Parsing (part 3)

Chapter 13 J&M'09

Parsing

CKY (or CYK)
Named after John Cocke, Daniel Younger and Tadao Kasami.

- Passive chart parser
- Very efficient (runs in polynomial time; $n^3 \times |G|$)
- Requires grammar transformation.

CKY requires grammars in Chomsky Normal Form (CNF)

In CNF, all rules must be of one of the following forms:

$X \rightarrow Y\,Z$
$K \rightarrow w$ (where 'w' is a word token)

This **binarization** step is crucial for efficient parsing.

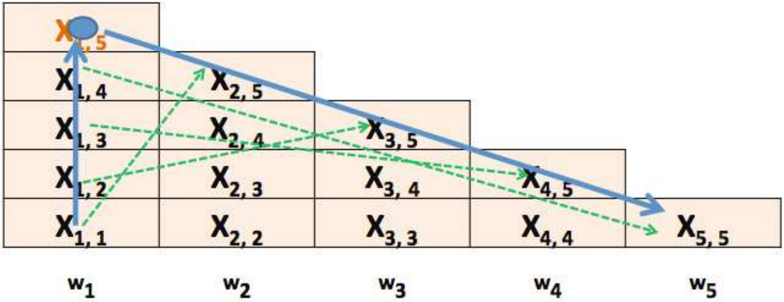
Conversion to CNF

- Copy all conforming rules to new grammar
- Eliminate unit productions:
  Rules like **NP** → **PN** and **PN** → **Tom**
  become a single rule **NP** → **Tom**
- Convert branching rules:
  Rules like **VP** → **DTV NP NP** become

  VP → **DVP NP**
  DVP → **DTV NP**

Parsing

For input w_1 ... w_t, build parse triangle:



Each row corresponds to a string of ascending length.
Each cell corresponds to all of the possible categories for the corresponding span. For example, in

(1) Doves dove

Each cell of row 1 would be {N,V}.

Parsing

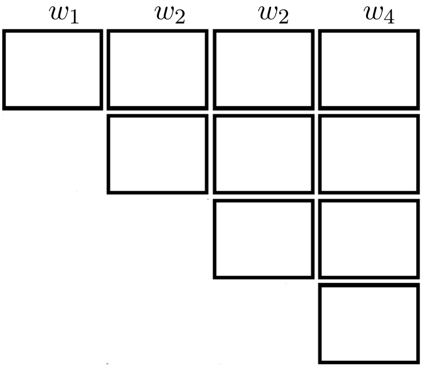
Input: *Tom saw a friend from Australia*

Span length						
6	S					
5	–	VP				
4	S	–	NP			
3	–	VP	–	N		
2	–	–	NP	–	PP	
1	NP	TV, N	DT	N,TV	P	NP
	Tom	saw	a	friend	from	Australia

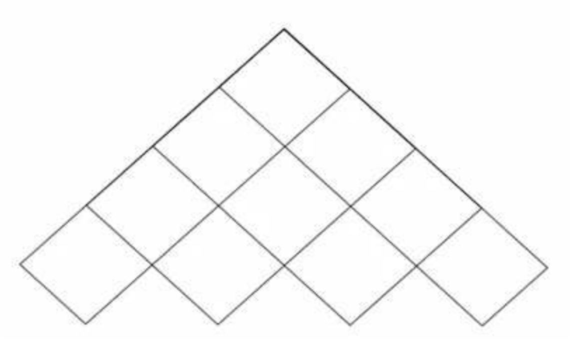
0 1 2 3 4 5 6

S → NP VP	VP → TV NP	VP → VP PP
NP → DT N	N → N PP	PP → P NP
NP → Tom	TV → saw	DT → a
NP → Australia	N → saw	TV → friend
N → friend	P → from	

Sometimes, the chart is shown as a parse triangle:



Sometimes, the chart is rotated:



w_1 w_2 w_2 w_4

Exercise:
Show that 'b a a b a' is parsed by CKY and the following grammar.

$S \rightarrow A\ B$
$S \rightarrow B\ C$
$A \rightarrow B\ A$
$A \rightarrow a$
$B \rightarrow C\ C$
$B \rightarrow b$
$C \rightarrow A\ B$
$C \rightarrow a$

Probabilistic Context-free Grammar (PCFG)

Rules are augmented with a probability:

NP → PN	[0.35]
NP → PRN	[0.30]
NP → DT N	[0.20]
NP → N	[0.15]
DT → *that*	[0.10]
DT → *a*	[0.30]
DT → *the*	[0.60]

The total probability for rules with the same left-hand side is 1:

$$\sum_{\beta} P(X \rightarrow \beta) = 1$$

Where:

- β is any sequence of (terminal/non-terminal) symbols.
- $P(X \rightarrow \beta)$ = probability of rule $X \rightarrow \beta$
  For example: $P(\text{NP} \rightarrow \text{PN}) = 0.35$

Statistical Parsing

T is a parse tree and $S = w_1...w_n$ a token sequence. Bayes' rule:

$$P(T|w_1...w_n) = \frac{P(w_1...w_n|T) \times P(T)}{P(w_1...w_n)}$$

But since $P(w_1...w_n|T)$ is always 1, then:

$$P(T|w_1...w_n) = \frac{P(T)}{P(w_1...w_n)}$$

In particular, we are interested in the most likely T for S:

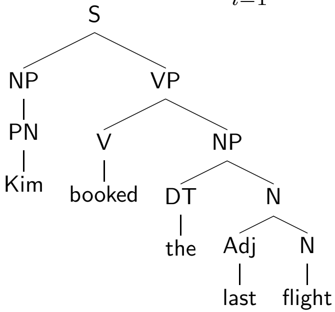
$$\hat{T} = argmax_{T:S=yield(T)} \frac{P(T)}{P(w_1...w_n)}$$

But since $P(w_1...w_n)$ is constant over all T's for a given S then:

$$\hat{T} \approx argmax_{T:S=yield(T)} P(T)$$

Statistical Parsing

$$P(T) = \prod_{i=1}^{n} P(X_i \rightarrow Y_1 ... Y_n)$$



Rule	P	Rule	P	Rule	P	Rule	P
S → NP VP	0.8	S → VP	0.2	VP → V	0.3	VP → V NP	0.5
VP → V NP NP	0.2	NP → DT N	0.2	NP → PN	0.35	NP → PRN	0.3
NP → N	0.15	N → Adj N	0.4	N → evening	0.2	N → flight	0.1
PN → Kim	0.15	V → booked	0.4	Adj → last	0.1	DT → the	0.4

$$P(T) = 0.8 \times 0.35 \times 0.15 \times 0.5 \times 0.4 \times 0.2 \times 0.4 \times 0.4 \times 0.1 \times 0.1 = 2.7 \times 10^{-6}$$

Statistical Parsing

We can use Treebanks to estimate the probabilities for CFG rules:

$$P(X \rightarrow Y_1...Y_n) = \frac{Count(X \rightarrow Y_1...Y_n)}{Count(X)}$$

Example:

$[_S$ $[_{NP}$ $[_{DT}$ This] $[_N$ text]] $[_{VP}$ $[_V$ is] $[[_{Adv}$ just] $[_{NP}$ $[_{DT}$ an] $[_N$ example]]]]]. $[_S$ $[_{NP}$ $[_{PRN}$ I]] $[_{VP}$ $[_V$ made] $[_{NP}$ $[_{PRN}$ it]] $[_{RP}$ up]]].

$P(\text{NP} \rightarrow \text{DT N}) = \frac{2}{4} = 0.5$

Penn Treebank:

```
(S (NP-SBJ-1 Jones)
    (VP followed
       (NP him)
       (PP-DIR into
             (NP the front room))))
```

Treebanks

- Linguistic Data Consortium (LDC)
- European Language Resources Association (ELRA)
- Stanford list
- NLTK data
- Others

Statistical Parsing

Or you can use a statistical parser to automatically parse a corpus you wish to use.

- Stanford Parser:

```
java -cp stanford-parser.jar:stanford-parser-3.4.1-models.jar
edu.stanford.nlp.parser.lexparser.LexicalizedParser -outputFormat penn
edu/stanford/nlp/models/lexparser/englishPCFG.ser.gz  input.txt >
output.txt
```

- Berkeley Parser: (demo)

```
java -jar berkeleyParser/BerkeleyParser-1.7.jar -gr
berkeleyParser/eng_sm6.gr < input.txt > output.txt
```

- More here

Parsing PCFG with the CKY parser

We need:

- A CNF grammar augmented with probabilities
- A way to resolve ambiguity:
  if there are two categories of the same type in the same span, then discard the less likely one.

$S_{[.7\times.1\times.00001=.0000007]}$						
$S_{[.7\times.1\times.0001=.000007]}$	$VP_{[.5\times.2\times.0001=.00001]}$ $\cancel{VP_{[.1\times.006\times.01=.000006]}}$					
–	$VP_{[.5\times.2\times.001=.0001]}$ $\cancel{VP_{[.1\times.006\times.072=.00004]}}$	$NP_{[.5\times.4\times.0009=.0001]}$				
$S_{[.7\times.1\times.006=.0004]}$	–	$NP_{[.5\times.4\times.006=.001]}$	$N_{[.3\times.3\times.01=.0009]}$			
–	$VP_{[.5\times.2\times.06=.006]}$	–	$N_{[.3\times.3\times.072=.006]}$	$PP_{[.9\times.4\times.03=.01]}$		
–	–	$NP_{[.5\times.4\times.3=.06]}$	–	$PP_{[.9\times.4\times.2=.072]}$	$NP_{[.5\times.2\times.3=.03]}$	
$NP_{[.1]}$	$Adj_{[.1]}$ $TV_{[.2]}$	$DT_{[.4]}$	$N_{[.3]}$	$P_{[.4]}$	$DT_{[.2]}$ $NP_{[.2]}$	$TV_{[.05]}$ $N_{[.3]}$
Mary	attacked	a	farmer	with	her	axe

S → NP VP	[.7]	VP → DVP NP	[.3]	VP → VP PP	[.1]
VP → TV NP	[.5]	NP → DT N	[.5]	NP → Mary	[.1]
NP → her	[.2]	DT → a	[.4]	DT → her	[.2]
N → Adj N	[.1]	N → N PP	[.3]	N → farmer	[.3]
N → axe	[.3]	Adj → attacked	[.1]	PP → P NP	[.9]
P → with	[.4]	TV → attacked	[.2]	TV → axe	[.05]

Evaluating PCFGS (PARSEVAL)

How do the constituents in the hypothesis parse tree match the constituents in a hand-labeled 'gold standard' (reference) parse tree?

T_C = set of constituents for S according to reference
T_G = set of constituents hypothesized for S

1. **labeled precision**

$$\frac{|T_C \cap T_G|}{|T_G|} = \frac{\text{\#of correctly identified constituents}}{\text{\#of constituents hypothesized}} = \frac{t_p}{t_p + f_p}$$

2. **labeled recall**

$$\frac{|T_C \cap T_G|}{|T_C|} = \frac{\text{\#of correctly identified constituents}}{\text{\#of constituents in reference}} = \frac{t_p}{t_p + f_n}$$

Rule of thumb: as precision increases, recall drops and vice versa.

Often precision and recall are reported as a single number:

$$\textbf{F-measure} : F_\beta = \frac{(\beta^2 + 1) \times P \times R}{\beta^2 \times P + R}$$

$\beta > 1$ favors Recall
$\beta < 1$ favors Precision

Problems with PCFGs

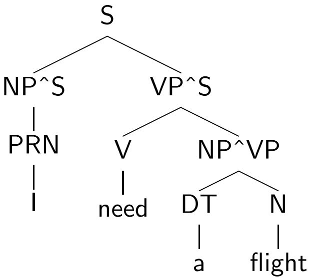
Poor independence assumption

Linguistic structures are not independent from each other. For example, the distribution of NP → PN is unbalanced:

91% of subject phrases are pronouns
34% of object phrases are pronouns

Solution: add information about the mother node in the daughters

Parent annotation: NP^S → PRN [.91] vs. NP^VP → PRN [.34]



PROBLEMS WITH PCFGs (continued)

Lack of lexical conditioning

Lexical items are important to resolve attachment ambiguities.

(2) a. * [Sam [dumped [the box into the bin]]].
    b. [Sam [dumped [the box] [into the bin]]].

(3) a. [Sam [dumped [the box in the bin]]].
    b. [Sam [dumped [the box] [in the bin]]].

(4) a. Sam likes [[green vegetables] and [music]].
    b. *Sam likes [green [vegetables and music]].

(5) a. *I need some [fresh [air and sunshine]].
    b. I need some [[fresh air] and [sunshine]].

Solution: add information about the token in the mother node

Dealing with the lack of lexical conditioning

grammar lexicalizaton