CS 378 Lecture 12
Constituency Parsing

Announcements

- Midterm: Weds - Sat of next week
- A3 due Tues
- A2 back soon

Recap

Viterbi: dynamic programming alg to compute $\max_{\bar{y}} P(\bar{y}, \bar{x})$ for an HMM

(similar alg today for parsing: CKY)

Beam search: approximates Viterbi by only preserving k states at a timestep

Today Constituency intro
Probabilistic context-free grammars
CKY algorithm

Constituency see slides

Context-free grammars

{ N T S R }

N: nonterminals

S, NP, VP, PP

DT, NN, VPD, IN, NNS → preterminals

T: terminals ~words

the
children
ate
cake
with
spoon

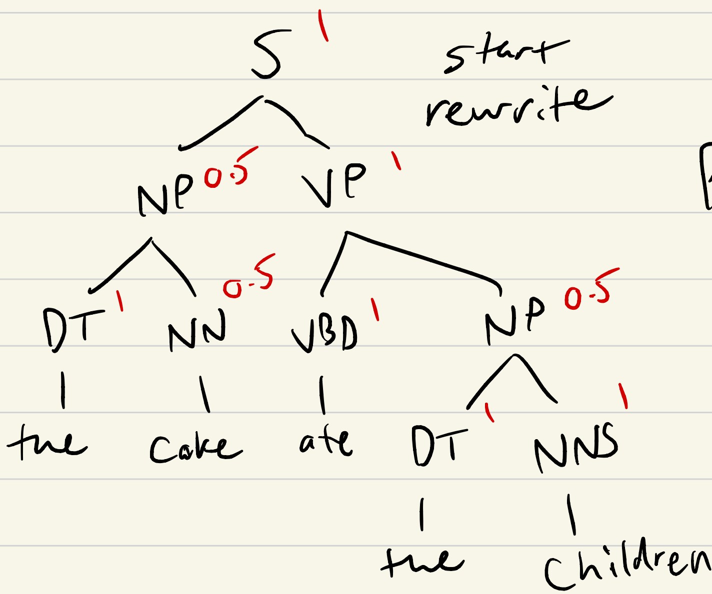
S → start symbol

S

R → rules

binary	unary
S → NP VP 1.0	DT → the 1.0
VP → VBD NP 1.0	NNS → children 1.0
NP → DT NN 0.5	NN → cake 0.5
NP → DT NNS 0.5	NN → spoon 0.5
	VBP → ate 1.0

CFG defines a set of trees



$P(\text{tree}) = \frac{1}{8}$

Probabilistic CFGs

Each rule has a prob.
Probs normalize per parent

$$P(\text{rule} \mid NP) = \begin{cases} NP \to DT\ NN & 0.5 \\ NP \to DT\ NNS & 0.5 \end{cases}$$

(~ HMM transitions)

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

Building a parser

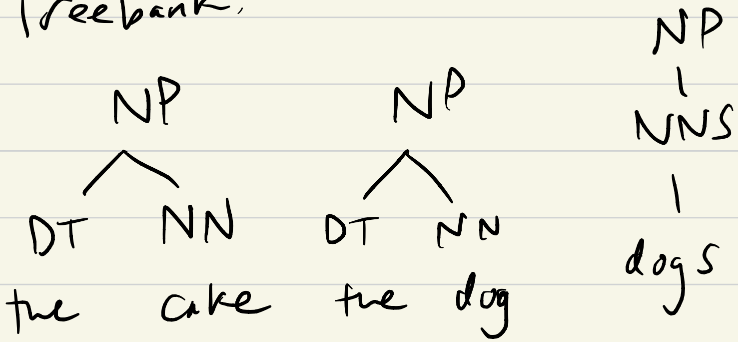
Input: treebank (sents w/ labeled trees)
Output: model + way to compute $\arg\max_T P(T \mid \bar{x})$

1. Grammar preprocessing (binarization) next time
2. Computing grammar + getting probs
3. Parsing

① Make every tree binary

② Read off grammar + count rules

Treebank:



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

$$P(w \mid NNS) = \{ 1.0 \ \text{dogs}$$

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

This is the maximum likelihood estimate of params for this data

③ CKY algorithm

Input: PCFG

sentence $\bar{x}$

Output: $\underset{T}{\text{argmax}}\ P(T \mid \bar{x})$

most likely tree for $\bar{x}$

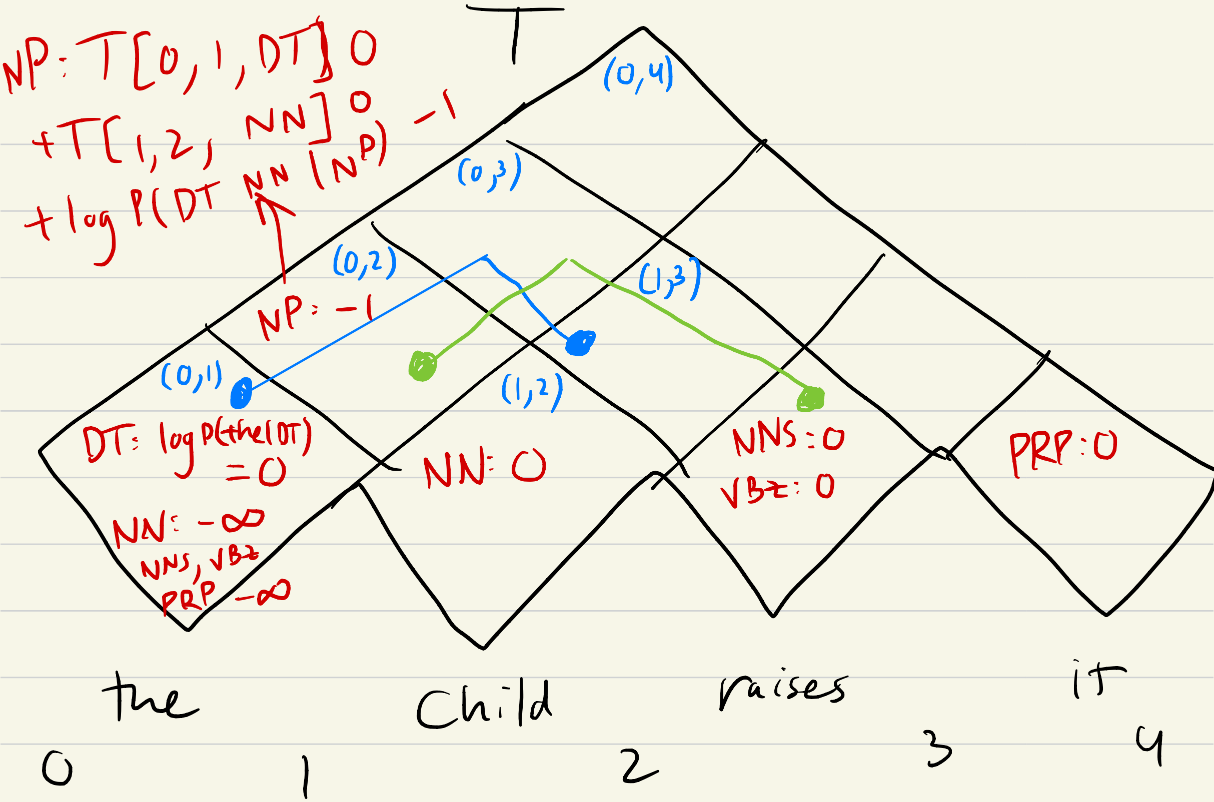
$$= \underset{T}{\text{argmax}}\ P(T, \bar{x})$$

dynamic program: track the best score for each nonterminal over each span of the sentence

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

sent len × sent len × number of symbols

python list indices



Grammar:

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

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

CKY

base: $T(i, i+1, X) = \log P(w_i \mid X)$

recursive:

$$T(i,j,X) = \max_{\underset{\text{split point}}{k:\, i<k<j}} \; \max_{\underset{\text{rule}}{X \to X_1 X_2}} \Big[\log P(X \to X_1 X_2) + T[i,k,X_1] + T[k,j,X_2] \Big]$$

cell $(0,3)$

$(0,1) + (1,3)$

$(0,2) + (2,3)$