# Probabilistic Context Free Grammars

#### Overview

- Probabilistic Context-Free Grammars (PCFGs)
- ► The CKY Algorithm for parsing with PCFGs

#### A Probabilistic Context-Free Grammar (PCFG)

S	$\Rightarrow$	NP	VP	1.0
VP	$\Rightarrow$	Vi		0.4
VP	$\Rightarrow$	Vt	NP	0.4
VP	$\Rightarrow$	VP	PP	0.2
NP	$\Rightarrow$	DT	NN	0.3
NP	$\Rightarrow$	NP	PP	0.7
PP	$\Rightarrow$	Р	NP	1.0

Vi	$\Rightarrow$	sleeps	1.0
Vt	$\Rightarrow$	saw	1.0
NN	$\Rightarrow$	man	0.7
NN	$\Rightarrow$	woman	0.2
NN	$\Rightarrow$	telescope	0.1
DT	$\Rightarrow$	the	1.0
IN	$\Rightarrow$	with	0.5
IN	$\Rightarrow$	in	0.5

Probability of a tree t with rules

$$\alpha_1 \to \beta_1, \alpha_2 \to \beta_2, \dots, \alpha_n \to \beta_n$$

is  $p(t) = \prod_{i=1}^{n} q(\alpha_i \to \beta_i)$  where  $q(\alpha \to \beta)$  is the probability for rule  $\alpha \to \beta$ .

#### **DERIVATION**

S

NP VP

DT NN VP

the NN VP

the dog VP

the dog Vi

the dog laughs

#### **RULES USED**

 $S \rightarrow NP VP$ 

 $\mathsf{NP} \to \mathsf{DT} \; \mathsf{NN}$ 

 $\mathsf{DT} \to \mathsf{the}$ 

 $\mathsf{NN} \to \mathsf{dog}$ 

 $\mathsf{VP} \to \mathsf{Vi}$ 

 $\mathsf{Vi} \to \mathsf{laughs}$ 

#### **PROBABILITY**

1.0

0.3

1.0

0.1

0.4

0.5

#### Properties of PCFGs

 Assigns a probability to each *left-most derivation*, or parse-tree, allowed by the underlying CFG

#### 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.

#### Properties of PCFGs

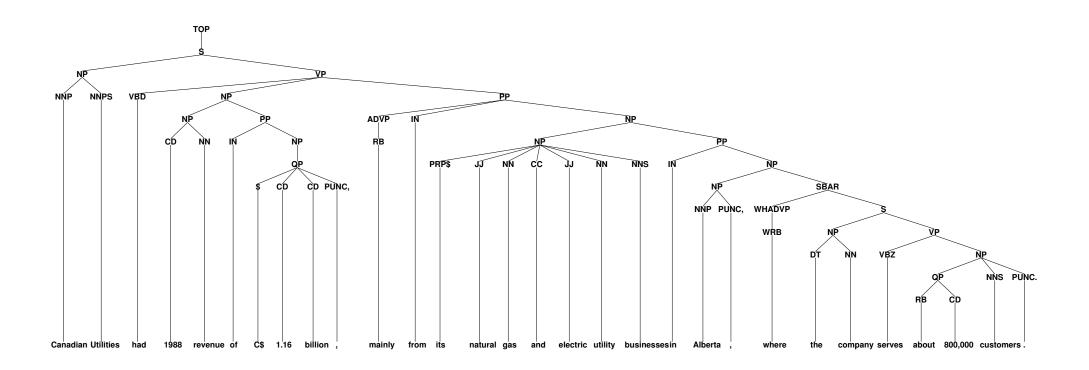
- Assigns a probability to each *left-most derivation*, or parse-tree, allowed by the underlying CFG
- Say we have a sentence s, set of derivations for that sentence is  $\mathcal{T}(s)$ . Then a PCFG assigns a probability p(t) to each member of  $\mathcal{T}(s)$ . i.e., we now have a ranking in order of probability.
- ightharpoonup The most likely parse tree for a sentence s is

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

#### Data for Parsing Experiments: Treebanks

- ▶ Penn WSJ Treebank = 50,000 sentences with associated trees
- ▶ Usual set-up: 40,000 training sentences, 2400 test sentences

#### An example tree:



#### Deriving 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:

$$q_{ML}(\alpha \to \beta) = \frac{\mathsf{Count}(\alpha \to \beta)}{\mathsf{Count}(\alpha)}$$

where the counts are taken from a training set of example trees.

▶ If the training data is generated by a PCFG, then as the training data size goes to infinity, the maximum-likelihood PCFG will converge to the same distribution as the "true" PCFG.

#### 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.
- ightharpoonup Given a PCFG and a sentence s, how do we find

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

#### Chomsky Normal Form

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

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

#### A Dynamic Programming Algorithm

ightharpoonup Given a PCFG and a sentence s, how do we find

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

Notation:

n= number of words in the sentence  $w_i=i$ 'th word in the sentence N= the set of non-terminals in the grammar S= the start symbol in the grammar

► Define a dynamic programming table

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

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

#### A Dynamic Programming 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  $i = 1 \dots n$ ,  $j = (i + 1) \dots n$ ,  $X \in N$ ,

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

#### The Full Dynamic Programming Algorithm

**Input:** a sentence  $s = x_1 \dots x_n$ , a PCFG  $G = (N, \Sigma, S, R, q)$ . **Initialization:** 

For all  $i \in \{1 \dots n\}$ , for all  $X \in N$ ,

$$\pi(i, i, X) = \begin{cases} q(X \to x_i) & \text{if } X \to x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

#### Algorithm:

- ▶ For  $l = 1 \dots (n-1)$ 
  - ▶ For  $i = 1 \dots (n l)$ 
    - Set j = i + l
    - ightharpoonup For all  $X \in N$ , calculate

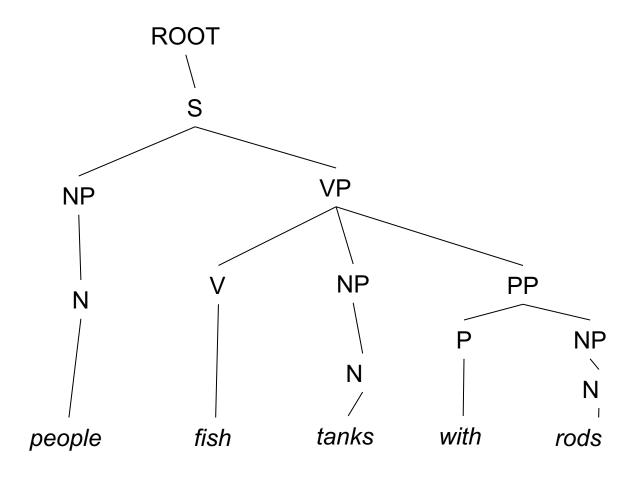
$$\pi(i,j,X) = \max_{\substack{X \to YZ \in R, \\ s \in \{i...(j-1)\}}} (q(X \to YZ) \times \pi(i,s,Y) \times \pi(s+1,j,Z))$$

and

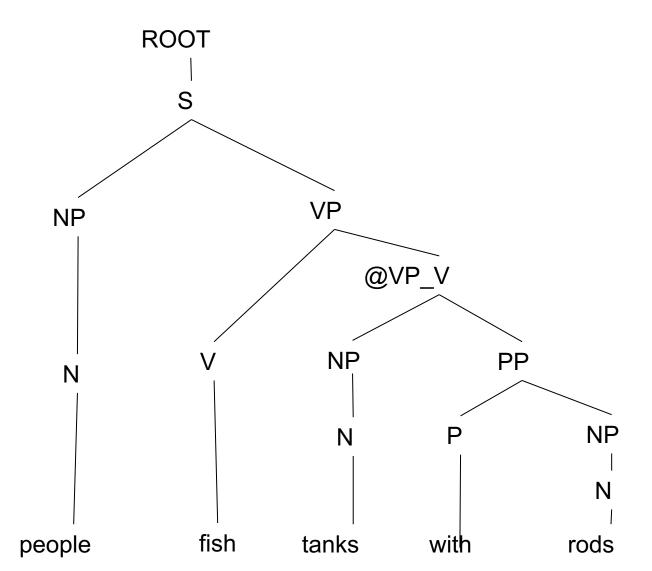
$$bp(i,j,X) = \arg\max_{\substack{X \to YZ \in R, \\ s \in \{i,\dots(i-1)\}}} (q(X \to YZ) \times \pi(i,s,Y) \times \pi(s+1,j,Z))$$

What's the run time Complexity?

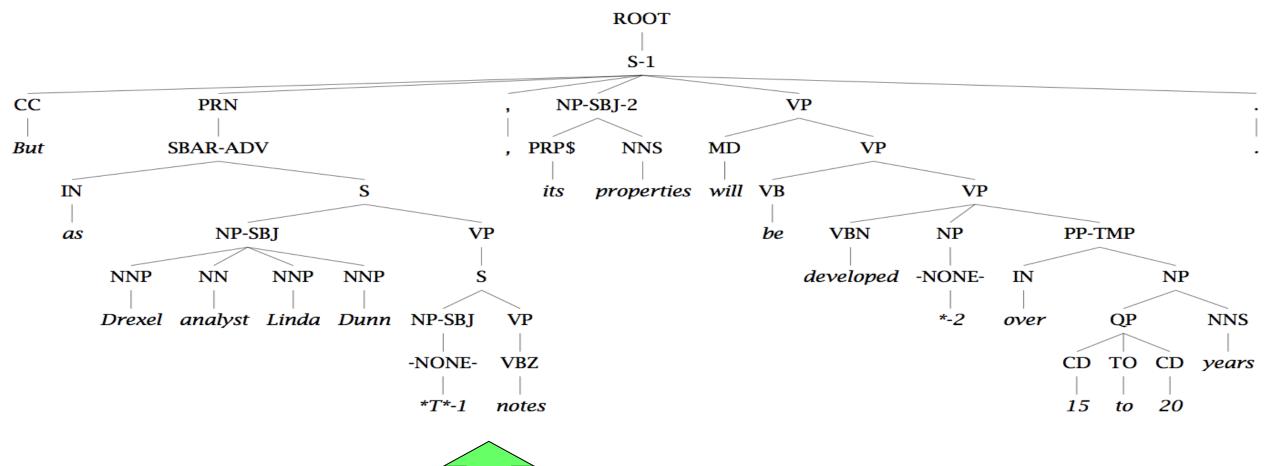
#### An example: before binarization...



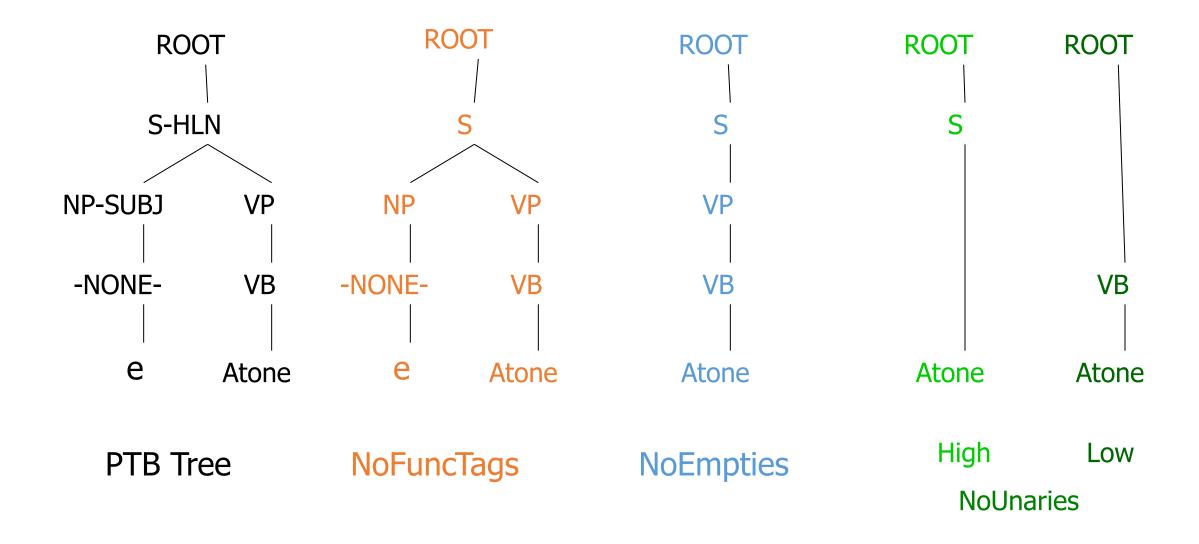
## After binarization...



## Unary rules: alchemy in the land of treebanks



## Treebank: empties and unaries



### Extended CKY parsing

- Unaries can be incorporated into the algorithm
  - Messy, but doesn't increase algorithmic complexity
- Empties can be incorporated
  - Doesn't increase complexity; essentially like unaries
- Binarization is vital
  - Without binarization, you don't get parsing cubic in the length of the sentence and in the number of nonterminals in the grammar

## The CKY algorithm (1960/1965) ... extended to unaries

```
function CKY(words, grammar) returns [most_probable_parse,prob]
  score = new double[#(words)+1][#(words)+1][#(nonterms)]
  back = new Pair[#(words)+1][#(words)+1][#nonterms]]
  for i=0; i<#(words); i++
    for A in nonterms
      if A -> words[i] in grammar
        score[i][i+1][A] = P(A \rightarrow words[i])
     else
        score[i][i+1][A] = 0
    //handle unaries
    boolean added = true
    while added
      added = false
      for A, B in nonterms
        if score[i][i+1][B] > 0 \&\& A->B in grammar
          prob = P(A->B)*score[i][i+1][B]
          if prob > score[i][i+1][A]
            score[i][i+1][A] = prob
            back[i][i+1][A] = B
            added = true
```

## The CKY algorithm (1960/1965) ... extended to unaries

```
for span = 2 to \#(words)
  for begin = 0 to \#(words) - span
    end = begin + span
    for split = begin+1 to end-1
      for A,B,C in nonterms
        prob=score[begin][split][B]*score[split][end][C]*P(A->BC)
        if prob > score[begin][end][A]
          score[begin]end][A] = prob
          back[begin][end][A] = new Triple(split,B,C)
    //handle unaries
    boolean added = true
   while added
      added = false
      for A, B in nonterms
        prob = P(A->B)*score[begin][end][B];
        if prob > score[begin][end][A]
          score[begin][end][A] = prob
          back[begin][end][A] = B
          added = true
return buildTree(score, back)
```

## **CKY Parsing**

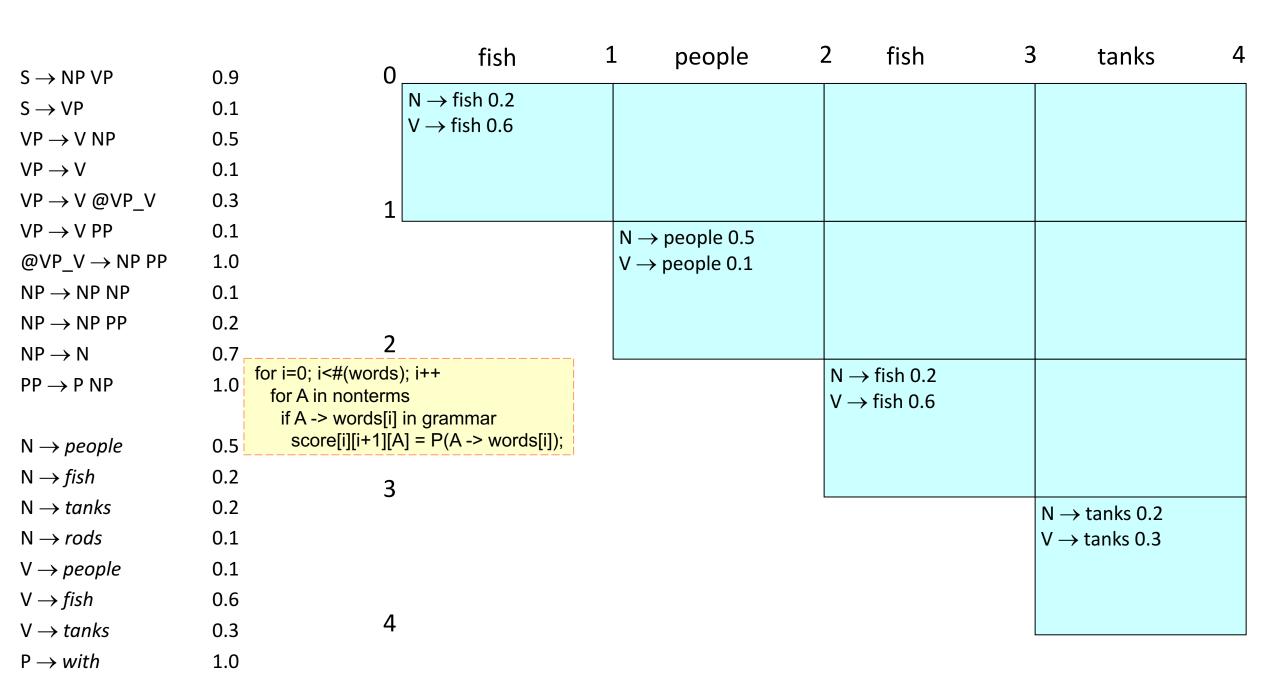
A worked example

## The grammar: Binary, Unaries, no epsilons,

$S \rightarrow NP VP$	0.9	$N \rightarrow people$	0.5
$S \rightarrow VP$	0.1	$N \rightarrow fish$	0.2
$VP \rightarrow V NP$	0.5	IN <del>-7</del> J1311	0.2
$VP \to V$	0.1	$N \rightarrow tanks$	0.2
$VP \rightarrow V @VP_V$	0.3	$N \rightarrow rods$	0.1
$VP \rightarrow VPP$	0.1		
$@VP_V \to NP\;PP$	1.0	$V \rightarrow people$	0.1
$NP \rightarrow NP NP$	0.1	$V \rightarrow fish$	0.6
$NP \rightarrow NP PP$	0.2	$V \rightarrow tanks$	U 3
$NP \rightarrow N$	0.7	v — turks	0.5
$PP \rightarrow P NP$	1.0	$P \rightarrow with$	1.0

0.	fish	1 people	2	fish	3	tanks	4
1	score[0][1]	score[0][2]		score[0][3]		score[0][4]	
2		score[1][2]		score[1][3]		score[1][4]	
3				score[2][3]		score[2][4]	
4						score[3][4]	

		•	fish	1	people	2	fish	3	tanks	4
$S \rightarrow NP VP$	0.9	0			· · ·					
$S \rightarrow VP$	0.1									
$VP \rightarrow V NP$	0.5									
$VP \rightarrow V$	0.1									
$VP \rightarrow V @VP_V$	0.3	1								
$VP \rightarrow VPP$	0.1	_								
$@VP_V \rightarrow NPPP$	1.0									
$NP \rightarrow NP NP$	0.1									
$NP \rightarrow NP PP$	0.2									
$NP \rightarrow N$	0.7	2								
$PP \rightarrow P NP$	1.0									
N  o people	0.5									
N → fish	0.2	3								
N  o tanks	0.2	3								
$N \rightarrow rods$	0.1									
$V \rightarrow people$	0.1									
$V \rightarrow fish$	0.6									
$V \rightarrow tanks$	0.3	4								
$P \rightarrow with$	1.0									



C NDVD	0.0	fish	1 people	2 fish	3 tanks 4
$S \rightarrow NP VP$	0.9	$0 \longrightarrow \text{fish } 0.2$			
$S \rightarrow VP$	0.1	$V \rightarrow \text{fish } 0.2$			
$VP \rightarrow V NP$	0.5	$NP \rightarrow N \ 0.14$			
$VP \rightarrow V$	0.1	$VP \rightarrow V 0.06$			
$VP \rightarrow V @VP_V$	0.3	$1 S \rightarrow VP 0.006$			
$VP \rightarrow VPP$	0.1		$N \rightarrow \text{people 0.5}$		
$@VP_V \rightarrow NPPP$	1.0		$V \rightarrow \text{people 0.1}$		
$NP \rightarrow NP NP$	0.1		$NP \rightarrow N 0.35$		
$NP \rightarrow NP PP$	0.2	_	$VP \rightarrow V 0.01$		
$NP \rightarrow N$	0.7	2	$S \rightarrow VP \ 0.001$		
$PP \rightarrow P NP$	1.0			$N \rightarrow \text{fish } 0.2$	
				$V \rightarrow \text{fish } 0.6$ NP $\rightarrow$ N 0.14	
$N \rightarrow people$	0.5			$VP \rightarrow V 0.06$	
$N \rightarrow fish$	0.2	// handle unaries		$S \rightarrow VP 0.006$	
N  o tanks	0.2	boolean added = true while added			N → tanks 0.2
$N \rightarrow rods$	0.1	added = false			$V \rightarrow tanks 0.3$
$V \rightarrow people$	0.1	for A, B in nonterms if score[i][i+1][B] > 0 && A->E	3 in grammar		$NP \rightarrow N \ 0.14$
$V \rightarrow f$ ish	0.6	prob = P(A->B)*score[i][i+1 if(prob > score[i][i+1][A])	][B]		$VP \rightarrow V 0.03$
$V \rightarrow tanks$	0.3	score[i][i+1][A] = prob			$S \rightarrow VP 0.003$
$P \rightarrow with$	1.0	back[i][i+1][A] = B added = true			

		fish	1 people	2 fish	3 tanks 4
$S \rightarrow NP VP$	0.9	0	$NP \rightarrow NP NP$		
$S \rightarrow VP$	0.1	$N \rightarrow \text{fish } 0.2$ V $\rightarrow \text{fish } 0.6$	0.0049		
$VP \rightarrow V NP$	0.5	$V \rightarrow 11SH 0.6$ NP $\rightarrow$ N 0.14	$VP \rightarrow V NP$		
$VP \rightarrow V$	0.1	$VP \rightarrow V 0.06$	0.105		
$VP \rightarrow V @VP_V$	0.3	1 > VP 0.006	$S \rightarrow NP VP$ 0.00126		
$VP \rightarrow VPP$	0.1		$N \rightarrow \text{people 0.5}$	$NP \rightarrow NP NP$	
$@VP_V \rightarrow NPPP$	1.0		$V \rightarrow \text{people 0.1}$	0.0049	
$NP \rightarrow NP NP$	0.1		$NP \rightarrow N \ 0.35$	$VP \rightarrow V NP$ 0.007	
$NP \rightarrow NP PP$	0.2		$VP \rightarrow V 0.01$	$S \rightarrow NP VP$	
$NP \rightarrow N$	0.7	2	$S \rightarrow VP 0.001$	0.0189	
$PP \rightarrow P NP$	1.0			$N \rightarrow \text{fish } 0.2$	$ \begin{array}{c} NP \to NP \ NP \\ 0.00196 \end{array} $
N  o people N  o fish	0.5 0.2	3		$V \rightarrow fish 0.6$ $NP \rightarrow N 0.14$ $VP \rightarrow V 0.06$ $S \rightarrow VP 0.006$	$VP \rightarrow V NP$ $0.042$ $S \rightarrow NP VP$ $0.00378$
N  o tanks	0.2				N → tanks 0.2
$N \rightarrow rods$	0.1	prob=score[begin][split][B]*score[sit] if (prob > score[begin][end][A])	split][end][C]*P(A->BC)		V → tanks 0.3
$V \rightarrow people$	0.1	score[begin]end][A] = prob			$NP \rightarrow N \ 0.14$
$V \rightarrow fish$	0.6	back[begin][end][A] = new Tripl	e(split,B,C)		$VP \rightarrow V 0.03$
V → tanks	0.3	4			$S \rightarrow VP 0.003$
$ extsf{P}  o  extsf{with}$	1.0				

		0	fish	1 ped	ople	2	fish	3	tanks	4
$S \rightarrow NP VP$	0.9	0	 → fish 0.2	$NP \rightarrow NP NI$	n					
$S \rightarrow VP$	0.1		$\rightarrow$ fish 0.6	0.00						
$VP \rightarrow V NP$	0.5		$P \rightarrow N 0.14$	$VP \rightarrow V NP$						
$VP \rightarrow V$	0.1		$P \rightarrow V 0.06$	0.10	)5					
$VP \rightarrow V @VP_V$	0.3	1 S -	→ VP 0.006	$S \rightarrow VP$ 0.01	.05					
$VP \rightarrow VPP$	0.1			$N \rightarrow peop$	le 0.5	$NP \rightarrow$	NP NP			
$@VP_V \rightarrow NPPP$	1.0			$V \rightarrow peopl$	le 0.1		0.0049			
$NP \rightarrow NP NP$	0.1			$NP \rightarrow N 0.$		$ VP \rightarrow$	V NP 0.007			
$NP  o NP \; PP$	0.2	2		$VP \rightarrow V 0.01$ S $\rightarrow VP 0.001$	$S \rightarrow N$					
$NP \rightarrow N$	0.7	2		$3 \rightarrow VP U.U$	VF 0.001		0.0189			
$PP \rightarrow P NP$	1.0						fish 0.2 fish 0.6		$NP \rightarrow NP NP$ $0.00196$	
N  o people N  o fish	0.5 0.2	<b>a</b> b	/handle unaries ooolean added = true			NP —	N 0.14 N 0.06 VP 0.006		$/P \rightarrow V NP$ $0.042$ $6 \rightarrow VP$ $0.0042$	
N  o tanks	0.2	_	vhile added added = false					N	$N \rightarrow tanks 0.2$	
$N \rightarrow rods$	0.1		for A, B in nonterms	:					$/ \rightarrow \text{tanks 0.1}$	
$V \rightarrow people$	0.1		prob = P(A->B)*score[being   prob > score[begin][en					L	$NP \rightarrow N \ 0.14$	
$V \rightarrow f$ ish	0.6		score[begin][end][A] =	•					$/P \rightarrow V 0.03$	
V → tanks	0.3	4	back[begin][end][A] = E added = true	)				S	$S \rightarrow VP \ 0.003$	
$P \rightarrow with$	1.0									

		fish	1 people	2 fish	3 tanks 4
$S \rightarrow NP VP$ $S \rightarrow VP$ $VP \rightarrow V NP$ $VP \rightarrow V$ $VP \rightarrow V @VP_V$	0.9 0.1 0.5 0.1 0.3	0 $N \rightarrow \text{fish } 0.2$ $V \rightarrow \text{fish } 0.6$ $NP \rightarrow N \ 0.14$ $VP \rightarrow V \ 0.06$ $S \rightarrow VP \ 0.006$	$NP \rightarrow NP NP$ $0.0049$ $VP \rightarrow V NP$ $0.105$ $S \rightarrow VP$ $0.0105$	$NP \rightarrow NP NP$ $0.0000686$ $VP \rightarrow V NP$ $0.00147$ $S \rightarrow NP VP$ $0.000882$	
$VP \rightarrow V PP$ $@VP_V \rightarrow NP PP$ $NP \rightarrow NP NP$ $NP \rightarrow NP PP$ $NP \rightarrow N$	0.1 1.0 0.1 0.2 0.7	2	$N \rightarrow \text{people } 0.5$ $V \rightarrow \text{people } 0.1$ $NP \rightarrow N \ 0.35$ $VP \rightarrow V \ 0.01$ $S \rightarrow VP \ 0.001$	$NP \rightarrow NP NP$ $0.0049$ $VP \rightarrow V NP$ $0.007$ $S \rightarrow NP VP$ $0.0189$	
$PP \rightarrow P NP$ $N \rightarrow people$ $N \rightarrow fish$	1.0 0.5 0.2	3		$N \rightarrow \text{fish } 0.2$ $V \rightarrow \text{fish } 0.6$ $NP \rightarrow N \ 0.14$ $VP \rightarrow V \ 0.06$ $S \rightarrow VP \ 0.006$	$NP \rightarrow NP NP$ $0.00196$ $VP \rightarrow V NP$ $0.042$ $S \rightarrow VP$ $0.0042$
$N \rightarrow tanks$ $N \rightarrow rods$ $V \rightarrow people$ $V \rightarrow fish$ $V \rightarrow tanks$ $P \rightarrow with$	0.2 0.1 0.1 0.6 0.3 1.0	if prob > sore[be		P(A->BC)	$N \rightarrow tanks 0.2$ $V \rightarrow tanks 0.1$ $NP \rightarrow N 0.14$ $VP \rightarrow V 0.03$ $S \rightarrow VP 0.003$

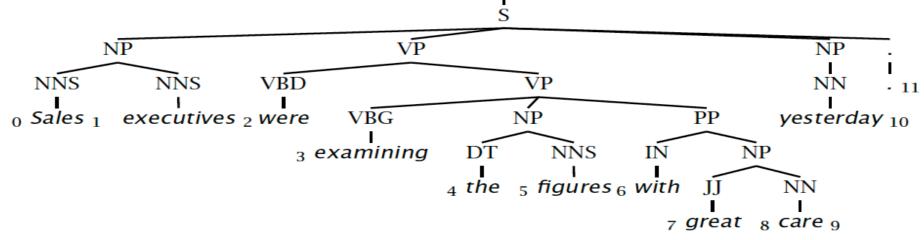
		fish	1 people	2 fish	3 tanks 4
$S \rightarrow NP VP$ $S \rightarrow VP$ $VP \rightarrow V NP$ $VP \rightarrow V$ $VP \rightarrow V @VP_V$ $VP \rightarrow V PP$	0.9 0.1 0.5 0.1 0.3 0.1	0 $N \rightarrow \text{fish } 0.2$ $V \rightarrow \text{fish } 0.6$ $NP \rightarrow N \ 0.14$ $VP \rightarrow V \ 0.06$ $S \rightarrow VP \ 0.006$	$NP \rightarrow NP NP$ $0.0049$ $VP \rightarrow V NP$ $0.105$ $S \rightarrow VP$ $0.0105$ $N \rightarrow people 0.5$	$NP \rightarrow NP NP$ $0.0000686$ $VP \rightarrow V NP$ $0.00147$ $S \rightarrow NP VP$ $0.000882$ $NP \rightarrow NP NP$	NP → NP NP
$@VP_V \rightarrow NPPP$ $NP \rightarrow NPNP$ $NP \rightarrow NPPP$ $NP \rightarrow N$ $PP \rightarrow PNP$	1.0 0.1 0.2 0.7 1.0	2	$V \rightarrow \text{people 0.1}$ $NP \rightarrow N 0.35$ $VP \rightarrow V 0.01$ $S \rightarrow VP 0.001$	$0.0049$ $VP \rightarrow V NP$ $0.007$ $S \rightarrow NP VP$ $0.0189$ $N \rightarrow fish  0.2$	$0.0000686$ $VP \rightarrow V NP$ $0.000098$ $S \rightarrow NP VP$ $0.01323$ $NP \rightarrow NP NP$
$N \rightarrow people$ $N \rightarrow fish$	0.5 0.2	3		$V \rightarrow fish 0.6$ $NP \rightarrow N 0.14$ $VP \rightarrow V 0.06$ $S \rightarrow VP 0.006$	$0.00196$ $VP \rightarrow V NP$ $0.042$ $S \rightarrow VP$ $0.0042$
$N \rightarrow tanks$ $N \rightarrow rods$ $V \rightarrow people$ $V \rightarrow fish$ $V \rightarrow tanks$ $P \rightarrow with$	0.2 0.1 0.1 0.6 0.3 1.0	if prob > score[be score[begin]end	ms i][split][B]*score[split][end][C]*P( gin][end][A]	A->BC)	N $\rightarrow$ tanks 0.2 V $\rightarrow$ tanks 0.1 NP $\rightarrow$ N 0.14 VP $\rightarrow$ V 0.03 S $\rightarrow$ VP 0.003

C \ ND VD	0.0	0	fish	1	people	2	fish	3	tanks	4
$S \rightarrow NP VP$	0.9		→ fish 0.2	NP -	→ NP NP	NP →	NP NP	NP	$\rightarrow$ NP NP	
$S \rightarrow VP$	0.1		$\rightarrow$ fish 0.6		0.0049		0.0000686		0.0000009604	
$VP \rightarrow V NP$	0.5		$P \rightarrow N \ 0.14$	VP -	$\rightarrow$ V NP	$VP \rightarrow$		VP	$\rightarrow$ V NP	
$VP \rightarrow V$	0.1		→ V 0.06	, c	0.105	C . A	0.00147		0.00002058	
$VP \rightarrow V @VP_V$	0.3	1   S -	→ VP 0.006	$ S \rightarrow$	0.0105	$S \rightarrow N$	0.000882	5 -	→ NP VP 0.00018522	
$VP \rightarrow VPP$	0.1			N -	→ people 0.5	$NP \rightarrow$	NP NP	NP	$\rightarrow$ NP NP	
$@VP_V \rightarrow NPPP$	1.0				→ people 0.1		0.0049		0.0000686	
NP  o NP  NP	0.1			NP	$\rightarrow$ N 0.35	$VP \rightarrow$	V NP 0.007	VP	$\rightarrow$ V NP 0.000098	
$NP \rightarrow NP PP$	0.2				→ V 0.01	$S \rightarrow N$		s –	→ NP VP	
$NP \rightarrow N$	0.7	2		S —	→ VP 0.001		0.0189		0.01323	
$PP \rightarrow P NP$	1.0						fish 0.2	NP	$\rightarrow$ NP NP 0.00196	
N  o people	0.5					NP —	fish 0.6 → N 0.14 → V 0.06	VP	$\rightarrow V NP$ $0.042$	
$N \rightarrow fish$	0.2						VP 0.006	S —	→ VP	
$N \rightarrow tanks$	0.2	3	or split = begin+1 to	 end-1			V1 0.000	N	0.0042 → tanks 0.2	
$N \rightarrow rods$	0.1		for A,B,C in nonter	rms	core[split][end][C]*P	P(A->BC)			$\rightarrow$ tanks 0.2 $\rightarrow$ tanks 0.1	
$V \rightarrow people$	0.1		if prob > score[b	egin][end][A	]	( /			$\rightarrow$ N 0.14	
$V \rightarrow f$ ish	0.6		score[begin]er back[begin][en		Triple(split,B,C)				$\rightarrow$ V 0.03	
V → tanks	0.3	4	23.2[209][01					S -	→ VP 0.003	
$P \rightarrow with$	1.0		Call buildTree(s	core, back) to	get the best parse					

# Constituency Parser Evaluation

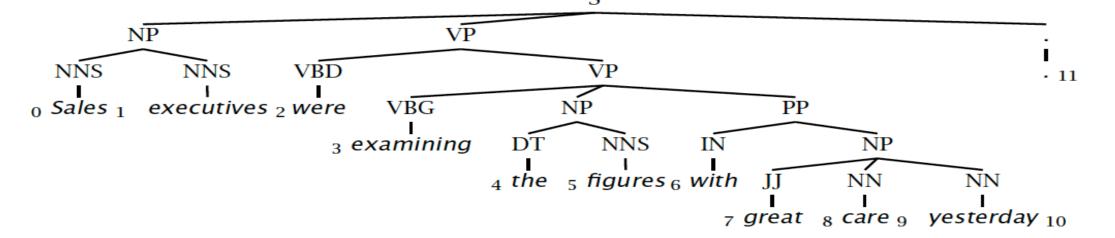
### Evaluating constituency parsing

Gold standard brackets: **S-(0:11)**, **NP-(0:2)**, VP-(2:9), VP-(3:9), **NP-(4:6)**, PP-(6-9), NP-(7,9), NP-(9:10)



Candidate brackets:

**S-(0:11)**, **NP-(0:2)**, VP-(2:10), VP-(3:10), **NP-(4:6)**, PP-(6-10), NP-(7,10)



### Evaluating constituency parsing

#### **Gold standard brackets:**

**S-(0:11), NP-(0:2)**, VP-(2:9), VP-(3:9), **NP-(4:6)**, PP-(6-9), NP-(7,9), NP-(9:10)

#### **Candidate brackets:**

**S-(0:11)**, **NP-(0:2)**, VP-(2:10), VP-(3:10), **NP-(4:6)**, PP-(6-10), NP-(7,10)

Labeled Precision 3/7 = 42.9%

Labeled Recall 3/8 = 37.5%

LP/LR F1 40.0%

Tagging Accuracy 11/11 = 100.0%

#### Summary

- ► PCFGs augments CFGs by including a probability for each rule in the grammar.
- ► The probability for a parse tree is the product of probabilities for the rules in the tree
- ► To build a PCFG-parsed parser:
  - 1. Learn a PCFG from a treebank
  - 2. Given a test data sentence, use the CKY algorithm to compute the highest probability tree for the sentence under the PCFG

## How good are PCFGs?

- Penn WSJ parsing accuracy: about 73% LP/LR F1
- Robust but not so accurate
  - Usually admit everything, but with low probability
  - A PCFG gives some idea of the plausibility of a parse
  - But not so good because the independence assumptions are too strong
- Give a probabilistic language model
  - But in the simple case it performs worse than a trigram model
- The problem seems to be that PCFGs lack the lexicalization of a trigram model