

Sommario

1. Introduzione	2
2. Fondamenti del Prompt Engineering.....	2
3. Tecniche Avanzate di Prompt Engineering.....	3
4. Prompting Multimodale: L'AI oltre il Testo	6
5. AI-Powered Prompt Generation: Prompt Auto-Generati	6
6. Framework e Strategie per Scrivere Prompt Efficaci	7
7. Il Framework di Greg Brockman.....	14
8. Strumenti e Risorse per il Prompt Engineering	16
9. Applicazioni Pratiche nei Vari Settori.....	18
10. Futuro del Prompt Engineering e Conclusioni	20

Prompt Engineering per l'Intelligenza Artificiale: Principi, Tecniche, Strategie, Framework e Applicazioni

1. Introduzione

Il **Prompt Engineering** è la disciplina che si occupa di progettare e ottimizzare i *prompt*, ovvero le istruzioni o domande fornite a un modello di intelligenza artificiale, per ottenere risposte pertinenti e utili. In altre parole, è l'arte di tradurre l'intento umano in un input comprensibile ed efficace per l'AI ([What is Prompt Engineering? A Detailed Guide For 2025 | DataCamp](#)). Questa pratica è diventata essenziale con la diffusione dei modelli di *linguaggio esteso* (LLM) come GPT-4, perché il modo in cui formuliamo una richiesta può fare la differenza tra una risposta accurata e un risultato fuori bersaglio ([What is Prompt Engineering? A Detailed Guide For 2025 | DataCamp](#)). Un prompt ben costruito funge da **ponte** tra l'intenzione dell'utente e la risposta della macchina, garantendo che l'AI comprenda il contesto e fornisca output utili.

L'importanza dei prompt si riflette nell'impatto su **produttività e creatività**. Un esempio semplice viene dall'uso quotidiano di assistenti vocali: chiedere *"Mettili un po' di musica rilassante"* rispetto a *"Riproduci la Sinfonia di Beethoven"* produce risultati molto diversi ([What is Prompt Engineering? A Detailed Guide For 2025 | DataCamp](#)). Nel contesto aziendale, un prompt chiaro può guidare un modello generativo a elaborare un riassunto preciso di un report, far scrivere un codice funzionante, o generare idee creative per una campagna di marketing. In tutti questi casi, prompt migliori significano risposte più pertinenti, risparmiando tempo e iterazioni.

Il Prompt Engineering ha un impatto crescente in **vari settori della vita quotidiana**. Nello sviluppo software, ad esempio, sviluppatori usano prompt accurati per far generare codice o trovare bug all'AI, accelerando il ciclo di programmazione. Nel marketing, professionisti della comunicazione formulano prompt studiati per ottenere testi pubblicitari accattivanti o immagini generate dall'AI conformi al *brand*. Persino utenti comuni, interagendo con chatbot avanzati come ChatGPT, beneficiano di prompt ben pensati per ottenere consigli, spiegazioni o contenuti personalizzati. In sintesi, man mano che l'AI si integra nelle nostre attività, la capacità di guidarla efficacemente tramite prompt diventa una competenza chiave sia per esperti sia per utenti comuni.

2. Fondamenti del Prompt Engineering

Per comprendere il Prompt Engineering, partiamo dalle basi: **che cos'è un prompt e come interagisce con un modello AI**. Un *prompt* è qualsiasi input testuale (domanda, comando, contesto) che forniamo a un modello di AI per ottenere una risposta. Nei modelli di linguaggio come GPT, il prompt condiziona la continuazione generata dal modello. Un prompt può essere semplice, ad esempio *"Riassumi questo testo"*, oppure molto elaborato, con istruzioni dettagliate e contesto aggiuntivo. In ogni caso, il modello analizza il prompt e, in base ai suoi parametri e ai dati con cui è stato addestrato, produce un output che ritiene probabilmente coerente con la richiesta.

Ci sono differenze importanti tra **prompt semplici e prompt avanzati**. Un *prompt semplice* di solito è breve e diretto, ma rischia di essere ambiguo o di non fornire abbastanza contesto. Questo può portare l'AI a generare risposte generiche o *inaspettate* se l'istruzione è troppo vaga. D'altro canto, un *prompt avanzato* include maggiori dettagli: può specificare un ruolo per l'AI, delimitare il formato della risposta, fornire esempi o passi da seguire. Tuttavia, anche qui serve equilibrio – un prompt eccessivamente complesso o lungo può confondere il modello ([Implementing advanced prompt engineering with Amazon Bedrock | AWS Machine Learning Blog](#)). La chiave è **bilanciare semplicità e completezza**, fornendo contesto sufficiente ma in modo chiaro e strutturato. Ad esempio, invece di chiedere genericamente *"Parlami dell'energia rinnovabile"*, un prompt più avanzato potrebbe essere *"Immagina di essere un esperto ambientale. Spiega in termini semplici l'importanza dell'energia solare e eolica, in circa 3 paragrafi, includendo almeno una statistica"*

recente". Quest'ultimo prompt definisce un ruolo, un tono, un ambito e perfino la lunghezza desiderata, aumentando le probabilità di ottenere una risposta utile e mirata.

Un concetto fondamentale nel Prompt Engineering è l'**in-context learning**, spesso declinato nei termini di *zero-shot* e *few-shot prompting*. Con il *zero-shot prompting*, si pone al modello un compito senza fornire alcun esempio: il modello deve generalizzare alla domanda usando solo ciò che ha appreso durante l'addestramento ([What is Prompt Engineering? A Detailed Guide For 2025 | DataCamp](#)). Ad esempio, chiedere *"Traduci in francese: Il gatto è sul tavolo"* è uno scenario *zero-shot* (nessun esempio di traduzione fornito). Nel *few-shot prompting*, invece, si includono nel prompt alcuni esempi di domanda-risposta pertinenti, così che il modello possa inferire il formato o la logica richiesta ([What is Prompt Engineering? A Detailed Guide For 2025 | DataCamp](#)). Ad esempio, per aiutare un modello a tradurre correttamente, potremmo fornire due frasi in italiano con la rispettiva traduzione francese, seguite da una nuova frase da tradurre. Questa tecnica sfrutta la capacità del modello di cogliere il *pattern* dai pochi esempi contestuali (da qui il termine *few-shot*). Una variante particolare è l'**one-shot prompting**, dove si fornisce un singolo esempio: funziona come via di mezzo, utile quando abbiamo un solo esempio rappresentativo di ciò che vogliamo.

Un altro concetto chiave è il **Chain-of-Thought (CoT) prompting**. Questa tecnica avanzata consiste nel fare in modo che l'AI espliciti una catena di ragionamenti passo-passo per arrivare alla risposta ([What is Prompt Engineering? A Detailed Guide For 2025 | DataCamp](#)). In pratica, anziché rispondere direttamente, il modello viene incoraggiato (tramite il prompt) a "pensare ad alta voce", enumerando le considerazioni intermedie. Ad esempio, per un problema matematico complesso, potremmo aggiungere al prompt la frase *"Ragiona passo per passo"* o fornire un esempio in cui la soluzione è spiegata a tappe. Il CoT sfrutta il fatto che i modelli di grandi dimensioni mostrano abilità emergenti di ragionamento se guidati opportunamente: chiedendo loro di spiegare il percorso logico, spesso otteniamo risposte più accurate su compiti difficili perché l'AI "si auto-corregge" lungo il tragitto. È stato dimostrato che aggiungere semplicemente frasi come *"Pensiamo passo per passo"* può far aumentare significativamente l'accuratezza in compiti aritmetici o di logica ([Chain-of-Thought Prompting | Prompt Engineering Guide](#)) ([Chain-of-Thought Prompting | Prompt Engineering Guide](#)). Dunque, il chain-of-thought prompting è uno strumento potente per sbloccare capacità di *reasoning* latenti nei modelli AI.

3. Tecniche Avanzate di Prompt Engineering

Con solide basi sui prompt, esistono numerose **tecniche avanzate** che i professionisti sfruttano per ottenere il massimo dai modelli AI. Di seguito approfondiamo alcune delle strategie più all'avanguardia:

1. **Chain-of-Thought Prompting (CoT)** – Come accennato, il CoT spinge il modello a fornire una spiegazione passo-passo prima di giungere alla conclusione. Questa tecnica, specie su modelli di grandi dimensioni, migliora la capacità di risolvere problemi complessi di matematica, logica o comprensione, perché simula un processo di ragionamento umano ([What is Prompt Engineering? A Detailed Guide For 2025 | DataCamp](#)). Ad esempio, per una domanda di ragionamento, un prompt CoT potrebbe terminare con: *"Vediamo come arrivarci passo per passo."* Indurre l'AI a scomporre il problema aiuta a individuare errori lungo il percorso e ad arrivare a risposte più corrette.
2. **Few-shot e One-shot Prompting** – L'uso di esempi nel prompt è un'arte in sé. Nel *few-shot prompting*, fornire 2-5 esempi rappresentativi può orientare fortemente il modello. Ad esempio, se stiamo sviluppando un assistente che risponda in modo formale, potremmo includere alcune domande di esempio con risposte formalmente strutturate. Il modello coglierà lo stile e lo replicherà. Con l'*one-shot*, a volte un

singolo esempio particolarmente ben scelto può bastare a impostare il tono o il format desiderato. Queste tecniche sono utili quando non vogliamo (o non possiamo) fine-tuning del modello, ma desideriamo comunque personalizzare la sua risposta in base a pochi esempi concreti.

3. **Role-based Prompting** – In questa strategia si assegna esplicitamente un ruolo o una persona virtuale all'AI all'interno del prompt. Ad esempio: *“Sei un avvocato esperto di diritto civile: spiega questo contratto in termini semplici...”*. Indicando un ruolo professionale (medico, insegnante, consulente, ecc.), il modello cercherà di adattare il registro e le conoscenze alla persona indicata ([What is Prompt Engineering? A Detailed Guide For 2025 | DataCamp](#)). Questo spesso produce output più pertinenti al contesto: un *role prompt* da “chef italiano” farà emergere ricette in stile tradizionale, mentre uno da “personal trainer” porterà consigli sul fitness, e così via. Il role-based prompting sfrutta il vasto corpus di conoscenza contestuale dei modelli, attivando il subset di informazioni e tono di voce pertinenti a quel ruolo.
4. **Self-Consistency Prompting** – Si tratta di una tecnica particolare che mira a migliorare l'accuratezza facendo leva sul concetto di maggioranza delle risposte. In pratica, si esegue il **medesimo prompt più volte** (ad esempio con leggeri random seed diversi, sfruttando la probabilità nelle risposte) e poi si adotta come risultato finale quello più comune o “consistente” fra quelli generati ([Self-Consistency Prompting: Enhancing AI Accuracy](#)) ([Self-Consistency Prompting: Enhancing AI Accuracy](#)). L'idea, proposta originariamente per potenziare il chain-of-thought, è che facendo “pensare” più volte l'AI, si possano mediare gli errori casuali. Ad esempio, se su 5 ragionamenti step-by-step diversi, tre convergono sulla risposta X e due sulla risposta Y, si assume che X sia quella corretta (auto-consistente). Questo metodo ha dimostrato di aumentare la **affidabilità** in compiti di ragionamento aritmetico e buon senso, riducendo risposte isolate errate dovute a variazioni nelle generazioni ([Self-Consistency Prompting: Enhancing AI Accuracy](#)) ([Self-Consistency Prompting: Enhancing AI Accuracy](#)).
5. **Tree-of-Thought Prompting (ToT)** – È un'evoluzione del chain-of-thought. Invece di un unico filo di ragionamento lineare, il modello viene guidato a esplorare **diverse ramificazioni** di pensiero, un po' come valutare vari scenari possibili nella soluzione di un problema ([Tree of Thoughts Prompting; How Does it Enhance AI Results?](#)) ([Tree of Thoughts Prompting; How Does it Enhance AI Results?](#)). Il prompt in questo caso può incoraggiare la generazione di più soluzioni candidate e una riflessione su di esse. Ad esempio, si può dire: *“Proponi tre possibili soluzioni e valuta quale sia la migliore.”* Il modello così genera un albero di considerazioni (diverse ipotesi) e le passa in rassegna. Questa tecnica ricorda una *ricerca in ampiezza/profondità*: il modello può “tornare indietro” se una strada pare portare a un vicolo cieco, e provare un'altra via ([Tree of Thoughts \(ToT\) | Prompt Engineering Guide](#)) ([Tree of Thoughts Prompting; How Does it Enhance AI Results?](#)). Il risultato è una capacità di **pianificazione migliorata**: il ToT permette di pianificare, fare *lookahead* e *backtracking*, trovando soluzioni più solide per problemi complessi rispetto al solo CoT lineare. Ad esempio, nella risoluzione di puzzle o nelle decisioni strategiche, un prompt ToT consente all'AI di valutare pro e contro di differenti approcci prima di finalizzare la risposta, simulando un processo deliberativo più approfondito.
6. **ReAct (Reason + Act)** – Questa è una metodologia che combina il ragionamento con la capacità di compiere azioni, tipicamente usata per agenti AI che possono interagire con strumenti esterni. In un framework ReAct, il prompt è strutturato affinché il modello produca alternatamente **tracce di ragionamento** (*Thoughts*) e **azioni** (*Acts*) ([ReAct Prompting | Prompt Engineering Guide](#)). Ad esempio, un

agente potrebbe ragionare sul da farsi (“Thought: la domanda richiede informazione X, potrei cercarla online”) e poi eseguire un’azione ambiente (“Act: [RICERCA] X”). Il prompt ReAct guida il modello a seguire questo schema, permettendogli di consultare fonti esterne o una base di conoscenza durante l’elaborazione della risposta ([ReAct Prompting | Prompt Engineering Guide](#)) ([ReAct Prompting | Prompt Engineering Guide](#)). Ciò porta a risultati più accurati e *factually correct* perché l’AI non si affida solo alla conoscenza interna (che potrebbe essere incompleta o obsoleta), ma la arricchisce con informazioni recuperate in tempo reale. In sostanza, ReAct trasforma il modello in un agente che **pensa e agisce**: pensa per decidere la prossima azione, agisce (es. chiamando un’API o effettuando un calcolo), ottiene un’osservazione, poi riprende a pensare, in un ciclo. Questa tecnica ha mostrato risultati all’avanguardia in compiti come domande su conoscenze esterne e decision-making, migliorando anche la trasparenza (il *chain of thought* è visibile) ([ReAct Prompting | Prompt Engineering Guide](#)) ([ReAct Prompting | Prompt Engineering Guide](#)).

7. **Self-Reflection Prompting** – Questa tecnica sfrutta la capacità del modello di rivedere e migliorare le proprie risposte. Indurre l’AI a valutare criticamente l’output prodotto permette di ottenere risposte più coerenti e accurate. **Esempio di prompt:**
*"Riformula la tua risposta migliorandone la chiarezza e l'accuratezza."
"Individua eventuali inesattezze nella tua risposta e correggile."*
8. **Meta-Prompting** – Il Meta-Prompting è una strategia che utilizza un prompt per generare altri prompt ottimizzati, migliorando la qualità delle richieste formulate all’AI. **Esempio di prompt:**
*"Scrivi un prompt per ottenere un elenco di strategie di digital marketing efficaci per il 2025."
"Genera tre diverse versioni di un prompt per ottenere il massimo da un modello AI per la scrittura di articoli."*
9. **Adaptive Prompting** – L’Adaptive Prompting consente al modello di adattarsi dinamicamente al contesto della conversazione, ottimizzando la risposta in base alle richieste ricevute. **Esempio di prompt:**
*"Se la mia richiesta è troppo ampia, chiedimi di restringerla. Se è troppo vaga, chiedimi di essere più specifico."
"Fornisci una risposta breve se la mia domanda è diretta, e una risposta dettagliata se è più complessa."*
10. **Progressive Prompting** – Il Progressive Prompting suddivide richieste complesse in più fasi, consentendo al modello di elaborare il contesto in modo più efficace. **Esempio di prompt:**
"Prima fornisci un elenco di punti chiave. Poi sviluppa ciascun punto in un paragrafo separato."
11. **Retrieval-Augmented Generation (RAG)** – Il RAG non è esattamente un formato di prompt, ma una strategia di architettura in cui il prompt viene arricchito con informazioni recuperate da un **database o corpus esterno**. In pratica, quando arriva una domanda, prima di generare la risposta si effettua una ricerca (ad esempio su documenti aziendali, su internet, o in un wiki interno) e si **inietta nel prompt** il contesto trovato, seguito dalla domanda originale. Questo consente al modello generativo di avere dati aggiornati o specifici su cui basare la risposta, invece di fare affidamento solo sulla propria memoria addestrata. Ad esempio, per rispondere a “Quali sono le ultime normative fiscali per le PMI?”, un sistema RAG potrebbe cercare nei documenti di legge aggiornati e poi fornire tali estratti al modello con un prompt del tipo: “Alla luce dei seguenti articoli di legge: [testo normativa] ... rispondi alla domanda: Quali sono ...?”. Il risultato è un output più preciso e fedele alle fonti. Questo approccio **aumenta l’accuratezza e la**

pertinenza delle risposte generative con fatti aggiornati ([What is RAG? - Retrieval-Augmented Generation AI Explained - AWS](#)). Molti assistenti avanzati (come ad esempio i chatbot integrati nei motori di ricerca, o gli helper su base documentale) utilizzano RAG per combinare le capacità linguistiche dell'LLM con un modulo di ricerca, ottenendo il meglio dei due mondi: generatività e affidabilità delle informazioni.

4. Prompting Multimodale: L'AI oltre il Testo

L'evoluzione dei modelli multimodali (es. GPT-4o, Gemini, Sora) ha introdotto nuove sfide e opportunità nel Prompt Engineering, consentendo di combinare testo, immagini, audio e video per ottenere output più sofisticati.

a. Hybrid Prompting (Prompting Ibrido)

Il Prompting Ibrido integra più modalità di input per ottenere risultati più ricchi e precisi.

Esempio di prompt:

"Analizza questa immagine e fornisci una spiegazione dettagliata del contesto."

"Esamina questo grafico e crea un riepilogo in 3 punti."

b. Multi-Agent Prompting

Il Multi-Agent Prompting sfrutta la collaborazione tra diversi modelli AI per ottimizzare la generazione dei contenuti.

Esempio di utilizzo:

- Un agente AI genera un prompt per un modello di visione artificiale, che poi fornisce input a un modello di linguaggio per una spiegazione dettagliata.

c. Cross-Modal Prompting

Nel Cross-Modal Prompting, un input testuale viene usato per generare output in formati diversi, come immagini o file audio.

Esempio di prompt:

"Crea un'infografica basata su questa descrizione testuale."

"Genera una sintesi audio di questa conversazione."

5. AI-Powered Prompt Generation: Prompt Auto-Generati

L'AI può essere utilizzata per generare prompt migliori in modo automatico, migliorando ulteriormente l'interazione uomo-macchina.

a. Meta-Prompting Avanzato

L'AI genera più versioni di un prompt e seleziona quella più efficace.

Esempio di prompt:

"Scrivi tre versioni alternative di questo prompt e scegli quella più efficace."

b. Self-Improving Prompts

Il modello analizza la propria risposta e riformula il prompt per ottenere una versione più precisa.

Esempio di prompt:

"Analizza la risposta generata e riformula il prompt per ottenere una risposta più dettagliata."

c. Dynamic Prompting

Il modello modifica dinamicamente il prompt in base al contesto e ai dati disponibili.

Esempio di prompt:

"Se l'utente chiede informazioni su un argomento complesso, proponi un approccio step-by-step."

d. AI-Guided Prompting

L'AI guida l'utente a scrivere prompt migliori, rendendo l'interazione più efficace.

Esempio di prompt:

"Chiedimi dettagli aggiuntivi prima di generare la risposta, se il prompt è troppo generico."

6. Framework e Strategie per Scrivere Prompt Efficaci

Nel tempo, sono emersi numerosi framework strutturati per la scrittura di prompt, pensati per aiutare gli utenti a non dimenticare elementi importanti e a organizzare le istruzioni in modo ottimale.

Questi framework forniscono un modello strutturato che guida l'utente nella formulazione del prompt, assicurando che l'AI riceva informazioni chiare, contestualizzate e ben formattate, riducendo ambiguità e iterazioni inutili.

In questa sezione, categorizziamo i framework in cinque principali gruppi, a seconda della complessità della richiesta e del livello di dettaglio richiesto.

a. Classic Frameworks (Framework Classici)

- i. **ERA – Expectation, Role, Action (Aspettativa, Ruolo, Azione):** Si concentra sul definire chiaramente cosa ci si aspetta dall'AI, qual è il ruolo che deve assumere e quali azioni deve compiere ([ERA framework - Expectation-driven AI prompt engineering](#)). Ad esempio, un prompt secondo ERA potrebbe essere: *"Aspettativa: ottenere una strategia di social media efficace. Ruolo: sei un consulente di marketing digitale. Azione: analizza i trend attuali e proponi un piano trimestrale per aumentare l'engagement del 30%."* Questo framework enfatizza l'allineamento dell'output alle **aspettative dell'utente**, contestualizzando il modello in un certo ruolo e dandogli istruzioni operative.
- ii. **APE – Action, Purpose, Expectation (Azione, Scopo, Aspettativa)** Struttura il prompt partendo dall'**azione** richiesta, lo **scopo** dell'output e le **aspettative** dell'utente. **Esempio**
"Azione: scrivi un articolo di 500 parole. Scopo: spiegare le basi della programmazione Python. Aspettativa: usa un linguaggio semplice e fornisci esempi pratici."

- iii. **TAE – Task, Action, Expectation (Compito, Azione, Aspettativa)**
Simile a ERA, ma enfatizza il **compito** primario che l'AI deve svolgere prima di definire l'azione. **Esempio**
"Compito: analizzare un dataset di vendite. Azione: individuare le tendenze chiave degli ultimi 12 mesi. Aspettativa: restituire una tabella con le cinque categorie di prodotti con la crescita più alta."

b. Strategic Frameworks (Framework Strategici)

- i. **CARE – Context, Action, Result, Example (Contesto, Azione, Risultato, Esempio)**: Il framework CARE aiuta a strutturare prompt dettagliati e azionabili, spesso usato per analisi di casi o report ([CARE framework - AI prompt engineering with precision - Juuzt AI](#)) ([ChatGPT Prompt Frameworks | Shelly Palmer](#)). Ad esempio: *"Contesto: la nostra azienda ha registrato un calo di vendite nell'ultimo trimestre. Azione: analizza i possibili motivi e suggerisci strategie di miglioramento. Risultato: vogliamo un piano in 3 punti per aumentare le vendite del 15% nel prossimo trimestre. Esempio: considera un caso simile in cui una campagna di marketing ha rilanciato le vendite di un prodotto tech."* Seguire CARE garantisce che forniamo all'AI il **contesto** necessario, chiediamo un'azione specifica, indichiamo il risultato atteso e, se possibile, includiamo un esempio per guidare la risposta.
- ii. **RACE – Role, Action, Context, Expectation (Ruolo, Azione, Contesto, Aspettativa)**: Framework semplice ed efficace per molti compiti, molto adottato in ambito marketing e non solo ([11 Must-Know ChatGPT Prompt Frameworks for Marketers | ButterCMS](#)). Con RACE prima si definisce il **ruolo** dell'AI, poi l'azione o compito, quindi si fornisce contesto aggiuntivo e infine si specifica l'aspettativa di output. Un esempio di prompt RACE potrebbe essere: *"Ruolo: Sei un analista finanziario esperto di borsa. Azione: Fornisci consigli di investimento sul titolo XYZ. Contesto: L'azienda ha appena pubblicato una trimestrale con utili oltre le attese ma un calo di ricavi in Europa. Aspettativa: Voglio un'analisi di 2 paragrafi sui pro e contro dell'acquisto di questo titolo ora, con una conclusione chiara."* La struttura RACE obbliga a dare all'AI una *missione* chiara e circostanze definite, più un'idea precisa di cosa vogliamo ottenere.
- iii. **RISE – Role, Input, Steps, Expectation (Ruolo, Input, Passaggi, Aspettativa)**
Questo framework enfatizza i **passaggi** da seguire per ottenere la risposta. **Esempio**
"Ruolo: Sei un esperto SEO. Input: un elenco di parole chiave. Passaggi: 1) Analizza la concorrenza, 2) Suggerisci strategie per migliorare il ranking, 3) Indica tool utili. Aspettativa: un report con consigli dettagliati."

c. Enhanced Frameworks (Framework Avanzati)

- i. **ROSES – Role, Objective, Scenario, Expected Solution, Steps (Ruolo, Obiettivo, Scenario, Soluzione Attesa, Passi)**: Framework utile soprattutto in ambito progettuale e decisionale ([ROSES framework - Blossoming AI prompt engineering strategies](#)). Ogni componente aggiunge un tassello: ruolo dell'AI, obiettivo da raggiungere, scenario o contesto specifico,

descrizione della soluzione attesa e indicazione dei passi da seguire. Un prompt costruito con ROSES potrebbe essere: *“Ruolo: Agisci come un project manager esperto in IT. Obiettivo: Elaborare un piano di progetto per implementare una nuova piattaforma e-commerce. Scenario: L'azienda ha 3 mesi di tempo e un budget limitato, team distribuito in remoto. Soluzione attesa: Un piano sintetico con milestone chiave e allocazione risorse ottimizzata. Passi: 1) Elenca i requisiti principali, 2) Definisci le fasi progettuali con timeline, 3) Suggerisci misure per mitigare rischi di ritardo.”*

Utilizzando ROSES, il prompt copre tutto il **processo** da contesto a esecuzione, guidando l'AI a rispondere in modo strutturato e completo.

- ii. **COAST – Context, Objective, Actions, Scenario, Task** : Il framework **COAST** è particolarmente utile quando si ha la necessità di costruire **prompt orientati all'azione** e basati su un contesto ben definito.

Struttura del Framework COAST

- 1) **Context (Contesto)** → Definisce la situazione iniziale, le informazioni di background e i vincoli rilevanti.
- 2) **Objective (Obiettivo)** → Determina lo scopo finale della richiesta.
- 3) **Actions (Azioni)** → Specifica cosa deve fare il modello AI.
- 4) **Scenario (Scenario)** → Fornisce dettagli aggiuntivi che possono influenzare il risultato.
- 5) **Task (Compito)** → Indica il compito specifico da svolgere.

Esempio di utilizzo del framework COAST

Prompt generico:

"Scrivi un'analisi sui cambiamenti climatici."

Prompt ottimizzato con COAST:

"Context: Le emissioni di gas serra sono aumentate del 15% negli ultimi dieci anni.

Objective: Fornire un'analisi dettagliata dei principali fattori che hanno contribuito a questa crescita.

Actions: Identificare le cause, valutare l'impatto su scala globale e proporre soluzioni attuabili.

Scenario: Consideriamo tre principali settori industriali (trasporti, energia, manifattura) e le politiche internazionali esistenti.

Task: Generare un report di 1000 parole con grafici esplicativi e una sezione sulle prospettive future."

Perché funziona?

Il **contesto** chiarisce la situazione e le informazioni di background.

L'**obiettivo** specifica lo scopo del prompt.

Le **azioni** delineano i compiti precisi che il modello deve eseguire.

Lo **scenario** restringe il focus a tre settori specifici, eliminando ambiguità.

Il **compito** finale definisce la struttura e il formato dell'output.

Quando usare COAST?

Creazione di report strutturati.

Analisi di scenari complessi.

Pianificazione strategica e business intelligence.

- iii. **TRACE – Task, Request, Actions, Context, Example** : Il framework **TRACE** è particolarmente utile per la progettazione di **prompt in ambito tecnico**,

scientifico e data-driven, in cui è necessario fornire esempi e specifiche precise.

Struttura del Framework TRACE

- 1) **Task (Compito)** → La richiesta principale da svolgere.
- 2) **Request (Richiesta)** → I dettagli esatti su cosa deve fare l'AI.
- 3) **Actions (Azioni)** → I passaggi specifici che l'AI deve eseguire.
- 4) **Context (Contesto)** → Le informazioni di background che aiutano il modello a rispondere con maggiore precisione.
- 5) **Example (Esempio)** → Un esempio concreto di output desiderato.

Esempio di utilizzo del framework TRACE

Prompt generico:

"Analizza i dati delle vendite e fornisci insight."

Prompt ottimizzato con TRACE:

*"Task: Analizzare il dataset di vendite dell'ultimo trimestre.

Request: Identificare i prodotti con il maggiore incremento e decremento di vendite.

Actions: 1) Eseguire un'analisi delle variazioni percentuali. 2) Generare un report con una tabella comparativa. 3) Evidenziare eventuali anomalie nei dati.

Context: I dati sono relativi a un'azienda di e-commerce con una distribuzione geografica su più paesi.

Example: Il report dovrebbe avere la seguente struttura:

- Intestazione con un riepilogo dei dati.
- Tabella con i prodotti e le variazioni di vendita.
- Sezione di insight con un'analisi delle tendenze."*

Perché funziona?

- Il **compito** (Task) è chiaro e ben definito.
- La **richiesta** (Request) specifica i criteri dell'analisi.
- Le **azioni** (Actions) forniscono un workflow strutturato.
- Il **contesto** (Context) riduce ambiguità sulle informazioni disponibili.
- L'**esempio** (Example) fornisce un modello per garantire che l'output sia conforme alle aspettative.

Quando usare TRACE?

- Data science e analisi statistica.
- Generazione di report dettagliati.
- Creazione di prompt altamente strutturati con output formattati.

d. Advanced Frameworks (Framework Tecnici Avanzati)

- i. **PRIME** – Questo framework pone enfasi sul *"priming"* dell'AI con le informazioni iniziali e sulla definizione di un processo strutturato. Anche se meno standardizzato degli altri, in genere include elementi come presentare il **problema o scopo** (Problem/Purpose), fornire un contesto o *background* per *priming* (dando al modello informazioni iniziali rilevanti), istruire sul **ruolo** o approccio (ad esempio "Interpreta questi dati..."), e spesso una parte di esempio o verifica del risultato atteso. L'idea di fondo del PRIME (da *to prime*, attivare) è assicurarsi che l'AI sia ben "sintonizzata" sul compito prima

di fornirle la domanda finale. Ad esempio: *“Scopo: Vogliamo un nuovo slogan pubblicitario. Contesto: azienda giovane, settore eco-sostenibile, pubblico under 30. Istruzione: Genera 5 slogan creativi e brevi che trasmettano innovazione verde. Esempio: (forniamo uno slogan esistente per far capire lo stile).”* Sebbene il termine esatto delle lettere possa variare, l'approccio PRIME ricorda di **preparare il modello** con contesto e istruzioni chiare prima di richiedere l'output.

- ii. **FOCUS – Function, Outcome, Context, Usage, Specifics** : Il framework FOCUS è progettato per la creazione di prompt altamente mirati, che richiedono una struttura ben definita e dettagliata. È particolarmente efficace in scenari in cui il contesto tecnico, il risultato desiderato e le specifiche del compito sono essenziali per ottenere un output preciso.

Struttura del Framework FOCUS

- 1) **Function** (Funzione) → Definisce il ruolo o la funzione del modello AI nel contesto del prompt.
- 2) **Outcome** (Risultato) → Stabilisce con chiarezza quale risultato si vuole ottenere.
- 3) **Context** (Contesto) → Fornisce le informazioni di background necessarie per l'elaborazione della risposta.
- 4) **Usage** (Utilizzo) → Specifica come il risultato verrà utilizzato o integrato in un flusso di lavoro.
- 5) **Specifics** (Specifiche) → Delinea i dettagli tecnici, i vincoli e i requisiti dell'output.

Esempio di utilizzo del framework FOCUS

Prompt generico:

"Scrivi un algoritmo di machine learning per classificare immagini."

Prompt ottimizzato con FOCUS:

"Function: Sei un esperto di machine learning e il tuo compito è generare un modello di classificazione immagini.

Outcome: Fornisci un codice Python che implementa una rete neurale convoluzionale (CNN) per classificare immagini in 5 categorie.

Context: Il dataset è composto da 10.000 immagini etichettate, bilanciate tra le classi. I dati devono essere pre-processati con normalizzazione e augmentation.

Usage: Il codice sarà utilizzato per addestrare un modello in TensorFlow/Keras con un set di test separato.

Specifics: Il modello deve essere scritto in TensorFlow, con una struttura CNN di almeno 3 livelli convoluzionali, dropout per la regolarizzazione e ottimizzatore Adam. Fornisci anche i parametri di addestramento e una funzione di valutazione del modello."

Perché funziona?

- La funzione chiarisce il ruolo del modello AI.
- L'outcome specifica il risultato richiesto.
- Il contesto fornisce dettagli tecnici e di background.
- L'utilizzo indica come il risultato sarà impiegato.
- Le specifiche definiscono vincoli chiari e requisiti precisi.

Quando usare FOCUS?

- Sviluppo di codice e automazione software.

- Compiti ingegneristici e scientifici avanzati.
- Generazione di contenuti con parametri tecnici dettagliati.

e. Expert Frameworks (Framework per Compiti Complessi e Specializzati)

- i. **CLEAR** – Acronimo di cinque principi guida: *Concise, Logical, Explicit, Adaptive, Reflective* (in italiano: **Conciso, Logico, Esplicito, Adattivo, Riflessivo**). Più che uno schema di prompt da compilare, è un framework per valutare e migliorare un prompt ([How to Use the CLEAR Framework to Get Better Results from ChatGPT](#)). **Conciso**: il prompt dev'essere chiaro e diretto, evitando parole superflue ([How to Use the CLEAR Framework to Get Better Results from ChatGPT](#)). **Logico**: organizzare le istruzioni in modo strutturato e sequenziale, così che l'AI segua facilmente il filo ([How to Use the CLEAR Framework to Get Better Results from ChatGPT](#)). **Esplicito**: non lasciare sottintesi, specificare esattamente cosa si vuole (anche con esempi di output voluto) ([How to Use the CLEAR Framework to Get Better Results from ChatGPT](#)). **Adattivo**: essere pronti a iterare e modificare il prompt in base alla risposta, non fermarsi alla prima stesura ([How to Use the CLEAR Framework to Get Better Results from ChatGPT](#)). **Riflessivo**: rivedere criticamente il prompt e il risultato, per capire se rispecchia davvero l'intenzione e migliorarlo ulteriormente. Il framework CLEAR aiuta a mantenere standard elevati: ad esempio, se un prompt iniziale non produce output soddisfacenti, applicando CLEAR possiamo scoprire che era troppo vago (mancava la parte "Explicit") o troppo prolisso (violando "Concise"), e correggerlo di conseguenza. In biblioteconomia e ambito accademico, questo modello è stato adottato per educare gli utenti a interagire in modo più efficace con i sistemi AI generativi, puntando sulla **chiarezza** e la capacità di adattamento.
- ii. **SCOPE** – *Scenario, Complications, Objective, Plan, Evaluation* (**Scenario, Complicazioni, Obiettivo, Piano, Valutazione**): Un framework orientato alla pianificazione strategica ([2025 Complete Guide to Prompt Frameworks: 18 Practical Frameworks to Boost Your AI Conversation Efficiency by 10x - AI WorkFlow Studio](#)). Aiuta a delineare prompt complessi per progetti articolati o analisi strutturate. Vediamone l'uso con un esempio: "*Scenario: Una piccola impresa tech intende lanciare un nuovo prodotto sul mercato internazionale nei prossimi 6 mesi. Complicazioni: budget di marketing limitato, forte concorrenza locale in alcuni paesi target, necessità di adattare il prodotto a normative diverse. Obiettivo: Definire una strategia di lancio che entro un anno porti a una quota di mercato del 5% in due paesi. Piano: Elenca le azioni chiave (marketing digitale, partnership, localizzazione) con relativa tempistica e risorse. Valutazione: Indica come misurare il successo di ogni azione (es. metriche di vendita trimestrali, tasso di adozione) e come effettuare eventuali correzioni di rotta.*" Un prompt così strutturato copre tutti gli aspetti: contesto iniziale, possibili criticità, ciò che si vuole ottenere, come si pensa di agire e come verificare i risultati. SCOPE assicura che l'AI risponda in modo **esaustivo**, toccando ogni punto nevralgico dalla definizione del problema alla misura delle soluzioni ([2025 Complete Guide to Prompt Frameworks: 18 Practical Frameworks to Boost Your AI Conversation Efficiency by 10x - AI WorkFlow Studio](#)) ([2025 Complete Guide to Prompt Frameworks: 18 Practical Frameworks to Boost Your AI Conversation Efficiency by 10x - AI WorkFlow Studio](#)). Questo è prezioso

quando si utilizzano modelli AI come consulenti strategici o planner, poiché li costringe a considerare sia gli ostacoli sia i criteri di successo.

- iii. **GUIDE – Goal, Use-case, Instructions, Details, Examples** : Il framework **GUIDE** è progettato per la scrittura di **prompt strutturati e contestualizzati**, particolarmente utili per compiti complessi che richiedono **un output chiaro e specifico**. Questo framework è ideale per **prompt di pianificazione strategica, creazione di contenuti dettagliati e generazione di output articolati**.

Struttura del Framework GUIDE

- 1) **Goal (Obiettivo)** → Definisce con precisione lo scopo della richiesta e il risultato atteso.
- 2) **Use-case (Caso d'uso)** → Specifica il contesto in cui l'output sarà utilizzato.
- 3) **Instructions (Istruzioni)** → Fornisce linee guida operative dettagliate per il modello AI.
- 4) **Details (Dettagli)** → Include informazioni specifiche, parametri o vincoli rilevanti per il compito.
- 5) **Examples (Esempi)** → Se necessario, offre esempi concreti di output desiderato.

Esempio di utilizzo del framework GUIDE

Prompt generico:

"Scrivi un articolo sulle nuove tendenze tecnologiche."

Prompt ottimizzato con GUIDE:

*"Goal: Creare un articolo di 1500 parole che descriva le principali tendenze tecnologiche emergenti nel 2025.

Use-case: Il contenuto sarà pubblicato su un blog di innovazione tecnologica e rivolto a un pubblico di professionisti IT e imprenditori digitali.

Instructions: Analizza almeno 5 trend chiave (es. AI generativa, quantum computing, 6G, blockchain per il settore sanitario, sostenibilità digitale). Struttura l'articolo con introduzione, sezioni per ogni trend e conclusione con prospettive future.

Details: Ogni sezione deve includere una spiegazione del trend, dati recenti e almeno un caso d'uso pratico. Usa un linguaggio tecnico ma accessibile.

Examples: Il tono deve essere simile agli articoli di Wired o MIT Technology Review. Un esempio di struttura potrebbe essere:

Introduzione: Contesto sull'accelerazione tecnologica

- **Tendenza 1:** AI Generativa – Evoluzione e impatti
- **Tendenza 2:** Quantum Computing – Progressi e limiti attuali
- **Tendenza 3:** 6G – Il futuro delle telecomunicazioni
- **Tendenza 4:** Blockchain nel settore sanitario – Sicurezza e trasparenza
- **Tendenza 5:** Sostenibilità digitale – Green computing e AI ecologica
- **Conclusione:** Previsioni per i prossimi 5 anni."

Perché funziona?

- L'**obiettivo** fornisce una direzione chiara.
- Il **caso d'uso** contestualizza l'output per il pubblico target.
- Le **istruzioni** delineano la struttura e il metodo di elaborazione.
- I **dettagli** specificano parametri e vincoli chiave.
- Gli **esempi** aiutano a standardizzare il formato e il tono desiderato.

Quando usare GUIDE?

- **Generazione di contenuti strutturati e articolati.**
- **Scrittura di documentazione tecnica o accademica.**
- **Creazione di report strategici o analisi di mercato.**

Questi framework non vanno seguiti ciecamente in ogni prompt, ma rappresentano ottime **guide pratiche**. Un prompt efficace spesso combina elementi di più modelli: ad esempio, potremmo usare RACE per strutturare un prompt e poi applicare CLEAR per rivederne la qualità linguistica e logica. L'obiettivo finale è sempre lo stesso: comunicare all'AI cosa vogliamo nel modo più chiaro e completo possibile, minimizzando fraintendimenti e massimizzando la pertinenza della risposta.

7. Il Framework di Greg Brockman

Greg Brockman, presidente di OpenAI, ha introdotto un framework strutturato per scrivere prompt efficaci, ottimizzando l'interazione con i modelli AI basati sul ragionamento (reasoning-based models). Questo framework si fonda su quattro elementi chiave che permettono di strutturare prompt chiari, mirati e ottimizzati per ottenere risposte precise e pertinenti.

- **Goal (Obiettivo)**

Definire con precisione l'obiettivo del prompt è il primo passo per ottenere una risposta mirata dal modello AI. Un obiettivo vago o generico può portare a risposte imprecise o poco utili, mentre un obiettivo ben formulato riduce il rischio di ambiguità.

Esempio generico (meno efficace):

"Dimmi qualcosa sulla dieta vegana."

Esempio avanzato (più efficace):

"Forniscimi un piano alimentare vegano settimanale per uno sportivo che necessita di 3000 kcal al giorno."

Perché è importante?

I modelli di ragionamento non necessitano di istruzioni dettagliate per elaborare il ragionamento, ma beneficiano di un obiettivo chiaro e ben definito.

- **Return Format (Formato della Risposta)**

Specificare il formato desiderato dell'output migliora la leggibilità e l'utilità della risposta. I modelli AI possono restituire risposte sotto forma di testo continuo, elenchi puntati, tabelle, codice strutturato o JSON.

Esempio senza formato specificato:

"Quali sono i migliori film di fantascienza?"

Esempio con formato specificato:

"Genera una tabella con tre colonne: Nome del film, Regista, Anno di uscita."

Perché è importante?

Specificare **struttura e formato** (lista, tabella, JSON) aiuta il modello a restituire risposte più precise e organizzate, senza necessità di ulteriori riformulazioni.

- **Warnings (Avvertenze e Limiti)**

Impostare vincoli chiari aiuta a prevenire la generazione di informazioni inesatte, obsolete o eccessivamente generiche. Questo è particolarmente utile in ambiti in cui l'accuratezza dei dati è fondamentale, come il settore medico, finanziario o storico.

Esempio troppo vago:

"Dimmi chi è il miglior calciatore della storia."

Esempio con avvertenza:

"Fornisci un elenco dei calciatori con più Palloni d'Oro, basandoti su dati FIFA ufficiali aggiornati al 2025."

Perché è importante?

A differenza degli LLM tradizionali, questi modelli **non necessitano di richieste esplicite per strutturare il loro ragionamento**. Troppi dettagli inutili possono addirittura compromettere l'efficacia del prompt.

- **Context Dump (Contesto Dettagliato, Organizzazione del Testo e delle Informazioni)**

Più dettagli si forniscono nel prompt, più accurata e personalizzata sarà la risposta. Aggiungere informazioni contestuali aiuta il modello a comprendere meglio l'intento dell'utente e a generare output più mirati. Strutturare il prompt in **sezioni ordinate** aiuta il modello a comprendere meglio le istruzioni, senza generare ambiguità.

Esempio senza contesto:

"Scrivimi una lettera di presentazione."

Esempio con contesto dettagliato:

"Scrivi una lettera di presentazione per un ruolo di Data Analyst, con un tono formale e mettendo in evidenza le mie competenze in Python e SQL."

Esempio senza struttura:

"Scrivi un riepilogo delle normative fiscali per PMI."

Esempio con struttura chiara:

Richiesta: Riassumi le normative fiscali italiane per le PMI aggiornate al 2024.

Formato: Testo continuo di massimo 300 parole.

Aspetti da includere:

- 1) Tassazione per PMI con fatturato inferiore a 1 milione di euro.
- 2) Regime fiscale agevolato e incentivi disponibili.
- 3) Impatti delle nuove leggi approvate nel 2024."

Perché è importante?

L'organizzazione del prompt con **tag strutturati, sezioni e punti elenco** aiuta il modello a interpretare meglio le informazioni e a generare risposte più pertinenti.

L'adozione di questo framework riduce il numero di iterazioni necessarie per ottenere risposte soddisfacenti, rendendo l'interazione con il modello più efficiente e migliorando la qualità dell'output.

Quando usare esempi?

A differenza degli LLM tradizionali, i modelli di ragionamento **funzionano già bene in zero-shot** (senza esempi). Tuttavia, se la risposta non è ottimale, si può ricorrere a **few-shot prompting**, fornendo pochi esempi ben selezionati.

Modularità e flessibilità

Non esiste una regola rigida per scrivere prompt efficaci. Tuttavia, seguire il **Framework di Greg Brockman** garantisce una formulazione più chiara e strutturata, riducendo il numero di iterazioni necessarie per ottenere risposte soddisfacenti.

Integrazione con altre modalità di AI

Quando necessario, il Prompt Engineering può essere combinato con tecniche come **analisi di**

documenti, ricerca web (RAG) o analisi di immagini, per fornire ai modelli reasoning un contesto più ampio e accurato.

L'adozione del **Framework di Greg Brockman** è un metodo efficace per ottimizzare le richieste ai modelli di ragionamento, migliorando la qualità e la precisione delle risposte. Seguendo questi principi:

Obiettivo chiaro → Evita ambiguità.

Formato definito → Organizza l'output in modo utile.

Evita istruzioni ridondanti → Lascia che il modello ragioni autonomamente.

Struttura organizzata → Migliora la comprensione e la coerenza della risposta.

Queste linee guida sono particolarmente utili per richieste **complesse** o per necessità di **output dettagliati e strutturati**. Integrando questi principi nel Prompt Engineering, è possibile sfruttare al meglio il potenziale dei modelli reasoning di nuova generazione.

Il **Framework di Greg Brockman** è particolarmente adatto per i **modelli di ragionamento** (reasoning models), come quelli della **serie-o di OpenAI**. Questi modelli si comportano in modo diverso rispetto agli LLM tradizionali, rendendo necessario un approccio strutturato per ottenere risposte più precise e pertinenti.

8. Strumenti e Risorse per il Prompt Engineering

Il crescente interesse per il Prompt Engineering ha portato allo sviluppo di molti **strumenti e risorse** utili sia per sperimentare che per automatizzare la creazione di prompt efficaci. Ecco alcuni dei principali a disposizione di sviluppatori e professionisti:

- **OpenAI Playground:** È un'interfaccia web fornita da OpenAI che permette di provare interattivamente i modelli di linguaggio (come GPT-3.5 o GPT-4) con vari prompt, regolando parametri come temperatura, lunghezza massima della risposta e altro. Il Playground consente di *costruire e modificare i prompt con risposte in tempo reale* da parte del modello scelto ([Top 5 Prompt Engineering Tools for Evaluating Prompts](#)). Questo strumento è prezioso per fare esperimenti rapidi: si può vedere come cambia l'output al variare del prompt, confrontare diversi approcci fianco a fianco e iterare finché non si ottiene il risultato desiderato. Per i principianti è anche un ottimo modo di **imparare** – provando vari esempi e utilizzando i template predefiniti si acquisisce intuizione su cosa funziona meglio.
- **PromptHero e librerie di prompt ottimizzati:** PromptHero è un motore di ricerca e community online dove gli utenti condividono prompt efficaci per vari modelli (da ChatGPT a sistemi di generazione immagini come Midjourney). Questo tipo di risorsa offre una **libreria di prompt** pronti all'uso o da cui trarre ispirazione. Ad esempio, un marketer può cercare prompt per "slogan pubblicitari" e trovarne di già sperimentati da altri utenti, votati e commentati. Oltre a PromptHero, esistono marketplace e raccolte come *PromptBase*, *FlowGPT* e repository open-source su GitHub (es. *Awesome ChatGPT Prompts*): tutti contribuiscono a diffondere le *best practice* di prompt engineering, permettendo di riutilizzare idee e format collaudati dalla comunità. Per un professionista, consultare queste librerie può far risparmiare tempo e dare spunti creativi, mentre per un neofita rappresentano un apprendimento per esempi.
- **LangChain e LlamaIndex:** Si tratta di framework di sviluppo pensati per costruire applicazioni avanzate che integrano modelli di linguaggio. **LangChain** (per Python/JavaScript) consente di concatenare chiamate ai LLM e gestire *prompt template*, memoria conversazionale, integrazione con strumenti esterni e molto altro. **LlamaIndex** (noto in precedenza come GPT Index) facilita l'indicizzazione di documenti e il recupero di informazioni pertinenti da fornire ai prompt (implementando di fatto tecniche di RAG). In

sostanza, questi strumenti aiutano a **automatizzare e orchestrare i prompt**: invece di scrivere manualmente ogni prompt, si possono creare catene (pipeline) dove l'output di un modello diventa input per il successivo, o dove il prompt viene arricchito dinamicamente con dati. Ad esempio, con LangChain si può realizzare un agente che: prende una domanda dell'utente, formula un prompt per cercare su Google (usando un modello o API), poi inserisce i risultati in un altro prompt per il modello finale che produce la risposta. L'uso di framework come questi è ormai comune nello sviluppo di *chatbot* personalizzati, assistenti virtuali e sistemi di domanda-risposta su base di conoscenza, perché forniscono le **infrastrutture modulari** per gestire prompt complessi e multi-turno in modo sistematico.

- **AutoGPT e agenti AI autonomi**: Auto-GPT, BabyAGI e progetti simili hanno guadagnato popolarità come esempi di agenti AI in grado di iterare su un compito in modo relativamente autonomo. In pratica, strumenti come AutoGPT sfruttano un modello di linguaggio per generare un **ciclo di prompt e azioni** senza intervento umano ad ogni passo. L'utente dà un obiettivo generale (es. *"Ricerca le tendenze di mercato e scrivi un rapporto"*) e l'agente si suddivide il compito, creando sotto-obiettivi, formulando prompt per interrogare il modello su ciascuno di essi, valutando i risultati e proseguendo fino al completamento del goal. Tecnicamente, AutoGPT utilizza un prompt iniziale "maestro" molto lungo che include istruzioni sul comportamento dell'agente, la capacità di auto-generare nuovi prompt (le *thoughts* e *actions* come nel modello ReAct) e di memorizzare risultati parziali in una memoria esterna ([AutoGPT - LangChain.js](#)). Questi agenti autonomi rappresentano un'evoluzione del prompt engineering: il prompt non è più solo la richiesta singola, ma l'intero **processo iterativo** è guidato da prompt generati dall'AI stessa. Pur essendo ancora sperimentali, dimostrano come un'AI ben *promptata* possa avviare catene di operazioni complesse – ad esempio, AutoGPT potrebbe decidere di fare ricerche web, poi chiedere all'LLM di analizzare ciascuna ricerca, poi aggregare i risultati in un report finale. Per i professionisti, giocare con questi strumenti offre uno sguardo su un futuro dove l'AI potrebbe fungere da "collega autonomo" su certi task, e sottolinea l'importanza di **prompt robusti** che garantiscano che l'agente segua gli obiettivi in modo affidabile e sicuro.
- **Strumenti di valutazione dei prompt**: Oltre a creare prompt, è importante valutarne le performance. In contesti aziendali o di ricerca, si utilizzano tool per effettuare **test sistematici** sui prompt. Ad esempio, OpenAI ha rilasciato un framework chiamato *OpenAI Evals* per valutare in maniera automatizzata le risposte dei modelli su una serie di query e confrontare prompt diversi o modelli diversi. Altri strumenti, come *PromptFlow* di Azure, offrono metriche e tracciamento delle versioni dei prompt ([Top 5 Prompt Engineering Tools for Evaluating Prompts](#)), permettendo di misurare quale formulazione fornisce risultati migliori su certi benchmark (ad esempio precisione delle informazioni, preferenza degli utenti, tasso di successo in un compito). Per chi sviluppa applicazioni con LLM, esistono anche librerie come **PromptLayer**, **LangSmith** o **Gradio** integrate con pipeline di testing, che aiutano a monitorare l'efficacia dei prompt nel tempo e sotto diverse condizioni. In mancanza di strumenti dedicati, una *best practice* è costruire internamente dataset di esempio con input e output attesi, e provare periodicamente i prompt su questi dataset per assicurarsi che l'AI continui a rispondere come voluto. Questo processo di **validazione** è cruciale soprattutto quando si aggiornano i modelli o si modificano i prompt: consente di individuare regressioni o bias introdotti e di intervenire con aggiustamenti (ad es. aggiungendo specifiche al prompt o cambiando approccio) per mantenere alta la qualità delle risposte.

9. Applicazioni Pratiche nei Vari Settori

Il Prompt Engineering, pur essendo trasversale, si manifesta in modo diverso a seconda del campo di applicazione. Vediamo alcune aree in cui i prompt ben progettati stanno rivoluzionando il modo di lavorare:

Sviluppo software e automazione – I programmatori sono stati tra i primi ad adottare l'AI generativa come “co-pilota” di coding. Strumenti come GitHub Copilot o Amazon CodeWhisperer utilizzano modelli di linguaggio addestrati sul codice per suggerire automaticamente parti di programma o aiutare nel debugging. In questo contesto, scrivere prompt efficaci significa, ad esempio, commentare il codice in modo chiaro per far sì che l'AI proponga la funzione mancante correttamente, oppure porre domande specifiche sull'errore ricevuto. Un developer può chiedere a ChatGPT: *“Ecco il mio codice e l'errore che ottengo: [codice] [errore]. Mi aiuti a trovare la causa e a risolverla?”*. Un prompt ben formulato includerà i pezzi di codice rilevanti e il messaggio di errore completo, ottenendo dall'AI una spiegazione dettagliata e magari snippet corretti. Nei flussi di automazione (DevOps, test), si usano prompt per generare script o configurazioni: ad esempio, descrivendo a parole cosa deve fare uno script, l'AI può produrre il codice Bash o Python corrispondente. Questo riduce il lavoro manuale e velocizza i tempi, ma richiede comunque verifica umana. In Cybersecurity, lato sviluppo sicuro, è possibile sfruttare prompt per far analizzare all'AI possibili vulnerabilità in un blocco di codice (es. *“Analizza il seguente codice alla ricerca di vulnerabilità XSS o SQL injection”*). In sintesi, il Prompt Engineering in ambito software consiste nel sapere **come chiedere all'AI di generare o analizzare codice**, includendo il giusto contesto (linguaggio, librerie, requisiti) per ottenere risultati utilizzabili e corretti.

Marketing e contenuti digitali – Questo è un settore dove creatività e ripetitività vanno a braccetto, e l'AI sta dando un contributo notevole. I marketer utilizzano prompt per generare testi pubblicitari, post per social media, descrizioni di prodotto, articoli di blog ottimizzati SEO e persino script per video. Ad esempio, un copywriter può istruire il modello: *“Sei un copywriter pubblicitario. Scrivi un testo di 50 parole, tono entusiasta, per promuovere un nuovo smartphone con fotocamera avanzata e batteria a lunga durata.”* In base a come si formula il prompt (tono, lunghezza, punti da enfatizzare), l'output varierà. La possibilità di iterare velocemente consente di provare diverse angolazioni creative modificando leggermente il prompt (ad es. *“tono ironico”* vs *“tono emotivo”*). Oltre al testo, l'AI generativa spinta dai prompt viene sfruttata per **creare immagini e video**: qui il prompt engineering visivo è altrettanto cruciale. Strumenti come DALL-E, Midjourney o Stable Diffusion traducono una descrizione testuale in un'immagine, e la qualità/aderenza dell'immagine dipende da quanto dettagliato e preciso è il prompt (esempio: *“Illustrazione in stile fumetto vintage di una famiglia che sorride attorno a un nuovo frigorifero ecologico, colori pastello”*). I professionisti del marketing stanno imparando a padroneggiare questi **prompt visivi** scegliendo accuratamente aggettivi, riferimenti stilistici, composizione, per ottenere immagini che rispecchino l'identità del brand. In ambito pubblicitario, si usano prompt anche per **brainstorming**: l'AI può proporre elenchi di idee per campagne o slogan dati certi vincoli, fungendo da “collega creativo instancabile”. Un grande vantaggio è la **personalizzazione su larga scala**: con opportuni prompt parametrizzati, un'azienda può far generare automaticamente centinaia di variazioni di un messaggio adattate a diversi segmenti di pubblico, cosa impraticabile manualmente. Il Prompt Engineering nel marketing dunque combina creatività e strategia, assicurando che dall'AI escano contenuti coinvolgenti, mirati e in linea col messaggio da comunicare.

Educazione e ricerca – In ambito educativo, gli assistenti AI stanno diventando tutor virtuali, e i prompt ben congegnati sono essenziali per ottenere spiegazioni chiare e corrette. Un insegnante, ad esempio, può chiedere a ChatGPT: *“Spiega la seconda legge della termodinamica in modo semplice, con un paragone concreto, per una presentazione a studenti di 15 anni.”* Qui specificare il livello di istruzione e il tipo di esempio richiesto è fondamentale per tarare la risposta sul pubblico

giusto. Gli studenti stessi possono usare l'AI come supporto nello studio: formulando buoni prompt possono farsi riepilogare un capitolo difficile, ottenere la soluzione spiegata di un problema matematico (magari chiedendo di non fornire direttamente la risposta ma di dare un suggerimento), o ancora generare quiz di autovalutazione sull'argomento che stanno studiando. Nella **ricerca accademica**, i modelli di linguaggio possono aiutare a sintetizzare letteratura o a esplorare connessioni tra concetti. Ad esempio, un ricercatore può istruire l'AI: *"Riassumi le differenze chiave tra le teorie X e Y in sociologia, citando eventualmente autori rilevanti."* Oppure: *"Elenca possibili applicazioni future della tecnologia descritta in questo articolo."* Ottenere queste risposte in bozza aiuta a risparmiare tempo, purché poi l'utente verifichi e ripulisca i contenuti (specialmente le citazioni, che vanno controllate per evitare *hallucinations*). Alcune università stanno integrando sistemi AI per assistere nella scrittura di documenti o nella traduzione automatica di paper scientifici, e insegnano agli studenti di dottorato come scrivere prompt efficaci per queste attività. Un altro uso in ricerca è l'**analisi di dati testuali**: con opportuni prompt, si può chiedere al modello di linguaggio di analizzare qualitativamente risposte aperte di un sondaggio, individuare temi ricorrenti o sentiment, fungendo da strumento preliminare di codifica qualitativa. In tutti questi esempi, la capacità di dire all'AI esattamente *come* svolgere il compito (livello di dettaglio, formato, lunghezza, ecc.) è ciò che distingue un risultato mediocre da uno realmente utile per il lavoro accademico.

Cybersecurity e analisi forense – In un campo dove l'attenzione al dettaglio è vitale, l'AI può essere impiegata come assistente per vagliare grandi moli di dati o per simulare scenari di attacco. Ad esempio, nel *penetration testing*, un esperto di sicurezza potrebbe usare un modello generativo per **simulare tentativi di phishing**: fornendo il prompt *"Sei un hacker black-hat esperto di social engineering. Componi un'email di phishing convincente rivolta a dipendenti di un'azienda finanziaria, fingendo di essere l'ufficio IT che chiede di reimpostare la password."* Questo ovviamente va fatto in ambiente controllato, ma aiuta i team di sicurezza a generare casistiche realistiche per addestrare i dipendenti al riconoscimento degli attacchi. Allo stesso modo, si possono chiedere suggerimenti su come sfruttare una vulnerabilità nota in un software (per poi prevenire tale exploit). Nell'analisi forense digitale, dove magari ci sono log di sistema enormi o conversazioni da analizzare, un prompt engineering efficace consente di interrogare l'AI per trovare *pattern* sospetti. Ad esempio: *"Analizza questi log di accesso e segnala eventuali anomalie o tentativi di intrusione, spiegando perché sono anomali."* L'AI può velocemente evidenziare orari insoliti, IP non riconosciuti o sequenze di azioni tipiche di un attacco, che poi l'analista umano approfondirà. Ancora, dopo un incidente informatico, si potrebbe utilizzare l'AI per sintetizzare report tecnici: *"Ecco i dati grezzi di un traffico di rete anomalo (...); descrivi in linguaggio comprensibile cosa potrebbe essere successo e quali passi di mitigazione intraprendere."* In questo caso il prompt deve istruire chiaramente l'AI a non inventare informazioni, ma a basarsi solo sui dati forniti e magari su conoscenze documentate di attacchi (eventualmente integrando un RAG con database di *malware* noti). Nel complesso, nel settore cybersecurity il Prompt Engineering sta diventando uno strumento in più: richiede rigore (i risultati vanno sempre verificati), ma può fungere da **moltiplicatore di forze** – generando ipotesi di attacco da testare, setacciando informazioni alla ricerca di *needle in the haystack*, e aiutando a stilare procedure di risposta agli incidenti in modo rapido e consistente.

Business e produttività – In uffici e aziende di ogni settore, l'AI viene adottata per automatizzare compiti quotidiani e supportare le decisioni. Un caso comune è l'**automazione di email e documenti**: molti professionisti ora usano strumenti basati su prompt per redigere bozze di email (specialmente in risposta a schemi ricorrenti), per riassumere lunghe conversazioni o verbali di riunioni, e per tradurre testi in altre lingue. Ad esempio, un manager dopo una lunga riunione può copiare nel prompt gli appunti grezzi e chiedere: *"Riassumi i punti chiave della riunione di progetto in un elenco puntato, evidenziando decisioni prese e azioni assegnate (chi fa cosa e scadenze)."* In pochi secondi ottiene una minuta ordinata, che poi può velocemente rivedere e inviare al team. I prompt ben fatti qui includono sempre dettagli su **forma e destinatari**: specificare "elenco puntato"

o “tono formale/informale” o “lunghezza massima” aiuta a ottenere un output immediatamente utilizzabile senza ulteriori editing. Un'altra applicazione è negli *assistant AI* integrati nei sistemi di produttività (come schedulatori o CRM): con prompt dietro le quinte, l'AI può prendere un comando generico dell'utente (*“Programma una riunione con il team vendite la prossima settimana per discutere il lancio X”*) e tradurlo in azioni concrete (trovare uno slot libero in calendario, invitare i partecipanti, predisporre un ordine del giorno). Per quanto riguarda il **decision making**, i dirigenti iniziano a sfruttare l'AI per avere rapidamente analisi preliminari su cui ragionare. Ad esempio: *“Confronta questi due progetti di investimento (A e B) elencandone vantaggi, rischi e ROI atteso, in forma tabellare.”* fornendo magari nel prompt i dati salienti di A e B. L'AI genererà una tabella comparativa che il dirigente userà come base per la discussione, ovviamente integrandola con la propria esperienza. Oppure, si possono chiedere *scenari what-if*: *“Se le vendite calano del 10% il prossimo trimestre, quali misure correttive potremmo adottare? Fornisci 3 possibili azioni con pro e contro ciascuna.”* Il modello restituisce idee (es. riduzione costi, promozioni speciali, diversificazione mercato) che stimolano la riflessione del management. In tutti questi casi, l'AI non prende la decisione ma **assiste** chi deve decidere, facendogli risparmiare tempo nelle analisi preliminari e offrendogli prospettive aggiuntive. Il Prompt Engineering entra in gioco perché l'utilità di queste analisi dipende da *come* viene posta la domanda: più il prompt circoscrive il problema e definisce il tipo di risposta (formato, criteri), più l'output sarà centrato e immediatamente fruibile nel contesto lavorativo.

10. Futuro del Prompt Engineering e Conclusioni

L'evoluzione rapida dell'AI pone inevitabilmente la domanda: **che ruolo avrà il Prompt Engineering in futuro?** Già oggi vediamo modelli sempre più capaci di seguire istruzioni complesse e contestuali, ma lungi dal rendere obsoleta questa disciplina, tali avanzamenti ne stanno cambiando forma e portata. Siamo all'alba di un'era in cui l'interazione uomo-AI sarà sempre più naturale e pervasiva, e il prompt engineering è destinato a rimanere un pilastro nel plasmare queste interazioni ([What is Prompt Engineering? A Detailed Guide For 2025 | DataCamp](#)) ([What is Prompt Engineering? A Detailed Guide For 2025 | DataCamp](#)).

Una direzione di sviluppo è quella dei **prompt adattativi e auto-generativi**. I ricercatori stanno esplorando metodi per cui sia lo stesso modello a poter riformulare o generare sotto-prompts in base al contesto, riducendo il bisogno di input manuali estremamente dettagliati ([What is Prompt Engineering? A Detailed Guide For 2025 | DataCamp](#)). Ad esempio, un futuro assistente potrebbe capire dal tono o da meta-istruzioni come approfondire una domanda, ponendo chiarimenti all'utente in autonomia (un po' come farebbe un essere umano chiedendo “mi confermi che intendi X?”) oppure attingendo a esperienze pregresse per auto-configurarsi. Già con tecniche come il *self-consistency* o agenti tipo AutoGPT intravediamo modelli che *si parlano da soli* per migliorare l'output. Questo però non elimina la figura del prompt engineer, anzi la sposta più *dietro le quinte*: bisognerà creare **meta-prompts** e istruzioni di sistema ancora più sofisticati, che definiscano i confini e gli obiettivi entro cui l'AI deve poi autogestirsi. In pratica, l'umano continuerà a dire “cosa fare” e in parte “come farlo”, ma l'AI potrebbe determinare autonomamente alcuni dettagli esecutivi.

Un'altra tendenza chiave è l'espansione verso **prompt multimodali** ([What is Prompt Engineering? A Detailed Guide For 2025 | DataCamp](#)). Con modelli capaci di gestire immagini, audio e video oltre al testo (come i recenti GPT-4 Vision o Google Gemini), il prompt engineering diventa più complesso e interessante. Già oggi possiamo chiedere a un'AI di descrivere un'immagine o di rispondere a domande combinando testo e figure. Nel prossimo decennio ci aspettiamo prompt che integrino istruzioni tipo *“Ecco una foto della scena di un incidente stradale, forniscimi un*

rapporto scritto evidenziando possibili cause” oppure *“Analizza questi dati di vendita (grafico allegato) e fammi un riassunto in 5 punti”*. I principi di chiarezza e contestualizzazione restano, ma bisognerà saper *fornire input eterogenei* e specificare l'output desiderato magari in più formati contemporaneamente (es. “spiegami a voce questo documento” implica generare audio). Il prompt engineer del futuro dovrà quindi capire non solo come parlare all'AI in linguaggio naturale, ma anche come collegare insieme diverse modalità di informazione, orchestrando testi, dati e media per ottenere l'effetto voluto.

Man mano che l'AI diventa più integrata nei processi decisionali e produttivi, emergerà anche il concetto di **interfacce più intuitive** per il prompting. Potrebbe darsi che gli utenti finali nemmeno si accorgano di stare “scrivendo un prompt” – ad esempio potrebbero interagire con un assistente vocale che, dietro le quinte, traduce le loro richieste colloquiali in prompt ben formati. Oppure interfacce grafiche dove si spuntano opzioni e il sistema compone il prompt per l'LLM. Questo significa che molte persone beneficeranno del prompt engineering senza conoscerlo direttamente, un po' come oggi usiamo motori di ricerca complessi con query semplici grazie all'ottimizzazione lato motore. Tuttavia, serviranno esperti che **progettino questi livelli di interazione**: definire prompt di sistema, personalità predefinite dell'AI, e linee guida su come tradurre input grezzi degli utenti in qualcosa che il modello possa capire. In sostanza, il ruolo del prompt engineer potrebbe evolvere verso quello di *conversation designer* o *AI behavior designer*, focalizzato sull'esperienza utente e sull'allineamento dell'AI agli obiettivi umani.

Un aspetto non trascurabile del futuro è l'**etica del prompt engineering**. Già oggi si discute di come i prompt possano essere usati per ottenere output scorretti o per aggirare filtri (il cosiddetto *prompt hacking*). Nei prossimi anni, i prompt engineer avranno anche la responsabilità di incorporare nei prompt principi di equità, inclusività e sicurezza ([What is Prompt Engineering? A Detailed Guide For 2025 | DataCamp](#)) ([What is Prompt Engineering? A Detailed Guide For 2025 | DataCamp](#)). Ad esempio, dovranno assicurarsi che un prompt non porti il modello a generare risposte distorte o offensive su certi gruppi, e che vengano previste istruzioni per *mitigare bias* (es. “dà prospettive diverse”, “usa un linguaggio rispettoso per tutti”). Inoltre, con modelli potentissimi, i prompt saranno strumenti altrettanto potenti – saper porre le domande giuste significherà anche poter influenzare decisioni importanti. Ci vorrà quindi un codice di condotta e linee guida, un po' come per qualsiasi disciplina ingegneristica, per usare questa capacità in modo responsabile. Vedremo probabilmente certificazioni o standard di settore sul prompt engineering etico e efficace.

In conclusione, il Prompt Engineering oggi è sia una competenza pratica immediata (che chiunque può iniziare a migliorare seguendo le linee guida come quelle illustrate in questo articolo), sia una nuova frontiera professionale in rapida evoluzione. Le **best practice** che conosciamo – chiarezza, contesto, esempi, iterazione – rimarranno fondamentali, ma dovremo applicarle in contesti sempre più complessi e dinamici. Il connubio uomo-AI diventerà più stretto: non parleremo più solo di “l'utente scrive un prompt e l'AI risponde”, ma di un **dialogo continuo** in cui entrambi si adattano. Saper orchestrare questo dialogo sarà il compito del prompt engineer del futuro, un ruolo a cavallo tra tecnologia e comunicazione, tra creatività e metodo scientifico. Chi oggi inizia a padroneggiare l'arte dei prompt si sta preparando a un domani in cui collaborare con l'intelligenza artificiale sarà all'ordine del giorno in ogni professione. E quando l'AI sarà ovunque, il valore di saperla dirigere con le parole giuste – come un direttore d'orchestra con i suoi musicisti – non potrà che crescere. Il consiglio è quindi di **continuare a sperimentare, imparare e affinare i prompt**, restando aggiornati sui nuovi strumenti e modelli, perché l'AI migliora di giorno in giorno e con essa anche la nostra capacità di interagire in modo produttivo e creativo. In definitiva, il Prompt Engineering non è che la moderna evoluzione dell'antica abilità di porre le domande giuste: un'abilità che, con gli opportuni adattamenti, manterrà la sua importanza indipendentemente da quanto intelligenti diventeranno le macchine. ([Implementing advanced prompt engineering with Amazon Bedrock | AWS Machine Learning Blog](#)) ([What is Prompt Engineering? A Detailed Guide For 2025 | DataCamp](#))

