Prompt Engineering

Scritto da Febogamedev

In questo capitolo ci concentreremo sul **Prompt Engineering**, cioè sulla progettazione del prompt e sull'analisi degli output che esso contribuisce a produrre (Liu et al., 2023; White et al., 2023).

È importante ricordare che il prompt costituisce soltanto una parte del contesto operativo. Un prompt ben progettato, da solo, non garantisce quindi un buon risultato: la qualità dell'output dipende anche dalle informazioni disponibili, dagli strumenti, dalle memorie, dalle regole e dai criteri di valutazione che compongono il sistema.

Il prompt rimane tuttavia una componente fondamentale, perché rappresenta il principale mezzo attraverso cui l'utente definisce il compito, esplicita il proprio obiettivo e orienta il comportamento del modello. La sua progettazione dovrebbe quindi avvenire dopo aver chiarito e organizzato il contesto necessario, del quale costituisce l'espressione linguistica più immediata (Reynolds & McDonell, 2021; White et al., 2023).

Formulazione del prompt

In senso generale, **il prompt** è l'input linguistico fornito a un sistema generativo (Liu et al., 2023).
Può assumere la forma di una domanda, di un'istruzione, di una descrizione, di un insieme di dati oppure di una combinazione di questi elementi.

Dal punto di vista progettuale, può essere definito in modo più preciso:
"Il prompt è la componente linguistica esplicita attraverso cui l'utente definisce il compito e fornisce una parte del contesto necessario a interpretarlo".

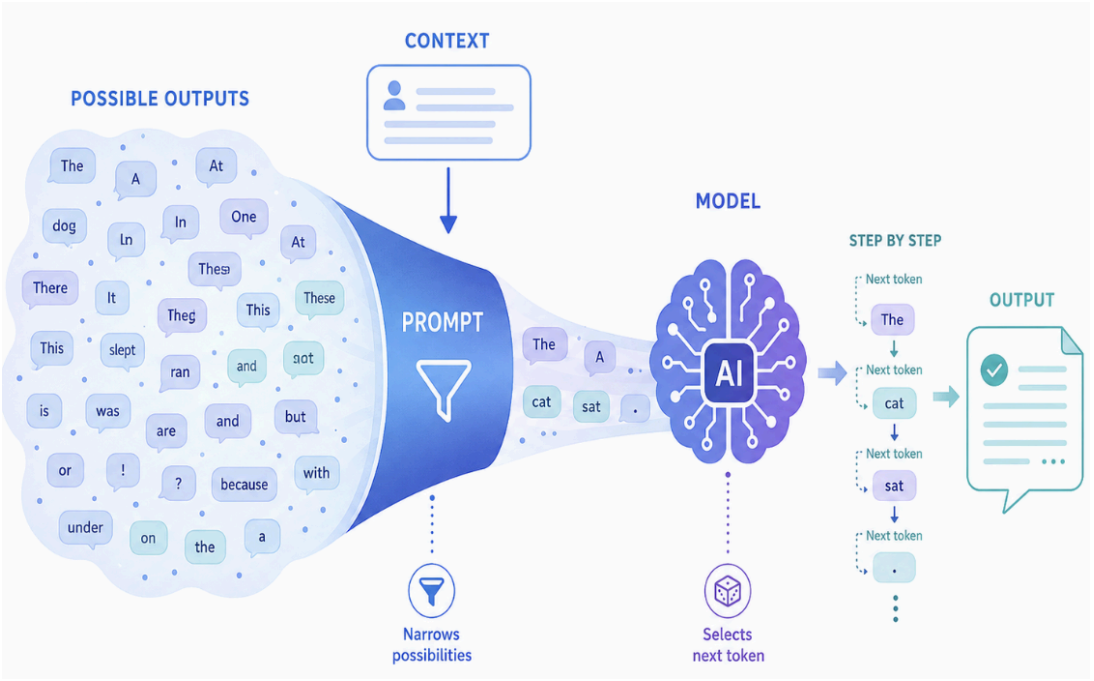
Il prompt non coincide quindi con l'intero contesto operativo. Il sistema può utilizzare anche istruzioni generali, messaggi precedenti, documenti, memorie, strumenti e risultati intermedi. Attraverso il prompt, però, l'utente rende esplicite le condizioni che considera rilevanti per il compito (Anthropic, n.d.-b).

Scrivere un prompt significa trasformare un'intenzione, spesso ancora implicita, in una struttura informativa utilizzabile dal sistema (Reynolds & McDonell, 2021).
Non significa impartire un ordine nel senso classico del termine, ma definire alcune delle condizioni entro cui verrà prodotto il risultato.

Questo perché un modello linguistico non interpreta intenzioni né comprende obiettivi nel senso umano del termine (Autio et al., 2024).
Non dispone inoltre di un criterio intrinseco con cui verificare la verità o la correttezza di ciò che produce (Autio et al., 2024; OpenAI, 2023).
Elabora il contesto disponibile e genera l'output producendo, a ogni passaggio, una distribuzione di probabilità sui token possibili, dalla quale il sistema seleziona o campiona

progressivamente una continuazione (Brown et al., 2020; OpenAI, 2023; Vaswani et al., 2017).

Per questo il prompt non controlla rigidamente il risultato, ma contribuisce a influenzare la distribuzione di probabilità delle possibili continuazioni o dei possibili output (Brown et al., 2020; Reynolds & McDonell, 2021).
Può essere considerato una struttura informativa che restringe lo spazio delle interpretazioni plausibili e rende alcune continuazioni più probabili di altre (Reynolds & McDonell, 2021).



Un errore comune consiste nel pensare che un buon prompt debba essere necessariamente lungo o molto dettagliato (Anthropic, n.d.-b; Zamfirescu-Pereira et al., 2023).

In realtà, come scopriremo a breve, un buon prompt è strutturato, non prolisso (Anthropic, n.d.-b; White et al., 2023).
La sua struttura serve a ridurre le ambiguità nei punti critici del compito.
Può chiarire, per esempio:

- Che cosa deve essere prodotto.
- Quale scopo deve avere l'output.
- A quale destinatario è rivolto.
- In quale dominio disciplinare deve operare il sistema.
- Quale livello di approfondimento è richiesto.
- Quali informazioni devono essere considerate.
- Quale formato deve assumere il risultato.
- Quali vincoli e criteri di qualità devono essere rispettati.

Non tutti questi elementi devono essere presenti in ogni prompt.
Una richiesta semplice può essere formulata in poche parole, mentre un compito ambiguo o complesso può richiedere una struttura più articolata (Anthropic, n.d.-b).

Esempio
Una richiesta come “Analizza questa interfaccia.” indica l’operazione generale da compiere, ma lascia aperte numerose interpretazioni.

Non specifica quale aspetto dell’interfaccia debba essere analizzato, a quale scopo servirà l’analisi, quale livello di approfondimento sia richiesto o secondo quali criteri debba essere valutato il progetto.

Il sistema potrebbe quindi produrre una descrizione generica, soffermandosi arbitrariamente sull’estetica, sull’usabilità o sull’organizzazione dei contenuti.

Una formulazione più precisa potrebbe essere:
“Analizza l’interfaccia allegata dal punto di vista dell’usabilità per utenti che utilizzano per la prima volta l’applicazione. Concentrati sulla gerarchia visiva, sulla riconoscibilità delle azioni principali e sulla chiarezza del feedback. Organizza la risposta in tre sezioni: punti di forza, criticità e proposte di miglioramento. Per ogni criticità, indica quale elemento dell’interfaccia la produce e quale modifica potrebbe ridurla.”

In questo caso vengono rese esplicite diverse dimensioni del compito:
- L’oggetto dell’analisi.
- La prospettiva disciplinare.
- Il tipo di utente considerato.
- Gli aspetti da osservare.
- La struttura dell’output.
- Il livello di concretezza richiesto.
- I criteri secondo cui formulare i suggerimenti.

La seconda formulazione è più articolata, ma la sua efficacia non dipende semplicemente dalla lunghezza. Dipende dal fatto che riduce le interpretazioni irrilevanti e rende espliciti gli elementi che devono orientare l’analisi (Anthropic, n.d.-b; White et al., 2023).

La struttura del prompt non deve quindi essere considerata una ricetta universale (Zamfirescu-Pereira et al., 2023).
È una scelta progettuale che deve essere proporzionata alla complessità del problema (Anthropic, n.d.-b).

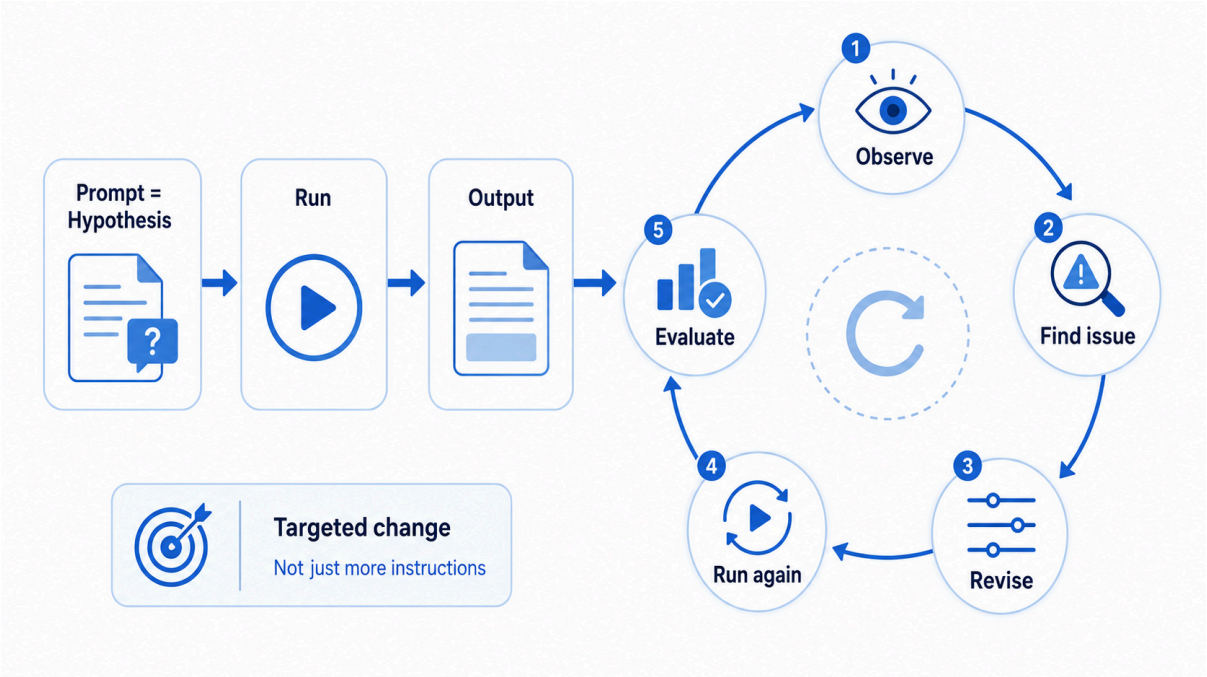
Specificare troppi dettagli può essere controproducente.
Un prompt sovraccarico può contenere informazioni irrilevanti, istruzioni duplicate o vincoli in conflitto tra loro (Anthropic, n.d.-b).
Il compito del progettista consiste nel rendere esplicito ciò che è necessario senza introdurre rumore inutile.

Un prompt non nasce necessariamente corretto o completo al primo tentativo (Anthropic, n.d.-a; Madaan et al., 2023).
Può essere considerato un’ipotesi sul modo più adeguato di configurare il compito (Zhou et al., 2022).

L’output permette di osservare gli effetti di tale configurazione e di individuare eventuali distanze rispetto al risultato atteso.
La progettazione del prompt è quindi spesso iterativa (Anthropic, n.d.-a; Madaan et al., 2023):

- Osservare il comportamento del sistema.
- Individuare eventuali ambiguità o problemi.
- Modificare in modo mirato istruzioni, vincoli, informazioni o esempi.
- Eseguire nuovamente il compito.
- Valutare il nuovo risultato.

Iterare non significa accumulare istruzioni indiscriminatamente. Significa formulare un’ipotesi sulla causa del problema e intervenire sulla componente che potrebbe averlo prodotto (Anthropic, n.d.-a).



Progettare un prompt richiede quindi di:

- Assumersi la responsabilità delle condizioni rese esplicite.
- Riconoscere che una parte del contesto rimarrà inevitabilmente implicita.
- Accettare l’incertezza come proprietà del processo generativo.
- Lavorare sulla probabilità di ottenere un risultato adeguato, non sulla certezza assoluta.

Diagnosi degli output insoddisfacenti

Nel lavoro con i sistemi generativi, un output insoddisfacente non deve essere interpretato soltanto come un malfunzionamento (Madaan et al., 2023; Zamfirescu-Pereira et al., 2023). Può diventare un segnale diagnostico utile per comprendere quale parte del prompt, del contesto o del sistema debba essere rivista.

La distanza tra il risultato ottenuto e quello atteso può dipendere da cause differenti.

- Richiesta troppo vaga: Una richiesta generica lascia aperto uno spazio molto ampio di interpretazioni possibili. Il sistema può quindi produrre una risposta corretta in senso generale, ma poco centrata rispetto allo scopo reale (Anthropic, n.d.-b; White et al., 2023).
- Vincoli contraddittori: Il prompt può contenere richieste difficilmente conciliabili. Chiedere, per esempio, un'analisi estremamente approfondita e contemporaneamente una risposta di pochissime righe costringe il sistema a trovare un compromesso che potrebbe non soddisfare nessuno dei due criteri.
- Destinatario o dominio non definiti: Una risposta può risultare formalmente corretta ma fuori target quando non è chiaro a chi sia rivolta, quale livello di competenza debba presupporre o all'interno di quale ambito disciplinare debba essere interpretata (White et al., 2023).
- Criteri di qualità insufficienti: Il modello può produrre un testo fluido e ben organizzato, ma inadatto allo scopo, se non sono stati chiariti i criteri secondo cui il risultato dovrà essere giudicato (Anthropic, n.d.-a).
- Informazioni incomplete o inaffidabili: Il problema può dipendere non dalla formulazione del prompt, ma dai documenti disponibili, dalle fonti recuperate, dalle memorie o dai risultati forniti dagli strumenti (Autio et al., 2024).
- Limiti del modello o del compito: Alcune richieste superano ciò che il sistema può garantire in modo affidabile. Un prompt non può assicurare la verità fattuale assoluta, la verifica autonoma di ogni affermazione o la produzione di giudizi normativi indiscutibili (Autio et al., 2024; OpenAI, 2023).

Un output insoddisfacente non dimostra quindi automaticamente che il prompt sia stato formulato male. Il problema può trovarsi nel prompt, ma anche nel modello, nelle informazioni, negli strumenti, nei criteri di valutazione o nella natura stessa del compito (Autio et al., 2024; Zamfirescu-Pereira et al., 2023).

Esempio
Supponiamo che un docente chieda a un LLM:
"Spiega il concetto di database."

Il sistema produce una risposta corretta ma molto generica:
"Un database è un insieme organizzato di dati che può essere utilizzato per memorizzare, gestire e recuperare informazioni."

La risposta non è necessariamente sbagliata.
Tuttavia il docente voleva utilizzarla in una lezione per studenti di scuola superiore che stanno iniziando a studiare informatica e si aspettava una spiegazione più intuitiva, con un esempio concreto e una distinzione tra database e semplice raccolta di file.

L'output insoddisfacente diventa quindi un segnale diagnostico.

Il problema potrebbe non essere una scarsa capacità del modello, ma una richiesta troppo vaga. Nel prompt non erano stati specificati:

- Il destinatario.
- Il livello di conoscenza iniziale.

- Lo scopo didattico.
- Il livello di approfondimento.
- Il tipo di esempio desiderato.
- I concetti che dovevano essere messi a confronto.

Il prompt può quindi essere rivisto in modo mirato:
“Spiega che cos’è un database a studenti di 16 anni che conoscono file e cartelle ma non hanno ancora studiato i database. Parti da un esempio concreto, come l’archivio degli studenti di una scuola, e spiega perché un database è diverso da una semplice raccolta di file. Usa un linguaggio accessibile ma tecnicamente corretto.”

Se il nuovo output risulta più adeguato, abbiamo ottenuto un’indicazione utile sulla causa del problema:
Output troppo generico → ipotesi: destinatario e scopo non definiti → modifica mirata del prompt → nuovo test

Supponiamo però che anche il secondo output contenga un errore, per esempio che affermi:
“Un database contiene sempre i dati in tabelle.”

In questo caso aggiungere altre indicazioni sul destinatario probabilmente non risolverebbe il problema.

La criticità riguarda invece la correttezza delle informazioni.
Potremmo quindi formulare una nuova ipotesi:
Output impreciso → possibile problema di conoscenza o informazioni disponibili → fornire una fonte affidabile o verificare esternamente

Il processo corretto non è “Output debole → aggiungere altre istruzioni” ma:
Output debole → formulare un’ipotesi sulla causa → modificare la componente interessata → ripetere il test

Anche la forma della risposta può fornire indizi sulla possibile origine del problema.

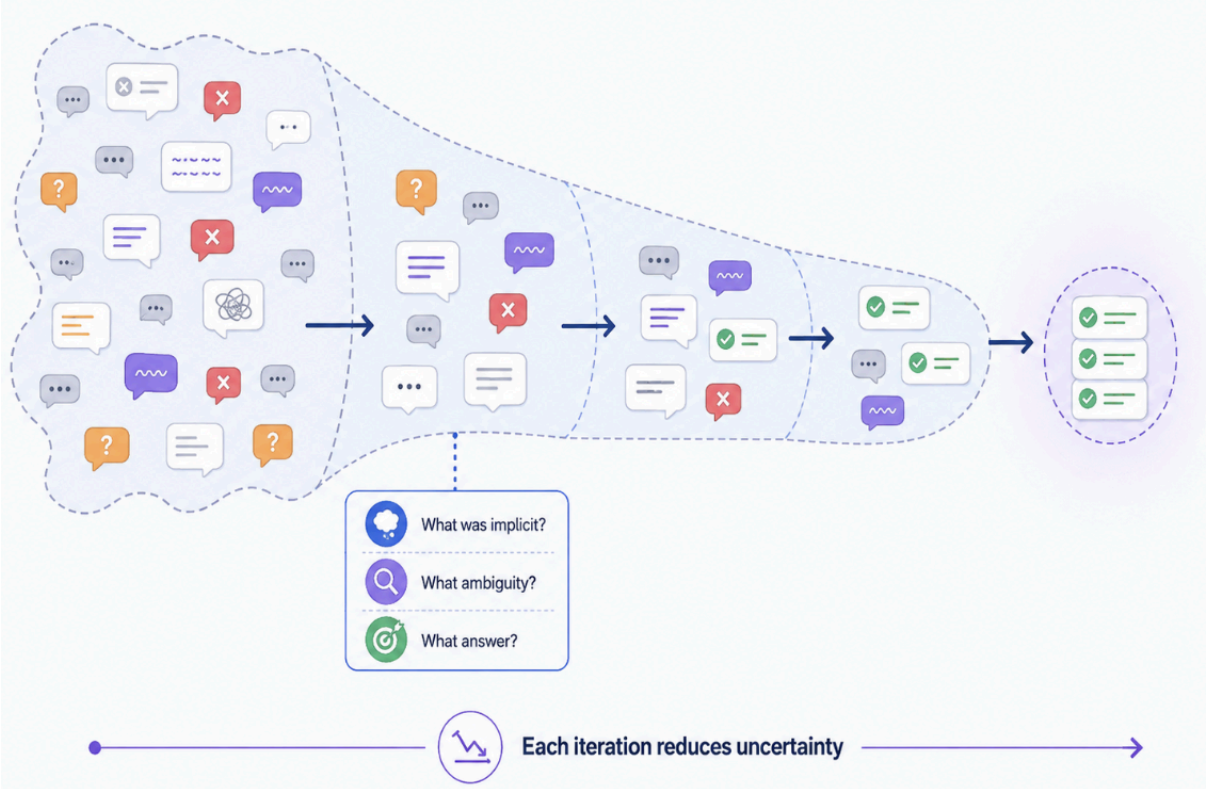
- Una risposta troppo lunga può suggerire che i vincoli sulla lunghezza fossero non sufficientemente definiti, poco espliciti, poco visibili o in conflitto con altre istruzioni.
- Una risposta banale può indicare che il dominio, il destinatario o il livello di approfondimento non erano stati definiti con sufficiente precisione.
- Una risposta tecnicamente corretta ma poco utile può segnalare uno scopo comunicativo poco chiaro o criteri di qualità insufficienti.
- Una risposta arbitraria può suggerire che il sistema sia stato spinto oltre il proprio spazio di affidabilità oppure che le informazioni disponibili non fossero adeguate.

Queste corrispondenze non costituiscono però diagnosi certe. Lo stesso risultato può avere cause differenti (Zamfirescu-Pereira et al., 2023).
L’output deve quindi essere interpretato come un insieme di indizi da verificare considerando l’intero contesto operativo.
In questi casi è utile chiedersi:

- Quale parte del compito era rimasta implicita?
- Quale ambiguità avevo lasciato aperta?
- Quale risultato stavo cercando realmente?
- Quali informazioni erano insufficienti?

- Quali vincoli erano in conflitto?
- Il modello e gli strumenti utilizzati erano adatti al compito?

Ogni iterazione efficace dovrebbe ridurre una specifica zona di indeterminatezza e restringere progressivamente lo spazio delle risposte indesiderate (Anthropic, n.d.-a; Madaan et al., 2023).



Un errore frequente consiste nel reagire a un output deludente aggiungendo istruzioni in modo caotico (Zamfirescu-Pereira et al., 2023).
Più testo, più vincoli e più richieste sovrapposte non producono necessariamente un prompt migliore.
Possono invece aumentare il rumore, nascondere le priorità e introdurre nuove contraddizioni (Anthropic, n.d.-b).
La revisione dovrebbe essere mirata.
A seconda del problema, può essere necessario:

- Chiarire la richiesta.
- Eliminare un dettaglio irrilevante.
- Aggiungere un’informazione mancante.
- Definire meglio il destinatario.
- Rendere esplicito lo scopo.
- Correggere un vincolo.
- Fornire un esempio.
- Modificare il formato richiesto.
- Sostituire una fonte.

- Utilizzare uno strumento differente.
- Suddividere il compito in più passaggi.

Modificare una componente alla volta permette di osservare meglio l'effetto della revisione. Cambiare simultaneamente molte parti del prompt rende invece più difficile comprendere quale intervento abbia realmente migliorato o peggiorato il risultato (Anthropic, n.d.-a).

Come avviene nella prototipazione, un risultato che non funziona non è necessariamente uno spreco.
Può rendere visibile una distanza tra ciò che il progettista intendeva ottenere e ciò che il sistema ha effettivamente prodotto.
Il fallimento non dimostra automaticamente che il prompt sia sbagliato, ma permette di formulare ipotesi sulle cause del problema e di intervenire in modo più consapevole (Madaan et al., 2023).
Può rendere visibili:

- Ambiguità concettuali.
- Informazioni rimaste implicite.
- Vincoli insufficienti o incompatibili.
- Limiti del modello.
- Problemi nelle fonti o negli strumenti.
- Criteri di valutazione non adeguati.

In questo senso, il fallimento può diventare un alleato del processo progettuale.
Interrompe l'illusione di correttezza, rende osservabili alcuni limiti del sistema e costringe il progettista a chiarire meglio il proprio obiettivo.

La prima risposta non dovrebbe quindi essere considerata automaticamente definitiva (Madaan et al., 2023).
Deve essere osservata, valutata e, quando necessario, sottoposta a revisione.

Il Prompt Engineering non consiste nel trovare una formula capace di produrre sempre il risultato corretto, ma nel progettare, osservare e migliorare progressivamente il modo in cui un compito viene presentato al sistema (Anthropic, n.d.-a; White et al., 2023; Zamfirescu-Pereira et al., 2023).

Materiale di supporto

Esercizi

Che cosa rende efficace un prompt?
Uno studente sostiene: «Un prompt è migliore quanto più è lungo e dettagliato.»

Analizza criticamente questa affermazione.

- Definisci che cosa si intende per prompt secondo la dispensa.
- Spiega perché il prompt non coincide con l'intero contesto operativo.
- Spiega perché il prompt non controlla rigidamente l'output, ma orienta le possibili continuazioni.
- Commenta l'affermazione: «Un buon prompt è strutturato, non prolisso.»
- Elenca almeno sei elementi che un prompt può rendere espliciti tra:
    - Oggetto.
    - Scopo.
    - Destinatario.
    - Dominio disciplinare.
    - Livello di approfondimento.
    - Informazioni da considerare.
    - Formato.
    - Vincoli.
    - Criteri di qualità.
- Spiega perché non è necessario inserire tutti questi elementi in ogni prompt.
- Descrivi un caso in cui un prompt molto breve può essere sufficiente.
- Descrivi invece un caso in cui una richiesta più articolata diventa necessaria.
- Spiega perché troppi dettagli possono introdurre rumore, duplicazioni o vincoli in conflitto.
- Concludi spiegando perché la struttura del prompt deve essere proporzionata alla complessità del problema.

Diagnosticare un output insoddisfacente
Un docente chiede a un sistema generativo: «Spiega che cos'è un database.»

Il sistema risponde:
«Un database è un insieme organizzato di dati utilizzato per memorizzare, gestire e recuperare informazioni.»

La risposta è corretta, ma il docente la considera poco utile per una classe di studenti di 16 anni.

- Spiega perché la risposta non può essere considerata semplicemente "sbagliata".
- Individua quali informazioni erano rimaste implicite nel prompt.
- Formula almeno tre possibili ipotesi sulla causa dell'insoddisfazione.
- Distingui tra problemi dovuti a:
    - Richiesta troppo vaga.
    - Destinatario non definito.

    - Scopo non definito.
    - Livello di approfondimento non definito.
    - Criteri di qualità insufficienti.
- Spiega perché la forma dell'output può fornire indizi diagnostici.
- Una risposta troppo lunga quale problema potrebbe suggerire?
- Una risposta banale quale problema potrebbe suggerire?
- Una risposta tecnicamente corretta ma poco utile quale problema potrebbe suggerire?
- Spiega perché questi indizi non costituiscono comunque una diagnosi certa.
- Immagina che, dopo aver migliorato il prompt, il sistema affermi: «Un database contiene sempre i dati in tabelle». Spiega perché aggiungere ulteriori indicazioni sul destinatario potrebbe non risolvere il problema.
- Indica quale potrebbe essere una revisione più appropriata in questo secondo caso.
- Concludi spiegando perché un output insoddisfacente può diventare uno strumento utile per migliorare il processo progettuale.

Dal prompt vago al prompt strutturato
Apri una nuova conversazione con un sistema generativo e inserisci: «Analizza questa interfaccia.»
Utilizza un'immagine di un sito, di un'app o di un'interfaccia digitale.
Salva la risposta ottenuta.
Successivamente crea una seconda versione del prompt specificando almeno:
- Oggetto dell'analisi.
- Prospettiva disciplinare.
- Tipo di utente.
- Aspetti da osservare.
- Scopo dell'analisi.
- Struttura dell'output.
- Livello di concretezza richiesto.
- Criteri di qualità.

Puoi utilizzare come riferimento la struttura proposta nella dispensa per l'analisi di usabilità.
Dopo aver ottenuto la seconda risposta:
- Confronta i due output.
- Individua quali decisioni il sistema ha dovuto prendere autonomamente nel primo caso.
- Individua quali ambiguità sono state eliminate nella seconda versione.
- Verifica se la seconda risposta è più pertinente al destinatario e allo scopo.
- Individua eventuali informazioni superflue presenti nel secondo prompt.
- Riduci il prompt eliminando ciò che ritieni non necessario.
- Esegui una terza prova.
- Confronta la seconda e la terza risposta.
- Valuta se la riduzione del prompt ha migliorato, peggiorato o lasciato invariata la qualità dell'output.
- Concludi indicando quali informazioni erano realmente necessarie e quali costituivano invece rumore.

Revisione iterativa controllata

Devi ottenere da un sistema generativo una spiegazione del concetto di algoritmo destinata a studenti di scuola superiore che conoscono già variabili e istruzioni semplici, ma non hanno ancora studiato formalmente gli algoritmi.

- Procedi per iterazioni.
- Prima prova.
- Scrivi un prompt iniziale e salva la risposta.

Successivamente:

- Individua un solo problema concreto nell'output.
- Formula un'ipotesi sulla sua causa.
- Modifica una sola componente del prompt.
- Esegui nuovamente la richiesta.
- Confronta il nuovo output con il precedente.
- Verifica se la modifica ha prodotto l'effetto previsto.

Ripeti il processo almeno tre volte utilizzando questa struttura:
Output → problema osservato → ipotesi sulla causa → modifica mirata → nuovo test → valutazione.

Durante le iterazioni prova a intervenire, quando necessario, su elementi differenti:

- Destinatario.
- Scopo.
- Livello di approfondimento.
- Esempi.
- Vincoli.
- Formato.
- Criteri di qualità.

Alla fine:

- Presenta il prompt iniziale e quello finale.
- Evidenzia tutte le modifiche effettuate.
- Spiega quale modifica ha prodotto il miglioramento maggiore.
- Indica almeno una modifica che non ha migliorato il risultato.
- Spiega perché modificare una componente alla volta rende più semplice comprendere la causa dei cambiamenti.
- Concludi spiegando la differenza tra iterare in modo progettuale e aggiungere istruzioni in modo caotico.

Bibliografia

Anthropic. (n.d.-a). *Define success criteria and build evaluations*. Claude Platform Docs. Retrieved September 4, 2026, from https://docs.anthropic.com/en/docs/build-with-claude/develop-tests

Anthropic. (n.d.-b). *Prompting best practices*. Claude Platform Docs. Retrieved September 4, 2026, from

https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/claude-4-best-practices

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*. Advances in Neural Information Processing Systems, 33, 1877–1901. https://doi.org/10.48550/arXiv.2005.14165

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

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B. P., Hermann, K., Welleck, S., Yazdanbakhsh, A., & Clark, P. (2023). *Self-refine: Iterative refinement with self-feedback*. arXiv. https://doi.org/10.48550/arXiv.2303.17651

OpenAI. (2023). *GPT-4 technical report*. arXiv. https://doi.org/10.48550/arXiv.2303.08774

Reynolds, L., & McDonell, K. (2021). *Prompt programming for large language models: Beyond the few-shot paradigm*. arXiv. https://doi.org/10.48550/arXiv.2102.07350

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention is all you need*. Advances in Neural Information Processing Systems, 30. https://doi.org/10.48550/arXiv.1706.03762

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

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 437, pp. 1–21). Association for Computing Machinery. https://doi.org/10.1145/3544548.3581388

Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., & Ba, J. (2022). *Large language models are human-level prompt engineers*. arXiv. https://doi.org/10.48550/arXiv.2211.01910

Approfondimenti

- Bonaventura Di Bello, Scrivere prompt efficaci For Dummies, Hoepli, 2026 — È probabilmente il testo che sceglierei come prosecuzione più diretta della dispensa. Parte dai fondamenti del contesto e della struttura del prompt, propone un processo di sviluppo e miglioramento delle richieste e arriva a framework e meta-prompt. È

particolarmente pertinente alla tua idea del prompt come struttura informativa progettata, piuttosto che come semplice domanda formulata “bene”.

- 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 — Approfondisce anatomia e metodologia del prompt, confronta prompt efficaci e inefficaci e introduce tecniche come zero-shot, few-shot, prompt chaining e ReAct. È particolarmente utile per sviluppare il principio della dispensa secondo cui la struttura deve essere proporzionata al problema e deve ridurre le ambiguità realmente rilevanti, evitando di aggiungere dettagli senza criterio.
- Flavio Corradini, Il prompt responsabile. Linguaggio e metodo per governare davvero l’IA generativa, 1ª ed., FrancoAngeli, 2026 — Affronta esplicitamente obiettivi, contesto, destinatari, strutturazione dell’output, iterazione, template ed esempi. È forse il libro più vicino alla dimensione metodologica della tua dispensa, perché rifiuta l’idea dei “prompt magici” e considera il prompting come un processo da progettare, osservare e correggere mantenendo la responsabilità umana sul risultato.
- Riccardo Ruffoli, Le basi del prompting. Manuale pratico per dialogare con l’intelligenza artificiale, Edizioni ETS, 2025 — È un testo breve e introduttivo, adatto soprattutto agli studenti. Organizza il prompting attorno a chiarezza e architettura della richiesta, specificità delle informazioni, stile, interazione e gestione dei compiti complessi, accompagnando la teoria con esempi concreti. Si presta quindi bene come lettura complementare a un capitolo di fondamenti.
- Sergio Sentinelli e Alessandro Placa, Fare la domanda giusta. L’arte di lavorare con ChatGPT e le AI, Apogeo, 2024 — Offre un approccio operativo al Prompt Engineering con metodologie, esempi e applicazioni professionali. È meno sistematico dei testi del 2026, ma è utile per osservare come obiettivo, formulazione della richiesta e contesto applicativo cambino concretamente l’utilità dell’output nei diversi domini professionali.

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.

