Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Generative Artificial Intelligence
Prompt Engineering Overview

Salvatore Palange

RT-ICAR-NA-2024-05 **Novembre 2024**

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Generative Artificial Intelligence
Prompt Engineering Overview

Salvatore Palange

Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche
(ICAR-CNR)
Via Pietro Castellino, 111 – 80131 Napoli, Italia

Email: salvatore.palange@icar.cnr.it

Abstract

Questo rapporto tecnico esplora il tema del "Prompt Engineering" nell'ambito dell'intelligenza artificiale generativa (GenAI). Il Prompt Engineering è una tecnica fondamentale che consente di ottimizzare le interazioni con modelli linguistici avanzati, come quelli offerti da ChatGPT[1], Claude[2] di Anthropic[3] o altri.

Il documento copre una panoramica delle diverse tecniche di scrittura del prompt[4], principalmente focalizzata sull'utilizzo di ChatGPT, il ruolo dei token e l'importanza del contesto nel guidare i risultati desiderati.

Viene inoltre descritto come sfruttare queste metodologie per migliorare l'efficienza e l'efficacia di applicazioni specifiche, con particolare attenzione all'ottimizzazione e all'uso di modelli AI per scopi aziendali e tecnici.

Parole chiave: GenAI, Prompt Engineering, Intelligenza Artificiale, Token, Modelli Linguistici, ChatGPT.

1. Introduzione al Prompt e al Prompt Engineering

Dal Novembre 2022, con il lancio del servizio ChatGPT ad opera di OpenAI, i Large Language Models (LLM) hanno catalizzato molta attenzione grazie alla loro capacità di svolgere una quantità notevole di task basati sul linguaggio naturale.

I modelli linguistici, nati negli anni '50, hanno compiuto progressi notevoli, evolvendosi fino agli attuali LLM basati su architettura transformer[5]. Questi modelli, come GPT-4[6], non solo elaborano il linguaggio naturale ma svolgono compiti complessi, ponendosi alla base dello sviluppo di intelligenze artificiali generaliste e dimostrando capacità che vanno oltre quelle per cui sono stati addestrati[7].

I modelli di linguaggio di grandi dimensioni sono chiamati appunto Large Language Models in quanto per ottenere le relative performance hanno un'architettura di rete neurale composta da miliardi di parametri, oltre che per il relativo training necessitano di consistenti capacità computazionali[8]. Esempi di LLM sono GPT-4, LLama[9] o simili. Gli LLM, basati sull'architettura transformer, sono diffusamente impiegati in prodotti per utenti finali, utenti interni o per ricerca[10].

Questi modelli sono strutturati per ricevere in input un "Prompt" per il quale producono una risposta conseguente. Questi prompt possono essere solitamente testuali, o di recente possono essere immagini, suoni, video o una combinazione di questi.

Il Prompt Engineering[11], di conseguenza, è il processo iterativo di progettazione e ottimizzazione dei "prompt" (input testuali) che vengono forniti a modelli linguistici di grandi dimensioni.

È dimostrato empiricamente che strutturare bene un prompt può influenzare positivamente i risultati su differenti task[12]. Ciò rende questo processo cruciale per applicazioni che vanno dalla creazione di contenuti al supporto decisionale.

Un prompt mal formulato può generare risposte imprecise o non rilevanti, mentre un prompt ben congegnato, tenendo conto del contesto e delle aspettative dell'utente, produce risultati ottimali e mirati.

Nella Fig. 1 è illustrato un esempio di incidenza di una strategia avanzata di Prompt Engineering e la relativa influenza sull'accuratezza dei risultati.

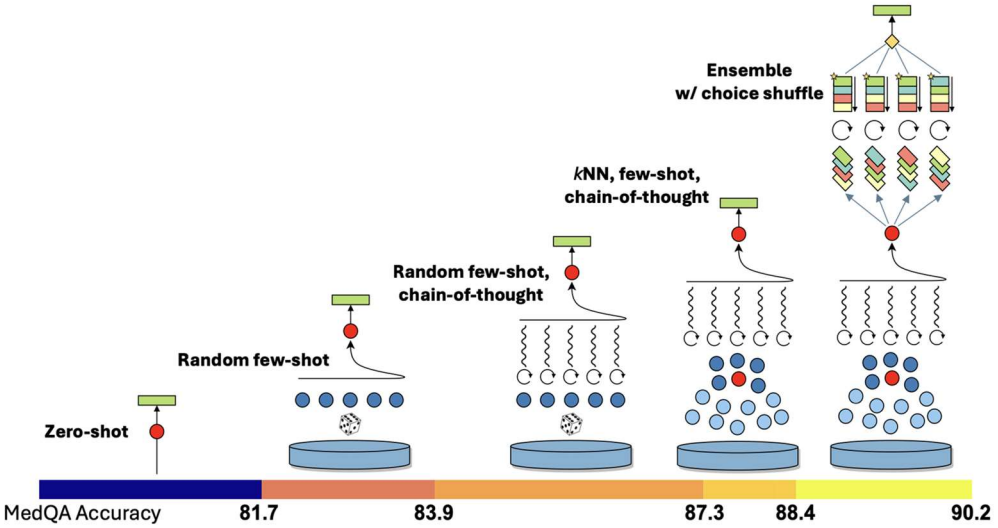

Fig. 1 The prompting strategy on MedQA benchmark for improved Accuracy. "Can Generalist Foundation Models Outcompete Special-Purpose Tuning? Case Study in Medicine" - Harsha Nori* et al. Microsoft - Nov 2023

Nella Fig. 1 sono evidenziati i risultati in termini di accuratezza dei risultati di diverse tecniche di composizione e redazione di prompt in ambito medico[13] testate con il MedQA Benchmark[14]. Si nota dal grafico che all'aumentare della complessità di scrittura del prompt, con tecniche sempre più avanzate, il risultato in termini di accuratezza aumenta, passando da un valore inferiore all'80% fino al 90% ed oltre.

Per comprendere come impostare correttamente un prompt, o quali sono i punti focali su cui porre attenzione, è necessario esaminare preventivamente, il funzionamento di questi modelli di linguaggio naturale per comprenderne le logiche sottostanti.

2. Token e Funzionamento dei Modelli Linguistici

I token sono unità fondamentali per il funzionamento dei modelli linguistici come GPT-4. Un token rappresenta una sequenza di caratteri che può corrispondere a una parola, a una parte di essa o a uno spazio. Quando un modello linguistico elabora un testo, scompone l'input in token, che vengono poi processati per generare una risposta. La tokenizzazione è un'attività comune nell'elaborazione del linguaggio naturale (NLP). È un passaggio fondamentale sia nei metodi tradizionali di NLP sia nelle architetture avanzate basate sul Deep Learning come i Transformers.

Per il seguente Rapporto prendiamo come riferimento ChatGPT di Open AI, consapevoli che ogni modello e ogni piattaforma può differire in architettura, interfaccia di utilizzo, sistemi di pre-processing del testo. La comprensione del concetto di token è importante per ottimizzare il prompt e ottenere risposte precise. Alcuni principi chiave per comprendere la lunghezza e la struttura dei token sono:

- 1 token ≈ 4 caratteri in inglese
- 1 token ≈ ¾ di parola
- 100 token ≈ 75 parole

Un aspetto critico del Prompt Engineering è la capacità di gestire in modo efficiente il numero di token. Un numero eccessivo di token può portare a una riduzione dell'efficacia del modello, mentre un numero insufficiente può limitare la complessità e l'informazione contenuta nella risposta.

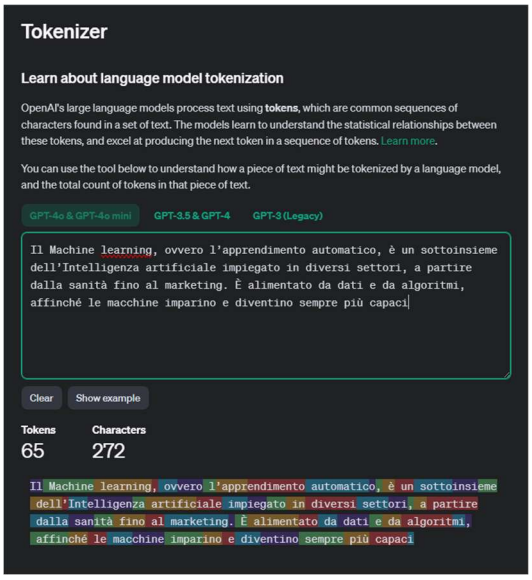

Fig. 2 Esempio di tokenizzazione con il tokenizer di OpenAI

Esistono diversi metodi di tokenizzazione ed ogni Large Language Model utilizza o modelli propri o modelli aperti[15]. Un esempio di tokenizzazione è quello in Fig. 2

Un esempio per comprendere come i token influenzano il risultato è porre la seguente domanda al LLM: "Quante R ha la parola Ramarro?". La risposta corretta è 3 ma l'LLM risponderà:

Fig. 3 Errore dell'LLM nel conteggio delle R nella parola "Ramarro"

La causa dell'errore nella Fig. 3 deriva direttamente da come la parola 'Ramarro' viene gestita nella pipeline di processo. Essa viene suddivisa in due token come illustrato in Fig. 4 e quindi l'LLM non avrà visibilità della parola intera ma dei due token e questo non gli renderà possibile svolgere correttamente questo quesito che per un essere umano è immediato ma ragionerà in funzione del peso che ogni token avrà nel modello.[16]

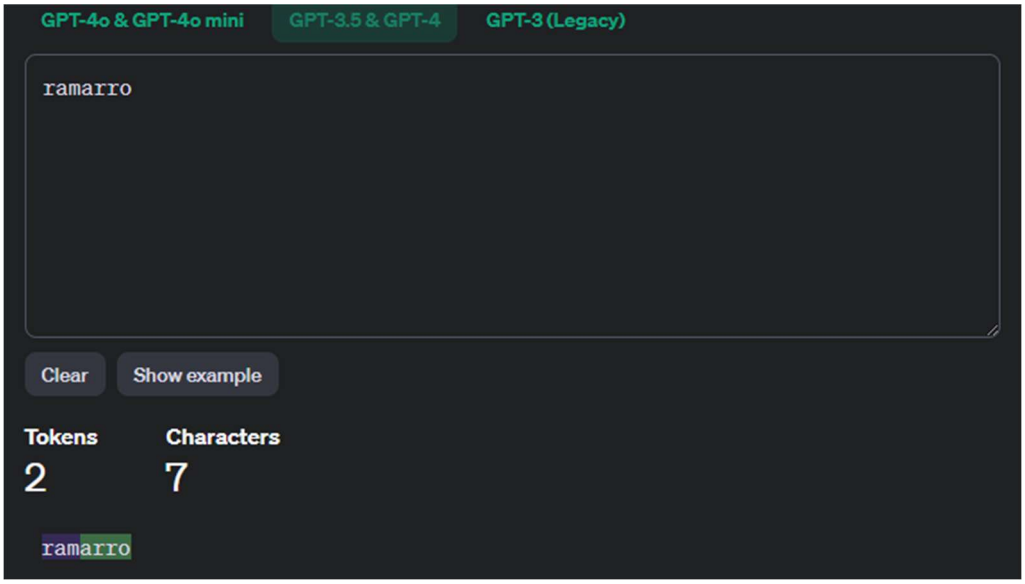

Fig. 4 Suddivisione in token della parola 'Ramarro'

3. Completezza del Prompt

Un elemento cruciale del Prompt Engineering è la formulazione efficace del prompt in termini di contenuto idoneo a circoscrivere correttamente non solo il quesito ma anche il contesto dello stesso per ottenere una risposta appropriata in linea con le aspettative.

Ogni LLM ha un proprio processo di Inferenza peculiare per determinare quelle che possono essere le parole più adatte a completare il Prompt.

Il LLM funzionando con un modello di rete neurale e coi relativi pesi, in breve, calcola la probabilità di quale sia la parola più probabile che segue la frase in input e fa questo lavoro ciclicamente completando man mano il prompt iniziale arricchito delle nuove parole generate.

Diverse ricerche hanno osservato che il design di specifici prompt possono obbligare modelli pre-addestrati ad effettuare particolari task, come question-answering, text generation, e comprensione semantica[17].
Si intuisce che diventa di fondamentale importanza la forma e l'articolazione del prompt proprio per una sua corretta rappresentazione nello spazio vettoriale[18] e una conseguente inferenza corretta.
Qui di seguito vengono forniti esempi di prompt difformi o conformi alle aspettative, come prova dell'importanza della struttura e del contesto.
Nei seguenti esempi descriveremo un utente che desidera produrre un tweet per la piattaforma X relativo ad una sua attività nel campo della Data Science denominata "CookingAI". Mostriamo i differenti risultati.
Esempio di prompt non conforme alle aspettative:

Fig. 5 Prompt non conforme

Risultato: Questo prompt fornisce una risposta errata perché "CookingAI" non è un'applicazione che riguarda la cucina, ma un sito dedicato all'intelligenza artificiale. L' LLM associa automaticamente la parola Cooking alla Cucina per cui tutto il resto della risposta viene composta in funzione di questo input non chiaro e completo.
Esempio di Prompt Conforme alle aspettative:

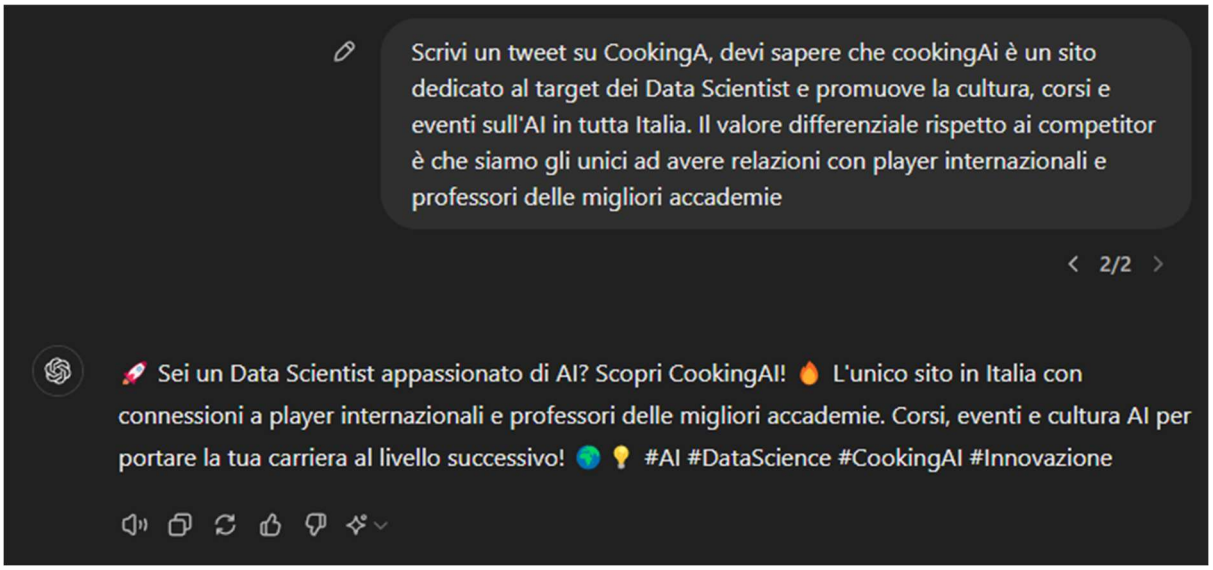

Fig. 6 Esempio di Prompt conforme

Risultato: Questo prompt riflette correttamente il focus tecnico del sito. Infatti, la differenza con il precedente è che abbiamo fornito maggiori informazioni di contesto in modo che il modello possa correttamente posizionare la nostra

richiesta in uno spazio più prossimo a ciò che noi ci aspettiamo come risultato ed in effetti il risultato risulta conforme a ciò che l'utente desidera come risposta.
Nei prossimi paragrafi l'obiettivo sarà quello di scomporre i singoli elementi che possono costituire un corretto prompt per indirizzare (cd. Steering o Sterzare l'LLM) correttamente le risposte nella direzione desiderata dall'utente.

4. Prompt Persona (Role Prompting)

Nel campo del Prompt Engineering, una delle tecniche più potenti e versatili è rappresentata dal "Prompt Persona"[19], noto anche come "Role Prompting". Questa strategia consente di configurare il modello linguistico per rispondere assumendo la prospettiva di una figura specifica, simulandone non solo l'expertise ma anche il linguaggio, il tono e lo stile comunicativo. Il risultato è un'interazione altamente personalizzata, che può adattarsi a una vasta gamma di scenari applicativi. L'idea centrale del Prompt Persona è quella di definire una "persona di riferimento" nel prompt, fornendo al modello un contesto chiaro per la generazione del contenuto. Questo approccio sfrutta la capacità intrinseca dei modelli linguistici di replicare stili e competenze specifiche, garantendo risposte mirate e coerenti.
I principali vantaggi includono:

- Flessibilità contestuale: permette di simulare il comportamento di figure professionali diverse.
- Adattabilità stilistica: calibra tono e registro linguistico in base alle esigenze comunicative.
- Incremento della precisione: migliora l'allineamento delle risposte con il dominio di applicazione richiesto.

Ad esempio:
"Agisci come un esperto SEO": il modello genera strategie dettagliate per migliorare il posizionamento nei motori di ricerca, utilizzando una terminologia tecnica specifica del settore.
"Agisci come un copywriter esperto": il contenuto prodotto è orientato alla persuasione, focalizzandosi su call-to-action e tecniche di storytelling efficaci.
La tecnica del Prompt Persona trova applicazione in molteplici contesti, sia accademici che professionali. Tra i principali possiamo ricordare l'Educazione e la formazione[20], per simulare un professore e spiegare concetti complessi o uno studente per verificare la comprensione. Altro contesto può essere il supporto aziendale: generare report, piani strategici o documentazione tecnica con il tono e il rigore di un consulente esperto. Sicuramente un altro ambito è quello creativo dove poter simulare figure come un autore o un editor per la stesura di testi narrativi o editoriali o creare assistenti virtuali che rispondano in modo naturale, emulando empatia e professionalità.
Come evidenziato nella Fig. 7, il Prompt Persona può influenzare significativamente la qualità e l'orientamento delle risposte del modello. Considerando una stessa richiesta — ad esempio, "scrivi un paragrafo su un argomento" — il risultato varia sensibilmente in base alla persona definita nel prompt iniziale.
Con il ruolo di "Esperto SEO", la risposta sarà focalizzata su aspetti tecnici come l'ottimizzazione delle parole chiave, l'architettura del sito e l'analisi del traffico.
Definendo invece il ruolo di "Copywriter esperto", il modello proporrà un testo persuasivo, strutturato per catturare l'attenzione e motivare un'azione specifica.
Questa versatilità si traduce nella possibilità di simulare approcci multidisciplinari in un unico ambiente operativo, aumentando significativamente l'efficienza e la produttività.

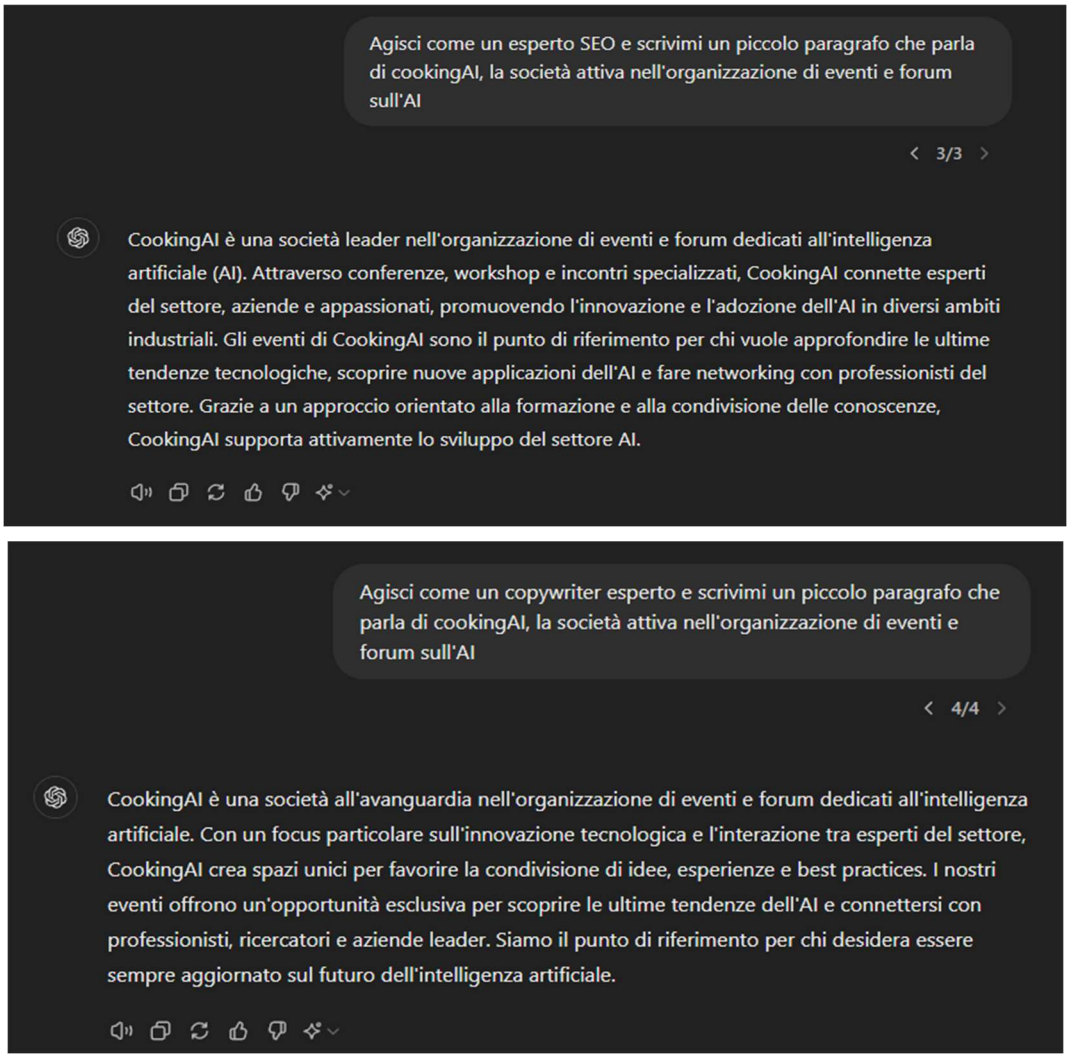

Fig. 7 Esempio di Prompt Persona

In conclusione si può dire che il Prompt Persona non è solo una tecnica, ma un vero e proprio paradigma nell'utilizzo dei modelli linguistici. Grazie alla sua capacità di adattare risposte e comportamenti simulati, rappresenta una leva strategica per sfruttare al massimo le potenzialità degli LLM. Questa metodologia si pone al crocevia tra innovazione tecnologica e applicazioni pratiche, rendendo i modelli linguistici strumenti essenziali per la ricerca, l'industria e la formazione.

5. L'Importanza del Contesto e delle Informazioni Ulteriori

Uno degli aspetti più sottovalutati nel Prompt Engineering è l'importanza del contesto[21]. Ogni prompt deve essere inserito in un contesto preciso per ottenere risposte pertinenti e utili. Fornire al modello linguistico un contesto chiaro è fondamentale per ridurre l'ambiguità delle risposte e migliorare la coerenza complessiva dei risultati. Per analogia, un contesto è come un perimetro che obbliga il LLM a rispondere al suo interno diminuendo quelle che sono le cosiddette "Allucinazioni"[22] ossia errori di contenuto impreciso ma che appare corretto agli occhi di un non esperto. Il contesto è fondamentale nell'ambito di applicazioni avanzate come il RAG (Retrieval Augmented Generation)[23] in cui l'LLM

genera risposte pertinenti ad un contesto specifico di conoscenza che viene imposto all'LLM stesso per circoscrivere la risposta.

Quando il contesto è assente o incompleto, il modello potrebbe rispondere in modo generico o addirittura sbagliato. Ad esempio, un prompt che chiede "Che cos'è il machine learning?" potrebbe generare una risposta valida ma piuttosto generica. Invece, un prompt come "Spiega cos'è il machine learning nel contesto delle applicazioni dell'analisi predittiva" offre un contesto specifico che guida il modello a fornire una risposta più mirata e rilevante.

L'importanza aumenta nel caso di omonimie o di possibili argomenti che hanno un vasto raggio contenutistico. Immaginiamo ad esempio di dover trattare della 'Divina Commedia'. Di seguito alcuni esempi di come il contesto cambia sostanzialmente il risultato del Prompt.

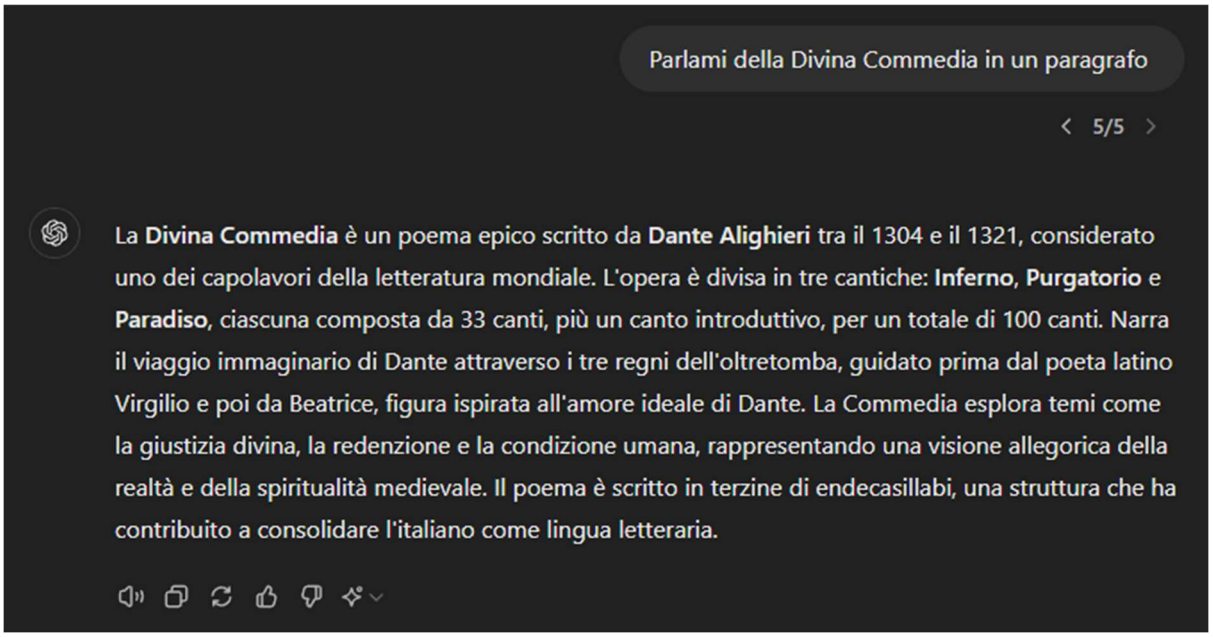

Fig. 8 Esempio di Prompt senza contesto

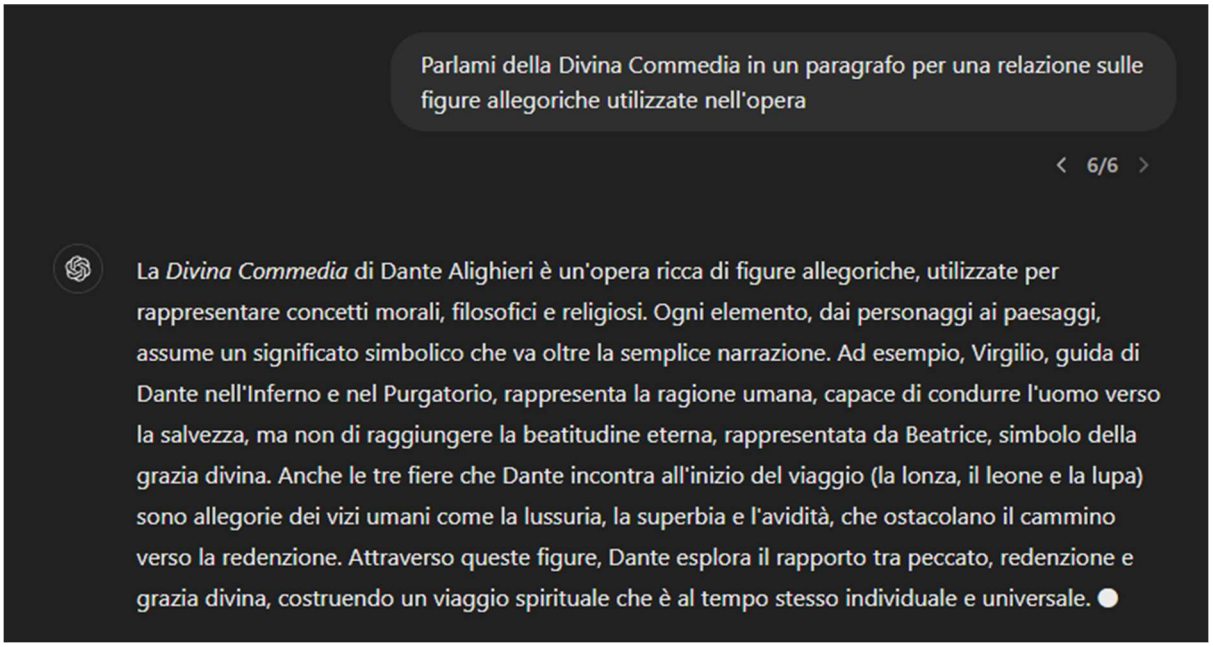

Fig. 9 Esempio di Prompt con un contesto specifico

In sintesi, il contesto permette al modello di:

- Ridurre le possibilità di interpretazione ambigua del prompt.
- Migliorare la coerenza logica delle risposte.
- Offrire risposte più pertinenti rispetto all'obiettivo dell'utente.
- Facilitare l'approfondimento tematico.

6. Target Persona Prompt

Definire il target del contenuto che si desidera produrre è una componente cruciale nel Prompt Engineering, così come lo è specificare il contesto all'interno del quale il modello deve operare.
Entrambi gli elementi agiscono come vincoli strategici, contribuendo a orientare la risposta dell'LLM entro un perimetro ben definito.
Questa capacità di circoscrivere le risposte consente non solo di evitare derive inefficaci, ma anche di massimizzare la pertinenza e la qualità del risultato. Proprio come il contesto aiuta a delimitare il dominio della richiesta, il target permette di definire il pubblico o l'utilizzo finale del contenuto, calibrando il tono, lo stile e la complessità del testo generato in funzione delle esigenze specifiche.
Negli esempi riportati nelle Fig. 10, Fig. 11 e Fig. 12, si può osservare chiaramente come il risultato prodotto dall'LLM vari in modo significativo in base al target specificato, pur mantenendo invariato l'oggetto della richiesta nel prompt. Per esempio, una richiesta che implica la spiegazione di un concetto scientifico avrà una formulazione completamente diversa se il target è costituito da esperti accademici rispetto a studenti delle scuole superiori. Nel primo caso, il linguaggio sarà più tecnico e articolato, con un'attenzione particolare ai dettagli e alla terminologia specialistica; nel secondo, il testo sarà semplificato, con un focus sulla chiarezza e sull'accessibilità. Questa capacità di adattare il contenuto in funzione del target rende il Prompt Engineering uno strumento particolarmente potente per chi deve creare comunicazioni mirate, che siano esse didattiche, divulgative o professionali.
Il target non agisce soltanto come parametro per personalizzare il risultato, ma rappresenta anche una leva fondamentale per indirizzare il percorso logico e stilistico del modello nella generazione del testo. In combinazione con un contesto ben definito, il target diventa parte di una strategia articolata che permette di ottenere risposte non solo corrette, ma anche performanti rispetto agli obiettivi dell'utente. Specificare il target equivale a fornire all'LLM un punto di riferimento chiaro, che guida il modello nella scelta del tono, della struttura e del contenuto più adeguati. Questo processo è particolarmente utile quando si lavora con prompt complessi che richiedono un elevato livello di precisione, come nel caso di report scientifici, strategie di marketing o testi destinati a un'audience specializzata.
Negli esempi seguenti si evidenzia come un differente Target prompt influenza il risultato a parità di oggetto richiesto nel prompt[24].

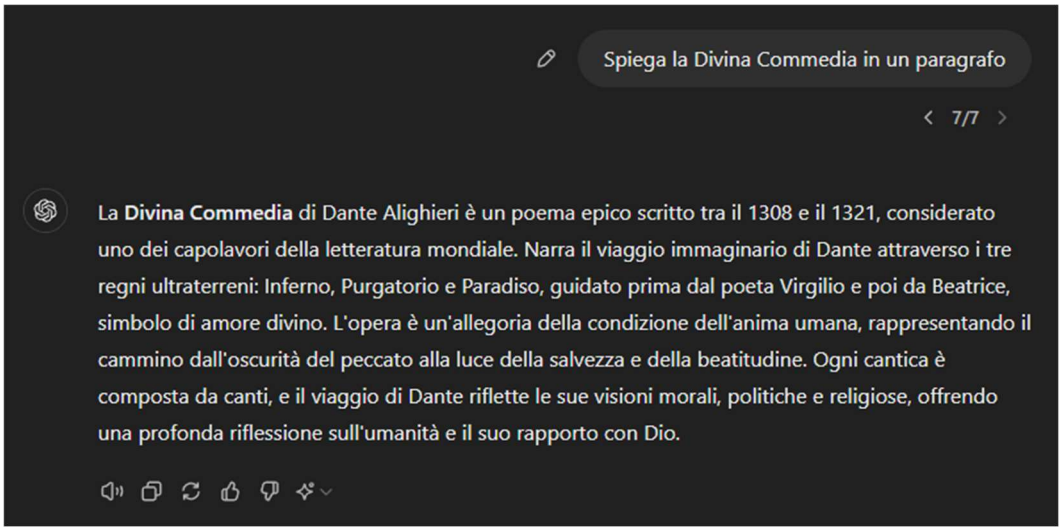

Fig. 10 Esempio di Prompt senza target

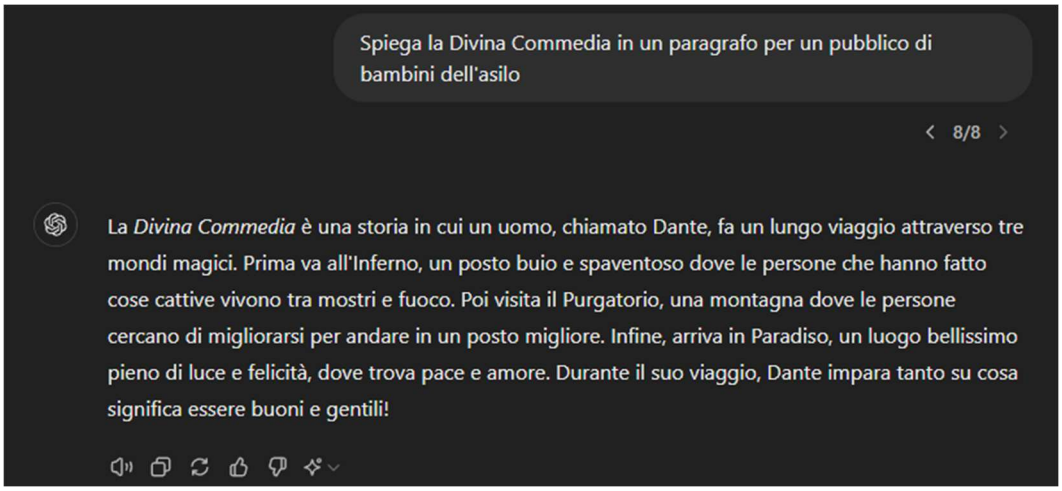

Fig. 11 Esempio di Prompt con un Target Persona ‘Non esperto’

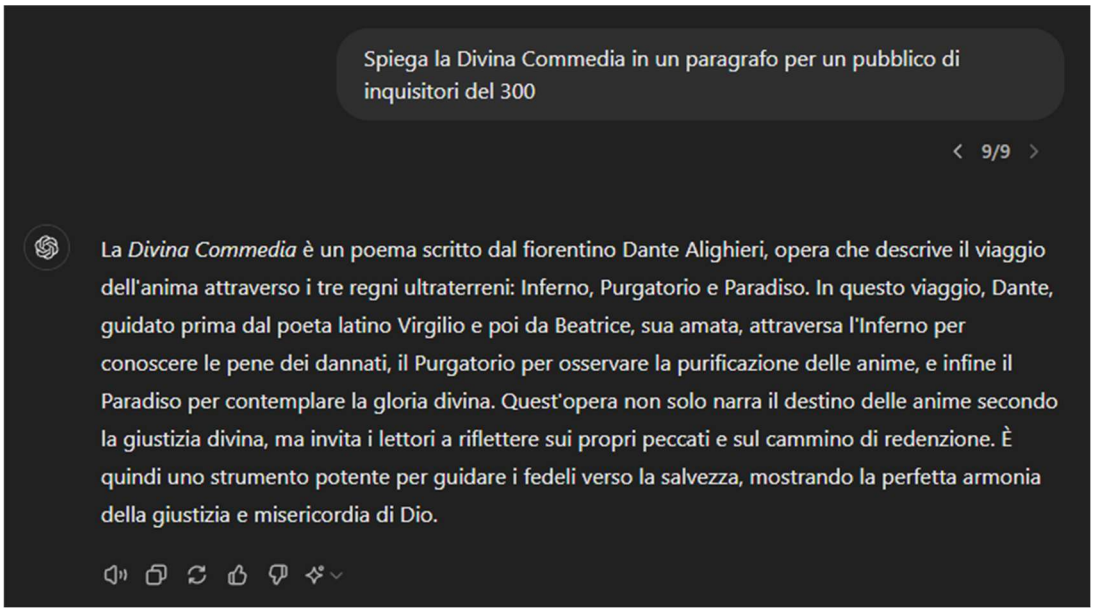

Fig. 12 Esempio di Prompt con un Target Personas ‘Esperto’

Un ulteriore aspetto rilevante è la sinergia che si crea combinando Prompt Persona, Target e Contesto.
Mentre il Prompt Persona definisce il punto di vista o il ruolo che il modello deve assumere (ad esempio, un esperto del settore o un divulgatore), il target indica per chi è pensato il contenuto generato, e il contesto delimita il perimetro entro cui il modello deve operare. Questa triade di elementi non solo aumenta la precisione delle risposte, ma offre all'utente un maggior grado di libertà e creatività nella formulazione delle richieste. Grazie a questa combinazione, l'utente può guidare attivamente il modello verso risultati che non solo soddisfano le aspettative iniziali, ma le superano, abilitando la generazione di contenuti altamente personalizzati e di valore.
La capacità di articolare le richieste in modo preciso, sfruttando i vincoli di contesto e target, rappresenta una vera e propria rivoluzione nel modo in cui si interagisce con i modelli linguistici. Questa metodologia permette di ottenere risposte non solo pertinenti, ma anche perfettamente allineate agli obiettivi dell'utente, rendendo il Prompt Engineering una disciplina fondamentale per chi desidera sfruttare appieno le potenzialità dei LLM. L'approccio che integra Target e contesto con il Prompt Persona non si limita a fornire risposte meccaniche, ma trasforma l'interazione con il modello in un processo collaborativo, in cui l'utente diventa il regista di un output che riflette fedelmente le sue intenzioni e le esigenze del pubblico di destinazione.

7. Ottimizzazione del Prompt per ridurre i costi di fruizione dell'API

Nel contesto dell'utilizzo di modelli linguistici l'ottimizzazione del prompt diventa essenziale per garantire la corretta allocazione delle risorse economiche e computazionali e per ottenere risposte più veloci ed efficienti.
Infatti se si utilizzano modelli LLM a pagamento (come OpenAI, Groq o simili) o modelli serviti su propri server, l'ottimizzazione del prompt permette un efficientamento computazionale e economico, di conseguenza riducendo sia i token prodotti in input sia quelli dell'output.
La formulazione dei prompt può incidere direttamente sul tempo di esecuzione del task e sull'efficienza complessiva del sistema.[25]
Alcuni esempi di come ottimizzare un prompt in questi casi includono:

- Utilizzare istruzioni concise e richiedere al modello di limitare il numero massimo di parole/token nella risposta.
- Prendere in considerazione l'utilizzo di modelli più piccoli se soddisfano i requisiti, poiché i modelli più grandi in genere richiedono più risorse computazionali. Questo è possibile sia se si è utenti finali, utilizzando altri provider anche diversi da OpenAI, sia se stiamo sviluppando software custom e dobbiamo scegliere i modelli più appropriati.
- Utilizzare librerie software Python come "**tiktoken"**[26] di OpenAI per contare preventivamente i token in una stringa di testo senza effettuare una chiamata API nel caso siamo nella fase di sviluppo software.

Di seguito alcuni esempi di prompt ottimizzato.

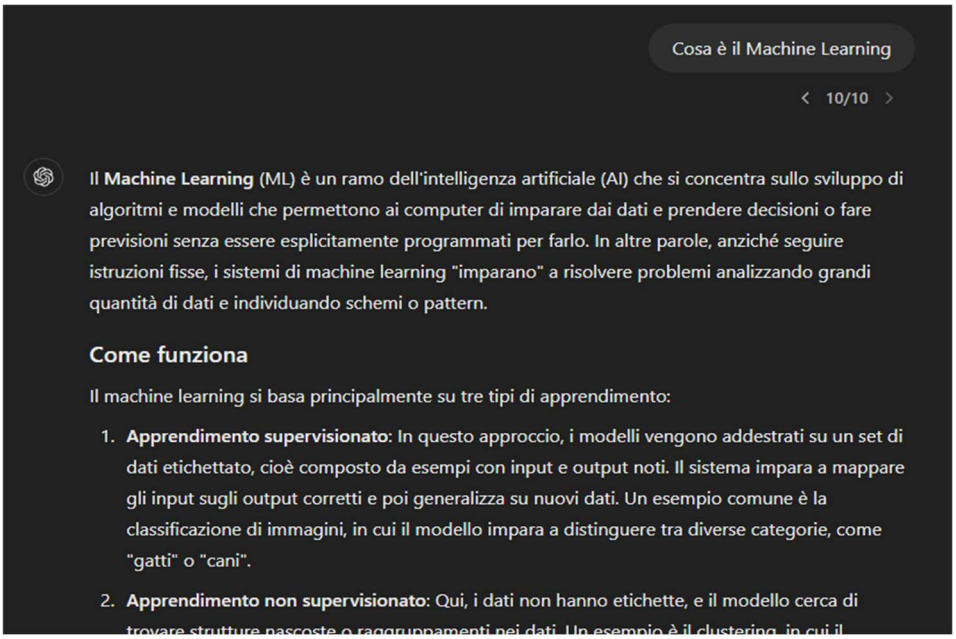

Fig. 13 Prompt non ottimizzato

Come si vede nella Fig 13, il prompt non ottimizzato ha generato una risposta generica, prolissa e articolata secondo un formato deciso autonomamente del modello in funzione di ciò che 'conosce' dal proprio training.
La richiesta ha generato **543** tokens.
Ora procederemo a generare la stessa richiesta costringendo a redigere il tutto in un sono paragrafo.

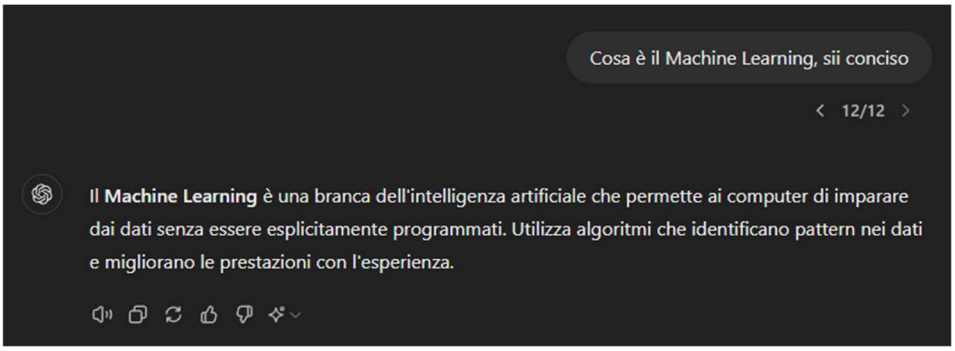

Fig. 14 Prompt ottimizzato per ottenere una risposta corta

Nell'esempio della Fig 14, il sistema risponde producendo **59** token con una riduzione dei token del 90%.
Operativamente, bisogna trovare il giusto trade-off tra sintesi nella risposta ed efficacia del contenuto in funzione dell'obiettivo della richiesta stessa, evitando risposte troppo sintetiche e prive di significato o risposte generiche lunghe.
La tecnica di ottimizzazione del prompt è una tecnica iterativa, ossia sarà l'utente a dover trovare il giusto mix di termini e di composizione del prompt per ottenere risposte efficaci e coincise. In alcuni casi può essere aiutato dallo stesso LLM che può essere istruito a creare prompt ottimizzati. In tal caso il risultato sarà il nuovo prompt da usare in una nuova interrogazione del modello.

8. Zero, One e Few Shot Prompting

In questo paragrafo trattiamo di una funzionalità implicita degli LLM ossia quella di prevedere correttamente strutture sintattiche del linguaggio anche in assenza di esplicite richieste od esempi[27]. Questa capacità è possibile grazie alla conoscenza inclusa nel modello stesso che riesce a produrre output in base al processo inferenziale della rete neurale da cui è costituito.

Questa funzionalità permette di fare le richieste che solitamente facciamo in modalità imperativa/direttiva classica[28][29] o in stile motore di ricerca. L'LLM anticipa quella che può essere la sequenza di parole successive anche senza alcun esplicito esempio. Questo si chiama **ZERO Shot**[30].

Durante il pre-training non supervisionato, un modello linguistico sviluppa un ampio insieme di competenze e capacità di riconoscimento di modelli. Quindi utilizza queste capacità al momento dell'inferenza per adattarsi o riconoscere rapidamente il compito desiderato[31]. Il termine "Context Learning" descrive il ciclo interno di questo processo, che avviene all'interno del forward-pass di ogni ciclo di addestramento.

Un esempio di Zero Shot è il seguente:

Fig. 15 Zero Shot Prompt

Come si vede nella Fig 15 per richiedere un task specifico all'LLM non viene dato alcun esempio di risultato e lui risponde esattamente.

Se invece diamo uno o più esempi siamo nel caso del **One Shot** e **Few Shot**.

Fig. 16 One Shot Prompting

Nella Fig 16 si vede che dando un risultato atteso il LLM completa la risposta non solo facendo correttamente la traduzione ma anche seguendo il formato dell'esempio con la lettera minuscola[32].

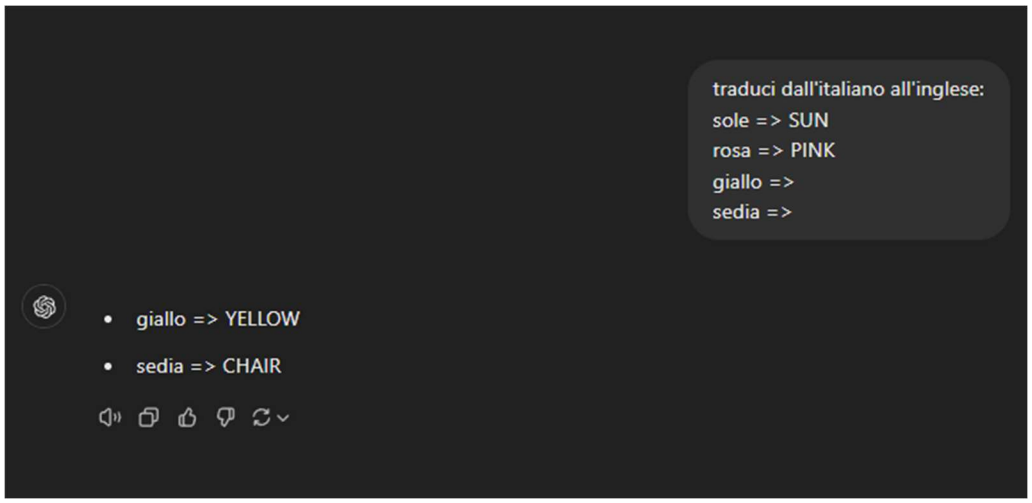

Fig. 17 Few Shot Learning

La Fig. 17 apparentemente non sembra identificare grandi differenze verso la Fig 16 a causa dell'esempio semplice. Tuttavia, la tecnica di fornire esempi più articolati in task più complessi come la simulazione, la summarization, o la produzione di sintesi concettuali più complesse porta l'LLM a seguire un binario preciso di strutturazione della risposta così come desiderato dall'utente.

La tecnica dei few-shot è essenziale nell'interpello degli LLM in flussi e pipeline composite in cui si richiede allo stesso di ottenere un risultato che segua una serie di istruzioni specifiche e di risultati attesi specifici. Il few-shot è anche alla base concettuale di tecniche avanzate come il CoT (Chain of Thoughts) che vedremo più avanti o come i Templates che sono trattati nel paragrafo successivo.

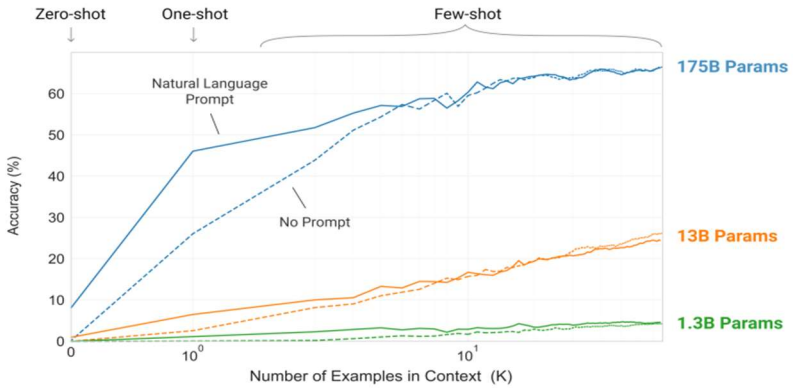

Fig. 18 Modelli più grandi, fanno un uso efficiente delle informazioni di contesto. https://arxiv.org/pdf/2005.14165

La Fig. 18 dimostra proprio questa capacità degli LLM di astrarre il metodo per dare una corretta risposta già nel caso degli Zero Shot. In questa immagine sono mostrate le prestazioni di apprendimento basate sul contesto su un compito semplice che richiede al modello di rimuovere simboli casuali da una parola, sia con che senza una descrizione del

compito in linguaggio naturale. Le "curve di apprendimento in contesto" più ripide per i modelli di grandi dimensioni dimostrano una capacità migliorata di apprendere un compito da informazioni contestuali. Queste capacità poi incrementano in modo diverso in base al numero di esempi forniti e alla grandezza dei modelli usati.

9. Templates

Una delle principali caratteristiche degli LLM, soprattutto quelli di grandi dimensioni come GPT4, è quella di poter strutturare il risultato richiesto secondo un preciso formato di output seguendo le indicazioni date nel prompt[33].
Quando si scrive il prompt bisogna richiedere espressamente il formato di output che si desidera utilizzando indicatori di posizione all'interno di parentesi quadre [ ] o altri placeholder facilmente comprensibili come <> seguendo la logica di tagging tipica dell'XML[25].
Di seguito un esempio in Fig 19.

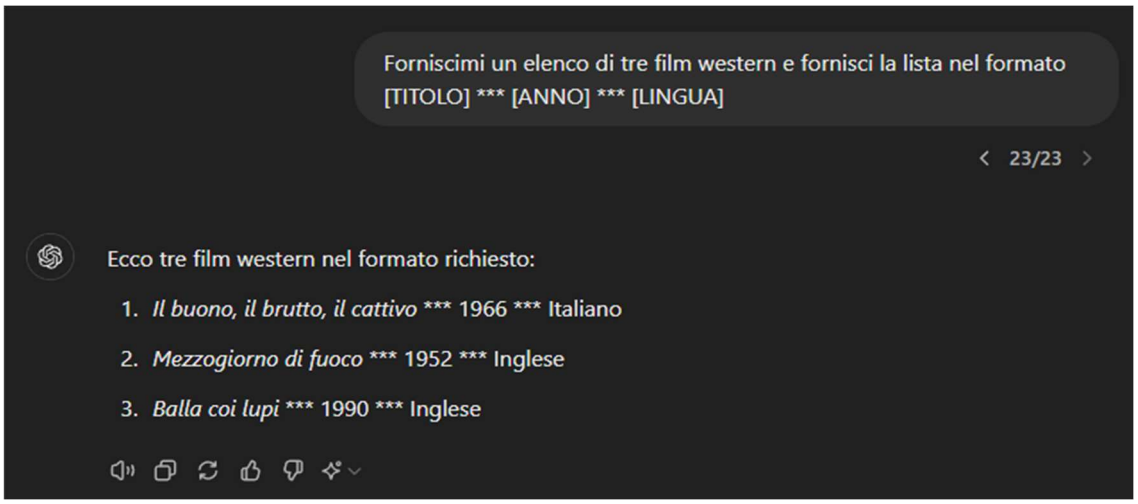

Fig. 19 Template Prompt

Un'estensione dei template è la possibilità di richiedere un output strutturato in formato specifico tipo Json come illustrato in Fig. 20..
Il template illustrato è molto utile per creare set di dati sintetici nel formato adatto per svariati usi (Software Testing o altro simile).
Il templating e i dati strutturati in output sono alla base della tecnica del 'Function Calling' usato all'interno del processo dell'Agentic AI per interpellare l'LLM su un task di classificazione e genericamente di decision making e dando come output un json strutturato.
In tal caso il mix di un LLM già istruito a dare risposte con template strutturato e un prompt ben costruito con la richiesta specifica per un template permettono a livello software di gestire l'attivazione di tool specifici in funzione del flusso di lavoro che emerge dall'interazione
Utente->LLM->Dati[34].

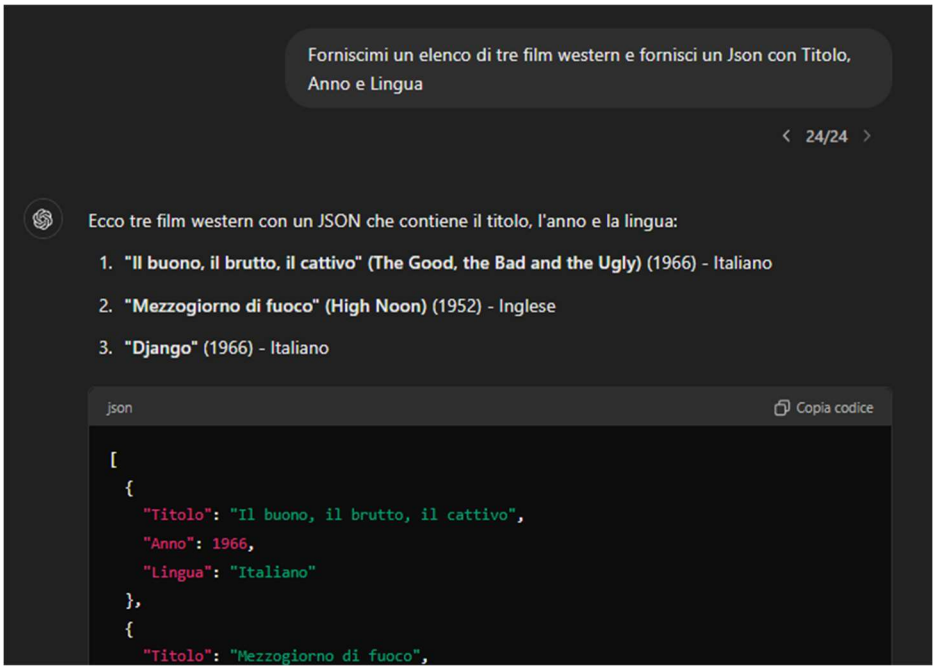

Fig. 20 Prompt con output format in Json

10. La Tecnica del Prompt a Catena di Pensieri (Chain of Thought Prompting)

La tecnica denominata "Prompting a Catena di Pensieri"[35] (Chain of Thought Prompting, CoT) rappresenta un approccio avanzato nel Prompt Engineering, particolarmente adatto per affrontare problemi complessi che richiedono un ragionamento strutturato e sequenziale.

A differenza di altri metodi che mirano direttamente a produrre una risposta, il CoT prompting guida il modello a esplicitare passo dopo passo il processo logico che conduce alla soluzione finale. Questa modalità di interazione si è dimostrata altamente efficace in contesti dove è fondamentale comprendere il percorso deduttivo seguito dal modello, come il calcolo matematico, l'analisi deduttiva o la risoluzione di problemi con molteplici variabili interdipendenti.

Un esempio emblematico dell'efficacia di questa tecnica è illustrato dai risultati riportati nella Fig. 1, dove l'applicazione del Chain of Thought Prompting su un dataset del settore sanitario (Health) ha determinato un significativo incremento dell'accuratezza delle risposte. Questo miglioramento non deriva soltanto dalla capacità del modello di seguire istruzioni dettagliate, ma soprattutto dall'attivazione di un processo di reasoning, ovvero un ragionamento strutturato, che si discosta dalla semplice generazione sequenziale di parole basata su inferenze statistiche. L'utilizzo del CoT prompting è dunque particolarmente indicato per quei compiti che richiedono una comprensione più profonda e articolata del problema, rendendolo uno strumento cruciale per la risoluzione di task complessi e per il miglioramento della qualità delle risposte generate dai modelli di linguaggio.

Esempio:

***D**: Roger ha 5 palline da tennis. Compra altre 2 lattine di palline da tennis, ognuna delle quali contiene 3 palline. Quante palline da tennis ha ora in totale?*

***Risposta con ragionamento passo-passo**:*

Roger ha iniziato con 5 palline.

Ogni lattina contiene 3 palline, quindi 2 lattine contengono 6 palline.

Sommando, 5 + 6 = 11 palline in totale.

***Risposta**: 11 palline.*

Questa metodologia risulta particolarmente utile in contesti educativi, analitici e ingegneristici, dove è importante seguire e validare ogni passaggio di un processo.

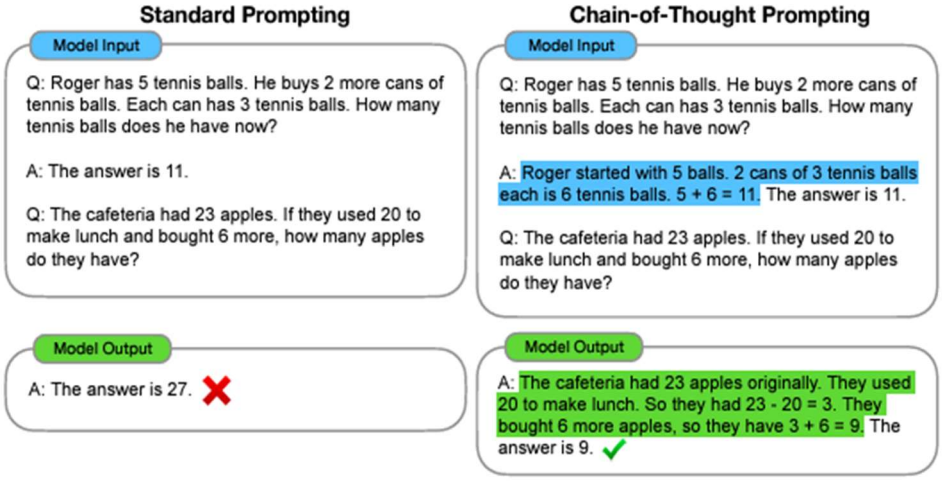

Fig. 21 Esempio di Chain of Thought Prompting - Chain-of-Thought Prompting Elicits Reasoning in Large Language Models - Jason Wei et al. 2023 - https://arxiv.org/pdf/2201.11903

Come evidenziato in alcune analisi statistiche tratte dal paper "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models", citato in precedenza e presente in Fig. 22, i risultati empirici confermano l'efficacia di questa tecnica su una vasta gamma di dataset e tipologie di task. In particolare, l'approccio CoT ha dimostrato una netta superiorità rispetto ai metodi tradizionali, evidenziando un miglioramento significativo non solo nella precisione delle risposte finali, ma anche nella capacità del modello di adattarsi a situazioni in cui è necessario articolare un ragionamento complesso. Questi dati confermano che il CoT prompting non è semplicemente una variazione stilistica nella formulazione dei prompt, ma rappresenta una vera e propria innovazione metodologica nell'interazione con modelli linguistici avanzati.

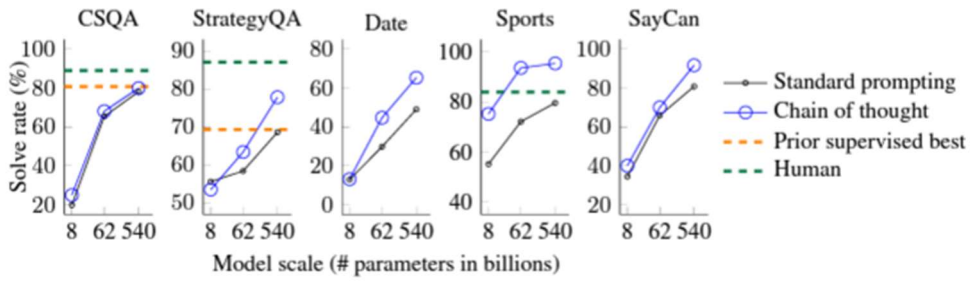

Fig. 22 Percentuale di risoluzione task con e senza CoT - Chain-of-Thought Prompting Elicits Reasoning in Large Language Models - Jason Wei et al. 2023 - https://arxiv.org/pdf/2201.11903

Un aspetto fondamentale del Chain of Thought Prompting risiede nella sua capacità di rendere trasparente il processo deduttivo del modello. Attraverso una serie di domande o passaggi intermedi, il modello viene guidato in un percorso graduale verso la soluzione, esplicitando ogni passo logico compiuto. Questa trasparenza non solo aumenta la comprensione del processo da parte dell'utente, ma consente anche di identificare eventuali errori o incongruenze lungo il percorso.

Ad esempio, nel calcolo matematico, il modello potrebbe scomporre un problema complesso in una serie di equazioni intermedie, mostrando chiaramente come ciascun risultato contribuisca alla soluzione finale. In questo modo, il CoT prompting non si limita a fornire una risposta corretta, ma permette anche di validare e comprendere il ragionamento sottostante, offrendo un livello di affidabilità e controllo superiore rispetto ad approcci meno strutturati.

L'efficacia del Chain of Thought Prompting non si limita tuttavia ai soli ambiti matematici o logici. Questa tecnica trova applicazione in una varietà di settori, inclusi quelli in cui la complessità del problema richiede una suddivisione in passaggi logici chiari e ben definiti. Ad esempio, nell'analisi deduttiva, il CoT prompting può essere utilizzato per costruire argomentazioni coerenti o per analizzare scenari con molteplici variabili. In ambito sanitario, come evidenziato dai risultati empirici, il modello può essere guidato nella valutazione di sintomi e dati clinici, costruendo un percorso logico che porta a una diagnosi ipotetica. Questa flessibilità rende il CoT prompting una soluzione estremamente versatile, capace di adattarsi a contesti e applicazioni molto diversi tra loro. Rappresenta una delle tecniche più promettenti e innovative nel campo del Prompt Engineering. La sua capacità di combinare accuratezza, trasparenza e reasoning lo rende uno strumento fondamentale per sfruttare appieno le potenzialità dei modelli linguistici di nuova generazione. L'esplicitazione del processo logico, unita alla possibilità di verificare ogni passaggio del ragionamento, non solo migliora la qualità delle risposte, ma contribuisce anche a instaurare un livello di fiducia più elevato nell'interazione uomo-macchina. Alla luce dei dati e delle analisi disponibili, il CoT prompting si configura non solo come un metodo efficace, ma come una strategia indispensabile per affrontare le sfide poste dai problemi complessi in molteplici ambiti applicativi.

11. Prompt Chaining: Una Catena di Istruzioni

La tecnica del **Prompt Chaining**[36], nota anche come **Chaining of Prompts**, rappresenta un approccio avanzato nella gestione delle interazioni con i modelli linguistici. Essa si basa sull'utilizzo sequenziale di prompt, in cui l'output generato da un'istruzione diventa l'input per la successiva.

Questo processo permette di affrontare compiti complessi suddividendoli in una serie di sotto-attività interconnesse, agevolando la concentrazione del modello su ciascun passaggio in modo mirato.

L'idea alla base di questo approccio è quella di scomporre una richiesta articolata in fasi più gestibili, ottimizzando così la qualità e la coerenza delle risposte. Suddividere un problema in step successivi non solo consente di semplificare la complessità del compito, ma garantisce anche una maggiore precisione in ogni fase del processo, evitando che il modello sia sopraffatto da istruzioni troppo dettagliate o conflittuali in un'unica soluzione.

Il Prompt Chaining trova applicazione in scenari particolarmente utili in cui la complessità di un'attività non può essere risolta attraverso un singolo passaggio.

Ad esempio, nei compiti che richiedono l'esecuzione di più step distinti, come la ricerca preliminare su un argomento, la stesura di una bozza e infine la formattazione di un documento completo, questa tecnica permette di affrontare ogni fase con un focus esclusivo e calibrato.

L'uso del Prompt Chaining assicura che ogni passaggio riceva un'attenzione adeguata, migliorando così la qualità complessiva del risultato finale. Un altro vantaggio significativo emerge quando si ha a che fare con prompt particolarmente complessi.

In questi casi, un unico input può risultare sovraccarico di istruzioni, portando il modello a generare risposte incoerenti o parzialmente errate. Spezzare il compito in sotto-task ben definiti consente invece di migliorare le prestazioni del modello, assicurandosi che ogni sotto-attività venga completata in modo ottimale e coerente rispetto agli obiettivi iniziali.

Un ulteriore ambito di applicazione particolarmente interessante del Prompt Chaining riguarda la possibilità di utilizzare il modello non solo come generatore di contenuti, ma anche come revisore[37][38].

Questo approccio permette di iterare sui risultati intermedi, chiedendo al modello di valutare la propria produzione per correggere eventuali errori o migliorarne l'efficacia. Ad esempio, dopo aver richiesto un elenco di argomenti per un blog, è possibile passare l'output al modello per verificarne la pertinenza e suggerire eventuali integrazioni o revisioni. In un passaggio successivo, il contenuto aggiornato può essere formattato per la pubblicazione finale. Questa capacità di auto-valutazione e miglioramento rende il Prompt Chaining una tecnica estremamente versatile e adatta a contesti che richiedono elevati standard di qualità e precisione. L'interazione sequenziale tra i vari prompt, quindi, non solo garantisce un controllo maggiore sui risultati, ma introduce anche un livello di supervisione che è difficile da ottenere con prompt singoli e isolati.

Tuttavia, l'efficacia del Prompt Chaining può essere limitata da alcune condizioni operative, in particolare quando si utilizza un'interfaccia web che non supporta la sequenzialità nativa delle richieste. In questi casi, l'implementazione di catene di prompt potrebbe risultare meno fluida, soprattutto in assenza di strumenti automatizzati per la gestione dei flussi. Questo limite è tuttavia superabile in contesti di programmazione con LLM integrati in workflow predefiniti, dove le Catene di Prompt possono essere configurate in modo parallelo o con diramazioni logiche. Questi flussi possono includere punti decisionali intermedi, seguendo una logica simile a quella dei flowchart utilizzati nella programmazione tradizionale.

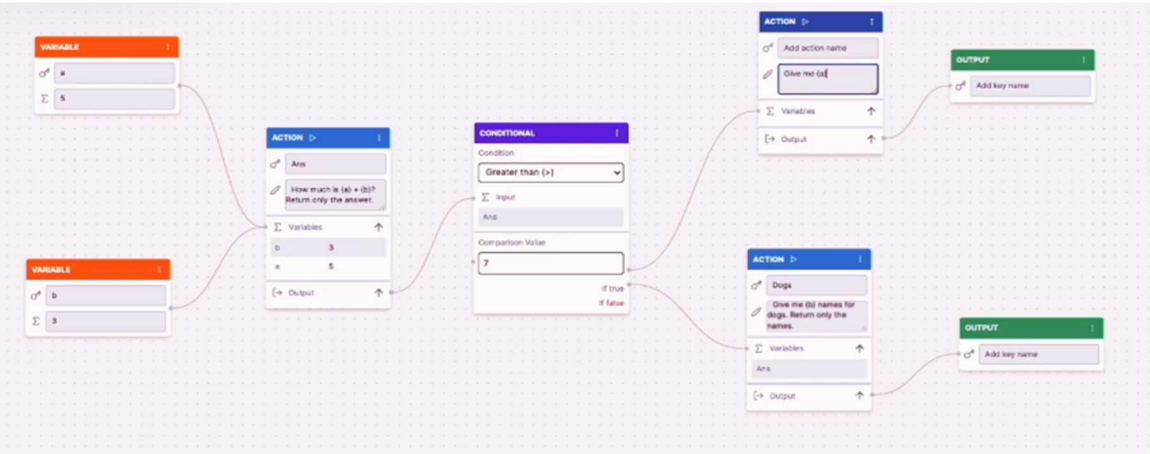

Fig. 23 Esempio di GUI per gestire il workflow del Prompt Chaining

Come evidenziato nella **Fig. 23**, dove è presentato un esempio di applicazione che simula il funzionamento di questo processo[39], è possibile simulare un processo strutturato che sfrutti il Prompt Chaining per generare, verificare e ottimizzare contenuti attraverso una sequenza di passaggi interconnessi.
In questo modo, si aprono nuove possibilità per utilizzare i modelli linguistici non solo come strumenti di generazione, ma come elementi attivi in flussi complessi di gestione e supervisione dei dati
In sintesi, il Prompt Chaining si configura come una metodologia avanzata per la gestione di compiti articolati, capace di migliorare significativamente la precisione e l'efficienza dei modelli linguistici.
La capacità di frammentare processi complessi in fasi autonome ma interconnesse rappresenta un valore aggiunto per numerosi ambiti applicativi, dalla creazione di contenuti alla programmazione di workflow automatizzati.
Pur presentando alcune limitazioni legate all'uso in ambienti meno strutturati, questa tecnica offre una versatilità senza pari quando integrata in sistemi che ne supportano appieno le potenzialità.
Grazie alla sua adattabilità e al controllo granulare che permette di esercitare su ogni fase del processo, il Prompt Chaining si rivela una risorsa fondamentale per sfruttare al meglio le capacità dei Large Language Models.

12. Strategie per ridurre i costi delle API OpenAI

Nel campo in continua evoluzione dell'intelligenza artificiale, l'utilizzo di API come quelle offerte da OpenAI è diventato essenziale per sviluppatori e aziende.
Tuttavia, i costi associati a questi servizi possono rappresentare un ostacolo significativo come già detto in precedenza.
Continuiamo la nostra analisi focalizzandoci su OpenAI, ma il processo è simile per ogni player del mercato degli LLM a pagamento.
Per gestire efficacemente i costi, è fondamentale comprendere il modello di prezzi; quello di OpenAI è presente all'indirizzo https://openai.com/api/pricing/.
I costi delle API OpenAI sono calcolati in base al numero di token elaborati, con tariffe differenti per i vari modelli disponibili.
Una delle principali strategie per la riduzione dei costi è l'ottimizzazione dell'uso dei token.
Questo consiste nel creare input per l'API che minimizzino il numero di token, senza compromettere la qualità delle risposte. Ad esempio, rendere le risposte JSON più leggere eliminando spazi e interruzioni di linea inutili può portare a risparmi fino al 30%.
Oltre al Prompt Engineering appena affrontato, è possibile operare sulla gestione delle logiche di retry. Nel caso in cui una richiesta all'API fallisca, una logica di retry ben progettata può evitare addebiti aggiuntivi, assicurando che l'API non venga richiamata inutilmente.
Inoltre, ogni modello ha anche delle variabili che possono essere tarate per influenzare il risultato. È questo il caso del parametro "Temperature".
Il parametro "Temperature" nelle API di OpenAI influenza la creatività e la casualità delle risposte generate. Impostando un valore di Temperature più basso, si ottengono risposte più determinate e strutturate, riducendo così il numero di token necessari e abbassando i costi dell'API.
La scelta del modello più efficiente per l'attività specifica può portare a significativi risparmi. OpenAI offre una varietà di modelli con capacità e costi diversi.

Ad esempio, il modello gpt-3.5-turbo è un'opzione economica e potente, spesso utilizzata per una vasta gamma di applicazioni.
Alla data di redazione di questo rapporto tecnico sono disponibili nuovi e più performanti modelli di OpenAI rilasciati nel mese di Settembre 2024. Di seguito le descrizioni dei modelli disponibili come presenti sul sito[40]:

GPT-4o

- Latest, fastest, highest intelligence model.
- 128k context length (i.e. an average to longer novel).
- Text and image input / text and image output.*
- Audio input / output.**

GPT-4o mini

- Lightest-weight intelligence model.
- 128k context length (i.e. an average to longer novel).
- Text and image input / text and image output.*
- Audio input / output.**
- Limitation: This model does not have access to the advanced tools that GPT-4o has.

GPT-4

- Previous high intelligence model.
- 128k context length (i.e. an average to longer novel).
- Text and image input / text and image output.*
- Audio input / output.**

GPT-3.5 (API only)

- Fast model for the simplest routine tasks.
- 16k context length (i.e. 1-2 dozen articles or a short story / novella).
- Text input / text output.
- Audio input / output.**

È importante, inoltre, rivedere regolarmente l'utilizzo dell'API e adattare le strategie di conseguenza. Questo approccio dinamico garantisce che le misure di risparmio siano sempre allineate con i più recenti modelli di prezzo e le capacità delle API.
In sintesi, la leva fondamentale è il costante aggiornamento dell'evoluzione dei modelli e del relativo pricing perché la tendenza del mercato degli LLM è quella del minor prezzo/token per i modelli a pagamento e dall'altra parte avremo modelli open sempre più performanti.

13. Strumenti di ottimizzazione per la corretta scrittura del Prompt

L'utente ordinario degli LLM proviene da un'esperienza consolidata dell'utilizzo dei motori di ricerca per cui è estremamente innaturale cambiare modalità di interrogazione di strumenti di ricerca testuale, finora perseguita con la logica delle Keyword.

Il Prompt Engineering va proprio in questa direzione, ossia quella di miscelare lo stile di ricerca tradizionale a query, con tecniche efficaci di interrogazione di modelli di Linguaggio Naturale multidimensionali e di grandi dimensioni[41].

Per venire incontro a queste difficoltà molti utenti e istituzioni in rete hanno iniziato a pubblicare tool utili per la scrittura assistita di prompt adeguati alle varie esigenze, dal marketing, alle vendite alla formazione ecc.

Una libreria utile è quella fornita da Anthropic che è presente all'indirizzo:

https://docs.anthropic.com/en/prompt-library/library .

Per OpenAI invece è possibile utilizzare i cosiddetti GPTs creati da altri utenti che creano prompt su richiesta come il seguente:

https://chatgpt.com/g/g-jlCv8ewMa-prompt-library ,

14. Conclusioni

Il Prompt Engineering si configura come una strategia fondamentale per valorizzare al massimo le potenzialità dei modelli linguistici avanzati, come GPT-4. Attraverso l'uso mirato di tecniche di ottimizzazione dei prompt, una gestione consapevole ed efficiente dei token, l'applicazione di contesti ben definiti e l'impiego di metodologie avanzate come il Prompt Chaining e il Few-Shot Prompting, è possibile ottenere risultati caratterizzati da maggiore precisione, coerenza e pertinenza rispetto agli obiettivi prefissati.

Più di una semplice tecnica, il Prompt Engineering rappresenta una competenza strategica che consente non solo di migliorare l'efficacia delle interazioni con i modelli di intelligenza artificiale, ma anche di adattarli con versatilità a una vasta gamma di contesti applicativi e operativi.

Questa disciplina, in continua evoluzione, riflette la crescente necessità di personalizzazione e ottimizzazione nelle interazioni uomo-AI, rispondendo alle sfide poste dalla complessità dei compiti e dalla diversificazione delle esigenze.

In un panorama tecnologico sempre più orientato all'efficienza e alla scalabilità, il Prompt Engineering non si limita a facilitare l'utilizzo dei modelli linguistici, ma li trasforma in strumenti potenti e dinamici, capaci di generare contenuti e soluzioni di alta qualità con maggiore rapidità. Questa capacità di coniugare personalizzazione e performance sottolinea il ruolo cruciale del Prompt Engineering come leva strategica per massimizzare il valore delle risorse computazionali e per aprire nuove frontiere nelle applicazioni dell'intelligenza artificiale.

15. Riferimenti

[1]OPENAI ChatGPT https://openai.com/chatgpt/

[2]CLAUDE https://claude.ai/

[3] https://www.anthropic.com/

[4] Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. - 2023 - Unleashing the potential of prompt engineering in large language models: a comprehensive review - https://arxiv.org/abs/2310.14735

[5]Attention Is All You Need - A Vaswani et al. - Advances in Neural Information Processing Systems, 2017 - https://arxiv.org/abs/1706.03762

[6] GPT-4 Technical Report 2023 https://arxiv.org/abs/2303.08774

[7] Large Language Models: A Survey - Shervin Minaee et al. Feb 2024 - https://arxiv.org/pdf/2402.06196

[8] Scaling Laws for Neural Language Models - Jared Kaplan et al. 2020 - https://arxiv.org/abs/2001.08361

[9] Llama 2: Open Foundation and Fine-Tuned Chat Models – Touvron et al. 2023 - https://arxiv.org/abs/2307.09288

[10] On the Opportunities and Risks of Foundation Models - Bommasani et al. 2021 - https://arxiv.org/abs/2108.07258

[11] Learning transferable visual models from natural language supervision. In International conference on machine learning - Radford et al. 2021 - https://arxiv.org/pdf/2103.00020

[12] Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing Liu et al. 2023 - https://dl.acm.org/doi/pdf/10.1145/3560815

[13] Harsha Nori, Yin Tat Lee, Sheng Zhang et al. - 2023 - Can Generalist Foundation Models Outcompete Special-Purpose Tuning? Case Study in Medicine - https://arxiv.org/abs/2311.16452

[14] Risposte a domande (Q&A) a scelta multipla basate sugli esami di licenza medica degli Stati Uniti (USMLE). Il set di dati viene raccolto dagli esami della commissione medica professionale. Copre tre lingue: inglese, cinese semplificato e cinese tradizionale e contiene rispettivamente 12.723, 34.251 e 14.123 domande per le tre lingue.

[15] Una guida ai principali Tokenizer SOTA è presente sul sito Huggingface

[16] 5 Technical Reasons Why GPT Misinterprets 'Ramarro': A Case Study in Multilingual Tokenization https://www.linkedin.com/posts/lorenzo-de-tomasi-ai-data-platforms_nlp-multilingualai-machinelearning-activity-7237752455948623872-LgIS

[17] Understanding LLMs: A Comprehensive Overview from Training to Inference – Liu et al. 2024 - https://arxiv.org/html/2401.02038v2

[18] Similarità per la ricerca del dominio di una frase - Morelli et al. 2020 - https://arxiv.org/pdf/2002.00757

[19] Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models - Wang et al. 2023 - https://arxiv.org/abs/2310.00746

[20] Ethan Mollick, Lilach Mollick - 2023 - Assigning AI: Seven Approaches for Students, with Prompts https://arxiv.org/abs/2306.10052

[21] Deconstructing In-Context Learning: Understanding Prompts via Corruption - Shivagunde et al. 2024 - https://arxiv.org/abs/2404.02054

[22] https://en.wikipedia.org/wiki/Hallucination_(artificial_intelligence)

[23] Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks - Lewis et al. 2020 - https://arxiv.org/abs/2005.11401

[24] When “A Helpful Assistant” Is Not Really Helpful: Personas in System Prompts Do Not Improve Performances of Large Language Models - Zheng et al. 2024 - https://arxiv.org/pdf/2311.10054v2

[25] Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4 - Bsharat et al. 2024 - https://arxiv.org/pdf/2312.16171

[26] https://github.com/openai/tiktoken

[27] Language Models are Few-Shot Learners - Brown et al. 2020 - https://arxiv.org/abs/2005.14165

[28] John R. Searle. 1969. Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press.

[29] The Prompt Report: A Systematic Survey of Prompting Techniques - Sander et al. 2024 - https://arxiv.org/abs/2406.06608

[30] Takeshi Kojima, Shixiang Shane Gu, Machel Reid - 2022 - Large Language Models are Zero-Shot Reasoners - https://arxiv.org/abs/2205.11916

[31] Hai Dang, Lukas Mecke, Florian Lehmann, Sven Goller, and Daniel Buschek - 2022 - How to prompt? opportunities and challenges of zero- and few-shot learning for human-ai interaction in creative applications of generative models. - http://arxiv.org/abs/2209.01390

[32] Robert L. Logan IV, Ivana Balažević, Eric Wallace et al. - 2021 - Cutting Down on Prompts and Parameters: Simple Few-Shot Learning with Language Models - https://arxiv.org/abs/2106.13353

[33] Let Me Speak Freely? A Study on the Impact of Format Restrictions on Performance of Large Language Models Rui et al. 2024 - https://arxiv.org/pdf/2408.02442

[34] Lennart Meincke, Ethan R. Mollick, Christian Terwiesch - 2024 - Prompting Diverse Ideas: Increasing AI Idea Variance - https://arxiv.org/abs/2402.01727

[35] Chain-of-Thought Prompting Elicits Reasoning in Large Language Models - Jason Wei et al. 2023 - https://arxiv.org/pdf/2201.11903

[36] Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. CHI Conference on Human Factors in Computing Systems. Wu et al. 2022.

[37] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han et al. - 2022 - Large Language Models Are Human-Level Prompt Engineers - https://arxiv.org/abs/2211.01910

[38] Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek - 2022 - Teaching Small Language Models to Reason - https://arxiv.org/abs/2212.08410

[39] https://promptchainer.io/

[40] https://help.openai.com/en/articles/7102672-how-can-i-access-gpt-4-gpt-4-turbo-gpt-4o-and-gpt-4o-mini

[41] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du - 2022 - ReAct: Synergizing Reasoning and Acting in Language Models - https://arxiv.org/abs/2210.03629