NLP - LEZIONE 11-12

INSIDE-OUTSIDE ALGORITHM

ALGORITMO INSIDE-OUTSIDE

L'idea è quella di ASSUMERE dei parametri iniziali ϕ e utilizzare un CORPUS al fine di aggiustare in modo ITERATIVO i valori delle varie stime. Ad esempio è possibile iniziare con una DISTRIBUZIONE UNIFORME su tutte le regole della grammatica con lo stesso non-terminale $A \to \bullet$.

La probabilità di una frase W è data da

$$P_\phi(W) = \sum_T P_\phi(W, T)$$

dove $\sum_T$ è la sommatoria su tutti i parse tree della frase W, mentre ϕ è l'insieme dei PARAMETRI utilizzati per calcolare le probabilità.

Dato un CORPUS formato dalle frasi $W_1, \dots, W_n$, la LIKELIHOOD $L(\phi)$ del corpus è data da

$$L(\phi) = P_\phi(W_1) \cdot P_\phi(W_2) \cdots P_\phi(W_n)$$

L'inside-outside algorithm parte da un insieme di parametri ϕ per OTTENERE un nuovo insieme di parametri ϕ' t.c. $L(\phi') \geq L(\phi)$. Questo ricalcolo di UPDATE dei parametri viene ripetuto fino a quando la LIKELIHOOD CONVERGE.

OSS: L'INSIDE-OUTSIDE ALGORITHM è un caso peculiare dell'EXPECTATION MAXIMIZATION algorithm e viene utilizzato per MASSIMIZZARE LOCALMENTE la LIKELIHOOD dei dati di TRAINING.

Assumiamo SENZA PERDITÀ DI GENERALITÀ che la nostra PCFG sia in CNF. Fissati i parametri ϕ l'algoritmo procede nel seguente modo per aggiornarli:

- $\phi'(A \to BC) := \dfrac{count(A \to BC)}{\sum_{\alpha} count(A \to \alpha)}$
- $count(A \to BC) := \sum_{i=1}^{N} C_\phi(A \to BC, W_i)$
- $C_\phi(A \to BC, W_i) :=$ NUMERO ATTESO di volte in cui la regola $A \to BC$ viene utilizzata per generare la stringa W_1.

- $\phi'(A \to w) := \dfrac{count(A \to w)}{\sum_{\alpha} count(A \to \alpha)}$
- $count(A \to w) := \sum_{i=1}^{N} C_\phi(A \to w, W_i)$
- $C_\phi(A \to w, W_i) :=$ NUMERO ATTESO di volte in cui la regola $A \to w$ viene utilizzata per generare la stringa W_1.

Al fine di calcolare il valore di $P_\phi(A \to \alpha, w_i)$, ovvero il # medio di volte in cui viene utilizzata la regola $A \to \alpha$ per generare la parola w_i, introduciamo le seguenti probabilità

- INSIDE PROBABILITY

  Denotata con $\alpha_{i,j}(A)$, rappresenta la probabilità che il non terminale A può essere utilizzato per DERIVARE la SOTTOSTRINGA $w_i \dots w_j$ nella FRASE $w = w_1 \dots w_n$ dati i parametri specificati in ϕ.

  In formule,

$$\alpha_{i,j}(A) := P_\phi(A \overset{*}{\Rightarrow} w_i \dots w_j)$$

- OUTSIDE PROBABILITY

  Denotata con $\beta_{i,j}(A)$, rappresenta la prob. di ottenere la stringa $w_1 \dots w_{i-1} A w_{j+1} \dots w_n$ partendo dal non-terminale S e utilizzando i parametri specificati in ϕ.

  In formule,

$$\beta_{i,j}(A) := P_\phi(S \overset{*}{\Rightarrow} w_1 \dots w_{i-1} A w_{j+1} \dots w_n)$$

Al fine di calcolare queste probabilità possiamo utilizzare le seguenti RELAZIONI RICORSIVE, che funzionano se la grammatica è in CNF

- $\alpha_{i,j}(A) = \sum_{B,C} \sum_{i \le k \le j} \phi(A \to BC) \cdot \alpha_{ik}(B) \cdot \alpha_{k+1,j}(C)$

  per $i < j$. Se $i = j$ invece

  $\alpha_{ii}(A) = \phi(A \to w_{ii})$

- $\beta_{i,j}(A) = \sum_{B,C} \sum_{i \le k < j} \phi(B \to CA)\, \alpha_{k,i-1}(C)\, \beta_{kj}(B) +$

  $+ \sum_{B,C} \sum_{m \ge k > j} \phi(B \to AC) \cdot \alpha_{j+1,k}(C) \cdot \beta_{i,k}(B)$

Utilizzando queste formule si è in grado di usare ϕ e ϕ'. L'algoritmo GARANTISCE che la LOG-LIKELIHOOD non decresce, ovvero che

$$LL(\phi) := \sum_{i=1}^{N} \log P_{\phi}(w_i) \le LL(\phi')$$

L'idea è quindi quella di SMETTERE quando la differenza $LL(\phi') - LL(\phi) > 0$ risulta essere ABBASTANZA PICCOLA.

L'implementazione di questi calcoli viene effettuata utilizzando l'algoritmo CYK: durante la fase di PARSING e di costruzione della CHART vengono calcolate le INSIDE PROBABILITIES con un approccio BOTTOM-UP; successivamente vengono calcolate le OUTSIDE PROBABILITIES con un approccio TOP-DOWN.

In generale poi l'algoritmo INSIDE-OUTSIDE si basa sul calcolo del NUMERO ATTESO di volte in cui applicare una data REGOLA al fine di generare una FRASE in modo da poter AGGIORNARE le stime dei parametri.