UNA CONTINUA PRESUNZIONE

di Marco Malvaldi

Potrò ritenere intelligente un computer il giorno in cui capirà quando scherzo.

Così diceva, una ventina di anni fa, Luciano de Crescenzo in uno dei suoi libri divulgativi di filosofia. Un computer, sosteneva de Crescenzo (che di computer un pochetto se ne intendeva, avendo fatto l'ingegnere informatico per una mezza vita), processa il linguaggio come se fosse una stringa di operatori e simboli logici, per cui se io dicessi 'il mio amico Claudio è una sagoma, ma ogni tanto lo ammazzerei' un elaboratore elettronico, alla domanda 'De Crescenzo Luciano desidera uccidere il suo amico Claudio?' risponderebbe 'Sì'.

Questa frase, all'epoca, mi fece una profonda impressione. Come potrebbe fare, un computer, a capire se sto scherzando?

Allora, mi ricordo che pensai che un computer avrebbe dovuto conoscermi direttamente. Avrebbe, in pratica, dovuto riconoscere una incoerenza tra quello che dicevo e la mia persona nel suo insieme. Per esempio, una frase come 'I calabresi sono terroni arretrati' è probabilmente scherzosa se la dice un calabrese, ma se viene pronunciata da un elettore di lungo corso della Lega Nord non mi sentirei di giurare che c'è dell'umorismo sotto; o perlomeno, non volontario.

Oggi come oggi, a livello computazionale, è possibile riconoscere una frase umoristica. Per capire come, è necessario guardare nel dettaglio come è fatto il linguaggio naturale e quali sono le sue differenze con il linguaggio matematico.

In primo luogo, il linguaggio naturale è ridondante; ovvero, in una frase è solitamente presente un gran numero di lettere di cui si potrebbe fare a meno, e riuscire a comprendere correttamente la frase. Già Claude Shannon, negli anni Cinquanta, aveva calcolato in modo abbastanza semiempirico (facendo leggere alla moglie articoli di giornale con delle lettere cancellate col pennarello, in numero crescente) che tale ridondanza si aggirava intorno al 50%. Tale ridondanza viene da due fatti distinti.

Il primo fatto è che le lettere non sono indipendenti le une dalle altre (dopo una 'q' viene sempre una 'u', e raramente in italiano si possono avere sette o otto consonanti di fila, come nel cecoslovacco 'zmrzlna').

Una possibile misura di questa interdipendenza è la cosiddetta mutua informazione (MI). Dati due testi scritti X, Y che prendono successivamente i valori x e y, ognuno con una probabilità finita, si ha:

$$MI(X,Y)=\sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}$$

La mutua informazione, come il coefficiente di correlazione, misura la relazione tra due serie di eventi X,Y per stabilire se siano indipendenti o meno. Se i due eventi sono indipendenti, e solo in quel caso, $p(x, y) = p(x)\,p(y)$ per qualsiasi x, y, e la mutua informazione tra le due serie è nulla. A differenza della correlazione, però, la mutua informazione è in grado di gestire correttamente relazioni non lineari o altamente simmetriche, dando risultati non nulli là dove invece il semplice coefficiente di correlazione darebbe un bello zero. Ad esempio, se riportassimo in grafico un quadrifoglio ideale, con due petali giacenti sulla retta $y = x$ e i due petali opposti, per simmetria, allineati lungo $y = -x + c$, il coefficiente di correlazione tra x e y sarebbe nullo: ogni singolo punto sui primi due petali verrebbe annullato da un punto corrispettivo appartenente ai secondi due. La mutua informazione, invece, riconoscerebbe un risultato differente da quello dovuto al mero caso.

Proprio grazie alle proprietà della mutua informazione, recentemente Max Tegmark ha dimostrato che il nostro linguaggio non può essere generato da un processo markoviano, anche se fosse organizzato gerarchicamente su più livelli.

Un processo markoviano è un processo in cui il passato determina senza ambiguità il futuro. Più precisamente, in un processo markoviano, a partire dagli elementi precedenti, si possono assegnare delle probabilità finite agli elementi successivi:

$$p(x_{i+1}) = p([x_i, x_{i-1}, \ldots, x_{i-n}])$$

In altri termini, se il linguaggio fosse un processo markoviano di ordine n, la probabilità della lettera i+1 sarebbe data senza equivoci dalla probabilità della stringa costituita dalle precedenti n lettere. Quello che Tegmark ha mostrato è che in un qualsiasi linguaggio formato da un processo markoviano, la mutua informazione ha delle proprietà di decadimento esponenziali; in altri termini, calcolando la MI tra una lettera X e la lettera Y da lei separata da altre n lettere, e facendo crescere la separazione n, la MI decade come Ae^{-n}. Nei linguaggi dell'essere umano (sia il linguaggio naturale sia la musica), tale misura decade invece secondo una legge di potenza. Il linguaggio naturale non è markoviano, come mostreremo tra breve, e proprio questa può essere la chiave per capire come farlo leggere ed interpretare a un calcolatore.

Abbiamo visto che il linguaggio è ridondante, e che circa il 50% delle lettere sia superfluo ai fini della comprensione. Questo, chiaramente, non significa che eliminando una lettera su due il testo sia sempre comprensibile; se per esempio, scrivo:

l cmerr rb

non è facilissimo capire la frase, ma se invece scrivo:

l Jvnts rb

che è quasi equivalente, come numero di lettere totali e numero di lettere cancellate, a quella precedente, credo che non pochi ci possano riconoscere quello che intendo dire. (Per i pochi che non masticano di calcio, la frase è 'la Juventus ruba'. Si fa per scherzare, ragazzi.)

Perché la frase precedente è più facile da ricostruire? Questo porta a un secondo aspetto del linguaggio naturale, che è quello dell'interdipendenza. Ovvero, io sono in grado di associare a una possibile parola tutto un ventaglio di parole che sono solite accompagnarla: siccome la parola 'Juventus', nel linguaggio comune, è spesso associata alla parola 'ruba' (vedi sopra), il riconoscimento è più veloce e più sicuro.

Mediamente, lo sappiamo tutti, il lettore è in grado di riconoscere una parola incompleta; un po' come se eseguisse una proiezione nello spazio di Hilbert di tutte le parole che conosce, assegnando una probabilità in funzione del grado di sovrapposizione.

Chiaramente, più è vasta la cultura di una persona, più questa operazione diventa potente e, al tempo stesso, ambigua. La stringa 'crct' può venire interpretata in maniera automatica come 'crociato' se siamo esperti di storia medievale o se stiamo studiando la *chanson de Roland*, ma se invece siamo veterinari è più facile che la leggiamo come 'criceto'. Attenzione, però: anche se siamo veterinari potremmo leggere 'crct' come 'crociato', nel caso in cui la parola precedente fosse 'legamento'.

In pratica, ogni parola che leggiamo crea in noi delle aspettative; man mano che leggiamo, il nostro cervello fa delle inferenze riguardo a quello che è lecito aspettarsi dopo.

Eccoci qua: la chiave è proprio questa. È proprio confrontando tali inferenze, tali previsioni, con lo sviluppo effettivo della frase che un computer potrebbe riconoscere una frase umoristica. Il divertimento, etimologicamente, nasce proprio da questo: *divertire* nel senso latino del termine, cioè cambiare direzione.

Il linguaggio naturale, abbiamo visto sopra, non può essere generato da un processo markoviano; ovvero, la conoscenza di ciò che viene prima non è in grado di assegnare in modo corretto una probabilità finita a ciò che viene dopo. Mostreremo questa cosa con un esempio. Supponiamo che leggiate la seguente frase:

I ragazzi si divertivano tantissimo a tuffarsi in piscina direttamente dall'albero, così abbiamo deciso di metterci un ...

Metterci cosa? Così, a intuito, credo che al lettore verrebbe naturale completare la frase con la parola 'trampolino'. Io, invece, decido di completare la frase in modo tale da renderla umoristica:

I ragazzi si divertivano tantissimo a tuffarsi in piscina direttamente dall'albero, così abbiamo deciso di metterci un po' d'acqua.

Da dove viene l'umorismo, in questa frase?

Dal fatto che, innanzitutto, siamo sorpresi. L'ultima parola non è quella che ci aspetteremmo.

Perché non ce l'aspettiamo? Perché, nel leggere la frase, il nostro cervello ha costruito una serie di azioni, ed ha assegnato loro un determinato significato. Nell'assegnare tale significato, ha dovuto per forza fare delle assunzioni, perché nel testo non erano specificate. In pratica, abbiamo assegnato dei valori di default.

Abbiamo letto che i ragazzi si divertono, e che per divertirsi si tuffano in una piscina; perché questa successione di parole abbia un senso a livello semantico, *ci sembra ovvio* che la piscina debba essere piena d'acqua. Che è la condizione standard, di default, di una piscina, almeno nella nostra testa; e se mai ci dovesse venire qualche dubbio, ci pensa la parola 'divertivano' a rassicurarci. Chi mai si divertirebbe a tuffarsi in una piscina piena di mattoni?

Poi, alla fine, l'ultima parola ('un po' d'acqua') ci costringe a rivedere tutte le nostre inferenze sottocorticali; tutte le assunzioni che abbiamo potuto fare a livello intuitivo fino a quel momento. E la visione di una fila di persone che si tuffano di testa in una piscina vuota si concretizza in un sorriso, o in una risata.

Dato che siamo in una rivista di matematica, proviamo a generalizzare. La frase che abbiamo letto è del tipo 'X è talmente Y che Z'. L'umorismo nasce dal generare una frase che faccia interpretare Y in termini di X nelle condizioni di default (piscina piena d'acqua) e per far questo sono necessari termini che attivino tutte le nostre capacità di simulazione: c'è il movimento (i ragazzi si tuffavano), c'è l'emozione (i ragazzi si divertivano) e c'è la descrizione di una successione temporale (albero-piscina). È tutto talmente completo che il nostro sistema di integrazione, il proiettore cinematografico che abbiamo in testa, deve solo mettere i dettagli. Quando leggiamo, infatti, nessuno di noi si mette lì con carta e penna a fare l'analisi grammaticale: siamo talmente abili che, in continuazione, integriamo la scena con particolari che non ci sono, e che vengono direttamente dalla nostra vita, vissuta o già letta non importa.

Un computer dovrebbe essere in grado di fare questo: individuare le condizioni di default che sono associate con questa scena e rendersi conto di quando una di queste viene violata. Il che significa ricostruire, a livello computazionale, tramite associazioni di parole, le usuali situazioni che queste parole sottintendono. Costruire un contesto, una cornice, per potersi rendere conto di quando casca in terra. In questo momento, non pochi informatici sono al lavoro su questo tema. L'esempio che avete letto viene dal lavoro di Benjamin Bergen e Seana Coulson, che propongono la teoria del *frame-shifting* (spostamento del contesto) come uno dei possibili modi per insegnare a un computer come ridere.

L'umorista abile individua, in questa continua assunzione di valori di default, il punto debole, quello che può essere cambiato. In questo senso, secondo me, l'umorismo è una delle più alte capacità dell'intelletto umano; essendo in grado di individuare il confine tra l'intuito e il ragionamento, punta il dito là dove ci sembra scontato non ragionare, e ce ne fa rendere conto con una emozione positiva. Una bella risata.

Riuscire ad insegnare ad un computer come riconoscere l'umorismo significherebbe aver capito appieno come funziona questo meccanismo. A questo proposito, c'è un ultimo punto di cui bisogna parlare.

Molto spesso, l'umorismo nasce dal riconoscere il contesto appropriato. Possiamo ridere tranquillamente di fronte ai ragazzi che si tuffano in una piscina vuota, perché in fondo sono persone di carta. Se avessimo letto la stessa notizia su una locandina di quotidiano, e uno dei ragazzi fosse un nostro parente stretto – magari nostro figlio –, la cosa ci atterrirebbe. La capacità di valutare una cosa come umoristica è strettamente associata a farsi domande legate alla nostra sopravvivenza, e a chiedersi se la situazione che stiamo esaminando la mette in pericolo o meno. Non so se saremo mai in grado di progettare e realizzare un elaboratore che abbia coscienza di esistere; ma, se ci vogliamo riuscire, credo fortemente che l'umorismo sia una strada maestra per arrivarci.

Marco Malvaldi
marcoampelio@hotmail.com