digitcult @Scientific Journal on Digital Cultures

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

Parlare con le macchine. Il prompt engineering come competenza epistemica

Talking to Machines. Prompt Engineering as an Epistemic Skill

Fabio Ciotti
0000-0001-9604-5980
Università di Roma Tor Vergata
ROR 02p77k626
fabio.ciotti@uniroma2.it

10.54103/2531-5994/31801

Abstract

[It.] Questo articolo propone di considerare il prompt engineering non come repertorio di tecniche per migliorare le prestazioni dei modelli linguistici generativi, ma come competenza epistemica che definisce l'interazione comunicativa tra esseri umani e sistemi di intelligenza artificiale. Individuate nell'apprendimento contestuale e nelle *personae* finzionali generate dagli LLM le sue condizioni di possibilità, si passa a una rapida illustrazione delle tecniche classiche di prompting, dal few-shot al chain-of-thought. L'avvento dei reasoning model e delle architetture agentiche assorbe nei modelli stessi molte di queste strategie formalizzate; ciò che si trasforma, senza scomparire, è la competenza nel costruire interazioni epistemicamente controllate. Il prompt engineering viene così ricollocato entro una più ampia competenza critica, in cui formulare le domande pertinenti, verificare le fonti e valutare l'affidabilità dell'output conta più della padronanza di protocolli codificati.

Parole chiave: Ingegneria dei prompt; Modello di ragionamento; Gioco di ruolo; LLM; Apprendimento contestuale; Alfabetizzazione critica nell'IA; IA agentica.

[En.] This paper proposes to regard prompt engineering not as a repertoire of techniques for improving the performance of generative language models, but as an epistemic competence that defines the communicative interaction between humans and artificial intelligence systems. Having identified its conditions of possibility in in-context learning and in the fictional personae generated by LLMs, it turns to a brief account of the classical prompting techniques, from few-shot to chain-of-thought. The advent of reasoning models and agentic architectures absorbs many of these formalized strategies into the models themselves; what is transformed, without disappearing, is the competence required to construct epistemically controlled interactions. Prompt engineering is thereby repositioned within a broader critical competence, in which asking pertinent questions, verifying sources and assessing the reliability of the output matter more than proficiency in codified protocols.

Keywords: Prompt engineering; Reasoning model; Role-play; LLM; In-context learning; Critical AI literacy; Agentic AI.

Received: 01/06/2026
Accepted: 16/07/2026
Published: 06/07/26

Milano University Press

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

1. Introduzione: le macchine conversazionali

L'introduzione dei modelli linguistici basati sui transformer alla fine del decennio scorso non è passata inosservata nella comunità del machine learning e del Natural Language Processing, ma modelli come BERT (Devlin et al., 2019) non hanno attirato l'attenzione della comunità accademica in generale, e ancor meno quella dell'opinione pubblica. Si trattava di una tecnologia interessante, destinata a potenziare le tecnologie di ricerca esistenti o le attività di NLP in ambiti specialistici, ma nessuno la considerava un'innovazione rivoluzionaria. Tuttavia, già dal 2018 OpenAI ha iniziato a esplorare una variante di tale architettura che adottava un approccio autoregressivo e permetteva al suo modello, battezzato *Generative Pre-trained Transformer* 1 (GPT-1) di generare testo a partire da un input linguistico[1]. Ma è solo con il rilascio di GPT-3.5 nel 2022 che la giovane azienda californiana definisce un innovativo flusso di addestramento del suo modello linguistico, potenziando la fase di adattamento (*fine-tuning*) attraverso una fase di *Instruction Tuning* (un apprendimento supervisionato basato su domande e risposte) e una di *Reinforcement Learning* (Ouyang et al., 2022), e crea qualcosa di totalmente diverso: una macchina conversazionale. L'impatto dello sviluppo di GPT-3.5 e del rilascio della prima interfaccia pubblica di ChatGPT ha determinato una reazione di enorme impatto sia nella comunità scientifica, sia nel dibattito pubblico, riportando la nozione di Intelligenza Artificiale al centro della discussione collettiva. Le implicazioni e le contraddizioni di questa innovazione sono così ampie, profonde e stratificate su molteplici livelli discorsivi, che non è possibile darne qui conto. In questa sede, desidero concentrarmi su una conseguenza dei cambiamenti architetturali introdotti per la prima volta in ChatGPT e poi diventati comuni in tutti i modelli linguistici successivi che hanno adottato strategie simili: l'emergenza del *prompt engineering* come strategia di interazione uomo-macchina (intelligente).

Fino all'introduzione degli LLM, interagire con le macchine di calcolo richiedeva il possesso di una profonda competenza tecnica oppure il ricorso a interfacce, solitamente visive e orientate all'utente finale, che nascondevano le complessità, ma anche la potenza delle architetture sottostanti. Interfacce linguistiche fondate su rudimentali predecessori degli LLM erano disponibili in alcuni settori verticali, ma la loro efficacia era piuttosto limitata, per usare un eufemismo. In generale, le precedenti tecnologie digitali, dai motori di ricerca alle interfacce grafiche, presupponevano un paradigma discreto di interrogazione basato sulla coppia domanda-risposta. I LLM sono stati la prima tecnologia di calcolo a risolvere in un colpo solo molti dei colli di bottiglia che limitavano le tecnologie di NLP, sia nella ricerca pura sia nell'applicazione alle interfacce uomo-macchina.

[1] Un modello è detto autoregressivo quando il processo di generazione avviene per passi successivi, in cui l'output di una fase di generazione viene reimmesso nella rete neurale fino al raggiungimento di una condizione di stop. La presentazione dei dettagli tecnici del funzionamento di una rete neurale a transformer esula dal presente lavoro, si rimanda ad Alammar & Grootendorst (2024) per una introduzione tecnica rigorosa ma comprensibile, e a Roncaglia (2023) per una spiegazione più discorsiva e indirizzata al lettore con un background umanistico.

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

Le capacità di questi sistemi, inoltre, sono andate ben oltre le aspettative dei loro creatori, e la gamma di compiti che gli LLM sono in grado di svolgere si è costantemente ampliata in profondità e in estensione. Tutto questo è stato e continua a essere realizzato semplicemente chiedendo, spiegando e parlando con le macchine. È così cresciuta una nuova area di esplorazione empirica e di ricerca, quella che potremmo definire l'arte e il mestiere di parlare alle macchine, che è stata chiamata prompt engineering. Nei primi anni della diffusione dei modelli linguistici generativi, il prompt engineering veniva spesso descritto con una retorica quasi ingegneristica, come se l'interazione con un modello linguistico di grandi dimensioni (LLM) dipendesse soprattutto dalla scoperta di formule, trucchi o strutture linguistiche in grado di «sbloccare» determinate capacità latenti del sistema. Questa rappresentazione non è del tutto infondata, poiché alcune configurazioni testuali producono effettivamente effetti misurabili sull'output, e tuttavia è limitativa, tanto più quanto gli stessi modelli si evolvono e acquisiscono capacità. Appare ormai chiaro, infatti, che il prompting è una pratica comunicativa, epistemica e culturale, in cui l'utente non si limita a impartire comandi, ma instaura progressivamente un contesto di cooperazione interpretativa con il modello.

Sin dall'inizio, e sempre più con lo sviluppo dei sistemi LLM dal 2022 a oggi, è apparso evidente che ciò che è stato definito «prompt engineering» dovrebbe essere compreso e ricollocato all'interno di una più ampia competenza critica nell'interazione con l'intelligenza artificiale generativa. Per adottare questa prospettiva, tuttavia, occorre chiarire quali proprietà dei modelli linguistici di grandi dimensioni rendano possibile tale forma di interazione, e in che senso un prompt sia qualcosa di diverso da una semplice query. Il punto è mostrare che la trasformazione attualmente in corso non è soltanto una questione di esperienza d'uso, ma una riconfigurazione dell'interfaccia computazionale, in cui il linguaggio naturale diventa al tempo stesso lo spazio dell'input e quello dell'output, e in cui le proprietà della risposta dipendono in modi non banali dalla forma, dalla struttura e dal contenuto di ciò che la precede.

2. Le condizioni di possibilità: apprendimento contestuale e sensibilità al contesto

I modelli linguistici di grandi dimensioni, dovrebbe essere ormai chiaro, non sono semplici archivi da interrogare tramite parole chiave, o linguaggi di interrogazione formalizzati, né sistemi simbolici che applicano regole esplicite e predeterminate. Essi rispondono in modo sensibile al contesto, modulando il proprio comportamento in base alle istruzioni, agli esempi, ai vincoli pragmatici e alle configurazioni discorsive presenti nella conversazione. La condizione che rende possibile questa sensibilità, e con essa l'esistenza stessa del prompt engineering come pratica, è il fenomeno dell'apprendimento contestuale.

Il primo articolo che ha tematizzato questa capacità degli LLM è il lavoro di Brown et al. (2020), in cui gli autori presentavano GPT-3, il primo modello autoregressivo che, con i suoi centosettantacinque miliardi di parametri, si qualificava propriamente come

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

un Large Language Model. Nel paper il team di OpenAI mostrava come un sistema di tale dimensione potesse svolgere un'ampia varietà di compiti verticali senza richiedere di adottare complesse e computazionalmente onerose procedure di fine tuning dei parametri del modello stesso. La messa a punto del modello per uno specifico compito, infatti, avveniva semplicemente specificando nel contesto di input alcune istruzioni linguistiche, possibilmente seguite da un numero limitato di esempi del compito richiesto. In particolare, Brown et al. individuavano tre possibili strategie di interazione con il modello al fine di elicitarne i comportamenti desiderati: un approccio *zero-shot*, che consiste nella sola descrizione in linguaggio naturale del compito; uno *one-shot*, che richiedeva l'aggiunta di un singolo esempio; e uno *few-shot*, con un numero variabile di esempi risolti del task, specificati entro la finestra di contesto. L'adattamento osservabile del modello avviene durante la fase di esecuzione, o inferenza, attraverso stati interni e attivazioni condizionati dal contesto, non mediante aggiornamenti persistenti dei parametri come nel fine-tuning.

Questa scoperta ha due conseguenze: in primo luogo, il prompt non è semplicemente una query, ma un insieme addizionale di dati, indicazioni informali di comportamento e di informazioni di contesto che hanno una funzione di orientamento della risposta. In secondo luogo, la scelta degli esempi, nella loro qualità, ordine e copertura, diventa un artefatto sperimentale con effetti misurabili sulle prestazioni. Risultati successivi hanno mostrato, ad esempio, una marcata sensibilità all'ordine degli esempi (Lu et al., 2022) e casi in cui la presenza di esempi corretti incide poco (Min et al., 2022), il che suggerisce che il prompt possa anche introdurre bias o rumore e non fungere uniformemente da ausilio o estensione della conoscenza del modello.

Le spiegazioni teoriche dell'apprendimento dal contesto rimangono oggetto di un dibattito attivo e si articolano in almeno tre filoni complementari. Un primo filone, sviluppato da von Oswald et al. (2023) e, in parallelo, da Dai et al. (2023), interpreta il fenomeno come una forma di discesa del gradiente implicita: il primo lavoro dimostra, in contesti di regressione controllata, l'equivalenza formale tra le trasformazioni indotte da un livello di *self-attention* lineare e quelle prodotte da un passo di ottimizzazione, mentre il secondo estende l'ipotesi ai modelli linguistici pre-addestrati, descrivendoli come meta-ottimizzatori che eseguono un fine-tuning implicito sugli esempi forniti. Un secondo filone, riconducibile a Xie et al. (2022), modella l'apprendimento nel contesto come inferenza bayesiana implicita, in cui il modello, avendo appreso durante il pre-addestramento una distribuzione su una pluralità di concetti latenti, utilizza gli esempi nel contesto per identificare quello più probabile. Un terzo filone, che risale a Min et al. (2022) ed è stato precisato da Pan et al. (2023), lo interpreta invece come una forma di recupero dalla memoria parametrica: gli esempi non insegnerebbero al modello un compito nuovo, ma gli consentirebbero di riconoscere e attivare capacità già acquisite in fase di pre-addestramento, come suggerisce l'osservazione che la sostituzione delle etichette corrette con etichette casuali degrada solo marginalmente le prestazioni. Queste letture non sono necessariamente incompatibili, e la ricerca recente tende a trattarle come descrizioni di meccanismi che operano a diversi livelli dell'architettura. Il punto rilevante è che, senza l'apprendimento contestuale, il prompt condizionerebbe l'output

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

soltanto come contesto statistico da completare, senza poter funzionare come specificazione di un compito: il modello proseguirebbe il testo secondo le regolarità apprese, ma non potrebbe estrarre dagli esempi e dalle istruzioni una struttura di compito cui adattare il proprio comportamento, e la pratica del prompting, che su questa capacità interamente si regge, non avrebbe alcun fondamento. Infine, occorre ricordare che l'evoluzione dei workflow di addestramento dei modelli ha modificato la natura dell'interazione, poiché oggi l'utente si rivolge a modelli già orientati al rispetto delle istruzioni e delle norme conversazionali, in un regime in cui il puro apprendimento nel contesto descritto da Brown et al. è mediato e modificato da complessi interventi di allineamento, in particolare mediante le varie tecniche di Reinforcement Learning.

Un passo teorico decisivo riguarda la nozione stessa di contesto. È utile distinguere tra la conoscenza incorporata nel modello come distribuzione dei parametri della rete neurale, che deriva dal processo di addestramento, e la conoscenza che viene attivata, cioè resa operativa nell'interazione specifica. Il contesto non si limita ad aggiungere informazioni esterne a un sistema già dato; esso orienta il modo in cui il modello seleziona, combina e rende pertinente il proprio spazio di possibilità. In questo senso, il prompt engineering può essere inteso come una tecnica per la costruzione del contesto: la capacità di organizzare un ambiente conversazionale condiviso in cui il modello possa produrre risposte coerenti con uno scopo, un pubblico, una disciplina, un genere discorsivo e un insieme di vincoli epistemici.

La possibilità di tale condizionamento, nella forma assunta dagli attuali LLM, affonda le sue radici nell'architettura transformer introdotta da Vaswani et al. (2017) e, in particolare, nel meccanismo di self-attention, che consente a ciascun token di una sequenza di «prestare attenzione» a tutti gli altri (o più specificamente nei transformer dei modelli autoregressivi, o *decoder-only*, a tutti quelli precedenti), stabilendo dipendenze a lungo raggio. Questa proprietà è cruciale, poiché gli esempi e le istruzioni forniti nel prompt possono trovarsi a grande distanza dall'output richiesto, e il modello può quindi trattare una lunga sequenza come una memoria computazionale in cui esempi e istruzioni vengono codificati e richiamati dall'attenzione. La finestra di contesto, ovvero la lunghezza massima della sequenza elaborabile in una singola inferenza, è quindi un parametro architettonico di primaria importanza. La sua progressiva estensione, dalle poche migliaia di token dei primi modelli ai limiti molto più ampi dei sistemi recenti (fino a 2 milioni di token), ha reso praticabili strategie quali l'inserimento di interi documenti nel contesto, la specificazione di istruzioni dettagliate e il mantenimento di conversazioni prolungate[2].

[2] Si deve precisare che gli attuali LLM sono tecnicamente privi di memoria di stato tra un'esecuzione e la successiva, e che quindi l'intera storia della conversazione deve essere reimmessa in input a ogni turno affinché il modello mantenga una coerenza conversazionale almeno locale. Ne consegue che anche modelli dotati di finestre di contesto molto ampie possono esaurire rapidamente il contesto disponibile, sia nelle conversazioni molto lunghe sia, a maggior ragione, nei modelli reasoning (cfr. infra) e nelle catene agentiche, dove la generazione di grandi quantità di token intermedi, spesso non visibili all'utente, erode lo spazio a disposizione. Per questo i modelli più avanzati, quando sono inseriti in ambienti chat o in contesti agentivi, vengono oggi corredati di tecniche di supporto, o di *harness*, che li dotano di una sorta di memoria a lungo termine adattiva, dalla compattazione periodica del contesto ai sistemi di memoria persistente esterna.

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

3. Personaggi nella macchina

Una delle proprietà fondamentali degli LLM che ne spiega la capacità di reagire in modo differenziato ai prompt conversazionali degli utenti è la stabilizzazione, nel corso dell'addestramento, di regolarità discorsive associate a voci, ruoli e registri riconoscibili, che è plausibile interpretare come tratti di personalità impliciti nel modello, dove il termine "personalità" va inteso in senso funzionale e narratologico, come maschera enunciativa, o ruolo, che il modello può assumere e modulare, e non in senso psicologico, come attribuzione di stati mentali o di un'identità soggettiva. Poiché il corpus di addestramento registra una vasta varietà di voci, registri e ruoli, il modello apprende le regolarità statistiche associate a ciascuno di essi, e può quindi riprodurle quando il contesto le rende salienti; questo meccanismo, discusso in termini teorici da Shanahan et al. (2023), spiega l'efficacia di un intero filone di strategie, quelle fondate sul *role prompting*. La costruzione di una persona non dovrebbe essere intesa ingenuamente come l'attribuzione di un'identità reale al modello, bensì come una configurazione pragmatica dell'interazione. Chiedere al sistema di agire come un esperto, un revisore, un consulente, un critico, un tutor o un interlocutore socratico significa imporre un quadro interpretativo che orienta la selezione della conoscenza, il tono, il grado di cautela, la struttura argomentativa e il tipo di inferenze considerate pertinenti.

Il riferimento scientifico più rigoroso per questa riformulazione è il lavoro di Shanahan et al. (2023), che propone il concetto di gioco di ruolo come chiave interpretativa centrale, sviluppando un'intuizione già presente in contributi informali ma influenti, come il saggio di Janus (2022) sulla nozione di simulatore. L'argomentazione di Shanahan et al. muove da una posizione deflazionista sulle proprietà cognitive dei modelli linguistici di larga scala: gli esseri umani acquisiscono la competenza linguistica attraverso l'interazione incarnata e la socializzazione all'interno di una comunità, mentre gli LLM sono addestrati alla previsione del token successivo; pertanto, l'attribuzione a tali modelli di stati mentali quali credenze, desideri o intenzioni nel senso comune della psicologia ingenua costituisce un errore di categoria. Ciononostante, i LLM, nel modellizzare il contenuto di un'enorme quantità di documenti scritti da esseri umani, individuano un'ampia serie di indizi sul modo in cui le persone si comportano diversamente a seconda della loro personalità, il che orienta la loro produzione linguistica. Questo spiega perché il ricorso al framing teorico del *role playing* ci consenta di utilizzare il vocabolario della psicologia ingenua, per dire che il modello "crede" o "mente", in un registro dichiaratamente figurato, preservandone l'utilità descrittiva senza vincolarci all'attribuzione di autentici stati mentali. L'analogia che gli autori propongono non è quella di un attore in un'opera teatrale con copione, ma quella dell'interprete improvvisatore, che costruisce il personaggio in tempo reale e che, proprio per questo, non interpreta un unico ruolo stabile, ma genera una distribuzione di personaggi possibili che il dialogo affina progressivamente.

Il riferimento alla modellizzazione delle personalità permette quindi di spostare la discussione dal piano dell'antropomorfizzazione ingenua a quello più preciso della simulazione di posizioni discorsive. L'LLM non possiede una personalità individuale nel

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

senso psicologico del termine, eppure è in grado di stabilizzare temporaneamente un profilo conversazionale che produce effetti rilevanti sulla qualità dell'interazione. Il collegamento con il prompting è ovvio: se il modello genera una distribuzione di possibili personaggi, il prompt funge da meccanismo che ne restringe la distribuzione, cosicché un prompt di sistema dettagliato riduce l'incertezza sul tipo di profilo simulato, mentre un prompt vago produce un profilo meno definito e quindi output più variabili[3].

Questa lettura, di impianto dichiaratamente concettuale, ha ricevuto negli anni successivi una convergente corroborazione empirica da parte della ricerca di interpretabilità meccanicistica, che, pur muovendo da interrogativi diversi e in larga parte autonomi, è approdata a risultati che ne rafforzano l'impianto. Lavorando sulla decomposizione delle attivazioni interne dei modelli tramite *sparse autoencoder*, una tecnica di *dictionary learning* che scompone lo spazio sovrapposto dei neuroni in un repertorio più ampio di feature tendenzialmente monosemantiche (Templeton et al., 2024), il gruppo di Anthropic ha mostrato che è possibile isolare nello spazio delle attivazioni direzioni lineari che corrispondono a tratti di carattere riconoscibili, come la malevolenza, l'adulazione o la propensione all'allucinazione, e che tali direzioni, denominate *persona vector*, non solo registrano ma controllano causalmente il comportamento del modello, poiché iniettandole artificialmente tramite steering si induce l'emergere del tratto corrispondente, mentre gli stessi sparse autoencoder permettono di scomporle in feature ancora più fini (R. Chen et al., 2025). Il dato che più interessa la nostra argomentazione è che queste direzioni si attivano prima della risposta, anticipando la personalità che il modello sta per assumere, in coerenza con la tesi del role playing quando descrive il prompt come un meccanismo che restringe la distribuzione dei personaggi possibili. La convergenza si è poi fatta esplicita nel cosiddetto persona selection model (Marks, Lindsey, & Olah, 2026), secondo cui il modello, nel simulare il personaggio dell'Assistant, seleziona e rifinisce personae e ruoli appresi durante il pre-addestramento, al punto che gli autori descrivono l'esecuzione dell'Assistant precedente al post-addestramento come un puro role play, riprendendo così, in chiave meccanicistica e a partire da evidenze sperimentali indipendenti, la medesima intuizione che Shanahan e colleghi avevano formulato su basi teoriche. Resta inteso, e gli stessi autori lo sottolineano, che si tratta di modelli esplicativi parziali e ancora in discussione, ma la loro rilevanza per la nostra argomentazione è che spostano la nozione di personalità dei modelli linguistici dal piano della metafora utile a quello dell'oggetto empiricamente individuabile e verificabile.

[3] La formulazione cauta che ho adottato non implica una chiusura sulla questione filosofica dello statuto cognitivo di questi profili. Chalmers (2026) ha recentemente argomentato che gli interlocutori artificiali con cui interagiamo sono quantomeno quasi-agenti, dotati di quasi-credenze e quasi-desideri, e che una versione sofisticata dell'interpretivismo di matrice dennettiana può trattare le singole personae come quasi-agenti distinti, realizzati dal modello e legati ai thread conversazionali, piuttosto che come mere finzioni prive di qualsiasi spessore cognitivo. La distinzione netta tra simulatore e simulacro, assunta dai framework del role play, andrebbe in questa prospettiva ammorbidita: che il modello simuli una posizione discorsiva non esclude che, nel simularla, la realizzi in un senso funzionalmente robusto. Ritengo questa apertura compatibile con la linea qui difesa: negare la natura personale in senso psicologico forte non equivale a negare che i profili conversazionali stabilizzati possiedano una realtà funzionale sufficiente a fondare attribuzioni intenzionali metodologicamente controllate. Lo stesso Shanahan (2024) ha di recente contrastato le interpretazioni più radicalmente deflazioniste sulla natura cognitiva degli LLM.

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

4. Le tecniche classiche di prompt engineering

Sulla base dell'apprendimento dal contesto degli LLM, si è sviluppato un repertorio di tecniche che ha svolto un ruolo importante nella prima fase della diffusione dei modelli di linguaggio, poiché ha dimostrato che le prestazioni del modello dipendono non solo dalle sue capacità latenti, ma anche dal modo in cui il compito viene formulato, contestualizzato e vincolato.

I primi approcci di prompting a essere formalizzati sono i cosiddetti prompt *zero-shot* e *few-shot* (Brown et al., 2020), che si distinguono per il numero di esempi risolti forniti nel contesto. Nel prompting zero-shot il compito è specificato unicamente mediante un'istruzione o una domanda in linguaggio naturale; nel prompting few-shot l'istruzione è accompagnata da un piccolo numero di esempi, tipicamente coppie di input e output che mostrano il compito richiesto, affinché il modello ne colga il formato e la logica sottostante e li applichi, per apprendimento contestuale, a un caso non visto. Nella progettazione dei prompt few-shot entrano in gioco almeno quattro variabili: la selezione degli esempi, il loro ordine, la calibrazione necessaria a evitare che si inducano scorciatoie o distorsioni e il formato. Tali variabili non sono fissate, poiché le migliori strategie di composizione dipendono dal modello, dalla finestra di contesto e dal regime di decodifica. La selezione degli esempi è di fondamentale importanza, ovviamente, e Liu et al. (2022) hanno dimostrato che la scelta di esempi semanticamente vicini produce prestazioni migliori rispetto alla selezione casuale, mentre Lu et al. (2022) hanno documentato variazioni considerevoli nelle prestazioni derivanti dal solo ordine.

La tecnica più influente elaborata nella ricerca sul prompt engineering è, senza dubbio, il *chain-of-thought prompting* (CoT), introdotto da Wei et al. (2022). L'idea fondamentale alla base di questa tecnica è includere negli esempi, oltre alle coppie input-output, anche i passaggi di ragionamento intermedi. Gli autori hanno dimostrato miglioramenti significativi nei compiti di ragionamento aritmetico, simbolico e di senso comune, con guadagni tanto più ampi quanto maggiore è la scala del modello. A seguito dello studio iniziale sono emerse diverse varianti. La prima è stata la CoT zero-shot di Kojima et al. (2022), che induce un ragionamento passo dopo passo mediante la semplice aggiunta di una formula come «pensiamo passo dopo passo». La seconda è stata la *self-consistency* di Wang et al. (2023), che campiona più catene e seleziona la risposta più frequente. La terza è stata il *tree of thoughts* di Yao et al. (2023a), che consente di esplorare percorsi alternativi con la possibilità di tornare indietro, una strategia che riprende l'architettura del *branch and bound* nella ricerca operativa combinatoria, sostituendo però al bound esatto una funzione di valutazione euristica implementata dal modello medesimo.

L'interpretazione funzionale più parsimoniosa che spiega l'efficacia della CoT è che la catena di ragionamento agisca come un'allocazione di capacità computazionale, poiché il modello è spinto a generare stati intermedi che riducono l'entropia dell'output finale. Feng et al. (2023) hanno formalizzato questa intuizione, dimostrando che i transformer dotati di CoT possono risolvere problemi che restano fuori dalla portata di un modello privo di una catena esplicita.

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

L'efficacia di queste tecniche, tuttavia, dovrebbe essere valutata con cautela, poiché non esiste una grammatica universale del prompt perfetto, quanto piuttosto pratiche situate, dipendenti dal modello, dal dominio, dal tipo di compito e dal grado di controllo epistemico esercitato dall'utente. La ricerca empirica successiva, ad esempio, ha mostrato due limiti delle strategie di prompting basate sull'elicitazione della chain-of-thought: il primo riguarda la dipendenza dal dominio della crescita di prestazioni, documentato dalla meta-analisi di Sprague et al. (2025), che mostra come i benefici della catena di ragionamento si concentrino su compiti matematici e simbolici e siano marginali o assenti nella classificazione, nell'estrazione di informazioni e nella generazione creativa. Il secondo, più insidioso, è l'inaffidabilità epistemica della catena prodotta: Turpin et al. (2023) hanno fornito prove del fatto che le catene di ragionamento generate dai modelli non sono sempre fedeli ai processi interni che producono la risposta, cosicché un modello può presentare una catena di ragionamento apparentemente coerente che non corrisponde al processo effettivamente implementato dalla rete neurale. Questa inaffidabilità, secondo alcune ricerche come quella di Y. Chen et al. (2025) e quella, più radicalmente critica, di Kambhampati (2024), si riscontra anche nei modelli di reasoning che hanno fatto leva sull'idea originale del chain-of-thought, inserendolo direttamente nei sistemi in fase di training; e si tratta di una questione critica, poiché l'uso delle tracce di ragionamento è stato indicato come una possibile fonte per la spiegabilità degli output dei modelli e dunque per la fiducia che un utente o un'intera organizzazione possono riporre in essi.

A questo repertorio delle tecniche di prompting occorre aggiungere una considerazione sul role-prompting, che la pubblicistica ha spesso presentato come una leva quasi magica. Le prove empiriche sono, infatti, contrastanti. Da un lato, Kong et al. (2023) hanno riscontrato che l'assegnazione di ruoli specifici può migliorare le prestazioni in compiti che richiedono conoscenze specialistiche; dall'altro, studi sistematici come quello di Wang, Mao et al. (2024), di Zheng, Pei et al. (2024) e soprattutto il lavoro di Meincke et al. (2025), dal titolo significativo *Prompt Engineering Is Complicated and Contingent*, hanno dimostrato che l'effetto del role-prompting è altamente variabile, dipende dal modello e dal compito, e in alcuni contesti è nullo o addirittura negativo. Una spiegazione plausibile, che propongo qui come ipotesi e non come fatto accertato, è che l'assegnazione del ruolo funzioni nella misura in cui il corpus di addestramento contenga testi sufficienti associati a quel ruolo specifico, e nella misura in cui il ruolo sia correlato alle proprietà dell'output desiderato. Se il ruolo è generico, il condizionamento è debole; se è specifico e informativo, può orientare la distribuzione verso un sottoinsieme del corpus che è realmente pertinente. Ne consegue che la competenza dell'utente nel dominio, ovvero la capacità di sapere quale ruolo assegnare e perché, è più importante della tecnica di prompting in sé.

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

5. I reasoning model e il mutato status del prompt

Una svolta nell'evoluzione dei modelli linguistici di grandi dimensioni che ha inciso profondamente sulle varie tattiche di prompt engineering è stata l'emergere dei cosiddetti modelli di reasoning. A partire dal rilascio di o1 da parte di OpenAI nel settembre 2024, seguito da o3 e da sistemi analoghi sviluppati da altri laboratori, ha preso forma una classe di modelli che implementa il ragionamento come capacità nativa, attraverso ciò che la documentazione tecnica definisce *test-time computation*, ossia il calcolo in fase di inferenza. A differenza del chain-of-thought elicitato tramite prompt, in cui è l'utente a richiedere esplicitamente un ragionamento passo dopo passo, i reasoning model nativi generano autonomamente catene estese di ragionamento prima di produrre l'output finale, catene tipicamente invisibili all'utente o solo parzialmente visibili in modalità dedicate. Il meccanismo sottostante, sebbene non completamente documentato, sembra combinare l'apprendimento per rinforzo applicato al ragionamento e la ricerca nello spazio delle soluzioni durante l'inferenza, barattando il tempo di calcolo con la qualità dell'output.

Questi sistemi sembrano incorporare, almeno in parte, strategie di "ragionamento" che in precedenza dovevano essere esplicitamente sollecitate dall'utente, dalla scomposizione del problema all'esplorazione delle alternative, dal controllo intermedio alla pianificazione di risposte complesse. In questo quadro, molte tecniche storiche di prompt engineering perdono centralità. Il chain-of-thought, da tecnica manuale imposta dall'esterno, tende a diventare una procedura interna, o comunque una modalità già incorporata nel modo di utilizzo del modello. Il dato più interessante, a questo proposito, è che la documentazione ufficiale di OpenAI per i reasoning model sconsiglia esplicitamente l'applicazione delle tecniche di chain-of-thought prompting, poiché, eseguendo questi modelli il ragionamento internamente, l'invito a «ragionare passo dopo passo» è ridondante e può interferire con il processo interno e degradare le prestazioni (OpenAI, 2025): una tecnica che era stata la scoperta più importante del settore diventa così controproducente con la generazione successiva di modelli. Questo suggerimento dovrebbe indurre alla cautela nei confronti di qualsiasi codificazione rigida delle migliori pratiche di interazione con i modelli, specialmente quelli di frontiera, poiché le tecniche di prompting dipendono non soltanto dal contesto di applicazione, ma anche, e soprattutto, dalla generazione tecnologica cui il modello appartiene, in un settore in cui le evoluzioni procedono ancora a ritmo molto sostenuto.

Ciò non significa che il prompt engineering scompaia, ma che il suo valore non consiste più principalmente nell'attivare capacità nascoste attraverso formule speciali, quanto nel governare la relazione tra compito, contesto, criterio di qualità e valutazione dell'output. L'attenzione del prompting si sposta dalla tecnica, cioè da come indurre il ragionamento, al contenuto, cioè a cosa chiedere al modello di elaborare e a quali vincoli imporre al risultato. Cambia, anche, il rapporto tra controllo e autonomia, poiché con i modelli precedenti l'utente poteva almeno in principio esercitare un controllo sul processo attraverso la struttura del prompt, mentre con i reasoning model nativi il percorso è in parte opaco e autonomo, e l'utente ne specifica l'obiettivo senza governarne

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

i passaggi. Questo cambiamento richiede inoltre una riflessione epistemologica sulla fiducia. La trasparenza non si ottiene semplicemente con la presenza di una catena di ragionamento nel modello, come mostrato da Turpin et al. (2023) e successivamente corroborato da Barez et al. (2025) nel loro saggio *Chain-of-Thought Is Not Explainability*. Ciò è dovuto al fatto che una catena di ragionamento plausibile potrebbe non riflettere accuratamente il processo interno, portando potenzialmente a una sopravvalutazione della fiducia negli output che sono, di fatto, inaffidabili. Di conseguenza, la catena di ragionamento non dovrebbe essere considerata come una finestra trasparente sulla "mente razionale" del modello, quanto piuttosto come un output testuale, utile a fini di debug e di verifica. La sua affidabilità deve essere valutata con metodi indipendenti, come abbiamo già osservato sopra sulla scorta di Y. Chen et al. (2025).

6. La frontiera dell'agentività: dalla delega alla governance dell'interazione

Il percorso evolutivo dei sistemi di IA generativa culmina in un'ulteriore trasformazione, attualmente in corso, i cui effetti restano per più versi poco chiari, ma che non va sottovalutata, soprattutto in compiti altamente specialistici come l'ingegneria del software, che non si limita alla sola codifica. Essa merita quindi un trattamento a sé stante, anche se esula dal nucleo argomentativo di queste pagine. Il passaggio dal prompting all'agentività rappresenta lo sviluppo concettualmente più significativo nella storia recente dell'interazione tra l'uomo e gli LLM. Nella sua formulazione più generale, e seguendo la rassegna sistematica di Wang, Ma, Feng et al. (2024), un agente basato su un LLM può essere definito come un sistema in cui il modello opera come controllore di un ciclo: definizione dello scopo → osservazione → pianificazione → azione → verifica: riceve un obiettivo, elabora un piano, esegue azioni tramite strumenti esterni, osserva i risultati e ripete il processo fino al raggiungimento dell'obiettivo o al soddisfacimento di un criterio di arresto. Il paradigma che ha reso operativa questa integrazione tra ragionamento e azione è stato formalizzato da Yao, Zhao, Yu et al. (2023b), che hanno mostrato come l'alternanza tra tracce di ragionamento e azioni consenta al modello di pianificare, monitorare e correggere il proprio operato, interagendo con fonti esterne. L'infrastruttura tecnica che ha reso possibile questa evoluzione è la chiamata di funzioni, introdotta nel 2023 e rapidamente adottata dai principali fornitori, che consente al modello di inviare chiamate strutturate a funzioni esterne, ricevendo i risultati come input per la generazione successiva. Di conseguenza, l'LLM si trasforma da generatore di testo in un orchestratore di flussi di lavoro computazionali.

Dal punto di vista dell'interazione, l'introduzione dei sistemi agenti sposta il focus dall'emissione di comandi alla definizione di un mandato, ovvero degli obiettivi, dei vincoli e dei criteri di accettazione entro i quali l'agente opera in modo autonomo. Nei contesti di sviluppo software in cui queste architetture sono più avanzate, le istruzioni più efficaci non sono più richieste passo dopo passo, bensì specifiche di un ciclo operativo. La documentazione delle piattaforme agenti descrive in modo esplicito cicli quali l'acquisizione del

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

contesto, l'azione e la verifica, e raccomanda di fornire all'agente criteri verificabili, quali test o vincoli, affinché possa convalidarsi autonomamente. Il prompt, pertanto, diventa una componente integrante di una specifica che comprende politiche, vincoli, criteri di verifica e procedure di escalation. Anche in questo caso, le enormi potenzialità di questa innovazione sono evidenti, ma non mancano i limiti e le conseguenze problematiche: poiché gli agenti possono operare in modo autonomo in un ambiente di produzione reale, la delega dei compiti introduce nuovi rischi, tra cui la possibilità che l'agente segua percorsi subottimali, propaghi errori lungo il ciclo o esegua azioni con conseguenze irreversibili. La letteratura sugli agenti, sintetizzata da Wang, Ma, et al. (2024) e da Huang, Liu et al. (2024), documenta diversi modi di fallimento, dal looping alle deviazioni dall'obiettivo iniziale, dalle azioni allucinate su risorse inesistenti agli errori a cascata, ai quali si aggiunge una specifica vulnerabilità di sicurezza, l'iniezione indiretta di prompt, cui sono soggetti tutti gli LLM, ma che in un contesto agentivo può divenire estremamente pericolosa per le conseguenze disastrose che possono derivarne. Questa vulnerabilità, descritta da Greshake et al. (2023), comporta l'inserimento di istruzioni dannose nei documenti recuperati o elaborati dal sistema, con il potenziale di alterarne il comportamento. I meccanismi di controllo, che vanno dall'intervento umano nelle azioni critiche alle misure di protezione, dall'esecuzione in sandbox alla registrazione per l'audit, costituiscono la risposta ingegneristica al problema; tuttavia, la ricerca sulla robustezza degli agenti è ancora agli albori, e la sua maturazione rappresenta una delle frontiere più urgenti del settore. Nelle architetture agentive, il punto rilevante per questa riflessione è il modo in cui l'interazione si avvicina a una teoria della delega, in cui i livelli di autonomia, la governance degli strumenti e la verificabilità assumono un ruolo centrale, e in cui la natura del prompt si trasforma anziché diventare obsoleta.

7. La «fine» del prompt engineering?

Da quanto abbiamo detto finora deriva l'attuale vivace discussione sulla possibile fine del prompt engineering. La diagnosi è vera, a mio avviso, solo in senso limitato, e vale la pena distinguere con precisione tra i due sensi in cui il termine può essere inteso. Se per prompt engineering intendiamo un insieme di espedienti testuali (e non testuali) fortemente strutturati e predefiniti, la sua rilevanza e utilità sono effettivamente destinate a diminuire man mano che i modelli diventano più robusti, più capaci di interpretare istruzioni vaghe e più autonomi nella gestione di compiti complessi. È infatti documentato che i modelli recenti sono molto meno sensibili alle variazioni nella formulazione del prompt rispetto a quanto lo fossero i modelli di prima generazione. Se invece con prompt engineering intendiamo la competenza di costruire interazioni epistemicamente controllate con i sistemi generativi, allora essa non scompare, ma si trasforma. Il dibattito tra gli specialisti del settore registra, inoltre, una polarizzazione delle posizioni. Da un lato, figure influenti hanno espresso scetticismo nei confronti del prompt engineering come mestiere specializzato, suggerendo che si tratti di una soluzione temporanea destinata a essere superata dall'evoluzione delle architetture. Dall'altro, Andrej Kar-

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

pathy, in un intervento su X del giugno 2025 (Karpathy, 2025), ha proposto di designare come *context engineering*, anziché prompt engineering, la disciplina che va acquisendo sempre maggiore importanza, intesa come l'arte di riempire la finestra di contesto con le informazioni esattamente pertinenti al passo successivo. Il cambiamento terminologico non è puramente lessicale, poiché «prompt engineering» evoca la formulazione di stringhe di testo, mentre «context engineering» descrive una disciplina sistemica che include la progettazione di tutto ciò che entra nella finestra di contesto: istruzioni esplicite, esempi, fatti recuperati tramite retrieval, dati, strumenti disponibili, stato della conversazione e memorie persistenti. In questa lettura, il prompting tradizionale è un sottoinsieme del context engineering, e ciò che l'utente scrive direttamente è solo una delle componenti di un contesto progettato in modo sistematico (Anthropic, 2025; Mei et al., 2025). Il context engineering, in definitiva, non sostituisce il prompt engineering, ma lo include ed estende, spostando il baricentro dalla formulazione locale all'ingegneria dell'intero ambiente informativo e strumentale in cui opera il modello.

Sul piano empirico, le prove già citate di Meincke et al. (2025) sulla contingenza del prompting corroborano questa posizione, poiché rendono implausibile l'idea che esista un insieme stabile di best practice universali e spostano l'attenzione sulla progettazione dei sistemi, sulla valutazione e sulla curatela del contesto. Il prompt engineering appare quindi come un termine ombrello che copre sotto-competenze distinguibili: formulazione conversazionale locale, progettazione di schemi e vincoli per output strutturati, recupero e curatela del contesto, progettazione di strategie e workflow per agenti e valutazione con dati di riferimento e metriche robuste. Ciascuna di queste competenze richiede una formazione tecnica specifica e si interseca con la conoscenza del dominio: ed è questa che in definitiva resta il fondamento di un uso appropriato ed efficace, in ogni contesto applicativo, dei sistemi di Intelligenza Artificiale.

8. Dal prompt engineering alla competenza critica

Le osservazioni del paragrafo precedente consentono di introdurre la mia tesi conclusiva: il prompt engineering, e ogni sua evoluzione o diluizione, deve essere ricollocato all'interno di una più ampia competenza critica di dominio nell'interazione con l'IA generativa. Per l'utente comune, molte tecniche specializzate saranno probabilmente assorbite dall'evoluzione dei modelli e delle interfacce, oltre che dal *meta-prompting*, ovvero dall'uso di un LLM per costruire il prompt più accurato ed efficace. Per gli usi professionali, scientifici, pedagogici e culturali rimarrà essenziale la capacità di progettare flussi di lavoro, definire ruoli, costruire contesti, iterare le risposte, verificare le fonti, rendere espliciti i criteri, testare le asserzioni del modello e integrare l'output in processi controllati di produzione di conoscenza.

Ma in ogni caso d'uso, il passaggio da «come scrivere un buon prompt» a «cosa chiedere e come valutare la risposta» è la trasformazione più significativa determinata dall'evoluzione dei modelli di frontiera a cui abbiamo assistito negli ultimi anni. Se nelle fasi iniziali la competenza cruciale era la conoscenza delle idiosincrasie del modello, quali

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

formati producessero risposte migliori e quali formule verbali suscitassero le capacità desiderate, con i modelli recenti questa conoscenza tecnica diventa marginale, mentre la competenza di dominio, la capacità di formulare le domande giuste e di riconoscere le risposte errate, diventa il fattore differenziante. Gli studi che mettono a confronto l'uso degli LLM da parte di esperti e principianti convergono su questo punto. Noy e Zhang (2023) hanno evidenziato un effetto di livellamento, per cui i lavoratori meno capaci migliorano più di quelli più capaci; Dell'Acqua et al. (2023), in uno studio sui consulenti aziendali, hanno riscontrato che l'uso del modello migliora le prestazioni nella maggior parte dei compiti ma le degrada in quelli che esulano dai limiti di capacità del sistema. Ne consegue che l'LLM funge da amplificatore di competenza, non da sostituto, e che la capacità cruciale diventa la valutazione[4].

La necessità di verificare, contestualizzare e integrare l'output deriva direttamente dai limiti noti di questi sistemi: dalle allucinazioni, ovvero la generazione (in notevole diminuzione nei modelli più avanzati e potenti, ma non scomparsa) di informazioni false ma plausibili, alla confidenza calibrata in modo inaffidabile, alla sensibilità al framing, per cui la stessa domanda, formulata in modi diversi, può produrre risposte contraddittorie[5]. Da ciò deriva un principio metodologico per un'interazione efficace: richiedere output verificabili, adottare protocolli di controllo basati su test, fonti e confronti, effettuati con lo stesso modello o meglio con più modelli in modo concorrente o avversario, e mantenere distinti la generazione di ipotesi, la verifica e la decisione finale. Una valutazione critica matura dovrebbe articolarsi lungo almeno tre dimensioni: la verifica fattuale di specifiche asserzioni rispetto a fonti autorevoli, il controllo della coerenza logica interna degli argomenti e la calibrazione epistemica, ovvero la valutazione di quanto l'incertezza espressa dal modello corrisponda all'incertezza effettiva. La padronanza congiunta di queste dimensioni costituisce ciò che si può definire un'alfabetizzazione critica in materia di IA.

[4] Se ne trae una conseguenza rilevante per la distinzione tra utente esperto e utente comune. Con i modelli recenti tale distinzione non passa più per l'abilità nel formulare prompt, poiché la forma dell'input incide sempre meno sull'esito, ma per il possesso delle conoscenze necessarie a valutare la pertinenza e la correttezza della risposta. Non è quindi la capacità di costruire un buon prompt a separare i due profili, ma la competenza di dominio che consente di riconoscere quando l'output è inaffidabile, la stessa che gli studi qui citati attribuiscono ai professionisti più capaci. La distinzione, in altri termini, si sposta dal piano della tecnica a quello della valutazione consapevole, in prima istanza dei propri stessi limiti. Ma questa questione temo abbia poco a che fare con il modo di operare e la natura dei sistemi di IA generativa.

[5] Questo limite, o meglio questo carattere dei LLM, è per certi versi più insidioso, poiché non riguarda la correttezza di un singolo output bensì la stabilità della relazione tra input e output. La natura stocastica del campionamento, unita alla dipendenza fine dei risultati dalla forma del contesto, di cui sono indizio la sensibilità all'ordine degli esempi (Lu et al., 2022) e la variabilità del role prompting (Meincke et al., 2025), fa sì che una medesima richiesta possa produrre risposte diverse in esecuzioni successive, e che una variazione minima nella formulazione possa spostare l'esito in modo non prevedibile. La non ripetibilità così intesa non è un difetto accidentale destinato a scomparire con il progresso dei modelli, ma una proprietà strutturale dei sistemi generativi, e costituisce in molti contesti applicativi, segnatamente quelli che esigono tracciabilità e verificabilità, il vincolo più severo all'affidabilità dell'interazione, perché mette in questione la nozione stessa di controllo epistemico su cui questa riflessione si fonda. Ma questo è vero solo se si assume come valore epistemico non negoziabile la certezza. Personalmente ritengo che non sia tale, epistemicamente ed eticamente, anche perché, come ci ha insegnato Wittgenstein, è fuori dall'orizzonte di uso del linguaggio umano.

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

Questa prospettiva richiama, sullo sfondo, la tradizione filosofica della mente estesa di Clark e Chalmers (1998), che il concetto di co-intelligenza proposto da Mollick (2024) ha riportato in primo piano. Se i criteri per l'estensione cognitiva (che potremmo sintetizzare nella forma disponibilità stabile - accesso affidabile - integrazione funzionale) fossero soddisfatti, il LLM potrebbe essere considerato una componente del sistema cognitivo dell'utente, e non un mero strumento esterno. La questione è filosoficamente controversa e le condizioni di Clark e Chalmers non sono pienamente soddisfatte dai sistemi attuali, poiché la disponibilità è condizionata dall'accesso alla rete, l'affidabilità è limitata dalle allucinazioni e l'integrazione funzionale è mediata dall'interfaccia testuale. La traiettoria tecnologica, tuttavia, sembra orientata verso un progressivo avvicinamento a tali condizioni, e ciò rende le implicazioni pedagogiche della transizione tutt'altro che marginali, poiché se un'interazione efficace richiede competenza nel dominio, pensiero critico e capacità di valutazione, allora l'istruzione deve formare queste competenze, e non limitarsi a vietare l'uso dei modelli o a insegnare trucchi di prompting. In questa prospettiva, il prompt diviene un atto comunicativo, un dispositivo di orientamento interpretativo e uno strumento per il governo dell'incertezza, e la sua funzione principale non è comandare il modello, ma configurare uno spazio di cooperazione cognitiva in cui le capacità statistiche, linguistiche e culturali dell'LLM possano essere rese utili entro vincoli espliciti di razionalità, verificabilità e responsabilità. Ma diviene soprattutto una pratica epistemica: una forma di interrogazione disciplinata in cui l'intelligenza dell'utente non consiste nel trovare la formula magica, ma nel saper costruire le condizioni per una conversazione affidabile, controllabile e produttiva con un sistema linguistico generativo artificiale, sì, ma anche troppo umano.

Ringraziamenti

Ringrazio i due revisori anonimi per la lettura attenta e per i suggerimenti puntuali, che mi hanno aiutato a precisare l'argomentazione e a rendere più chiara l'esposizione. La responsabilità di quanto sostengo, e di ogni residua imprecisione, resta soltanto mia.

Bibliografia

Alammar, J., & Grootendorst, M. (2024). *Hands-on large language models: Language understanding and generation* (1st ed.). O'Reilly Media.

Anthropic. (2025). *Effective context engineering for AI agents*. https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents

Barez, F., Wu, T.-Y., Arcuschin, I., Lan, M., Wang, V., Siegel, N., Collignon, N., Neo, C., Lee, I., Paren, A., Bibi, A., Trager, R., Fornasiere, D., Yan, J., Elazar, Y., & Bengio, Y. (2025). Chain-of-thought is not explainability [Preprint]. *alphaXiv*. https://www.alphaxiv.org/abs/2025.02v1

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … & Amodei, D. (2020). *Language*

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

models are few-shot learners. In Advances in Neural Information Processing Systems, 33, 1877–1901.

Chalmers, D. J. (2026). What we talk to when we talk to language models [Preprint]. *PhilArchive*. https://philarchive.org/rec/CHAWWT-8

Chen, R., Arditi, A., Sleight, H., Evans, O., & Lindsey, J. (2025). Persona vectors: Monitoring and controlling character traits in language models [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2507.21509

Chen, Y., Benton, J., Radhakrishnan, A., Uesato, J., Denison, C., Schulman, J., Somani, A., Hase, P., Wagner, M., Roger, F., Mikulik, V., Bowman, S. R., Leike, J., Kaplan, J., & Perez, E. (2025). Reasoning models don't always say what they think [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2505.05410

Clark, A., & Chalmers, D. (1998). The extended mind. *Analysis*, 58(1), 7–19. https://doi.org/10.1093/analys/58.1.7

Dai, D., Sun, Y., Dong, L., Hao, Y., Ma, S., Sui, Z., & Wei, F. (2023). Why can GPT learn in-context? Language models secretly perform gradient descent as meta-optimizers. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Findings of the Association for Computational Linguistics*: ACL 2023 (pp. 4005–4019). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-acl.247

Dell'Acqua, F., McFowland, E., III, Mollick, E. R., Lifshitz-Assaf, H., Kellogg, K., Rajendran, S., Krayer, L., Candelon, F., & Lakhani, K. R. (2023). Navigating the jagged technological frontier: Field experimental evidence of the effects of AI on knowledge worker productivity and quality (Working Paper No. 24-013). *Harvard Business School*. https://doi.org/10.2139/ssrn.4573321

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *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) (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Feng, G., Zhang, B., Gu, Y., Ye, H., He, D., & Wang, L. (2023). Towards revealing the mystery behind chain of thought: A theoretical perspective. In *Advances in Neural Information Processing Systems* (Vol. 36, pp. 70757–70798). Curran Associates.

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec '23)* (pp. 79–90). Association for Computing Machinery. https://doi.org/10.1145/3605764.3623985

Huang, X., Liu, W., Chen, X., Wang, X., Wang, H., Lian, D., Wang, Y., Tang, R., & Chen, E. (2024). Understanding the planning of LLM agents: A survey [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2402.02716

Janus. (2022, September 2). *Simulators. LessWrong*. https://www.lesswrong.com/posts/vJFdjigzmcXMhNTsx/simulators

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

Kambhampati, S. (2024). Can large language models reason and plan? *Annals of the New York Academy of Sciences*, 1534(1), 15–18. https://doi.org/10.1111/nyas.15125

Karpathy, A. [@karpathy]. (2025, June 25). *+1 for "context engineering" over "prompt engineering". People associate prompts with short task descriptions you'd give an LLM in your day-to-day use. When in every industrial-strength LLM app, context engineering is the delicate art and science of filling the context window* [Post]. X. https://x.com/karpathy/status/1937902205765607626

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems* (Vol. 35, pp. 22199–22213). Curran Associates.

Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., & Chen, W. (2022). What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures* (pp. 100–114). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.deelio-1.10

Lu, Y., Bartolo, M., Moore, A., Riedel, S., & Stenetorp, P. (2022). Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers) (pp. 8086–8098). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.556

Marks, S., Lindsey, J., & Olah, C. (2026, February 23). *The persona selection model: Why AI assistants might behave like humans*. Anthropic Alignment Science Blog. https://www.anthropic.com/research/persona-selection-model

Mei, L., Yao, J., Ge, Y., Wang, Y., Bi, B., Cai, Y., Liu, J., Li, M., Li, Z.-Z., Zhang, D., Zhou, C., Mao, J., Xia, T., Guo, J., & Liu, S. (2025). A survey of context engineering for large language models [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2507.13334

Meincke, L., Mollick, E. R., Mollick, L., & Shapiro, D. (2025). Prompting science report 1: Prompt engineering is complicated and contingent [Working paper]. *SSRN*. https://doi.org/10.2139/ssrn.5165270

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., & Zettlemoyer, L. (2022). Rethinking the role of demonstrations: What makes in-context learning work? In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (pp. 11048–11064). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.759

Mollick, E. (2024). *Co-intelligence: Living and working with AI*. Portfolio/Penguin.

Noy, S., & Zhang, W. (2023). Experimental evidence on the productivity effects of generative artificial intelligence. *Science*, 381(6654), 187–192. https://doi.org/10.1126/science.adh2586

OpenAI. (2025). *Reasoning best practices. OpenAI Platform Documentation*. Retrieved June 1, 2026, from https://platform.openai.com/docs/guides/reasoning-best-practices

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell,

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems* (Vol. 35, pp. 27730–27744). Curran Associates.

Pan, J., Gao, T., Chen, H., & Chen, D. (2023). What in-context learning "learns" in-context: Disentangling task recognition and task learning. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Findings of the Association for Computational Linguistics: ACL 2023* (pp. 8298–8319). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-acl.527

Roncaglia, G. (2023). *L'architetto e l'oracolo: Forme digitali del sapere da Wikipedia a ChatGPT*. Laterza.

Shanahan, M. (2024). Still "talking about large language models": Some clarifications [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2412.10291

Shanahan, M., McDonell, K., & Reynolds, L. (2023). Role play with large language models. *Nature*, 623(7987), 493–498. https://doi.org/10.1038/s41586-023-06647-8

Sprague, Z., Yin, F., Rodriguez, J. D., Jiang, D., Wadhwa, M., Singhal, P., Zhao, X., Ye, X., Mahowald, K., & Durrett, G. (2025). To CoT or not to CoT? Chain-of-thought helps mainly on math and symbolic reasoning [Conference paper]. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*. https://doi.org/10.48550/arXiv.2409.12183

Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A., Cunningham, H., Turner, N. L., McDougall, C., MacDiarmid, M., Freeman, C. D., Sumers, T. R., Rees, E., Batson, J., Jermyn, A., … & Henighan, T. (2024). Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*. https://transformer-circuits.pub/2024/scaling-monosemanticity/

Turpin, M., Michael, J., Perez, E., & Bowman, S. R. (2023). Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in Neural Information Processing Systems* (Vol. 36, pp. 74952–74965). Curran Associates.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 5998–6008). Curran Associates.

Von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., & Vladymyrov, M. (2023). Transformers learn in-context by gradient descent. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, & J. Scarlett (Eds.), *Proceedings of the 40th International Conference on Machine Learning* (pp. 35151–35174). PMLR. https://proceedings.mlr.press/v202/von-oswald23a.html

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W. X., Wei, Z., & Wen, J. (2024). A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), Article 186345. https://doi.org/10.1007/s11704-024-40231-1

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., & Zhou, D. (2023). Self-consistency improves chain of thought reasoning in language

DigitCult
ISSN 2531-5994
VOL. 11, NO. 1
(2026)

models [Conference paper]. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*. https://doi.org/10.48550/arXiv.2203.11171

Wang, Z., Mao, S., Wu, W., Ge, T., Wei, F., & Ji, H. (2024). Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Volume 1: Long Papers) (pp. 257–279). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.15

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems* (Vol. 35, pp. 24824–24837). Curran Associates.

Xie, S. M., Raghunathan, A., Liang, P., & Ma, T. (2022). An explanation of in-context learning as implicit Bayesian inference [Conference paper]. In *The Tenth International Conference on Learning Representations (ICLR 2022)*. https://doi.org/10.48550/arXiv.2111.02080

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., & Narasimhan, K. (2023a). Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems* (Vol. 36, pp. 11809–11822). Curran Associates.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023b). ReAct: Synergizing reasoning and acting in language models [Conference paper]. In *The Eleventh International Conference on Learning Representations* (ICLR 2023). https://doi.org/10.48550/arXiv.2210.03629

Kong, A., Zhao, S., Chen, H., Li, Q., Qin, Y., Sun, R., Zhou, X., Wang, E., & Dong, X. (2024). Better Zero-Shot Reasoning with Role-Play Prompting. In K. Duh, H. Gomez, & S. Bethard (A c. Di), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)* (pp. 4099–4113). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.228

Zheng, M., Pei, J., Logeswaran, L., Lee, M., & Jurgens, D. (2024). When "a helpful assistant" is not really helpful: Personas in system prompts do not improve performances of large language models. In Y. Al-Onaizan, M. Bansal, & Y.-N. Chen (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024* (pp. 15126–15154). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-emnlp.888