MAPPE DEL LABIRINTO

UNA SERIE DI PHANTOM MAZE — AI & LANGUAGE LAB

SAGGIO 20

Embedding: la mappa senza cartografo

Claudia Marsico · Hernán Inverso

DOI 10.5281/zenodo.23647190

Il linguaggio può diventare geometria. Gli embedding mostrano fin dove può arrivare una macchina che si orienta per relazioni



Marc Chagall, Io e il villaggio (1911)

Antistene, il socratico che derideva Platone, rifiutava le definizioni e brandiva la propria ricetta: per sapere che cosa è una cosa bisogna collocarla tra le altre. L'idea ha un'aria sorprendentemente moderna. Una parola sta in una rete di relazioni con cui ha somiglianze, opposizioni o associazioni abituali. La proto-teoria dei campi semantici di Antistene ha la sua eco negli embedding, che collocano unità linguistiche in uno spazio matematico secondo le regolarità che hanno appreso dai

loro usi. La somiglianza finisce lì, ma il movimento è simile: invece di chiedere che cosa significa qualcosa bisogna chiedere dove si colloca rispetto al resto.

Quando parliamo di token ci riferiamo al testo tagliato in pezzi discreti di un vocabolario computazionale, che non corrispondono necessariamente a parole, lettere, morfemi né unità di senso. Ma una tessera da sola non serve ancora a una rete neurale, perché per calcolare ha bisogno di numeri. Il passo successivo consiste quindi nel trasformare ogni token in un vettore, cioè un elenco ordinato di numeri che possono essere immaginati come coordinate. Con due coordinate collochiamo un punto su un foglio; con tre, un punto in una stanza. Con tremila continuiamo ad avere un punto, anche se non possiamo più disegnare lo spazio in cui vive, maneggevole computazionalmente.

Marc Chagall, Io e il villaggio (1911)

Come riceve ogni token un elenco simile? In un modello c'è una grande tabella appresa durante l'addestramento, la matrice di embedding. Ogni tessera del vocabolario ha associata una riga di quella tabella. Quando appare un token, il sistema prende la riga corrispondente e ne ottiene così la sua prima rappresentazione numerica. L'identificatore del token, supponiamo che "cane" fosse un unico token e avesse l'identificatore 1734, non significa nulla di per sé. È soltanto un indirizzo dove andare a cercare il suo vettore. Quel vettore è denso, perché è composto da molti valori numerici diversi da zero, è appreso, perché nessuno ha deciso a mano quali dovessero essere i suoi numeri, ed è distribuito, perché una proprietà rilevante non vive in una sola coordinata, come se il numero 82 significasse "animale" e il 417 "plurale". Le regolarità si distribuiscono tra molte dimensioni e acquistano senso nell'insieme.

Il verbo inglese *to embed* significa incastrare, fissare qualcosa dentro una materia che lo circonda, come una pietra nel cemento. Un *embedding* incastra un'unità di linguaggio in uno spazio popolato da altre. Il token che nel saggio precedente era una tessera separata da tutte le altre diventa ora una posizione dalla quale possono calcolarsi prossimità, differenze e direzioni. L'intuizione linguistica che rende utile quella geometria veniva dalla metà del XX secolo. Zellig Harris propose di studiare le unità di una lingua osservandone la distribuzione, cioè i contesti in cui compaiono. Due parole che possono occupare luoghi simili in molte costruzioni hanno qualcosa in comune rilevabile mediante analisi distribuzionale senza definizioni. John Firth condensò pochi anni dopo l'intuizione nella frase: "conoscerai una parola dalla compagnia che tiene". L'uso lascia tracce. "Cane" appare più e più volte in ambienti dove troviamo abbaiare, guinzaglio, passeggiata, animale domestico, veterinario; "mutuo", vicino a banca, rata, interesse, abitazione, prestito. Se riuniamo quantità sufficienti di testo, quelle regolarità cominciano a formare una specie di firma statistica.

Nel 2013, Tomas Mikolov e il suo gruppo presentarono word2vec, che trovò un modo efficace di comprimere quelle tracce in vettori. In una variante, il compito di addestramento usava una parola per prevedere quali sogliono comparire vicino; nell'altra, ricostruiva una parola a partire da quelle che la circondano. Per migliorare le sue previsioni, senza dizionario né classificazione preliminare dei termini, l'addestramento modificava ripetutamente i numeri associati a ogni parola. Alla fine, le parole usate in contesti simili tendevano a restare rappresentate da vettori anch'essi simili.

Ma che cosa vuol dire che due elenchi di centinaia di numeri siano "simili"? Immaginiamo ogni vettore come una freccia che parte dall'origine di quello spazio enorme. Se due frecce puntano approssimativamente verso lo stesso lato, hanno una grande similarità del coseno; se formano un angolo maggiore, quella similarità diminuisce. Così nasce la mappa. Parole usate in modi simili tendono a occupare regioni vicine. "Cane" e "gatto" possono restare vicini perché compaiono in contesti simili, ma lo stesso può accadere con termini opposti. "Freddo" e "caldo" condividono moltissime compagnie associate a temperatura, acqua, clima, sensazione, inverno o estate. Lo spazio registra prima regolarità d'uso e a partire dalla combinazione di molte di esse può distinguere classi di relazioni differenti.

La sorpresa di word2vec fu che, oltre a formare vicinati, alcune relazioni ripetute dalla lingua apparivano come spostamenti approssimativamente regolari. Certi plurali rispetto ai loro singolari, alcuni tempi verbali, capitali rispetto a paesi o forme femminili rispetto a maschili potevano trovarsi seguendo direzioni simili. Di lì nacque l'operazione che rese celebre il modello: prendere il vettore di "re", sottrargli quello di "uomo", sommargli quello di "donna" e finire vicino a "regina". Le analogie non funzionano sempre, dipendono molto dal corpus e dal modo di addestramento e non provano che esista un'algebra universale nascosta dietro le parole, ma una relazione che nessuno aveva programmato esplicitamente acquisì una forma geometrica. La statistica dell'uso produsse direzioni.

Nel XVII secolo, Leibniz sognò una *characteristica universalis*, un alfabeto dei concetti in cui ogni idea avesse assegnato un segno, così che ragionare fosse calcolare. Davanti a qualsiasi disputa, invece di discutere basterebbe sedersi e dire *calculemus*, calcoliamo, e fare i conti fino a vedere chi ha ragione. Gli *embedding* sfiorano quel sogno stravagante fatto di concetti convertiti in vettori, analogie esprimibili come spostamenti aritmetici, anche se hanno perso per strada la tassonomia razionale che Leibniz si aspettava. Mettono anche sottosopra l'ideale di lingua adamica dove il nome riflette l'essenza della cosa. L'embedding si disinteressa di che cosa è un cane e gli interessa solo con che cosa appare e in quali contesti si muove. Dove la lingua adamica aveva essenza, lo spazio vettoriale trova vicinanza.

La geometrizzazione del linguaggio si affacciò nell'analisi semantica latente, sviluppata intorno al 1990. Il procedimento partiva da un'enorme tabella che registrava quali termini apparivano in quali documenti e applicava tecniche algebriche per comprimerla. Quella compressione faceva emergere prossimità che non erano evidenti nella tabella originale, ma word2vec portò il principio a una scala e a una precisione nuove apprendendo rappresentazioni mediante predizione locale. GloVe, presentato da ricercatori di Stanford nel 2014, dal canto suo sfruttò statistiche globali di co-occorrenza. Sebbene i metodi differissero, condividevano il limite che ogni parola riceveva un unico vettore fisso. "Banco" doveva vivere sempre nello stesso punto, fosse il sedile di una piazza o l'istituzione dove si deposita denaro. Il vettore finiva per essere una specie di media dei suoi usi.

E se una parola non avesse una posizione unica? ELMo esplorò nel 2018 l'idea di costruire rappresentazioni che dipendessero dalla frase completa. Quello stesso anno BERT, appoggiato già all'architettura transformer, generalizzò e popolarizzò il cambiamento. Da allora conviene distinguere due cose. Entrando nel modello, un token riceve un embedding iniziale relativamente stabile, preso

dalla tabella che abbiamo descritto prima. Ma quello è solo il punto di partenza, dato che man mano che attraversa gli strati il suo vettore cambia per effetto del contesto. Così, il token corrispondente a "banco" parte dallo stesso embedding lessicale di base, anche se la rappresentazione d'ingresso incorpora anche informazione sulla sua posizione e finisce per occupare posizioni diverse dopo che l'architettura incorpora il contesto. Quei vettori successivi siamo soliti chiamarli rappresentazioni contestuali o, in un senso più ampio, embedding contestuali. Invece di un unico punto per "banco" c'è una posizione costruita per quella comparsa concreta. L'attenzione è uno dei meccanismi centrali che producono quella trasformazione e in questo modo la mappa, invece di un atlante stampato che assegna a ogni parola una città fissa, risulta una mappa che si ridisegna mentre si legge la frase.

La stessa idea geometrica si espanse poi oltre le parole. Sentence-BERT, nel 2019, produsse vettori particolarmente utili per rappresentare frasi e paragrafi completi, così che due testi potessero confrontarsi per la loro vicinanza semantica. CLIP, nel 2021, portò il principio nel territorio di immagini e testi perché entrambi i tipi di oggetto potessero occupare regioni compatibili di uno stesso spazio. Una fotografia di un cane e una descrizione appropriata del cane potevano restare vicine pur essendo cose fisicamente incomparabili, facendo convivere modalità diverse dentro una stessa geometria appresa.

A che cosa serve questo? Una ricerca tradizionale si appoggia su coincidenze di termini, mentre una ricerca vettoriale può convertire domande o documenti in vettori e recuperare quelli che sono rimasti più vicini, anche se non usano lo stesso vocabolario. Chiedere "come abbassare il costo di riscaldare una casa" può recuperare un passo su "migliorare l'isolamento termico", anche se le frasi non condividono termini. Per fare quel tipo di ricerche su larga scala si usano basi di dati vettoriali, progettate per immagazzinare e trovare rapidamente vicini dentro collezioni immense di vettori. Quella ricerca è anche un pezzo centrale di molti sistemi RAG, sigla di Retrieval-Augmented Generation, generazione aumentata dal recupero. Prima di ricevere domande, il sistema prepara una collezione di testi: li divide in frammenti, calcola un vettore per ognuno e li lascia indicizzati. Quando arriva una richiesta, calcola anche il suo vettore e cerca quali di quei frammenti sono rimasti più vicini. Poi colloca i passaggi recuperati accanto alla domanda e li consegna al modello di linguaggio. L'embedding aiuta a decidere quale informazione conviene mettere davanti al modello prima che risponda. La stessa logica serve a raccomandare contenuti simili a ciò che qualcuno ha già consumato, a raggruppare documenti per affinità, a individuare casi atipici, a ordinare grandi archivi o a mettere in relazione testi e immagini. Ciò che cominciò come un modo di rappresentare parole finì convertito in una procedura generale per trasformare oggetti complessi in punti confrontabili e misurare tra loro diverse forme di somiglianza.

Torniamo al punto più strano. Nessun cartografo ha disegnato questa mappa né ha deciso in anticipo che cosa dovesse significare ciascun asse. Latitudine e longitudine sono una decisione umana, con un meridiano zero che rimase in un osservatorio di Londra come una bandiera di potere navale nel XIX secolo. In uno spazio di embedding nessuno cominciò dicendo che una certa dimensione avrebbe rappresentato il genere, un'altra la temporalità, un'altra l'autorità e un'altra il rapporto tra un animale e il suo suono. L'addestramento si concentrò sul prevedere contesti e aggiustò i numeri fino a farlo

sempre meglio. La geometria apparve come risultato. Possiamo esplorarla addestrando sonde che individuino informazione grammaticale o semantica e persino intervenire su alcuni schemi, ma la mappa funziona meglio di quanto possiamo glossarla.

Cominciammo con Antistene che abbandonava l'essenza per la relazione, cambiando la domanda che cosa è un cane con dove si colloca. In questo l'embedding è una sorta di figlio suo. Il vincolo filiale esplode quando Antistene afferma che ogni nome si riferisce a una cosa esistente fino al paradosso che l'errore è impossibile: chi dice qualcosa, dice qualcosa che è e chi dice qualcosa che è, dice la verità. L'embedding ha rotto gli ormeggi con il mondo. "Cane" riceve la sua posizione perché la parola è apparsa milioni di volte vicino ad abbaiare, a guinzaglio e a veterinario. Si confrontano parole per la loro vicinanza e il risultato è una teoria del senso che funziona senza che il mondo intervenga. Abbiamo progettato una tecnologia che può prescindere dalla domanda sulla corrispondenza. La mappa rappresenta le tracce che hanno lasciato alcuni termini comparendo accanto ad altri. Vale la pena ricordarlo perché a noi invece interessano il mondo e le cose. L'embedding mostra fin dove può arrivare un'intelligenza che si orienta per relazioni senza partire da una definizione essenziale e rende anche palese il costo di quella libertà, dato che può imparare a muoversi con una precisione sbalorditiva tra le nostre parole senza che per questo i suoi criteri di rilevanza coincidano con i nostri. Così, questa mappa può essere eccellente per percorrere il linguaggio e tuttavia non dirci dove sono le frontiere che contano nel mondo.

◊

COME CITARE

Marsico, Claudia e Hernán Inverso. 2026. "Embedding: la mappa senza cartografo". *Mappe del labirinto*, saggio 20. Phantom Maze — AI & Language Lab. https://doi.org/10.5281/zenodo.23647190

Mappe del labirinto è una serie di Phantom Maze, AI & Language Lab. La collezione vive su **phantommaze.cloud**. L'iscrizione è gratuita.

Alcune delle idee che inquadrano questo testo sono sviluppate in *Il fantasma senza ego. Una mappa nel labirinto dell'intelligenza artificiale*, di Hernán Inverso e Claudia Marsico (Phantom Maze Press, 2026; ISBN 978-987-3729-19-5).

PHANTOM MAZE

Leggi questo saggio online · Tutti i saggi · Il libro

Fonti. Scott Deerwester et al., "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, 1990, pp. 391–407; Jacob Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv*, 11 Oct. 2018, arXiv:1810.04805; J. R. Firth, "A Synopsis of Linguistic Theory, 1930–1955," *Studies in Linguistic Analysis*, special volume of the Philological Society, Basil Blackwell, 1957, pp. 1–32; Zellig S. Harris, "Distributional Structure," *Word*, vol. 10, nos. 2–3, 1954, pp. 146–62; Gottfried Wilhelm Leibniz, *Dissertatio de arte*

combinatoria, Fick and Seubold, 1666; Tomas Mikolov et al., "Efficient Estimation of Word Representations in Vector Space," *arXiv*, 16 Jan. 2013, arXiv:1301.3781; Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig, "Linguistic Regularities in Continuous Space Word Representations," *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2013, pp. 746–51; Jeffrey Pennington et al., "GloVe: Global Vectors for Word Representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2014, pp. 1532–43; Matthew E. Peters et al., "Deep Contextualized Word Representations," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, Association for Computational Linguistics, 2018, pp. 2227–37; Alec Radford et al., "Learning Transferable Visual Models from Natural Language Supervision," *Proceedings of the 38th International Conference on Machine Learning*, edited by Marina Meila and Tong Zhang, vol. 139, Proceedings of Machine Learning Research, 2021, pp. 8748–63; Nils Reimers and Iryna Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, 2019, pp. 3982–92.