CS371N Lecture 16
Trees, PCFGs, CKY

Announcements

- A4 due Tuesday
- Midterm next Thursday in class
- My OHs are virtual today

Recap HMMs, Viterbi

HMMs $P(\bar{y}, \bar{x})$

We want $\underset{\bar{y}}{\text{argmax}}\ P(\bar{y} | \bar{x})$

Viterbi algorithm:

$v_i(\tilde{y})$ = score of best sequence ending in tag $\tilde{y}$ at index i

$$v_1(\tilde{y}) = \log P(x_1 | \tilde{y}) + \log P(\tilde{y})$$

$$v_i(\tilde{y}) = \log P(x_i|\tilde{y}) + \max_{\tilde{y}_{prev}} \left[\log P(\tilde{y}|\tilde{y}_{prev}) + v_{i-1}(\tilde{y}_{prev}) \right]$$

n words $|\mathcal{T}|$ tags, chart is $n|\mathcal{T}|$

$O(n|\mathcal{T}|^2)$

Today

- Constituency syntax
- PCFGs
- CKY

Context-free Grammars

$\{ N \quad T \quad S \quad R \}$

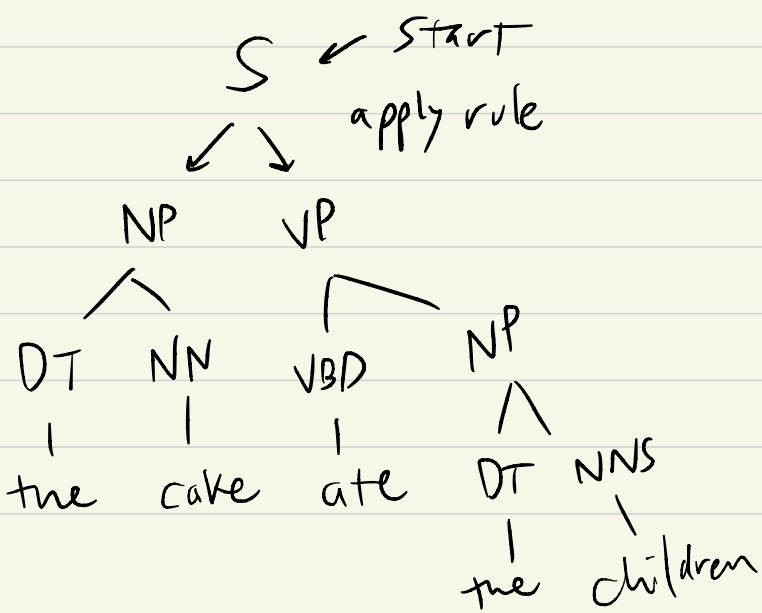
N: nonterminals — T: terminals — S: start symbol — R: rules

S, VP, NP... words S

Rules

binary			unary	
$S \rightarrow NP\ VP$	1		$DT \rightarrow$ the	1
$VP \rightarrow VBD\ NP$	1		$NNS \rightarrow$ children	1
$NP \rightarrow DT\ NN$	1/2		$NN \rightarrow$ cake	1/2
$NP \rightarrow DT\ NNS$	1/2		$NN \rightarrow$ spoon	1/2
			$VBD \rightarrow$ ate	1

CFG defines a set of trees



Probabilistic CFGs

Each rule has a prob.

Probs. normalize per parent

$P(\text{rule} \mid \text{parent})$

Ex. $P(\text{rule} \mid NP) = \begin{cases} 1/2 & NP \to DT\ NN \\ 1/2 & NP \to DT\ NNS \end{cases}$

$$P(\text{tree}) = \prod_{\text{rules}} P(\text{rule} \mid \text{parent}(\text{rule}))$$

Generative model of sentences

$P(T, \bar{x})$ HMM: $P(\bar{y}, \bar{x})$

What we can do: (1) generate sents

(2) Compute $\underset{T}{\text{argmax}}\ P(T \mid \bar{x})$

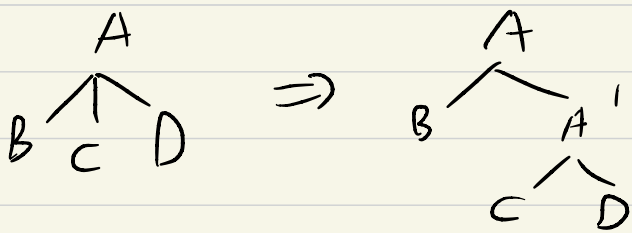
Steps to parsing

Input: "treebank": sents labeled w/trees

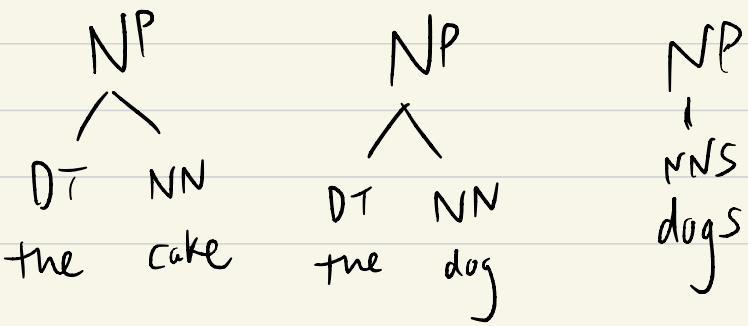
Output: grammar (PCFG)

① Grammar preprocessing

Binarize trees



② Read grammar off treebank and compute probs. Count + normalize



$$P(\text{rule} \mid \text{NP}) = \begin{cases} \text{NP} \to \text{DT NN} & 2/3 \\ \text{NP} \to \text{NNS} & 1/3 \end{cases}$$

$$P(\text{word} \mid \text{NN}) = \begin{cases} \text{cake} & 1/2 \\ \text{dog} & 1/2 \end{cases}$$

$$P(\text{word} \mid \text{NNS}) = \{ \text{dogs} \quad 1$$

(3) Parsing algorithm CKY

Inputs: PCFG, sentence $\bar{x}$

Output: $\underset{T}{\arg\max}\ P(T \mid \bar{x})$

most likely tree T for $\bar{x}$

CKY

Dynamic program: track the best score for building a nonterminal over each span of a sentence

≈ V: (4)

$T(i, j, X)$ = score (log prob) of the best way to build constituent X over span (i, j)

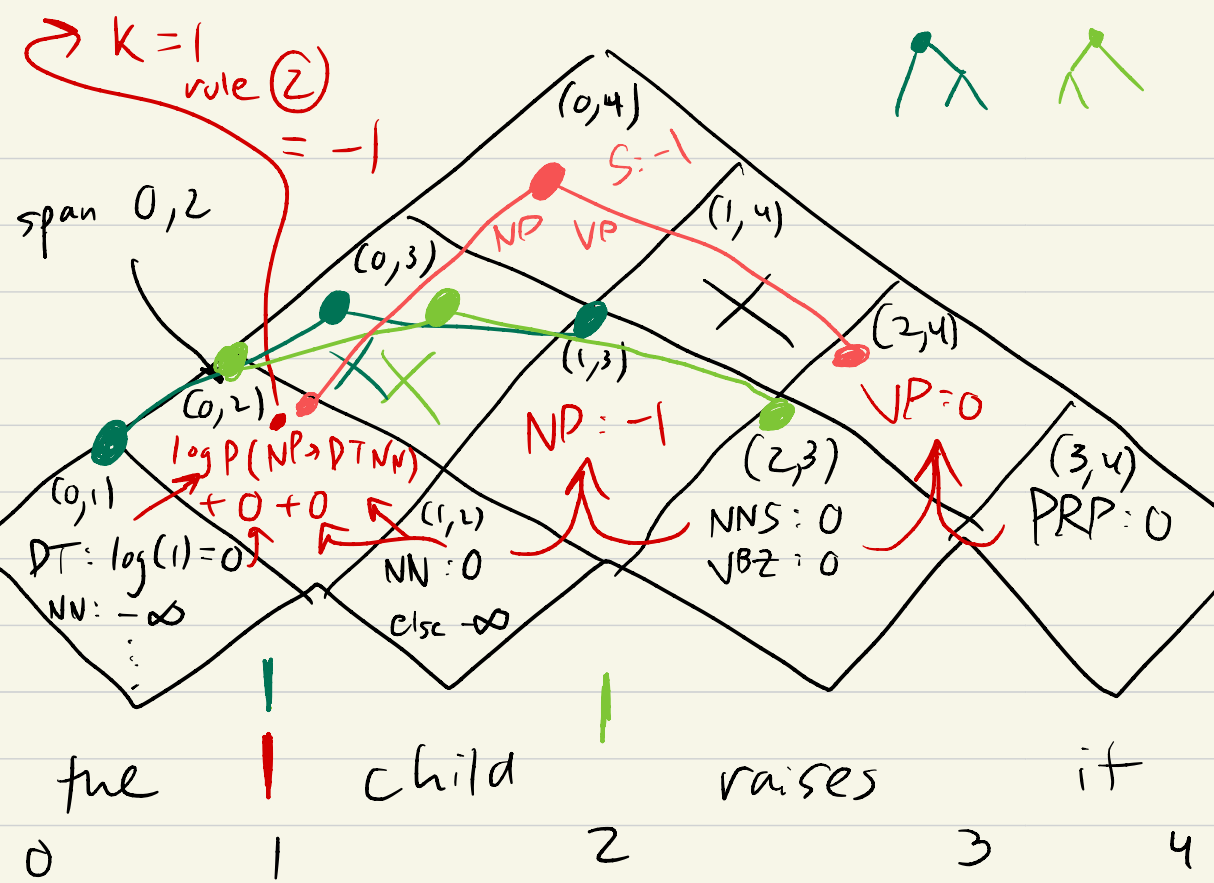
Base case: $T(i, i+1, X) = \log P(w_i | X)$

Recursive case: $T(i, j, X)$

loop over all is, js, Xs

loop over all "split points" k "transition"

$$= \max_{k: i<k<j} \max_{X \to X_1 X_2} \left[\log P(X \to X_1 X_2) + T(i, k, X_1) + T(k, j, X_2) \right]$$



Grammar

	S → NP VP	1
②	NP → DT NN	1/2
	NP → NN NNS	1/2
	VP → VBZ PRP	1
	DT → the	1
	NN → child	1
	NNS → raises	1
	VBZ → raises	1
	PRP → it	1

Assume $\log(1/2) = -1$