

L'Intelligenza Artificiale

di Raffaello De Masi

Proviamo un po' a discorrere

Seconda parte

Finora abbiamo comunicato con il calcolatore in un modo molto rigido e ristretto. Il calcolatore, cioè, ha la capacità di comprendere da noi solo poche parole (nella puntata precedente erano i quattro punti cardinali, ma è ovvio che per quanto sia possibile allargare l'esempio, tanto per dire, alle direzioni intermedie, SE, SW ecc. e magari, ancora ampliando la scelta a quelle subintermedie, SSW, MNW, WNW e così via, le scelte saranno però sempre ristrette ad un certo range), per di più fornite ad esso solo in un certo modo. Tanto per essere un po' più elastici abbiamo previsto anche la possibilità di errori di battitura alla tastiera, errori che, data la lunghezza di due delle possibili risposte, (EST, SUD), sono stati ridotti a non più di uno. Certo è che, anche in questo caso, l'intelligenza sta di casa ancora molto lontano. Tanto per intenderci, basta che, nella risposta, venga inserito inizialmente uno spazio, che ci si ritrova in una situazione di errore, anzi di completo disastro, visto che il confronto stringa d'input — risposta darebbe risultati del tutto negativi, come

se avessimo risposto, alla domanda su quale direzione prendere, la parola «Rigatoni» della più famosa pubblicità.

Il fatto è che, nel mondo reale, al contrario di quanto si immaginerebbe a prima vista, il linguaggio è qualcosa di estremamente sofisticato e complesso e, soprattutto, sovente dotato di significato diverso a seconda delle circostanze. La comprensione corretta di una frase scritta o parlata è una cosa così ardua, legata com'è al contesto, alla posizione nel discorso e, addirittura, al tono con cui viene detta, oltre che, ovviamente allo stretto significato lessicale, che solo una mente umana è capace di intenderne davvero il senso. Non solo, ma ve lo immaginate il problema di far intendere al computer un modo di dire, come ad esempio «Questione di lana caprina» o magari «Qui casca l'asino», visto che, talvolta, potrebbe anche intendersi che l'asino casca davvero?

Un linguaggio umano è fatto di frasi aventi o no tra loro un nesso logico. Il vocabolario italiano Garzanti della lingua italiana definisce, come frase, un'espressione linguistica avente senso comune.

Semplice, no?

Ma provate a renderlo comprensibile per un calcolatore! E se già la definizione di frase è così complicata, fi-

guriamoci una frase stessa che cosa sarà!

Nelle note di questo mese cercheremo di scoprire il modo di far intendere al calcolatore una frase di senso compiuto, vale a dire vedremo di stabilire uno standard di frase che il computer possa analizzare, e di cui possa riconoscere le parti e le rispettive funzioni nel contesto.

La prima cosa da fare è l'eliminazione di tutti i «leading and trailing blank» per dirla in gergo, vale a dire di tutti gli spazi inutili. Tale problema sarà parso importante a molti estensori di linguaggi visto che generalmente non manca, in alcuno di essi, una funzione adatta allo scopo. Noi siamo in Basic e, generalmente, la funzione a noi riservata è TRIM \$ che viene così usata:

```
.....
200 INPUT IN$
210 IN$ = TRIM $(IN $)
per cui la stringa in input
... VIVA ... L'AVELLINO.....
```

dove... rappresentano spazi bianchi di viene

VIVA L'AVELLINO
senza spazi bianchi all'inizio ed alla fine. Nel caso il Basic in vs. possesso non prevedesse la funzione, e dato che questa è necessaria per il nostro dire, occorrerà costruirvi una routine adatta allo scopo. Essa potrà essere simile alla figura 1.

Questa routine funziona come TRIM \$; elimina cioè gli spazi iniziali e finali e, inoltre, quelli intermedi superiori ad 1. Per chi usa un linguaggio che consente la costruzione di procedure risulta semplice costruirsi un comando (magari proprio TRIM \$) adatto all'uso. Come al solito, anche stavolta, chi ha pratica di Assembler può farsi tutto da solo.

Bene, cominciamo a stabilire delle regole generali per la costruzione di una frase comprensibile al calcolatore. La prima è che la frase debba essere conclusa con un segno di interpunzione che, nella forma generale può essere rappresentato da un punto fermo, (vedremo, poi, che sarà necessario eliminarlo). È importante per poter consentire al calcolatore di riconoscere la fine della frase che, in input, potrebbe

Figura 1

```
15000 REM ROUTINE DI ELIMINAZIONE SPAZI BIANCHI SUPERIORI AD 1
15010 B2 = LEN(IN$) : REM ELIMINAZIONE BLANK INIZIALI
15020 FOR H = 1 TO B2
15030 IF IN$[1,1] = CHR$(32) THEN IN$=IN$[2,LEN(IN$)] ELSE H =
B2
15040 NEXT H
15050 REM ELIMINAZIONE BLANK FINALI
15060 REM ESSA PUO' ESSERE INTEGRATA ALLA ROUTINE BLANK
15070 REM INIZIALI , MA VIENE QUI SEPARATA PER CHIAREZZA.
15080 B2 = LEN(IN$) : REM LA VERIFICA DI B2 VIENE ESEGUITA
SULLA NUOVA LUNGHEZZA.
15090 FOR H = 1 TO B2
15100 IF IN$[LEN(IN$),LEN(IN$)] = CHR$(32) THEN IN$ = IN$[1,
LEN(IN$)-1] ELSE H = B2
15110 NEXT H
15120 REM ROUTINE ELIMINAZIONE BLANK INTERMEDI
15130 B2 = LEN(IN$) : REM NUOVO RIAGGIORNAMENTO VARIABILE B2
15140 FOR H = 2 TO B2-1 : REM INUTILE PROVARE PER LA PRIMA ED
ULTIMA LETTERA, VISTO CHE NON POSSONO ESSERE BLANK.
15150 IF IN$[H,H] = CHR$(32) AND IN$[H+1,H+1] = CHR$(32) THEN
IN$= IN$[1,H-1]&IN$[H+1,LEN(IN$)]
15160 NEXT H
```

essere rappresentata da stringhe diverse.

Il punto non verrà distanziato dalla fine della frase con uno spazio bianco. È questa ancora una convenzione necessaria, in quanto la presenza dello spazio di separazione porta a condizioni di ambiguità non facilmente risolvibili.

Cominciamo adesso ad affrontare la frase di senso compiuto; ed iniziamo osservando appunto alcuni esempi di proposizione.

La prima

IO MANGIO

(da questo momento non inseriremo mai il punto finale, presupponendo che esso venga inserito automaticamente dalla routine addetta).

È formata dal solo soggetto e dal verbo.

IO MANGIO CARNE

Inserendo un (complemento) oggetto si esercita una drastica risoluzione dei significati possibili mentre in

IO MANGIO CARNE LESSA

si introduce, con un aggettivo, una ulteriore limitazione e precisazione dell'oggetto. Ancora

IO TALVOLTA MANGIO CARNE LESSA

si qualifica stavolta non l'oggetto, ma il verbo.

La parola più importante in tutti gli esempi precedenti è MANGIO, che rappresenta il concetto principale. Il cambiamento di qualunque altro termine, infatti, non modifica l'azione principale, cosa che succede, invece, cambiando il verbo. Il secondo esempio è più informativo indicando che solo un particolare tipo di cibo, CARNE, viene mangiato (l'aggiunta di un aggettivo, LESSA, esegue una particolare scelta del tipo di carne) e, infine, la vita si complica ancora quando un avverbio, TALVOLTA, si aggiunge alla compagnia.

Come può, un programma, analizzare questa frase? La risposta sta nella ricerca di una struttura logica nel periodo, magari da poter poi generalizzare convenientemente.

Vediamo come ciò è possibile:

a) la frase inizia con un soggetto (IO), e termina con un punto (messo dall'operatore, o dalla routine apposita).

b) La parola dopo il verbo MANGIO, è l'oggetto.

c) Se la frase dopo l'oggetto continua (la parola successiva non è un punto fermo), si tratta di uno o più aggettivi.

d) Se la parola prima del verbo non è il soggetto, allora si tratta di un avverbio.

Certo che è una forma un poco rigida di costruzione di una frase, ma al-

meno, in questo stadio, dobbiamo accontentarci; si vedrà poi in seguito come sarà possibile rendere più elastica la cosa.

Anzi, per non appesantire nuovamente la trattazione, stabiliremo di usare, dove non specificato direttamente, un solo aggettivo. Inoltre, proprio per onore della semplicità, supporremo sempre frasi senza articoli e preposizioni. Dobbiamo pure adattarci; siamo solo alla base dell'Everest dell'A.I.; Robinson Crusoe non poteva pretendere, il giorno dopo aver trovato Venerdì, di leggergli il «Sogno di una notte di mezza estate» e sperare di essere applaudito!

È necessario, a questo punto, fornire al computer un vocabolario, che gli servirà da manuale di confronto per le parole componenti la frase che gli forniremo. Ciò può essere fatto con una nuova subroutine (fig. 2), dotata di sequenze READ e DATA che assegnino ad array separate (monodimensionali) vocaboli relativi a classi sintattiche diverse.

Le array sono dimensionate al numero delle parole disponibili (non viene utilizzato il posto 0 pur senza imporre l'option base, rappresentante sempre una limitazione più che un vantaggio) e la lunghezza delle stringhe inseribili (10 caratteri) è funzione della più lunga delle parole (PROSCIUTTO) precedenti.

Adesso il testo è frantumato nei suoi componenti sia sullo schermo (il che è il meno) sia nella matrice monodimensionale PAROLE S. La riga 17055 può essere eliminata senza particolari problemi.

A questo punto è possibile eseguire un confronto tra le parole della frase e quelle previste nella array di vocabolario. Cominciamo, così, a cercare l'oggetto: ovvio che questo può trovarsi, in base alla nostra convenzione solo come terza o quarta parola.

Perciò:

```

17500 REM Ricerca dell'oggetto
17510 FLAG10, FLAG11 = 0
17520 FOR H = 1 TO 10
17530 IF PAROLE $ (3) = OGGETTO
$ (H) THEN FLAG 10 = 1
NEXT H
17540 IF FLAG10 = 1 THEN PRINT
«La frase non contiene averbi»
FOR H = 1 TO 10
17560 IF PAROLE $ (3) = OGGETTO
$ (H) THEN FLAG11 = 1
NEXT H
17580 IF FLAG11 = 1 THEN PRINT
«È presente un avverbio»
17590 IF FLAG10 = 0 AND FLAG11
= 0 THEN PRINT «Attenzione,
manca il complemento oggetto,
o la frase non rispetta la
convenzione adottata».
```

La routine è scritta in maniera piuttosto elementare, proprio per consentire l'immediata visione del procedi-

```

16000 REM DOTAZIONE VOCABOLARIO
16010 DIM OGGETTO$(10)[10]
16020 DIM AGGETTIVO$(5)[10]
16030 DIM AVVERBIO$(4)[10]
16040 DATA CARNE, PASTA, PANE, FAGIOLI, PESCE, PATATE, FUNGHI,
PROSCIUTTO, FORMAGGIO,UOVA
16050 DATA ARROSTO, LESSO, GRATINATO, AFFUMICATO, VECCHIO
16060 DATA RARAMENTE, TALVOLTA, SPESSO, SEMPRE
16070 FOR H = 1 TO 10
16080 READ OGGETTO$(H)
16090 NEXT H
16100 FOR H = 1 TO 5
16110 READ AGGETTIVO$(H)
16120 NEXT H
16130 FOR H = 1 TO 4
16140 READ EVVERBIO$(H)
16150 NEXT H
```

Figura 2 - Routine di assegnazione del vocabolario di base.

A questo punto si può introdurre la nostra frase ed essa va spezzettata nelle sue parole componenti. Per comodità è però opportuno inserire, al fondo della frase uno spazio bianco.

```
IN $ = IN $ & " "
```

In questo modo tutte le parole in cui spezzetteremo la frase avranno un blank finale. Infine, per motivi di comodo, che vedremo in seguito, conviene introdurre una parola fantoccio finale con la sequenza.

```
IN $ = IN $ & " # "
```

Costruiamo adesso una routine (fig. 3) di spezzettamento della frase.

mento di confronto. Essa può essere così ridotta e perfezionata:

```

17500 REM
17510 FLAG10 = 3
17520 FOR H = 1 TO 10
17530 IF PAROLE $ (3) = OGGETTO
$ (H) THEN FLAG10 = 1 : H
= 10 : GOTO17550
17540 IF PAROLE $ (4) = OGGETTO
$ (M) THEN FLAG10 = 2 : H
= 10
17550 NEXT H
```

Fermiamoci un momento per fare una considerazione: possiamo applicare lo stesso ragionamento (e la stessa routine) all'aggettivo, che può esse-

```

17000 REM DIVIDE LA FRASE IN PAROLE E LE INCAMERA IN UNA ARRAY
17010 DIM PAROLE$(6)[10]
17020 B2=LEN(IN$) : FLAG1=6
17030 FOR H = 1 TO 6
17040 IF IN$ = "*" THEN FLAG1 = H : H=6 : GOTO 17060
17050 PAROLE$(H) = IN$[1,POS(IN$," ")] : IN$ = IN$[POS(IN$," ")+1, LEN(IN$)]
17060 PRINT PAROLE$(H)
17060 NEXT H
17070 IF FLAG1 = 6 AND PAROLE$(6) <> "*" THEN PRINT "FRASE TROPPO LUNGA"

```

Figura 3 - Spezzettamento della frase nelle sue parole.

re presente solo come 4° o 5° termine. Se assegniamo FLAG 20 alla routine aggettivo, eventualmente anche qui dopo aver fatto una verifica simile alla linea 17590, resta solo da verificare, con una linea simile, e dopo i debiti confronti logici con le routine precedenti, la presenza e la correttezza dell'avverbio che, sempre in base alla convenzione adottata, può essere presente solo come 2° termine. Se, a tal uopo, viene utilizzato il FLAG30, ci resterà solo da aggiungere una serie di confronti logici e test condizionali tra FLAG10, FLAG20 e FLAG30 (ed eventualmente un FLAGERRORE ancora una volta analogo alla linea 17530, inseribile ad ogni routine ed avente la precedenza, destinato ad evidenziare la non correttezza della frase inserita) per avere un'analisi, un «parsing» completo, della frase.

Abbiamo affrettato un poco questa parte per un motivo: ci interessava evidenziare la tecnica del parsing più che i risultati ottenuti, visto che, in effetti, la struttura rimane comunque piuttosto rigida. Vedremo tra un momento come è possibile snellire il tutto in maniera efficace ed elegante.

Ci preme, solo, di fare qualche precisazione riguardo al punto finale. Sebbene esso serva per individuare la fine della frase, nei nostri esempi rappresentava la lettera finale dell'ultima parola. Conviene farlo, per comodità, sparire; ciò può essere eseguito con una semplice routine, da inserire magari dopo l'input della frase completa e l'eliminazione di trailing e leading blank.

```

10 REM Routine eliminazione punto finale.
20 B2 = LEN (IN$)
30 IF IN$ [B2,B2] = "." THEN IN$ = IN$ [1,B2-1]

```

Ciò in quanto, in fase di confronto, qualsiasi parola, anche se battuta alla tastiera in modo esatto, non darebbe risposta positiva proprio per quel punto in più.

Come abbiamo visto, però, esiste un grosso problema: la frase va scritta in maniera piuttosto rigida, seguendo un prefissato ordine di disposizione delle parti logiche.

Ad esempio, se si scrive la nostra

frase come:

TALVOLTA IO MANGIO
VECCHIO PROSCIUTTO
ARROSTO

è come se dessimo una martellata in testa al calcolatore. Per poter affrontare adeguatamente il problema dovremo aumentare, in maniera esponenziale, le strutture condizionali, fino a che il programma scoppierebbe per elefantiasi. Cerchiamo di vedere le cose in maniera diversa e, perché no, più semplice. La prima ipotesi sarebbe quella di confrontare ciascuna parola con il vocabolario e determinarne la specie. Abbastanza logico, e peraltro valido, ma esiste una maniera un poco più raffinata di affrontare il problema. Vediamo come!

È possibile effettuare la scansione della frase, senza spezzettarla nelle sue parole, alla ricerca delle sue parti logiche. Come? Ad esempio, immaginiamo di voler cercare il complemento oggetto: dalla tabella dei DATA del dizionario vediamo che il più piccolo sostantivo della sequenza oggetto è formato da 4 lettere. A partire dalla prima lettera della frase individuiamo una stringa, comunque formata da 4 lettere, e confrontiamola con le prime 4 lettere delle stringhe presenti nel dizionario oggetti. Se il test dà risultato negativo spostiamoci alla seconda lettera, e così via.

Immaginiamo, tanto per provare, di cercare il complemento oggetto nella frase già detta: il minimo oggetto è formato da 4 lettere: cominciamo:

test n°	Stringa confrontata	Risultato test
1	IO M	0
2	IO MA	0
3	MAN	0
4	MANG	0
.....		
31	PROS	1

A questo punto, se lo si desidera, e sarebbe preferibile, può essere eseguito un confronto più approfondito della stringa completa. Ovvio, no?

C'è però, e poteva mancare, qualche problema da risolvere: UOVA (sostantivo, complemento oggetto), NUOVA (aggettivo), o GROSSEUOVA (errore di battitura) danno lo stesso risultato

positivo. La soluzione è abbastanza semplice ed ovvia. Le parole del vocabolario vanno inserite con un blank iniziale, e la verifica va, ovviamente, fatta su n + 1 caratteri. Se si aggiunge un test di verifica che evidenzia che, nella frase in input, la parola testata non faccia parte, in coda, di una parola più lunga (es. NUOVA[MENTE]), cosa che può essere fatta agevolmente solo controllando la presenza di un blank finale, il gioco è fatto. L'unica cosa a cui fare attenzione è l'inserimento della riga

```
...10 IN$ = " " & IN$
```

onde consentire un corretto test anche della prima parola.

Nel modo che abbiamo esposto, è possibile inserire la frase senza troppe precauzioni e senza badare molto al costruito sintattico. Non solo, ma sarà possibile scrivere frasi come:

A ME PIACE MANGIARE CARNE
E PESCE ARROSTO, TALVOLTA
SPESSO, TALVOLTA
RARAMENTE

Inserendo una semplice routine che, preliminarmente, separi le virgole dalle parole, la frase è perfettamente comprensibile al calcolatore, che si farà ben poco cruccio del ciarpame presente, ma analizzerà e terrà conto solo di quello che è esistente nel vocabolario.

Ci pare infine ovvio come sia possibile distinguere il tipo di parola trovato applicandovi un marker iniziale.

Tanto per intenderci, le parole trovate potranno essere conservate in una array, all'uopo predisposta, in cui ognuna sarà preceduta da una o più lettere indicanti il tipo (ES O-PESCE, per indicare che è un oggetto, oppure A-LESSO, per indicare un aggettivo). A questo punto siamo solo legati ai dizionari presenti ed alla loro ampiezza: si può crearne uno contenente i soggetti, uno i verbi (anche se qui problemi semantici di declinazione, difettività o irregolarità porrebbero problemi un poco più complicati), un altro i complementi, ecc. così, ad una tastiera di un computer destinato alla gestione di un ristorante di un albergo sarà possibile battere:

— Il signor ROSSI desidera a cena pollo lesso e patate fritte
e, anche qui superando qualche piccolo problema di desinenze, farsi capire e magari vedersi rispondere
— «PATATE non disponibili. Scegliere altro contorno»
e magari battendo
— «PORTAMI CARMEN RUSSO»
avere come risposta
— «Se ne sconsiglia l'abbinamento con BUCATINI ALL'AMATRICIANA - TROPPO PESANTE!».

MC

IL DATABASE DI FIDUCIA!

Volete gestire le informazioni in modo facile veloce ed accurato? La sola risposta è DELTA 4, il database in grado di soddisfare le vostre esigenze in brevissimo tempo!

DELTA 4 permette di inserire, selezionare, ordinare le informazioni, eseguire calcoli dai più semplici ai più complessi, stampare prospetti ed etichette e persino inviare lettere personalizzate! DELTA 4 può trasferire i dati su/da altri programmi quali Multiplan, Lotus 1-2-3, Wordstar . . . ed anche elaboratori centrali!

DELTA 4 è facilissimo da usare perchè è in ITALIANO e non richiede alcuna conoscenza di linguaggi di programmazione. Seguendo il menu principale potrete scrivere il vostro programma senza commettere errori! Ideale per l'uomo d'affari, DELTA 4 è usato ogni giorno anche da esperti programmatori in piccole o grandi aziende in tutto il mondo!

Il menu principale di DELTA 4 consente inoltre di creare il vostro menu di opzioni automatiche, in grado di caricare altri programmi e/o altri menu personalizzati.

DELTA 4 È UN GENERATORE DI APPLICAZIONI ED È IDEALE PER:

- Gestione magazzino
- Gestione ordini
- Agenzie viaggio/pubblicità
- Farmacie
- Biblioteche/videoteche
- Gestioni beni immobiliari
- Ospedali/ambulatori
- Gestione clienti
- Gestione di portafoglio
- Banche
- Alberghi, Ristoranti
- . . . e mille altre!

**NON PERDETE ALTRO TEMPO! RIVOLGETEVI
SUBITO AL VOSTRO RIVENDITORE.**

Disponibile su:

IBM PC/XT/AT
OLIVETTI M24/M21
HP150
DEC RAINBOW 100/100 +
Victor/Apricot

Distribuito da:

IBM Italia S.p.A.
Olivetti/Syntax S.p.A.
Hewlett Packard Italiana S.p.A.
Digital Italiana S.p.A.
Harden Italia S.p.A.

. . . attraverso la loro rete di concessionari e negozi.



**Compssoft
Italia s.r.l.**

Viale Campania 4, 20133, Milano
Telefoni: 02 7380092; 02 7388325;
02 7381836