CS 378 Lecture 12: Constituency Parsing

Announcements

- A3 due in one week
- Monday: talk by Preslav Nakov
- CRFs: see video on course website

Recap Viterbi: dynamic programming algo to compute $\max_{\bar{y}} P(\bar{y}, \bar{x})$

(similar algo today: CKY)

Beam search: faster approximation to Viterbi by ruling out low-scoring things

Today Constituency intro
Probabilistic context-free grammars
CKY algorithm (start)

Constituency see slides

Context-free grammars

{N T S R

nonterminals terminals start symbol rules

S, NP, VP, PP... words S

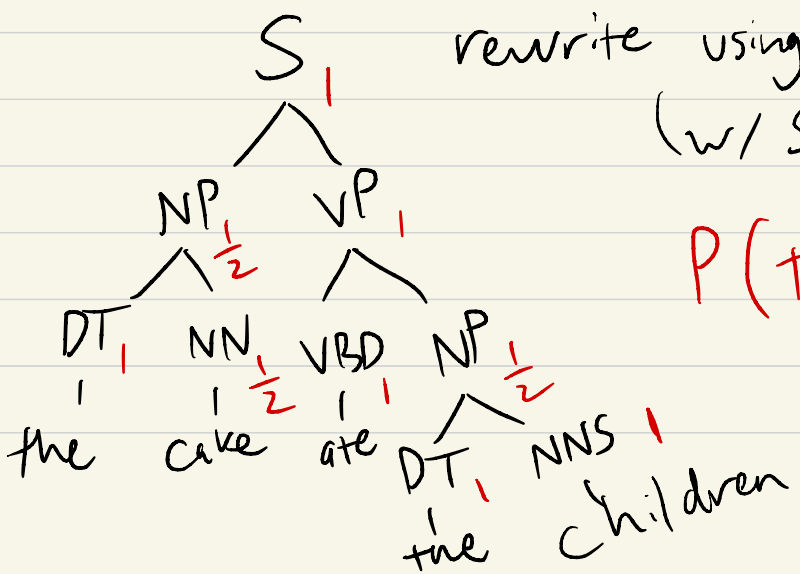
tags (preterminals)

DT, NN, IN, NNS

Rules

binary	probs	unary	
S → NP VP	1	DT → the	1
VP → VBD NP	1	NNS → children	1
NP → DT NN	1/2	NN → cake	1/2
NP → DT NNS	1/2	NN → spoon	1/2
		VBD → ate	1

CFGs define a set of trees



rewrite using a rule (w/ S on LHS)

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

Probabilistic CFG (PCFG)

Each rule has a prob.

Probs normalize per parent

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

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

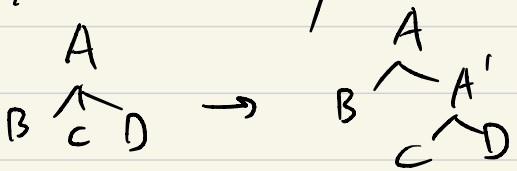
Building a Parser

Input: treebank (sents w/labeled trees)

Output: grammar + model + way to compute $P(T \mid \bar{x})$

① Grammar preprocessing

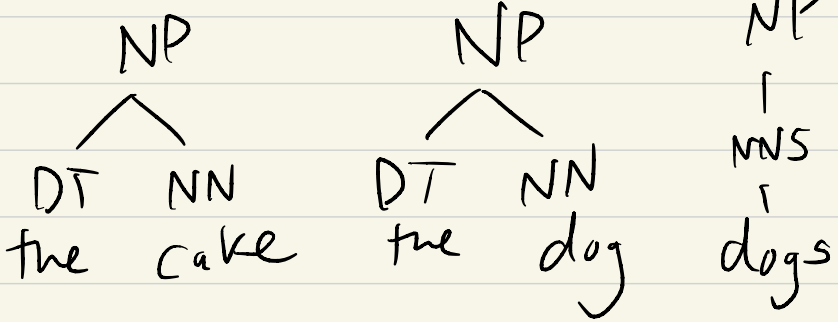
- Make every tree binary + unary



② Computing probs. for grammar

maximum likelihood estimate (MLE) probs

Treebank:



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

$P(w \mid NNS)$ 1.0 dogs

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

③ CKY algorithm

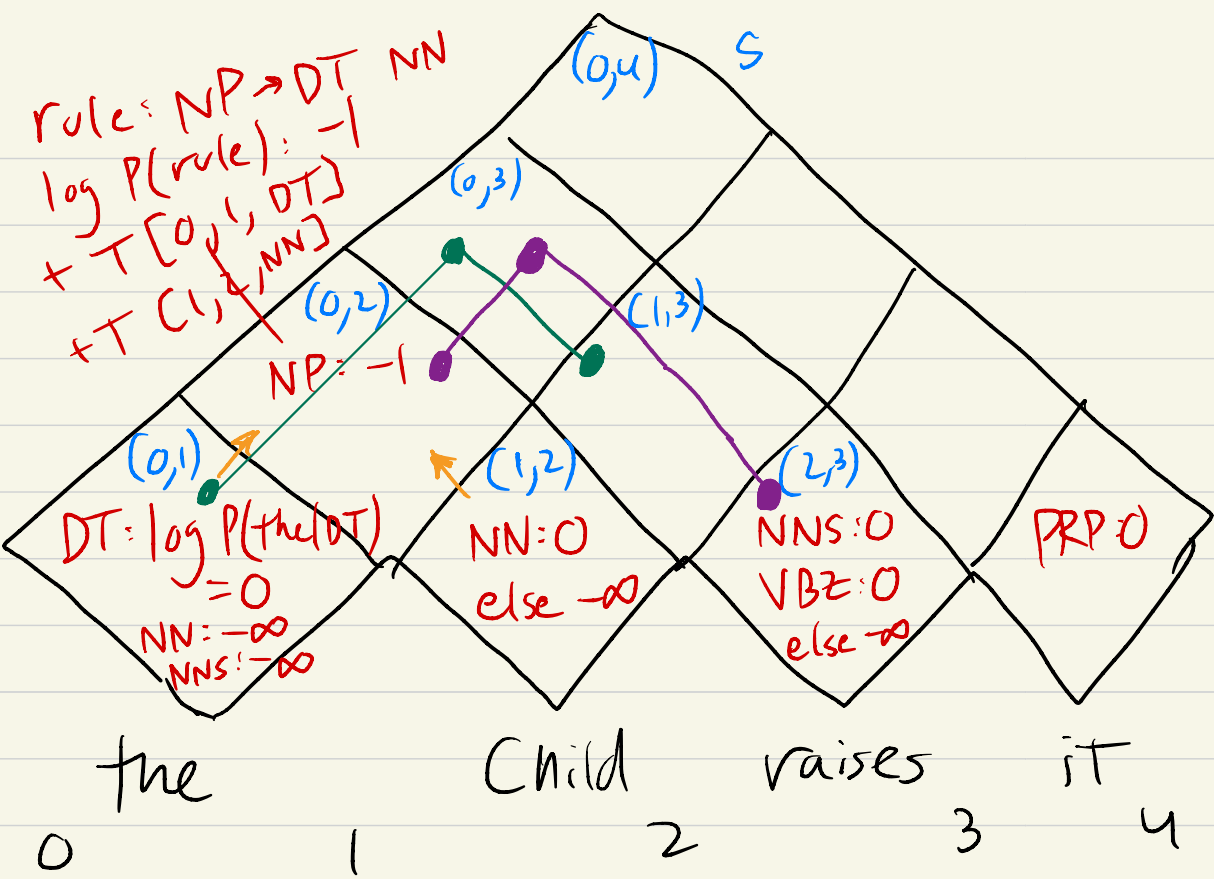
Inputs: PCFG
sentence $\bar{x}$

Output: $\underset{T}{\text{argmax}}\ P(T|\bar{x})$ most likely tree T 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 X over $\bar{x}_i \dots \bar{x}_j$ (span (i, j))



Grammar:

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

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

CKY:

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

(loop over all $X \in$ nonterminals)

Recursive case:

$T(i, j, X)$ loop over all is + js + Xs

loop over split points k

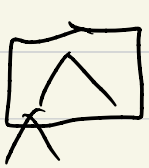
$$= \max_{k: i<k<j} \max_{X \to X_1 X_2} \left[\log P(X \to X_1 X_2) + T[i, k, X_1] + T[k, j, X_2] \right]$$

$(0, 3)$

0 • 1 • 2 • 3

k k

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



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