

Università degli studi di Pisa
Facoltà di scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica

ATN & CHART-DIAGRAM

Corso di Intelligenza Artificiale: Trattamento Automatico
del Linguaggio Naturale

Prof. Amedeo Cappelli
Presentazione: Elena Fiore

1

Il del linguaggio naturale nella società dell'informazione

- L'informazione scientifica, tecnica e relativa alle imprese raddoppia, oggi, in meno di un anno;
- Tra il 70% e il 90% di tutta l'informazione digitale prodotta viene memorizzata in forma di testo;
- Crescono le possibilità di calcolo sia in termini di velocità che di massa di dati memorizzati;
- Velocità e massa devono essere correlate alla possibilità di utilizzare selettivamente l'informazione ;
- Si rende necessaria la creazione e la reperibilità di contenuti ovvero di significati legati a parole, frasi e altre parti del discorso e del testo;



Da qui la necessità dell'elaborazione elettronica della lingua per adeguarla ai nuovi supporti sui quali i suoi prodotti vengono generati, trasmessi e recuperati.

2

Sistemi basati sul linguaggio naturale e parser

- Un sistema basato sul linguaggio naturale procede all'analisi di una frase in due momenti distinti:
 - l'analisi sintattica o parser, cui spesso è associata l'interpretazione semantica;
 - l'analisi contestuale.
- Un parser è un programma che si articola in un numero ristretto di funzioni:
 - scansione della frase in input;
 - scansione della grammatica per il reperimento delle regole rilevanti;
 - strutturazione della frase e memorizzazione progressiva.

3

Augmented grammars

- L'uso di una grammatica CF \Rightarrow limitazione del parser ad accettore di linguaggio.
- Per memorizzare la struttura di una frase e per rifiutare frasi mal-formate la grammatica viene estesa con features o augmentations:

NP \rightarrow Det N



(NP NUMBER ?n) \rightarrow (Det NUMBER ?n) (N NUMBER ?n)

4

ATN Augmented Transition Networks

Un ATN è un grafo in cui:

- (a) i nodi sono etichettati con nomi di altri puntatori di tipo ATN;
- (b) gli archi sono contrassegnati da:
 - categorie di parole
 - identificatori (puntatori) di altri ATN
 - test.

Esistono altri due tipi di archi:

- archi di tipo done
- archi di tipo jump

5

ATN Augmented Transition Networks

Si entra in un ATN quando si incontra un nodo di tipo NP.
Si esce quando si attraversa un arco di tipo "done".

Per attraversare "legalmente" un arco occorre che:

- tutti i test siano soddisfatti dal valore della variabile corrente.
- se l'arco è etichettato con una categoria di parole, la parola corrente deve appartenere a tale categoria.

Durante l'attraversamento degli archi l'ATN assegna dei valori agli slots (o augmentations) delle strutture create.

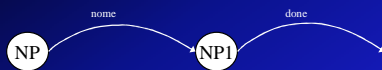
Per attraversare un arco etichettato con l'identificatore di un altro ATN si invoca la corrispondente network.

Il risultato è utilizzato nel proseguimento dell'analisi.

6

Alcuni esempi di ATN

Consideriamo il parser della frase: "Mary"



Arco	Azione
NP/nome	NAME=* NUM=3s

Parola	Tipo
Mary	nome

7

Alcuni esempi di ATN

Consideriamo ora la frase: "Mary cammina".



Arco	Test	Azione
S/NP		SUBJ=* MOOD=dichiarativo
S1/verbo	NUM(SUBJ)∩NUM(*)	VERB=* NUM=NUM(SUBJ)∩NUM(*)

8

Alcuni esempi di ATN

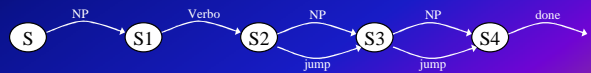


Arco	Test	Azione
NP/Articolo		DET=*, NUM=NUM(*)
NP/Pronome		Pronome=*, NUM=NUM(*)
NP/Sostantivo	NUM(*)=3p	Sostantivo=*, NUM=3p
NP/Nome		Nome=*, NUM=3s
NP1/Aggettivo		AGG=append(Agg.*)
NP1/Sostantivo	NUM(*)∩NUM	Sostantivo=*, NUM=NUM(*)∩NUM

9

Alcuni esempi di ATN

Arco	Test	Azione
S/NP		SUBJ=*, MOOD=dichiarativo
S1/Verbo	NUM(SUBJ)∩NUM(*)	VERBO=*
S2/NP	Transitivo(VERBO)	NUM=NUM(SUBJ)∩NUM(*)
S2/jump	¬Transitivo(VERBO)	OBJ=*
S3/NP	Bi-Transitivo	IND-OBJ=OBJ
		OBJ=*



10

Alcuni esempi di ATN

Un ATN per i verbi composti:

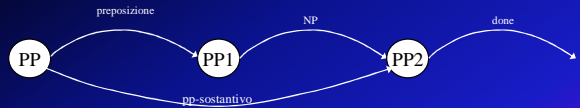


Arco	Test	Azione
VERBO/do		AUX=do FIRST-V=*
V1/verbo	If AUX=do, TENSE(*)=infinito Else TENSE(*)=presente o pass.	VERBO=*
V2/done		If FIRST-V=nil, FIRST-V=*, TENSE=TENSE(FIRST-V)

11

Alcuni esempi di ATN

Un ATN per le prepositional-phrases:



Arco	Azione
PP/preposizione	PREP=*
PP/pp-sostantivo	PP=*
PP1/NP	OBJ=*

12

L'algoritmo di parsing tramite ATN

1. Settare il puntatore ATN ad (S) e il puntatore alla locazione nella frase all'inizio della frase in questione.
2. Scegliere un arco di uscita dal nodo corrente. Per poter attraversare "legalmente" l'arco dovranno essere verificate le seguenti condizioni:
 - (a) Tutti i test dovranno essere soddisfatti dal valore della variabile corrente.
 - (b) Se l'arco è etichettato con una categoria di parole, la parola corrente dovrà appartenere a tale categoria.

13

3. Eseguire le azioni associate all'arco. In più:
 - (a) Se l'arco è una categoria di parole, aggiornare, di una parola, la posizione corrente nella frase e sostituire il nodo corrente con il destinatario dell'arco in esame;
 - (b) Se l'arco corrisponde ad un altro ATN, inserire il nodo iniziale dell'ATN nella pila dei puntatori ATN;
 - (c) Se l'arco è di tipo jump, sostituire il nodo corrente con il destinatario dell'arco in esame;
 - (d) Se l'arco è di tipo done, eliminare il nodo corrente dai puntatori ATN e settare * al valore restituito da questo nodo. Se il puntatore ATN è ora vuoto e tutte le parole del testo sono state processate, restituire *. Se il puntatore ATN è vuoto e rimane testo da analizzare, fallire. Altrimenti ritornare al passo (2).

14

TOP-DOWN VS BOTTOM-UP

- La strategia top-down è altamente predittiva indicando già la possibile struttura della frase \Rightarrow essa riduce l'ambiguità delle parole.
- La strategia bottom-up non possiede, invece, questo vantaggio dato che espande i nuovi archi tenendo conto solo del primo costituente riconosciuto.
- D'altra parte, ogni volta che l'algoritmo top-down ha bisogno di fare backtracking esso torna ad esaminare nuovamente i componenti della frase che aveva precedente processato per predizioni che si sono rivelate errate.
- L'algoritmo bottom-up, invece, esamina ogni elemento di una frase solo una volta evitando di riscriverlo più volte se utilizzato in diverse regole.
- E' possibile, combinando i due metodi, tentare di avvantaggiarsi delle peculiarità di uno e dell'altro.

15

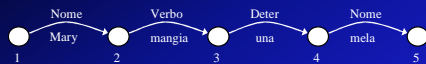
Chart Parsing

- Un chart può essere visualizzato come un grafo i cui nodi rappresentano punti nella frase e i cui archi rappresentano, invece, i costituenti.
- Ogni nodo corrisponde allo spazio tra due parole (più l'inizio e la fine).
- Ogni arco ha il nome del costituente che inizia e finisce ai nodi che connette.
- Il chart iniziale contiene solo gli archi corrispondenti alle singole parole dell'input e alle loro categorie lessicali.
- Non appena un costituente viene trovato viene aggiunto al chart.

16

Alcuni esempi di Chart

Il Chart iniziale della frase: "Mary mangia una mela"



In cui assumiamo una grammatica con seguenti regole:

$S \rightarrow NP VP$
 $NP \rightarrow \text{Determiner } NP2 \mid NP2$
 $NP2 \rightarrow \text{Nome} \mid \text{Aggettivo } NP2 \mid NP2 PP$
 $PP \rightarrow \text{Preposizione } PP$
 $VP \rightarrow \text{Verbo} \mid \text{Verbo } NP \mid VP PP$

17

Le componenti di un Chart

- Gli archi di un chart possono essere completi o attivi.
- Un arco è completo quando la sua remainder è vuota, altrimenti è attivo.
- Un arco completo rappresenta un costituente che è stato individuato.
- Un arco attivo rappresenta uno stadio nella ricerca di un costituente.
- La sua remainder rappresenta cosa deve ancora essere trovato.
- Un arco può essere specificato tramite il numero del nodo di inizio e del nodo di fine insieme con la rappresentazione della regola della grammatica su cui è basato e la remainder di quella regola ancora da completare.
- Il numero del nodo di fine viene utilizzato per separare la parte della regole che è già stata analizzata dalla remainder.

18

Alcuni esempi di archi di un Chart

- Una regola del tipo: (${}_4NP \rightarrow \text{DET } {}_5NP2$) rappresenta un arco attivo con nodo di inizio 4, nodo di fine 5, che sta analizzando un costituente di tipo NP, che ha già trovato il suo DET e che ha ancora bisogno di NP2 per diventare completo.
- Un arco completo avrà il numero di nodo di fine alla fine della regola come in: (${}_4NP \rightarrow \text{DET } NP2 {}_6$).

19

Chart Parser

- Il parser si serve del chart introducendovi un nuovo costituente ogni volta che una regola viene applicata con successo.
- Ogni volta che c'è un non-terminale all'inizio della remainder cerca per altri costituenti all'interno del chart.
- Se c'è un arco per il corrispondente simbolo nel punto corrente nel chart, il parser può avanzare nella frase fino al punto di fine di quell'arco senza bisogno di eseguire nuovamente l'analisi del costituente.
- Di conseguenza possiamo ottenere il risparmio di una notevole quantità di inefficienza combinatoriale rispetto al parser top-down.

20

L' algoritmo di parsing tramite Chart

1. Il chart viene inizializzato in modo da rappresentare le parole della frase di input con le relative categorie lessicali e propone il simbolo iniziale della grammatica come primo nodo del chart.
2. Ripete i seguenti passi:
 - (a) rimuove ciascun membro della lista degli archi pendenti del chart e lo considera come nuovo arco;
 - (b) combina il nuovo arco con il chart (questo potrebbe produrre nuovi archi pendenti);
 - (c) se il nuovo arco è attivo, propone il primo simbolo del suo remainder come suo nodo di fine nel chart (questo può produrre nuovi archi pendenti)

21

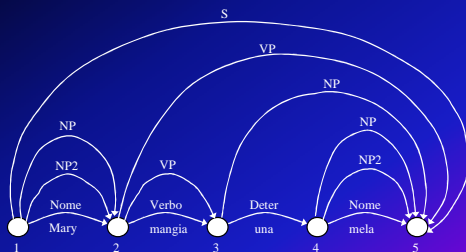
L' algoritmo di parsing tramite Chart

3. Il parser riesce se, ad un certo punto, tra gli archi o tra gli archi pendenti ce n'è uno completo tale che:
 - (a) il suo nodo di inizio è uguale al primo elemento dei nodi del chart;
 - (b) il suo nodo di fine è uguale all'ultimo elemento dei nodi del chart;
 - (c) la sua label è uguale al simbolo iniziale della grammatica.
4. Se non ci sono archi pendenti quando deve esserne scelto uno, il parser fallisce.

22

Alcuni esempi di Chart

Il Chart finale della frase: "Mary mangia una mela"



23