



ICT Training Center

Il tuo partner per la Formazione e la Trasformazione digitale della tua azienda



SPRING AI

GENERATIVE ARTIFICIAL INTELLIGENCE CON JAVA

Simone Scannapieco

Corso base per Venis S.p.A, Venezia, Italia

Settembre 2025

PROMPT ENGINEERING

10:44

ChatGPT >

MI spieghi cosa è il prompt engineering?

Il **prompt engineering** è l'arte e la tecnica di progettare e formulare input specifici, detti **prompt**, per ottenere risposte desiderate da un modello di intelligenza artificiale (AI) come me. In parole semplici, è il processo di creare domande o richieste ben strutturate per guidare il modello a fornire risultati pertinenti, accurati e utili.

I modelli AI, come quelli basati su **transformer** (ad esempio, GPT), sono progettati per rispondere a input testuali, ma la qualità e la precisione delle loro risposte dipendono molto da come viene formulato il prompt. Ecco alcuni aspetti importanti del prompt engineering:

1. **Chiarezza e specificità:** Un prompt ben formulato deve essere chiaro e specifico, così che l'AI possa comprendere esattamente cosa gli viene chiesto. Ad esempio, anziché chiedere "Dimmi qualcosa sulla storia", si potrebbe specificare "Quali furono le cause principali della Prima Guerra Mondiale?".

+ Messaggio

10:45

ChatGPT >

1. **Chiarezza e specificità:** Un prompt ben formulato deve essere chiaro e specifico, così che l'AI possa comprendere esattamente cosa gli viene chiesto. Ad esempio, anziché chiedere "Dimmi qualcosa sulla storia", si potrebbe specificare "Quali furono le cause principali della Prima Guerra Mondiale?".
2. **Contesto e dettagli:** Fornire contesto o dettagli aggiuntivi all'interno del prompt aiuta il modello a capire meglio ciò di cui hai bisogno. Ad esempio, "Scrivi una breve introduzione sul Rinascimento per un pubblico scolastico".
3. **Struttura e vincoli:** A volte, può essere utile specificare il formato della risposta, come "Dammi una lista di cinque motivi" o "Scrivi una risposta di massimo 100 parole".
4. **Uso di esempi:** Se necessario, includere esempi nel prompt può aiutare l'AI a comprendere meglio il tono, lo stile o il tipo di risposta atteso. Ad esempio, "Rispondi come se fossi un personaggio storico, anziché un personaggio moderno".

+ Messaggio

10:45

ChatGPT >

2. **Contesto e dettagli:** Fornire contesto o dettagli aggiuntivi all'interno del prompt aiuta il modello a capire meglio ciò di cui hai bisogno. Ad esempio, "Scrivi una breve introduzione sul Rinascimento per un pubblico scolastico".
3. **Struttura e vincoli:** A volte, può essere utile specificare il formato della risposta, come "Dammi una lista di cinque motivi" o "Scrivi una risposta di massimo 100 parole".
4. **Uso di esempi:** Se necessario, includere esempi nel prompt può aiutare l'AI a comprendere meglio il tono, lo stile o il tipo di risposta atteso. Ad esempio, "Rispondi come se fossi un personaggio storico, come Leonardo da Vinci".
5. **Iterazioni e perfezionamento:** Il prompt engineering è spesso un processo iterativo. Si potrebbe iniziare con un prompt di base, vedere la risposta del modello, e poi modificarlo o perfezionarlo per ottenere un risultato più accurato.

+ Messaggio

➡ Definiamo meglio il concetto di **prompt**. . .

Istruzione indicazioni/suggerimenti specifici che si presentano al modello per raggiungere l'azione desiderata (designano la *task* da compiere)

Contesto informazioni aggiuntive che possono indirizzare e addestrare il modello in modo che generi risposte migliori, pertinenti e coerenti con l'obiettivo

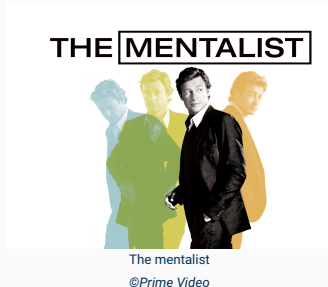
Dati di *input* il testo contenente ciò a cui si è interessati a trovare una risposta

Indicatore di *output* la tipologia o il formato che si vuole ottenere nella risposta

➡ Dati di *input* obbligatori

➡ Altri dati opzionali

- ➔ Sceglie uno **specifico** modello da ottimizzare
- ➔ Capisce come è stato addestrato il modello
- ➔ Capisce come è stato configurato il modello
- ➔ Comprende come riconfigurare il modello
- ➔ Si esprime efficacemente con il modello
 - ➔ Stile
 - ➔ Tono
 - ➔ Struttura
 - ➔ Scelta delle parole
- ➔ Dedica il suo lavoro a tutto questo
- ⚠ Spinge il LLM verso un comportamento **pseudo-deterministico**



- 1 **Lunghezza massima:** numero di *token* massimo generato dal modello
 - ➔ Evitare alti consumi (energetici, economici, ...)
 - ➔ Abbattere i tempi di risposta
 - ➔ Cruciale in alcuni tipi di strategie di *prompting* (ReAct) e per alcune tipologie di *task*
 - ⚠ **Non** impone modifiche stilistiche al modello!

2 Creatività del modello: libertà nella scelta del *token* successivo

➔ Temperatura:

➔ **Bassa** se vogliamo risposte più deterministiche

➔ **Alta** per risposte “creative”

⚠ Intervallo $[0, +\infty]$

➔ Top-K: seleziona i primi K *token* più probabili dalla distribuzione

⚠ Intervallo $[0, +\infty]$

➔ Top-P: seleziona i *token* più probabili e la cui probabilità cumulativa non supera P

⚠ Intervallo $[0, 1]$

⚠ Non sempre tutti disponibili!

➔ Attenzione alle configurazioni estreme

Setup	Effetto
Temp = 0	Top-P, Top-K irrilevanti (<i>greedy decoding</i>)
Temp \gg 0	Temp irrilevante
Top-K = 1	Temp, Top-P irrilevanti
Top-K \gg 1	Temp, Top-P irrilevanti (<i>greedy decoding</i>)
Top-P = 0	Temp, Top-K irrilevanti
Top-P = 1	Top-P irrilevante



Yzma e Kronk

©Disney

Creatività del modello quanta libertà lasciamo al LLM nella scelta del *token* successivo

Temperatura più è bassa, più deterministica la generazione dei *token* successivi (ovvero, preso sempre il *token* più probabile)

Top P definisce una finestra di probabilità entro la quale poter scegliere il *token* successivo (da abbassare quando si vogliono risposte fattuali, da alzare se si vuole un testo più “fuori dagli schemi”)

 Non modificare entrambe!!!

Lunghezza massima numero di *token* massimo generato dal modello

Stop sequences lista di *token* che dice al modello di terminare la generazione

Token penalties contro la ripetizione dei medesimi *token*

Frequency penalty proporzionalmente al n° di volte usato nella sequenza

Presence penalty globale per tutti i *token* ripetuti

 Non modificare entrambe!!!

Direct solo dati di *input*, senza nemmeno definire la *task* obiettivo

- ➔ Utilizzare il modello secondo la sua natura specifica, senza specializzarlo

0-Shot *task* specificata, ma la richiesta dell'utente è priva di informazioni sul contesto o priva di esempi

Few-shots la richiesta dell'utente è seguita da alcuni esempi su cui la LLM tara il *template* delle risposte da fornire

Role-based le informazioni del *prompt* sono suddivise tra una entità *superuser* (*system* o *admin*) e una entità *user*

- ➔ Contesto, definizione *task* e formattazione *output* al *system*
- ➔ *Input* allo *user*

Chain-of-Thought il *prompt* viene costruito affinché il LLM possa “ragionare” per passi sequenziali, sfruttando man mano i dati che ha generato

- ➔ Utilizzato per le *task* di risoluzione problemi di logica, matematica e simili