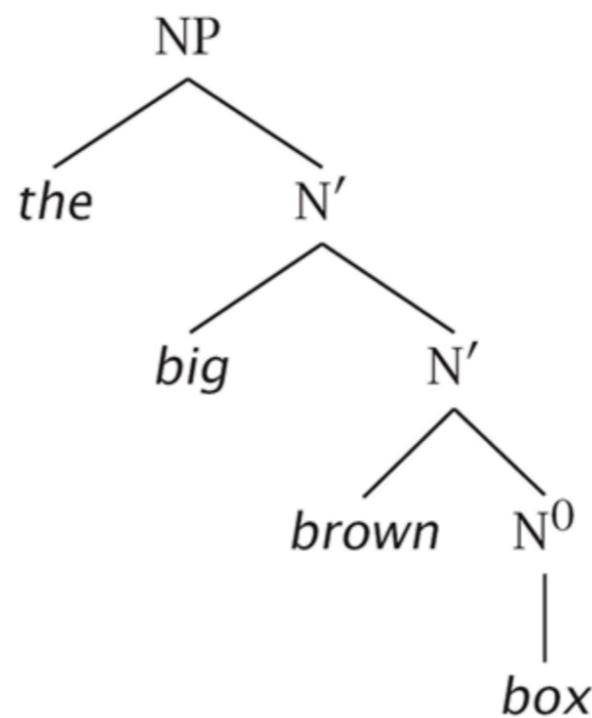
- Smoothing is helpful (esp. for rules that produce one word)
- If we don't have training data, use EM algorithm to estimate the probability

Comparison HMM vs PCFG

HMM: Linear Markov Chain

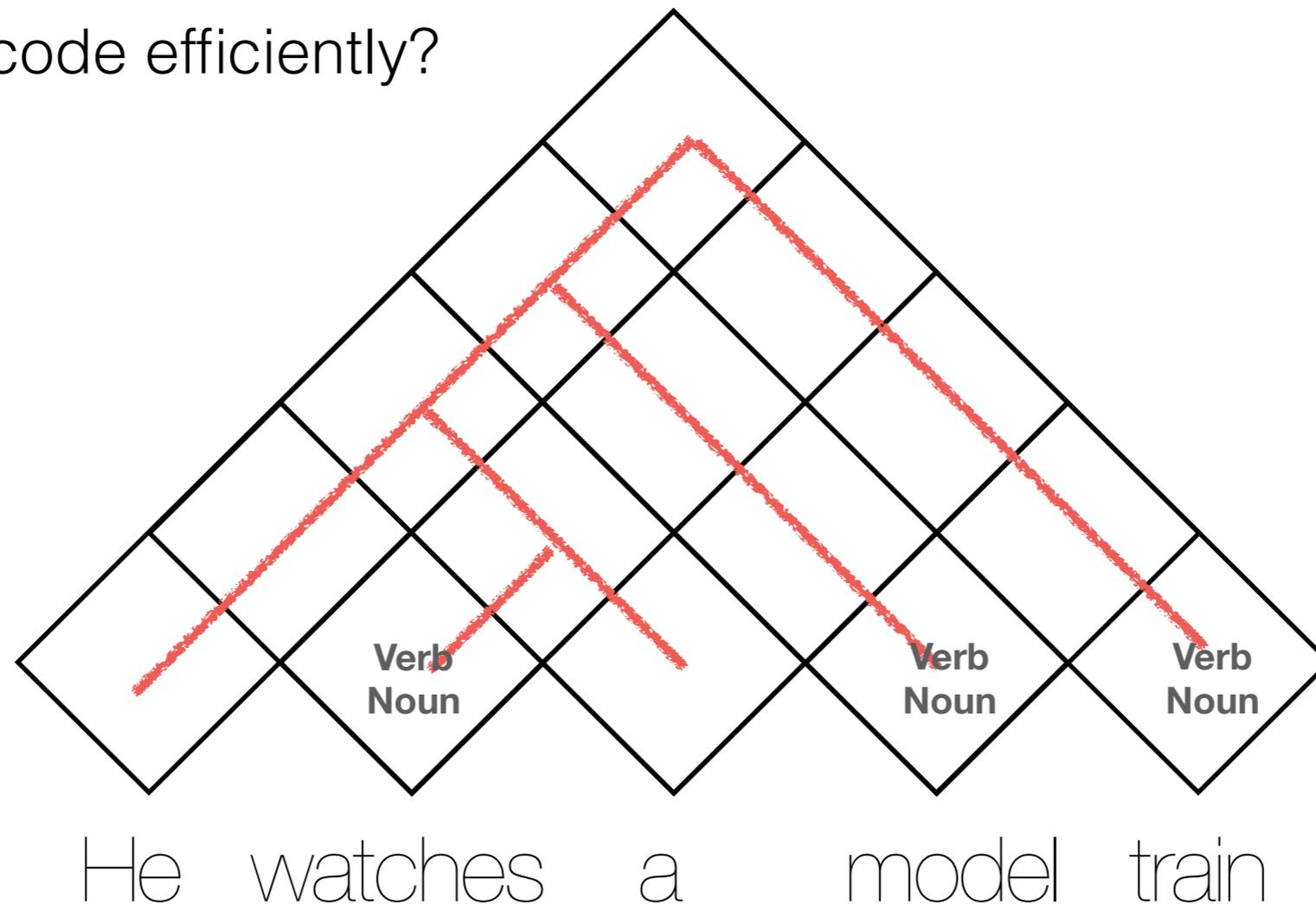


PCFG: tree



PCFG Decoding

- Brute force solution: enumerate all possible binary trees, score them, find the tree with maximum score
- For a sentence of n words, there are $(n-1)!$ possible binary trees. Each word may have more than 1 possible POS tags
- How to decode efficiently?



PCFG Decoding: CYK Algorithm

Bottom-up Dynamic Programming

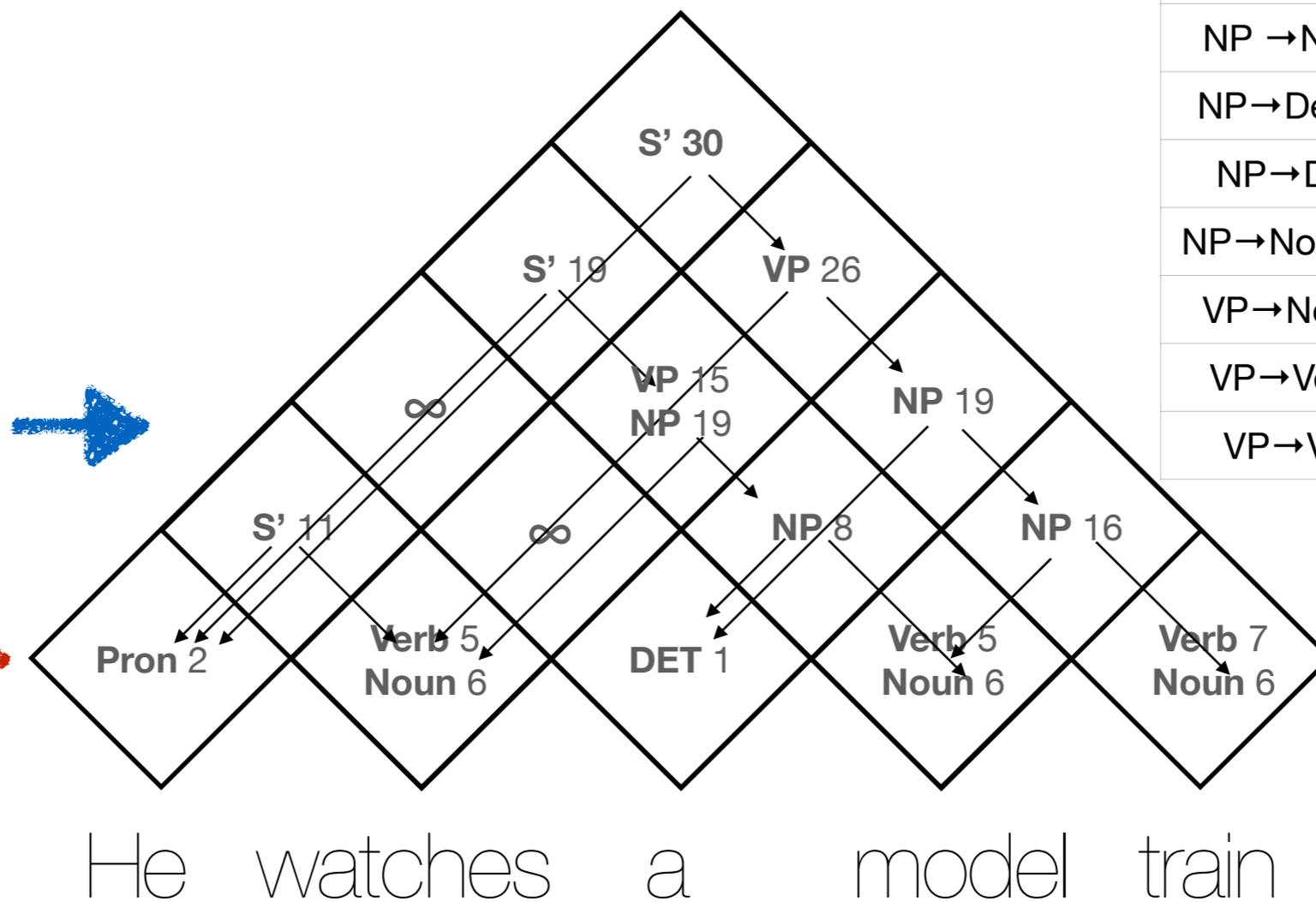
Remember to store back-pointer!

Binary Rule	-log prob
$S' \rightarrow \text{Pron Verb}$	4
$S' \rightarrow \text{Pron VP}$	2
$S' \rightarrow \text{NP VP}$	2
$\text{NP} \rightarrow \text{NP Verb}$	5
$\text{NP} \rightarrow \text{Det Noun}$	2
$\text{NP} \rightarrow \text{Det NP}$	2
$\text{NP} \rightarrow \text{Noun Noun}$	4
$\text{VP} \rightarrow \text{Noun NP}$	5
$\text{VP} \rightarrow \text{Verb NP}$	2
$\text{VP} \rightarrow \text{VP NP}$	2

Binary rule
 $A \rightarrow BC$

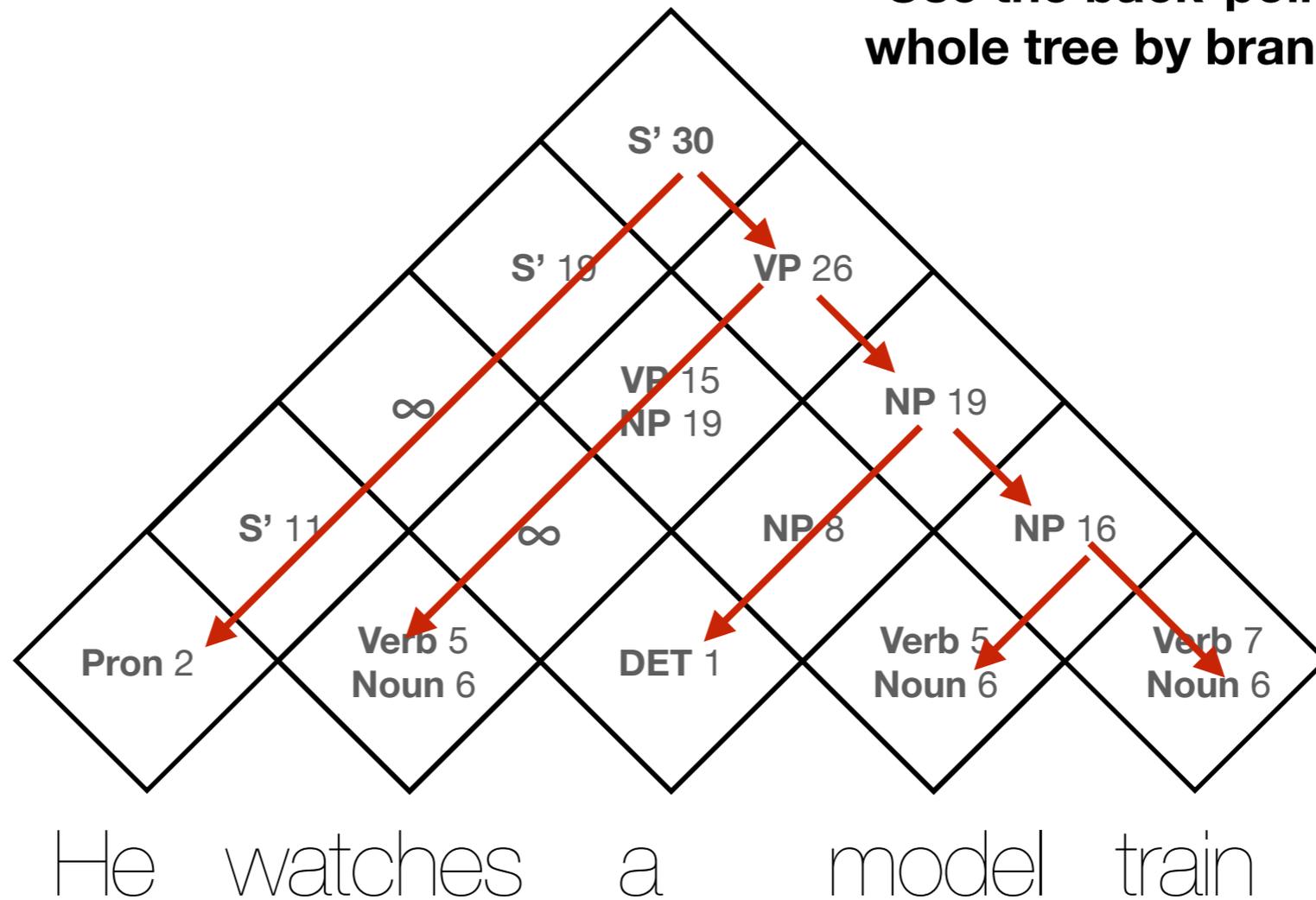
Lexical rule
(Unary rule)

$A \rightarrow \alpha, \alpha \in \Sigma$



PCFG Decoding: CYK Algorithm

Use the back-pointer to build the whole tree by branching from root.



PCFG Decoding: CYK Algorithm

- The score (or unnormalized probability) of a constituent with a non-terminal is often called inside probability $s_{\text{label}}(i, j, A)$
 - Computed by dynamic programming

$$s_{\text{label}}(i, j, A) = \max_{k, B, C} P(A \rightarrow BC) \times s_{\text{label}}(i, k, B) \times s_{\text{label}}(k + 1, j, C)$$

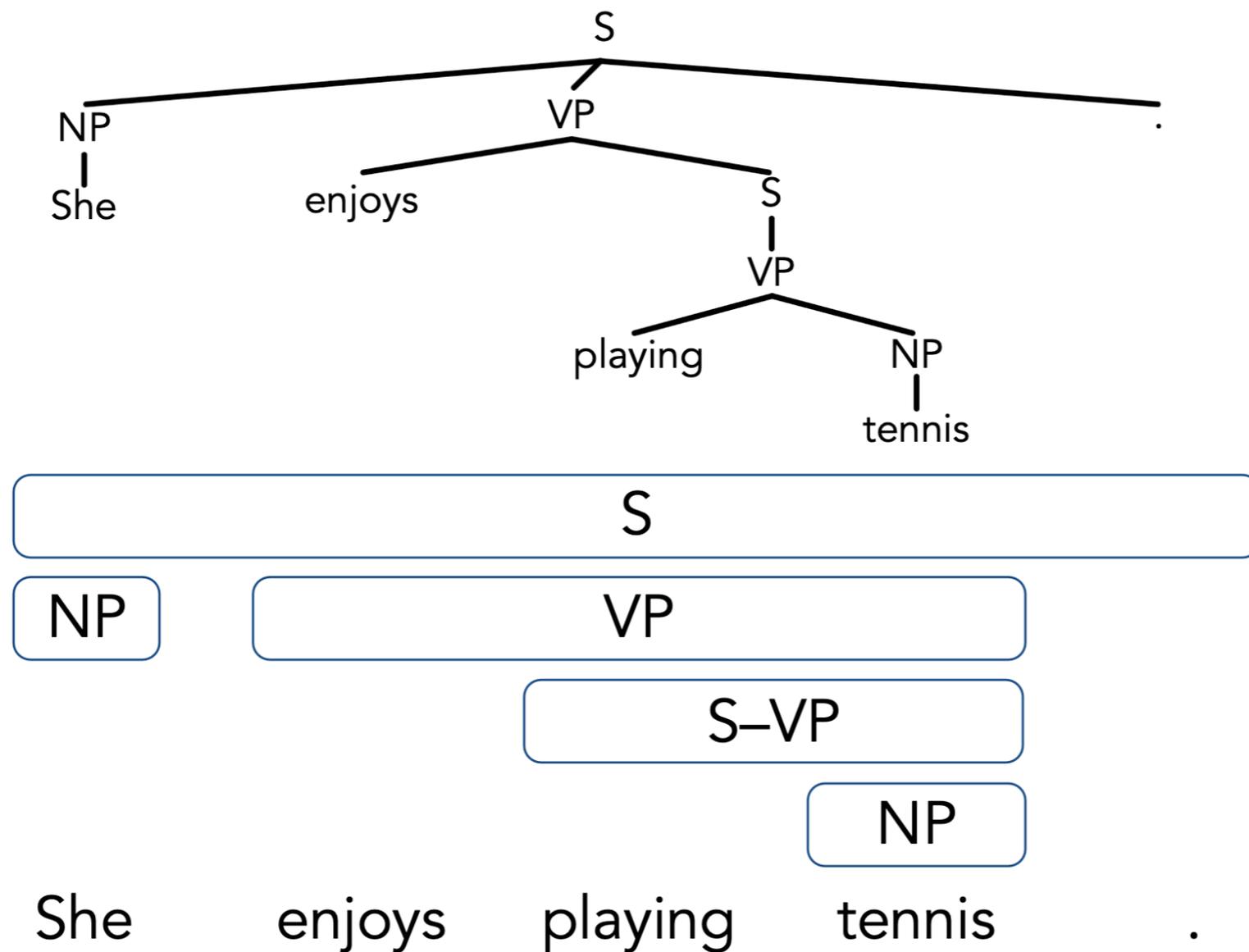
- The best optimal score of the whole sentence of length n is derived by

$$s_{\text{label}}(1, n, S)$$

Supervised Parsing: Span-based Neural Models

Span-Based Parsing

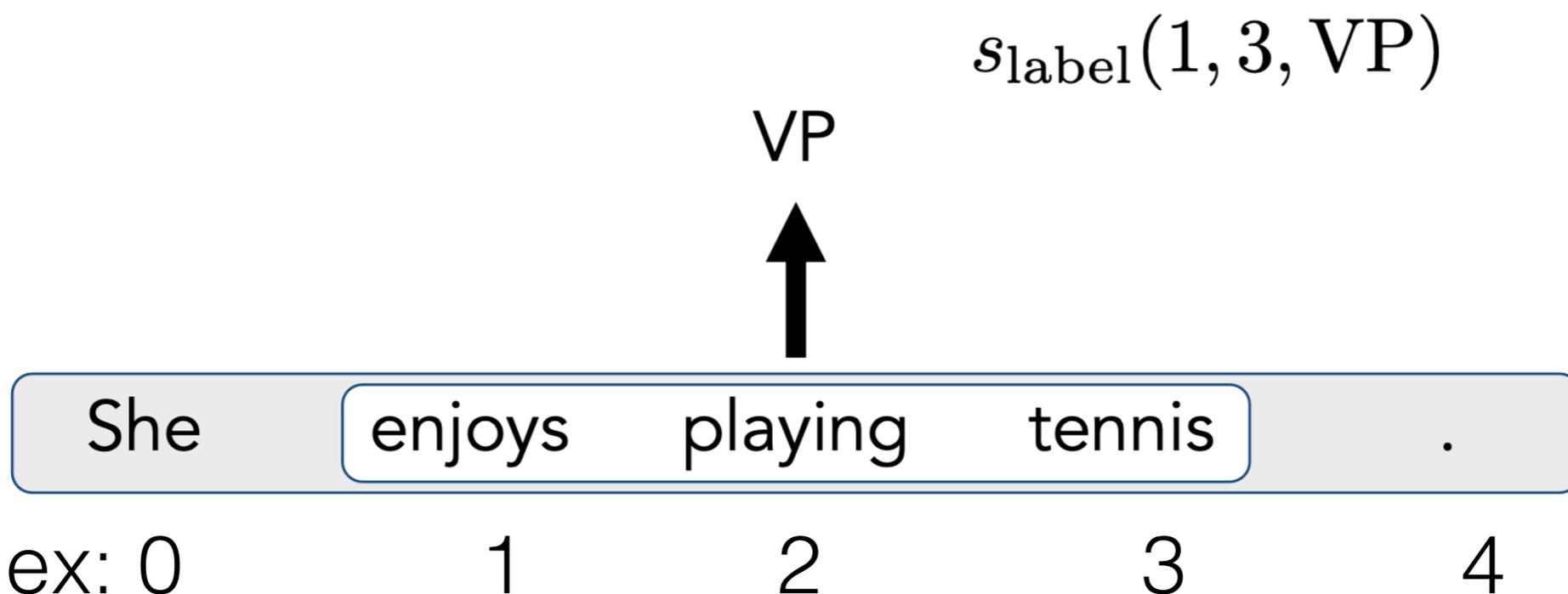
$$P(Y_{i:j} = c | X_{i:j}) = w_c \cdot F_c(X_{i:j})$$



Span-Based Parsing

$$s_{\text{label}}(i, j, \ell)$$

Scoring a span from the i -th word to j -th word being the label of ℓ



Word index: 0

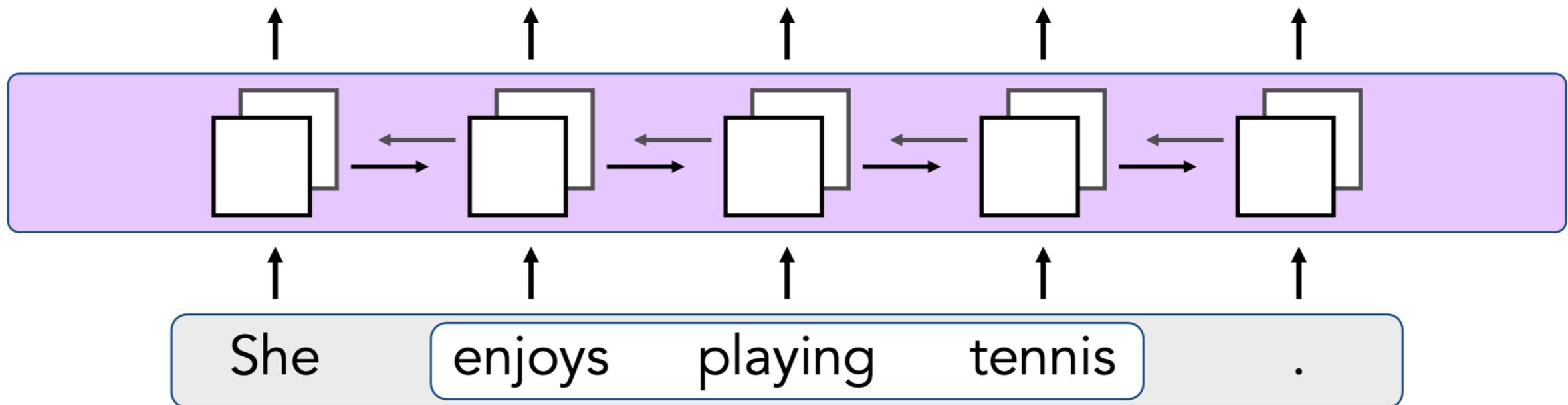
1

2

3

4

Span-Based Parsing

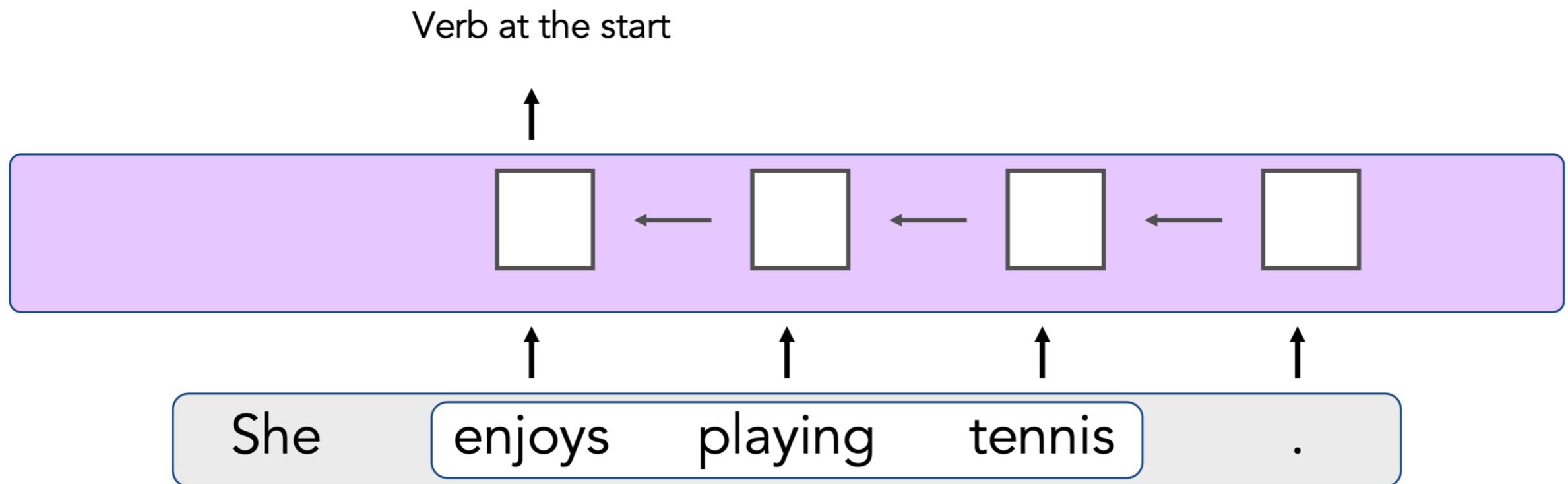


Span-Based Parsing

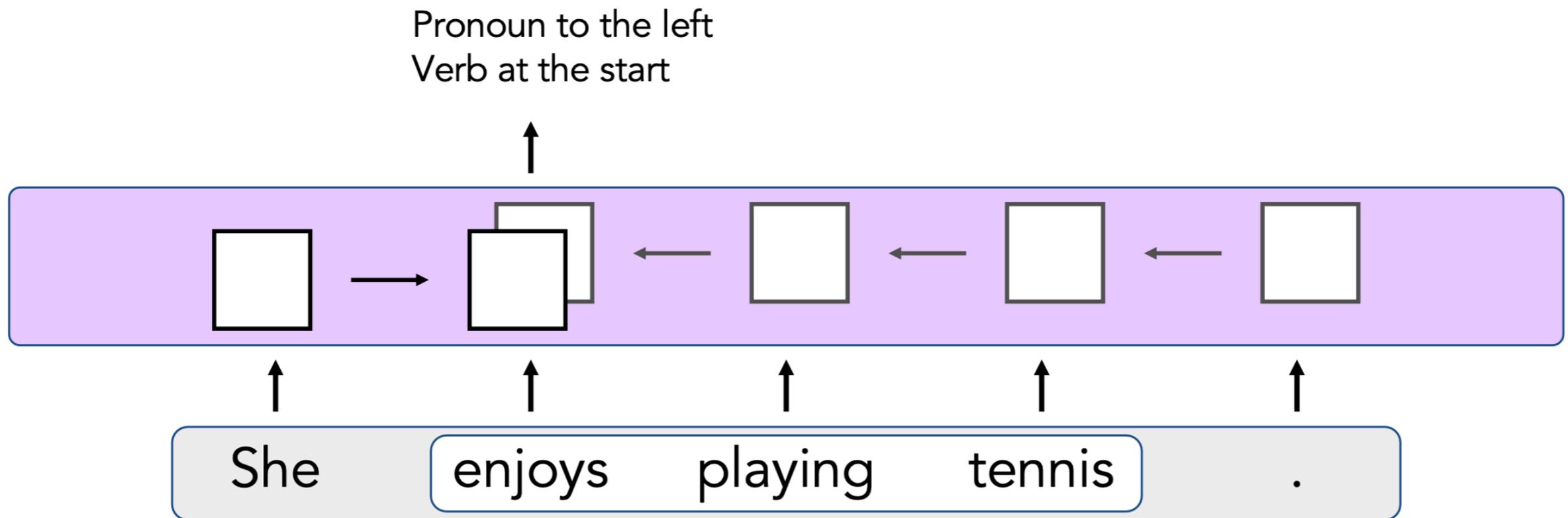
Pronoun to the left



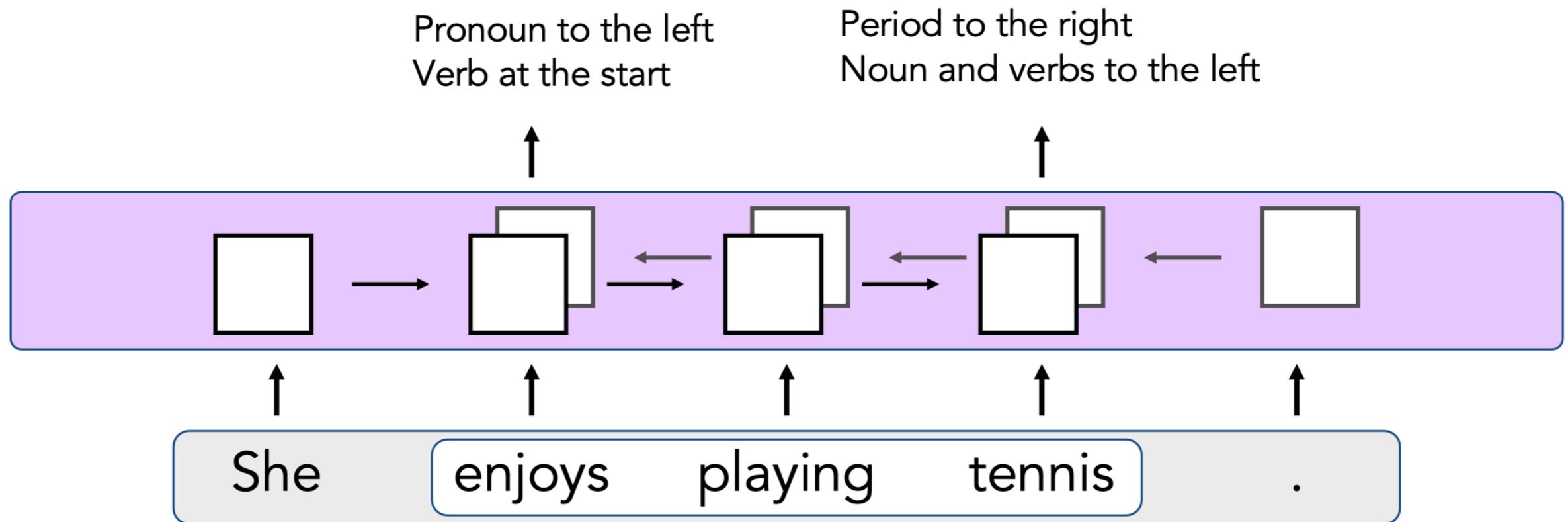
Span-Based Parsing



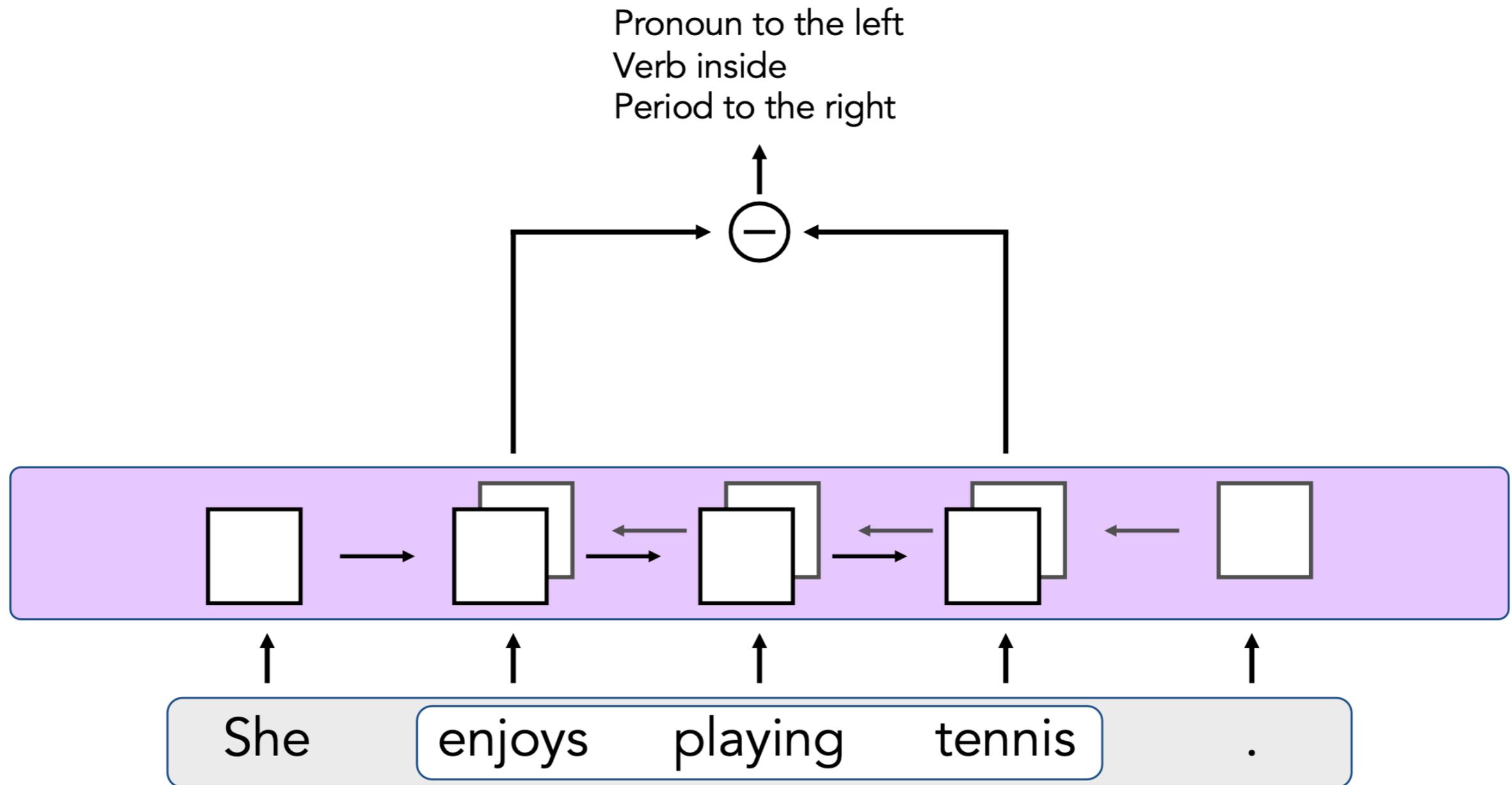
Span-Based Parsing



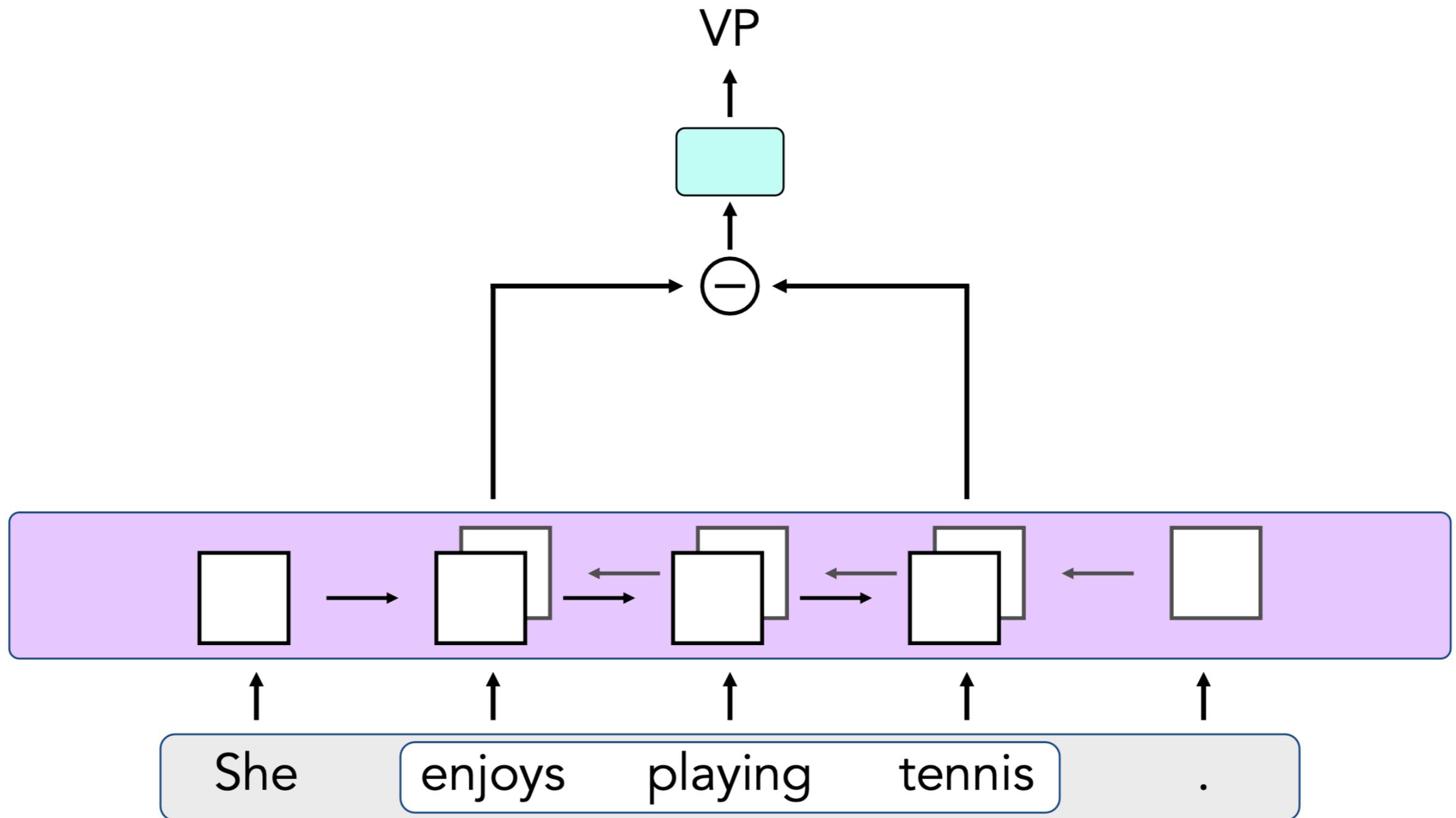
Span-Based Parsing



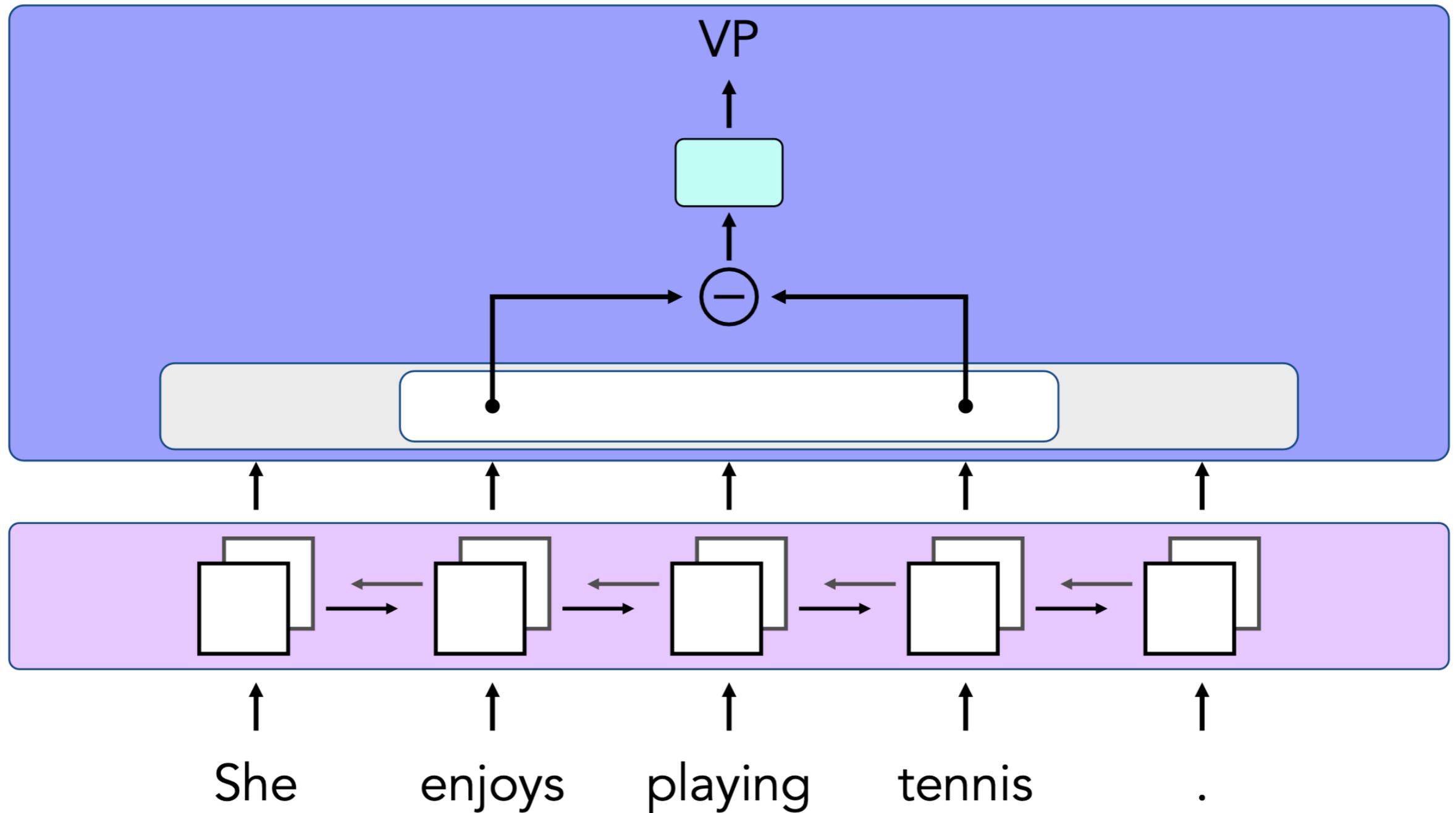
Span-Based Parsing



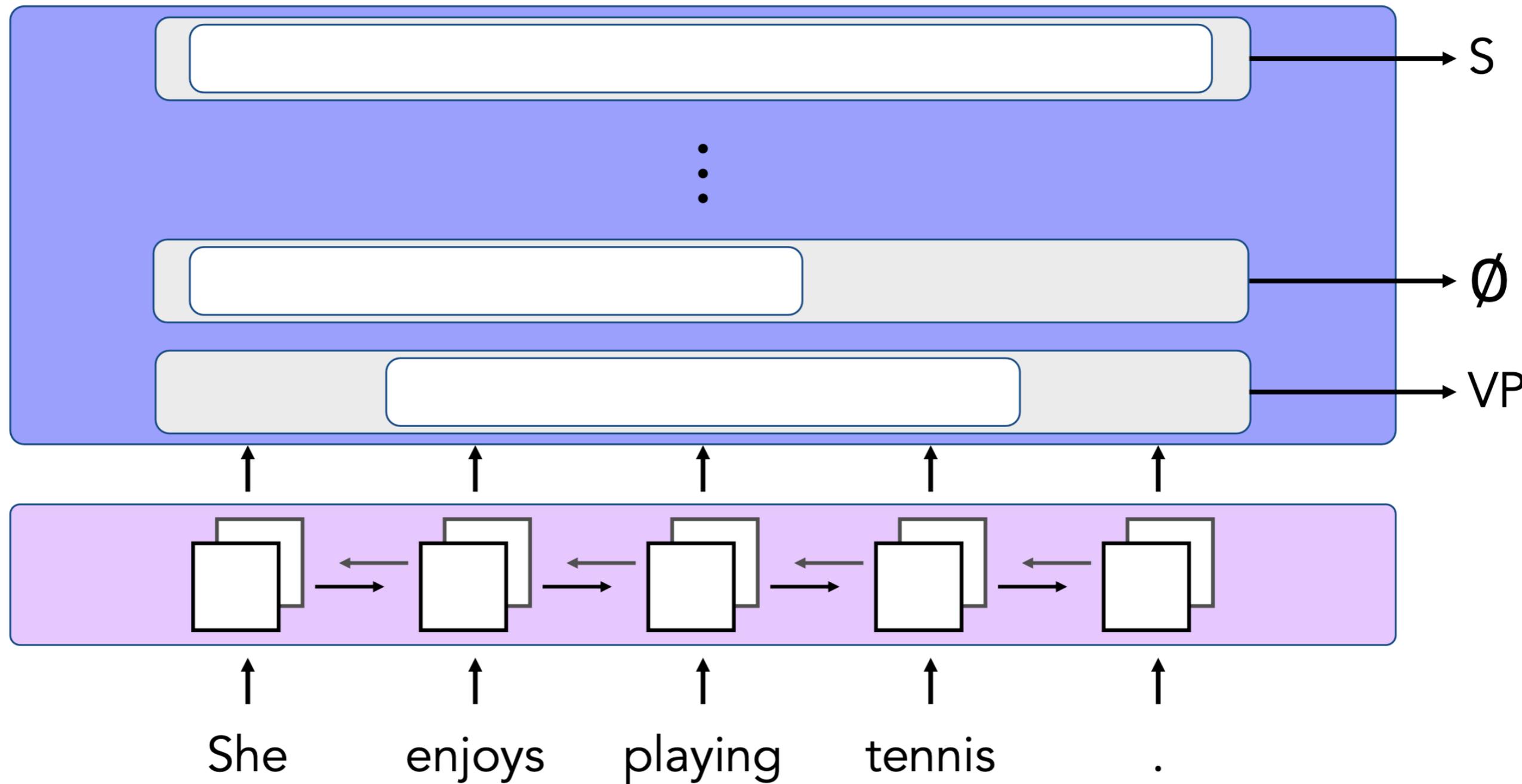
Span-Based Parsing



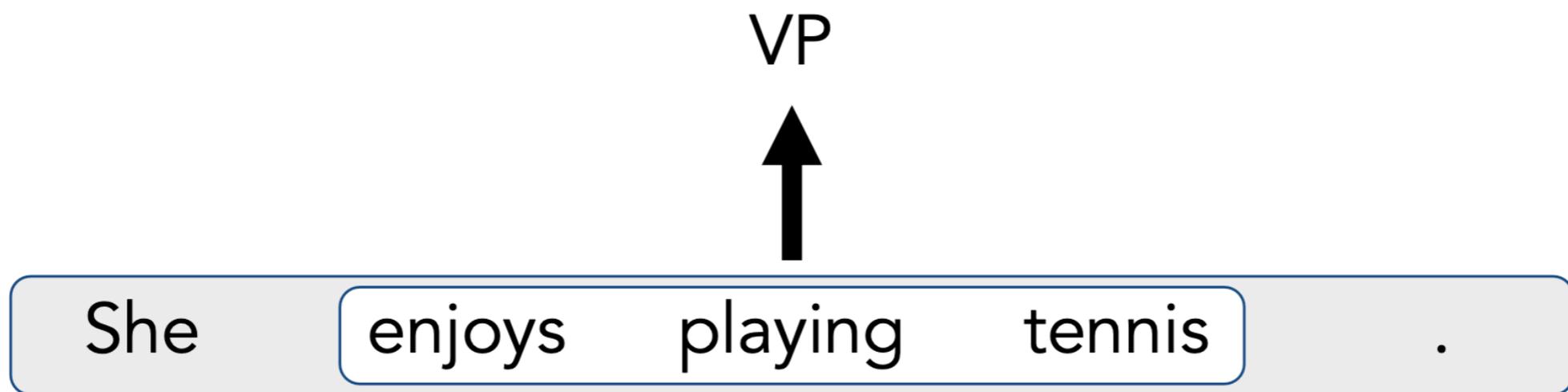
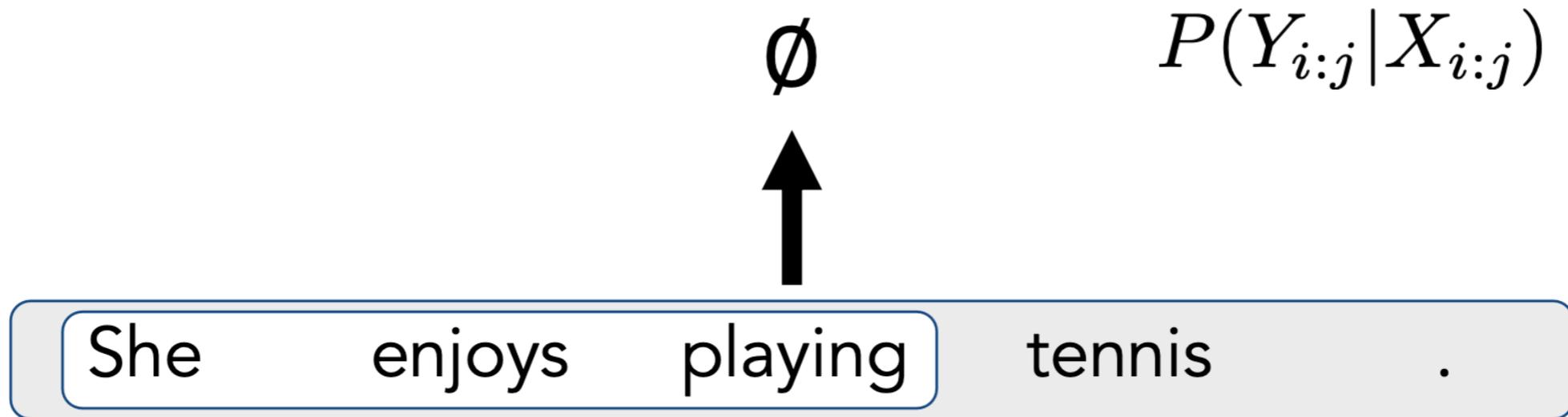
Span-Based Parsing



Span-Based Parsing



Span-Based Parsing



Training: Margin Loss

- Find the best tree using the current model

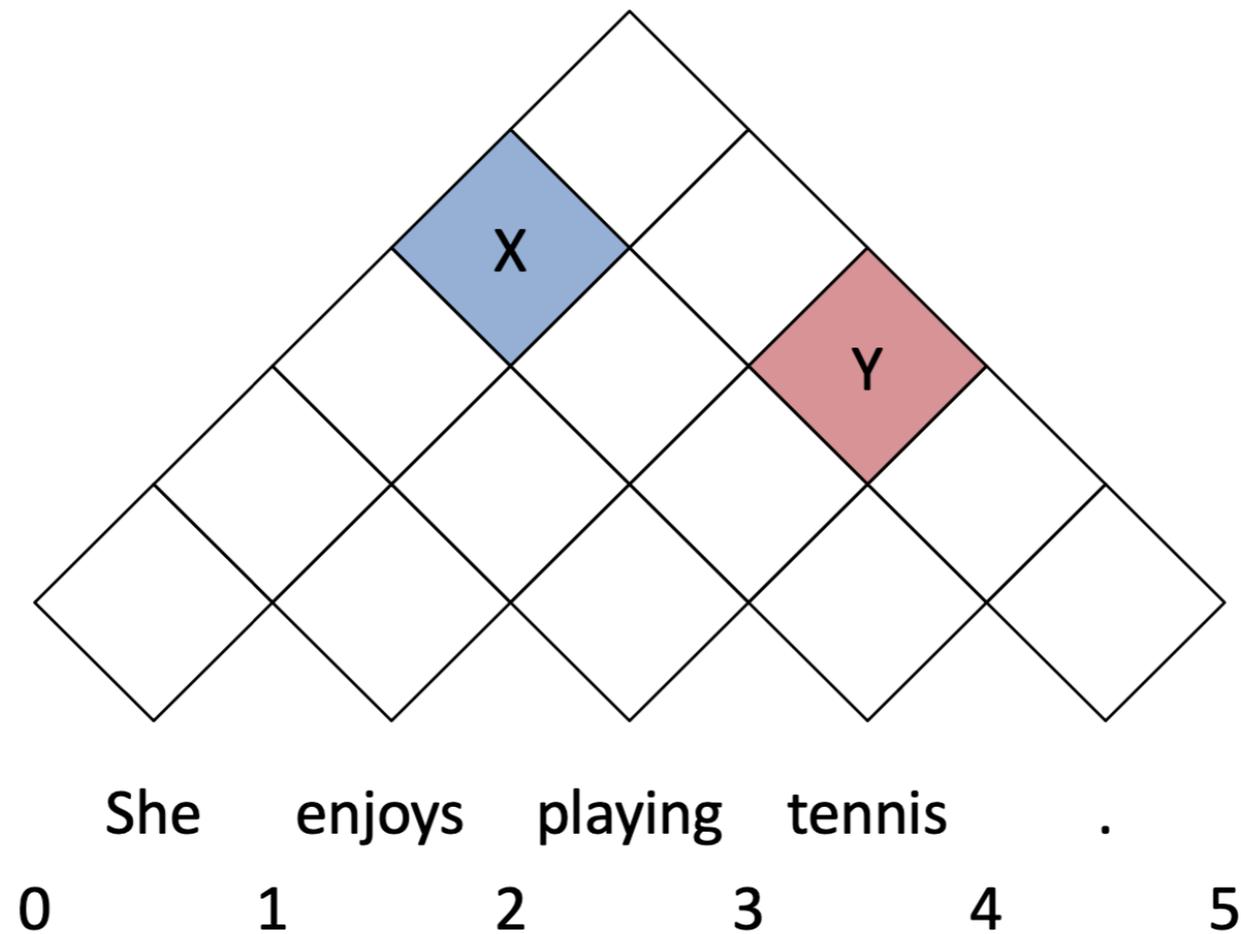
$$\hat{T} = \operatorname{argmax}_T [s_{\text{tree}}(T)].$$

- Margin loss:

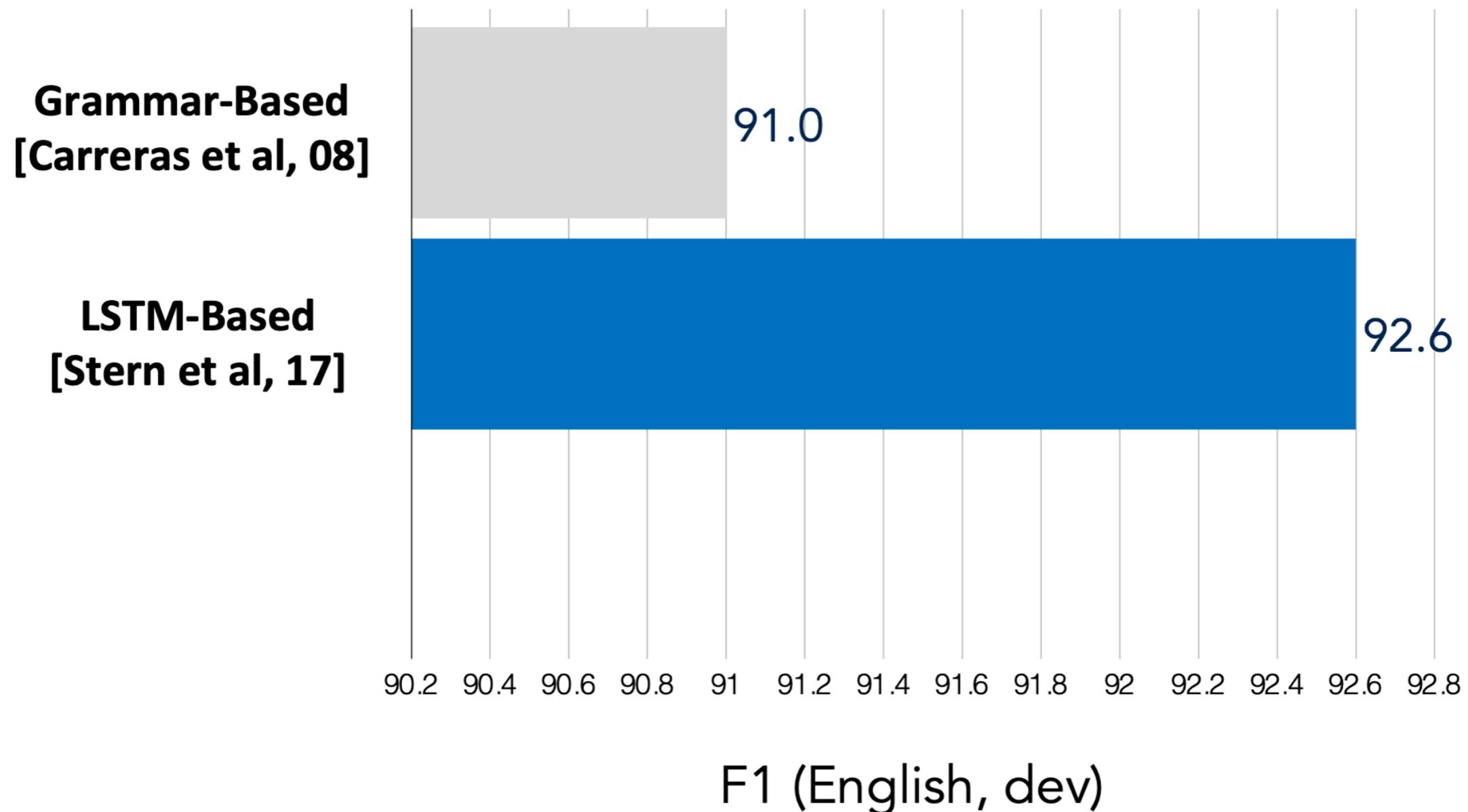
$$\max \left(0, 1 - s_{\text{tree}}(T^*) + s_{\text{tree}}(\hat{T}) \right)$$

Decoding: CYK

- Same as counting-based PCFG
- Use the learned scores for possible spans in the following chart



Improves over non-neural methods



Questions?