

Natural Language Processing

CSE 325/425



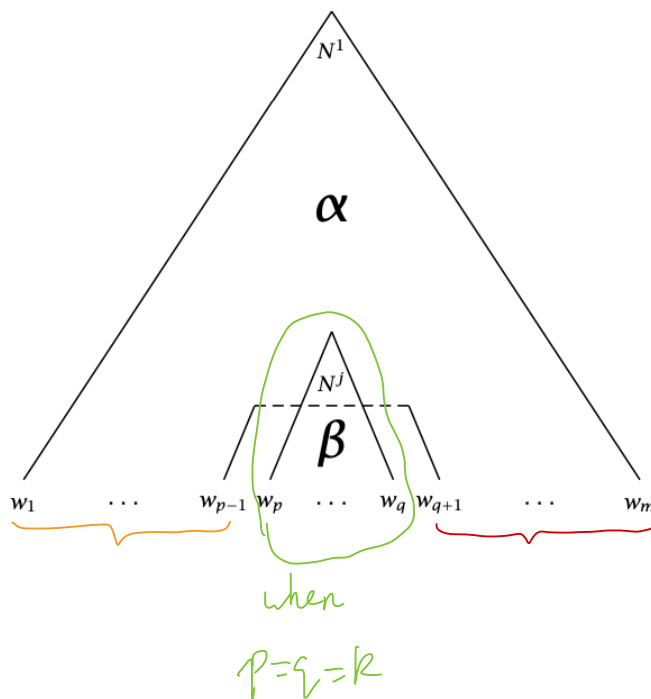
Sihong Xie

Lecture 18:

- 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.

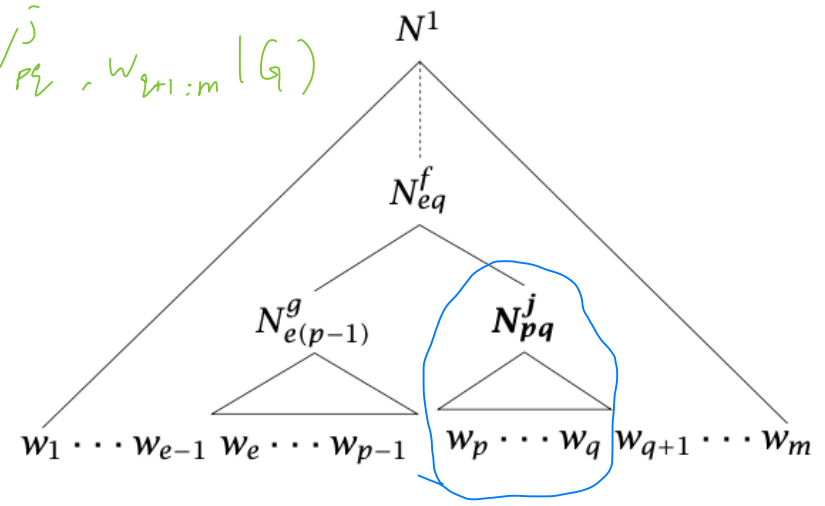
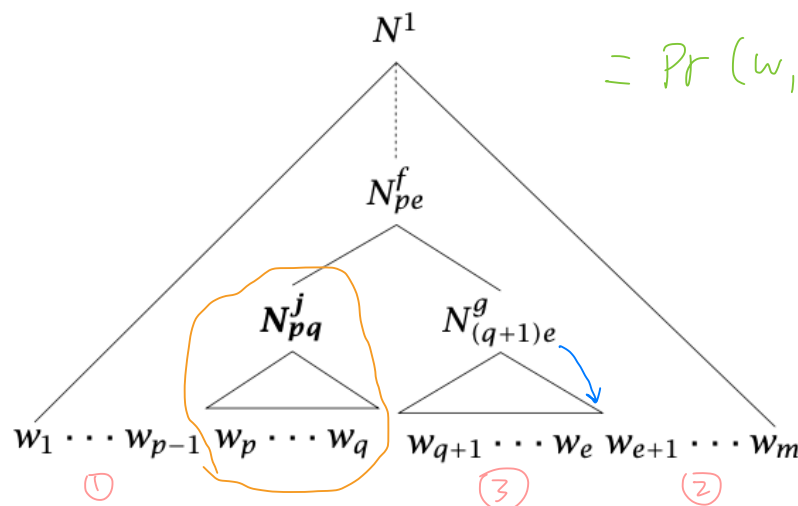


$$\begin{aligned}
 P(w_{1m}|G) &= \sum_j P(w_{1(k-1)}, w_k, w_{(k+1)m}, N_{kk}^j | G) \\
 &= \sum_j P(w_{1(k-1)}, \underbrace{N_{kk}^j}_{\text{circled}}, \underbrace{w_{(k+1)m}}_{\text{underlined}} | G) \\
 &\quad \times P(w_k | w_{1(k-1)}, N_{kk}^j, w_{(k+1)m}, G) \\
 &= \sum_j \alpha_j(k, k) P(N^j \rightarrow w_k)
 \end{aligned}$$

Outside algorithm

Compute the outside probabilities using dynamic programming.

- Base case $\rightarrow N^1 = S$
 $\alpha_1(1, m) = 1 \Leftrightarrow \Pr(\phi \mid S \rightarrow w_{1:m}) = 1$
 $\alpha_j(1, m) = 0 \text{ for } j \neq 1$
- Induction due to CNF: the target outside probability can be for
 - the left child
 - the right child



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.

$$\begin{aligned}
 \alpha_j(p, q) &= \left[\sum_{f, g \neq j} \sum_{e=q+1}^m P(w_{1(p-1)}, w_{(q+1)m}, N_{pe}^f, N_{pq}^j, N_{(q+1)e}^g) \right] \\
 &\quad + \left[\sum_{f, g} \sum_{e=1}^{p-1} P(w_{1(p-1)}, w_{(q+1)m}, N_{eq}^f, N_{e(p-1)}^g, N_{pq}^j) \right] \\
 &= \left[\sum_{f, g \neq j} \sum_{e=q+1}^m P(\overset{①}{w_{1(p-1)}}, \overset{②}{w_{(e+1)m}}, N_{pe}^f) P(N_{pq}^j, N_{(q+1)e}^g | N_{pe}^f) \right. \\
 &\quad \left. \times P(\overset{③}{w_{(q+1)e}} | N_{(q+1)e}^g) \right] + \left[\sum_{f, g} \sum_{e=1}^{p-1} P(w_{1(e-1)}, w_{(q+1)m}, N_{eq}^f) \right. \\
 &\quad \left. \times P(N_{e(p-1)}^g, N_{pq}^j | N_{eq}^f) P(w_{e(p-1)} | N_{e(p-1)}^g) \right]
 \end{aligned}$$

Handwritten notes:
 N_{pq}^j left \rightarrow
 N_{pe}^f right \rightarrow

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

Need inside probabilities \downarrow

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:

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

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

$$\begin{aligned}\Pr(w_{1m} | G) &= P(w_{1m}, N_{pq} | G) = \sum_j \alpha_j(p, q)\beta_j(p, q) \\ &\times \Pr(N_{pq} | w_{1m}, G) \\ &= 1\end{aligned}$$

○ but this is just the probability of the sentence.

- Two special cases: $P(w_{1m} | G) = P(N^1 \xRightarrow{*} w_{1m} | G)$ or $= \sum_j \alpha_j(k, k) P(N^j \rightarrow w_k)$
(Why?) $= P(w_{1m} | N_{1m}^1, G) = \beta_1(1, m)$

HW 5

$$p=1, q=m$$

$$\alpha_1(1, m) = ?$$

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

$$= \Pr(w_{1:k-1}, N_k^j, w_{k+1:m} | G)$$

Find the optimal parsing tree

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

- Viterbi \approx forward, CYK \approx optimal-tree-finding

- Filling out the CYK matrix:

- On the diagonal $\delta_i(p, p) = P(N^i \rightarrow w_p)$

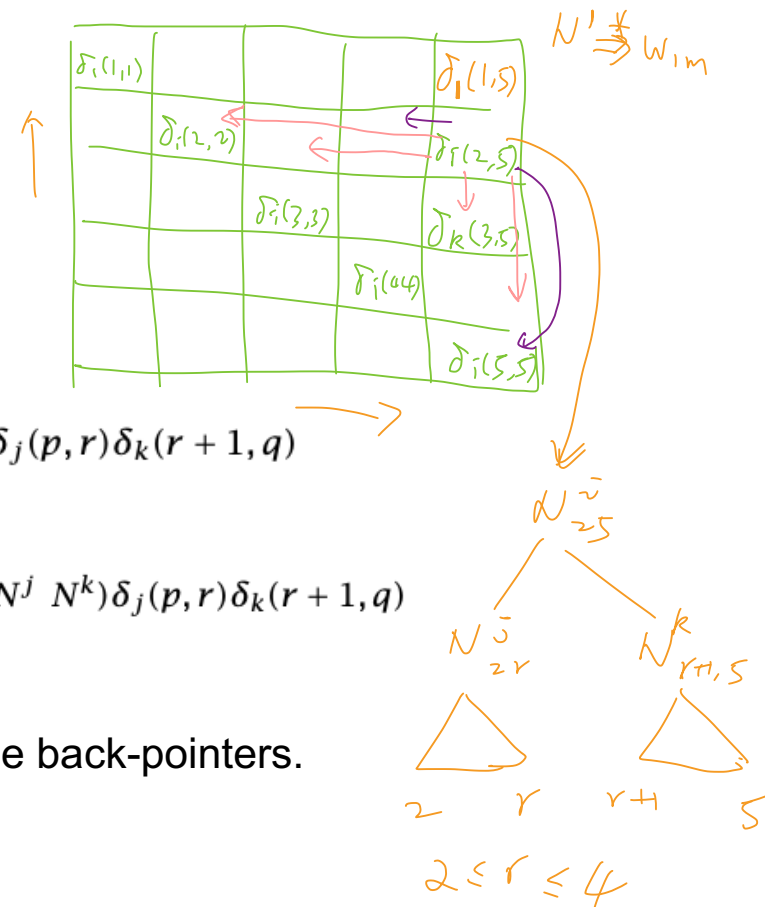
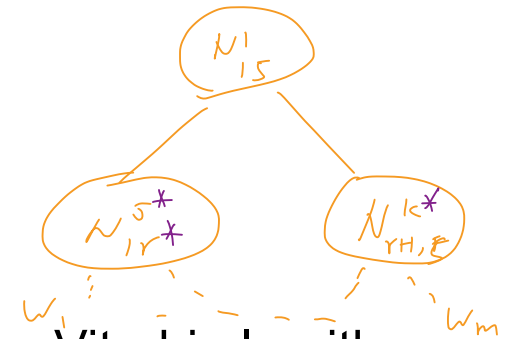
- In cell (p, q) :

- max probability $\psi_i(p, q) = \arg \max_{(j,k,r)} P(N^i \rightarrow N^j N^k) \delta_j(p, r) \delta_k(r+1, q)$

- record the backpointers $\delta_i(p, q) = \max_{\substack{1 \leq j, k \leq n \\ p \leq r < q}} P(N^i \rightarrow N^j N^k) \delta_j(p, r) \delta_k(r+1, q)$

- At the top-right corner $P(\hat{t}) = \delta_1(1, m)$

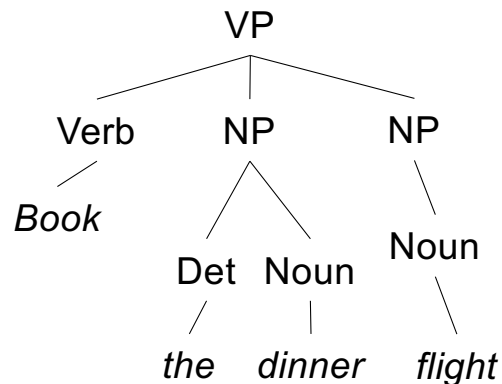
- Reconstructing the optimal tree by back-tracing using the back-pointers.



Training of PCFG

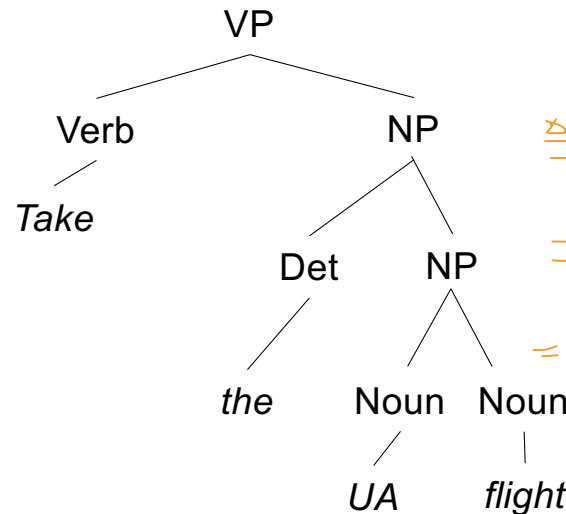
- MLE: given a training corpus with sentences and their parsing trees (e.g., Penn Treebank),
 - extract rules observed in the corpus.
 - probability of a rule: how often it appears / how often the LHS appears.

Example: with two training parsing trees



Noun → dinner
Noun → flight

NP → Det NP
- - -



$$\begin{aligned}
 &Pr(\text{dinner} | \text{Noun}) \\
 &\stackrel{\Delta}{=} Pr(\text{Noun} \rightarrow \text{dinner}) \\
 &= \frac{1}{4} \\
 &= 0.25
 \end{aligned}$$

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 $[p, q]$:

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

- Expected frequency of the non-terminal:

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

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

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 $[p, q]$:

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

- Expected frequency of a rule:

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

- 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:

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

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

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