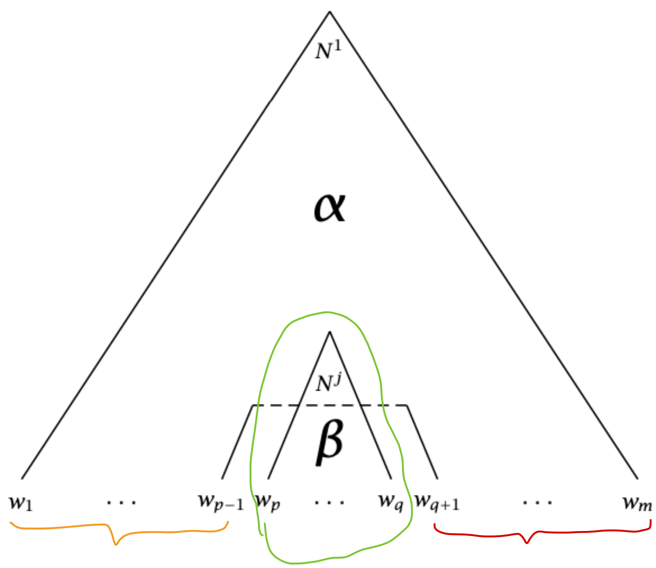
Natural Language Processing
CSE 325/425

Sihong Xie

<u>Lecture 18:</u>
- Outside probability and algorithm
- Training of PCFG

Outside probability

Alternative to inside probability, the probability of a sentence given a PCFG can be calculated using outside probability.



when

$p = q = k$

for $\forall k = 1, \cdots, m$

$$
\begin{aligned}
P(w_{1m}|G) &= \sum_j P(w_{1(k-1)}, w_k, w_{(k+1)m}, N^j_{kk}|G) \\
&= \sum_j P(w_{1(k-1)}, N^j_{kk}, w_{(k+1)m}|G) \\
&\qquad \times P(w_k|w_{1(k-1)}, N^j_{kk}, w_{(k+1)n}, G) \\
&= \sum_j \alpha_j(k,k) P(N^j \rightarrow w_k)
\end{aligned}
$$

Outside algorithm

Compute the outside probabilities using dynamic programming.

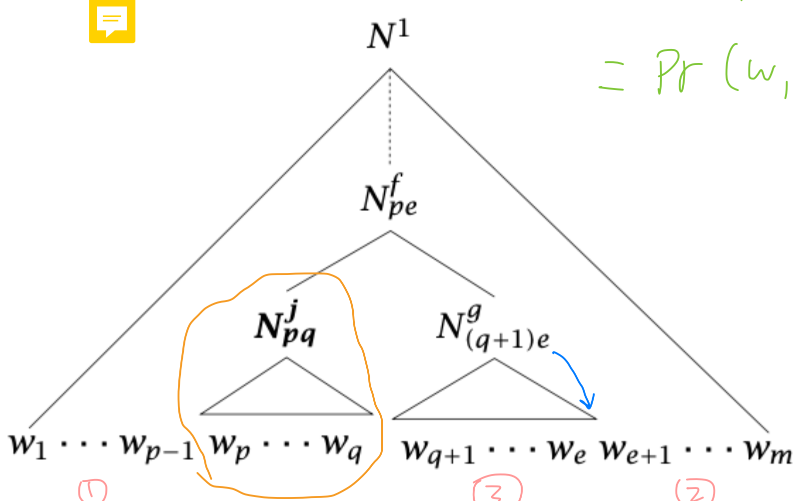
- Base case $\to N^1 = S$

$$\alpha_1(1,m) = 1 \quad \Leftrightarrow \quad Pr(\phi \mid S \to w_{1m}) = 1$$

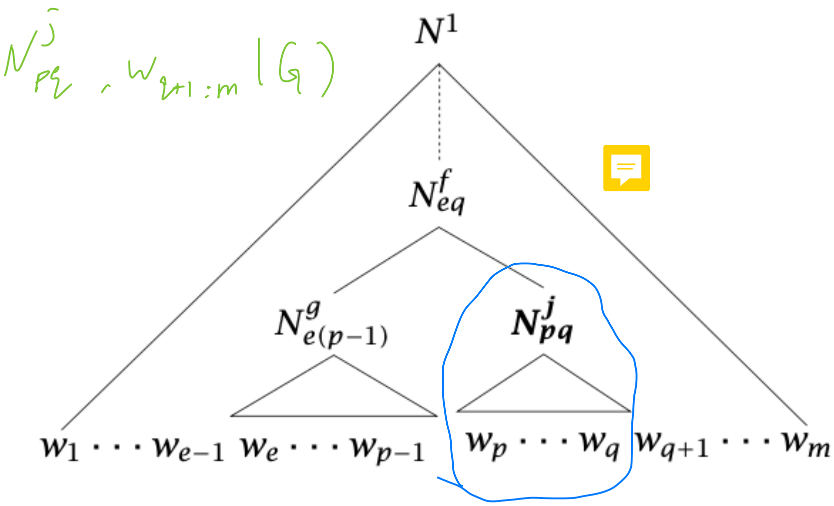
$$\alpha_j(1,m) = 0 \quad \text{for } j \neq 1$$

- Induction due to CNF: the $\underbrace{\text{target outside probability}}_{\alpha_j(p,q)}$ can be for
  - the left child the right child

$= Pr(w_{1:p-1}, N^j_{pq}, w_{q+1:m} \mid G)$



Solved by the inside Alg.



Outside algorithm

Dynamic programming: subproblems for the parent and the sibling are solved. The sub-problems are created by conditioning on:

- the left or right child, and
- the end or starting point of the sibling.

N^j_{pq} left

N^j_{pq} right

$$
\begin{aligned}
\alpha_j(p,q) &= \Big[\sum_{f,g\neq j}\sum_{e=q+1}^{m} P(w_{1(p-1)}, w_{(q+1)m}, N^f_{pe}, N^j_{pq}, N^g_{(q+1)e})\Big] \\
&\quad + \Big[\sum_{f,g}\sum_{e=1}^{p-1} P(w_{1(p-1)}, w_{(q+1)m}, N^f_{eq}, N^g_{e(p-1)}, N^j_{pq})\Big] \\
&= \Big[\sum_{f,g\neq j}\sum_{e=q+1}^{m} P(\overset{1}{w_{1(p-1)}}, \overset{2}{w_{(e+1)m}}, N^f_{pe}) P(N^j_{pq}, N^g_{(q+1)e} \mid N^f_{pe}) \\
&\qquad \times P(\overset{3}{w_{(q+1)e}} \mid N^g_{(q+1)e})\Big] + \Big[\sum_{f,g}\sum_{e=1}^{p-1} P(w_{1(e-1)}, w_{(q+1)m}, N^f_{eq}) \\
&\qquad \times P(N^g_{e(p-1)}, N^j_{pq} \mid N^f_{eq}) P(w_{e(p-1)} \mid N^g_{e(p-1)})\Big]
\end{aligned}
$$

Need inside probabilities

$$
\begin{aligned}
&= \Big[\sum_{f,g\neq j}\sum_{e=q+1}^{m} \alpha_f(p,e) P(N^f \to N^j\, N^g) \beta_g(q+1,e)\Big] \\
&\quad + \Big[\sum_{f,g}\sum_{e=1}^{p-1} \alpha_f(e,q) P(N^f \to N^g\, N^j) \beta_g(e,p-1)\Big]
\end{aligned}
$$

Probability of a sentence

Suppose the inside and outside probabilities are computed at some non-leaf node that is the non-terminal *j* and spans [p, q].

- Then the probability of the sentence with that leaf node is:

$$\begin{aligned}\alpha_j(p,q)\beta_j(p,q) &= P(w_{1(p-1)}, N^j_{pq}, w_{(q+1)m}|G)P(w_{pq}|N^j_{pq}, G) \\ &= P(w_{1m}, N^j_{pq}|G)\end{aligned}$$

- The probability of the sentence with some non-terminal spanning [p, q]:

$Pr(w_{1m}|G) \times \underline{Pr(N_{pq}|w_{1m}G)}_{=1}$ =

$$P(w_{1m}, N_{pq}|G) = \sum_j \alpha_j(p,q)\beta_j(p,q)$$

  - but this is just the probability of the sentence.

- Two special cases: $P(w_{1m}|G) = P(N^1 \overset{*}{\Longrightarrow} w_{1m}|G)$ or $= \sum_j \underline{\alpha_j(k,k)}P(N^j \to w_k)$

  (Why?) $= P(w_{1m}|N^1_{1m}, G) = \beta_1(1,m)$

$= Pr(w_{1:k-1}, N^j_k, w_{k+1:m} | G)$

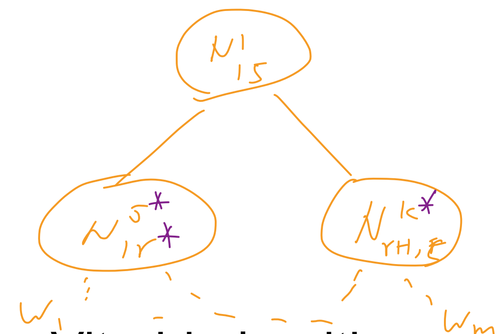
HW 5

$p=1,\ q=m$

$\alpha_1(1,m) = ?$

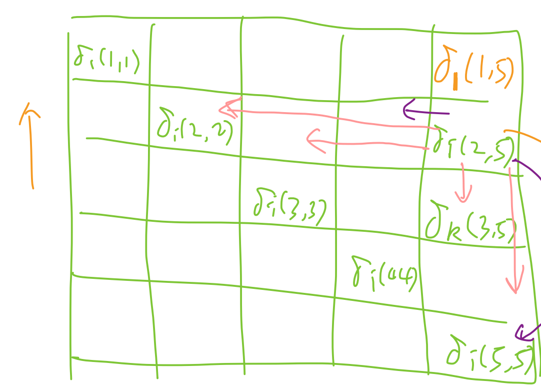
$\alpha_j(1,m) = ?$ when $j \neq 1$

Find the optimal parsing tree

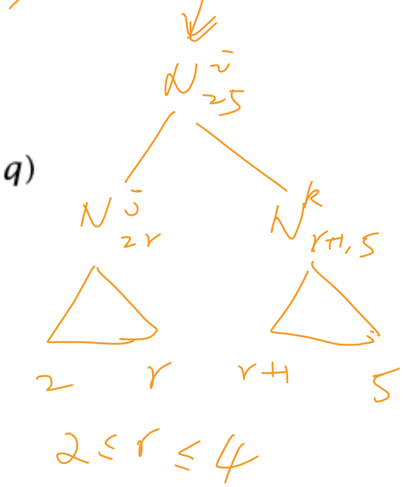


Finding the optimal tree is called “decoding”, similar to the Viterbi algorithm for HMM.

$N^1 \Rightarrow^* w_{1m}$



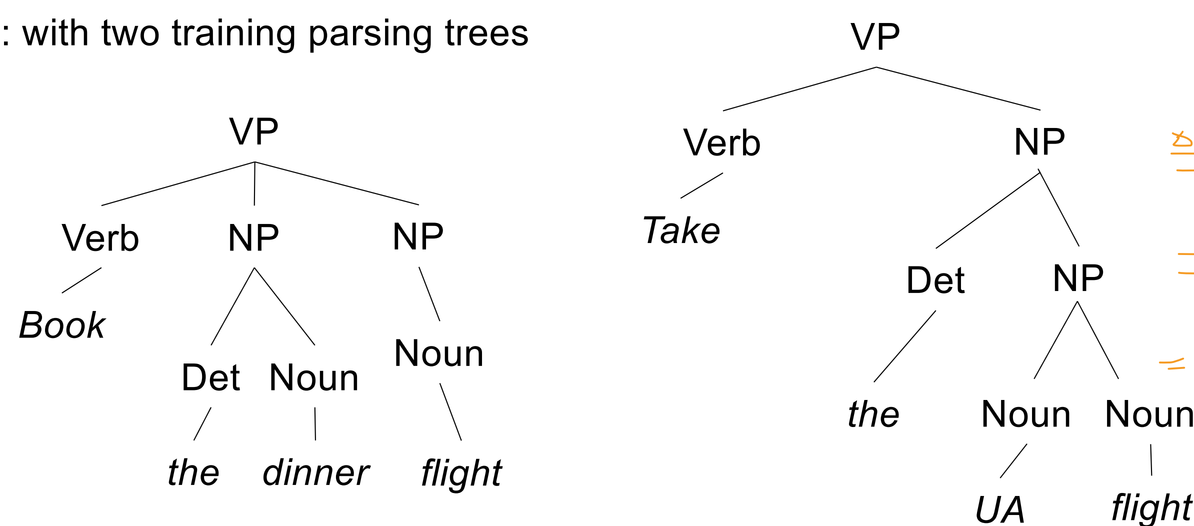
- Viterbi ≈ forward, CYK ≈ optimal-tree-finding
- Filling out the CYK matrix:
  - On the diagonal $\delta_i(p,p) = P(N^i \rightarrow w_p)$
  - In cell (*p, q*):
    - max probability $\psi_i(p,q) = \underset{(j,k,r)}{\arg\max}\, P(N^i \rightarrow N^j\ N^k)\delta_j(p,r)\delta_k(r+1,q)$
    - record the backpointers $\delta_i(p,q) = \max_{\substack{1 \le j,k \le n \\ p \le r < q}} P(N^i \rightarrow N^j\ N^k)\delta_j(p,r)\delta_k(r+1,q)$
  - At the top-right corner $P(\hat{t}) = \delta_1(1,m)$
- Reconstructing the optimal tree by back-tracing using the back-pointers.



Training of PCFG

- MLE: given a training corpus with sentences and their parsing trees (e.g., Penn Treebank),
  - extract rules observed in the corpus.
  - probability of a rule: how often it appears / how often the LHS appears.

Example: with two training parsing trees



$Pr(\text{dinner} \mid \text{Noun})$

$\triangleq Pr(\text{Noun} \rightarrow \text{dinner})$

$= \frac{1}{4}$

$= 0.25$

Noun → dinner

Noun → flight

NP → Det NP

- - -

Training of PCFG

What if we have a large number of sentences without being parsed?

- Need to deal with the unknown trees as latent structures;
- Similar to the unknown POS-tag sequences in the learning of HMM.
- EM algorithm.
  - E-step: run the inside and outside algorithms to find the inside and output probabilities. (In HMM, this is the forward-backward algorithm).
  - M-step: estimate the rule probability based on the expectation of frequencies of occurrence using the inside/outside probabilities.

Training of PCFG

EM algorithm

- Focus on the M-step, given the inside and outside probabilities.
  - Probability of a non-terminal for the range [*p, q*]:

$$P(N^j \stackrel{*}{\Longrightarrow} w_{pq} | N^1 \stackrel{*}{\Longrightarrow} w_{1m}, G) = \frac{\alpha_j(p,q)\beta_j(p,q)}{\pi}$$

  - Expected frequency of the non-terminal:

$$E(N^j \text{ is used in the derivation}) = \sum_{p=1}^{m} \sum_{q=p}^{m} \frac{\alpha_j(p,q)\beta_j(p,q)}{\pi}$$

$$P(N^j \to N^r\ N^s \stackrel{*}{\Longrightarrow} w_{pq} | N^1 \stackrel{*}{\Longrightarrow} w_{1m}, G)$$
$$= \frac{\sum_{d=p}^{q-1} \alpha_j(p,q) P(N^j \to N^r\ N^s) \beta_r(p,d) \beta_s(d+1,q)}{\pi}$$

Training of PCFG

EM algorithm

- Focus on the M-step, given the inside and outside probabilities.
  - Probability of a rule deriving words in the range [*p, q*]:

$$P(N^j \rightarrow N^r\ N^s \stackrel{*}{\Rightarrow} w_{pq} | N^1 \stackrel{*}{\Rightarrow} w_{1m}, G) = \frac{\sum_{d=p}^{q-1} \alpha_j(p,q) P(N^j \rightarrow N^r\ N^s) \beta_r(p,d) \beta_s(d+1,q)}{\pi}$$

  - Expected frequency of a rule:

$$E(N^j \rightarrow N^r\ N^s, N^j \text{ used}) = \frac{\sum_{p=1}^{m-1} \sum_{q=p+1}^{m} \sum_{d=p}^{q-1} \alpha_j(p,q) P(N^j \rightarrow N^r\ N^s) \beta_r(p,d) \beta_s(d+1,q)}{\pi}$$

  - Estimation of the conditional probability of RHS given the LHS:

$$\hat{P}(N^j \rightarrow N^r\ N^s) = \frac{\sum_{p=1}^{m-1} \sum_{q=p+1}^{m} \sum_{d=p}^{q-1} \alpha_j(p,q) P(N^j \rightarrow N^r\ N^s) \beta_r(p,d) \beta_s(d+1,q)}{\sum_{p=1}^{m} \sum_{q=p}^{m} \alpha_j(p,q) \beta_j(p,q)}$$

Motivations

EM algorithm

- Focus on the M-step, given the inside and outside probabilities.
  - Probability of the rule $N^j \rightarrow w^k$ deriving the word w^k somewhere in the sentence:

$$P(N^j \rightarrow w^k | N^1 \stackrel{*}{\Longrightarrow} w_{1m}, G) = \frac{\sum_{h=1}^{m} \alpha_j(h,h) P(N^j \rightarrow w_h, w_h = w^k)}{\pi} = \frac{\sum_{h=1}^{m} \alpha_j(h,h) P(w_h = w^k) \beta_j(h,h)}{\pi}$$

  - Conditional probability of w^k given N^j:

$$\hat{P}(N^j \rightarrow w^k) = \frac{\sum_{h=1}^{m} \alpha_j(h,h) P(w_h = w^k) \beta_j(h,h)}{\sum_{p=1}^{m} \sum_{q=p}^{m} \alpha_j(p,q) \beta_j(p,q)}$$