

Probabilistic Parsing

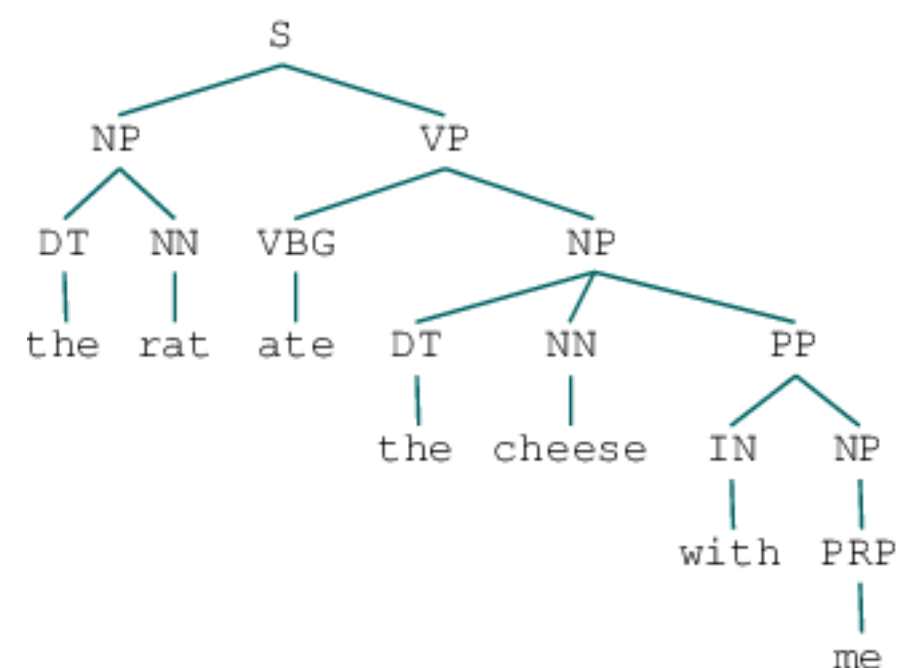
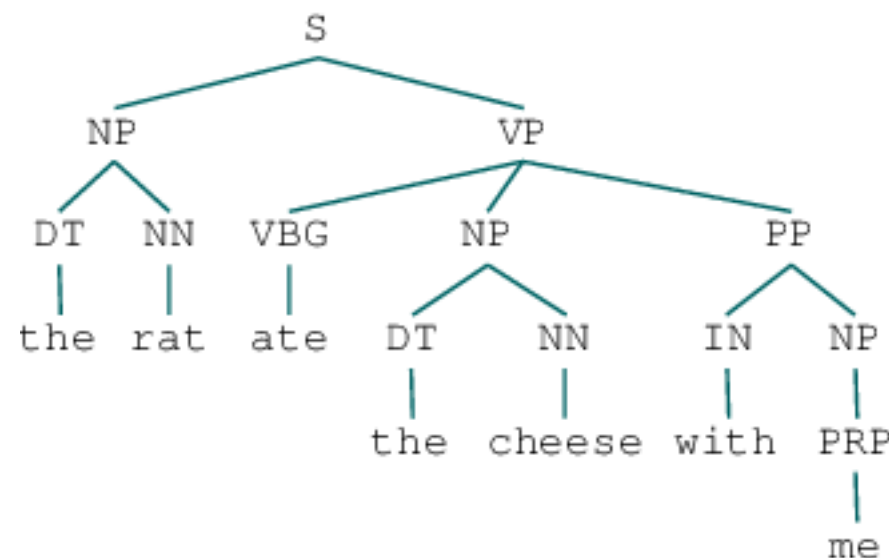
COMP90042 LECTURE 18



THE UNIVERSITY OF
MELBOURNE

Ambiguity in parsing

- Context-free grammars assign hierarchical structure to language
 - * Linguistic notion of a '*syntactic constituent*'
 - * Formulated as generating all strings in the language; or
 - * Predicting the structure(s) for a given string
- Raises problem of ambiguity, e.g., which is better?



Outline

- Probabilistic context-free grammars (PCFGs)
- Parsing using dynamic programming
- Limitations of ‘context-free’ assumption and some solutions:
 - * parent annotation
 - * head lexicalisation

Basics of Probabilistic CFGs

- As for CFGs, same symbol set:
 - * Terminals: words such as *book*
 - * Non-terminal: syntactic labels such as NP or NN
- Same productions (rules)
 - * LHS non-terminal \rightarrow ordered list of RHS symbols
- In addition, store a **probability** with each production
 - * $NP \rightarrow DT\ NN$ [p = 0.45]
 - * $NN \rightarrow cat$ [p = 0.02]
 - * $NN \rightarrow leprechaun$ [p = 0.00001]
 - * ...

Probabilistic CFGs

- Probability values denote *conditional*
 - * $\Pr(\text{RHS} \mid \text{LHS})$
- Consequently they:
 - * must be positive values, between 0 and 1
 - * must sum to one for given LHS
- E.g.,
 - * $\text{NN} \rightarrow \text{aadvark} \quad [p = 0.0003]$
 - * $\text{NN} \rightarrow \text{cat} \quad [p = 0.02]$
 - * $\text{NN} \rightarrow \text{leprechaun} \quad [p = 0.0001]$
 - * $\sum_x \Pr(\text{NN} \rightarrow x \mid \text{NN}) = 1$

A Probabilistic grammar

Grammar		Lexicon
$S \rightarrow NP VP$	[.80]	$Det \rightarrow that [.10] \mid a [.30] \mid the [.60]$
$S \rightarrow Aux NP VP$	[.15]	$Noun \rightarrow book [.10] \mid \text{flights} [.30]$
$S \rightarrow VP$	[.05]	$\mid meal [.015] \mid money [.05]$
$NP \rightarrow Pronoun$	[.35]	$\mid flight [.40] \mid dinner [.10]$
$NP \rightarrow Proper-Noun$	[.30]	$Verb \rightarrow book [.30] \mid include [.30]$
$NP \rightarrow Det Nominal$	[.20]	$\mid prefer [.40]$
$NP \rightarrow Nominal$	[.15]	$Pronoun \rightarrow I [.40] \mid she [.05]$
$Nominal \rightarrow Noun$	[.75]	$\mid me [.15] \mid you [.40]$
$Nominal \rightarrow Nominal Noun$	[.20]	$Proper-Noun \rightarrow Houston [.60]$
$Nominal \rightarrow Nominal PP$	[.05]	$\mid NWA [.40]$
$VP \rightarrow Verb$	[.35]	$Aux \rightarrow does [.60] \mid can [.40]$
$VP \rightarrow Verb NP$	[.20]	$Preposition \rightarrow from [.30] \mid to [.30]$
$VP \rightarrow Verb NP PP$	[.10]	$\mid on [.20] \mid near [.15]$
$VP \rightarrow Verb PP$	[.15]	$\mid through [.05]$
$VP \rightarrow Verb NP NP$	[.05]	
$VP \rightarrow VP PP$	[.15]	
$PP \rightarrow Preposition NP$	[1.0]	

Source JM3, Fig 12.1

Stochastic Generation with PCFGs

Déjà vu, it's almost the same as for CFG, with one twist:

1. Start with S, the sentence symbol
2. Choose a rule with S as the LHS
 - * **Randomly select a RHS** according to $\text{Pr}(\text{RHS} \mid \text{LHS})$
e.g., $S \rightarrow VP$
 - * Apply this rule, e.g., substitute VP for S
3. Repeat step 2 for each non-terminal in the string
(here, VP)
4. Stop when no non-terminals remain

Gives us a tree, as before, with a sentence as the yield

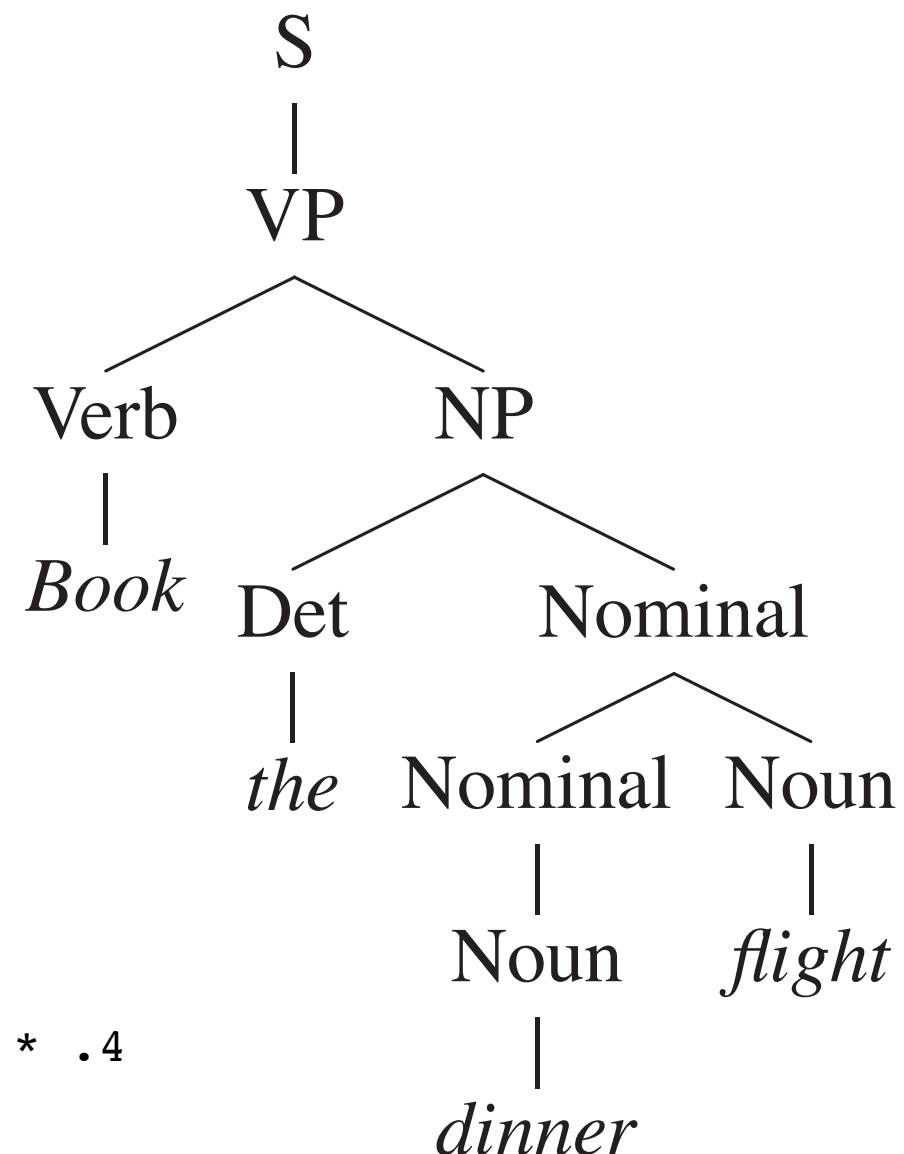
How likely is a tree?

- Given a tree, we can compute its probability
 - Decomposes into probability of each production

- E.g., for tree on right,

* $P(\text{tree}) =$

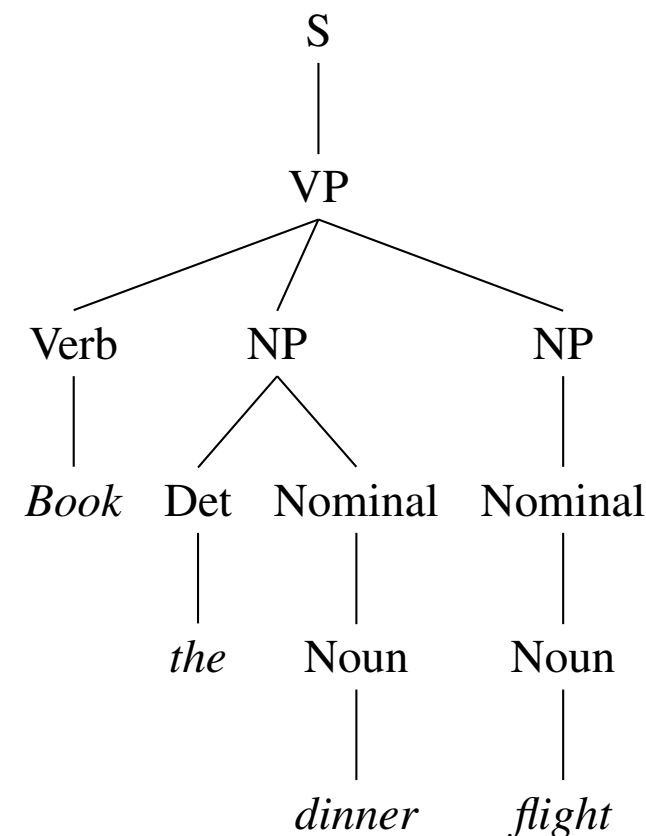
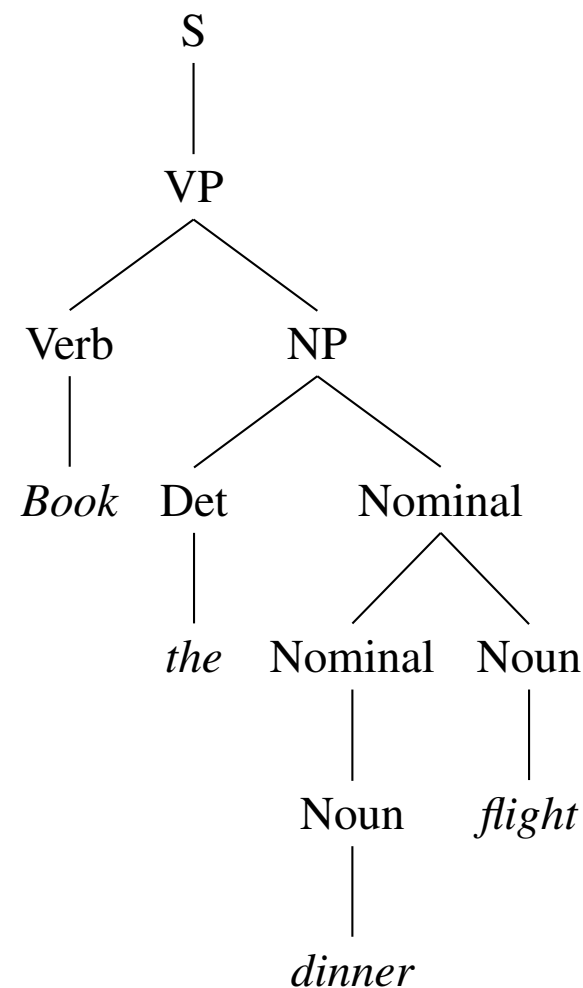
$P(S \rightarrow VP) \times$
 $P(VP \rightarrow \text{Verb NP}) \times$
 $P(\text{Verb} \rightarrow \textit{Book}) \times$
 $P(NP \rightarrow \text{Det Nominal}) \times$
 $P(\text{Det} \rightarrow \textit{the}) \times$
 $P(\text{Nominal} \rightarrow \text{Nominal Noun}) \times$
 $P(\text{Nominal} \rightarrow \text{Noun}) \times$
 $P(\text{Noun} \rightarrow \textit{dinner}) \times$
 $P(\text{Noun} \rightarrow \textit{flight}) = 2.16 \times 10^{-6}$



I.e., $.05 * .2 * .3 * .2 * .6 * .2 * .75 * .1 * .4$

Resolving parse ambiguity

- Can select between different trees based on $P(T)$



Source: JM3
Fig 12.2

Mistakes in
textbook

$.05 * .05 * .20 * .15 * .75 * .75 * .30 * .60 * .10 * .40$

- $P = 2.16 \times 10^{-6}$

$$P = 3.04 \times 10^{-7}$$

Parsing PCFGs

- Instead of selecting between two trees, can we select a tree from the set of all possible trees?
- Before we looked at
 - * CYK and Early
 - * for unweighted grammars (CFGs)
 - * finds **all possible trees**
- But there are often 1000s, many completely nonsensical $\arg \max_T \text{ s.t. } \text{yield}(T)=\mathbf{w} P(T)$
- Can we solve for the **most probable tree**

CYK for PCFGS

- CYK finds *all trees* for a sentence; we want **best** tree
- Prob. CYK follows similar process to standard CYK
- Convert grammar to Chomsky Normal Form (CNF)

* E.g., $VP \rightarrow \text{Verb NP NP}$ **[0.05]**

becomes $VP \rightarrow \text{Verb NP+NP}$ **[]**
 $NP+NP \rightarrow \text{NP NP}$ **[]**

where NP+NP is a new symbol.

- Issues with unary productions (*see ipython notebook*)

Prob. CYK

```

function PROBABILISTIC-CYK(words, grammar) returns most probable parse
                                     and its probability

for  $j \leftarrow$  from 1 to LENGTH(words) do
    for all  $\{ A \mid A \rightarrow words[j] \in grammar \}$ 
         $table[j-1, j, A] \leftarrow P(A \rightarrow words[j])$ 
    for  $i \leftarrow$  from  $j-2$  downto 0 do
        for  $k \leftarrow i+1$  to  $j-1$  do
            for all  $\{ A \mid A \rightarrow BC \in grammar,$ 
                        and  $table[i, k, B] > 0$  and  $table[k, j, C] > 0 \}$ 
                if ( $table[i, j, A] < P(A \rightarrow BC) \times table[i, k, B] \times table[k, j, C]$ ) then
                     $table[i, j, A] \leftarrow P(A \rightarrow BC) \times table[i, k, B] \times table[k, j, C]$ 
                     $back[i, j, A] \leftarrow \{k, B, C\}$ 
    return BUILD_TREE( $back[1, LENGTH(words), S]$ ),  $table[1, LENGTH(words), S]$ 

```

Figure 12.3 The probabilistic CYK algorithm for finding the maximum probability parse

chart now stores
probabilities for each
span and symbol

CYK can be thought
of as storing all events
with probability = 1

function CKY-PARSE(*words*, *grammar*) **returns** *table*

for *j* ← **from** 1 **to** LENGTH(*words*) **do**

for all {*A* | $A \rightarrow \text{words}[j] \in \text{grammar}$ }

$\text{table}[j-1, j] \leftarrow \text{table}[j-1, j] \cup A$

for *i* ← **from** *j* - 2 **downto** 0 **do**

for *k* ← *i* + 1 **to** *j* - 1 **do**

for all {*A* | $A \rightarrow BC \in \text{grammar}$ **and** *B* ∈ *table*[*i*, *k*] **and** *C* ∈ *table*[*k*, *j*]}

$\text{table}[i, j] \leftarrow \text{table}[i, j] \cup A$

Figure 11.5 The CKY algorithm.

validity test now looks to
see that the child chart cells
have non-zero probability

function PROBABILISTIC-CKY(*words*, *grammar*) **returns** most probable parse
and its probability

for *j* ← **from** 1 **to** LENGTH(*words*) **do**

for all {*A* | $A \rightarrow \text{words}[j] \in \text{grammar}$ }

$\text{table}[j-1, j, A] \leftarrow P(A \rightarrow \text{words}[j])$

for *i* ← **from** *j* - 2 **downto** 0 **do**

for *k* ← *i* + 1 **to** *j* - 1 **do**

for all {*A* | $A \rightarrow BC \in \text{grammar}$,

and *table*[*i*, *k*, *B*] > 0 **and** *table*[*k*, *j*, *C*] > 0 }

if (*table*[*i*, *j*, *A*] < $P(A \rightarrow BC) \times \text{table}[i, k, B] \times \text{table}[k, j, C]$) **then**

$\text{table}[i, j, A] \leftarrow P(A \rightarrow BC) \times \text{table}[i, k, B] \times \text{table}[k, j, C]$

back[*i*, *j*, *A*] ← {*k*, *B*, *C*}

return BUILD_TREE(*back*[1, LENGTH(*words*), *S*]), *table*[1, LENGTH(*words*), *S*])

Instead of storing set
of symbols, store the
probability of best scoring
tree fragment covering span
[*i*, *j*] with root symbol *A*

Overwrite lower scoring
analysis if this one is better,
and record the best production.

Figure 12.3 The probabilistic CKY algorithm for finding the maximum probability parse

Illustration

chopsticks

[illegible]

		$\psi(\cdot)$	$\exp \psi(\cdot)$
S	\rightarrow NP VP	0	1
NP	\rightarrow NP PP	-1	$\frac{1}{2}$
	$\rightarrow we$	-2	$\frac{1}{4}$
	$\rightarrow sushi$	-3	$\frac{1}{8}$
	$\rightarrow chopsticks$	-3	$\frac{1}{8}$
PP	\rightarrow IN NP	0	1
IN	$\rightarrow with$	0	1
VP	\rightarrow V NP	-1	$\frac{1}{2}$
	\rightarrow VP PP	-2	$\frac{1}{4}$
	\rightarrow MD V	-2	$\frac{1}{4}$
V	$\rightarrow eat$	0	1

Illustration

[illegible]

		$\psi(\cdot)$	$\exp \psi(\cdot)$
S	\rightarrow NP VP	0	1
NP	\rightarrow NP PP	-1	$\frac{1}{2}$
	$\rightarrow we$	-2	$\frac{1}{4}$
	$\rightarrow sushi$	-3	$\frac{1}{8}$
	$\rightarrow chopsticks$	-3	$\frac{1}{8}$
PP	\rightarrow IN NP	0	1
IN	$\rightarrow with$	0	1
VP	\rightarrow V NP	-1	$\frac{1}{2}$
	\rightarrow VP PP	-2	$\frac{1}{4}$
	\rightarrow MD V	-2	$\frac{1}{4}$
V	$\rightarrow eat$	0	1

Illustration

[illegible]

		$\psi(\cdot)$	$\exp \psi(\cdot)$
S	\rightarrow NP VP	0	1
NP	\rightarrow NP PP	-1	$\frac{1}{2}$
	$\rightarrow we$	-2	$\frac{1}{4}$
	$\rightarrow sushi$	-3	$\frac{1}{8}$
	$\rightarrow chopsticks$	-3	$\frac{1}{8}$
PP	\rightarrow IN NP	0	1
IN	$\rightarrow with$	0	1
VP	\rightarrow V NP	-1	$\frac{1}{2}$
	\rightarrow VP PP	-2	$\frac{1}{4}$
	\rightarrow MD V	-2	$\frac{1}{4}$
V	$\rightarrow eat$	0	1

Illustration

we eat sushi with chopsticks

NP 1				
	V 1			
		NP 1/8		

		$\psi(\cdot)$	$\exp \psi(\cdot)$
S	\rightarrow NP VP	0	1
NP	\rightarrow NP PP	-1	$\frac{1}{2}$
	$\rightarrow we$	-2	$\frac{1}{4}$
	$\rightarrow sushi$	-3	$\frac{1}{8}$
	$\rightarrow chopsticks$	-3	$\frac{1}{8}$
PP	\rightarrow IN NP	0	1
IN	$\rightarrow with$	0	1
VP	\rightarrow V NP	-1	$\frac{1}{2}$
	\rightarrow VP PP	-2	$\frac{1}{4}$
	\rightarrow MD V	-2	$\frac{1}{4}$
V	$\rightarrow eat$	0	1

Illustration

we eat sushi with chopsticks

NP 1				
	V 1	VP 1/8 * 1 * 1/2 = 1/16		
		NP 1/8		

		$\psi(\cdot)$	$\exp \psi(\cdot)$
S	\rightarrow NP VP	0	1
NP	\rightarrow NP PP	-1	$\frac{1}{2}$
	$\rightarrow we$	-2	$\frac{1}{4}$
	$\rightarrow sushi$	-3	$\frac{1}{8}$
	$\rightarrow chopsticks$	-3	$\frac{1}{8}$
PP	\rightarrow IN NP	0	1
IN	$\rightarrow with$	0	1
VP	\rightarrow V NP	-1	$\frac{1}{2}$
	\rightarrow VP PP	-2	$\frac{1}{4}$
	\rightarrow MD V	-2	$\frac{1}{4}$
V	$\rightarrow eat$	0	1

Illustration

we eat sushi with chopsticks

NP 1				
	V 1	VP 1/16		
		NP 1/8		
			IN 1	

		$\psi(\cdot)$	$\exp \psi(\cdot)$
S	\rightarrow NP VP	0	1
NP	\rightarrow NP PP	-1	$\frac{1}{2}$
	$\rightarrow we$	-2	$\frac{1}{4}$
	$\rightarrow sushi$	-3	$\frac{1}{8}$
	$\rightarrow chopsticks$	-3	$\frac{1}{8}$
PP	\rightarrow IN NP	0	1
IN	$\rightarrow with$	0	1
VP	\rightarrow V NP	-1	$\frac{1}{2}$
	\rightarrow VP PP	-2	$\frac{1}{4}$
	\rightarrow MD V	-2	$\frac{1}{4}$
V	$\rightarrow eat$	0	1

Illustration

we eat sushi with chopsticks

NP 1				
	V 1	VP 1/16		
		NP 1/8		
			IN 1	
				NP 1/8

		$\psi(\cdot)$	$\exp \psi(\cdot)$
S	\rightarrow NP VP	0	1
NP	\rightarrow NP PP	-1	$\frac{1}{2}$
	$\rightarrow we$	-2	$\frac{1}{4}$
	$\rightarrow sushi$	-3	$\frac{1}{8}$
	$\rightarrow chopsticks$	-3	$\frac{1}{8}$
PP	\rightarrow IN NP	0	1
IN	$\rightarrow with$	0	1
VP	\rightarrow V NP	-1	$\frac{1}{2}$
	\rightarrow VP PP	-2	$\frac{1}{4}$
	\rightarrow MD V	-2	$\frac{1}{4}$
V	$\rightarrow eat$	0	1

Illustration

we eat sushi with chopsticks

NP 1				
	V 1	VP 1/16		
		NP 1/8		
			IN 1	PP 1/8
				NP 1/8

		$\psi(\cdot)$	$\exp \psi(\cdot)$
S	\rightarrow NP VP	0	1
NP	\rightarrow NP PP	-1	$\frac{1}{2}$
	$\rightarrow we$	-2	$\frac{1}{4}$
	$\rightarrow sushi$	-3	$\frac{1}{8}$
	$\rightarrow chopsticks$	-3	$\frac{1}{8}$
PP	\rightarrow IN NP	0	1
IN	$\rightarrow with$	0	1
VP	\rightarrow V NP	-1	$\frac{1}{2}$
	\rightarrow VP PP	-2	$\frac{1}{4}$
	\rightarrow MD V	-2	$\frac{1}{4}$
V	$\rightarrow eat$	0	1

Illustration

we eat sushi with chopsticks

NP 1				
	V 1	VP 1/16		
		NP 1/8		NP 1/128
			IN 1	PP 1/8
				NP 1/8

		$\psi(\cdot)$	$\exp \psi(\cdot)$
S	\rightarrow NP VP	0	1
NP	\rightarrow NP PP	-1	$\frac{1}{2}$
	$\rightarrow we$	-2	$\frac{1}{4}$
	$\rightarrow sushi$	-3	$\frac{1}{8}$
	$\rightarrow chopsticks$	-3	$\frac{1}{8}$
PP	\rightarrow IN NP	0	1
IN	$\rightarrow with$	0	1
VP	\rightarrow V NP	-1	$\frac{1}{2}$
	\rightarrow VP PP	-2	$\frac{1}{4}$
	\rightarrow MD V	-2	$\frac{1}{4}$
V	$\rightarrow eat$	0	1

Illustration

we	eat	sushi	with	chopsticks
NP 1				
	V 1	VP 1/16		VP $\frac{1}{4} * \frac{1}{16} * \frac{1}{8} = \frac{1}{512}$
		NP 1/8		NP 1/128
			IN 1	PP 1/8
				NP 1/8

		$\psi(\cdot)$	$\exp \psi(\cdot)$
S	\rightarrow NP VP	0	1
NP	\rightarrow NP PP	-1	$\frac{1}{2}$
	$\rightarrow we$	-2	$\frac{1}{4}$
	$\rightarrow sushi$	-3	$\frac{1}{8}$
	$\rightarrow chopsticks$	-3	$\frac{1}{8}$
PP	\rightarrow IN NP	0	1
IN	$\rightarrow with$	0	1
VP	\rightarrow V NP	-1	$\frac{1}{2}$
	\rightarrow VP PP	-2	$\frac{1}{4}$
	\rightarrow MD V	-2	$\frac{1}{4}$
V	$\rightarrow eat$	0	1

Illustration

we eat sushi with chopsticks

NP 1				
	V 1	VP 1/16		VP $\frac{1}{2} * 1 * \frac{1}{128} =$ 1/256
		NP 1/8		NP 1/128
			IN 1	PP 1/8
				NP 1/8

1/256 > 1/512
→ this is a better analysis, so replace old value

		$\psi(\cdot)$	$\exp \psi(\cdot)$
S	→ NP VP	0	1
NP	→ NP PP	-1	$\frac{1}{2}$
	→ we	-2	$\frac{1}{4}$
	→ sushi	-3	$\frac{1}{8}$
	→ chopsticks	-3	$\frac{1}{8}$
PP	→ IN NP	0	1
IN	→ with	0	1
VP	→ V NP	-1	$\frac{1}{2}$
	→ VP PP	-2	$\frac{1}{4}$
	→ MD V	-2	$\frac{1}{4}$
V	→ eat	0	1

Illustration

we eat sushi with chopsticks

NP 1				
	V 1	VP 1/16		VP 1/256
		NP 1/8		NP 1/128
			IN 1	PP 1/8
				NP 1/8

		$\psi(\cdot)$	$\exp \psi(\cdot)$
S	\rightarrow NP VP	0	1
NP	\rightarrow NP PP	-1	$\frac{1}{2}$
	$\rightarrow we$	-2	$\frac{1}{4}$
	$\rightarrow sushi$	-3	$\frac{1}{8}$
	$\rightarrow chopsticks$	-3	$\frac{1}{8}$
PP	\rightarrow IN NP	0	1
IN	$\rightarrow with$	0	1
VP	\rightarrow V NP	-1	$\frac{1}{2}$
	\rightarrow VP PP	-2	$\frac{1}{4}$
	\rightarrow MD V	-2	$\frac{1}{4}$
V	$\rightarrow eat$	0	1

Illustration

we eat sushi with chopsticks

NP 1				S 1/1024
	V 1	VP 1/16		VP 1/256
		NP 1/8		NP 1/128
			IN 1	PP 1/8
				NP 1/8

		$\psi(\cdot)$	$\exp \psi(\cdot)$
S	\rightarrow NP VP	0	1
NP	\rightarrow NP PP	-1	$\frac{1}{2}$
	$\rightarrow we$	-2	$\frac{1}{4}$
	$\rightarrow sushi$	-3	$\frac{1}{8}$
	$\rightarrow chopsticks$	-3	$\frac{1}{8}$
PP	\rightarrow IN NP	0	1
IN	$\rightarrow with$	0	1
VP	\rightarrow V NP	-1	$\frac{1}{2}$
	\rightarrow VP PP	-2	$\frac{1}{4}$
	\rightarrow MD V	-2	$\frac{1}{4}$
V	$\rightarrow eat$	0	1

Prob CYK: Retrieving The parses

- S in the top-right corner of parse table indicates success
- Retain back-pointer to best analysis
 - * for each chart cell, store the split point and the non-terminal for the left and right children
- To get parse(s), follow pointers back for each match
- Convert back from CNF by removing new non-terminals

Complexity of CYK

- What's the space and time complexity of this algorithm?
 - * in terms of n the length of the input sentence

Problems with (P)CFGs

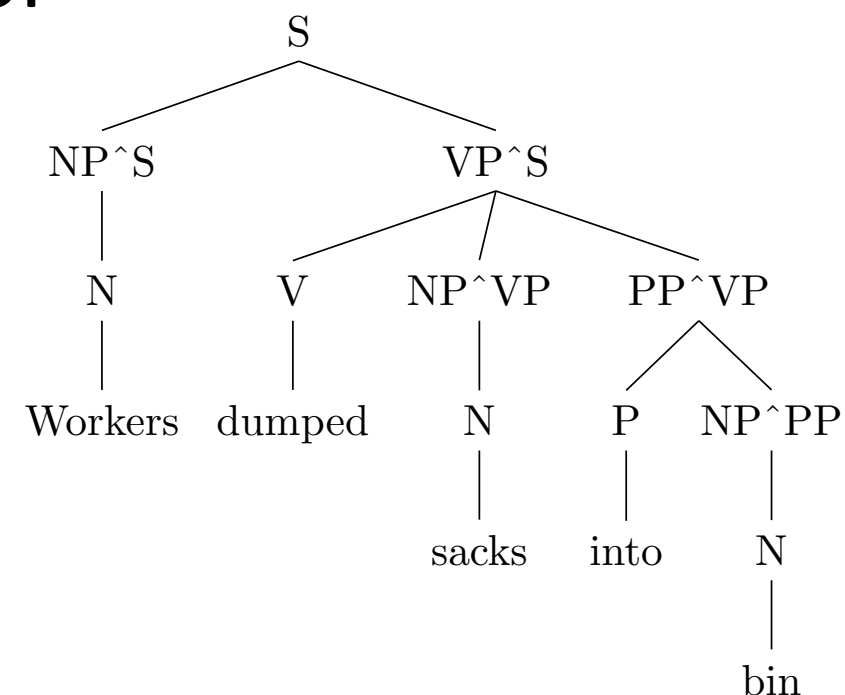
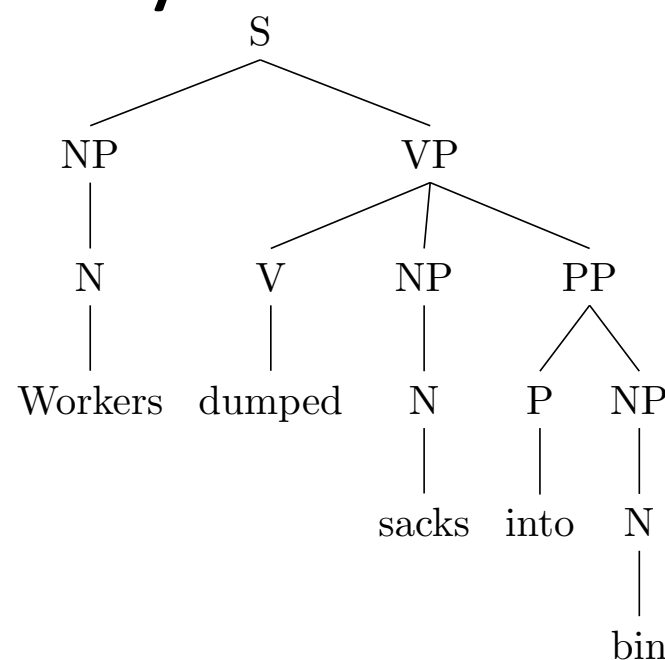
- **poor independence assumptions:** rewrite decisions made independently, whereas inter-dependence is often needed to capture global structure.
 - * E.g., NP → PRP used often as subject (first NP), much less often as object (second NP)
- **lack of lexical conditioning:** non-terminals representation behaviour of the actual words, but are much too coarse.
Problems with
 - * preposition attachment ambiguity;
 - * subcategorisation (*[forgot NP]* vs *[forgot S]*);
 - * coordinate structure ambiguities (*dogs in houses and cats*)

PP Attachment

- Consider sentences (PP shown bracketed)
 - (1) *Workers dumped sacks [into bin].*
 - (2) *Fishermen caught tons [of herring].*
- Both have same POS tag sequence, but different structure
 - * PP attaches either high (to the verb) or low (to the noun)
 - * how to make this attachment decision? Difference between the two analyses comes down to rules:
 - $VP \rightarrow \text{Verb NP PP}$ vs. $VP \rightarrow \text{Verb NP}; NP \rightarrow NP PP$
- The probabilities of these three rules drive attachment, *irrespective of the verb, preposition and noun*

One solution: parent conditioning

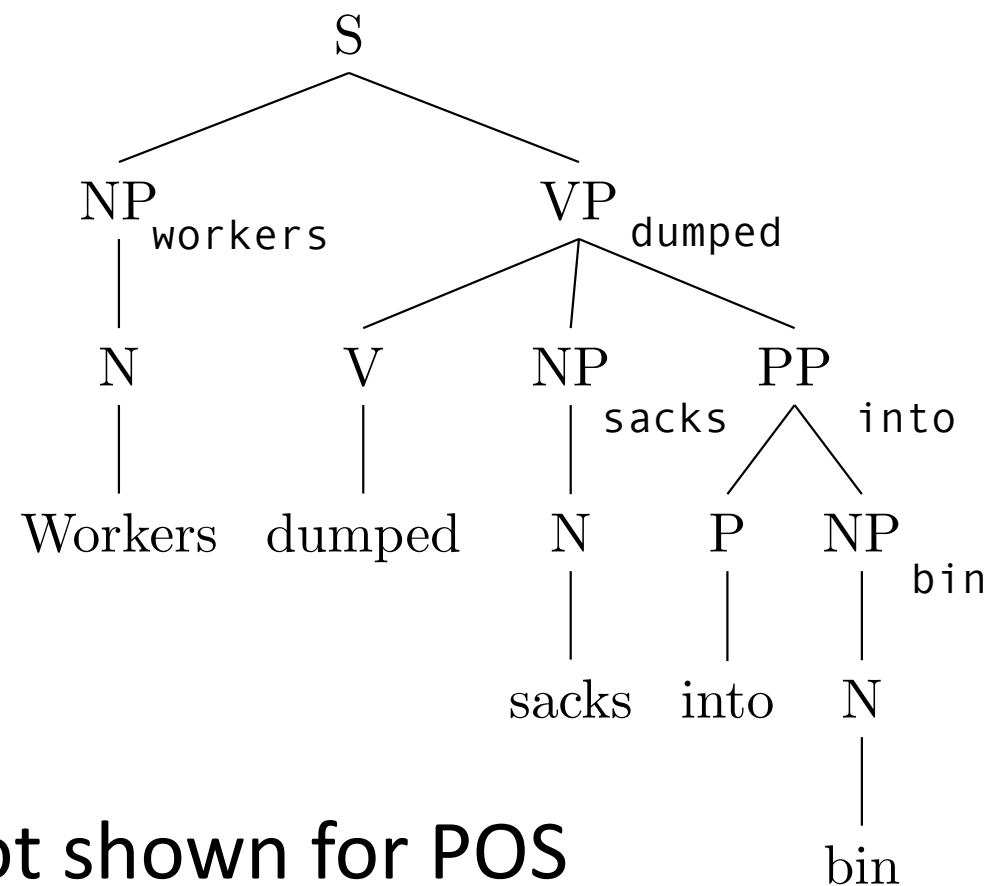
- Make non-terminals more explicit by incorporating parent symbol into each symbol



- NP^S represents subject position (left); NP^VP denotes object position (right); PP^VP is different to PP^NP
- Helps to make general tags more specific, used for a number of different purposes, e.g., *He said **that** I saw ...*

Another solution: Head Lexicalisation

- Record **head word** with parent symbols
 - the most salient child of a constituent, usually the noun in a NP, verb in a VP etc



- head words not shown for POS
- $VP \rightarrow V NP PP \Rightarrow$
 $VP(dumped) \rightarrow V(dumped) NP(sacks) PP(into)$

Head lexicalisation

- Incorporate head words into productions, such that the most important links between words is captured
 - * rule captures correlations between head tokens of phrases
- Grammar symbol inventory expands massively!
 - * Many of the productions much too specific, seen very rarely
 - * Learning more involved to avoid sparsity problems (e.g., zero probabilities)

A final word

- PCFGs widely used, and are some of the best performing parsers available. E.g.,
 - * Collins parser, Berkeley parser, Stanford parser
 - * all use some form of lexicalisation or change to non-terminal set with CFGs
- But not used universally, a competing method is to treat parsing as a sequential process of “transitions”
 - * next week, dependency parsing

Required Reading

- J&M3 Ch. 12 – 12.6
 - * Warning: several errors in the computations, and grammar used for PCYK is not in CNF