NLP Recitation: Parser

Huayun Xu hx2283@columbia.edu

11/16/2020

Topic

- Parsing
  - CKY
  - Earley
- PCFG
- Example

Recap

- 2 grammars:
- Context Free Grammar

A context free grammar is defined by:

- Set of **terminal symbols** Σ.
- Set of **non-terminal symbols** N.
- A **start symbol** $S \in N$.
- Set R of **productions** of the form $A \rightarrow \beta$, where $A \in N$ and $\beta \in (\Sigma \cup N)^*$, *i.e.* β is a string of terminals and non-terminals.

Chomsky Normal Form

A CFG $G=(N, \Sigma, R, S)$ is in Chomsky Normal Form (CNF) if the rules take one of the following forms:

- $A \rightarrow B\ C$, where $A \in N$, $B \in N$, $C \in N$.
- $A \rightarrow b$, where $A \in N$, $b \in \Sigma$.

S → NP VP	V → saw
VP → V NP	P → with
VP → VP PP	D → the
PP → P NP	N → cat
NP → D N	N → tail
NP → NP PP	N → student

Two Approaches to Parsing

- Bottom-up: Start at the words (terminal symbols) and see which subtrees you can build. Then combine these subtrees into larger trees. (Driven by the input sentence.)
  CKY algorithm - requires Grammars in Chomsky Normal Form.

- Top-down: Start at the start symbol (S), try to apply production rules that are compatible with the input.
  (Driven by the grammar - next week)
  Earley algorithm

- Both approaches can be seen as a kind of search problem (next week).

CKY Algorithm

for $i=0...(n\text{-}length)$:
  $j = i+length$
  for $k=i+1...j\text{-}1$:
  ....

S	→ NP VP	NP	→ she
VP	→ V NP	NP	→ glasses
VP	→ VP PP	D	→ the
PP	→ P NP	N	→ cat
NP	→ D N	N	→ glasses
NP	→ NP PP	V	→ saw
		P	→ with

length=5

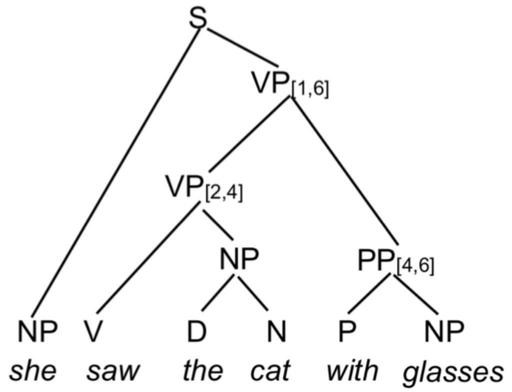
i=0,k=5,j=6

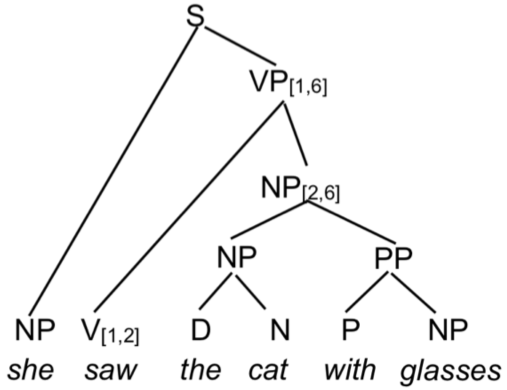
0 *she* 1 *saw* 2 *the* 3 *cat* 4 *with* 5 *glasses*

	she	saw	the	cat	with	glasses
0	NP			S		S
1		V		VP		VP
2			D	NP		NP
3				N		
4					P	PP
5						NP,N
6						

Syntactic Ambiguity

S	→ NP VP	NP	→ she
VP	→ V NP	NP	→ glasses
VP	→ VP PP	D	→ the
PP	→ P NP	N	→ cat
NP	→ D N	N	→ glasses
NP	→ NP PP	V	→ saw
		P	→ with

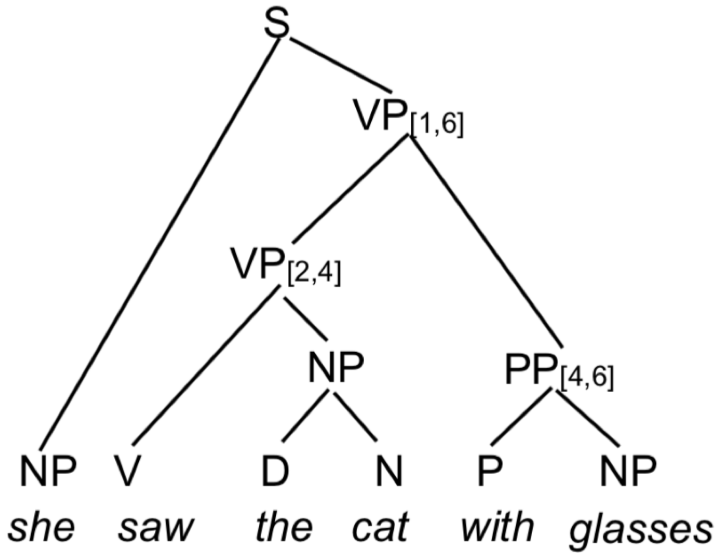


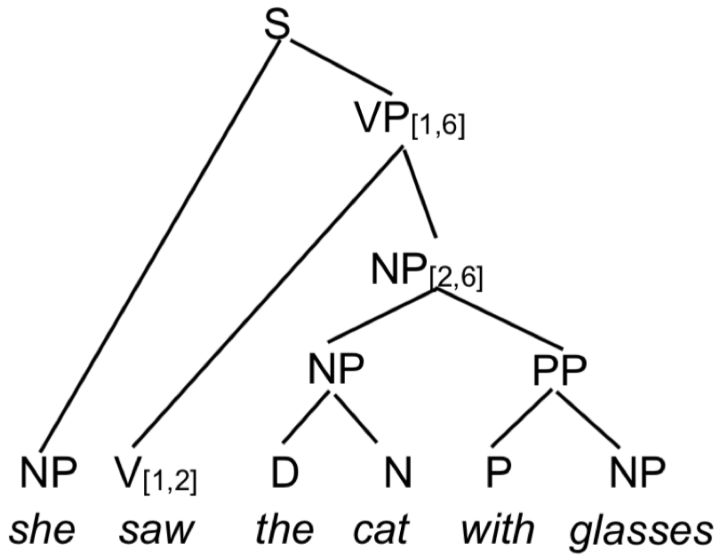


Probabilistic Context Free Grammars (PCFG)

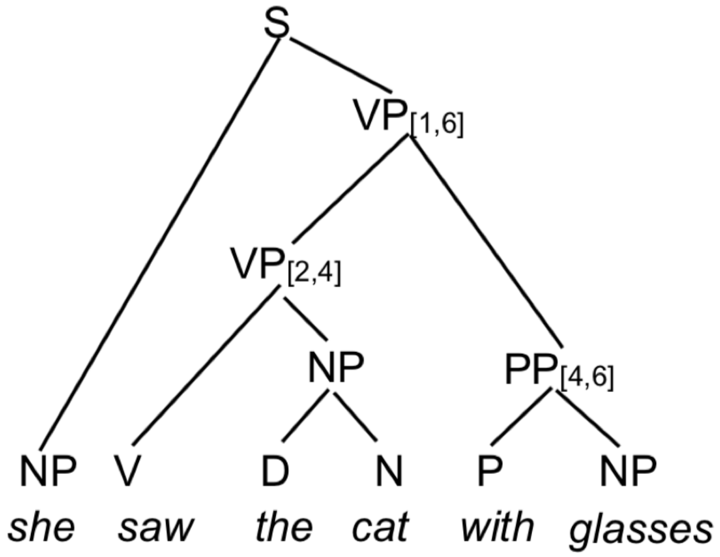
- PCFGs are probabilistic augmentations of CFGs in which each rule is associated with a probability.
- We can estimate PCFG probabilities from a treebank.
- A problem with PCFGs is that they make poor independence assumptions. CFG rules impose an independency assumption on probabilities, resulting in poor modeling of structural dependencies across the parse tree.
- Each PCFG rule is treated as if it were conditionally independent; thus, the probability of a sentence is computed by multiplying the probabilities of each rule in the parse of the sentence.
- Can also use CKY algorithm with PCFGs to also compute the probabilities for a sentence in the grammar.

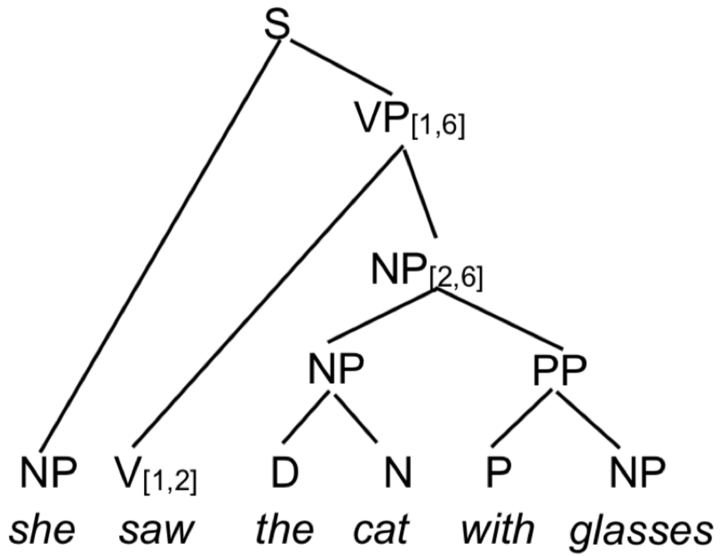
S	$\rightarrow$ NP VP	[1.0]	NP	$\rightarrow$ she	[0.05]
VP	$\rightarrow$ V NP	[0.6]	NP	$\rightarrow$ glasses	[0.05]
VP	$\rightarrow$ VP PP	[0.4]	D	$\rightarrow$ the	[1.0]
PP	$\rightarrow$ P NP	[1.0]	N	$\rightarrow$ cat	[0.3]
NP	$\rightarrow$ D N	[0.7]	N	$\rightarrow$ glasses	[0.7]
NP	$\rightarrow$ NP PP	[0.2]	V	$\rightarrow$ saw	[1.0]
			P	$\rightarrow$ with	[1.0]





S	→ NP VP	[1.0]	NP	→ she	[0.05]
VP	→ V NP	[0.6]	NP	→ glasses	[0.05]
VP	→ VP PP	[0.4]	D	→ the	[1.0]
PP	→ P NP	[1.0]	N	→ cat	[0.3]
NP	→ D N	[0.7]	N	→ glasses	[0.7]
NP	→ NP PP	[0.2]	V	→ saw	[1.0]
			P	→ with	[1.0]





$P(A) = 0.05 * 1 * 1 * 0.3 * 1 * 0.05 * 0.7 * 1 * 0.6 * 0.4 * 1$

$P(B) = 0.05 * 1 * 1 * 0.3 * 1 * 0.05 * 0.7 * 1 * 0.2 * 0.6 * 1$

Earley Parser

- The Earley parser is a top-down approach to figuring out if a sentence is in the grammar, because it starts at the start symbol S and tries to apply production rules that are compatible with the input.
- The Earley parser discards derivations that are incompatible with the sentence. The grammar doesn’t need to be in CNF for the Earley parser to be used.
- The Earley parser has three operations: scan, predict, and complete.

Earley Parser Example

S → NP VP

NP → D N
NP → N

VP → V NP
VP → V PREP NP
VP → V

D → the
D → a

N → cup
N → ball

PREP → with

V → fell

Input sentence: the cup fell

Chart[0]

s0 S → * NP VP [0,0] init
s1 NP → * D N [0,0] predict s0
s2 NP → * N [0,0] predict s0
s3 D → * the [0,0] predict s1
s4 D → * a [0,0] predict s1
s5 N → * cup [0,0] predict s2
s6 N → * ball [0,0] predict s2

Chart[1]

s7 D → the * [0, 1] scan s3
s8 NP → D * N [0, 1] complete s1 with s7
s9 N → * cup [1, 1] predict s8
s10 N → * ball [1, 1] predict s8

Earley Parser Example

S → NP VP

NP → D N
NP → N

VP → V NP
VP → V PREP NP
VP → V

D → the
D → a

N → cup
N → ball

PREP → with

V → fell

Input sentence: the cup fell

Chart[2]

s11 N → cup * [1, 2] scan s9
s12 NP → D N * [0, 2] complete s8 with s11
s13 S → NP * VP [0, 2] complete s0 with s12
s14 VP → * V NP [2, 2] predict s13
s15 VP → * V PREP NP [2, 2] predict s13
s16 VP → * V [2, 2] predict s13
s17 V → * fell [2, 2] predict s14, s15, s16

Earley Parser Example

S → NP VP

NP → D N
NP → N

VP → V NP
VP → V PREP NP
VP → V

D → the
D → a

N → cup
N → ball

PREP → with

V → fell

Input sentence: the cup fell

Chart[3]

s18 V → fell * [2, 3] scan s17
s19 VP → V * NP [2, 3] complete s14 with s18
s20 VP → V * PREP NP [2, 3] complete s15 with s18
s21 VP → V * [2, 3] complete s16 with s18s
22 S → NP VP * [0, 3] complete s13 with s21

Acknowledgments

- Some slides from the course and Raefah Wahid’s notes.