NLP - LEZIONE 08

DEL 28/10/2019

In questa lezione cominciamo ad analizzare il livello SINTATTICO nell'analisi del linguaggio.

GERARCHIA DI CHOMSKY ~ (10:00 min)

Data delle frasi P_1 e P_2

P_1: Mario mangia una mela con il coltello.

P_2: Mario mangia una mela con il verme.

Vogliamo trovare una rappresentazione delle STRUTTURE SINTATTICHE delle frasi P_1 e P_2. Vogliamo quindi ESPLICITARE le relazioni che sussistono tra le varie parole.

Una volta che abbiamo effettuato l'ANALISI GRAMMATICALE e assegnato ad ogni PAROLA la sua rispettiva CLASSE, possiamo analizzare come le varie parole delle frasi sono collegate tra loro.

Per rappresentare queste relazioni possiamo utilizzare l'approccio di Chomsky, che consiste nel definire il concetto di GRAMMATICA FORMALE che ci permette di definire un LINGUAGGIO FORMALE.

Questi linguaggi nei sono stati classificati nella famosa GERARCHIA DI CHOMSKY in base alla loro COMPLESSITÁ, dove la complessità di un linguaggio può essere vista nei seguenti due modi:

i) Come la complessità del macchinario (AUTOMA) necessario e sufficiente per ACCETTARE il linguaggio.

ii) Come la complessità delle REGOLE della grammatica che GENERA il linguaggio.

La gerarchia di Chomsky può essere rappresentata come segue:

LIVELLO	GRAMMATICA	AUTOMA
0	?	TURING MACHINE (INDECIDIBILE)
1	CONTEXT-SENSITIVE	TURING MACHINE (DECIDIBILE)
2	CONTEXT-FREE	PUSH DOWN AUTOMATON
3	REGOLARE	FINITE STATE AUTOMATON

Chomsky era interessato a capire in quale livello potesse essere inserito il LINGUAGGIO NATURALE.

GRAMMATICHE CONTEXT-FREE ~ (28:00 min)

Lavorando con una grammatica CF è possibile rappresentare le informazioni sintattiche tramite delle STRUTTURE ARBOREE in quanto le regole sono della forma $\alpha \to \beta$, con $|\alpha| = 1$.

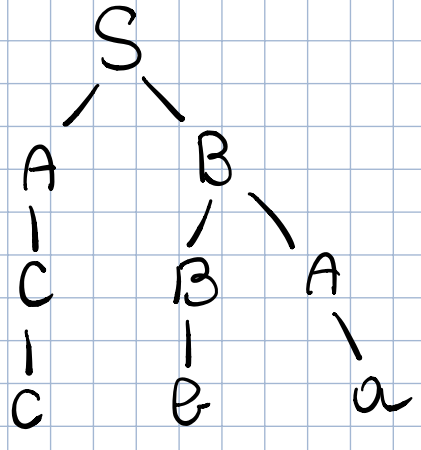
ESEMPIO:

$S \to AB \mid BC$
$A \to C \mid a$
$B \to BA \mid b$
$C \to c$

La frase cba è ottenuta dalla seguente DERIVAZIONE

$S \to AB$
$\to CB$
$\to CBA$
$\to cba$

Possiamo rappresentare la derivazione di cba nel seguente modo



Andiamo adesso a rispondere alle seguenti domande:

i) Perché ci servono le grammatiche CF e NON ci bastano le grammatiche REGOLARI.

ii) Perché ci bastano le CF e non ci servono le grammatiche CONTEXT-SENSITIVE.

Per rispondere a queste domande possiamo utilizzare il PUMPING LEMMA, che ci permette di capire a quale livello appartiene un linguaggio. Abbiamo due versioni del pumping lemma:

1) REGULAR → CF

La prima versione è utilizzata per dim. che il linguaggio sotto analisi non è regolare.

In particolare ci dice che se nel linguaggio non presenti stringhe della forma $a^n b^n$, allora il linguaggio non è regolare.

2) CF $\longrightarrow$ CS

Utilizzato per dim. che un linguaggio non è CF.

Ci dice che stringhe della forma $a^n b^n c^n$ non fanno parte del livello context-free.

Per capire dove si colloca il linguaggio naturale devo quindi analizzare le strutture prodotte e capire se ci sono strutture della forma $a^n b^n$ o $a^n b^n c^n$.

Q: Le lingue naturali come l'italiano o l'inglese sono REGOLARI o no?

R: Non sono regolari in quanto l'italiano, ad esempio, quando trovo un "CHE" devo anche contare un VERBO. Tra il "CHE" e il VERBO sono ovvero anche varie frasi tra loro. Dovendo contare qualcosa, non concludere che l'italiano non è un linguaggio REGOLARE.

! Potrei però fissare un NUMERO MASSIMO DI INNESTI DA CONTARE e utilizzare un automa a stati finiti per analizzare il CORPUS.

OSS: In inglese WITCH viene utilizzato per indicare che una frase è ACCESSORIA, mentre il THAT indica che la frase non è ACCESSORIA.

OSS: Gli LSTM sono in grado di riconoscere i linguaggi naturali anche se non sono context-free.

Consideriamo ora la frase

[Pino, Rino e Gino sono fratelli, figlio e nuora di GAETANO]

Notiamo che lavorando con una grammatica CF non siamo in grado di analizzare strutture del genere della forma $(_1 \; (_2 \;)_1 \;)_2$.

Dato però che questi casi SONO POCHI, l'idea è quella di rimanere nel livello CF e qualora dovessimo trovare frasi del genere, possiamo rimandare la loro analisi a livelli successivi.

PARSING ~ (1:05:00 min)

Data una grammatica $G = \langle R, NT, T, S \rangle$ CF e data una frase $w_1 \dots w_n$, $w_i \in T^*$, la mossa che costruisce l'ALBERO SINTATTICO di $w_1 \dots w_n$ secondo G è detta PARSING.

Il PARSING ci permette di capire se la frase $w_1 \dots w_n$ è generata da G, e in caso positivo costruisce l'albero sintattico.

Consideriamo la grammatica

$S \to AB$
$S \to ASB$
$A \to a$
$B \to b$

e sia 'aabb' la frase da analizzare.
Abbiamo varie tecniche per effettuare il parsing, tra cui:

- <u>TOP-DOWN PARSING</u>
  L'idea è quella di guardare tutte le regole della grammatica ogni singole volte.

$S \to AB \to aB$

$S \to ASB \to AABB \to aaBB$

A questo punto ho ottenuto la FRONTIERA e rigetto.

È possibile migliorare questa procedura utilizzando dei LOOK-AHEAD a partire dalla frontiera.

- BOTTOM-UP PARSING

Si parte dalle foglie della frase per vedere se possiamo applicare delle regole della grammatica per ottenere la radice in analisi.

CYK - PARSING ~ (1:23:00 min)

Il CYK è un algoritmo di parsing di tipo BOTTOM-UP utilizzando la PROGRAMMAZIONE DINAMICA.

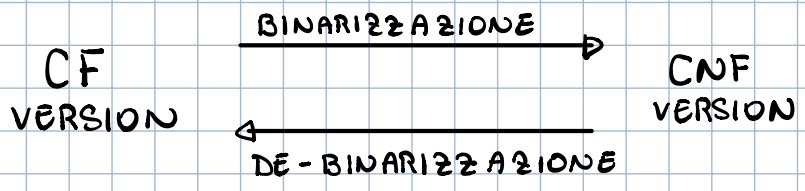
L'algoritmo lavora con una forma particolare della grammatica CF detta CNF (CHOMSKY NORMAL FORM), che è la seguente

$$CNF \begin{cases} NT \to NT_1\ NT_2 \\ NT \to T \end{cases}$$

Notiamo che gli alberi ottenuti da grammatiche di tipo CNF sono ALBERI BINARI.

Notiamo poi che le grammatiche CNF sono (DEBOLMENTE) equivalenti a quelle CF, nel senso che generano le stesse frasi ma non gli stessi alberi.

Tramite dei processi di conversione possiamo passare dalla versione CF generale a quella in CNF data una grammatica CF.



Consideriamo la seguente grammatica

G:

S	→	AB | BC
A	→	BC
B	→	BB | CC
C	→	a
B	→	b
A	→	a

Andiamo a vedere come passare la stringa

b a a b e

Per parsare, il CYK costruisce pure pure una MATRICE che contiene le informazioni nel corso di PARSING. La matrice utilizzata è la seguente

b	a	a	b	a
B	S, A	B	B	S, A
	A, C	B	B	S, A
		A, C	S	Ø
			B	S, A
				A, C

Ogni cella contiene i non terminali che permettono di ottenere una sottostringa della stringa in input.

Una possibile esecuzione riempie le diagonali della matrice a partire dalla diagonale principale. Ad ogni passo andiamo a confrontare delle coppie di celle riempite nel passo precedente.

Alla fine del processo se riesco ad inserire lo START SYMBOL S nella cella in alto a destra, quella associata all'intera stringa, allora la stringa è generata dalla grammatica.

Se la stringa è generata, per costruire l'ALBERO SINTATTICO dobbiamo ogni volta mantenerci dei RIFERIMENTI alle alle utilizzate per produrre i non terminali.

In realtà molto spesso non abbiamo un solo albero, in quanto la grammatica utilizzata potrebbe ambigua AMBIGUA.

In caso di grammatiche eccessivamente ambigue possiamo avere TANTI alberi sintattici. In questi casi capire quale albero scegliere può essere MOLTO DIFFICILE.