

# When the Lab of CompLing was started at the University of Venice - Preface to the proceedings of the first workshop held in 1982

Rodolfo Delmonte<sup>1</sup>

<sup>1</sup>*Ca Foscari University of Venice*

## Abstract

The chapters contained in the book - now out of print - and in its second edition published in 1988 with the title "Studi di Linguistica Computazionale"/Studies of Computational Linguistics, are some of the presentations held at the "Seminario Introduttivo alla Linguistica Computazionale"/Introductory Workshop in Computational Linguistics, on May 13/14/1982, at the University of Venice.

## Keywords

what CL really is, Pisa contribution, Zampolli opinion, Venice contribution

## Prefazione al libro

I contributi raccolti in questo volume sono alcuni degli interventi tenuti al Seminario Introduttivo alla Linguistica Computazionale, il 13-14 maggio 1982, all'Università di Venezia. Lo scopo dell'incontro era principalmente quello di presentare una panoramica dettagliata degli sviluppi di una disciplina, la linguistica computazionale, che in Italia ha per ora poco seguito. I motivi di questo stato di cose sono svariati, di tipo economico storico e sociale: tradizionalmente le Facoltà umanistiche infatti, non si servono di tecnologia per svolgere le proprie ricerche, a differenza di quelle scientifiche. Il letterato, il filologo, il glottologo e il linguista in Italia solo sporadicamente si è rivolta all'elaboratore per avere un ausilio nei propri studi. All'estero è prassi comune che gli studiosi in campo umanistico, se non dispongono direttamente di un centro ricerche di linguistica computazionale, si siano rivolti all'elaboratore per verificare o convalidare ipotesi teoriche. Non vi è alcun dubbio però che gli elaboratori costano, e il livello di investimento nella ricerca in Italia è ben noto a tutti quelli che vi operano come basso. Ugualmente nota poi in Italia è la separazione che tuttora esiste tra ricerca umanistica attuta con carta, penna e intuito dallo scolaro, e ricerca scientifica che deve giocoforza affidarsi alla tecnologia. In più, storicamente, si aggiunge l'anatema crociano contro tutti quelli che per essere creativi non si affidino all'intuito e intelligenza individuale, ma magari vogliono fare lavoro d'equipe o "sacrilegio", utilizzare macchine. I risultati sono ovviamente quelli che dicevamo: benché l'interesse per la LC sia cresciuto negli ultimi sei o sette anni, ciò è dovuto quasi unicamente all'opera di organizzazione

---

*CLiC-it 2023: 9th Italian Conference on Computational Linguistics, Nov 30 – Dec 02, 2023, Venice, Italy*


✉ [delmont@unive.it](mailto:delmont@unive.it) (R. Delmonte)

🌐 <http://rondelmo.it/> (R. Delmonte)

🆔 0000-0003-0282-7661 (R. Delmonte)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

e ramificazione compiuta dall'attuale Istituto di Linguistica Computazionale di Pisa, diretto dal prof. Antonio Zampolli.

Come ha anche lucidamente chiarito Zampolli in un suo intervento[1] e nella relazione presentata a Venezia, è bene distinguere perlomeno tra linguistica quantitativa e linguistica computazionale: nel primo caso infatti l'elaboratore viene utilizzato principalmente allo scopo di condurre analisi di testi di tipo statistico, probabilistico e quantitativo in genere. L'elaboratore funge in questo caso da ausilio al linguista, il quale è primariamente interessato agli aspetti stilistici, filologici, glottologici e letterari del testo/i in esame: l'elaboratore fornirà allora spogli elettronici dalla cui consultazione ragionata il linguista potrà rilevare gli elementi di suo interesse. Ovviamente, una volta trasferito uno o più testi su un supporto appropriato per la macchina, schede perforate o nastro magnetico, l'analisi potrà essere ripetuta variando eventualmente di volta in volta gli elementi che si vuole che l'elaboratore estragga per il linguista. Analisi di questo genere, anche se compiute su corpora estesi, ad esempio di 100mila occorrenze, non richiedono all'elaboratore più di 50 secondi di tempo macchina e un tempo variabile tra i 15 e i 30 minuti per stampare i risultati.

La LC invece ha come scopo l'analisi del linguaggio e non solamente della lingua, e l'elaboratore non è più semplicemente un ausilio, ma lo strumento di simulazione dei processi linguistici sottostanti a un qualche modello della produzione/comprendimento della lingua, implementati dal linguista, che vorrà verificarne la bontà di funzionamento. Il programma utilizzato in questo caso non servirà a produrre spogli di corpora o testi, ma rappresenterà una grammatica, tradotta in un linguaggio di programmazione. Il linguista sarà allora interessato a studiare gli effetti prodotti da modificazione, introduzioni o cancellazione di regole della grammatica sulla interpretazione e descrizione del campione di lingua utilizzato.

Più in generale, il linguista computazionale sarà interessato non soltanto al raggiungimento di obiettivi esplicativi e cioè di descrizione del linguaggio mediante regole e rappresentazioni strutturali, ma vorrà anche sviluppare strumenti adeguati alla generazione o produzione e alla comprensione e percezione del linguaggio. Se quindi le ricerche quantitative operano in ambito matematico e statistico, utilizzando un approccio tassonomico e descrittivo ai fenomeni linguistici studiati, quelle computazionali sono interessate alla elaborazione di modelli della competenza e della realizzazione linguistica, in altre parole alla grammatica e al meccanismo che la realizza o processore.

Come spiega Zampolli[1], il sogno di giungere velocemente alla comprensione dei meccanismi della lingua attraverso la traduzione meccanica svanì lentamente negli anni '60. L'attenzione degli operatori in quel tempo si distolse dai problemi di implementazione su elaboratore di algoritmi intesi a risolvere questioni linguistiche, per rivolgersi più direttamente al funzionamento del linguaggio: da problemi legati a tecniche di programmazione e linguaggi, tecniche di immagazzinamento dati e altri aspetti del software e hardware intesi ad influire direttamente sulla lingua in esame, per dirigersi allora al problema costituito dalla struttura del linguaggio e dalle sue regole sottostanti.

Benché l'alto livello di formalizzazione e di esplicitazione nella descrizione strutturale e delle regole proposta da Chomsky possa indurre il linguista a considerare la grammatica generativa trasformazionale(GGT) come un esempio di algoritmo linguistico per elaboratore: ma in realtà non è così, e lo ha ribadito lo stesso Chomsky[2]. La GGT è sì esplicita nei suoi formalismi, ma i suoi obiettivi descrittivo/esplicativi non si adeguano se non in parte a quelli che ci si

pone in ambito LC. Nella GGT manca infatti una qualsiasi preoccupazione per esplorare e studiare i meccanismi relativi alle operazioni di produzione e comprensione del linguaggio che Chomsky relega alla cosiddetta esecuzione o realizzazione linguistica, che in quanto tale non può costituire un campo di ricerca scientificamente valido per un linguista. E se il modello di competenza proposto da Chomsky costituisce fondamentalmente un modello psicologico, esso è però anche intenzionalmente il più astratto e il più distante possibile dalla realtà della realizzazione linguistica. Le regole contenute nella grammatica infatti secondo i principi della GGT, non potranno mai essere desunte o indotte da corpora linguistici attraverso procedimenti di scoperta per quanto ampi essi possano essere, saranno pur sempre deficitari rispetto alla quantità di materiale linguistico che produrrà/comprenderà nella sua vita un parlante qualsiasi della lingua.

Applicazioni computazionali della GGT come quella della Joyce Friedman[?] o altre, discusse in particolare dalla Prodanof, in realtà sono ispirate solo in parte alle posizioni teoriche della GGT. Infatti, la teoria e il processore non necessariamente coincidono, ed è soltanto il secondo che servirà da verifica della prima, che essenzialmente è e resterebbe solamente una ipotesi di funzionamento del linguaggio. In particolare poi, le ultime teorie generative hanno sostituito alla centralità della sintassi con il suo componente trasformazionale, il lessico e le categorie funzionali come primitive. I lavori di Bresnan[?], Kaplan e Bresnan[3] e Gazdar[4] hanno relegato il ruolo della sintassi alla sottocategorizzazione e al funzionamento di una grammatica a struttura sintagmatica che proietta strutture sintattiche direttamente dall'analisi superficiale. In termini di analizzatore sintattico o "parser", non sarà quindi più necessario procedere a ritroso con le trasformazioni alla rovescia per recuperare una struttura profonda difficilmente individuabile. Un parser ATN, o quello deterministico del Parsifal saranno allora sufficienti con sole regole di tipo context-free a descrivere la struttura sintattica sottostante - ma di questo tratterà estesamente il saggio della Prodanof.

Tradurre e comprendere discorsi e testi sono invece attività che richiedono ad un processore abilità alquanto diverse da quelle necessarie per la verifica di una teoria e del modello che essa rappresenta. Infatti il linguista computazionale dovrà simulare il comportamento linguistico di un parlante/ascoltatore in situazioni reali. In questo caso non sarà sufficiente l'informazione lessicale, morfologica, sintattica e semantica ma si dovrà utilizzare anche quella pragmatica; e le regole non potranno essere solo quelle contestuali, o context-sensitive, ma saranno del testo o del discorso, cotestuali o transfrastiche, in modo da catturare i processi di inferenza che dall'enunciato risalgono alla conoscenza o enciclopedia del parlante/ascoltatore. Solo così la codifica-decodifica del messaggio o produzione-comprensione dell'enunciato potrà realizzarsi efficacemente. Le ricerche in questo campo vanno da quelle di Petoefi[5] e Van Dijk[6] a quelle documentate in Conte[7], Parisi[8] e Castelfranchi e Parisi[9], nonché all'ambito di ricerca definito con il termine Intelligenza Artificiale, a cui accenna la relazione di Ferrari, all'interno di lavori orientati a compiere analisi automatiche del contenuto semantico del linguaggio.

Ed è proprio di questo campo più vicino alla realizzazione linguistica che si interessa l'ultimo saggio, che non vuole né può essere una panoramica dei lavori svolti in questa area di ricerca, in quanto le soluzioni adottate per l'inglese mal si adattano all'italiano. Il saggio propone un modello e la sua simulazione in un processore basato su teorie fonetico-fonologiche e prosodiche, o di quella parte più standardizzata di variabili implicate nella fase di realizzazione linguistica rappresentata dalla produzione del parlato. In pratica, il processore simula le operazioni di

codifica e decodifica compiute da un parlante nel leggere un testo ad alta voce. In questo senso esso è uno strumento adeguato alla produzione di voce sintetica attraverso macchine comandate da elaboratori. I primi lavori in questo ambito sono di provenienza inglese e cioè di Holmes e Mattingly[10], e Mattingly[11] che propongono un approccio segmentale di tipo fonemico; i lavori del gruppo di Padova, composto da Francini, Debiasi e Spinabelli [12], propongono una soluzione al problema delle unità minime con i difoni, che sono di numero superiore ai fonemi ma risolvono interamente il problema delle transizioni tra due suoni contigui. Sempre nell'ambito della sintesi della voce, si registrano poi i lavori americani, in particolare quelli di Allen[13] e di Umeda[14], nonché il sistema di Klatt[15], basato di nuovo su difoni. Tutti questi sistemi di sintesi contengono un processore della lingua basato su teorie fonetico-fonologiche: ad esempio Allen utilizza le teorie chomskiane per il suo modello, e per prevedere la posizione dell'accento ha costruito un dizionario di 12mila morfi che assieme ad un algoritmo morfologico gli permettono di recuperare la struttura sillabica della parola in esame, quindi il suo riconoscimento e l'assegnazione dell'accento di parola per regole. Anche il nostro modello, come vedremo, prevede la localizzazione dell'accento di parola, utilizzando un lessico esiguo ma solo per le eccezioni alle regole.

## RINGRAZIAMENTI

Le persone da ringraziare in questi casi sono ovviamente tantissime: innanzitutto chi ha permesso materialmente che avesse luogo l'incontro e cioè il prof. Volpato, direttore del CEDOSTA, il prof. Castellani, direttore del Centro di Calcolo e il prof. Csillaghy, direttore del Centro Linguistico dell'Università di Venezia. Quindi chi mi ha spronato ad organizzare l'incontro e cioè il prof. Zampolli, direttore dell'Istituto di LC di Pisa, e tutti gli amici, Riccardo Zipoli, Alberto Tomasin, Irina Prodanof, Giacomo Ferrari, Luciano Canepari, e gli altri che hanno seguito gli interventi. Per ultimo chi più di ogni altro ha contribuito di persona, Giorgio Vercellin. Il libro non sarebbe potuto uscire in questa forma senza il contributo del Centro Linguistico e l'aiuto dell'amico Marco Vio.

## References

- [1] A. Zampolli, Trattamento automatico di dati linguistici e linguistica quantitativa, in: *Linguaggi e Formalizzazione - SLI*, Bulzoni, Roma, 1979, pp. 349—370.
- [2] N. Chomsky, *Current Issues in Linguistic Theory*, Prentice-Hall, Englewood Cliffs N.J., 1964, pp. 50—118.
- [3] R. K. J. Bresnan, *Lexical Functional Grammar: a Formal Syntax for Grammatical Representation*, ????
- [4] G. Gazdar, *Phrase Structure Grammar*, ????
- [5] J. S. Petoefi, Text representation and lexicon as semantic network, in: *Linguaggi e Formalizzazione - SLI*, Bulzoni, Roma, 1979, pp. 573—589.
- [6] T. A. V. Dijk, *Text and Context*, Longman, London, 1977.
- [7] M. E. Conte, *La linguistica testuale*, Feltrinelli, Milano, 1977.

**TOMASIN FERRARI PRODANOF DELMONTE**

# **LINGUISTICA COMPUTAZIONALE**

**a cura di rodolfo delmonte**

**Figure 1:** Copyright © 1983 by C.L.E.S.P. s.r.l. Via del Santo, 57/7 – 35123 Padova – Italy All rights reserved

- [8] D. Parisi, Studi per un modello del linguaggio, volume 89, Quaderni della Ricerca Scientifica - CNR, Roma, 1975.
- [9] C. C. . D. Parisi, Linguaggio conoscenza e scopi, Il Mulino, Bologna, 1980.
- [10] J. N. H. . I. G. M. . J. N. Shearman, Speech synthesis by rule, Language and Speech 7 (1964) 127–143.
- [11] I. G. Mattingly, Synthesis by rule of prosodic features, Language and Speech 9 (1966) 1–23.
- [12] G. I. F. . G. B. D. . R. D. Spinabelli, Study of a system of minimal speech-reproducing units for italian speech, JASA 43 (1968) 1282–1286.
- [13] J. Allen, Synthesis of speech from unrestricted text, in: Proceedings of the IEEE - "Man

Machine Communication by Voice", volume 64, 1976, pp. 433—442. doi:10.1109/PROC.1976.10152.

- [14] N. Umeda, Consonant duration in american english, JASA 61 (1977) 846–858.
- [15] D. H. Klatt, Structure of a phonological component for a synthesis-by-rule program, ASSPT - IEEE Transactions Acoustic Speech Signal Process 24 (1976) 391–398.