

Minerva e il pappagallo. IA generativa e modelli linguistici nel laboratorio dell'umanista digitale

Fabio Ciotti

Università di Roma "Tor Vergata"
(fabio.ciotti@uniroma2.it)

Abstract

Da quando è stato rilasciato da OpenAI nello scorso autunno il sistema di interazione linguistica ChatGPT, basato a sua volta sul modello linguistico GPT-4, è assunto agli onori della cronaca e del dibattito, sia specialistico sia pubblicistico. In questo paper, dopo averne esaminato i principi, intendo esaminare le possibili ricadute di questa classe di sistemi sulle pratiche della ricerca scientifica in ambito umanistico e in particolare in ambito umanistico digitale o computazionale. La tesi che propongo è che ChatGPT già nello stato di avanzamento in cui si trova oggi 1) rappresenta un ottimo strumento di supporto per la progettazione e implementazione di workflow di ricerca; 2) potrebbe avere un impatto rilevante nella qualità e nella efficacia della ricerca umanistica digitale/computazionale.

Parole chiave

Intelligenza artificiale (AI), Large Language Models (LLM), ChatGPT

DOI

<https://doi.org/10.58015/2036-2293/671>

Diritto d'autore

Questo lavoro è fornito con la licenza *Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale*: <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

Gli autori mantengono il diritto d'autore sui propri articoli e materiali supplementari e mantengono il diritto di pubblicazione senza restrizioni.

1. Introduzione¹

L'evoluzione prodigiosa dei sistemi di Intelligenza Artificiale che operano in ambito linguistico cui abbiamo assistito nell'ultimo decennio è un fenomeno di enorme impatto non solo scientifico e tecnologico, ma anche sociale e culturale. Le reazioni evocate da questa evoluzione, e in particolare dall'introduzione al pubblico del sistema conversazionale online ChatGPT da parte di OpenAI² nel novembre del 2022, assurto agli onori della cronaca e del dibattito – sia di quello specialistico sia di quello della pubblicistica non specializzata e del chiacchiericcio dei social media – lo testimoniano. ChatGPT³ in realtà è una interfaccia utente basata sul Web che permette di interagire in modo semplice con alcuni esemplari della famiglia di grandi modelli linguistici (Large Language Model, LLM) generativi sviluppati da OpenAI. Di recente sono stati resi pubblici anche Bard, servizio simile sviluppato da Google, basato sul modello linguistico PALM2⁴, e Claude 2, sviluppato da Anthropic⁵. E sono ormai numerosi i modelli e le applicazioni conversazionali derivati da LLAMA e LLAMA 2, i modelli sviluppati da Meta, l'unico tra i grandi protagonisti dell'economia digitale globale a rilasciare i suoi modelli in open access⁶.

Il complesso di tecnologie alla base della IA generativa – non esclusivamente quella linguistica, poiché altrettanto clamorosi sono stati gli sviluppi nei modelli in grado di generare immagini statiche, come Dall-E⁷, Stable Diffusion⁸ e Midjourney⁹, e già sono avanzate le sperimentazioni nella generazione di immagini in movimento e ambienti digitali 3D – e dei cosiddetti modelli fondazionali¹⁰ rappresenta con ogni probabilità la

¹ Questo articolo estende e rielabora il paper presentato al convegno AIUCD 2023 Fabio Ciotti, «ChatGPT: un Pappagallo Stocastico può essere di aiuto a un Vero Ricercatore (Umanistico)?», *La memoria digitale: forme del testo e organizzazione della conoscenza. Atti del 12° Convegno annuale AIUCD. Siena: Università degli Studi di Siena, Venezia, AIUCD - Ass. per l'informatica umanistica e la cultura digitale, 2023*, pp. 245–250.

² OpenAI, «Introducing ChatGPT», *OpenAI*, novembre 30, 2022, <https://openai.com/blog/chatgpt>.

³ Nel momento in cui scriviamo il servizio web ChatGPT può usare due diverse versioni della generazione 3.5 e la più recente ed estesa, ancorché assai meno trasparente, generazione 4 del modello OpenAI, «GPT-4 Technical Report», 2023, <https://arxiv.org/abs/2303.08774>.

⁴ Ghahramani, «Introducing PaLM 2», *Google Blog*, maggio 10, 2023, <https://blog.google/technology/ai/google-palm-2-ai-large-language-model/>.

⁵ <https://www.anthropic.com/index/claude-2>.

⁶ Hugo Touvron et al., «Llama 2: Open Foundation and Fine-Tuned Chat Models», 2023, <https://arxiv.org/abs/2307.09288>.

⁷ <https://openai.com/research/dall-e-3-system-card>. Ma questa ultima versione del modello è stata integrata in ChatGPT, che fornisce ora un ambiente integrato per generare sia testo sia immagine, e molto altro.

⁸ <https://stability.ai/stablediffusion>.

⁹ <https://www.midjourney.com>.

¹⁰ L'appellativo "generativa" deriva dal fatto che questi sistemi sono in grado di produrre nuova informazione a partire da informazione in ingresso, e non si limitano a individuare pattern e classificare l'input come le tradizionali applicazioni di reti neurali nel passato. L'espressione *foundation models* è stata introdotta dai ricercatori dello "Stanford Institute for Human-Centered Artificial Intelligence's (HAI) Center for Research on Foundation Models (CRFM)" nel report pubblicato nell'agosto 2021, per indicare "any model that is trained on broad data (generally using self-supervision at scale) that can be adapted (e.g., fine-tuned) to a wide range of downstream tasks". Rishi Bommasani et al., «On the Opportunities and Risks of Foundation Models», *ArXiv*, 2021, <https://crfm.stanford.edu/assets/report.pdf>: 3.

più rilevante innovazione a far data dall'invenzione del Web. Ma, mentre l'idea geniale di Tim Berners Lee passò del tutto inosservata per quasi un decennio, prima di divenire il motore della trasformazione digitale di inizio millennio, i modelli generativi, e in particolare ChatGPT, hanno immediatamente catturato l'attenzione della pubblica opinione mondiale, nonché scatenato l'ennesima controversia tra apocalittici e integrati (solo con toni assai più apocalittici del solito, se si considera che ci sono persone, anche molto intelligenti, che considerano questi sistemi una possibile minaccia esistenziale per l'umanità). Probabilmente gli effetti sulla comunicazione sociale dell'invenzione del Web e poi dei social media, hanno un ruolo in tutto questo affollato dibattito; ma è altrettanto rilevante il fatto che queste tecnologie, per la prima volta dopo 70 anni di tentativi falliti nella storia dell'Intelligenza Artificiale, mostrano di possedere competenze cognitive di alto livello, tanto da essere paragonabili a quelle umane in vari compiti; in particolare, quello di usare il linguaggio naturale.

Naturalmente la genuina capacità di comprendere di GPT e simili, o di essere dotato di vere capacità razionali, e persino di poter avere una qualche forma di coscienza, è oggetto di un acceso dibattito, che vede una maggioranza di scettici (o persino di critici radicali), capitanati da Emily Bender, coautrice del famigerato saggio sugli *Stochastic Parrots*¹¹, ma alcuni importanti fautori, come il filosofo della mente David Chalmers¹². Prendere posizioni serie e fondate su temi come questi – ma anche su questioni meno serie ma più fascinosamente apocalittiche, come il rischio esistenziale legato a una possibile presa del potere delle intelligenze artificiali, stile *Matrix* e *Terminator* – così come valutare le possibili conseguenze di queste innovazioni sul mondo del lavoro, o sulla comunicazione sociale, richiede di avere una idea chiara di cosa siano e di come funzionino i modelli linguistici. Cosa non esattamente facile, visto che la matematica alla base di questi sistemi non è proprio alla portata di tutti, e che ogni descrizione per analogia rischia di essere fuorviante. Ma il meglio è nemico del bene, e quindi in questo articolo scommettiamo sul fatto che una valida approssimazione sia di sicuro più utile di tante immaginifiche narrazioni fantascientifiche.

Tra i molti temi sui cui si è incentrato tale dibattito, che non è possibile riassumere in questa sede, non mancano quelli relativi alle applicazioni degli LLM generativi e delle relative conseguenze, nel mondo della ricerca e della didattica. Queste ultime in particolare hanno destato una notevole apprensione, almeno in parte giustificata poiché è evidente che un sistema di generazione linguistica così efficiente ed efficace sia nella coerenza sintattica sia in quella semantica dei testi generati, e dotato di una interfaccia semplice da usare, potrebbe sicuramente essere utilizzato dagli studenti per produrre almeno parte di quella serie di contributi scritti mediante i quali vengono giudicati e valutati¹³. Ma anche il settore dell'editoria scientifica è stato agitato da allarmi che hanno

¹¹ Emily M. Bender et al., «On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?», *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, presentato al FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event Canada, ACM, 2021, <https://dl.acm.org/doi/10.1145/3442188.3445922>, pp. 610–623.

¹² David J. Chalmers, «Could a Large Language Model Be Conscious?», *Boston Review*, 2023, <https://www.bostonreview.net/articles/could-a-large-language-model-be-conscious/>.

¹³ Stephen Marche, «The College Essay Is Dead. Nobody is prepared for how AI will transform academia», *The Atlantic*, dicembre 6, 2022,

portato alcuni importanti editori¹⁴ e repository di *preprints*¹⁵ a definire dei veri e propri documenti di policy, nonché la MLA a fornire una prima indicazione sulle modalità di citazione del contributo di un LLM al contenuto di un lavoro scientifico¹⁶.

Mi interessa piuttosto in questa sede riflettere sulle possibili ricadute di un sistema come questo – o dei suoi consimili che a breve inonderanno il mercato della cosiddetta intelligenza artificiale –, sulle pratiche della ricerca scientifica in ambito umanistico e in particolare in ambito umanistico digitale o computazionale. Intendo dunque fornire alcuni spunti di riflessione relativamente al fatto che ChatGPT già nello stato di avanzamento in cui si trova oggi:

1. rappresenta un ottimo strumento di supporto per la progettazione e implementazione di *workflow di ricerca*;
2. potrebbe avere un impatto rilevante nella qualità e nella efficacia della ricerca umanistica digitale/computazionale.

Queste due osservazioni valgono in primo luogo se è verificata la condizione necessaria che studenti e studiosi di area umanistica abbiano una sufficiente conoscenza del modo in cui un LLM come GPT funziona, come è vero per ogni altro metodo matematico/computazionale¹⁷.

2. Un po' di storia dell'Intelligenza Artificiale

La nozione di Intelligenza Artificiale (IA) ha una storia piuttosto lunga: infatti risale alla metà degli anni 50 del secolo scorso quando, sulla spinta dello sviluppo dei primi computer digitali, un gruppo di studiosi provenienti da campi diversi come la matematica, la logica, la psicologia, la linguistica, iniziarono a teorizzare ed esplorare sperimentalmente la possibilità di replicare i comportamenti intelligenti superiori della mente umana attraverso programmi per computer. L'atto di nascita ufficiale del termine IA e del relativo campo di studi viene fatto risalire al Dartmouth Workshop organizzato nell'estate del 1956 da John McCarthy (cui la vulgata attribuisce il conio dell'espressione *Artificial Intelligence*) e Marvin Minsky, allora giovani ricercatori emergenti, con la collaborazione dei già affermati Claude Shannon e Nathan Rochester. In realtà i mattoni concettuali alla base dell'IA erano già da tempo stati sviluppati, in particolare grazie al contributo di Alan Turing che già nel 1950 aveva pubblicato il suo famoso saggio «Computing machinery and intelligence» in cui tra le altre cose viene presentata per la

<https://www.theatlantic.com/technology/archive/2022/12/chatgpt-ai-writing-college-student-essays/672371>.

¹⁴ «Tools such as ChatGPT threaten transparent science; here are our ground rules for their use», *Nature*, vol. 613, fasc. 7945, gennaio 2023, pp. 612–612.

¹⁵ «arXiv announces new policy on ChatGPT and similar tools – arXiv.org blog», s.d., <https://blog.arxiv.org/2023/01/31/arxiv-announces-new-policy-on-chatgpt-and-similar-tools/?fbclid=IwAR05arV1UDPVN5bb1Bh4MchaylYxWvcYfX9mslCUHCKY8-Sm4Fe5gwGmKYM>.

¹⁶ «How do I cite generative AI in MLA style?», *MLA Style Center*, marzo 17, 2023, <https://style.mla.org/citing-generative-ai/>.

¹⁷ Ted Underwood, *Distant horizons: digital evidence and literary change*, Chicago, The University of Chicago Press, 2019, par. par. The Missing Curricular Foundation; Michael Gavin, *Literary mathematics: quantitative theory for textual studies*, Stanford text technologies, Stanford, California, Stanford University Press, 2023.

prima volta l'idea del "gioco dell'imitazione" (noto poi come "test di Turing") come approccio operativo alla valutazione dell'intelligenza di un sistema artificiale¹⁸.

L'idea alla base di questa impresa scientifica era che non ci fosse una differenza sostanziale tra ciò che avviene in un cervello e ciò che avviene in un processore elettronico. Ovvero che la differenza nel sostrato materiale fosse del tutto inessenziale ai fini di spiegare come funzionano i processi cognitivi come la comprensione di un testo, la soluzione di un problema complesso o anche il giocare a scacchi. Secondo questa idea ciò che caratterizza una mente in quanto soggetto di comportamenti cognitivi intelligenti è il fatto di elaborare rappresentazioni interne discrete (simboli), connesse causalmente alla realtà, sulla base di regole, esattamente ciò che fa un computer, ovvero una qualsiasi macchina formale equivalente alla macchina di Turing¹⁹. Il problema pratico, dunque, consisteva dunque nella formulazione dei processi funzionali eseguiti da un cervello sottospecie di algoritmi, e nella loro implementazione su un computer digitale.

A questa idea originale dell'intelligenza artificiale si oppose molto presto una visione diversa che riteneva impossibile replicare comportamenti intelligenti superiori senza tenere conto della struttura biologica di basso livello e della fisiologia del cervello. In primis, il fatto che il cervello fosse composto da cellule interconnesse, i neuroni, che si scambiano messaggi elementari, attivandosi reciprocamente, e che il cervello apprende induttivamente dall'esperienza. Questo modello biologico poteva essere in qualche modo simulato attraverso dei processi computazionali, ma di tutt'altra natura rispetto a quelli implementati nell'IA classica, processi sub-simbolici, cioè, elaborazione di valori numerici attraverso semplici funzioni matematiche, i cui risultati si propagano attraverso una rete di neuroni artificiali.

¹⁸ A. M Turing, «Computing machinery and intelligence», *Mind*, vol. 59, fasc. 236, 1950, pp. 433–460.

¹⁹ I fondamenti filosofici di questa tesi, la teoria funzionalista della mente, furono dati pochi anni dopo da Hilary Putnam nel saggio «Minds and Machines», Sidney Hook (a cura di), *Dimensions of Minds*, New York University Press, 1960, pp. 138–164. Ma assai rilevante fu anche il contributo di Noam Chomsky, che proprio in quegli anni aveva pubblicato il testo fondativo della sua teoria linguistica *Syntactic Structures*, Berlin, Boston, De Gruyter Mouton, 1957, <https://doi.org/10.1515/9783112316009>.

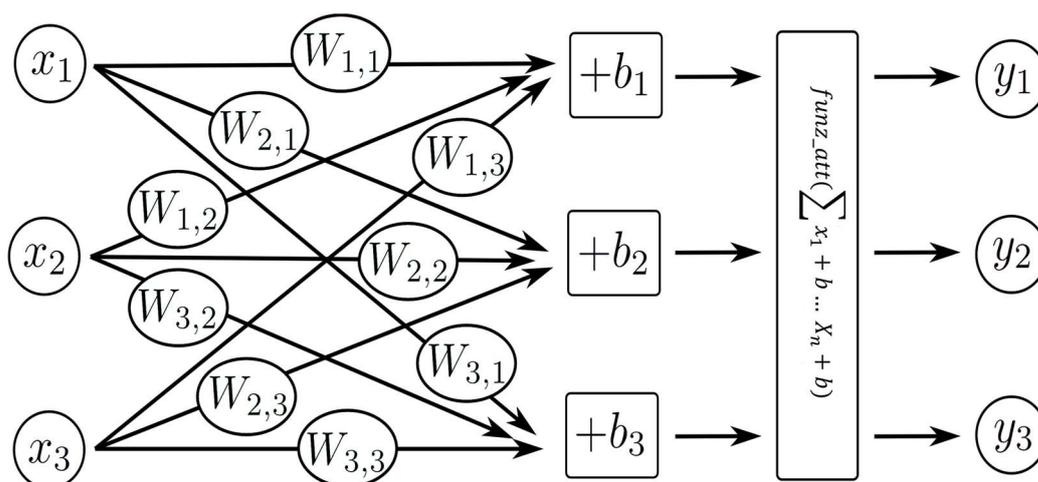


Figura 3 Schema di una rete neurale a due strati con nodi completamente connessi

Una rete come questa è composta da nodi interconnessi. Ogni nodo riceve in ingresso n valori numerici dai neuroni dello strato precedente (a partire dallo strato di input superficiale), che vengono moltiplicati per un peso, o parametro, specifico per ciascuna connessione, e poi sommati (a questa somma si aggiunge anche un termine fisso detto *bias*, per ragioni che è troppo lungo spiegare). La risposta del neurone, infine, viene fornita da una *funzione di attivazione* che applica una trasformazione *non lineare* a tale somma, e invia il risultato ai neuroni dello strato successivo. Da un punto di vista matematico una rete neurale artificiale applica una serie di operazioni algebriche su vettori e matrici. Il numero dei nodi, la loro stratificazione, la funzione e i valori dei parametri ne determinano il comportamento complessivo, ma solo questi ultimi possono essere cambiati una volta definita l'architettura della rete. Questo cambiamento è l'esito di un processo di apprendimento (anche se in teoria si possono predisporre con una distribuzione predefinita).

La prima formulazione del concetto di neurone artificiale (*Threshold Logic Unit*) fu avanzata da Warren McCulloch e Walter Pitts nel 1943²⁰, in un modello che era ancora legato all'elaborazione di simboli discreti binari per calcolare funzioni booleane, sebbene introducesse molti degli aspetti caratterizzanti delle future reti neurali. Ma il primo vero esploratore di questa architettura computazionale fu Frank Rosenblatt, che a partire dalla metà degli anni Cinquanta implementò le idee di McCulloch e Pitts sviluppando il primo *Perceptron hardware*, il *Mark 1 Perceptron*. Si trattava di una rete neurale fisica a uno strato che eseguiva il riconoscimento di immagini approssimando una funzione di classificazione lineare binaria mediante un rudimentale processo di apprendimento.

Nonostante il promettente lavoro di Rosenblatt, questa idea della computazione neurale venne ritenuta errata o, nel migliore dei casi, inefficace da parte della maggioranza della comunità di ricerca della IA, rispetto a un approccio che tendeva a modellizzare direttamente le funzioni cognitive superiori. Le reti neurali, infatti, almeno le reti neurali nelle loro iniziali implementazioni, i Perceptron a singolo strato, non erano

²⁰ Warren S. McCulloch, Walter Pitts, «A logical calculus of the ideas immanent in nervous activity», *The bulletin of mathematical biophysics*, vol. 5, fasc. 4, dicembre 1943, pp. 115–133.

in grado di scalare da elementari compiti di riconoscimento di forme semplici nell'ambiente a problemi che richiedessero astrazione o argomentazione o razionalità come la dimostrazione di un teorema, laddove i metodi simbolici eccellevano proprio in questo settore. Nel 1967 Minsky e Papert pubblicarono un libro in cui dimostravano formalmente queste limitazioni (in particolare, dimostrarono l'impossibilità per questi sistemi di apprendere la funzione booleana XOR) e con questo chiusero di fatto la ricerca in questa direzione per molti anni

Solo a partire dagli anni 80 il paradigma neurale riprende vigore, proprio in contemporanea con la grande crisi dell'intelligenza artificiale classica sia dal punto di vista delle teorie e filosofie della mente sia dal punto di vista delle implementazioni ingegneristiche. La ripresa del paradigma neurale è dovuta a diversi fattori: in primo luogo lo sviluppo di un metodo generale di simulazione dell'apprendimento, necessario affinché una rete neurale possa essere addestrata a compiere compiti specifici, denominato *back propagation*, che è alla base degli sviluppi più recenti del *machine learning*. Si tratta di un algoritmo per l'aggiornamento dei pesi della rete introdotto nello sviluppo di reti neurali sin dagli anni 80 dal gruppo *Parallel Distributed Processing* di Rumelhart, e McClelland²¹, basato su un procedimento matematico di ottimizzazione detto *gradient descent*, il cui scopo è individuare i minimi locali di una funzione differenziabile²².

Altro fattore di sviluppo è stato l'introduzione di modelli di reti neurali assai più complessi rispetto ai modelli originali, che prevedono numerosi strati o livelli intermedi di nodi della rete neurale. Queste architetture innovative, introdotte a partire dalla fine degli anni 90, hanno determinato un vero e proprio salto generazionale nella capacità di elaborazione delle reti neurali, per arrivare a quello che oggi vediamo nei modelli linguistici come Bert, GPT e tutte le loro ulteriori derivazioni dirette o indirette, così come nelle reti convoluzionali, in grado di riconoscere complesse configurazioni visuali e così via.

Un fattore che ha facilitato lo sviluppo della nuova intelligenza artificiale è stata anche la disponibilità di potenza di calcolo, inimmaginabile negli anni 50, così come la disponibilità di una enorme quantità di dati su cui poter addestrare le reti neurali. Questi due elementi, apparentemente esterni, sono in realtà dirimenti poiché i grandi modelli linguistici e i vari sistemi di riconoscimento di immagini, ampiamente utilizzati anche in ambito industriale, non sarebbero pensabili se non si avesse la possibilità di addestrarli su milioni di esempi.

3. Cosa è e come funziona GPT, in breve

ChatGPT è, come detto, una interfaccia Web che permette di interagire con un modello linguistico della famiglia GPT di OpenAI, acronimo di *Generative Pre-trained Transformer*.

²¹ David E. Rumelhart, James L. McClelland, AU, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, The MIT Press, 1986, <https://direct.mit.edu/books/book/4424/Parallel-Distributed-ProcessingExplorations-in-the>.

²² Sul Web si trovano numerose introduzioni su questi e altri aspetti tecnici, in particolare sul Blog *Medium*. Tra i tanti segnaliamo Hunter Phillips, «A Simple Introduction to Gradient Descent», *Medium*, giugno 10, 2023, <https://medium.com/@hunter-j-phillips/a-simple-introduction-to-gradient-descent-1f32a08b0deb>.

Questi a loro volta sono modelli probabilistici del linguaggio naturale costituiti da reti neurali artificiali basate su una complessa architettura detta *decoder-only transformer* con *self-attention*²³. I modelli come GPT sono reti di enormi dimensioni, caratterizzate da una grandissima quantità di parametri (i pesi assegnati alle connessioni tra i nodi): il modello più vasto della generazione 3 conta 175 miliardi di parametri²⁴; purtroppo per la generazione 4 OpenAI ha assunto una policy di assoluta segretezza, per cui non sono noti dettagli sia architetturali sia sulle fonti utilizzate per l'addestramento del modello.

Obiettivo di questo genere di modelli generativi è appunto produrre una risposta a un input di testo (*prompt*) in modo linguisticamente coerente (sia dal punto di vista sintattico sia da quello semantico) e auspicabilmente pertinente. Poiché un testo è, in prima istanza, una sequenza di unità, o token atomici, concatenate in modo sequenziale, possiamo descrivere un LLM come un sistema che data una sequenza di token iniziale è in grado predire in modo ricorsivo (o autoregressivo) il token successivo a sinistra sulla base di una stima probabilistica condizionale: $P(W_{k+1} | W_0 \dots W_k)$ dove W_n indica i token del testo²⁵.

In generale, l'architettura degli attuali modelli linguistici generativi è costituita da una serie di moduli *decoder-only transformer* – anche se la prima formulazione prevedeva dei moduli *encoder-decoder*, poiché era pensata per lo specifico task della traduzione automatica. Ogni modulo a sua volta è costituito da uno strato di *multihead masked self-attention* seguito da una rete *feed-forward* (di norma un *perceptron* multistrato), inframmezzati da vari strati di normalizzazione. La pila di transformers è preceduta da uno strato di codifica che prende i testi in input e li codifica in modo tale che il resto della rete possa utilizzarli, e da uno di decodifica che applica il processo inverso e restituire il testo di input più il token aggiunto. In fase di esecuzione (o di inferenza) questo output viene automaticamente reinserito nella rete e il processo viene ripetuto finché non venga generato un simbolo di fine sequenza o si raggiunga un limite prefissato di token da generare: per questo i modelli linguistici vengono detti anche *modelli auto-regressivi*.

Non è possibile in questa sede esplorare tutti i dettagli tecnici e matematici che spiegano il funzionamento di questo genere di sistemi. Vale la pena soffermarsi brevemente su alcuni aspetti rilevanti. Il primo problema da risolvere per costruire un generatore di parole e di discorsi sensati e coerenti è: come *rappresentare* il linguaggio in un modo che sia accessibile a un procedimento di natura fondamentalmente matematica? Insomma, come trasformare le parole in numeri, e come assicurarci che tale rappresentazione catturi in un senso profondo le strutture e il funzionamento del

²³ Ashish Vaswani et al., «Attention Is All You Need», 2017, <https://arxiv.org/abs/1706.03762>; Jacob Devlin et al., «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding», *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, Association for Computational Linguistics, 2019, <https://aclanthology.org/N19-1423>, pp. 4171–4186.

²⁴ Tom Brown et al., «Language Models are Few-Shot Learners», H. Larochelle et al. (a cura di), *Advances in Neural Information Processing Systems*, vol. xxxiii, Curran Associates, Inc., 2020, https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf, pp. 1877–1901.

²⁵ Stephen Wolfram, «What Is ChatGPT Doing ... and Why Does It Work?», *Stephen Wolfram: Official Website*, febbraio 14, 2023, <https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work/>.

linguaggio naturale. La tecnica detta *word embedding* fornisce una soluzione efficace ed efficiente a questo problema, e si basa sulla teoria semantica distribuzionale, secondo cui il significato di un termine in una lingua è fissato dall'insieme dei contesti di uso di quel termine²⁶.

Il word embedding è un procedimento computazionale che converte le parole in sequenze di numeri, codificando il loro significato in un formato che la macchina può comprendere e manipolare²⁷. Per fare questo vengono esaminati un numero molto grande di contesti verbali in cui compare la parola in questione: l'algoritmo o, meglio, gli algoritmi visto che ce ne sono diversi, sui cui dettagli non ci soffermiamo, sintetizza tutti questi contesti in un vettore (una lista) di numeri reali. Un LLM è dunque dotato di una sorta di vocabolario di uso in cui tutte le parole che conosce sono codificate in questo modo (parlare di parole è un'ulteriore semplificazione, poiché GPT lavora su parti di parole, token, ma questa distinzione è ininfluente al livello di approssimazione in cui ci siamo collocati²⁸). L'input del modello, dunque viene in prima istanza formalizzato sotto forma di sequenze di token, che vengono convertiti in numeri utilizzando un vocabolario che assegna poi a ogni token il suo *embedding vettoriale*. Ad esempio²⁹:

1. Il - gatto - saltò - sul - tavolo
2. [2, 4, 7, 10, 12]
3. [099837849898..., 934898689573589..., 093745282902...]

Una volta provveduto alla trasformazione del prompt in una sequenza di vettori di embedding (altro dettaglio: all'embedding dei token viene aggiunto l'embedding della posizione di ogni token nella sequenza, necessaria poiché i transformer sono reti *feed-forward*, unidirezionali, che leggono l'intera sequenza di input nello stesso momento), tali vettori vengono passati ai vari strati di transformer. Qui gioca un ruolo cruciale il processo di calcolo dell'attenzione. Si tratta di una tecnica utilizzata per elaborare il testo di input e per capire quali parole del contesto a sinistra del token corrente sono più rilevanti per proseguire il discorso in modo coerente. Ad esempio, se in una frase compare un pronome l'attenzione sarà in grado di individuare l'entità a cui esso si riferisce, e di conseguenza le parole successive proposte dal modello saranno selezionate in modo opportuno: "Il **gatto** saltò sul tavolo e **lo**": "*sgridai*". In realtà l'attenzione calcola il contributo di ogni token precedente, assegnando loro una certa probabilità; inoltre, in

²⁶ John Rupert Firth, *Papers in Linguistics 1934–1951*, London, Oxford University Press, 1957. Una trattazione recente ed estensiva del tema in Alessandro Lenci e Magnus Sahlgren, *Distributional Semantics (Studies in Natural Language Processing)*, Cambridge: Cambridge University Press, 2023. doi:10.1017/9780511783692.

²⁷ Tomas Mikolov et al., «Efficient Estimation of Word Representations in Vector Space», 2013, <https://arxiv.org/abs/1301.3781>.

²⁸ Si veda una semplice introduzione all'algoritmo di tokenizzazione usato da gran parte degli LLM di nuova generazione, denominato *Byte-Pair Encoding*, su <https://huggingface.co/learn/nlp-course/chapter6/5?fw=pt#byte-pair-encoding-tokenization>. Più formalmente si veda Rico Sennrich, Barry Haddow, Alexandra Birch, «Neural Machine Translation of Rare Words with Subword Units», 2015, <https://arxiv.org/abs/1508.07909>. Per vedere come GPT applica questo processo a una frase si può sperimentare con l'applicazione su <https://platform.openai.com/tokenizer>.

²⁹ L'esempio è fittizio e fornisce una visione semplificata della tokenizzazione di GPT, che, come detto, non è basata su parole ma su unità sub-parola.

ogni modulo transformer ci sono molti blocchi (o *head*) di attenzione paralleli, ciascuno dei quali valuta diverse relazioni sintattiche e semantiche dell'input. La matematica dell'attenzione è piuttosto astrusa, ma intuitivamente è chiaro come questo meccanismo sia in grado di determinare in modo causale e motivato la generazione del linguaggio da parte di un LLM, rendendolo in grado di adattare le sue risposte allo specifico contesto discorsivo e dunque di trattare in modo sorprendentemente efficace tutte le ambiguità sintattiche e semantiche del linguaggio naturale.

Infine, occorre dedicare qualche parola alla fase finale del processo, quella di decodifica e di selezione del nuovo token da aggiungere alla sequenza di input. Come detto, l'architettura di questi modelli prevede che la sequenza di vettori di input passi attraverso diversi strati di transformer (GTP-3, ad esempio, ha 96 strati). Al termine di questa serie di elaborazioni viene prodotta una sequenza di vettori di dimensione identica a quella dell'input ma con i valori aggiornati e arricchiti di tutte le informazioni prodotte dal processo precedente. A questo punto in fase di esecuzione il vettore di embedding dell'ultimo token viene inviato a uno strato di decodifica che produce un vettore la cui struttura e dimensione è identica a quella del dizionario (nei modelli più recenti dovrebbe ammontare a circa centomila *type*) e i cui valori sono la probabilità assegnata al token corrispondente di essere il successore dell'ultimo token della sequenza di input. Ad, esempio, se l'input è "Il gatto salto sul" questo vettore potrebbe le seguenti probabilità:

tavolo	0.15
divano	0.12
letto	0.11
...	...
vettore	0.001

Data questa distribuzione di probabilità ci sono diverse strategie che si possono adottare per selezionare attualmente un token. La più ovvia potrebbe sembrare quella di optare per la parola con la maggiore probabilità (*greedy decoding*). Tuttavia, questa strategia porterebbe a una fortissima ripetitività della generazione linguistica (che in molti casi trasformerebbe GPT in un pappagallo in senso stretto). Per questo gli LLM usano diverse strategie per innalzare la creatività linguistica. La prima è normalizzare la distribuzione di probabilità usando un parametro detto *temperatura*, al crescere della quale cresce l'uniformità della distribuzione (ovvero si restringe l'intervallo di varianza dei valori). A questo punto invece di emettere sempre il token con probabilità massimale, viene effettuata un campionamento casuale sugli n token più probabili, o su quelli che hanno una probabilità cumulativa superiore a un certo valore. In conclusione, la scelta finale di ogni parola generata è soggetta a un algoritmo stocastico per evitare che sia sempre la parola *più* probabile a essere scelta ed aumentare così la "creatività" del modello³⁰. Il

³⁰ Questo è in effetti l'unico processo strettamente stocastico (i.e. aleatorio) nel funzionamento di un LLM a transformer, contrariamente alla vulgata che li vuole pappagalli stocastici, poiché come abbiamo visto il meccanismo dell'attenzione fa sì che la generazione del *token* successivo sia anche causalmente determinata.

modello, si potrebbe dire, combina la sua conoscenza del linguaggio con la sua capacità di valutare la probabilità di ogni parola in base al contesto in cui si trova unita, a una certa dose di casualità.

4. La fase di apprendimento di un LLM

La complessa architettura che abbiamo visto, come detto, è di fatto basata su una rete di reti neurali, e come tale il suo comportamento non è programmato (non è algoritmico in senso tradizionale), ma emerge dopo un complesso processo di apprendimento. Come GPT (e i vari altri modelli linguistici oggi disponibili) acquisisce la sua conoscenza del linguaggio, dunque? Alla base dell'addestramento di un transformer c'è sempre una variante del già noto algoritmo di *back-propagation*. Ma la generazione più recente di modelli linguistici deve il suo successo alla convergenza di varie fasi di apprendimento, basate su strategie diverse, come rappresentato in figura 3.

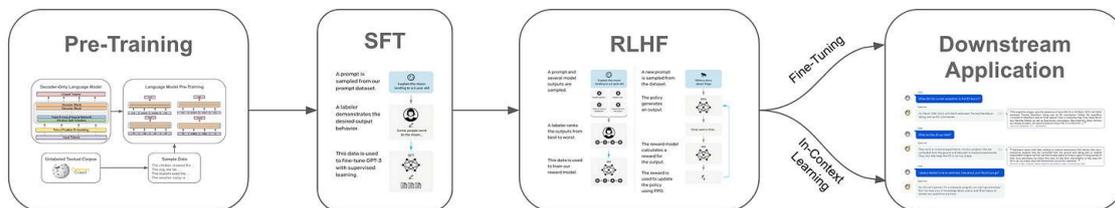


Figura 4 Le varie fasi del processo di addestramento di un modello linguistico (fonte: <https://cameronrwolfe.substack.com/p/data-is-the-foundation-of-language>)

La fase fondamentale e di gran lunga la più complessa e onerosa è quella detta di *self pre-training*. Il prefisso “pre” sta a indicare il fatto che modelli come questi vengono in prima istanza addestrati in modo generico a usare il linguaggio, e poi per essere utilizzati in compiti determinati devono essere adattati o specializzati, ovvero devono passare attraverso una fase di *fine-tuning*. Ma l'aspetto più innovativo di questa fase di addestramento negli LLM è quella legata all'idea di autoapprendimento. Infatti, a differenza di quanto è sempre avvenuto con le reti neurali, gli LLM possono apprendere da soli dai dati, senza che questi debbano essere etichettati manualmente da esperti umani. Data la quantità di testo necessaria ad addestrare un LLM di qualità pari a quella di GPT-4, senza questa possibilità sarebbe stato impossibile sviluppare concretamente questa tecnologia. Infatti, GPT viene addestrato selezionando blocchi di testo da enormi dataset testuali estratti dal Web, quali CommonCrawl (commoncrawl.org), WebText2 (post su Reddit, openwebtext2.readthedocs.io), Wikipedia e altri numerosi archivi di testi e pubblicazioni digitali³¹. Il pre-training richiede mesi di elaborazione continua con migliaia di unità GPU funzionanti in parallelo, e ha di conseguenza costi ingenti.

La quasi completa automazione dell'addestramento sfrutta il fatto che in una sequenza testuale (ma la cosa si può generalizzare a ogni tipo di dati discreti sequenziali), il target del processo è di fatto implicito nei dati di training stessi. Data una sequenza di k token, infatti, basta usare i primi $t_1 \dots t_{k-1}$ token come prompt e il k -esimo token come target. Riprendendo l'esempio della frase “il gatto saltò sul tavolo” visto

³¹ Per dare dei numeri il training set di GPT-3 è composta da circa da circa 400 miliardi di token, che corrisponde a circa 570 Gbyte, selezionati da un corpus di 45 terabyte di testo. Cfr. T. Brown et al., *op.cit.* Purtroppo, come già detto nulla di preciso si sa della versione 4.

sopra, in fase di addestramento, il modello riceve la sequenza di token [2, 4, 7, 10] come prompt di input e il token [12] come target. Nel primo passo della fase di addestramento, con i parametri inizializzati in modo casuale (o semi-casuale) il modello fornirà una risposta altrettanto casuale e scorretta – mettiamo “lago” – alla fine del processo di inferenza. A questo punto il vettore del risultato (lago) viene confrontato con il vettore del target (tavolo), e la differenza tra questi due vettori, definita *loss*, misura quanto il modello è lontano dalla risposta corretta e viene usata come input della funzione di ottimizzazione usata nel processo di *back propagation*. Questo aggiorna tutti gli innumerevoli pesi (parametri) del modello utilizzando un algoritmo di ottimizzazione, che aggiusta gradualmente i pesi in modo da approssimare la minimizzazione della ‘perdita’ calcolata durante la fase precedente. Iterando questo processo di ottimizzazione durante l’addestramento per decine di volte per miliardi di token, il modello continua a migliorare la sua precisione e ad aumentare la sua capacità di generare un output coerente e pertinente. Che, nel caso dei modelli linguistici, significa individuare la distribuzione di probabilità che un insieme di parole (il dizionario) ha di seguire una data sequenza testuale di input.

Ma la capacità conversazionale che apprezziamo usando ChatGPT, non è solo il prodotto di questa fase di massivo auto-addestramento alla conoscenza della sintassi e della semantica di decine di lingue. Infatti, un modello base, come quello prodotto da questa fase, non sarebbe in realtà in grado di conversare, scrivere poesie, risolvere problemi matematici, sviluppare teorie e ipotesi, analizzare dati e programmare come vediamo fare con efficacia incredibile (e con risultati che in molti casi superano il livello delle prestazioni umane, almeno se si misurano mediante i benchmark più comuni) da parte dei modelli linguistici più potenti e di recente generazione. Per ottenere queste prestazioni sono necessarie una combinazione di altre tecniche di *fine tuning* supervisionato, che richiedono dunque l’intervento umano³². In particolare, nell’addestramento di GPT-3.5 e dei successori hanno avuto un ruolo importante una fase di addestramento guidato a seguire istruzioni e domande espresse in forma linguistica (*supervised fine-tuning*, SFT, o più precisamente *instruction fine-tuning*) e una volta a migliorare le capacità conversazionali e argomentative (nonché l’aderenza ad assunti etici) basata sul *reinforcement learning* (*RL with Human Feedback*, RLHF).

³² Long Ouyang et al., «Training language models to follow instructions with human feedback», 2022, <https://arxiv.org/abs/2203.02155>.

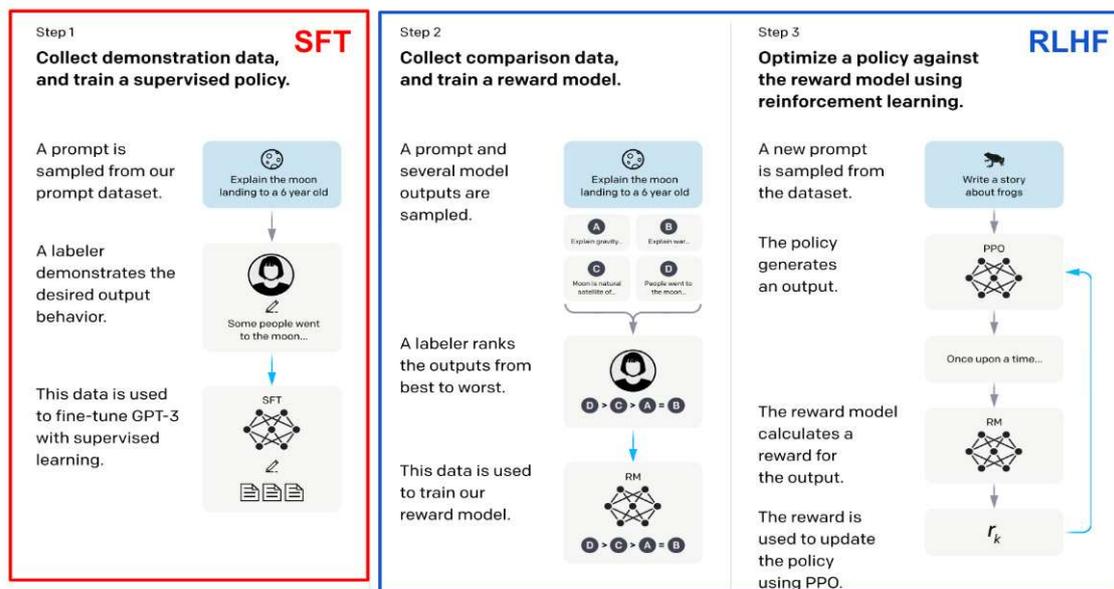


Figura 5 Diagramma che illustra le fasi di SFT e RLHF di GPT (fonte: Ouyang, Long, et al., cit.)

La fase SFT assomiglia molto all'addestramento tradizionale di una rete neurale: il modello riceve delle coppie prompt-risposta dove le risposte sono redatte da esperti umani; in genere si tratta di prompt che specificano istruzioni, come per la prima volta sperimentato nel predecessore di ChatGPT, InstructGPT, sviluppato nel 2022 da OpenAI. Queste coppie vengono poi usate per una ulteriore fase di addestramento del modello in modo sostanzialmente omologo al processo di pre-training (ma assai meno lungo e oneroso).

Il reinforcement learning è una tecnica di apprendimento automatico che consente a un modello neurale di apprendere tramite la sperimentazione e il feedback. Ci sono diverse modalità di implementare un processo di RLHF, ma in linea di massima le cose si svolgono in questo modo: al modello viene sottoposto un prompt in risposta al quale genera n diverse risposte; un esperto umano etichetta queste n risposte in base a una scala di qualità (nel contesto di ChatGPT, il modello riceve una ricompensa positiva se la risposta generata è rilevante e coerente con la conversazione in corso. Al contrario, riceve una ricompensa negativa se la risposta è errata o inappropriata); queste risposte pre-valutate sono usate per addestrare un modello di supporto detto *reward model*, che emette un valore scalare come valutazione di una sequenza linguistica; il reward model viene infine usato per valutare le risposte a una serie di prompt del modello primario, usando un algoritmo denominato *Proximal Policy Optimization*. Questo feedback può essere utilizzato per modificare i pesi del modello, migliorando la sua capacità di generare risposte coerenti e appropriate in una conversazione. Ad esempio, se il modello genera una risposta che viene valutata come poco pertinente o inappropriata, i pesi della rete neurale possono essere modificati in modo da rendere meno probabile la generazione di risposte simili in futuro.

Naturalmente i dettagli matematici e tecnici di questi processi sono al di là della portata e degli scopi di questo lavoro. Ma ciò che è importante rilevare è che queste successive fasi di fine tuning hanno un ruolo fondamentale nella formazione della capacità di interazione linguistica del modello. Esse provvedono infatti il processo di

allineamento di un LLM, ovvero la sua convergenza verso le attese cognitive e linguistiche di un utente umano, che include la capacità di essere utile e informativo, cooperativo, non offensivo. Il dibattito teorico su questi aspetti è assai ampio, e le opinioni divergenti, come più volte detto. Potremmo dire che, se la fase di self pre-training corrisponde all'apprendimento della competenza linguistica (sia sintattica sia semantica), la fase SFT fornisce al modello una competenza conoscitiva e argomentativa e la fase RLHF una competenza conversazionale e pragmatica (alla Grice). Non abbiamo spazio per esplorare le possibili conseguenze teoriche e pratiche di queste osservazioni. Ciò che è comunque evidente è che nell'insieme l'addestramento di un LLM fa sì che il modello mostri un insieme di capacità "emergenti", per le quali non è stato esplicitamente addestrato.

Ovvero, dal combinato di una vasta modellizzazione probabilistica delle lingue naturali, e del raffinamento di conoscenze di dominio e attitudini comportamentali, emerge un agente linguistico in grado di svolgere compiti universali con efficacia altissima, tanto da superare numerosi test di valutazione di competenze, spesso con risultati di livello superiore a quelli dei soggetti umani. Ma ciò che più ci interessa è che un LLM mostra di possedere una utilissima meta-capacità: quella di saper apprendere dal contesto conversazionale, ciò che fa sì che l'interazione con un modello, se opportunamente guidata da adeguate strategie argomentative, possa essere considerata una forma di apprendimento contestuale. E questo rende il modello capace di fornire prestazioni efficaci anche in ambiti e su compiti complessi. Non stiamo affermando che un LLM come GPT-4 non faccia errori o, come è invalso dire, non produca allucinazioni (ovvero risposte errate dal punto di vista fattuale, o vere e proprie affabulazioni di invenzione, che, si badi però, sono sempre linguisticamente coerenti). Ma che, usando il prompting in modo adeguato, si possa fortemente limitare l'evenienza di tali errori e far convergere il modello su risposte altamente qualificate anche in ambiti molto specialistici.

Il cosiddetto *prompt engineering* è appunto un campo di studi emergente dedicato allo studio delle strategie e delle tecniche volte a ottimizzare l'interazione con i modelli generativi linguistici e no. Ci riserviamo di esplorare questo ambito di ricerca, e le sue conseguenze teoriche e applicazioni in ambito umanistico in successivi lavori. In questa sede, sulla base di quanto appena detto, vogliamo concludere osservando come un LLM come GPT-4 o simili, proprio in virtù di questa capacità emergente di apprendere dal contesto interazionale, possa rivelarsi un formidabile supporto nell'attività di ricerca scientifica, in particolare in quella nell'ambito delle Digital Humanities.

5. ChatGPT nel *digital lab* dell'umanista digitale

Dopo avere riassunto i fondamenti teorici degli LLM, torniamo dunque a riflettere sul ruolo che essi possono assumere nelle pratiche di ricerca sperimentale che sono tipiche del campo umanistico digitale³³. Ciò che intendo sostenere è che i sistemi linguistici generativi possono fornire un valido contributo in tutte le fasi di un progetto di ricerca:

³³ Fabio Ciotti, «La svolta empirico-computazionale negli studi culturali e letterari: una nuova scienza della cultura», *Culture digitali. Intersezioni: filosofia, arti, media. Proceedings AIUCD2022*, AIUCD - Ass. per l'informatica umanistica e la cultura digitale, 2022, <https://t.co/2WzCpVb9ho>.

1. formulazione di domande di ricerca sulla base di specifici framework teorici;
2. formulazione di setting sperimentali ed esecuzione di procedimenti analitici;
3. formulazione e valutazione di ipotesi esplicative.

Tra queste fasi, GPT si mostra di particolare supporto nella fase di esplorazione nello spazio dei metodi e degli strumenti applicabili alle ipotesi di ricerca individuate dal ricercatore. Questo è dovuto al fatto che GPT si è rivelato uno strumento particolarmente versato nella prototipazione rapida di applicazioni software, che altrimenti richiederebbero molto tempo (sempre ammesso che il ricercatore abbia le necessarie competenze di sviluppo software) o il coinvolgimento nel gruppo di ricerca di studiosi di area tecnico-informatica già disponibili a partecipare al progetto stesso anche nella sua fase puramente esplorativa e aurorale. Ovviamente è ben lungi da me ritenere che, in mancanza di una adeguata conoscenza dettagliata del dominio di ricerca, delle possibilità offerte dai metodi computazionali e di una certa competenza almeno approssimativa nello sviluppo di applicazioni in un qualche linguaggio di programmazione o formalismo adeguato, sia possibile produrre alcunché di qualificato e significativo. Tuttavia, in presenza di queste condizioni i modelli conversazionali sono in grado di fornire:

1. supporto alla creazione di workflow di ricerca
2. prototipazione software per il test dei metodi individuati di aiuto nella predisposizione
3. formalizzazione dei dati e dei dataset da utilizzare nella ricerca.

Ad esempio, un ricercatore di ambito letterario che volesse applicare tecniche di *machine learning* per analizzare grandi quantità di testi potrebbe utilizzare ChatGPT per ottenere indicazioni su come strutturare il workflow di analisi dei dati, quali tecniche di analisi testuale utilizzare, come preparare i dati per l'analisi, e come interpretare i risultati dell'analisi. Inoltre, GPT può anche fornire indicazioni su come selezionare e predisporre i dati appropriati per l'analisi, (ad esempio quali fonti di dati utilizzare e come selezionare i dati in modo da limitare il rischio di *bias*). In definitiva un sistema come ChatGPT può essere considerato un *artificial research assistant* in grado di supportare studiosi umanistici con competenze di *data science* non professionali, nella fase di *operazionalizzazione* dell'analisi.

A supporto di queste tesi riporto tre casi di uso. Il primo è relativo a interazione con ChatGPT volta a definire un workflow metodologico e a predisporre il codice Python necessario all'individuazione, mediante il modello linguistico BERT, di brani testuali che siano tematicamente attinenti al tema del lavoro³⁴:

User

³⁴ Questa interazione è stata condotta dall'autore nella preparazione di una ricerca presentata al seminario "Émotions, travail et récit" (Aix-en Provence, Marzo 2023) con il titolo "The 'feeling' of labor in Italian Literature 1800-2000: a distant reading approach"

I would like you to help me design a workflow for the following task: I want to use BERT to detect passages of texts in a corpus of multilingual novels that are examples of a certain theme. For instance: I want to extract excerpts of text that are about a working activity. How could I proceed?

ChatGPT

Sure, here's a suggested workflow for using BERT to detect passages of text in a corpus of multilingual novels:

Data preparation: Collect a corpus of multilingual novels and pre-process them to remove any noise and unwanted characters. You can also tokenize the text into sentences or paragraphs for easier processing.

Fine-tune a pre-trained BERT model: Fine-tune a pre-trained BERT model on a large corpus of text that includes examples of the theme you are interested in (in this case, working activity). You can use a pre-trained multilingual BERT model to handle the multilingual aspect of your corpus. The fine-tuning process involves training the BERT model on your specific task by providing it with examples of text that are about the theme you are interested in, and then fine-tuning the model to identify passages that match that theme.

Text classification: Use the fine-tuned BERT model to classify each sentence or paragraph in your corpus as either relevant to the theme or not. You can use a threshold for the model output to determine if a given sentence or paragraph is about the theme.

Extract relevant passages: Once you have identified the relevant sentences or paragraphs, you can extract the corresponding passages of text from the original corpus. You can combine adjacent sentences or paragraphs that are relevant to create longer passages.

Post-processing: Finally, you can apply any necessary post-processing to the extracted passages, such as removing duplicates or sorting by relevance.[...]

User

How many tagged examples are needed for fine tuning?

CahtGPT

The number of tagged examples needed for fine-tuning a BERT model depends on several factors, such as the complexity of the task, the size and quality of the training data, and the performance requirements of the model. In general, the more examples you have, the better your model is likely to perform, as it has more information to learn from.

However, it's also important to keep in mind that fine-tuning a BERT model can be computationally expensive, so you may need to balance the number of training examples with the available computing resources and training time.

[...]

User

Tell me if this can work use a fixed vocabulary of work-related terms to extract 200 random passages from my corpus; use them to fine tune a Bert model; then use the Bert model fine-tuned to detect passages in the rest of the corpus

ChatGPT

Using a fixed vocabulary of work-related terms to extract random passages from your corpus can be a useful approach for creating a labeled dataset for fine-tuning a BERT model. However, there are a few potential issues with this approach:

Limited diversity of passages: By using a fixed vocabulary to extract passages, you may end up with a limited set of passages that all use similar or related terminology. This can limit the diversity of the training data and result in a model that is overly focused on specific keywords, rather than the broader theme of work-related activities.

Bias in the labeled dataset: Depending on how you select and extract the 200 passages, there may be some bias in the labeled dataset that could affect the performance of the model. For example, if the passages are all from a specific genre of novels or from a particular time period, the model may not perform as well on passages from different genres or time periods.

Difficulty in scaling: If you plan to use the fine-tuned BERT model to detect passages in the rest of the corpus, you may find that the fixed vocabulary approach does not scale well to larger or more diverse datasets. For example, there may be work-related terms or phrases that are not included in your fixed vocabulary, which could result in missed passages.

[...]

User

Can you write a python program to test my approach?

ChatGPT

Sure! Here is some sample code in Python that demonstrates your approach for extracting work-related passages and fine-tuning a BERT model: [...]

Il secondo caso di uso invece mostra come GPT sia in grado di produrre direttamente analisi approfondite di testi letterari (in questo caso l'oggetto dell'analisi è il primo capitolo de *Il turno* di Luigi Pirandello). La limitazione maggiore per questo tipo di applicazioni di GPT è la dimensione predefinita del prompt che il modello è in grado di gestire (*context window*). Nella prima versione di GPT-4 essa era fissata intorno ai 30 mila token; il nuovo modello GPT-4-Turbo, rilasciato nel novembre 2023, ha esteso questo limite, almeno in teoria, a 128 mila token, ma dalle prime sperimentazioni si evidenzia come la capacità di 'tenere in memoria' il contesto degradino quando si supera il limite

dei 73 mila token³⁵. Questo ovviamente rende piuttosto complicato lavorare su testi lunghi come un romanzo³⁶. Il caso d'uso mostra una particolare tecnica di elaborazione del prompt detta *one shot Chain of Thought* (CoT), in cui nella richiesta si specifica non solo il task da eseguire e i dati o contesto di applicazione, ma si fornisce anche un esempio di esecuzione dello stesso e una spiegazione dei passi di ragionamento necessari per eseguirlo, allo scopo di guidare il modello nella corretta esecuzione del compito.

User

I will make you read a text chunk in Italian. You must extract all passages of direct discourse.

I will show you an example:

TEXT SAMPLE: — Ragioniamo!

ANSWER: "Ragioniamo"

REASON: the bar symbol introduces a direct discourse. The direct discourse end when you meet a new bar symbol or a punctuation mark.

Here is the text chunk

TEXT: [...]

END TEXT

You must format the answer in the following way:

Direct discourse passages

1 : "<text of the direct discourse passage>"

ChatGPT

³⁵ Si vedano a su questo i test effettuati da Greg Kamradt e documentati su X/Twitter <https://twitter.com/GregKamradt/status/1722386725635580292?s=20&fbclid=IwAR0RtN6iNc0PiVHn-UoDxJPe2ni9pZywY2UnaoKP0PWnzI6jvnpjrrXBu1k>.

³⁶ In realtà la questione è abbastanza controversa, poiché questo limite sembra alquanto variabile almeno nella versione del modello che opera in ChatGPT. Si tenga conto, tuttavia, che un LLM non ha qualcosa che possa funzionare da memoria a lungo termine su cui trasferire il contenuto delle sue interazioni, e che ogni volta che viene inviato un prompt esso riceve tutta la sequenza di domande e risposte precedenti. Si capisce come sia molto facile superare il limite prefissato della *context window*. Purtroppo, la mancanza di informazioni sulla architettura del sistema non permette di avere certezze a riguardo, ma è probabile che, se il contesto eccede il limite, in fase di pre-processing le prime parti dell'interazione vengano scartate. Gli studi sperimentali, tuttavia, evidenziano come il modello tenda a ricordare con maggiore precisione anche le parti iniziali di un prompt.

Direct discourse passages

1 : "Giovane d'oro, sì sì, giovane d'oro, Pepè Alletto!"

2 : "Ragioniamo!"

3 : "Don Diego Alcozèr?"

4 : "Aspettate... Ragioniamo!"

5 : "Don Diego Alcozèr?"

6 : "Scusate tanto, credevo che foste persone ragionevoli."

7 : "Avete voluto la libertà, santo Dio! il re che regna [...]"

8 : "Ragioniamo..."

User

Now I want you to perform a second task on the same text chunk: identify passages containing reported thoughts. I will show you some examples with words introducing a reported thought.

TEXT SAMPLE: Perché lui, veramente, ci ragionava su quel partito...

ANSWER: "Perché lui, veramente, ci ragionava su quel partito, ci ragionava con la più profonda convinzione che fosse una fortuna per la figliuola. E s'era intestato di persuaderne anche gli altri, "

REASON: the pronoun "lui" refers to a character and the following verb "ragionava" introduces a passage of reported thought

TEXT SAMPLE: Gli sarebbe piaciuto maritar la figlia col consenso popolare...

ANSWER: "Gli sarebbe piaciuto maritar la figlia col consenso popolare "

REASON: the past conditional tense of the verb "gli sarebbe piaciuto" is often used in the context of reported thought or soliloquy. The verb "piacere" in Italian indicates a state of mind; verbs referring to thought, beliefs, state of minds or emotional statuses are usually used in narrative passages reporting thoughts or cognitive activity of the characters in a novel.

You must format the answer in the following way:

Reported thought passages

1 : "<text of the direct discourse passage>"

ChatGPT

Reported thought passages

1 : "il Ravì si sarebbe guardato bene dal negarlo; ma, quanto a concedergli la mano di Stellina, no via: non voleva se ne parlasse neanche per ischerzo."

2 : "Gli sarebbe piaciuto maritar la figlia col consenso popolare, come diceva; e andava in giro per la città, fermando amici e conoscenti per averne un parere."

3 : "Perché lui, veramente, ci ragionava su quel partito, ci ragionava con la più profonda convinzione che fosse una fortuna per la figliuola. E s'era intestato di persuaderne anche gli altri, quelli almeno che gli permettevano di sfogare l'exasperazione crescente di giorno in giorno."

User

Now I want you to perform a third task on the same text chunk: identify all characters and agents mentioned in the narrative. I will give you some rules to follow to identify characters.

1) A personal character is identified by a proper noun

2) A personal character is identified by pronouns

3) A character plays a role in the narrated action; a character is an agent

4) A narrative text can also have collective characters that are not identified by name or surname but play a role and have agency

You must format the answer in the following way:

Characters in the text

1 : "<character> - <reason>"

ChatGPT

Characters in the text

1 : Ravì: He is identified by a proper noun and is a central figure in the text, considering marrying his daughter to someone.

2 : Stellina: Identified by a proper noun, she is the daughter of Ravì whom he wishes to marry off.

3 : Pepè Alletto: Identified by a proper noun at the beginning of the text, he is mentioned as a "giovane d'oro" but doesn't seem to be a chosen suitor for Stellina.

4 : Don Diego Alcozèr: Identified by a proper noun, he appears to be the intended groom for Stellina, causing reactions from others.

5 : Friends and Acquaintances: These are collective characters, playing the role of the public opinion that Ravì seeks out for advice on marrying his daughter. They react in various ways to the mention of Don Diego Alcozèr.

6 : Young Men ("giovanotti"): Identified as a collective character through a common noun, they are described as lacking the qualities a "previdente" father like Ravì would want for his daughter.

7 : Old Men ("un vecchio"): This is a collective character, presented as the alternative option for marrying Stellina. They are described as preferable if they are rich.

These are the characters in the text chunk based on the rules you've provided.

Come si vede anche nei task più complessi e astratti le risposte del modello sono di buon livello, e soprattutto nel terzo task le argomentazioni sono molto solide e pertinenti, sebbene il modello abbia esteso troppo la nozione di personaggio.

Infine, voglio riportare un caso assai più complesso di uso di ChatGPT per effettuare analisi testuali, che fa uso di un plug-in sviluppato da OpenAI per ChatGPT, *Advanced data analysis*³⁷. Si tratta di una modalità di interazione specializzata per l'analisi di dati che permette all'utente di inviare data set in vari formati e di chiedere al modello di applicare su di essi procedimenti di analisi (il modello sviluppa il codice Python e lo esegue in modo autonomo), nonché di visualizzare i risultati e trarre alcune conclusioni. Nel caso di uso ho inviato al modello un file CSV che contiene l'analisi di un corpus di oltre seicento testi narrativi italiani condotta mediante il software LWIC – un analizzatore testuale che calcola le frequenze di varie classi di parole in un corpus, raggruppate per ruolo e funzione semantica, molto usato in negli studi di psicologia e sociologia quantitativa³⁸. Lo scopo dello studio (che ho già condotto in precedenza sviluppando direttamente codice e visualizzazione³⁹) è verificare se sia possibile

³⁷ Come tutti i plug-in è disponibile solo per la versione con abbonamento di ChatGPT.

³⁸ James W. Pennebaker et al., «The Development and Psychometric Properties of LIWC2015», 2015, <http://hdl.handle.net/2152/31333>.

³⁹ Fabio Ciotti, «Computational approaches to literary periodization: an experiment in Italian narrative of 19th and 20th century», *Digital Humanities 2022. Conference Abstracts*, Tokyo, DH2022 Local Organizing Committee, 2022, <https://dh2022.dhii.asia/dh2022bookofabsts.pdf>, pp. 181–183.

individuare una correlazione tra la distribuzione di frequenza delle classi di parole relative alla sfera della cognizione e dell'introspezione e la periodizzazione letteraria della narrativa italiana tra 800 e 900. In questa sede ovviamente non intendo discutere presupposti e risultati dello studio in sé, ma solo mostrare come ChatGPT abbia fornito un valido supporto a tutte le fasi di analisi quantitative e di interpretazione dei risultati. In questo caso la lunghezza dell'interazione non permette di riportare tutti le fasi della stessa, e quindi riproduco solo alcuni estratti che mostrano in modo significativo le potenzialità di questo ambiente di lavoro basata su GPT.

ChatGPT

Thank you for providing the dataset. Let's first load the data and take a look at the structure and the first few entries to understand what we are working with.

[...]

The dataset contains 31 columns. Here's a brief description of the first few rows:

Filename: This seems to contain the source of each text.

decade: The decade in which the text was written.

Other columns: These appear to be different language and semantic features, possibly derived from a text analysis tool such as LIWC (Linguistic Inquiry and Word Count). Some examples are pronomi, Io, Noi, Se, Tu, Altri, Affett, Sen_Pos, and so forth.

Before proceeding, could you provide some additional context on the columns, especially those following the decade column? What do they represent?

And, what are you hoping to achieve with the cluster analysis? For instance, are you interested in grouping texts based on these features, or is there a different goal?

User

The decade column contains the decade in which a book was written, approximately.

ChatGPT

Thank you for the clarification. So, the decade column indicates the approximate decade during which each book was written. The other columns seem to be various language and semantic features.

With respect to your cluster analysis, could you provide more details on your objective? Are you aiming to group these books based on their linguistic and semantic characteristics? If so, do you have any specific features you're particularly interested

in? Or should all the features be considered for the clustering? Your guidance will help me better tailor the analysis to your needs.

User

The overall aim of my research is checking if those text, written from 1820 to 2000, forms groups on the base of the semantic features extracted by LWIC that are period sensitive. That is to check if for instance all text before 1900 forms a cluster that is clearly distinct from the texts written after 1900

ChatGPT

I see, thank you for the clarification. In this case, we'll perform a cluster analysis on the text features in order to identify distinct groups, and then we'll examine whether these groups correlate with the period in which the texts were written.

For the cluster analysis, we can use an algorithm like K-means or hierarchical clustering, which are commonly used for this type of task.

Before we begin, we need to pre-process the data:

We'll first remove the Filename column because it's not a numerical feature.

Next, we'll standardize the remaining features so they have a mean of 0 and a standard deviation of 1. This is a common preprocessing step when performing cluster analysis since it ensures that all features contribute equally to the distance computations, regardless of their original scale.

[...]

To select the number of clusters, we can use the Elbow Method or the Silhouette Score. Both methods involve creating several models with different numbers of clusters, and selecting the number that results in the best score according to a certain criterion.

The Elbow Method involves plotting the sum of squared distances from each point to its assigned cluster center for different numbers of clusters. The "elbow" of the plot, or the point where adding more clusters doesn't significantly improve the fit, indicates an appropriate number of clusters.

The Silhouette Score measures how close each point in one cluster is to the points in the neighboring clusters. A higher score indicates that the points are well clustered.

Which method would you prefer to use?

User

Adopt the Elbow method

ChatGPT

Great, let's proceed with the Elbow Method. This method involves fitting the K-means algorithm to the data for a range of values of k (number of clusters), and for each value of k , the sum of squared distances of samples to their closest cluster center (also known as inertia) is calculated.

We'll plot these inertia values for different k 's. The appropriate number of clusters is typically at the "elbow" of the plot, i.e., the point where the inertia starts to decrease more slowly.

Let's perform this analysis.

From the Elbow Method graph, there's a noticeable "elbow" at around $k=4$. This suggests that 4 is a good number of clusters for this dataset, as adding more clusters after this point doesn't result in a significant decrease in inertia.

[...]

ChatGPT

The table above shows the mean values of all the features for each cluster. This can give us some insights into the distinguishing characteristics of the books in each cluster.

However, with 29 features, it can be challenging to discern clear patterns just by looking at this table. To get a clearer view, let's visualize how the average decade of the books in each cluster changes. This can help us understand if the clusters correspond to different periods of time as you hypothesized.

The bar plot displays the mean decade for each cluster. It appears that there is a progression of time from Cluster 1 to Cluster 0, with the books in Cluster 1 tending to be older and those in Cluster 0 being more recent.

This is a promising result, as it suggests that the semantic features extracted by LIWC may have some sensitivity to the period in which a text was written.

To get a more complete picture of how well the clusters correspond to different time periods, we could also examine the distribution of decades within each cluster. This could be done using a histogram or boxplot for each cluster. Would you like to proceed with this additional analysis?

[...]

The boxplots display the distribution of decades within each cluster. Here are some observations:

Cluster 1 primarily contains books from the earliest decades in the dataset, with a few outliers extending into the later decades.

Cluster 2 has a wide range, spanning from around 1860 to almost 2000, with a median around 1940.

Cluster 3 contains books mainly from the mid-1900s, with a few extending into the early 2000s.

Cluster 0 predominantly contains books from more recent decades, with a median around 1980.

These distributions align with the results we observed in the bar plot of mean decades for each cluster. The clusters appear to capture some aspects of the time progression in the dataset, with Cluster 1 primarily containing older books and Cluster 0 mainly containing more recent ones.

This provides some evidence supporting your hypothesis that the text features extracted by LIWC can form clusters that are period-sensitive. However, it's worth noting that there is considerable overlap in the decade distributions of the clusters, and further analysis may be needed to fully understand the nature of these clusters and their relationship to the time period.

6. Conclusioni, problemi aperti, prospettive

Gli esempi e i casi d'uso che abbiamo fornito sopra mostrano come i sistemi LLM, con tutti i loro limiti, se opportunamente e consapevolmente utilizzati sono di grande utilità nella fase di definizione e prototipazione rapida della ricerca, e permettono di automatizzare e documentare parti rilevanti del workflow (con ricadute positive anche sul piano della apertura del processo di ricerca⁴⁰). I noti problemi di “allucinazioni” – la produzione di risposte false a livello referenziale ancorché linguisticamente accettabili – e l'inesattezza nei riferimenti bibliografici e fattuali ovviamente vanno tenuti nel dovuto conto, e impediscono (per fortuna diremmo) di delegare totalmente il processo di ricerca. Tuttavia, una accurata predisposizione dei prompt, mediante quello che chiameremmo un *research oriented prompt engineering*, può ovviare almeno in parte a questi problemi: ci riferiamo in particolare all'uso di tecniche di *contextual prompting*⁴¹ e di *Chain-of-Thought*

⁴⁰ «Open Science Workflows | FOSTER», s.d., <https://www.fosteropenscience.eu/foster-taxonomy/open-science-workflows>.

⁴¹ Sandesh Swamy et al., «Contextual Dynamic Prompting for Response Generation in Task-oriented Dialog Systems», 2023, <https://arxiv.org/abs/2301.13268>.

*prompting*⁴² che sono state sviluppate e testate di recente come alternative a costose, complesse e non sempre efficaci operazioni di fine-tuning verticali, con risultati notevoli.

I modelli linguistici conversazionali possono dunque facilitare la costituzione di un *common ground* concettuale e metodologico tra ricercatori formati in domini differenti, prerequisito necessario per l'efficacia della ricerca interdisciplinare che caratterizza ogni settore delle Digital Humanities. E di conseguenza possono anche essere un fattore di innalzamento della produttività del lavoro scientifico⁴³.

Assai più controverso è l'uso degli LLM nella fase di interpretazione dei risultati di generalizzazione e di spiegazione causale dei risultati dell'analisi dei dati. In un certo senso presumere che un modello linguistico possa fornire autonomamente risultati in questi ambiti presuppone che esso sia dotato di una qualche capacità metacognitiva, mentre è ancora ampiamente dibattuto se essi siano, anche in misura molto limitata, qualcosa di più di cosiddetti *stochastic parrot*⁴⁴. Non è mia intenzione in questo contesto addentrarmi nel dibattito filosofico circa la natura "veramente" intelligente o meno, per non dire cosciente o, come si usa dire oggi, senziente, di questi modelli⁴⁵. Mi sembra che buona parte delle posizioni scettiche a riguardo siano fondamentalmente varianti moderne del dualismo cartesiano, del vecchio *Ghost in the machine* che Gilbert Ryle aveva cercato di esorcizzare filosoficamente già negli anni 50 del secolo scorso⁴⁶. A fronte delle capacità emergenti di questi modelli, ritengo, come molti, che l'atteggiamento scientificamente più appropriato sia quello di indagare con un approccio empirico e uno spirito critico ma non pregiudiziale tali capacità.

Ma anche senza concedere troppo alle facoltà cognitive di GPT e simili, vorrei far notare come, se opportunamente interrogato, un modello linguistico di complessità elevata sia in grado di fornire un valido supporto nella attività di *theory building* e *hypothesis testing*. Come ha osservato in una conversazione personale Gino Roncaglia, riflettendo su alcune sue sperimentazioni volte a usare ChatGPT come generatore di teorie filosofiche⁴⁷, è forte l'impressione che «questi modelli diano il meglio proprio nella co-generazione di ragionamenti astratti, mentre sono di sicuro poco utili come generatori di informazioni fattuali (dove il ricorso a un classico motore di ricerca è, ancora oggi, di certo più efficace).

Certo è che assegnare funzioni di valutazione e spiegazione a un modello linguistico, ancorché avanzato come GPT-4 e i suoi simili, richiederebbe una adeguata

⁴² Jason Wei et al., «Chain-of-Thought Prompting Elicits Reasoning in Large Language Models», 2023, <https://arxiv.org/abs/2201.11903>.

⁴³ Ci sono numerose analisi sull'impatto positivo degli LLM nella produttività in ambito di sviluppo software e un recente lavoro sperimentale di due ricercatori del MIT sui loro effetti nei compiti di redazionali di professionisti di medio livello mostra come "exposure to ChatGPT increases job satisfaction and self-efficacy and heightens both concern and excitement about automation technologies" Shakked Noy, Whitney Zhang, «Experimental Evidence on the Productivity Effects of Generative Artificial Intelligence», *SSRN Electronic Journal*, 2023, <https://www.ssrn.com/abstract=4375283>.

⁴⁴ E.M. Bender et al., *op.cit.*

⁴⁵ D.J. Chalmers, *op.cit.*; John Rogers Searle, *Intelligenza artificiale e pensiero umano. Filosofia per un tempo nuovo*, Angela Condello (a cura di), Roma, Castelvecchi, 2023.

⁴⁶ Gilbert Ryle, *The concept of mind*, London, Hutchinson, 1949.

⁴⁷ <https://chat.openai.com/share/675628d3-17eb-48a7-b256-85ec3ab013d0>.

soluzione al problema della esplicabilità dei risultati dei modelli stessi⁴⁸, problema al centro della ricerca sulla *explainable AI* e tutt'altro che risolto (e forse non empiricamente risolvibile per i grandi modelli). In linea di principio possiamo immaginare di chiedere al modello di spiegare tutte le ragioni delle sue scelte; il fatto è che, come studiosi e ricercatori, in un contesto di indagine scientifica non possiamo assumere le motivazioni discorsive addotte dal modello stesso come giustificazione di una spiegazione causale o di una interpretazione dei risultati di un esperimento. Almeno finché non sia più chiaro se e quanto questi modelli abbiano sostanziali capacità di ragionamento causale (non sto sostenendo che *non* ne abbiano, solo che *non sappiamo* quanta e di che tipo ne abbiano). Siamo dunque di fronte a un limite epistemologico di fronte a sistemi artificiali la cui evoluzione tecnica è stata così rapida da superare di gran lunga la nostra capacità di comprensione e spiegazione teorica. Tuttavia, è assai ragionevole pensare che sia l'evoluzione della ricerca nell'ambito stretto del *machine learning* e dell'IA, attraverso la produzione di modelli sempre più potenti ed efficienti, sia la riflessione teorica sugli stessi fornirà risposte a molte delle domande che rimangono allo stato attuale aperte e "flottanti". Per il momento possiamo comunque essere più che soddisfatti di come questi Pappagalli Stocastici possono aiutare noi, Veri Ricercatori Umani, nel progettare e condurre ottima ricerca umanistica computazionale.

⁴⁸ W. James Murdoch et al., «Definitions, methods, and applications in interpretable machine learning», *Proceedings of the National Academy of Sciences*, vol. 116, fasc. 44, ottobre 2019, pp. 22071–22080; Roberto Confalonieri et al., «A historical perspective of explainable Artificial Intelligence», *WIREs Data Mining and Knowledge Discovery*, vol. 11, fasc. 1, gennaio 2021, <https://onlinelibrary.wiley.com/doi/10.1002/widm.1391>.