Dal Prompt Engineering al Context Engineering

Scritto da Febogamedev

Il prompt

Il Prompt Engineering è la pratica di progettare, strutturare e raffinare le istruzioni fornite a un sistema di intelligenza artificiale generativa, con lo scopo di ottenere risultati più pertinenti, coerenti, controllabili e adeguati al compito (Liu et al., 2023; White et al., 2023).

Il prompt consiste nell'input fornito al modello, che può comprendere una o più istruzioni e altre informazioni (Liu et al., 2023).

In termini più generali, il prompt engineering può essere considerato una forma di progettazione dell'interazione linguistica tra una persona e un sistema generativo (Zamfirescu-Pereira et al., 2023).

Non consiste quindi soltanto nello scrivere correttamente una domanda o nel trovare una particolare formula capace di produrre sempre il risultato desiderato.
Significa piuttosto rendere espliciti l'obiettivo dell'interazione, le informazioni necessarie, i vincoli da rispettare e i criteri secondo cui valutare la risposta (White et al., 2023; Zamfirescu-Pereira et al., 2023).

Da questo punto di vista, il Prompt Engineering si colloca all'intersezione di più ambiti:

- È vicino al design dell'interazione, perché il prompt costituisce uno dei punti di contatto attraverso cui l'utente comunica con il sistema.
- È vicino alla comunicazione tecnica, perché richiede chiarezza, precisione, organizzazione delle informazioni e capacità di ridurre le ambiguità.
- È vicino al design dell'informazione, perché stabilisce quali dati debbano essere presentati, in quale ordine e con quale gerarchia.
- È legato, infine, alla progettazione del comportamento di un sistema probabilistico, perché la risposta non viene determinata rigidamente come l'esito di una sequenza di istruzioni tradizionali, ma viene generata in funzione del contesto disponibile (Amershi et al., 2019; White et al., 2023).

Un prompt può essere molto semplice, come una domanda breve, oppure più complesso e strutturato. Può contenere una richiesta, informazioni di contesto, esempi, vincoli formali, indicazioni sul destinatario, criteri di qualità e specifiche sul formato dell'output (White et al., 2023).

Chi progetta un prompt non stabilisce necessariamente, passo per passo, tutto ciò che il sistema dovrà fare.
Definisce piuttosto una configurazione del problema che orienta la generazione verso alcune soluzioni e ne rende meno probabili altre (Brown et al., 2020; Liu et al., 2023).

In questo senso:
Il prompt non ordina: orienta.

Questa caratteristica distingue il Prompt Engineering dalla programmazione tradizionale.

- In un programma deterministico, il progettista specifica una sequenza di operazioni che, date le stesse condizioni, dovrebbe produrre lo stesso risultato.
- In un sistema generativo, invece, le istruzioni delimitano uno spazio di possibilità entro cui viene prodotto l'output (Brown et al., 2020; Liu et al., 2023).

Il prompt come interfaccia

Il prompt può essere interpretato come una particolare forma di interfaccia (Zamfirescu-Pereira et al., 2023).

Nelle interfacce grafiche tradizionali, l'utente agisce attraverso pulsanti, finestre, menu, icone e campi di testo.
Le possibilità disponibili sono generalmente visibili e delimitate dalla struttura stessa dell'interfaccia.

In un'interfaccia linguistica, invece, lo spazio delle richieste possibili è molto più aperto.
L'utente può formulare obiettivi, descrivere situazioni, fornire esempi, correggere la direzione del lavoro e negoziare progressivamente il significato del compito.

Questa apertura rende l'interazione flessibile, ma anche più ambigua (Amershi et al., 2019; Zamfirescu-Pereira et al., 2023).

Una richiesta come "Spiegami l'inerzia."

non specifica il livello di approfondimento, il destinatario, il dominio applicativo, la lunghezza desiderata o lo scopo della spiegazione. Il sistema dovrà quindi inferire implicitamente molte di queste dimensioni e tenderà a produrre una risposta generalista (Zamfirescu-Pereira et al., 2023).

Una formulazione più precisa potrebbe essere:
"Spiega il concetto di inerzia a studenti di animazione che conoscono velocità e accelerazione, ma non hanno ancora studiato formalmente le leggi di Newton. Collega il concetto alla partenza e all'arresto dei corpi animati, usando un linguaggio rigoroso ma accessibile."

In questo caso non cambia soltanto la forma della domanda. Viene configurato più chiaramente il problema:

- È indicato il destinatario.
- Viene delimitato il dominio.
- È specificato lo scopo didattico.
- Viene stabilito il livello di conoscenza iniziale.

- È definito un criterio di qualità (White et al., 2023).

Il prompt funziona quindi come una rappresentazione linguistica parziale delle condizioni entro cui il sistema deve interpretare il compito.

È necessariamente parziale perché non può contenere tutto ciò che l'utente sa, pensa o desidera.
Progettare un prompt significa scegliere quali elementi debbano essere resi espliciti e quali possano essere lasciati impliciti (Zamfirescu-Pereira et al., 2023).

Nel corso del testo verrà utilizzata più volte l'espressione **nuova sessione di dialogo**.

Per nuova sessione di dialogo si intende una nuova conversazione avviata con il modello senza includere i turni della conversazione precedente.
Domande, risposte, correzioni, ipotesi e giustificazioni accumulate fino a quel momento non fanno quindi più parte del contesto conversazionale corrente, salvo che vengano reinserite esplicitamente o recuperate attraverso eventuali meccanismi persistenti del sistema (Anthropic, 2025).

Aprire una nuova sessione permette quindi di riproporre un problema senza trascinare automaticamente il percorso costruito nella conversazione precedente.
Questo può essere utile, per esempio, quando si vuole verificare se una risposta dipenda dalle premesse, dalle correzioni o dalle giustificazioni accumulate durante il dialogo.

A livello di interfaccia, nella maggior parte dei sistemi, significa semplicemente avviare una nuova chat.

Dal prompt al contesto

Il prompt è soltanto una delle componenti che influenzano il comportamento di un sistema generativo (Mei et al., 2025).

Quando interagiamo con un assistente basato su modelli linguistici, la risposta può dipendere anche da:
- Dalle istruzioni generali stabilite dal sistema.
- Dai messaggi precedenti della conversazione.
- Dai documenti messi a disposizione.
- Dagli esempi forniti.
- Dalle informazioni recuperate da fonti esterne.
- Dagli strumenti utilizzabili.
- Dallo stato dell'applicazione.
- Dalle memorie associate all'utente o al progetto.
- Dai criteri di valutazione.
- Dalle regole di sicurezza.
- Dai risultati prodotti durante i passaggi precedenti del compito.

L’insieme delle informazioni, delle istruzioni, degli stati e delle risorse disponibili costituisce il **contesto operativo** entro cui il sistema interpreta la richiesta e produce il proprio comportamento (Anthropic, 2025; Mei et al., 2025).



Visto che il prompt è solo una piccola parte del contesto, è necessario progettare tale contesto, tramite il Context Engineering (Anthropic, 2025; Mei et al., 2025).

Il Context Engineering è definito nel modo seguente:
la progettazione delle condizioni informative, operative e valutative entro cui un sistema generativo interpreta un obiettivo, compie delle operazioni e produce un risultato (Anthropic, 2025; Mei et al., 2025).

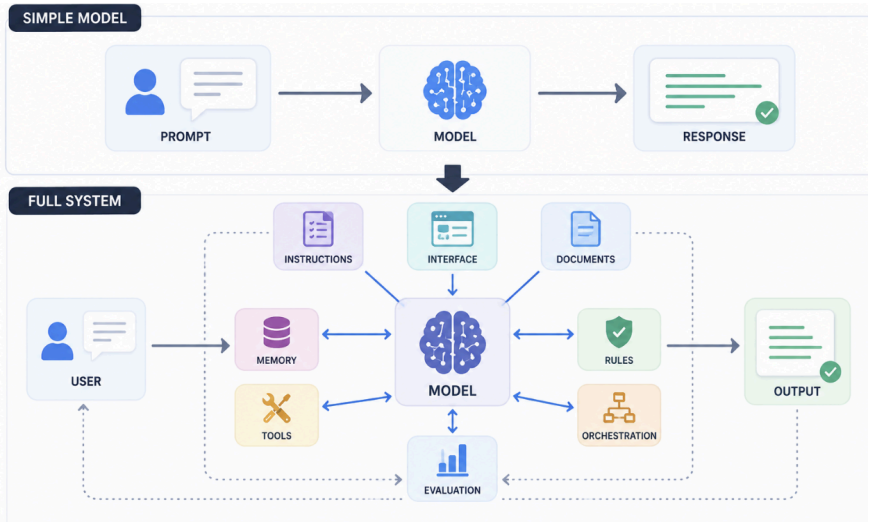
Quindi:

- Il Prompt Engineering concentra l’attenzione soprattutto sulla formulazione delle istruzioni.
- Il Context Engineering allarga il campo e considera l’intero ambiente entro cui tali istruzioni vengono interpretate ed eseguite (Anthropic, 2025; Mei et al., 2025).

Il prompt appartiene quindi al contesto, ma non coincide con esso.
Questa distinzione è diventata particolarmente importante con l’evoluzione dei sistemi generativi.
Nei primi utilizzi pubblici delle LLM, l’interazione assumeva spesso una forma relativamente semplice:
richiesta → risposta

Nei sistemi contemporanei, invece, il modello può essere inserito in processi più articolati. Può consultare documenti, interrogare database, utilizzare strumenti, eseguire codice, mantenere uno stato operativo, confrontare risultati e sviluppare il compito attraverso più passaggi (Anthropic, 2024; Lewis et al., 2020; Yao et al., 2023).

Il problema progettuale non è più soltanto: “Quale domanda devo scrivere?”
Diventa: “Quali condizioni deve trovare il sistema per poter svolgere correttamente il compito?”



Errore nel modello

Un buon prompt non rende infallibile un modello.

Non corregge automaticamente informazioni errate apprese durante l'addestramento, non garantisce che le fonti recuperate siano corrette, non trasforma una risposta plausibile in una risposta vera e non sostituisce la competenza disciplinare o la necessità della verifica (Autio et al., 2024).

Anche un contesto molto ben progettato può produrre risultati errati, incompleti o inadatti.
Può contenere dati sbagliati, fonti insufficienti o vincoli incompatibili.
Il sistema può interpretare male una richiesta, scegliere uno strumento inadeguato o produrre una sintesi che altera il significato dei materiali (Autio et al., 2024).

Un buon contesto può tuttavia:

- Ridurre l'ambiguità.
- Aumentare la pertinenza.
- Rendere espliciti i criteri.
- Facilitare la verifica.
- Diminuire la variabilità indesiderata.
- Limitare alcune classi di errore.
- Rendere il processo più osservabile (Anthropic, 2025; White et al., 2023).

Non produce conoscenza certa, ma costruisce condizioni migliori per utilizzare le capacità disponibili (Autio et al., 2024).
Nei sistemi agentici la responsabilità progettuale è ancora maggiore.
Il sistema può infatti intervenire su risorse e ambienti esterni, e non limitarsi a produrre un testo (Anthropic, 2024; Yao et al., 2023).

Il contesto operativo deve quindi comprendere anche:

- Limiti dei permessi.
- Gestione degli errori.

- Validazione degli input.
- Controllo delle azioni irreversibili.
- Eventuale conferma umana.
- Tracciabilità delle operazioni (Anthropic, 2024; Autio et al., 2024).

Un sistema che produce un testo errato può generare disinformazione.
Un sistema che esegue un'azione errata può modificare dati, inviare comunicazioni o intervenire su un ambiente esterno (Autio et al., 2024).

La progettazione del contesto e dei meccanismi di controllo deve pertanto essere proporzionata alle possibili conseguenze del compito (Autio et al., 2024).

Esempio
Immaginiamo un sistema agentico utilizzato da un'azienda per gestire automaticamente le richieste dei clienti.

Un cliente scrive: "Vorrei annullare il mio ordine perché ho cambiato idea."

Il sistema agentico potrebbe:
leggere il messaggio → identificare l'ordine → verificare lo stato della spedizione → decidere se l'annullamento è consentito → utilizzare il gestionale → modificare l'ordine → inviare una conferma al cliente.

Se il modello interpreta male la richiesta e produce soltanto una risposta testuale sbagliata, può essere relativamente limitato: "Il suo ordine è stato annullato", anche se in realtà non lo è stato

Il cliente riceve un'informazione errata, ma l'ordine nel sistema aziendale rimane invariato. In un sistema agentico, invece, lo stesso errore può trasformarsi in un'azione reale.

Supponiamo che il modello associ per errore la richiesta all'ordine sbagliato e utilizzi gli strumenti a sua disposizione per annullarlo.
Il processo potrebbe diventare:
Interpretazione errata → selezione dell'ordine sbagliato → chiamata allo strumento → annullamento reale → modifica dei dati aziendali

A questo punto non abbiamo più soltanto un output scorretto. Il sistema ha modificato lo stato del mondo esterno.
Le conseguenze potrebbero includere:
- Annullamento dell'ordine di un altro cliente.
- Rimborso non dovuto.
- Modifica del magazzino.
- Comunicazioni automatiche errate.
- Necessità di intervento umano per ripristinare la situazione.
- Possibili problemi economici o di fiducia.

Per questo, prima di eseguire un'azione importante, l'architettura potrebbe imporre controlli aggiuntivi.

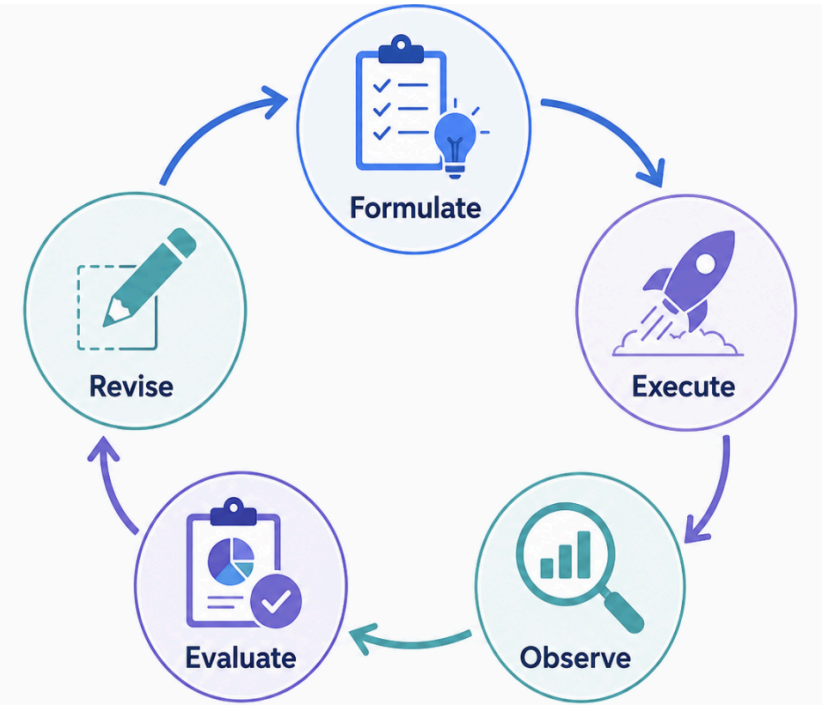
Per esempio, il modello può individuare l'ordine e proporne l'annullamento, ma il sistema verifica che il cliente corrisponda realmente al proprietario dell'ordine e, oltre una certa

soglia economica, richiede una conferma umana prima di procedere.

La differenza fondamentale è quindi:
- Errore generativo → produce un'informazione sbagliata.
- Errore agentico → può produrre una conseguenza sbagliata.

Poiché il comportamento del sistema non è completamente prevedibile o stabile in ogni condizione, non esiste un prompt perfetto che produca risultati adeguati indipendentemente dal modello, dal contesto e dal compito (Autio et al., 2024; Zamfirescu-Pereira et al., 2023).

Esiste invece un processo iterativo di progettazione:
formulazione → esecuzione → osservazione → valutazione → revisione (Amershi et al., 2019; Anthropic, 2025)



Quando l'output non è adeguato, il problema non si risolve necessariamente aggiungendo più istruzioni. Può essere necessario eliminare informazioni superflue, correggere una fonte, modificare il recupero delle informazioni, chiarire il destinatario, introdurre una verifica o riconoscere che il sistema utilizzato non è adatto al compito (Anthropic, 2025).

Il Context Engineering sposta quindi l'attenzione dalla ricerca della frase ideale alla progettazione e al miglioramento dell'intero sistema (Anthropic, 2025; Mei et al., 2025).

Perché il Context Engineering è importante

Il passaggio dal Prompt Engineering al Context Engineering modifica quindi l'oggetto della progettazione.

Non si interviene più soltanto sulla formulazione della richiesta, ma sull'insieme delle informazioni, degli strumenti, delle regole e delle procedure che orientano il comportamento del sistema (Anthropic, 2025; Mei et al., 2025).

Il prompt rimane una componente fondamentale, ma acquista significato all'interno del contesto più ampio in cui viene interpretato ed elaborato (Anthropic, 2025; Mei et al., 2025)

Il Context Engineering non consiste quindi semplicemente nel chiedere meglio, ma nel progettare le condizioni che rendono possibile una risposta o un'azione più pertinente, verificabile e adeguata allo scopo (Anthropic, 2025; Mei et al., 2025).
Per questo il Context Engineering è importante almeno per tre ragioni:

- Riduce l'indeterminatezza
- Esternalizza il pensiero strutturato
- Trasforma il linguaggio in un ambiente operativo

Riduce l'indeterminatezza
Una richiesta vaga lascia aperte molte interpretazioni possibili.
Specificare il dominio, il destinatario, i materiali e i criteri restringe lo spazio delle risposte plausibili e rende più probabile un risultato pertinente (Anthropic, 2025).

Esternalizza il pensiero strutturato
Progettare il contesto obbliga l'utente a rendere esplicite informazioni che spesso rimangono implicite.
Definire obiettivi, fonti, assunti e criteri aiuta a comprendere meglio il problema prima ancora che il sistema produca una risposta (White et al., 2023).

Trasforma il linguaggio in un ambiente operativo
L'interazione con i sistemi generativi non avviene più soltanto attraverso comandi formalizzati o interfacce grafiche. Avviene attraverso ambienti linguistici, documentali e operativi nei quali istruzioni, dati, strumenti e risultati intermedi vengono combinati (Anthropic, 2025; Mei et al., 2025).

- Quando questo ambiente è progettato male, anche un modello avanzato può produrre risultati mediocri.
- Quando è progettato bene, il sistema può diventare uno strumento più efficace di analisi, scrittura, progettazione, trasformazione e supporto decisionale (Anthropic, 2025; Mei et al., 2025).

Materiale di supporto

Esercizi

Dal prompt al contesto
Uno studente sostiene:
«Per ottenere una buona risposta da un'IA basta scrivere un prompt molto preciso.»

Analizza criticamente questa affermazione.

- Definisci che cosa si intende per Prompt Engineering.
- Spiega perché progettare un prompt non significa semplicemente "fare una domanda migliore".
- Spiega il significato dell'affermazione: «Il prompt non ordina: orienta».
- Confronta il comportamento di un programma deterministico con quello di un sistema generativo.
- Spiega perché il prompt può essere considerato una forma di interfaccia linguistica.
- Individua almeno cinque informazioni che un prompt complesso potrebbe rendere esplicite, per esempio destinatario, obiettivo, vincoli o criteri di qualità.
- Spiega perché una richiesta come «Spiegami l'inerzia» lascia al modello numerose decisioni implicite.
- Spiega invece quali informazioni aggiunge una richiesta che specifichi destinatario, conoscenze iniziali, dominio e criterio di qualità.
- Definisci il contesto operativo di un sistema generativo.
- Elenca almeno sei elementi del contesto che possono influenzare la risposta oltre al prompt.
- Spiega la differenza fondamentale tra Prompt Engineering e Context Engineering.
- Commenta infine questa affermazione: il prompt appartiene al contesto, ma non coincide con esso.

Un buon contesto elimina gli errori?
Un'azienda utilizza un sistema di IA per rispondere alle richieste dei clienti. In una seconda versione decide di permettere al sistema anche di modificare direttamente gli ordini nel gestionale.

Rispondi alle seguenti domande.

- Spiega perché un buon prompt non rende infallibile un modello.
- Indica almeno quattro cause per cui una risposta può rimanere errata anche con un contesto ben progettato.
- Indica almeno quattro vantaggi che un buon contesto può comunque offrire.
- Spiega la differenza tra errore generativo ed errore agentico.
- Perché l'errore di un sistema che produce soltanto testo può avere conseguenze diverse dall'errore di un sistema che può modificare dati esterni?
- Immagina che il sistema annulli l'ordine del cliente sbagliato. Individua almeno quattro possibili conseguenze.
- Indica quali controlli potrebbero essere introdotti prima di consentire l'annullamento.
- Spiega perché i limiti dei permessi sono una componente del Context Engineering.

- Spiega il ruolo della validazione degli input.
- Spiega quando potrebbe essere appropriata una conferma umana.
- Perché può essere importante conservare una traccia delle operazioni svolte dal sistema?
- Ricostruisci il ciclo iterativo: Formulazione → esecuzione → osservazione → valutazione → revisione.
- Spiega perché, davanti a un risultato insoddisfacente, aggiungere semplicemente altre istruzioni al prompt potrebbe non essere la soluzione corretta.
- Concludi spiegando perché il Context Engineering sposta l'attenzione dalla ricerca della "frase perfetta" alla progettazione dell'intero sistema.

Dal prompt vago al prompt progettato
Apri una nuova conversazione con un sistema di IA e inserisci:
«Spiegami il machine learning.»

Salva la risposta ottenuta.
Successivamente apri una seconda nuova conversazione e riprogetta il prompt specificando almeno:

- Destinatario.
- Conoscenze iniziali.
- Obiettivo.
- Dominio o contesto applicativo.
- Livello di approfondimento.
- Vincoli.
- Esempi desiderati.
- Criteri di qualità.
- Formato dell'output.

Per esempio, puoi immaginare che il destinatario sia uno studente di comunicazione visiva che non possiede conoscenze di programmazione.

Dopo aver ottenuto la seconda risposta:

- Confronta le due risposte.
- Individua quali decisioni il modello ha dovuto inferire autonomamente nel primo caso.
- Individua quali decisioni hai invece reso esplicite nel secondo.
- Valuta le risposte secondo almeno quattro criteri definiti prima del confronto.
- Indica quale risposta è più pertinente rispetto al destinatario scelto.
- Individua eventuali informazioni inutili presenti nella prima risposta.
- Individua eventuali problemi rimasti anche nella seconda.
- Modifica il prompt una terza volta sulla base dei problemi osservati.
- Confronta la terza risposta con la seconda.
- Documenta il processo attraverso la sequenza: Formulazione → esecuzione → osservazione → valutazione → revisione.
- Spiega perché aprire una nuova sessione per ogni prova rende più controllabile il confronto.
- Concludi indicando quali elementi del prompt hanno inciso maggiormente sul risultato e quali invece hanno avuto un effetto limitato.

Progettare il contesto di un assistente agentico
Un negozio online vuole utilizzare un assistente di IA per gestire richieste come:
«Vorrei annullare il mio ordine perché ho cambiato idea.»

Il sistema può consultare il database degli ordini e dispone potenzialmente di uno strumento capace di annullarli.

Il tuo compito non è semplicemente scrivere il prompt, ma progettare il contesto operativo necessario affinché il sistema lavori in modo controllabile.

Definisci:
- L'obiettivo del sistema.
- Le informazioni che deve ricevere dal messaggio del cliente.
- I documenti o le regole aziendali che dovrebbe poter consultare.
- I dati dell'ordine che dovrebbe recuperare.
- Gli strumenti che dovrebbe poter utilizzare.
- Le operazioni che dovrebbe poter eseguire autonomamente.
- Le operazioni che invece dovrebbero essere vietate.
- I controlli necessari per verificare che l'ordine appartenga realmente al cliente.
- Le condizioni che stabiliscono se un ordine può essere annullato.
- Le situazioni nelle quali deve essere richiesta una conferma umana.
- Le azioni considerate irreversibili o ad alto impatto.
- Il sistema di gestione degli errori.
- Le informazioni che devono essere registrate per garantire la tracciabilità delle operazioni.
- I criteri con cui valutare se il risultato è corretto.
- Il comportamento da adottare quando le informazioni disponibili sono insufficienti o contraddittorie.

Successivamente prova il sistema, o simula il suo comportamento, con almeno quattro casi:
- Ordine corretto e ancora annullabile.
- Ordine già spedito.
- Numero d'ordine appartenente a un altro cliente.
- Richiesta ambigua che non permette di identificare con certezza l'ordine.

Per ogni caso descrivi: Input → informazioni recuperate → controlli → decisione → eventuale azione → verifica finale.

Concludi indicando quali elementi appartengono al prompt e quali invece appartengono al più ampio contesto operativo.

Bibliografia

Amershi, S., Weld, D. S., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P. N., Inkpen, K., Teevan, J., Kikin-Gil, R., & Horvitz, E. (2019).

Guidelines for human-AI interaction. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (Paper 3, pp. 1-13). Association for Computing Machinery. https://doi.org/10.1145/3290605.3300233

Anthropic. (2024, December 19). *Building effective agents*. https://www.anthropic.com/engineering/building-effective-agents

Anthropic. (2025, September 29). *Effective context engineering for AI agents*. https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents

Autio, C., Schwartz, R., Dunietz, J., Jain, S., Stanley, M., Tabassi, E., Hall, P., & Roberts, K. (2024). *Artificial intelligence risk management framework: Generative artificial intelligence profile* (NIST AI 600-1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.600-1

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 models are few-shot learners*. In Advances in Neural Information Processing Systems (Vol. 33, pp. 1877-1901). https://arxiv.org/abs/2005.14165

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). *Retrieval-augmented generation for knowledge-intensive NLP tasks*. In Advances in Neural Information Processing Systems (Vol. 33, pp. 9459-9474). https://arxiv.org/abs/2005.11401

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). *Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing*. ACM Computing Surveys, 55(9), Article 195. https://doi.org/10.1145/3560815

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*. arXiv. https://doi.org/10.48550/arXiv.2507.13334

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). *A prompt pattern catalog to enhance prompt engineering with ChatGPT*. arXiv. https://doi.org/10.48550/arXiv.2302.11382

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). *ReAct: Synergizing reasoning and acting in language models*. In International Conference on Learning Representations. https://openreview.net/forum?id=WE_vluYUL-X

Zamfirescu-Pereira, J. D., Wong, R. Y., Hartmann, B., & Yang, Q. (2023). *Why Johnny can't prompt: How non-AI experts try (and fail) to design LLM prompts*. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (Article 113, pp. 1-21). Association for Computing Machinery. https://doi.org/10.1145/3544548.3581388

Approfondimenti

- Riccardo D'Iglio e Alessandro Mattavelli, Prompt efficaci per professionisti e imprese. Guida pratica per governare l'intelligenza artificiale generativa, ottenere output affidabili e creare valore nei processi di lavoro, Maggioli Editore, 2026 — È probabilmente il testo italiano più vicino alla direzione della dispensa. Parte dal

Prompt Engineering ma arriva a tecniche avanzate, prompt chaining, ReAct, automazioni, integrazione nei processi di lavoro e dedica esplicitamente spazio al Context Engineering. È particolarmente coerente con l'idea che non basti formulare bene una richiesta, ma sia necessario organizzare contesto, strumenti e workflow.

- Flavio Corradini, Il prompt responsabile. Linguaggio e metodo per governare davvero l'IA generativa, 1ª ed., FrancoAngeli, 2026 — Affronta il prompt non come una formula magica, ma come parte di un processo strutturato nel quale devono essere definiti obiettivi, contesto, destinatari, output, iterazione, controllo e responsabilità umana. È particolarmente adatto alla parte della dispensa che sottolinea come un buon prompt non renda infallibile il modello e come verifica e competenza disciplinare rimangano indispensabili.
- Massimiliano Nicotra, Tecniche di prompt e agenti AI. Guida all'uso dell'intelligenza artificiale per professionisti e imprese, Giuffrè, 2026 — Collega Prompt Engineering, limiti dei modelli, agenti AI e automazione dei workflow. È molto pertinente alla seconda metà della dispensa, dove il problema passa dalla qualità di un output testuale alle conseguenze delle azioni compiute da un sistema agentico. Il volume affronta inoltre rischi, responsabilità e condizioni nelle quali non è opportuno delegare completamente un'operazione all'IA.
- Micheal Lanham, AI Agent in pratica. Creare assistenti intelligenti e su misura per ogni attività, Apogeo, 2025 — È l'approfondimento tecnico più utile per comprendere cosa significhi concretamente costruire un sistema in cui il modello dispone di strumenti, conoscenza esterna, RAG, planning, feedback, valutazione dei prompt e sistemi multi-agente. Permette quindi di vedere in pratica molti degli elementi che nella dispensa vengono ricondotti al più ampio contesto operativo del modello.
- Chiara Martino, Intelligenza Artificiale Conversazionale. Processi, strumenti e professioni per creare chatbot e assistenti vocali, 1ª ed., FrancoAngeli, 2024 — Approfondisce Conversation Design, Prompt Engineering, Knowledge Engineering e progettazione dei workflow conversazionali. È particolarmente interessante rispetto alla tua interpretazione del prompt come forma di interfaccia: permette di spostare l'attenzione dalla singola domanda alla progettazione dell'interazione complessiva tra persona e sistema.

Nota editoriale

Per la realizzazione di questi materiali ho utilizzato strumenti di intelligenza artificiale generativa come supporto alla scrittura, in particolare per migliorare la forma del testo, riorganizzare i contenuti, correggere la formulazione e velocizzare alcune attività redazionali.

Lavorando autonomamente alla produzione di questi materiali, cerco infatti di automatizzare tutte le attività che possono esserlo, così da poter dedicare più tempo alla ricerca, alla progettazione e allo sviluppo dei contenuti.

L'intelligenza artificiale non determina però i contenuti dell'opera: la scelta degli argomenti, la struttura, le idee, le interpretazioni, gli esempi e l'impostazione didattica sono elaborati da me. L'IA viene quindi utilizzata principalmente come strumento di supporto alla produzione e alla revisione formale, mentre la responsabilità autoriale e progettuale dei contenuti rimane mia.

Salvo diversa indicazione, questo materiale è distribuito con licenza Creative Commons Attribuzione – Non commerciale – Condividi allo stesso modo 4.0 Internazionale (CC BY-NC-SA 4.0).

È quindi possibile condividere, ridistribuire, modificare e creare opere derivate a partire dal materiale, a condizione di attribuirne correttamente la paternità, di non utilizzarlo per scopi commerciali e di distribuire eventuali versioni modificate o derivate con la stessa licenza.

