

Il prompt prima del prompt

di Stefano Penge



In fondo “scrivere prompt” è solo una particolare forma di programmazione. Si dice ad un computer (no, ad un software, anzi all’interfaccia di un software) quello che si vuole che faccia.

Se questo sembra lontano dalla programmazione tradizionale forse è solo perché non la si conosce a fondo e si pensa invariabilmente a una serie di stupidi comandi dati ad uno stupido robot che li esegue. Si pensa che programmare significhi specificare esattamente il risultato e il modo di raggiungerlo passo passo (l’algoritmo).

Che i prompt possano usare parole italiane, anziché astrusi comandi in un simil-inglese separati da segni di interpunzione usati a casaccio, non è nemmeno tanto strano: la storia dei linguaggi di programmazione ha già visto tanti tentativi di staccarsi dalle lingue specialistiche per avvicinare alla programmazione anche le persone normali: è il caso del COBOL, inventato negli anni ‘60 per far programmare anche i contabili e non solo gli ingegneri. L’idea poi che i linguaggi di programmazione debbano per forza assomigliare all’inglese è falsa; ci sono linguaggi basati sull’arabo, sul cinese e persino sul latino [https://www.codeshow.it/Linguaggi/Linguaggi_nazionali].

[VAI ALLA RUBRICA SULLA INTELLIGENZA ARTIFICIALE](#)

Ma nemmeno è detto che programmare sia per forza “dare ordini”. Ad esempio, nella programmazione detta “orientata agli oggetti” si definiscono le conoscenze e le abilità di un tipo di oggetti. Più oggetti insieme, collegati gerarchicamente oppure messi in relazione tramite la possibilità di scambiarsi informazioni, costituiscono un modello. Una volta costruito, il modello viene eseguito e se la vede da solo, senza interazioni con gli umani. E’ come scrivere il copione e poi mettere in scena una rappresentazione teatrale.

O ancora: nella programmazione logica non si impartiscono comandi ma si specificano tre cose: dei fatti, delle regole, un obiettivo (o meglio un teorema da dimostrare). Dati e regole costituiscono un modello. L’interprete cerca di dimostrare il teorema a partire dai fatti che gli sono noti usando le regole che ha appreso. Se ce la fa, si limita a dire “Vero”. Il terreno su cui si muove, le vie e gli incroci sono disegnati dal programmatore; ma la strada che segue non viene tracciata da nessuno.

In tutti questi casi, non vengono specificati tutti i passi da compiere nell’ordine in cui vanno compiuti, ma si fornisce una rappresentazione ad alto livello del sistema (nella programmazione ad oggetti) o degli obiettivi (nella programmazione logica). Resta il fatto che le regole secondo le quali il programma viene eseguito dal computer sono note ed eventualmente ricostruibili. Se proprio si vuole, a posteriori si può chiedere all’interprete Prolog di mostrare tutti i tentativi che ha fatto, quelli falliti e quelli che hanno avuto successo. Anche le regole “costruite” dal programma stesso durante la sua esecuzione – che è una delle caratteristiche che rende Prolog così flessibile e capace di risolvere problemi complessi – sono spiegabili secondo la grammatica che gli è stata fornita.

Software che interagiscono con le persone in linguaggio naturale sono stati immaginati e realizzati dai programmatori

già da molti decenni. A partire da **Joseph Weizenbaum**, che nel 1966 per dimostrare quanto le persone sono pronte ad attribuire coscienza a qualsiasi cosa, dalle nuvole ai calcolatori, scrive polemicamente “Eliza”, un programma che simula una sessione con uno psicoterapeuta della scuola rogersiana. Eliza non capisce nulla, ma si limita a reagire a certe parole chiave con risposte preconfezionate. Anche così, c’è chi ci passa pomeriggi interi per provare a risolvere i propri problemi.

Poi si può ricordare **Will Crowther**, un programmatore appassionato di speleologia e giocatore di Dungeons&Dragons che nel 1975 scrive “Colossal Cave Adventure”, il primo gioco testuale in cui il giocatore per esplorare una caverna infinita deve rivolgersi al suo alter-ego virtuale e *scrivere* piccole frasi come “guarda in alto, vai a destra, prendi il martello”. [https://www.codeshow.it/Codici/Colossal_Cave]

Lo stesso **Prolog**, il linguaggio principe per la programmazione logica, era stato inventato da **Alain Comerauer** negli anni ‘70 all’università di Marsiglia per scrivere programmi che permettessero di parlare al computer in francese. [<https://www.codeshow.it/Linguaggi/Prolog>]

Insomma la capacità di analizzare una richiesta scritta da una persona e produrre una risposta comprensibile è una competenza sofisticata, ma non nuova. Come non è nuova la capacità di analizzare un testo per individuarne le parti e “marcarle” in base all’argomento. Lo fanno già tutti i software dei social network quando analizzano quello che scriviamo per costruire un profilo dei nostri interessi e delle nostre opinioni, facendo quella che si chiama “sentiment analysis” (in cui “sentiment” è quell’opinione di pancia, non razionalizzata, che ci rende insopportabile la Ferragni e ci rende simpatico Gigi Proietti).

Nel caso dei modelli GPT non si sta interagendo direttamente col modello linguistico (che è un oggetto enorme con miliardi

di parametri), ma appunto con un software specializzato che è in grado di tradurre quello che scrive la persona in una serie di istruzioni che interrogano il modello. Questo software è abbastanza sofisticato da sapere dire “grazie, scusa, ho sbagliato” e individuare alcune parole importanti in un mucchio di altre.

Ma non è il solo. Lo fanno anche i “prompt” dei motori di ricerca da un bel po’ di tempo; si potrebbe scrivere:

“per favore, Google adorato, potresti dirmi se qualcuno da qualche parte ha scritto qualcosa sul terzo canto dell’inferno? Grazie”

ma in pratica è sufficiente scrivere:

“terzo canto inferno”

che sono le uniche parole importanti. Il risultato è lo stesso; se non ci credete, provate.

The screenshot shows a Google search interface with a dark theme. The search bar contains the text "per favore, Google adorato, potresti dirmi se qualcuno da qualche parte ha scritto qualcosa sul terzo canto dell'inferno? Grazie". Below the search bar, there are navigation tabs for "Siti di luoghi", "Immagini", "Video", "Libri", "Notizie", "Voli", and "Finanza". The search results show "Circa 9 risultati (0,37 secondi)". The first result is from "Studenti.it" with the URL "https://www.studenti.it > canto-3-inferno-dante-parafrasi...". The title of the result is "Canto III Inferno di Dante: testo, parafrasi e analisi". The description reads: "Testo e parafrasi del canto 3 dell'Inferno di Dante Alighieri con spiegazione, commento e figure retoriche. Gli ignavi e la legge del contrappasso. Mancanti: Google adorato, qualcosa". Below the search results, there is a section titled "Le persone hanno chiesto anche" with four expandable questions: "COSA DICE IL 3 canto dell'Inferno?", "Quale figura retorica è presente nei primi tre versi del Canto III?", "Chi si trova nel terzo canto?", and "Che cosa c'è scritto sulla porta dell'Inferno?". A "Feedback" link is visible at the bottom right of the page.

In realtà l'interfaccia del motore di ricerca utilizza *tutte* le parole per eliminare possibili ambiguità; ma è in grado di filtrare il rumore di fondo costituito da articoli, congiunzioni, verbi predicativi e parole troppo comuni per concentrarsi sulle altre.

La particolarità di questa maniera di programmare computer è che lo si fa *senza specificare esattamente come*. Non è necessario specificare quello che si vuole ottenere, se non in maniera piuttosto vaga; ci pensa il software a estrarre le informazioni rilevanti e tradurle (abbiamo detto che i computer lo sanno fare bene) in istruzioni eseguibili. D'altra parte il concetto di ricerca si è trasformato parecchio: se prima era un bibliotecario a cui si forniscono le parole che devono comparire nell'oggetto cercato, oggi è un maggiordomo che propone soluzioni prima ancora che la domanda venga formulata e che il bisogno venga espresso. La trasparenza e l'anticipazione dei bisogni sono, a mio avviso, due degli obiettivi perseguiti con più costanza nel progresso dei servizi digitali. Non necessariamente a vantaggio nostro, come ho provato a raccontare qui [<https://www.stefanopenge.it/wp/trasparenze/>].

Quello che è davvero nuovo è la capacità di generare testi complessi sulla base di un modello anche stilistico oltre che di contenuto. Ed è abbastanza nuova la maniera in cui questa performance avviene: non applicando regole ad un insieme di dati, ma costruendo un enorme modello linguistico in cui le parole siano collegate ad altre parole sia per vicinanza che

per area semantica. Questo super-modello permette di predire quale parola è più probabile che segua una serie di parole precedenti. Un po' quello che fa il T-9 del nostro telefono, che ad ogni parola che scriviamo ci propone la parola successiva più probabile sulla base delle sequenze più frequenti, ma molto meglio.

Non c'è una vera comprensione del testo, se per comprensione intendiamo qualcosa di vicino a quello che noi immaginiamo di fare quando leggiamo un testo: una specie di ricerca delle parole nel dizionario dei significati, seguita da una somma di tutti i significati trovati. Ma c'è chi dice che in fondo anche noi, quando ascoltiamo e parliamo, utilizziamo strategie superficiali che hanno a che fare più con la parentela tra le parole che con la proiezione sul piano lessicale e grammaticale dei significati di una serie di concetti. Per esempio, quando ascoltiamo qualcuno che parla una lingua nota ma non troppo tendiamo a cercare di cogliere frammenti e ricostruire il senso generale a partire da quelli e dalle intersezioni possibili tra i campi semantici relativi. Ad esempio, se intercettiamo "cane" e "grilletto" la conversazione potrebbe riguardare delle armi da fuoco; ma se insieme c'è "gatto" allora probabilmente si tratta di Pinocchio. Capito, o meglio deciso, questo contesto, tutte le altre parole verranno interpretate nella direzione selezionata. Il cane sarà Melampo, il legno quello del ciocco usato da Geppetto eccetera.

Forse addirittura i significati non sono niente altro che queste relazioni tra parole che rendono più probabili certe costellazioni di parole rispetto ad altre nelle conversazioni. Se fosse così, ChatGPT non farebbe niente di diverso da quello che facciamo noi perché è così che funziona il linguaggio in pratica.

Quindi: o noi *non* siamo più intelligenti dei grandi modelli linguistici, oppure questi modelli non sono affatto intelligenti.

Ma arriviamo all'uso didattico del prompt, che era l'obiettivo dichiarato di queste riflessioni.