

Obiettivi del manuale

L'obiettivo di questa guida è presentare delle competenze di base sulla scrittura dei prompt in grado di fornire risultati immediati a problemi comuni. La guida è divisa in due sezioni principali e da un'appendice tecnica. Nella prima sezione viene definito il dominio operativo con una descrizione sintetica dei Large Language Models (LLM), dei modelli GPT e dell'interfaccia più conosciuta verso questi modelli: ChatGPT.

Cosa troverete nella guida:

1. Descrizione dei LLM e dei modelli GPT
2. Descrizione di ChatGPT e dei suoi campi di applicazione più frequenti
3. Indicazioni pratiche su come costruire una efficace strategia di scrittura del prompt
4. Libreria dei prompt commentata, con esempi concreti e originali, provati tutti in lingua italiana
5. Uso semplificato dei modelli GPT-3.5, GPT-4o e GPT-4 tramite script python generati da ChatGPT utilizzando le API dedicata
6. Descrizione delle strategie di scrittura dei prompt
7. Lista di modelli di prompt personalizzabili catalogati per area di interesse.

Cosa non troverete nella guida:

1. Metodi di integrazione di ChatGPT con applicazione terze
2. Guida all'uso delle API di OpenAI
3. Uso avanzato dei modelli GPT-3.5, GPT-4o e GPT-4 tramite script ed API
4. Guida all'integrazione di servizi terzi per la creazione di prompt concatenati, ad esempio LangChain
5. Tecniche di scrittura avanzata dei prompt fuori dall'interfaccia di ChatGPT

Attraverso questa guida potrete utilizzare al meglio ChatGPT per la produttività quotidiana con esempi pratici e documentati da spiegazioni esaustive.

Cosa sono i LLM

I Large Language Models (LLM) sono modelli di intelligenza artificiale sviluppati appositamente per analizzare, generare e manipolare il linguaggio naturale umano. L'evoluzione dei LLM inizia negli anni '50 e '60 con le prime esperienze nell'ambito delle intelligenze artificiali, esperienze limitate però dalla potenza di calcolo disponibile e da un approccio principalmente basato su regole linguistiche dichiarate in fase di programmazione. Negli anni '80 e '90 inizia una seconda fase per le intelligenze artificiali con un percorso e delle soluzioni che si avvicinano ai moderni LLM, questo avvicinamento si deve alla scelta di utilizzare modelli statistici come alternativa alla dichiarazione di rigide regole. Una strategia

efficace resa possibile grazie anche alle maggiori capacità di calcolo e alla disponibilità di ampi corpora di testo, questo periodo è caratterizzato dall'uso di modelli di Markov nascosti e di approcci probabilistici per il processamento del linguaggio naturale (NLP).

L'inizio del XXI secolo ha visto l'introduzione dei *word embeddings*, che hanno migliorato la capacità dei modelli di catturare significati e relazioni semantiche, tuttavia, è stato soltanto con l'avvento dell'architettura Transformer nel 2017¹ che si è assistito a un vero e proprio salto di qualità, grazie alla sua capacità di gestione delle relazioni semantiche su testi complessi. Queste architetture utilizzano un metodo per mantenere il software focalizzato sul contesto del testo chiamato "*multi head attention model*", una soluzione che consente al modello di concentrarsi su diverse parti del testo simultaneamente e di ponderare l'importanza relativa di differenti parole in una frase, focalizzandosi su quelle più rilevanti per il contesto specifico. Questo meccanismo di attenzione, combinato con la capacità dei Transformer di gestire sequenze di dati di lunghezza variabile, ha notevolmente migliorato la capacità dei modelli di elaborare e generare linguaggio naturale in modo coerente e contestualmente accurato.

L'ultimo tassello nella nascita dei LLM come li conosciamo avviene nel 2018, con l'introduzione dei modelli GPT (GPT-2, GPT-3) da parte di OpenAI caratterizzati da capacità di generazione del testo e comprensione linguistica avanzata, un capacità resa possibile grazie all'addestramento su dataset di grandi dimensioni. Questi modelli hanno stabilito nuovi standard per l'interazione naturale uomo-macchina e hanno trovato applicazioni in numerosi campi anche oltre il linguaggio naturale come lo sviluppo software e l'analisi dati. La terza versione di GPT, GPT-3, lanciata nel 2020, ha rappresentato un punto di svolta, dimostrando capacità di generazione del linguaggio ancora più evolute, inoltre la scelta di OpenAI di rilasciare un'interfaccia pubblica, conversazionale e gratuita del modello (ChatGPT) ha permesso l'accesso alle funzionalità dei LLM ad un pubblico vasto e non specializzato.

Parallelamente allo sviluppo di GPT, aziende come Cohere e Anthropic hanno presentato i propri modelli LLM, puntando su aspetti diversi come un maggiore rispetto di principi etici e una flessibilità nelle integrazioni con prodotti terzi quali app e più in generale software. Questi modelli offrono capacità di comprensione e generazione del linguaggio che trovano applicazione in una vasta gamma di contesti, dall'assistenza clienti alla moderazione dei contenuti, dalla personalizzazione delle esperienze utente all'analisi sentimentale.

L'addestramento dei LLM richiede enormi dataset di testo, che possono includere libri, articoli, siti web e altri testi disponibili pubblicamente, da cui i modelli apprendono come predire la parola successiva in una frase, acquisendo così una comprensione implicita non solo della lingua e della grammatica, ma anche del contesto e di vari aspetti culturali. Un processo che consente ai modelli di affinare le proprie capacità e di adattarsi a sfide linguistiche complesse, offrendo risposte sempre più accurate e pertinenti. Questo processo si avvale di tecniche di apprendimento automatico su larga scala, in particolare

¹ <https://arxiv.org/abs/1706.03762>

l'apprendimento supervisionato in cui viene fornito al computer una grossa mole di set di dati con informazioni complete e descrizioni dettagliate, così da rendere questo corpus di informazioni la “storia” delle esperienze del computer. Quando questa macchina si troverà di fronte ad un problema attingerà alla sua memoria di esperienze per cercare una soluzione idonea. In altri casi vengono utilizzate tecniche di apprendimento non supervisionato in cui vengono forniti al computer set di dati senza alcuna indicazione del risultato atteso. Lo scopo del software è quello di trovare un pattern comune a tutti quei dati e risalire ad uno schema che li legghi.

Un elemento distintivo dell'addestramento dei LLM è il concetto di pre-training seguito da fine-tuning. Durante la fase di pre-training, un modello viene addestrato su una raccolta estesa di testo generico per imparare una comprensione generale della lingua, della sua struttura e del contesto in cui le parole sono utilizzate. Successivamente, nella fase di fine-tuning, il modello può essere ulteriormente addestrato su un dataset più specifico o per compiti particolari, come la risposta a domande o la traduzione tra lingue diverse, questo approccio permette ai modelli di mantenere una versatilità ampia, pur essendo in grado di specializzarsi per compiti o domini specifici.

ChatGPT

ChatGPT è un'interfaccia per l'uso di modelli di intelligenza artificiale sviluppata da OpenAI, e basata sull'architettura dei modelli di linguaggio trasformativi GPT (Generative Pre-trained Transformer). La sua prima versione, GPT-1, è stata lanciata nel 2018, seguita da GPT-2 nel 2019 e GPT-3 nel 2020. ChatGPT è una versione specializzata, progettata specificamente per generare risposte in forma di dialogo, simili a quelle umane.

Come abbiamo visto in precedenza, i modelli GPT sono basati sull'architettura Transformer e sono addestrati su grandi dataset di testo, imparando progressivamente a prevedere la prossima parola in una frase, basandosi sulle parole precedenti e a mantenere la previsione coerente con il contesto dell'intera frase o in alcuni casi dell'intero paragrafo.

I modelli GPT, vengono addestrati in due fasi: pre-addestramento e affinamento, nella fase di pre-addestramento, il modello viene esposto a grandi quantità di testo e impara a prevedere la prossima parola in una sequenza. Durante la fase di affinamento, il modello viene ulteriormente addestrato su dati più specifici per adattarsi a compiti particolari, come nel caso di ChatGPT rispondere alle domande per intraprendere una possibile conversazione con l'utente. ChatGPT, in particolare, è stato ottimizzato attraverso tecniche di addestramento supervisionato e di apprendimento con rinforzo, fasi in cui gli sviluppatori forniscono al modello esempi specifici di dialoghi e conversazioni, aiutandolo a modellare risposte più coerenti e contestualmente appropriate, questo addestramento aggiuntivo è fondamentale per migliorare la qualità delle interazioni in termini di pertinenza, accuratezza e naturalezza del linguaggio.

L'interazione con ChatGPT avviene attraverso l'invio di un comando sotto forma di domanda o esplicita richiesta, la costruzione di queste indicazioni non segue una sintassi rigida, come avviene ad esempio in un linguaggio di programmazione, ma è strutturata in linguaggio naturale, tuttavia nella stesura è importante mantenere un rigore logico e una forma chiara e senza ambiguità così da permettere al modello di individuare l'esatto contesto e generare delle risposte pertinenti. La tecnica di progettazione di un comando esatto per i modelli simili a ChatGPT è definita *prompt engineering*.

Il prompt engineering, ovvero la progettazione di prompt, è un aspetto cruciale nell'uso di ChatGPT, il risultato infatti dell'interazione con il modello dipende da un prompt ben progettato in grado di guidare la generazione di risposte più precise e aderenti alle richieste dell'utente. Quando si fornisce un prompt a ChatGPT, il modello utilizza il contesto fornito per guidare la sua risposta, tuttavia, non si tratta solo di una semplice ricerca nel database di informazioni, il modello, avendo analizzato miliardi di parole e testi durante la fase di addestramento, genera una risposta basata sulle probabilità di quale parola o frase potrebbe seguire logicamente il prompt.

Possiamo immaginare il prompt come l'inizio di una fiaba e la risposta di ChatGPT come lo svolgimento della fiaba inventato per l'occasione sulla base dell'analisi di tante fiabe.

Con l'avvento delle architetture multimodali nei modelli GPT, la scrittura del prompt permette anche la generazione di contenuti non testuali, ad esempio è possibile avviare la generazione di immagini oppure l'analisi dati oppure l'analisi di interi documenti di testo inviati come file. Questi modelli sfruttano la potenza dei Large Language Models (LLM) per comprendere e produrre linguaggio naturale, integrando allo stesso tempo capacità visive per analizzare e interpretare immagini, ciò è reso possibile attraverso tecniche di deep learning avanzate, come le reti neurali trasformative, che sono state adattate per lavorare con dati non solo testuali ma anche visivi. GPT-4, ad esempio, può comprendere una domanda espressa in parole e cercare risposte pertinenti analizzando sia testi che immagini, oppure generare descrizioni testuali dettagliate a partire da input visivi.

ChatGPT 4 utilizza l'architettura multimodale del modello GPT-4 che ha permesso di passare da una semplice elaborazione del linguaggio naturale, alla generazione di immagini oppure all'analisi di dataset in formati come CSV, JSON e XML.

Architetture multimodali

Le architetture multimodali quindi permettono ai modelli di elaborare diverse forme di input, come testo, immagini e dati, questi modelli non si limitano a "leggere" il linguaggio: possono "vedere" e "comprendere" dati strutturati, questa versatilità rende semplice per gli utenti caricare e analizzare diversi tipi di file con risultati ottimali soprattutto con file che contengono dati strutturati come appunto dataset o fogli di calcolo. Dopo il caricamento, il modello può eseguire diverse operazioni sui dati, come filtrare le informazioni, eseguire aggregazioni o trovare correlazioni e al contempo essere di supporto anche nella rappresentazione visiva dei dati, come grafici o tabelle.

Il prompt

Il prompt è un input testuale fornito dall'utente che funge da punto di partenza per la generazione delle risposte da parte del sistema di intelligenza artificiale (IA) di ChatGPT, questo input può assumere diverse forme: una domanda, un'affermazione, un'istruzione, o una combinazione di questi elementi. In termini più ampi, un prompt è una sorta di innesco per la conversazione, una richiesta di informazioni o un comando che guida l'IA nella comprensione del contesto e delle aspettative dell'utente. I prompt sono un elemento fondamentale per le interazioni con ChatGPT poiché determinano la direzione e la natura delle risposte generate dall'IA. Un prompt chiaro e specifico consente a ChatGPT di comprendere meglio l'intento dell'utente, rendendo la risposta più rilevante e utile, in questo senso, i prompt non solo avviano la conversazione, ma ne modellano anche il percorso, influenzando l'accuratezza e la pertinenza delle risposte fornite dall'IA.

La forma e la struttura di un prompt possono variare notevolmente a seconda dell'obiettivo dell'utente, dalla semplice domanda diretta su un fatto specifico alla richiesta di analisi di file con dati strutturati o alla successione di prompt in sequenza per ottenere un risultato più raffinato, scrivere un prompt si configura come un'azione determinante, una sorta di codice scritto in linguaggio naturale per l'estrazione di informazioni in una forma a noi congeniale, il prompt è quindi un vero e proprio codice di linguaggio con delle regole e delle strutture sintattiche attraverso il quale ci è permesso interagire con l'interfaccia "amichevole" (come ChatGPT) di un modello in grado di manipolare enormi archivi di dati utilizzati appunto per il suo addestramento.

I prompt efficaci hanno alcune caratteristiche comuni: sono chiari e diretti, evitano ambiguità che potrebbero portare l'IA fuori strada, inoltre, definiscono adeguatamente il contesto, fornendo al modello con il quale interagiamo, le informazioni di base necessarie per generare una risposta pertinente. Questo fattore è particolarmente importante quando si tratta di argomenti complessi o specialistici, dove l'assenza di dettagli chiave potrebbe portare a risposte imprecise o addirittura completamente inattendibili.

OpenAI nel dicembre 2023 ha fornito delle linee guida ufficiali per la scrittura di prompt per i propri modelli. Ne emergono 5 istruzioni fondamentali che qui di seguito sono analizzate in dettaglio e corredate di esempi.

Cinque parole chiave: Chiarezza - Esempi - Suddivisione - Riflessione - Prove

Chiarezza

Nella scrittura di prompt per ChatGPT, uno dei compiti più critici è evitare l'uso di espressioni gergali e di ambiguità nelle richieste, la chiarezza e la linearità dell'esposizione è particolarmente importante perché, pur essendo un modello linguistico avanzato, ChatGPT

può avere difficoltà nel decodificare termini eccessivamente tecnici o espressioni approssimative.

Il concetto di chiarezza nella scrittura del prompt è possibile riassumerlo in 5 punti:

1. **Attenzione al lessico:** nell'impiego di termini tecnici o gergali specifici di un determinato ambito, risulta fondamentale fornire una definizione chiara. L'uso di un linguaggio gergale con espressioni idiomatiche o termini altamente specializzati dovrebbe essere limitato esclusivamente alle situazioni in cui è strettamente necessario. Allo stesso modo è importante evitare frasi che possano prestarsi a interpretazioni multiple, poiché l'ambiguità nel linguaggio può causare risposte confuse o inesatte da parte di ChatGPT.
2. **Struttura logica e ordinata:** organizzare il prompt in una struttura logica aiuta a mantenere chiarezza e coerenza. Ad esempio, se si stanno chiedendo informazioni su più argomenti, è utile separarli chiaramente nel prompt, in questo caso potrebbe essere utile avvalersi di delimitatori come le virgolette triple `'''` usate anche in Python per identificare una stringa su più righe, oppure una combinazione di caratteri speciali inusuali (`####` oppure `<<< >>>` o `///`) sufficientemente distintivi per essere riconosciuti da ChatGPT come separatori di contenuti e non come semplice punteggiatura.
Un'alternativa per l'uso di caratteri speciali come delimitatori è impiegare i tag XML. Questi sono tag racchiusi in parentesi angolari, con tag di apertura e di chiusura, ad esempio `<tag>` e `</tag>`. Questo metodo è efficace perché gli LLM sono formati su un grande quantità di contenuti web formattati in XML e grazie a questo addestramento sono in grado di interpretarne la struttura e le specifiche di formattazione. Un vantaggio dello standard XML è nella possibilità di creare dei tag autoesplicativi, ad esempio `<personaggio-principale>`
`</personaggio-principale>` o `<testo-esempio>` `</testo-esempio>`, questa possibilità facilita la comprensione, da parte di ChatGPT del legame tra le istruzioni fornite e i tag XML impiegati come delimitatori.
3. **Evitare il sovraccarico di informazioni:** sebbene possa essere tentante includere molti dettagli in un unico prompt, ciò può confondere sia ChatGPT sia l'utente. Se necessario, dividere le richieste complesse in più prompt separati può essere un approccio più efficace. Un prompt efficace è diretto e focalizzato sull'obiettivo, include soltanto le richieste che permettono il raggiungimento dell'obiettivo ed evita i contenuti ridondanti.
4. **Esempi contestualizzati:** offrire esempi pratici o scenari in cui il processo per raggiungimento del nostro obiettivo viene utilizzato, può aiutare a chiarire il significato della nostra richiesta. Questo metodo è particolarmente efficace in ambienti tecnici o specialistici dove i termini possono avere significati molto specifici.
5. **Fornisci dei vincoli e delle regole:** nella formulazione del prompt, può essere utile includere vincoli e linee guida espliciti per assicurarsi che ChatGPT rimanga concentrato sul compito e produca l'output desiderato. Questi vincoli possono essere un particolare formato della risposta, il numero di parole, o qualsiasi altro requisito

rilevante per il risultato che vuoi ottenere. Chiedendo di rielaborare e riassumere un testo potremmo indicare un vincolo di lunghezza caratteri in questo modo:

Analizza il testo che ti fornirò alla fine di questo messaggio, riscrivilo e riassumilo in massimo 500 caratteri. ''' testo da riassumere '''

E' chiaro quindi che per ottenere risposte che siano efficaci è importante assicurarsi che le richieste forniscano tutti i dettagli o il contesto importanti, in caso contrario stiamo lasciando al modello il compito di indovinare le nostre intenzioni con il rischio di avere come risultato delle interpretazioni fantasiose o semplicemente approssimative. Come abbiamo già visto, per i compiti semplici è sufficiente descrivere in dettaglio e chiaramente la nostra richiesta: poche semplici parole che racchiudono interamente il nostro obiettivo. Per compiti più complessi, in cui ci aspettiamo un risultato con un formato preciso e in cui forniamo dei contenuti da interpretare, è utile utilizzare dei delimitatori per indicare le diverse parti dell'input, i delimitatori possono essere dei tag XML o delle virgolette triple. Questa tecnica è molto utile proprio per evitare ambiguità nel prompt e separare la nostra richiesta dal contenuto su cui il modello si troverà a lavorare. Un'altra tecnica che permette di ottenere un risultato preciso, utile alle nostre intenzioni è quella che viene definita "few-shot" e che viene descritta in dettaglio nella sezione relativa di questa guida. Con la tecnica "few-shot" forniamo al modello un set ristretto di esempi del risultato che ci aspettiamo, questi esempi saranno per il modello una guida per affinare la formattazione del risultato sulle nostre necessità e limiteranno le interpretazioni e le possibili anomalie nella risposta.

Riferimento

ChatGPT sfrutta un modello addestrato per compiti generici e al contrario dell'espressione che li definisce come "intelligenza artificiale" non hanno reale comprensione del testo che viene fornito, si tratta di modelli probabilistici e sia di ChatGPT 4o che ChatGPT 4 hanno la tendenza a fornire sempre una risposta alla richiesta dell'utente anche quando il risultato non è perfettamente attendibile. Per alcuni casi è utile fornire un testo di riferimento verificato sul quale chiediamo di operare con delle iterazioni o delle riscritture senza cambiare il contenuto e quello che per noi è il "valore semantico" del testo. In questo scenario l'uso di delimitatori come le virgolette triple, tag nel formato XML o Markdown è una tecnica efficace per focalizzare il modello sul testo da elaborare.

In scenari meno articolati fornire dei riferimenti è comunque una buona strategia, in particolare fornire esempi aiuta a rendere esplicito e comprensibile al modello il risultato che ci aspettiamo di ottenere.

Come già anticipato quando abbiamo definito il concetto di "chiarezza" del prompt, includere nel prompt degli esempi efficaci a definire il contesto in cui il modello si troverà ad operare permette una "centratura" che lo aiuterà a fornirci delle risposte in linea con le nostre aspettative.

Suddivisione

ChatGPT sia nella versione 4o che nella versione 4 possiede una finestra di contesto limitata ad un numero di token preciso, questo significa che può essere necessario far "ricordare" al modello passaggi precedenti di una lunga conversazione. Una finestra di contesto, nei modelli di intelligenza artificiale come GPT-4o o 4 si riferisce alla quantità di informazioni o testo che il modello può considerare in un dato momento durante la generazione di una risposta. È un concetto importante perché rappresenta il limite di memoria del modello, oltre questo limite, il modello non può ricordare o fare riferimento a informazioni precedenti. Questo significa che in una conversazione lunga o nel trattamento di un documento ampio, il modello potrebbe "dimenticare" parti del dialogo o del testo precedente e se un aspetto cruciale di una conversazione o di un documento cade fuori dalla finestra di contesto, la risposta del modello potrebbe non tenere conto di quell'informazione. Quindi solo il testo all'interno di questa finestra può essere utilizzato direttamente per influenzare la generazione del testo e in applicazioni come i chatbot (vd ChatGPT), se la conversazione supera la lunghezza della finestra di contesto, diventa necessario riassumere parti precedenti del dialogo per mantenere la coerenza.

La suddivisione di un prompt in più parti permette quindi di includere nel prompt anche parti della risposta del modello così da mantenere alta la concentrazione del software sui contenuti che vogliamo siano al centro della nostra elaborazione testuale, in applicazioni più avanzate che sfruttano direttamente i modelli GPT tramite le API di OpenAI, le query possono contenere molte delle risposte precedenti del modello e anche testi di riferimento citati in precedenza nella conversazione e compressi tramite tecniche di embeddings. La capacità di "far ricordare" informazioni in una lunga conversazione è una caratteristica fondamentale dei Large Language Models (LLM) come GPT-4. Questa tecnica migliora notevolmente l'efficacia delle interazioni, rendendo il dialogo con il modello più fluido, contestualizzato e utile.

Per illustrare come si possa applicare efficacemente questa tecnica, esaminiamo alcuni esempi pratici.

Immaginiamo un contesto in cui un utente interagisce con GPT-4 per pianificare un viaggio. L'utente inizia chiedendo informazioni su possibili destinazioni. Il modello suggerisce diverse opzioni basate sulle preferenze dell'utente, come le città europee famose per l'arte e la cultura. Durante la conversazione, l'utente esprime un particolare interesse per Roma e Parigi. A questo punto, il modello "ricorda" questa preferenza e, nelle interazioni successive, propone informazioni specifiche su queste due città, come luoghi di interesse, ristoranti e consigli di viaggio.

Un altro potrebbe essere un dialogo continuo su un argomento complesso come la programmazione. Se un utente sta lavorando su un progetto di sviluppo software e chiede consigli su specifici linguaggi di programmazione, il modello può "ricordare" il contesto del progetto, così, quando l'utente chiede successivamente informazioni su un problema di

debug o su una libreria specifica, il modello fornisce risposte che non solo sono pertinenti alla domanda immediata ma che si inseriscono anche nel contesto più ampio del progetto software dell'utente.

Riflessione

A volte otteniamo risultati migliori quando istruiamo esplicitamente il modello a ragionare partendo dai principi di base prima di arrivare a una conclusione. Supponiamo, ad esempio, di voler che un modello valuti la soluzione di un problema matematico da parte di uno studente. Il modo più ovvio per affrontare ciò è semplicemente chiedere al modello se la soluzione dello studente è corretta o meno, è possibile però che il modello stesso venga influenzato dalla risposta dell'utente, è utile quindi preparare ChatGPT con ragionamenti intermedi che portano verso la soluzione e solo successivamente sottoporre la valutazione del problema risolto. Un altro caso concreto sono le operazioni su più file CSV per analisi dati, in questo scenario fornire un prompt con diverse operazioni per ottenere un unico risultato salvato su un nuovo file, potrebbe generare delle incongruenze e delle incomprensioni. E' efficace in questo caso scomporre il problema in più passaggi e guidare il chatbot verso la soluzione che ci aspettiamo lasciando a lui i calcoli complessi e la scrittura del file, ad esempio possiamo far elaborare un risultato sul primo file, poi descrivere che utilizzeremo un secondo file per un secondo elaborato e poi chiedere conferma della comprensione delle operazioni eseguite per poi chiedere la creazione di un unico file finale in cui le due soluzioni sono integrate.

Prove

Un elemento ulteriore per ottimizzare i risultati degli scambi con ChatGPT è costituito dalle attività di iterazione del prompt inviato, è raro infatti che con una singola interazione il sistema riesca a centrare perfettamente le nostre esigenze in particolare sulla replica di stili e forme sintattiche, sperimentare con varie modalità di formulazione del prompt diventa quindi un esercizio in grado di guidare il software verso dei risultati più vicini alle nostre aspettative.

Attraverso un processo iterativo di domande e risposte, l'utente può guidare ChatGPT verso risposte che soddisfino meglio i propri bisogni o rispondano più precisamente alle proprie domande. Se la risposta iniziale di ChatGPT non raggiunge il risultato atteso, è utile iterare e migliorare la risposta ad esempio chiedendo al modello di fornire più informazioni oppure di riformulare completamente la sua risposta o magari di esplorare una prospettiva diversa. Richiedere analisi e informazioni da diverse prospettive su un argomento specifico si rivela particolarmente utile in contesti multidimensionali, come lo sviluppo di un prodotto o la gestione di un progetto, dove la comprensione delle diverse visioni e interessi degli stakeholder è cruciale. L'idea alla base di questa strategia è di utilizzare l'intelligenza

artificiale per considerare un argomento da molteplici angolazioni, integrando le prospettive di vari attori coinvolti.

La sperimentazione e l'uso di diversi approcci nella formulazione dei prompt può mostrare come variazioni anche minime nel linguaggio o nella struttura della domanda possano avere un impatto significativo sulla risposta dell'IA, questo processo di sperimentazione facilita una migliore comprensione di come ChatGPT interpreta i prompt e offre indicazioni su come affinare le richieste per ottenere le risposte desiderate, concettualmente è una forma di "sintonizzazione" tra utente, obiettivi attesi e modello di AI. Replicando, modificando e producendo diversi tipi di interazioni accordiamo il modello al nostro linguaggio e alle nostre aspettative così da poter ottenere delle risposte efficaci, ricche di sfumature e significati.

Agisci come - *strategia bonus* -

Una strategia efficace, anche se non esplicitamente menzionata nelle linee guida di OpenAI, è quella di invitare ChatGPT ad operare come in un gioco di ruolo e di agire nelle vesti di un personaggio chiave all'esecuzione del prompt. Questo approccio, che include l'uso della frase "act as" nel prompt, rende possibile la creazione di dialoghi efficaci, risposte contestualizzate e la simulazione di scenari realistici. Ad esempio, potresti dire a ChatGPT di "agire come un agente di viaggio" per fornire raccomandazioni su destinazioni di vacanza basate sulle preferenze dell'utente, oppure di "agire come un detective" per risolvere un crimine immaginario. Per utilizzare il "trucco Act as", è sufficiente includere la frase "act as" seguita dalla descrizione del ruolo o della personalità che ChatGPT dovrebbe assumere.

Il modello può essere istruito a imitare lo stile di scrittura o il modo di pensare di una persona famosa, ad esempio, "Scrivi come se fossi Stephen King", oppure è possibile chiedere al modello di assumere un ruolo particolare, come un mediatore, un consulente, o un amico, per ottenere una prospettiva o un tipo di risposta che si adatti a quel ruolo.

Questa tecnica è efficace perché consente agli utenti di stabilire un contesto chiaro e specifico per il modello, il quale può poi generare risposte più mirate e appropriate. Tuttavia, è importante ricordare che, nonostante il modello possa imitare uno stile o un ruolo, le sue risposte sono sempre simulate e non riflettono la vera conoscenza o competenza di una vera persona esperta o del ruolo specificato.

Custom instructions:

Le Custom Instructions di ChatGPT sono una funzionalità che consente agli utenti di personalizzare le interazioni con il modello linguistico di OpenAI. Questa caratteristica permette di incorporare nelle risposte di ChatGPT preferenze o specifiche dettate dall'utente, attraverso le "custom instructions", le istruzioni fornite vengono considerate costantemente dal modello in ogni interazione, evitando la necessità di ripetere le stesse indicazioni più volte.

Dal punto di vista operativo, l'uso delle Custom Instructions si traduce in un'interazione più mirata con il modello, in cui le risposte fornite da ChatGPT possono essere più aderenti alle esigenze specifiche dell'utente, poiché il modello tiene conto delle istruzioni personalizzate inserite. Questo aspetto può risultare utile in vari contesti, come l'ambito accademico o professionale, dove le esigenze informative possono variare notevolmente. Le funzionalità offerte dalle Custom Instructions comprendono la possibilità di definire il tono e lo stile delle risposte di ChatGPT, oltre a specificare informazioni o preferenze che il modello deve considerare, è possibile, inoltre, integrare queste istruzioni con i plugin di ChatGPT per migliorare l'interazione in specifici ambiti di utilizzo.

Il pannello di controllo di ChatGPT permette l'accesso alle Custom Instructions, all'interno di questo pannello, gli utenti possono inserire o modificare istruzioni personalizzate. Il processo prevede la selezione della sezione desiderata, l'inserimento delle istruzioni e il salvataggio delle modifiche, consentendo un controllo diretto e flessibile sull'output del modello.

Libreria dei Prompt

In questa sezione sono elencati e catalogati prompt per diversi scopi e vengono dettagliate con degli esempi le linee guida che sono state esplicitate in precedenza. Gli esempi qui elencati sono stati tutti verificati e provati in lingua italiana, per molti di questi esempi è possibile visualizzare la chat con ChatGPT per visualizzare lo scambio completo tra utente e chatbot.

Analizzando i prompt sarà evidente come l'efficacia aumenta quando:

- Iniziamo con un prompt chiaro e preciso.
- Incoraggiamo ChatGPT ad espandere le sue risposte.
- Indichiamo tono e linguaggio utilizzati nella conversazione.
- Monitoriamo l'andamento della conversazione aggiustando il tiro e adeguando le nostre richieste alle risposte per migliorare il risultato finale.

Formazione:

In questa prima sezione sono elencate delle strategie di scrittura di prompt dedicati al supporto per l'apprendimento. Attraverso un uso bilanciato, ChatGPT diventa uno strumento efficace per studenti, insegnanti e per la propria formazione personale. L'uso di ChatGPT per scopi formativi deve tenere in considerazione i limiti del modello, le tecniche per focalizzare le risposte in linea con le nostre aspettative e soprattutto dei prompt precisi in grado di mitigare il rischio di allucinazioni o eccessiva creatività nei testi.

Prompt:

Spiega il concetto di [concetto] in termini semplici e comprensibili per un pubblico non specializzato. La tua spiegazione avrà questa struttura:

- introduzione // spiega in maniera semplice [concetto]
- caratteristiche // descrivi le caratteristiche fondamentali del [concetto]
- applicazioni // riporta degli esempi in cui il [concetto] viene applicato

Esempio:

Utente: Spiega il concetto di fisica quantistica in termini semplici e comprensibili per un pubblico non specializzato. La tua spiegazione avrà questa struttura:

- introduzione // spiega in maniera semplice fisica quantistica*
- caratteristiche // descrivi le caratteristiche fondamentali della fisica quantistica*
- applicazioni // riporta degli esempi in cui la fisica quantistica viene applicato*

ChatGPT: La fisica quantistica è un campo affascinante e complesso della scienza, che studia il comportamento e le interazioni delle particelle subatomiche. Ecco una spiegazione semplificata:

Introduzione: Cos'è la Fisica Quantistica?

La fisica quantistica è la teoria scientifica che si occupa del mondo dei piccolissimi, cioè delle particelle più elementari dell'universo come elettroni, fotoni e quark. A differenza della fisica classica, che descrive il mondo che vediamo intorno a noi, la fisica quantistica si occupa di fenomeni che sfidano la nostra intuizione e esperienza quotidiana. Per esempio, in questo strano mondo, una particella può trovarsi in più posti contemporaneamente o essere intrecciata con un'altra particella a distanze enormi.

Caratteristiche: Le Fondamenta della Fisica Quantistica

Dualità onda-particella: Le particelle subatomiche possono comportarsi sia come particelle (con una posizione definita nello spazio) che come onde (distribuite su una regione dello spazio).

Principio di Indeterminazione di Heisenberg: Non è possibile conoscere contemporaneamente con precisione sia la posizione sia la velocità di una particella. Più sappiamo di uno, meno possiamo sapere dell'altro.

Entanglement Quantistico: Due particelle possono essere correlate (o "intrecciate") in modo tale che il cambiamento dello stato di una influenzi immediatamente l'altra, indipendentemente dalla distanza che le separa.

Superposizione: Una particella può esistere in molti stati diversi contemporaneamente, fino a quando non viene misurata.

Applicazioni: Dove la Fisica Quantistica Ci Influenza

Computer Quantistici: Usano le proprietà della superposizione e dell'intreccio per eseguire calcoli a velocità incredibilmente superiori rispetto ai computer tradizionali.

Telecomunicazioni e Criptografia Quantistica: Migliorano la sicurezza delle comunicazioni sfruttando le proprietà uniche dell'entanglement.

Imaging Medico: Tecniche come la risonanza magnetica nucleare (RMN) si basano sui principi quantistici per creare immagini dettagliate dell'interno del corpo umano.

Materiali e Tecnologie Innovative: La comprensione delle proprietà quantistiche ha portato allo sviluppo di nuovi materiali e dispositivi, come i LED e i semiconduttori.

In sintesi, la fisica quantistica, pur essendo controintuitiva e astratta, è fondamentale per comprendere la natura a livello più profondo e ha applicazioni pratiche che influenzano molte tecnologie moderne.

Prompt

Elabora un piano didattico per una lezione su [contenuto] dedicata a studenti [destinatari]

Il piano che definirai sarà organizzato in questi punti:

Obiettivi: Quali conoscenze o competenze si prevede che gli studenti acquisiscano entro la fine della lezione?

Introduzione: Una panoramica breve o un elemento accattivante per coinvolgere gli studenti.

Procedura: Attività passo dopo passo che verranno svolte per raggiungere gli obiettivi della lezione.

Valutazione: Metodi per verificare se gli obiettivi sono stati raggiunti, come quiz, discussioni o compiti a casa.

Conclusione: Un riassunto della lezione, che solitamente sintetizza i punti chiave o rivede gli obiettivi principali.

Esempio:

Utente: Elabora un piano didattico per una lezione su Il ciclo dell'acqua dedicata a studenti di terza elementare.

Il piano che definirai sarà organizzato in questi punti:

***Obiettivi:** Quali conoscenze o competenze si prevede che gli studenti acquisiscano entro la fine della lezione?*

***Introduzione:** Una panoramica breve o un elemento accattivante per coinvolgere gli studenti.*

***Procedura:** Attività passo dopo passo che verranno svolte per raggiungere gli obiettivi della lezione.*

Valutazione: Metodi per verificare se gli obiettivi sono stati raggiunti, come quiz, discussioni o compiti a casa.

Conclusione: Un riassunto della lezione, che solitamente sintetizza i punti chiave o rivede gli obiettivi principali.

ChatGPT:

Piano Didattico per la Lezione sul Ciclo dell'Acqua (Terza Elementare)

Obiettivi

Comprendere i concetti base del ciclo dell'acqua, inclusi evaporazione, condensazione, precipitazione e raccolta.

Riconoscere l'importanza del ciclo dell'acqua per l'ambiente.

Sviluppare abilità di osservazione e ragionamento attraverso esperimenti e discussioni.

Introduzione

Inizio con un breve video animato che illustra il ciclo dell'acqua in modo semplice e divertente.

Domande interattive: "Da dove viene l'acqua della pioggia?" e "Cosa succede all'acqua dopo che piove?"

Procedura

Discussione Guidata: Spiegazione dei concetti chiave con l'aiuto di diagrammi colorati.

Esperimento "Mini Ciclo dell'Acqua":

Materiali: una ciotola grande, una tazza piccola, pellicola trasparente, acqua.

I bambini riempiono la ciotola con acqua, pongono la tazza al centro, coprono con la pellicola.

Osservano come l'acqua si evapora e si condensa, simulando il ciclo dell'acqua.

Attività Creativa: Disegnare il ciclo dell'acqua e colorare le diverse fasi.

Storia Interattiva: "Il Viaggio di una Goccia d'Acqua", dove i bambini seguono le avventure di una goccia d'acqua attraverso il ciclo.

Valutazione

Quiz Veloce: Domande a risposta multipla per verificare la comprensione dei concetti base.

Discussione di Gruppo: Riflessioni guidate sull'esperimento e sulle attività creative.

Compito a Casa: Creare un breve racconto o disegno che descriva il ciclo dell'acqua.

Conclusione

Riassunto dei punti chiave della lezione.

Domande di riflessione: "Perché il ciclo dell'acqua è importante per noi e per il pianeta?"

Incoraggiamento a osservare il ciclo dell'acqua nella vita quotidiana, come durante la pioggia o la formazione di rugiada.

Questo piano didattico mira a rendere il concetto del ciclo dell'acqua accessibile e coinvolgente per gli studenti di terza elementare, combinando teoria, esperimenti pratici e creatività.

Prompt semplificati:

Agisci come un docente di [disciplina], suggeriscimi tre domande a risposta multipla, con tre opzioni ciascuna, di cui una sola corretta, su [argomento].

Elenca le similitudini e le differenze tra [concetto A] e [concetto B]. Rispondi utilizzando un tono neutrale, senza giudizi qualitativi e riportando soltanto informazioni descrittive.

Forniscimi degli esempi concreti che possano spiegare in maniera semplice il concetto di [concetto]

Descrivi passo passo quali sono i passaggi per [contenuto].

Utilizza un linguaggio semplice adatto ad un pubblico di [pubblico di riferimento]

Esempio: <https://chat.openai.com/share/21bace04-a6de-453c-809d-a08c1dd5c6b2>

Descrivi l'impatto che [fenomeno] ha avuto nella [disciplina]

Esempio: descrivi l'impatto che la seconda rivoluzione industriale ha avuto in letteratura

Prompt iterativo

I prompt iterativi sono utili quando si desidera esplorare un argomento in modo più approfondito. Inizi con una domanda di base e poi continui a costruire su quella con domande di follow-up.

Prompt 1:

Sei un docente di [disciplina] e devi scrivere una dispensa per una lezione approfondita sul tema del [contenuto]. Scrivi l'abstract della dispensa elencando i concetti chiave che contraddistinguono [contenuto] e spiega in maniera semplice [contenuto].

Il testo che scriverai sarà di minimo 500 parole.

Prompt 2:

Agisci come un docente di [disciplina] e scrivi una dispensa approfondita sul [contenuto] a partire dal testo di riferimento da me fornito in allegato utilizzando uno stile formale e neutrale in un testo unico, fluido e senza troppe interruzioni di circa 1500 parole.

Il testo di riferimento è allegato in formato TXT

Prompt 3:

Come docente di storia scrivi una dispensa di approfondimento con uno stile formale e neutrale, dal linguaggio semplice fluido e senza troppe interruzioni sul tema [contenuto] Attingi al testo di riferimento che trovi alla fine di questo messaggio per descrivere il [contenuto] e integra con le informazioni presenti nella tua base di conoscenza.

Il testo di riferimento è:

''' [risposta del prompt1 + risposta del prompt2] '''

Esempio: <https://chat.openai.com/share/1e9e1fae-34f6-4661-b237-ec1c2b2e2e1d>

Scrittura di codice:

In questa sezione i prompt saranno descritti con casi di studio concreti per migliorare la comprensione delle possibilità e per fornire delle risorse già pronte da poter essere utilizzate in casi analoghi.

Scrivere abstract di molti testi

In questo scenario abbiamo una cartella che contiene diversi file di testo e abbiamo bisogno di avere un abstract per ciascuno di loro. Una delle soluzioni è sfruttare uno dei modelli GPT tramite le API di OpenAI, per far questo scegliamo di scrivere uno script Python in grado di leggere in cartella, inviare il file di testo al modello e restituire un abstract del testo salvato in un file CSV con due colonne, una con il nome file originale e nella seconda l'abstract da 500 caratteri.

Prompt:

Ho bisogno di uno script python in grado di elaborare un file di testo utilizzando il modello gpt-3.5-turbo-0125 con l'endpoint v1/chat/completions e tramite le API di OpenAI. Scrivi lo script python in grado di recuperare dei file TXT da una cartella chiamata "Testi", per ciascun file estrarre un abstract da 500 caratteri del contenuto e scrivere un file CSV con due colonne, una con il nome del file e una con l'abstract elaborato.

Risposta:

Per creare uno script Python che elabora file di testo utilizzando GPT-3.5-Turbo e le API di OpenAI, devi seguire alcuni passaggi. Lo script dovrebbe fare quanto segue:

Leggere ogni file TXT da una cartella chiamata "Testi".

Per ogni file, usare GPT-3.5-Turbo per generare un abstract di 500 caratteri del contenuto. Scrivere i risultati in un file CSV con due colonne: una per il nome del file e l'altra per l'abstract.

Prima di procedere, assicurati di avere un account OpenAI e le relative chiavi API per accedere al servizio GPT-3.5-Turbo.

Ecco uno script di esempio che realizza quanto richiesto:

```
import os
import openai
import csv

# Imposta la chiave API di OpenAI
openai.api_key = 'la_tua_chiave_api'

# Percorso della cartella che contiene i file TXT
folder_path = 'Testi'

# Percorso del file CSV di output
csv_file = 'output.csv'

# Funzione per generare l'abstract utilizzando GPT-3.5-Turbo
def generate_abstract(text):
    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo-0125",
        messages=[{"role": "system", "content": "Fornisci un
abstract di 500 caratteri del seguente testo:"},
        {"role": "user", "content": text}]
    )
    return response['choices'][0]['message']['content']

# Elabora i file nella cartella
with open(csv_file, mode='w', newline='', encoding='utf-8') as
file:
    writer = csv.writer(file)
    writer.writerow(['File Name', 'Abstract'])

    for filename in os.listdir(folder_path):
        if filename.endswith('.txt'):
            file_path = os.path.join(folder_path, filename)
            with open(file_path, 'r', encoding='utf-8') as f:
                content = f.read()
                abstract = generate_abstract(content[:1000]) #
Usa solo i primi 1000 caratteri per l'abstract
```

```
        writer.writerow([filename, abstract[:500]])    #  
Scrive i primi 500 caratteri dell'abstract  
  
print('Elaborazione completata.')
```

Interrogare WikiData con Query SPARQL

SPARQL è un linguaggio di query utilizzato per interrogare e manipolare dati rappresentati nel formato RDF (Resource Description Framework). RDF è un modello dati utilizzato per rappresentare informazioni basate su triple, che consistono in soggetti, predicati e oggetti, ed è comunemente utilizzato per rappresentare dati strutturati e metadati semantici.

WikiData così come il Catalogo centrale Beni Culturali utilizzano le query SPARQL per permettere interrogazioni sulle basi dati.

Prompt:

Scrivi una query SPARQL per elencare tutti gli elementi presenti in WikiData associati al Rijksmuseum

Risposta:

Per elencare tutti gli elementi presenti in Wikidata che sono associati al Rijksmuseum, la seguente query SPARQL può essere utilizzata:

```
SELECT ?item ?itemLabel  
WHERE  
{  
    ?item wdt:P195 wd:Q190804.  
    SERVICE wikibase:label { bd:serviceParam wikibase:language  
"[AUTO_LANGUAGE],en". }  
}
```

Questa query seleziona tutti gli elementi (?item) e le loro etichette (?itemLabel) che hanno una proprietà di "collezione" (wdt:P195) associata all'ID di Wikidata del Rijksmuseum (wd:Q190804). Il servizio wikibase:label è utilizzato per ottenere le etichette degli elementi nella lingua preferita dell'utente o in inglese.

[WikiData link](#)

Scrittura di più file da un solo prompt

ChatGPT, soprattutto nella versione 4 permette delle interazioni complesse a partire anche da prompt non perfettamente esplicativi.

In questo caso abbiamo chiesto di scrivere uno script PHP in grado di estrarre i metadati di un'immagine caricata. Nella richiesta non abbiamo citato la necessità di scrivere anche il codice HTML per il form di presentazione, il modello però ha compensato e ha elencato

passaggi e soluzioni.

Prompt:

Ho bisogno di un codice in grado di estrarre i metadati da immagini salvate in JPG, in particolare i dati EXIF dell'immagine e tutti gli altri attributi. Scrivi uno script PHP che permetta l'upload di una foto, la elabori e ne estragga i metadati. Dopo l'elaborazione mostra in una tabella i metadati estratti.

Risposta:

Per realizzare uno script PHP che permetta l'upload di una foto, elabori e ne estragga i metadati, possiamo suddividere il compito in diverse parti. Prima di tutto, è necessario creare una semplice interfaccia HTML per l'upload della foto. Poi, lo script PHP gestirà l'upload del file e l'estrazione dei metadati EXIF. Infine, i metadati estratti verranno mostrati in una tabella HTML.

Codice HTML generato:

```
<!DOCTYPE html>
<html>
<head>
    <title>Upload e Estrazione Metadati di un'Immagine</title>
</head>
<body>
    <form action="upload.php" method="post"
enctype="multipart/form-data">
        Seleziona immagine:
        <input type="file" name="fileToUpload" id="fileToUpload">
        <input type="submit" value="Upload Image" name="submit">
    </form>
</body>
</html>
```

Codice PHP generato:

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $target_dir = "uploads/";
    $target_file = $target_dir .
basename($_FILES["fileToUpload"]["name"]);
    $uploadOk = 1;
    $imageFileType =
strtolower(pathinfo($target_file,PATHINFO_EXTENSION));

    // Controlla se il file è un'immagine
```

```

        if(isset($_POST["submit"])) {
            $check =
getimagesize($_FILES["fileToUpload"]["tmp_name"]);
            if($check !== false) {
                echo "File is an image - " . $check["mime"] . ".";
                $uploadOk = 1;
            } else {
                echo "File is not an image.";
                $uploadOk = 0;
            }
        }

        // Carica il file
        if ($uploadOk == 0) {
            echo "Sorry, your file was not uploaded.";
        } else {
            if
(move_uploaded_file($_FILES["fileToUpload"]["tmp_name"],
$target_file)) {
                echo "The file ". htmlspecialchars( basename(
$_FILES["fileToUpload"]["name"])) . " has been uploaded.";

                // Estrazione dei metadati EXIF
                $exif = exif_read_data($target_file);
                if ($exif) {
                    echo "<h2>Metadati Estratti:</h2>";
                    echo "<table border='1'>";
                    foreach ($exif as $key => $value) {
                        echo "<tr><td>$key</td><td>$value</td></tr>";
                    }
                    echo "</table>";
                } else {
                    echo "No EXIF data found for this image.";
                }
            } else {
                echo "Sorry, there was an error uploading your file.";
            }
        }
    }
}
?>

```

Revisioni: al codice PHP è stato chiesto di aggiungere una REGEX per evitare l'upload di file che non sono immagini.

Il file risultante, nonostante funzionasse, rilasciava un warning derivato alla presenza di array di dati nei dati EXIF dell'immagine, quindi è stato richiesto di elaborare nuovamente il codice per risolvere l'errore:

Prompt:

ho questo errore:

```
Warning: Array to string conversion in  
/home/mhd-01/www.xxx/htdocs/test/upload.php on line 39
```

Risposta:

L'errore "Warning: Array to string conversion" si verifica quando si tenta di stampare o utilizzare un array come una stringa. Nel contesto del tuo script PHP, questo potrebbe accadere quando si tenta di visualizzare i metadati EXIF, poiché alcuni dei valori EXIF possono essere array anziché stringhe singole.

Per risolvere questo problema, dovresti controllare se ciascun valore è un array e, in caso affermativo, gestirlo in modo appropriato. Ad esempio, puoi convertire l'array in una stringa leggibile con la funzione `print_r()` o `implode()`. Modifica la parte dello script che stampa i metadati EXIF come segue:

[File corretto](#)

[Pacchetto di file HTML+PHP](#)

Esercizio:

Provate a fornire il file HTML e chiedete di integrare il framework Bootstrap al codice per migliorare l'aspetto dell'interfaccia e renderlo compatibile per mobile.

Realizzare una pagina HTML da un documento Word

In questo scenario viene richiesto di scrivere il codice di una pagina HTML che descriva le informazioni chiave contenute in un file .docx, viene chiesto anche di applicare fogli di stile ed eventuali Javascript utilizzando il framework Bootstrap.

Il file docx viene allegato e viene utilizzato il modello ChatGPT 4.

Prompt:

Ti fornirò un file docx con del testo e delle informazioni di contesto.

Riporta le informazioni del testo in una pagina HTML strutturata come una dispensa didattica.

Utilizza il framework Bootstrap per definire i CSS ed eventuali JS.

Il file docx da elaborare è in allegato

Risposta:

Basandomi sul contenuto del file `Demo-doc.docx` che hai fornito, ho creato una struttura per una pagina HTML che servirà come dispensa didattica. Questa pagina utilizzerà il framework Bootstrap per il CSS e, se necessario, per il JavaScript. Il contenuto del documento si concentra sull'importanza del "prompt" in un sistema di intelligenza artificiale come ChatGPT, evidenziando come i prompt influenzano le risposte generate e offrendo alcuni consigli su come formulare prompt efficaci.

Struttura della Pagina HTML

```
<!DOCTYPE html>
<html lang="it">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Dispensa su Prompt in ChatGPT - Paolo De Gasperis,
2024</title>
  <link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootst
rap.min.css" rel="stylesheet">
</head>
<body>
  <div class="container mt-5">
    <h1 class="mb-4">L'Importanza del Prompt in ChatGPT</h1>
    <p class="lead">Autore: Paolo De Gasperis</p>
    <p class="lead">Anno: 2024</p>

    <section class="mt-4">
      <h2>Introduzione al Prompt</h2>
      <p>Il prompt è un input testuale fornito dall'utente
che funge da punto di partenza per la generazione delle risposte
da parte del sistema di intelligenza artificiale (IA) di ChatGPT.
Questo input può assumere diverse forme: una domanda,
un'affermazione, un'istruzione o una combinazione di questi
elementi.</p>
    </section>

    <section class="mt-4">
      <h2>Natura e Funzione dei Prompt</h2>
      <p>In termini più ampi, un prompt è una sorta di
innesco per la conversazione, una richiesta di informazioni o un
```

comando che guida l'IA nella comprensione del contesto e delle aspettative dell'utente. I prompt sono un elemento fondamentale per le interazioni con ChatGPT, poiché determinano la direzione e la natura delle risposte generate dall'IA.</p>

</section>

<section class="mt-4">

<h2>La Formulazione dei Prompt</h2>

<p>Scrivere un prompt si configura come un'azione determinante, una sorta di codice scritto in linguaggio naturale per l'estrazione di informazioni in una forma a noi congeniale. I prompt efficaci hanno alcune caratteristiche comuni: sono chiari e diretti, evitano ambiguità che potrebbero portare l'IA fuori strada e definiscono adeguatamente il contesto.</p>

</section>

<footer class="mt-5">

<p>© 2024 Paolo De Gasperis. Tutti i diritti riservati.</p>

</footer>

</div>

<script

src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script

src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.0.5/dist/umd/popper.min.js"></script>

<script

src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>

</html>

Note sulla Pagina HTML:

- Ho utilizzato Bootstrap 4.5.2, ma puoi aggiornarlo alla versione più recente se lo desideri.
- La struttura è divisa in sezioni per organizzare logicamente il contenuto.
- Sono inclusi i collegamenti a jQuery e Popper.js, che sono richiesti da alcune funzionalità di Bootstrap.
- Il contenuto è racchiuso in un container di Bootstrap per una migliore formattazione e layout responsivo.

Puoi utilizzare questo codice HTML come base per la tua dispensa didattica e modificarlo secondo le tue esigenze specifiche.

[La pagina HTML](#)

Il metodo “Act as... / Behave as...” per la scrittura di codice

In questa sezione sono elencati dei metodi generalizzati che permettono di utilizzare la personificazione di un ruolo per la scrittura di codice o per la soluzione di errori e bug.

Esempio 1

Agisci come: Programmatore PHP

Software stack: CakePHP, MySQL

Funzionalità: Calcolo IVA

Campi obbligatori: Importo, Aliquota IVA, Anno

Task: Scrivi il codice per un servizio in grado di calcolare l'aliquota IVA di un prodotto

<https://chat.openai.com/share/0d18c768-7776-47f4-9fa5-98a6ee043c30>

Esempio generalizzato:

Agisci come: [programmatore di linguaggio specifico]

Software stack: [la serie di tecnologie necessarie per lo sviluppo del prodotto]

Funzionalità: [cosa deve fare il codice]

Campi obbligatori: [i parametri che sono sempre presenti]

Task: [indica qui cosa deve scrivere ChatGPT e la funzione del codice]

Esempio generalizzato ridotto:

Agisci come un programmatore [linguaggio] e scrivi il codice per [descrizione dettagliata della funzione del codice]

Correzione di errori

Il codice che segue mostra un errore. Individua l'errore e scrivi la versione corretta del codice fornito.

Codice: [il codice]

Errore: [l'errore che viene mostrato]

Esempio:

Il codice che segue mostra un errore. Individua l'errore e scrivi la versione corretta del codice fornito.

Codice: <?php


```
function moltiplica($num1, $num2) {
    return $num1 * $num2;
// Esempio di utilizzo della funzione
$risultato = moltiplica(5, 10);
echo "Il risultato della moltiplicazione è: " . $risultato; //
Stampa 50
?>
Errore: Parse error:
Unclosed '{' on line 2 in
/home/mhd-01/www.xxx.xx/htdocs/guida-prompt/molt.php on line 9
```

Risposta:

L'errore nel codice PHP fornito è dovuto alla mancanza della chiusura delle parentesi graffe `}` nella definizione della funzione `moltiplica`. La parentesi graffa di apertura è presente, ma manca quella di chiusura.

Ecco il codice corretto:

```
<?php
function moltiplica($num1, $num2) {
    return $num1 * $num2;
} // Chiusura mancante della funzione aggiunta qui

// Esempio di utilizzo della funzione
$risultato = moltiplica(5, 10);
echo "Il risultato della moltiplicazione è: " . $risultato; //
Stampa 50
?>
```

Con questa modifica, il codice ora include la parentesi graffa di chiusura per la funzione `moltiplica`, risolvendo l'errore di parsing.

Analisi dei dati

ChatGPT 4 integra funzioni di analisi dati e dal 6 novembre 2023 OpenAI ha rilasciato delle funzioni specializzate di ChatGPT, permette inoltre agli utenti di utilizzare il modello per produrre delle proprie versioni personalizzate di ChatGPT.

Tra le funzioni specializzate è presente Data Analyst che permette di utilizzare ChatGPT 4 come strumento di analisi e interpretazione dati.

Di seguito proveremo una serie di prompt utili ad:

- pulire dati
- costruire dataset
- recuperare informazioni da pagine web (web scraping)
- produrre visualizzazione dati

- documentare dataset
- estrarre informazioni incrociando dataset

Pulizia dei dati

Prompt:

Scrivi un codice Python per la pulizia di un dataset. Includi nel tuo esempio la gestione dei valori assenti, l'eliminazione dei duplicati e la normalizzazione dei formati.

Usa un dataset con le colonne 'Nome,' 'Età,' 'Professione' ed 'Email'.

Prompt:

Scrivi il codice Python per pulire un dataset contenente dati anagrafici. Il dataset include 'Nome', 'Data di Nascita', 'Luogo di Nascita', e 'Numero di Telefono'. Il focus dovrebbe essere sulla correzione degli errori di formattazione, sulla standardizzazione delle date, e sulla validazione dei numeri di telefono.

Prompt:

Scrivi il codice Python per pulire un dataset di vendite. Il dataset include colonne come 'ID Prodotto', 'Prezzo Unitario', 'Quantità Venduta', e 'Data Vendita'. Il codice dovrebbe affrontare problemi comuni come valori mancanti, formati di data incoerenti, e anomalie nei prezzi o nelle quantità.

Prompt generalizzato:

Scrivi uno script [{opzionale indicare il linguaggio}] per pulire un dataset [tipologia di dati]. Il dataset include colonne come [colonne principali del dataset]. Il codice dovrebbe affrontare problemi comuni come valori mancanti, formati incoerenti, e anomalie nei [casi specifici]

Costruire Dataset

In questo scenario abbiamo una tabella di un database MySQL denominata opere su cui vogliamo fare delle operazioni CRUD attraverso il software che stiamo scrivendo.

Per procedere con i test abbiamo bisogno di popolare la tabella con dati di esempio.

Chiediamo a ChatGPT di scrivere una query MySQL per popolare la tabella con dati fittizi.

Prompt generalizzato

Sto scrivendo un software e ho bisogno di dati di esempio nella tabella del database

Funzionalità del software: [specificare la funzionalità]

Dati da generare:[campi della tabella]

Formato dei dati: [formato del database, della query e quantità di dati]

Prompt di esempio

Sto scrivendo un software e ho bisogno di dati di esempio nella tabella del database

Funzionalità del software: catalogare opere d'arte

Dati da generare: artista, titolo opera, anno creazione

Formato dei dati: query MySQL per inserire 10 record di esempio. La tabella di chiama "opere"

Lo stesso metodo può essere generalizzato per compilare con dati fittizi dataset generici.

Prompt generalizzato

Agisci come un generatore di dati fittizi. Ho bisogno di un insieme di dati che abbia [x] righe e [y] colonne: [inserisci i nomi delle colonne].

Web Scraping

Il web scraping è una pratica che coinvolge l'automazione del recupero di dati da siti web. Il processo utilizza bot o script per estrarre informazioni, quali testo, immagini, e dati strutturali, dalle pagine web ed è comunemente impiegato in vari campi come l'analisi dei dati, il monitoraggio dei prezzi, la ricerca di mercato, l'aggregazione di contenuti e la costruzione di dataset.

Va fatto notare come alcuni siti web abbiano termini di servizio che limitano o vietano esplicitamente il web scraping e che la violazione di questi termini potrebbe avere conseguenze legali. Inoltre, intense attività di scraping dalle pagine di un sito potrebbero essere identificate da eventuali sistemi anti-intrusione come veri e propri attacchi, l'attività infatti se ripetuta in maniera intensiva può causare un sovraccarico del server che ospita il sito tanto da essere interpretata come un attacco DDoS.

Prompt:

Scrivi un codice Python in grado di recuperare i dati dalla pagina

'https://it.wikipedia.org/wiki/OpenAI' utilizzando le librerie BeautifulSoup e requests. Estrai e visualizza il titolo della pagina, gli URL e il contenuto testuale delle intestazioni principali (ad esempio, h1, h2) sulla pagina.

Salva i dati in un file CSV.

Prompt generalizzato:

Scrivi un codice Python in grado di recuperare i dati dalla pagina '[URL]' utilizzando le librerie BeautifulSoup e requests. Estrai e visualizza il titolo della pagina, gli URL e il contenuto testuale delle intestazioni principali (ad esempio, h1, h2) sulla pagina.

Salva i dati in un file CSV.

Un metodo efficace per recuperare contenuti dal web è sfruttare le interfacce API fornite da alcuni siti web oppure sfruttare il feed RSS. Qui di seguito abbiamo due prompt per lo sviluppo di codice Python in grado di interrogare le API della Staten Museum Kunst e un secondo prompt in grado di recuperare dati attraverso il feed RSS

API scrape data

Prompt:

Scrivi un script python in grado di recuperare le informazioni dall'endpoint API della Staten Museum Kunst disponibile a questo indirizzo:

<https://api.smk.dk/api/v1/art/search/?keys=&offset=0&rows=10&lang=en>

Salva le informazioni rispettando la struttura del file JSON mostrato in un file CSV denominato SMK.csv

Prompt generalizzato:

Scrivi un script python in grado di recuperare le informazioni dall'endpoint API a questo indirizzo:[URL]

Salva le informazioni rispettando la struttura del file JSON mostrato in un file CSV

Feed RSS scrape data

Prompt:

Scrivi uno script python per interpretare il file XML del feed RSS di Ansa.it

Recupera i dati dal feed e salvali in un file CSV denominato ansa.csv

Il feed RSS è questo:

https://www.ansa.it/canale_tecnologia/notizie/tecnologia_rss.xml

Prompt generalizzato:

Scrivi uno script python per interpretare il file XML del feed RSS di [nome del sito]

Recupera i dati dal feed e salvali in un file CSV denominato [nome del CSV]

Il feed RSS è questo:

[URL del feed RSS]

E' possibile interagire ulteriormente su questo feed chiedendo di realizzare uno script in PHP in grado di salvare i dati in una pagina HTML.

Prompt

Scrivi uno script in PHP in grado di recuperare i dati dal feed RSS di

https://www.ansa.it/canale_tecnologia/notizie/tecnologia_rss.xml e salvarli in una pagina

HTML. Aggiorna lo script ogni 5 ore.

Risposta

Per creare uno script PHP che recupera i dati da un feed RSS e li salva in una pagina

HTML, seguito da un'automatizzazione per eseguire lo script ogni 5 ore, puoi seguire questi passi:

Passo 1: Script PHP per il Parsing del Feed RSS

Prima di tutto, crea uno script PHP che legge il feed RSS e genera un file HTML. Ecco un esempio di tale script:

```
<?php

function updateRSSFeed() {
    $rssUrl =
"https://www.ansa.it/canale_tecnologia/notizie/tecnologia_rss.xml"
;
    $xml = simplexml_load_file($rssUrl);

    $htmlContent = '<html><head><title>Feed RSS ANSA
Tecnologia</title></head><body>';
    $htmlContent .= '<h1>Articoli ANSA Tecnologia</h1>';

    foreach ($xml->channel->item as $item) {
        $title = $item->title;
        $link = $item->link;
        $description = $item->description;

        $htmlContent .= "<h2>$title</h2>";
        $htmlContent .= "<p>$description</p>";
        $htmlContent .= "<a href='$link'>Leggi di
più</a><br><br>";
    }

    $htmlContent .= '</body></html>';
    file_put_contents('ansa_feed.html', $htmlContent);
}

updateRSSFeed();

?>
```

Questo script legge il feed RSS di ANSA e scrive il contenuto in `ansa_feed.html`.

Passo 2: Automatizzazione con Cron Job

Per eseguire automaticamente lo script ogni 5 ore, puoi impostare un Cron Job su un server Linux. Se non hai accesso a un server Linux, potresti dover cercare alternative specifiche del sistema operativo che stai utilizzando.

Ecco come impostare un Cron Job su un server Linux:
Apri il terminale e digita `crontab -e` per modificare il crontab.
Aggiungi la seguente riga per eseguire lo script ogni 5 ore:

```
0 */5 * * * /usr/bin/php /percorso/dello/script/rss-to-html.php
```

Sostituisci `/percorso/dello/script/rss-to-html.php` con il percorso effettivo del tuo script PHP.
Salva e chiudi l'editor del crontab.
Questo Cron Job eseguirà lo script `rss-to-html.php` ogni 5 ore. Il percorso `/usr/bin/php` è quello comune per l'eseguibile PHP, ma può variare a seconda della configurazione del server. Verifica il percorso corretto per PHP sul tuo sistema se necessario.

Visualizzare dati

ChatGPT è in grado di produrre grafici in grado di interpretare visivamente i dati contenuti in un dataset. In questo esempio è stato utilizzato il dataset degli artisti del MoMa rilasciato dal museo in CC0 <https://github.com/MuseumofModernArt/collection>
Il set di dati che è stato utilizzato è `Artist.csv`

Prompt

Ti ho allegato un dataset che contiene informazioni su artisti del MoMa.
Come esperto in visualizzazioni dati, fornisci dei suggerimenti su quali dati possa visualizzare in una dashboard e che tipo di grafici possa usare.

Prima di scrivere la risposta ChatGPT analizza il dataset e ne identifica le colonne.

Risposta:

Basandomi su queste informazioni, ecco alcuni suggerimenti per visualizzazioni che potresti creare per una dashboard:

- Distribuzione degli Artisti per Nazionalità: Un grafico a barre o una mappa geografica che mostra il numero di artisti per ogni nazionalità.
- Analisi di Genere: Un grafico a torta o a barre che rappresenta la distribuzione di genere degli artisti.
- Linea Temporale degli Artisti: Un grafico a linee o a barre che mostra il numero di artisti nati in ciascun decennio o anno. Questo può aiutare a visualizzare le tendenze storiche.
- Distribuzione dell'età degli Artisti al Momento della Morte: Se hai dati sufficienti sugli anni di morte, potresti creare un istogramma che mostri a che età gli artisti sono generalmente deceduti.

- Artisti Ancora in Vita vs Deceduti: Un grafico a torta o a barre che confronta il numero di artisti ancora in vita con quelli deceduti.
- Collegamenti a Wikidata e ULAN: Una visualizzazione tabellare o una mappa di rete che mostra i collegamenti tra gli artisti e le loro identità in Wikidata e ULAN, se i dati sono sufficientemente completi.

Posso anche effettuare alcune analisi preliminari per darti un'idea migliore di come potresti procedere.

Ad ogni richiesta di grafico viene sviluppato un codice python come questo:

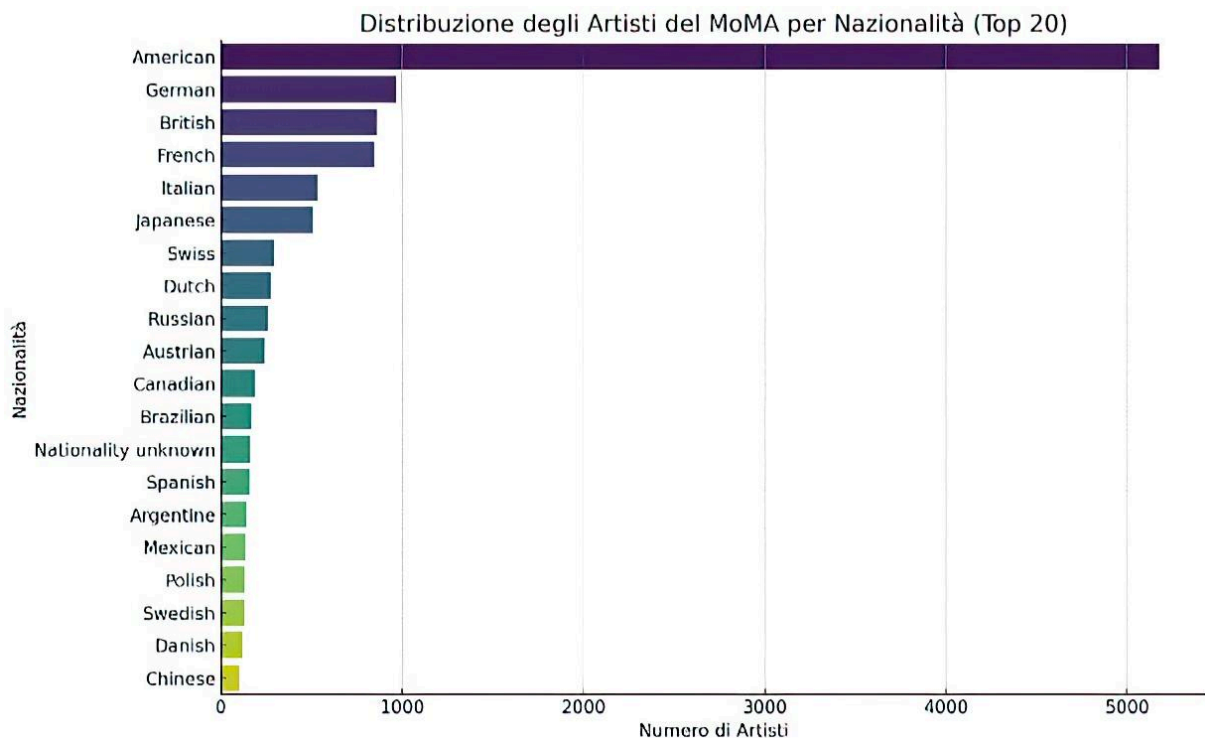
```
import matplotlib.pyplot as plt
import seaborn as sns

# Calcolo della distribuzione degli artisti per nazionalità
nationality_counts = moma_artists['Nationality'].value_counts()

# Filtro per mostrare solo le prime 20 nazionalità
top_nationalities = nationality_counts.head(20)

# Creazione del grafico a barre
plt.figure(figsize=(12, 8))
sns.barplot(x=top_nationalities.values, y=top_nationalities.index,
palette="viridis")
plt.xlabel('Numero di Artisti')
plt.ylabel('Nazionalità')
plt.title('Distribuzione degli Artisti del MoMA per Nazionalità
(Top 20)')
plt.show()
```

e la relativa immagine



Prompt generalizzato:

Ti ho allegato un dataset che contiene informazioni su [descrizione del dataset].

Come esperto in visualizzazioni dati, fornisci dei suggerimenti su quali dati possa visualizzare in una dashboard e che tipo di grafici possa usare.

Documentare codice e dati

La capacità di ChatGPT di analizzare dei dati presenti in tabelle, in dataset o in documenti tecnici permette di utilizzare il modello per redigere della documentazione relativa al set di dati a disposizione. Per documentazione si intende la descrizione dettagliata delle funzioni dello script e l'analisi della sintassi, una possibilità che non solo aiuta a comprendere meglio il codice, ma fornisce anche una base per migliorarlo, ottimizzarlo o correggerne eventuali errori. Un risultato analogo accade con i grandi set di dati da cui possono essere estratte informazioni di contesto in grado di fornire un'interpretazione globale del documento oppure una descrizione esauriente delle sue caratteristiche: dalla natura delle varie colonne a come questi dati possono essere sfruttati per analisi specifiche. Questo non solo semplifica la comprensione del dataset, ma apre anche la strada a un suo utilizzo più efficiente, fornendo indicazioni su possibili interrogazioni SQL o analisi tramite linguaggi come R o Python.

Prompt generalizzato: Analizza questo dataset e scrivi il file README come documentazione tecnica

(il dataset fornito è il CSV utilizzato in precedenza)

[File README del dataset Artists_MoMa.csv](#)

Prompt generalizzato:

Compila un file Readme per un repository GitHub.

Includi un titolo descrittivo e breve, un breve sommario in un linguaggio semplice e divulgativo.

Racchiudi l'output in una parentesi di codice con linguaggio markdown.

Dettagli del repository da usare nel sommario

Nome del Repository:

Descrizione del Repository:

Prompt generalizzato:

Alla fine di questo messaggio trovi del codice [che tipologia di codice].

Analizza e scrivi una documentazione tecnica in grado di descrivere:

1- funzione del file

2- eventuali errori di sintassi

3- struttura sintattica

Esempio:

Alla fine di questo messaggio trovi del codice PHP.

Analizzalo e scrivi una documentazione tecnica in grado di descrivere:

1- funzione del file

2- eventuali errori di sintassi

3- struttura sintattica

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $target_dir = "uploads/";
    $target_file = $target_dir .
basename($_FILES["fileToUpload"]["name"]);
    $uploadOk = 1;
    $imageFileType =
strtolower(pathinfo($target_file,PATHINFO_EXTENSION));

    // Controlla se il file è un'immagine
    if(isset($_POST["submit"])) {
        $check =
getimagesize($_FILES["fileToUpload"]["tmp_name"]);
```

```

        if($check !== false) {
            echo "File is an image - " . $check["mime"] . ".";
            $uploadOk = 1;
        } else {
            echo "File is not an image.";
            $uploadOk = 0;
        }
    }
}

// Verifica che l'estensione sia di un formato immagine
accettato (jpg, png, gif, etc.)
if (!preg_match("/(jpg|jpeg|png|gif)$/i", $imageFileType)) {
    echo "Only JPG, JPEG, PNG & GIF files are allowed.";
    $uploadOk = 0;
}

// Carica il file
if ($uploadOk == 0) {
    echo "Sorry, your file was not uploaded.";
} else {
    if
(move_uploaded_file($_FILES["fileToUpload"]["tmp_name"],
$target_file)) {
        echo "The file ". htmlspecialchars( basename(
$_FILES["fileToUpload"]["name"])) . " has been uploaded.";

// ... [parte precedente dello script] ...

// Estrazione dei metadati EXIF
$exif = exif_read_data($target_file);
if ($exif) {
    echo "<h2>Metadati Estratti:</h2>";
    echo "<table border='1'>";
    foreach ($exif as $key => $value) {
        echo "<tr><td>$key</td><td>";
        if (is_array($value)) {
            echo "<pre>" . print_r($value, true) . "</pre>"; //
Gestisce array
        } else {
            echo htmlspecialchars($value); // Gestisce stringhe
normali
        }
        echo "</td></tr>";
    }
}

```

```

    }
    echo "</table>";
} else {
    echo "No EXIF data found for this image.";
}

    } else {
        echo "Sorry, there was an error uploading your file.";
    }
}
}
?>

```

Nota: la finestra di chat di ChatGPT - Data Analyst permette l'upload di documenti, tuttavia le restrizioni di sicurezza dell'ambiente di esecuzione del modello non permettono di eseguire direttamente molti linguaggi. Per ovviare a questo problema è sufficiente inserire lo script che volete analizzare alla fine delle istruzioni.

Prompt per analisi e visualizzazione dati

Prompt generalizzato: Crea un grafico [tipologia di grafico] che rappresenti i seguenti valori {"Valore1":N1, "Valore2": N2, "Valore3": N3, "Valore4": N4} utilizzando uno schema di colori [colori scelti].

Esempio: Crea un grafico a torta che rappresenti i seguenti valori {"Studio":5, "Tempo libero": 1, "Sport": 4, "Riposo": 3} utilizzando uno schema di colori pastello.

Prompt generalizzato: crea una palette per rappresentare [argomento] in un grafico [tipologia di grafico]

Esempio: suggerisci una palette di colori idonea a rappresentare in un grafico a barre dati relativi al cambiamento climatico e al riscaldamento globale.

Esempio: suggerisci una tipologia di grafico ed un set di colori che renda la rappresentazione dei miei dati accessibile agli utenti ipovedenti.

Prompt generalizzato: quale tipo di grafico posso utilizzare per visualizzare i dati relativi a [campi da rappresentare] che trovi nel dataset allegato?

Esempio: quale grafico posso utilizzare per rappresentare la distribuzione degli artisti per nazionalità nel dataset che ti ho allegato?

Esempio: suggerisci una soluzione per ottimizzare un grafico a barre per la visualizzazione da mobile.

Prompt: Agisci come un analista di dati per creare una rappresentazione grafica che illustri
Agisci come un analista di dati. Voglio condurre un'analisi statistica su [inserisci argomento].
Nell'analisi approfondirò il tema [inserisci argomento]. Potresti suggerirmi cinque dataset più pertinenti per il mio progetto?

Prompt: come si distribuisce [caratteristica specifica] all'interno del [dataset specifico].

Prompt: Agisci come un analista di dati, sviluppa un codice per analizzare e visualizzare il [dataset specificato].

Prompt:

Prompt: Sei un esperto di fogli di calcolo. Elabora una formula per Microsoft Excel che soddisfi [descrizione dei requisiti].

Fornisci la stessa formula adattata per Google Sheets

Prompt: Agisci come un programmatore. Scrivi un'espressione regolare in Python che soddisfi [descrizione regex].

Scrittura a partire da un testo

ChatGPT è un modello che permette di instaurare conversazioni e interfacciarsi facilmente con un LLM ed è progettato per comprendere e generare testi utilizzando un linguaggio naturale. Tuttavia, come accade con altri modelli, anche ChatGPT soffre di “allucinazioni”, delle situazioni in cui, durante la conversazione, vengono generate delle risposte con informazioni false. Le “allucinazioni” sono un elemento critico, soprattutto con ChatGPT, un modello in grado di rendere la conversazione e le informazioni riportate estremamente realistiche, anche se false. La frequenza di “allucinazioni” è sensibilmente diminuita con l'avvento del modello GPT-4 su cui si basa ChatGPT-4, nonostante questo miglioramento è di fondamentale importanza verificare autonomamente l'attendibilità delle informazioni in testi generati completamente dal modello.

Una strategia per mitigare la possibile presenza di “allucinazioni” nella creazione di testi in ChatGPT è affiancare il nostro prompt con fonti affidabili, dei testi di riferimento attendibili e verificati, in questi casi nel prompt viene indicato di attingere al testo come base di conoscenza primaria per l'elaborazione del contenuto richiesto. E' una tecnica che non assicura una totale attendibilità delle risposte di ChatGPT ma permette un controllo migliore nella guida alla generazione di testi.

Prompt generalizzato:

Ho un breve testo dedicato a [argomento]. Te lo condivido e ti chiedo di ampliarlo.

Approfondisci con maggiori dettagli e citando testi di riferimento.

Il testo è qui di seguito: [testo]

Prompt generalizzato:

Devo scrivere un progetto dedicato a [argomento].

Ho scritto un abstract, potresti ampliarlo portandolo a circa 1000 caratteri?

[abstract]

Prompt generalizzato:

Ho un elenco puntato di azioni e risultati di progetto, avrei la necessità di trasformare il punto elenco in un breve discorso con uno stile naturale, formale e motivazionale.

Il progetto è dedicato a [argomento] e gli interventi nell'elenco sono destinati a [obiettivo delle azioni].

Riesci a trasformare il punto elenco di azioni che ti condividerò in un breve discorso?

L'elenco puntato è qui [elenco puntato]

Estrazione di informazioni

La Named-entity recognition (NER), nota anche come estrazione di entità o segmentazione di entità, rappresenta una sottocategoria dell'estrazione di informazioni, una tecnica che si concentra su individuare e classificare entità nominate menzionate in testi non strutturati in categorie predefinite, quali nomi di persone, organizzazioni, luoghi, codici medici, espressioni temporali, quantità, valori monetari, percentuali, ecc. La NER è fondamentale nel contesto dell'elaborazione del linguaggio naturale (NLP), poiché consente di trasformare i dati testuali non strutturati in informazioni strutturate e facilmente analizzabili.

Un esempio classico di NER implica la trasformazione di un blocco di testo non annotato, come "Mario ha visitato la città di Roma nel 2023", in un blocco di testo annotato che evidenzia le entità nominate: "[Mario]Persona ha visitato la città di [Roma]Luogo nel [2023]Tempo". In questo esempio, vengono identificati e classificati un nome di persona, un luogo e un'espressione temporale.

Il processo completo di NER è spesso suddiviso concettualmente, e talvolta anche nelle implementazioni, in due problemi distinti: la rilevazione dei nomi e la classificazione dei nomi secondo il tipo di entità a cui si riferiscono (ad esempio, persona, organizzazione o luogo).

La prima fase è tipicamente semplificata in un problema di segmentazione: i nomi sono definiti come intervalli contigui di token, senza annidamenti, in modo che "Banca d'Italia" sia un unico nome, nonostante all'interno di questo nome, la sottostringa "Italia" sia essa stessa un nome. Questo problema di segmentazione è formalmente simile al chunking, la seconda fase invece richiede la scelta di un'ontologia per organizzare le categorie di cose.

Espressioni temporali e alcune espressioni numeriche (ad esempio, denaro, percentuali, ecc.) possono anche essere considerate entità nominate nel contesto del compito di NER.

Un esempio pratico dell'uso di ChatGPT per la NER potrebbe essere l'analisi di un articolo di giornale. ChatGPT può scansionare l'articolo e identificare le entità principali, classificandole in base al loro tipo. Ad esempio, se l'articolo menziona "Bill Gates", "Microsoft" e "Seattle", ChatGPT è in grado di riconoscere "Bill Gates" come persona, "Microsoft" come organizzazione e "Seattle" come luogo. La capacità di ChatGPT di gestire grandi volumi di testo lo rende particolarmente utile nella NER su testi estesi, ad esempio è possibile elaborare attraverso ChatGPT interi capitoli di un libro, estrarre personaggi, luoghi, eventi temporali e altri elementi rilevanti, fornendo così un riassunto strutturato delle entità principali. Va notato come le finestre di testo disponibili per il modello GPT 4 siano diventate molto ampie dopo l'aggiornamento del 6 novembre 2023 ed è effettivamente possibile fornire in chat un testo di 20 pagine senza difficoltà. La finestra di contesto complessiva del modello GPT 4 è infatti di 128.000 token che corrispondono a circa 500.000 caratteri, questa enorme quantità di dati permette di poter effettuare operazioni di NER direttamente dalla chat con interi documenti. In un elenco, come un inventario o un database, ChatGPT può rapidamente categorizzare gli elementi in base a parametri specifici, come tipo di prodotto, produttore o modello, costruendo un dataset manipolabile con strumenti di calcolo.

Estrazione NER da dataset Pubblici o da API

Prompt: Utilizzando il dataset pubblico [inserire nome del dataset o URL dell'API], identifica e classifica tutte le entità nominate presenti, concentrandoti su nomi di persone, organizzazioni e luoghi.

Estrazione NER da testi lunghi

Prompt: Analizza il testo fornito come allegato ed elenca tutte le entità nominate identificate, classificandole in categorie come personaggi, luoghi e eventi temporali.

Salva il risultato in un file CSV denominato lista.csv

Estrazione NER da documenti tecnici e da elenchi

Prompt: Esamina il seguente documento tecnico [inserire link o testo del documento] e identifica tutte le entità tecniche come tecnologie, nomi di aziende, standard e protocolli.

Crea un elenco categorizzato di queste entità e salvalo in un file CSV con tre colonne: nome_azienda, termini_tecnologici, protocolli.

Esempi di operazioni NER tramite API

L'estrazione di entità come luoghi o nomi da testi articolati e complessi, può essere automatizzata nel caso si abbia una serie di testi relativi allo stesso dominio. In questo esempio vengono analizzati una serie di articoli con contenuti storico-artistici e viene chiesto allo script di estrarre una lista di "artisti noti" per poi salvare i risultati in un file CSV composto da 2 colonne, nella prima colonna viene indicato il nome del file di testo analizzato e nella seconda colonna gli artisti rilevati all'interno del testo.

Questa soluzione permette di eseguire velocemente un'analisi preliminare su una grande collezione di documenti utile ad identificare i testi con maggiori riferimenti a figure storiche precise.

Codice di esempio:

```
import openai
import os
import csv

# Configura la tua chiave segreta API
openai.api_key = 'LA TUA CHIAVE API OPENAI'

# Percorsi delle cartelle
INPUT_DIR = "NER"
OUTPUT_CSV = "artisti_estrazione.csv"

# Lista per i risultati
results = []

# Elenca tutti i file nella cartella di input
for filename in os.listdir(INPUT_DIR):
    if filename.endswith(".txt"):
        # Leggi il contenuto del file
        with open(os.path.join(INPUT_DIR, filename), 'r',
encoding='utf-8') as file:
            content = file.read()

            # Invia il contenuto al modello GPT-3.5-turbo per
l'estrazione delle entità
            response = openai.ChatCompletion.create(
                model="gpt-3.5-turbo-0125",
                messages=[
                    {"role": "system", "content": "Identify all
entities that exactly match this type of artists:
Painters, Sculptors, Architects, Female painters, Decorators,
Engravers, Photographers.
Exclude writers, authors, art critics, philosophers, cardinals,
clergy, kings, queens, popes, governors, dukes, and everything
that cannot be categorized in this list:
Painters, Sculptors, Architects, Female painters, Decorators,
Engravers, Photographers."}]
            )
```

List all these entities in a list separated by comma. Avoid in the list any other words to describe entities.

```
"}",
        {"role": "user", "content": content}
    ]
)

# Estrai le entità (artisti) dal testo restituito
extracted_artists =
response.choices[0].message['content'].strip()

# Aggiungi al risultato
results.append([filename, extracted_artists])

# Scrivi i risultati in un file CSV
with open(OUTPUT_CSV, 'w', newline='', encoding='utf-8') as
csvfile:
    writer = csv.writer(csvfile)
    writer.writerow(['Nome File', 'Artisti']) # Intestazioni
delle colonne
    writer.writerows(results)

print("Estrazione completata!")
```

Traduzione e descrizione

Capacità di contestualizzazione di ChatGPT

L'uso di ChatGPT nella traduzione di testi è caratterizzato da una differenza chiave rispetto alle traduzioni con altri software, ChatGPT è in grado di interpretare e riflettere il contesto in cui un testo è inserito, una caratteristica che permette di migliorare l'accuratezza delle traduzioni prodotte, in questo scenario, diventa così essenziale per l'utente fornire non soltanto il testo da tradurre, ma anche dettagli relativi al contesto in cui esso è situato. Ad esempio, la traduzione di un dialogo tratto da un romanzo può beneficiare della menzione del genere letterario o del tono narrativo, queste informazioni di contesto assistono ChatGPT nella scelta di parole e frasi congruenti con il testo sorgente, così come per la traduzione di espressioni idiomatiche o terminologie settoriali, può essere efficace specificare l'origine culturale o regionale così da accordare la traduzione alle sfumature specifiche della lingua originale.

Il modello Transformer su cui si basa ChatGPT consente una comprensione approfondita di contesto e significato, portando a traduzioni che catturano con maggiore efficacia le sfumature linguistiche, per utilizzare le funzionalità dei Transformer in grado di catturare il

contesto dell'intero documento significa per l'utente un cambio di approccio nella traduzione che introduce una certa "creatività" nella formulazione della richiesta. Una metodologia efficace consiste nel richiedere sinonimi contestualizzati per termini specifici, come nel prompt: "Fornisci un sinonimo in [Lingua di Destinazione] per la parola [Parola], insieme alla sua definizione". Ciò permette di selezionare termini che migliorano la qualità della traduzione, avvalendosi delle spiegazioni dettagliate fornite da ChatGPT per una comprensione più approfondita dell'uso contestuale e delle sfumature dei termini.

Una differenza importante rispetto alle traduzioni tradizionali è la possibilità per l'utente di cambiare, per lo stesso testo, stile di scrittura, tono e grado di formalità. Una traduzione destinata a un pubblico accademico, per esempio, richiederà l'uso di un linguaggio formale e di una terminologia specifica, mentre un testo destinato ai social media potrà adottare un tono più colloquiale e diretto. Gli utenti sono incoraggiati a sperimentare con diverse formulazioni di richieste di traduzione per valutare come ChatGPT adatta il suo output, garantendo una flessibilità superiore rispetto ad altri strumenti di traduzione.

Prompt generalizzati:

Agisci come un traduttore specializzato in lingua [lingua]. I testi forniti saranno in lingua [lingua originale], traduci i testi in una versione corretta e migliorata del mio testo, in [lingua]. Correggi gli errori senza cambiare il significato.

Il testo che produrrai sarà in un [lingua] con una grammatica perfetta e con uno stile formale. Rispondi solo con la correzione senza scrivere spiegazioni.

Traduci il seguente brano '[testo sorgente]' in [lingua di destinazione], tenendo conto del contesto [tipologia di contesto] e del pubblico di riferimento [tipo di pubblico].

Come si dice il proverbio/modo di dire '[testo sorgente]' in [lingua di destinazione]? Fornisci anche una breve spiegazione del significato culturale.

Traduci questo estratto letterario '[testo sorgente]' in [lingua di destinazione], mantenendo lo stile e il tono dell'autore originale.

Traduci questo testo tecnico/scientifico '[testo sorgente]' in [lingua di destinazione], utilizzando terminologia specifica del campo [settore di riferimento].

Traduci la seguente frase '[testo sorgente]' in [lingua di destinazione] e fornisci una spiegazione grammaticale per aiutarmi a capire come si struttura la frase nella lingua di destinazione.

Fornisci un riassunto in [lingua di destinazione] di questo articolo/lungo testo'[testo sorgente], evidenziando i punti chiave e mantenendo le informazioni essenziali.

Traduci i seguenti dialoghi di una sceneggiatura '[testo sorgente]' in [lingua di destinazione] , conservando le intonazioni e le sfumature emotive dei personaggi.

Consigli utili

Per garantire che ChatGPT produca testi coesi e scorrevoli, è possibile indicare nel prompt l'istruzione *fluida e senza interruzioni*, ad esempio:

Il testo che scriverai dovrà essere di almeno 500 parole in uno stile neutrale, fluido e senza interruzioni.

Questo approccio aiuta a prevenire la creazione di testi frammentati, caratterizzati da un eccesso di elenchi, paragrafi brevi e liste, inoltre, per assicurarsi che il tono del testo rimanga neutrale e privo di giudizi qualitativi, è utile specificare che si desidera uno stile *neutrale e senza giudizi qualitativi*, questo evita l'inclusione di aggettivi superlativi e opinioni non necessarie. Per ottenere un risultato ancor più incentrato sulla descrizione, si può indicare la preferenza per:

un testo senza espressioni positive, un testo neutrale e descrittivo dedicato a [argomento].

Si tratta di indicazioni semplici che in molti casi aiutano nel guidare la stesura dei testi in ChatGPT verso dei risultati che siano più vicini ad una scrittura umana, continua e omogenea.

Template e GPTs

Il 14 marzo 2023 viene pubblicato il modello GPT-4 e in ChatGPT, anche se con delle limitazioni, viene introdotta la possibilità di conversare utilizzando il modello più recente di OpenAI. Una delle caratteristiche emerse è la facilità del modello di autoconfigurarsi dietro specifiche istruzioni fornite tramite prompt, impostando così delle specifiche funzioni e delle regole di interazione con l'utente, ChatGPT si configura così come un'interfaccia per la creazione di testi e risultati variabili. Un esempio facilmente comprensibile di questa tecnica lo possiamo trovare nell'articolo "New ChatGPT Prompt Engineering Technique: Program Simulation" di Scalamogna,² e nel successivo "Prompt Engineering Evolution: Defining the New Program Simulation Prompt Framework"³, in questi esempi viene mostrato come il concetto di "template", ovvero un prompt generalizzato con parametri sostituibili e

²

<https://towardsdatascience.com/new-chatgpt-prompt-engineering-technique-program-simulation-56f49746aa7b>

³

<https://towardsdatascience.com/prompt-engineering-evolution-defining-the-new-program-simulation-prompt-framework-d8a1ee096904>

personalizzabili, si possa spingere fino alla creazione di veri e propri prompt “attivatori” in grado di trasformare una chat con il modello in un’interfaccia per la creazione guidata di contenuti o per la creazione di nuovi prompt.

Qui di seguito è riportato il template testato da Scalamogna a settembre 2023:

“Behave like a self-assembling program whose purpose is to create illustrated children’s stories. You have complete flexibility on determining the program’s functions, features, and user interface. For the illustration function, the program will generate prompts that can be used with a text-to-image model to generate images. Your goal is to run the remainder of the chat as a fully functioning program that is ready for user input once this prompt is received.”

Va notato come lo stesso prompt tradotto da ChatGPT in lingua italiana genera risultati sostanzialmente diversi e con un’interattività ridotta.

Tra i template per prompt si sono diffuse diverse strategie che adottano proprio questa tecnica per generalizzare interi comportamenti da far assumere a ChatGPT. Un template di prompt molto popolare è destinato ad esempio alla realizzazione di prompt efficaci:

I want you to become my Prompt engineer. Your goal is to help me craft the best possible prompt for my needs. The prompt will be used by you, ChatGPT. You will follow the following process:

1. Your first response will be to ask me what the prompt should be about. I will provide my answer, but we will need to improve it through continual iterations by going through the next steps.
2. Based on my input, you will generate 2 sections. a) Revised prompt (provide your rewritten prompt. it should be clear, concise, and easily understood by you), b) Questions (ask any relevant questions pertaining to what additional information is needed from me to improve the prompt)
3. We will continue this iterative process with me providing additional information to you and you updating the prompt in the Revised prompt section until I say we are done.

Adattato in italiano con:

Comportati come un ingegnere dei prompt. Il tuo obiettivo è aiutarmi a creare il prompt migliore possibile per le mie esigenze. Il prompt sarà utilizzato da te, ChatGPT. Seguirai il seguente processo:

1. La tua prima risposta sarà chiedermi di cosa dovrebbe trattare il prompt. Io fornirò la mia risposta, ma dovremo migliorarla attraverso continue iterazioni seguendo i passaggi successivi.

2. In base al mio input, genererai 2 sezioni.

a) Prompt rivisto (fornisci il tuo prompt riscritto. Dovrebbe essere chiaro, conciso e facilmente comprensibile per te),

b) Domande (fai qualsiasi domanda pertinente riguardo a quali informazioni aggiuntive sono necessarie da parte mia per migliorare il prompt)

3. Continueremo questo processo iterativo con me che ti fornisco informazioni aggiuntive e tu che aggiorni il prompt nella sezione Prompt rivisto fino a quando non dirò che abbiamo finito.

Il template si dimostra una guida efficace per scrivere un prompt attraverso un dialogo con la macchina stessa e diventa un utile esercizio di riflessione sui dettagli e sulle sfumature di significato che vogliamo ottenere nel risultato finale.

A novembre 2023, solo otto mesi dopo la pubblicazione del modello GPT-4, OpenAI introduce GPTs, versioni personalizzabili di ChatGPT per scopi specifici, che possono essere create senza bisogno di conoscenze di programmazione. Queste permettono di personalizzare ChatGPT per l'uso quotidiano e di condividerle pubblicamente. La creazione di una GPT è guidata direttamente dall'interfaccia offerta nel pannello di controllo di ChatGPT e avviene attraverso una conversazione in cui vengono definite le capacità del modello personalizzato come la possibilità di fare ricerche sul web o la specializzazione verso l'analisi di dati.

Con la pubblicazione dei GPTs, i template passano così in secondo piano essendo formalmente integrati in un prodotto fornito direttamente da OpenAI, tuttavia sono una tecnica di ingegneria del prompt che permette una comprensione migliore delle potenzialità del modello e permettono un controllo diretto e personalizzato sull'output.

Prompt Zero-Shot

La strategia zero-shot nel campo del prompt engineering rappresenta un approccio per il miglioramento della comprensione e dell'elaborazione delle richieste da parte dei modelli di intelligenza artificiale come GPT-4. Il zero-shot si basa sulla capacità del modello di rispondere a domande o eseguire compiti senza aver ricevuto esempi specifici in precedenza. Questa metodologia sfrutta l'addestramento del modello su vasti dataset, consentendogli di generalizzare e applicare le sue conoscenze a nuove situazioni con notevole efficacia. Il cuore della tecnica zero-shot risiede nella formulazione di prompt chiari

e diretti, questi prompt devono essere concepiti per guidare il modello verso l'obiettivo desiderato senza fare affidamento su esempi precedenti. Un elemento fondamentale quindi è la chiarezza e la specificità delle istruzioni: i prompt devono essere dettagliati e diretti, fornendo al modello tutte le informazioni necessarie per comprendere il contesto e l'obiettivo del compito. Un altro aspetto cruciale del zero-shot è la sua flessibilità, non basandosi su esempi specifici, questa strategia è estremamente versatile e può essere applicata a un'ampia varietà di compiti, da quelli che richiedono una comprensione contestuale profonda a quelli che implicano creatività o ragionamento astratto. Questo rende il zero-shot particolarmente utile in situazioni dove non si dispone di esempi pertinenti o in compiti inediti per i quali il modello non è stato specificamente addestrato.

L'assenza di esempi concreti da cui attingere rappresenta anche una difficoltà per il modello nell'adattarsi a compiti che richiedono una comprensione molto specifica o dettagliata. In questi casi, il modello deve fare affidamento esclusivamente sulle sue capacità di inferenza e sulle conoscenze acquisite durante la fase di addestramento.

Prompt Few-Shot

Il "Few-Shot Prompting" è una tecnica utilizzata per migliorare la qualità e la precisione delle risposte fornite da modelli come GPT-4 e viene utilizzata quando è necessario perfezionare il risultato per un compito specifico o un dominio senza avere accesso a un vasto dataset. Attraverso l'approccio Few-Shot, forniamo al modello alcuni esempi (gli "shots" appunto) prima di fare la nostra domanda effettiva, questo aiuta il modello a comprendere meglio il contesto e il tipo di risposta che ci aspettiamo. Possiamo immaginare il few-shot prompting come una mini-sessione di formazione per il modello attraverso una sequenza di prompt forniti dall'utente. La quantità di esempi forniti influenza direttamente la qualità della risposta, specialmente per compiti più complessi, tuttavia il numero rimane un elemento da calibrare contesto per contesto e in alcuni una quantità eccessiva di esempi potrebbe avere un effetto controproducente sovraccaricando il modello e generando incongruenze.

La tecnica Few-Shot Prompt è estremamente utile per guidare i LLM verso il tipo specifico di risposta desiderata, fornendo esempi rilevanti e ben scelti, mantenendo la struttura del prompt coerente e uniforme, è possibile migliorare significativamente la qualità e la rilevanza delle risposte ottenute indirizzando il modello nella giusta direzione. E' importante tuttavia, riconoscere i limiti di questa tecnica, in particolare per compiti che richiedono un ragionamento più elaborato come per i calcoli matematici o i problemi logici che richiedono più elaborazioni ed iterazioni, in tali scenari, il few-shot può essere efficacemente combinato con altre tecniche come il Chain-of-Thought aumentando l'efficacia nei compiti complessi e articolati.

Linee guida ed esempi

Scegliere esempi rilevanti: i pochi esempi che fornisci devono essere attentamente selezionati per essere il più rilevanti possibile alla tua domanda, questi esempi dovrebbero rappresentare il tipo di risposta o il formato che stai cercando. Ad esempio, se vuoi che il modello generi poesie, fornisci esempi di poesie.

Chiarezza: anche se stai fornendo esempi, questi devono essere chiari e concisi. Evita esempi troppo lunghi o complessi, poiché potrebbero confondere il modello anziché aiutarlo.

Coerenza: gli esempi dovrebbero essere coerenti tra loro in termini di stile, tono e formato, questo aiuta il modello a "capire" meglio quale tipo di risposta ti aspetti.

Formulazione della domanda finale: dopo aver presentato gli esempi, formula la tua domanda finale in modo chiaro assicurandoti che sia in linea con gli esempi forniti.

Di seguito troverai alcuni esempi in cui viene utilizzata la tecnica Few-Shot:

Scrittura creativa:

Utente:

Scrivi una storia breve che inizia con la seguente frase: "Era una notte tempestosa, il vento ululava attraverso le cime degli alberi e la pioggia cadeva a dirotto."

Esempi di contesto:

- Un vecchio faro solitario si ergeva su una scogliera battuta dalle onde.
- Il protagonista è un giovane capitano di nave naufragato su quest'isola deserta.
- Il faro sembra misteriosamente acceso, nonostante sia stato abbandonato da anni.

Risposta:

Era una notte tempestosa, il vento ululava attraverso le cime degli alberi e la pioggia cadeva a dirotto. Un vecchio faro solitario si ergeva su una scogliera battuta dalle onde, illuminando a intermittenza il mare agitato. Il giovane capitano di nave, naufragato su quest'isola deserta, era esausto e stremato dal naufragio. Strisciò a fatica verso il faro, sperando di trovare rifugio.

Mentre si avvicinava, notò con sorpresa che il faro sembrava misteriosamente acceso, nonostante fosse stato abbandonato da anni. La luce emessa era irregolare, come se qualcuno o qualcosa fosse al controllo. Con il cuore in gola, il capitano salì le scale in rovina fino alla stanza della lanterna.

Lì, nella penombra, scoprì un vecchio marinaio barbuto che operava freneticamente i meccanismi del faro. Il marinaio sembrava sorpreso quanto il capitano di trovarsi in quella situazione. Senza una parola, i due uomini si guardarono negli occhi, e in quel momento, compresero che il destino li aveva uniti in quella notte tempestosa.

Il marinaio spiegò che aveva passato anni a mantenere il faro acceso, sperando che un giorno qualcuno sarebbe arrivato in cerca di aiuto. Il giovane capitano era quel qualcuno. Insieme, decisero di lavorare per segnalare la loro posizione e cercare un modo per farsi salvare.

Mentre la tempesta infuriava all'esterno, i due uomini lavorarono fianco a fianco, con una luce di speranza che bruciava intensamente. Quella notte tempestosa avrebbe segnato l'inizio di una straordinaria amicizia e di un'avventura che li avrebbe portati lontano dalle scogliere battute dalle onde.

<https://chat.openai.com/share/0b85b1a6-83fc-484d-9839-81f8cca9a37e>

Prompt generalizzato:

Scrivi una storia breve che inizia con la seguente frase: "[frase]."

Esempi di contesto:

- [esempio1]
- [esempio2]
- [esempio3]

Scrittura di una lettera di presentazione

Ti fornirò alcuni esempi di lettere di presentazione. Utilizza questi esempi per comprendere il tono, lo stile e il contenuto richiesti, e poi scrivi una lettera di presentazione per un candidato che si sta applicando per la posizione di responsabile dei servizi IT.

Esempio 1: Candidato per Ruolo di Responsabile Marketing

Estratto della Lettera: "Come professionista esperto nel marketing con oltre una decade di esperienza nel guidare team e campagne di successo, sono entusiasta di portare la mia creatività e le mie competenze strategiche nel ruolo di Responsabile Marketing presso la vostra azienda..."

Esempio 2: Candidato per Ruolo di Ingegnere del Software

Estratto della Lettera: "Con una solida formazione in ingegneria del software e una passione per lo sviluppo di soluzioni tecnologiche innovative, mi candido con entusiasmo per il ruolo di Ingegnere del Software nella vostra azienda. La mia esperienza nello sviluppo di software ad alte prestazioni e nel lavoro in team agile mi rende un candidato ideale..."

Esempio 3: Candidato per Ruolo di Analista Finanziario

Estratto della Lettera: "In qualità di analista finanziario con un'acuta comprensione dei mercati e delle tendenze economiche, sono pronto a contribuire con analisi approfondite e strategie finanziarie nel ruolo di Analista Finanziario presso la vostra azienda..."

Richiesta Specifica:

Ora, basandoti sugli esempi forniti, scrivi una lettera di presentazione per un candidato che si sta candidando per la posizione di responsabile dei servizi IT. La lettera dovrebbe riflettere un forte background in IT, competenze di leadership, e la capacità di gestire efficacemente le operazioni IT in un'organizzazione.

Risposta:

Gentili Signori,

Mi rivolgo a voi per manifestare il mio interesse per la posizione di Responsabile dei Servizi IT presso la vostra stimata azienda. Con un'esperienza pluriennale nel settore dell'informatica e una comprovata capacità di guidare team IT verso il successo, sono convinto di poter apportare un valore significativo al vostro team.

Nel corso della mia carriera, ho acquisito una solida esperienza nella gestione di infrastrutture IT complesse, nell'implementazione di soluzioni tecnologiche innovative e nella conduzione di progetti IT di ampia scala. La mia competenza non si limita alla sola conoscenza tecnica; ho sviluppato forti capacità di leadership, che mi hanno permesso di guidare i miei team con efficacia, promuovendo un ambiente di lavoro collaborativo e incentrato sugli obiettivi.

In qualità di responsabile IT, ho sempre dato priorità all'ottimizzazione dei processi e alla sicurezza dei sistemi, garantendo allo stesso tempo che le esigenze tecnologiche dell'organizzazione fossero soddisfatte in modo efficiente. La mia abilità nel comprendere e anticipare le esigenze aziendali, unita alla capacità di tradurre queste necessità in soluzioni IT efficaci, mi rende un candidato ideale per questo ruolo.

Sono entusiasta all'idea di portare la mia esperienza, le mie competenze e la mia passione per la tecnologia nel vostro team. Sono convinto che la mia proattività e la mia visione strategica possano contribuire in modo significativo al successo del vostro dipartimento IT.

Vi ringrazio per aver preso in considerazione la mia candidatura e sono a disposizione per ulteriori informazioni o per un incontro di persona, al fine di discutere più approfonditamente del mio contributo alla vostra azienda.

Cordiali saluti,
[Il Tuo Nome]

<https://chat.openai.com/share/86cae9af-96f4-4aa7-bced-01c8fdaaabce>

Prompt generalizzato

Ti fornirò alcuni esempi di [tipologia di documento]. Utilizza questi esempi per comprendere il tono, lo stile e il contenuto richiesti, e poi scrivi [tipologia di documento] per [dettagli su destinatario e contesto]

Sentiment Analysis in few-shot

Esempio 1: "Amo il sole e il mare!" // Positive

Esempio 2: "Roma è una città bellissima" // Positive

Esempio 3: "La guerra è una maledizione" // Negative

Esempio 4: "La situazione in cui siamo è orribile!" // Negative

Domanda

- "Queste mele hanno un ottimo sapore" //

Risposta: "These apples have a great taste." // Positive

<https://chat.openai.com/share/3e0adbf2-96bc-4fc1-9afb-68a66edca1da>

Chain of Thought

Il metodo di "Prompt Chain-of-Thought" (CoT), è un tecnica di scrittura di prompt basata sull'impiego di una serie di fasi intermedie di ragionamento per guidare il modello verso una risposta complessa, incrementando così la sua capacità di risolvere problemi complessi che richiedono più passaggi logici. La tecnica CoT si distingue per la sua analogia con il "pensare ad alta voce", ovvero esporre i passaggi intermedi del ragionamento in modo esplicito, questa caratteristica aiuta il modello a destrutturare e affrontare sistematicamente le varie parti di un problema complesso, ad esempio, in un problema matematico, come la somma dei numeri dispari di un insieme. In questo caso invece di limitarsi a calcolare e fornire la somma, il modello elenca e somma esplicitamente i numeri dispari, mostrando il processo di pensiero che conduce alla conclusione se la somma è pari o dispari esponendo quindi tutti i calcoli intermedi. E' una strategia di scrittura del prompt che quindi si adatta bene a problemi che richiedono un ragionamento sequenziale e dettagliato.

La tecnica CoT può essere combinata con il metodo di prompting few-shot per migliorare ulteriormente le prestazioni su task complessi, in questo scenario il modello viene istruito con pochi esempi in cui ogni esempio include la fase di ragionamento dettagliata.

Per usare questa tecnica con ChatGPT, è necessario sollecitare il modello con istruzioni in cui viene richiesto di spiegare il suo processo di pensiero passo dopo passo. Ad esempio, nel risolvere un problema matematico, potresti chiedere a ChatGPT di descrivere ogni fase della risoluzione dell'equazione, o nel formulare un'opinione su un argomento, di esporre i vari argomenti e controargomenti che considera.

Linee guida ed esempi

Spiegazione di concetti articolati

Prompt:

Vorrei che spiegassi il Teorema di Pitagora. Inizia con una breve introduzione al concetto del teorema, che afferma che in un triangolo rettangolo, l'area del quadrato costruito sull'ipotenusa (il lato opposto all'angolo retto) è uguale alla somma delle aree dei quadrati costruiti sugli altri due lati. Prosegui poi analizzando un esempio specifico, come un triangolo con lati di lunghezza 3, 4 e 5, mostrando come il teorema si applichi in questo caso. Infine, discuti brevemente le applicazioni pratiche di questo teorema nella vita quotidiana o in altri campi della scienza.

Risposta:

Il Teorema di Pitagora è un principio fondamentale della geometria euclidea, che riguarda i triangoli rettangoli. Secondo questo teorema, in un triangolo rettangolo, l'area del quadrato costruito sull'ipotenusa (il lato opposto all'angolo retto) è uguale alla somma delle aree dei quadrati costruiti sugli altri due lati, noti come cateti. Matematicamente, questo è espresso dalla formula $a^2 + b^2 = c^2$, dove c è la lunghezza dell'ipotenusa, e a e b sono le lunghezze dei cateti.

Per esemplificare, consideriamo un triangolo rettangolo con i lati di lunghezza 3, 4 e 5. Qui, i lati più corti (3 e 4) sono i cateti e il lato più lungo (5) è l'ipotenusa. Applicando il teorema di Pitagora:

$$3^2 + 4^2 = 5^2$$

$$9 + 16 = 25$$

$$25 = 25$$

Questo mostra che il teorema funziona in questo caso, poiché la somma delle aree dei quadrati sui cateti ($9 + 16$) è uguale all'area del quadrato sull'ipotenusa (25).

Le applicazioni pratiche del Teorema di Pitagora sono numerose e varie. In architettura e ingegneria, è spesso utilizzato per determinare distanze inaccessibili o per calcolare l'inclinazione di strutture e terreni. In fisica, aiuta a risolvere problemi riguardanti vettori e spostamenti in due dimensioni. Inoltre, il concetto trova applicazione in vari campi come il design, la navigazione, l'analisi spaziale in geografia, e persino in alcuni aspetti dell'informatica e della grafica digitale. In sintesi, il Teorema di Pitagora è uno strumento essenziale per molteplici discipline scientifiche e tecniche.

<https://chat.openai.com/share/215c3c3e-f054-4896-8f61-a3157ec8727c>

Prompt generalizzato:

Vorrei che spiegassi il [argomento]. Inizia con una breve introduzione al concetto [descrizione verificata del concetto]. Prosegui poi analizzando un esempio specifico, come [esempio realistico], mostrando come [argomento] si applichi in questo caso. Infine, discuti brevemente le applicazioni pratiche di questo [argomento] nella vita quotidiana o in altri campi [eventuale campo specifico di applicazione].

Self Consistency

La tecnica di "Self-Consistency" è un approccio che punta a generare più percorsi di ragionamento e a confrontarli, al fine di identificare la risposta più coerente e logica. Questo metodo si rivela particolarmente utile in contesti che richiedono elaborazioni complesse, come problemi aritmetici o questioni di senso comune, contribuendo ad aumentare l'accuratezza delle risposte del modello grazie al confronto tra diverse soluzioni, e una maggiore versatilità, essendo applicabile a una vasta gamma di domande.

La tecnica assicura che la risposta fornita sia coerente con il contenuto generato e può essere paragonata ad un processo di autoverifica in cui il modello, come uno studente che rivede le proprie risposte, fa riferimento alle risposte precedenti durante una conversazione. Questo orientamento guida a fondare le nuove risposte sui contenuti già presentati, ad esempio chiedendole di fornire una definizione per ciascun termine di una lista precedentemente generata. L'approccio è un'iterazione in cui vengono chieste più risposte alla stessa domanda e il modello viene invitato a prendere la risposta che compare maggiormente come definitiva, questo metodo segue la tecnica del Chain of Thought (CoT) e risulta particolarmente efficace quando utilizzato in combinazione con esso.

La self-Consistency, inoltre incoraggia il modello a considerare vari approcci al problema, promuovendo un ragionamento più approfondito anche con il rischio di incontrare risposte contraddittorie che rendono difficile la selezione della soluzione corretta. Il processo inizia con la creazione di una collezione di percorsi di ragionamento per uno specifico prompt, attraverso la generazione di molteplici campioni, in seguito, l'AI determina la risposta più coerente sfruttando la sua knowledge base e le informazioni fornite nel prompt, infine fornisce la risposta selezionata come output finale.

In pratica, si chiede alla stessa intelligenza artificiale di rispondere alla stessa domanda più volte e poi si considera la risposta che fornisce più spesso come quella finale. Questo è utile perché aiuta a dare risposte che hanno più senso e sono più precise.

Uno degli ambiti principali di applicazione della "Self-Consistency" è il miglioramento della capacità di ragionamento aritmetico dei modelli di linguaggio. Spesso, i LLM come i GPT lottano con compiti che richiedono un'elaborazione aritmetica, come la somma, la sottrazione, la moltiplicazione e la divisione, attraverso tecniche che sfruttano la "Self-Consistency" vengono generate diverse soluzioni a un problema e selezionate quella più coerenti, basandosi sulle varie interazioni con il modello.

Linee guida

Immagina di avere una squadra di esperti che, indipendentemente l'uno dall'altro, risolvono lo stesso problema e poi confrontano le loro soluzioni per scegliere quella che ha più senso. Scrivere un prompt utilizzando la tecnica della Self-Consistency con un'interfaccia come ChatGPT funziona in modo simile.

Quando usi la Self-Consistency, chiedi fondamentalmente a ChatGPT di pensare a un problema in diversi modi e di fornire più risposte. Poi, il modello analizza tutte le risposte che ha generato e sceglie quella che appare più spesso o che sembra la più coerente e logica. Questo metodo aiuta a produrre una risposta finale che è più probabile sia corretta perché è stata verificata in più modi.

Ecco i passaggi semplificati di come scrivere un prompt con questa tecnica:

1. Definisci chiaramente il problema che vuoi risolvere.
2. Fai la stessa domanda più volte a ChatGPT, come se ogni volta chiedessi a un esperto diverso.
3. Guarda tutte le diverse risposte che ChatGPT ha dato.
4. Seleziona la risposta che viene data più frequentemente o quella che sembra essere la più ragionevole e coerente con ciò che ChatGPT sa già.

In pratica, stai invitando il modello ad essere più preciso facendo in modo che verifichi il proprio lavoro, proprio come farebbe un essere umano se stesse cercando di assicurarsi che la risposta a un problema difficile sia corretta.

Prompt per analisi linguistica

La linguistica computazionale è un campo interdisciplinare che coinvolge l'uso di modelli di intelligenza artificiale per analizzare e comprendere il linguaggio naturale. Di seguito sono presentati alcuni prompt utili per diversi compiti di linguistica computazionale, insieme a esempi concreti.

Analisi Morfologica

Prompt:

Analizza la morfologia della seguente frase: "[frase]". Identifica le radici delle parole, i prefissi, i suffissi e altre morfologie rilevanti.

Esempio:

Analizza la morfologia della seguente frase: "La linguistica computazionale è un campo interdisciplinare che coinvolge l'uso di modelli di intelligenza artificiale per analizzare e comprendere il linguaggio naturale.". Identifica le radici delle parole, i prefissi, i suffissi e altre morfologie rilevanti.

Risposta:

Ecco un'analisi dettagliata della morfologia della frase data:

1. **Linguistica:**
 - **Radice:** linguist- (derivato da "lingua", indica lo studio o la scienza)
 - **Suffisso:** -ica (usato per formare aggettivi o, in questo caso, sostantivi che indicano campo di studio o caratteristica)
2. **Computazionale:**
 - **Radice:** comput- (derivato da "computare", che significa calcolare)
 - **Suffisso:** -azionale (forma aggettivi che indicano relazione o appartenenza)
3. **Campo:**
 - **Radice:** campo (parola base che significa area o settore)
4. **Interdisciplinare:**
 - **Prefisso:** inter- (significa tra, attraverso, indica una nozione di reciprocità o scambio)
 - **Radice:** disciplin- (derivato da "disciplina", indica un ramo di conoscenza)
 - **Suffisso:** -are (forma aggettivi)
5. **Coinvolge:**
 - **Radice:** coinvolg- (derivato da "involgere", significa includere)
 - **Suffisso:** -e (desinenza verbale)
6. **Uso:**
 - **Radice:** uso (parola base che indica l'atto di utilizzare)
7. **Modelli:**
 - **Radice:** model- (derivato da "modello", che indica un esempio o schema)
 - **Suffisso:** -i (plurale italiano)
8. **Intelligenza:**
 - **Radice:** intelligenz- (derivato da "intelligere", che significa comprendere)
 - **Suffisso:** -a (sostantivo femminile)
9. **Artificiale:**
 - **Radice:** artifici- (derivato da "artificio", che indica qualcosa di fatto con abilità)
 - **Suffisso:** -ale (forma aggettivi che indicano appartenenza o relazione)
10. **Analizzare:**
 - **Radice:** analizz- (derivato da "analisi", processo di esame dettagliato)
 - **Suffisso:** -are (verbo)

11. **Comprendere:**

- **Radice:** comprend- (derivato da "comprehendere", che significa capire completamente)
- **Suffisso:** -ere (verbo)

12. **Linguaggio:**

- **Radice:** lingu- (relativo alla lingua)
- **Suffisso:** -aggio (indica mezzo o risultato)

13. **Naturale:**

- **Radice:** natur- (derivato da "natura", che indica le caratteristiche innate)
- **Suffisso:** -ale (forma aggettivi che indicano appartenenza o relazione)

Questa frase incorpora una varietà di morfemi derivativi che aiutano a formare termini specifici rilevanti in ambito scientifico e tecnico, illustrando bene l'uso di prefissi e suffissi per modificare e specializzare il significato delle parole radice.

Analisi Sintattica

Prompt:

Esegui l'analisi sintattica della seguente frase: "[frase]". Identifica il soggetto, il predicato, e i complementi.

Esempio:

Esegui l'analisi sintattica della seguente frase: "Il nobile caccia il cinghiale."

Risposta:

L'analisi sintattica della frase "Il nobile caccia il cinghiale." è la seguente:

Frase: "Il nobile caccia il cinghiale."

Tipo di frase: Dichiarativa, affermativa, semplice.

Analisi dei sintagmi:

Sintagma nominale (Soggetto): "Il nobile"

Determinante: "Il" (articolo determinativo)

Nome: "nobile"

Sintagma verbale (Predicato verbale): "caccia"

Verbo: "caccia" (verbo transitivo, coniugato al presente, indicativo, terza persona singolare)

Sintagma nominale (Oggetto diretto): "il cinghiale"

Determinante: "il" (articolo determinativo)

Nome: "cinghiale"

Funzioni sintattiche:

Soggetto: "Il nobile"

Predicato verbale: "caccia"

Oggetto diretto: "il cinghiale"

In sintesi, la frase è costituita da un soggetto ("Il nobile"), un predicato verbale ("caccia") e un oggetto diretto ("il cinghiale").

Analisi Semantica

Prompt:

Esegui un'analisi semantica della seguente frase: "[frase]". Identifica il significato delle parole e le loro relazioni semantiche.

Esempio:

Esegui un'analisi semantica della seguente frase: "Il manoscritto è ben conservato "

Risposta:

L'analisi semantica della frase "Il manoscritto è ben conservato" consiste nell'esaminare il significato dei suoi elementi e la relazione tra di essi.

1. Significato delle parole:

- **Il:** Articolo determinativo maschile singolare. Indica che il sostantivo che segue è noto o specifico.
- **Manoscritto:** Sostantivo maschile singolare. Si riferisce a un documento scritto a mano.
- **è:** Verbo essere, coniugato al presente, terza persona singolare. Serve come copula, collegando il soggetto con il predicato nominale.
- **ben:** Avverbio di modo. Indica la qualità della conservazione in maniera positiva.
- **conservato:** Participio passato del verbo conservare. In questo contesto, agisce come aggettivo, descrivendo lo stato del manoscritto.

2. Relazioni semantiche:

- **Soggetto:** "Il manoscritto". Si tratta dell'entità di cui si parla nella frase.
- **Predicato nominale:** "è ben conservato". Esprime una proprietà o uno stato del soggetto.
 - **Verbo copulativo:** "è". Collega il soggetto con il suo complemento.
 - **Complemento predicativo del soggetto:** "ben conservato". Fornisce informazioni sullo stato del manoscritto.

3. Significato globale:

- La frase comunica che un particolare manoscritto è mantenuto in buone condizioni.
- L'uso dell'avverbio "ben" indica che la conservazione è di qualità superiore, sottolineando positivamente lo stato del manoscritto.

In sintesi, la frase "Il manoscritto è ben conservato" trasmette l'informazione che il manoscritto è in uno stato di buona conservazione, con un'implicazione positiva sulla qualità della conservazione stessa.

Riconoscimento di Entità Nominate (NER)

Prompt:

Esegui il riconoscimento delle entità nominate nella seguente frase: "[frase]". Identifica le persone, i luoghi, le organizzazioni e altre entità rilevanti.

CLIP

CLIP (Contrastive Language-Image Pre-training) è un modello multimodale sviluppato da OpenAI, che è in grado di comprendere e correlare testi e immagini. Utilizzando CLIP, è possibile estrarre embeddings (rappresentazioni vettoriali) da immagini e testi, e poi eseguire operazioni su questi embeddings per vari compiti, come la ricerca di immagini, il clustering, e l'analisi di somiglianze.

Qui di seguito un breve script python per l'uso di un modello ridotto di CLIP per la classificazione delle immagini.

//codice

```
import os

import shutil

from pathlib import Path

import numpy as np

import pandas as pd

import torch
```



```

from PIL import Image

from tqdm.auto import tqdm

from collections import defaultdict

from transformers import CLIPProcessor, CLIPModel

from sklearn.cluster import AgglomerativeClustering

from sklearn.metrics.pairwise import cosine_similarity

from sklearn.preprocessing import StandardScaler


def load_clip_model(device):

    model =
CLIPModel.from_pretrained("openai/clip-vit-base-patch32").to(device)

    processor =
CLIPProcessor.from_pretrained("openai/clip-vit-base-patch32")

    return model, processor


def generate_embeddings(image_directory, model, processor,
device):

    image_paths = list(Path(image_directory).glob("*.jpg")) +
list(Path(image_directory).glob("*.jpeg")) +
list(Path(image_directory).glob("*.png"))

    embeddings = []

    valid_image_paths = []

    for image_path in tqdm(image_paths, desc="Generating
embeddings"):

        try:

```

```

        image = Image.open(image_path).convert("RGB")

        inputs = processor(images=image, return_tensors="pt",
padding=True).to(device)

        with torch.no_grad():

            outputs = model.get_image_features(**inputs)

            embeddings.append(outputs.cpu().numpy().flatten())

            valid_image_paths.append(image_path)

    except Exception as e:

        print(f"Error processing {image_path}: {e}")

return np.array(embeddings), valid_image_paths

def cluster_images(embeddings, similarity_threshold):

    # Calcola la similarità coseno

    cosine_sim_matrix = cosine_similarity(embeddings)

    # Converti la similarità in una matrice di distanza

    distance_matrix = 1 - cosine_sim_matrix

    # Esegui il clustering agglomerativo utilizzando la matrice di
distanza

    clustering_model = AgglomerativeClustering(n_clusters=None,
metric='precomputed', linkage='average', distance_threshold=1 -
similarity_threshold)

    labels = clustering_model.fit_predict(distance_matrix)

```

```

    return labels

def organize_images_into_clusters(labels, image_paths,
output_directory):

    cluster_to_images = defaultdict(list)

    for label, image_path in zip(labels, image_paths):

        cluster_to_images[label].append(image_path)

    for cluster_label, images in cluster_to_images.items():

        cluster_dir = output_directory /
f"cluster_{cluster_label}"

        cluster_dir.mkdir(parents=True, exist_ok=True)

        for image_path in images:

            shutil.move(str(image_path), cluster_dir /
image_path.name)

    print(f"Organized images into {len(cluster_to_images)}
clusters.")

def save_embeddings_to_csv(embeddings, image_paths, output_file):

    df = pd.DataFrame(embeddings, index=[p.name for p in
image_paths])

    df.to_csv(output_file, index_label="filename")

    print(f"Embeddings saved to {output_file}")

```

```
def main(image_directory="img", output_directory="output",
similarity_threshold=0.8):

    device = "cuda" if torch.cuda.is_available() else "cpu"

    model, processor = load_clip_model(device)

    embeddings, valid_image_paths =
generate_embeddings(image_directory, model, processor, device)

    # Normalizza gli embeddings

    scaler = StandardScaler()

    embeddings = scaler.fit_transform(embeddings)

    # Save embeddings to CSV

    output_file = Path(image_directory) / "embeddings.csv"

    save_embeddings_to_csv(embeddings, valid_image_paths,
output_file)

    labels = cluster_images(embeddings, similarity_threshold)

    output_directory = Path(output_directory)

    output_directory.mkdir(parents=True, exist_ok=True)

    organize_images_into_clusters(labels, valid_image_paths,
output_directory)
```

```

if __name__ == "__main__":

    import argparse

    parser = argparse.ArgumentParser(description="Cluster images
based on conceptual similarity using CLIP embeddings.")

    parser.add_argument("--image_directory", type=str,
required=True, help="Path to the directory containing images.")

    parser.add_argument("--output_directory", type=str,
default="output", help="Path to the directory to save clustered
images.")

    parser.add_argument("--similarity_threshold", type=float,
default=0.8, help="Threshold for clustering based on cosine
similarity. Lower values create more clusters.")

    args = parser.parse_args()

    main(args.image_directory, args.output_directory, args.similarity_threshold)

```

il codice è scaricabile anche da questo indirizzo: <https://www.paolodegasperis.it/clip.zip>

E' possibile utilizzare un modello multimodale come GPT-4 o GPT-4o per calcolare la somiglianza di un insieme, anche molto esteso, di immagini utilizzando gli embeddings estratti tramite il modello preaddestrato CLIP.

Ad esempio:

caricando il file CSV (embeddgins.csv) generato tramite lo script sopra condiviso, è possibile chiedere al modello GPT di calcolare le prime 5 coppie di immagini simili.

Prompt: Calcola le prime 5 coppie di immagini più simili analizzando i vettori di embeddings forniti nel CSV

Risposta:

Le prime 5 coppie di immagini più simili nel nuovo set di embeddings sono:

- 'Danish-Melchior_Lorck-1576-12.jpg' e 'Danish-Melchior_Lorck-1576-37.jpg' con una distanza di 15.90.



- 'Danish-Melchior_Lorck-1576-37.jpg' e 'Danish-Melchior_Lorck-1581-01.jpg' con una distanza di 15.98.



- 'German-Lucas_Cranach_dÆ-1512-01.jpg' e 'Italian-Bartolomeo_di_Zanobi_di_Benedetto_Ghetti-1518-01.jpg' con una distanza di 16.20.



- 'Henry_Moore_Cut_0006.jpg' e 'Henry_Moore_Cut_0058.jpg' con una distanza di 16.57.



- 'Danish-Melchior_Lorck-1576-10.jpg' e 'Danish-Melchior_Lorck-1576-38.jpg' con una distanza di 17.15.



Questi risultati mostrano le coppie di immagini con le distanze euclidee più basse tra i loro embeddings, indicando una notevole somiglianza visiva tra di loro nel nuovo set di dati fornito.

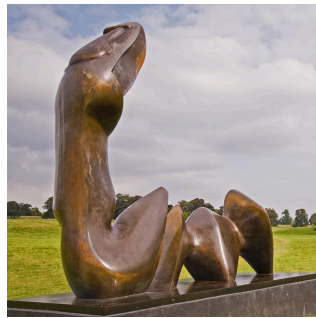
Nota: La risposta è basata sugli embeddings delle immagini di esempio fornite nell'archivio zip e la parola "distanza" si riferisce alla distanza euclidea tra i vettori di embeddings delle immagini. Questa misura calcola la distanza "lineare" tra due punti in uno spazio multidimensionale e dà un'idea della somiglianza visiva o di contenuto tra le immagini basata sui loro attributi codificati nei vettori.

Utilizzando la similarità coseno, ecco le prime 5 coppie di immagini più simili basate sui loro embeddings. Questi risultati indicano le coppie di immagini che hanno l'orientamento più simile nei loro vettori di attributi, mostrando una grande somiglianza visiva o di contenuto basata sui dati di embeddings, normalizzata tra 0 e 1, dove 1 indica la massima similarità.

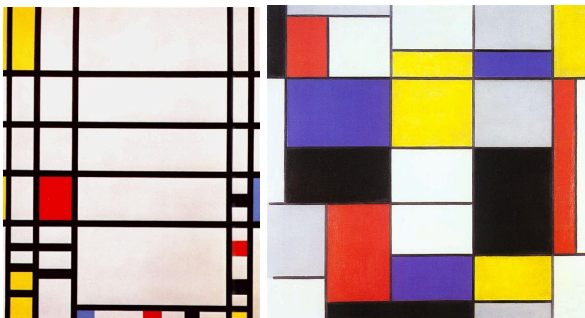
- 'Henry_Moore_Cut_0006.jpg' e 'Henry_Moore_Cut_0058.jpg' con una similarità di 0.797.



- 'Henry_Moore_Cut_0006.jpg' e 'Henry_Moore_Cut_0070.jpg' con una similarità di 0.774.



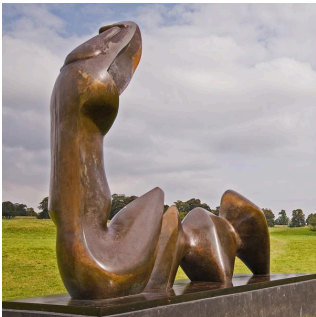
- '201.jpg' e '6.jpg' con una similarità di 0.774.



- 'Henry_Moore_Cut_0009.jpg' e 'Henry_Moore_Cut_0070.jpg' con una similarità di 0.774.



- 'Henry_Moore_Cut_0058.jpg' e 'Henry_Moore_Cut_0070.jpg' con una similarità di 0.711.



Analisi linguistica per testi antichi: Greco e Latino

GPT-4 e GPT-4o hanno capacità avanzate di comprensione e generazione di testo che si estendono anche a lingue antiche come il latino. Questi modelli possono essere utilizzati per l'interpretazione di testi latini, compresi manoscritti e epigrafi. Di seguito sono presentati alcuni prompt di esempio per l'analisi di testi in latino, che possono essere utili per un approccio didattico alle epigrafi.

Analisi Grammaticale di testi latini

Prompt: Esegui un'analisi grammaticale del seguente testo latino, identificando parti del discorso, coniugazioni verbali e declinazioni nominali: "[testo latino]".

Esempio: Esegui un'analisi grammaticale del seguente testo latino: "Gallia est omnis divisa in partes tres."

Risposta Attesa:

- **Gallia:** nominativo singolare, sostantivo femminile, "la Gallia"

- **est**: terza persona singolare del verbo essere, "è"
- **omnis**: nominativo singolare, aggettivo, "tutta"
- **divisa**: participio passato, "divisa"
- **in**: preposizione, "in"
- **partes**: accusativo plurale, sostantivo femminile, "parti"
- **tres**: accusativo plurale, aggettivo, "tre"

Interpretazione di Epigrafi Latine

Prompt: Interpreta la seguente epigrafe latina, fornendo una traduzione e un'analisi del contesto storico: "[testo epigrafe]".

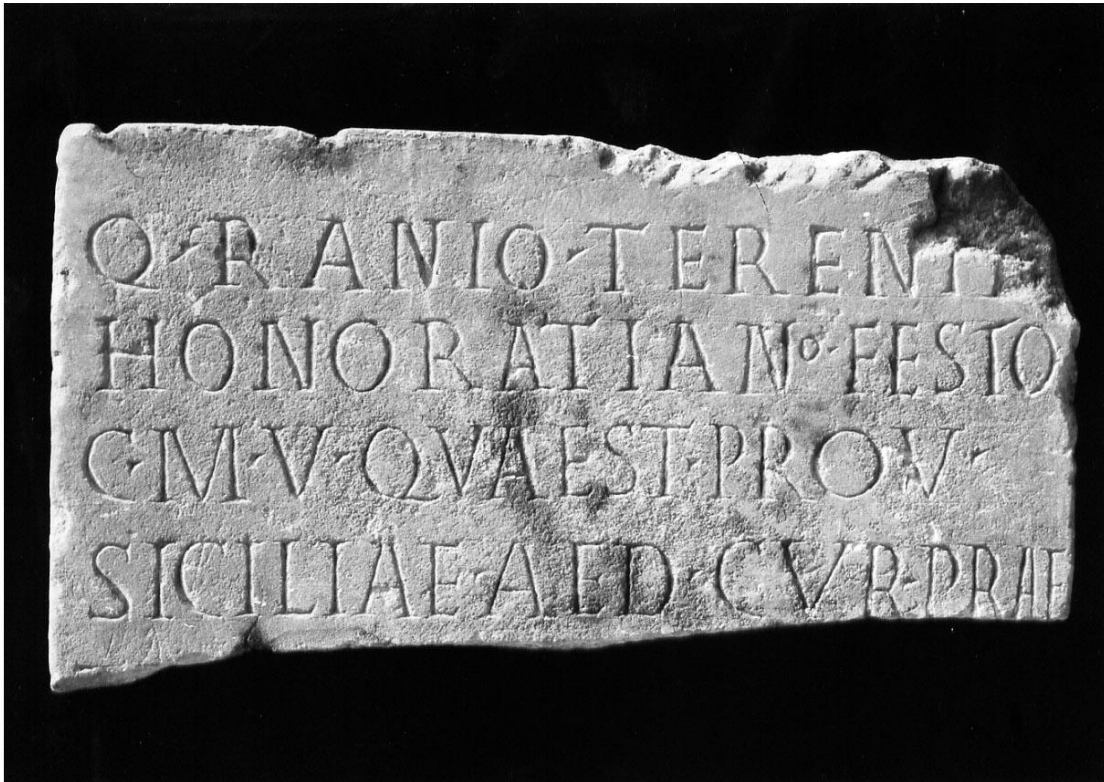
Esempio: Interpreta la seguente epigrafe latina: "D.M. Lucius Cornelius Scipio Barbatus consul."

Risposta Attesa: Traduzione: "Agli dei Mani. Lucio Cornelio Scipione Barbato, console."

Analisi: Questa epigrafe commemorativa onora Lucio Cornelio Scipione Barbato, un importante politico romano e console. Gli dei Mani erano spiriti dei defunti venerati nell'antica Roma, e l'epigrafe suggerisce un'alta posizione sociale e politica.

GPT-4 e GPT-4o sono modelli multimodali, questo ci permette di caricare un'immagine per poi interrogare il modello sull'immagine stessa.

Nel caso delle epigrafi in latino, è possibile in alcuni casi caricare la foto dell'iscrizione e chiedere una traduzione e un'interpretazione del possibile contesto storico.



iscrizione onoraria- URBINO, post 193 d.C - ante 217 d.C

<https://catalogo.cultura.gov.it/detail/ArchaeologicalProperty/1100015141>

Prompt: Interpreta questa epigrafe, traducila in italiano e fornisci un contesto storico

Risposta:

Testo dell'Epigrafe

L'epigrafe riportata nell'immagine presenta la seguente iscrizione:

Q. RANIO TEREN

HONORATIANO FESTO

CMV QVAEST PROV

SICILIAE AED CVR PRAEF

Traduzione in Italiano

A Quinto Ranio Terenzio

Honoraziano Festo

Che fu questore della provincia

Di Sicilia, edile curule e prefetto.

Interpretazione e Contesto Storico

Q. RANIO TEREN: Abbreviazione per "Quintus Ranius Terentius". Quinto era un prenome romano comune, Ranius è il nomen (nome della gens o famiglia), e Terentius potrebbe essere un cognomen aggiunto per indicare una specifica linea familiare o una caratteristica particolare.

HONORATIANO FESTO: Probabilmente un aggettivo onorifico che si riferisce a Festo. "Honoratianus" potrebbe essere un aggettivo derivato da "honor", significando qualcosa come "onorato" o "onorifico".

CMV QVAEST PROV: Abbreviazione per "Cum Questore Provinciae". Indica che Quinto Ranio Terenzio Festo ha ricoperto la carica di questore di una provincia, in questo caso, Sicilia.

SICILIAE AED CVR PRAEF: Abbreviazione per "Siciliae Aedilis Curulis Praefectus". Indica che ha ricoperto la carica di edile curule e prefetto in Sicilia. L'edile curule era un magistrato responsabile della manutenzione di edifici pubblici, strade, e altri aspetti dell'amministrazione urbana, mentre il prefetto poteva avere diversi ruoli, spesso legati all'amministrazione e alla giustizia.

Contesto Storico

L'iscrizione onora Quinto Ranio Terenzio Festo, che ha ricoperto diverse cariche pubbliche importanti nella provincia romana di Sicilia.

Questore Provinciale: Nella struttura amministrativa romana, il questore era un funzionario responsabile delle finanze. I questori provinciali erano inviati nelle province per gestire le finanze pubbliche, raccogliere tasse e amministrare la giustizia finanziaria.

Edile Curule: Gli edili curuli erano magistrati superiori rispetto agli edili plebei e avevano compiti più vasti, inclusa la supervisione delle feste pubbliche e dei giochi, la manutenzione dei templi, delle strade, e delle altre infrastrutture pubbliche.

Prefetto: Il ruolo di prefetto variava, ma spesso includeva compiti amministrativi e giudiziari. Nelle province, i prefetti potevano essere incaricati di compiti specifici come la supervisione delle forniture di grano o la gestione di una città.

Provincia di Sicilia: Sicilia era una delle prime province romane, acquisita durante le guerre puniche contro Cartagine. Era una provincia agricola importante, nota per la produzione di grano, che era vitale per l'approvvigionamento di Roma.

Conclusione

Questa epigrafe celebra Quinto Ranio Terenzio Festo, un funzionario romano che ha servito come questore, edile curule e prefetto nella provincia di Sicilia. L'iscrizione riflette l'importanza del ruolo amministrativo e giudiziario di Festo in una delle province chiave dell'Impero Romano, sottolineando il suo contributo al mantenimento e alla gestione delle strutture pubbliche e delle finanze.



lapide tombale - ambito romano (sec. IV)

<https://catalogo.beniculturali.it/detail/HistoricOrArtisticProperty/1200237645E>

L'immagine mostra un gruppo di epigrafi, ognuna delle quali contiene iscrizioni in latino o greco. Ecco l'interpretazione e la traduzione di alcune delle epigrafi visibili:

Epigrafe 1 (In alto a sinistra)

Testo:

ARCADIUS

S QVI VIXIT

IMENSX DIE

VII IN PACE

Traduzione:

Arcadius

Che visse 1 mese e 7 giorni

Riposa in pace

Epigrafe 2 (In alto al centro)

Testo:

AVR X X X

SEPTIMIAE

PIETATI

CONIVGI

CARISSIMAE

Traduzione:

Aurelius

A Septimia

Alla sua cara moglie

Epigrafe 3 (In alto a destra)

Testo:

S V T I PETE

PRONOS

VIS ALVIS SIMVS

Traduzione:

S V T I PETE

PRONOS

VIS ALVIS SIMVS

Nota: Questa iscrizione è meno chiara e potrebbe necessitare di ulteriori indagini per una traduzione accurata.

Epigrafe 4 (In basso a sinistra)

Testo:

ΑΥΡ' ΑΓΙΑΝΟΣ

ΠΑΦΛΑΓΩΝ ΘΕ

ΟΥΔΟΥΛΟΣ ΠΙΕΤΟ

ΕΚΡΟΙ ΜΗΘΕΝ ΕΙΡΗ

ΝΗ ΜΝΗΘΗΑΥΤΟΥ

ΘΕΟΣ ΕΙΣ ΤΟΥΣ ΑΙΩΝΑΣ

Traduzione:

Aurelius Agianos

Di Paflagonia

Servo di Dio

Riposa in pace

Che Dio lo ricordi

Per l'eternità

Epigrafe 5 (In basso a destra)

Testo:

INNOCENTISSIMAEETATI

INDVICIAMDIDICISIHI IN MARCHIA

NONAMENTENE VIBIA PAISEN

PISTA S 2 ANN XX MES

Traduzione:

Alla più innocente età

Per l'invidia appresa in marcia

Non mentendo Vibia Paise

Fedeltà 2 anni 20 mesi

Nota: Questa iscrizione è parzialmente incompleta e potrebbe richiedere ulteriori indagini per una traduzione più precisa.

Contesto Storico

Le epigrafi funerarie romane e greche spesso forniscono dettagli sulle vite dei defunti, comprese le loro età, titoli, relazioni familiari e auguri di pace eterna. Queste iscrizioni offrono uno sguardo intimo sulle relazioni personali e sulle pratiche funerarie dell'antica Roma e Grecia.

1. **Epigrafe di Arcadius:** Indica un bambino di nome Arcadius che visse solo un mese e sette giorni. Questo tipo di iscrizione era comune per commemorare brevemente le vite dei bambini.
2. **Epigrafe di Septimia:** Dedicata da Aurelius alla sua cara moglie Septimia, riflette l'affetto e il rispetto per i coniugi defunti.
3. **Epigrafe di Agianos:** In greco, commemorando Aurelius Agianos di Paflagonia, servo di Dio. Riflette una devozione religiosa e l'augurio di memoria eterna.
4. **Epigrafe di Vibia Paise:** Rende omaggio a Vibia Paise, menzionando la fedeltà e un periodo di 2 anni e 20 mesi.

Conclusione

Queste epigrafi offrono un'idea delle pratiche commemorative e delle relazioni personali nell'antichità. Sono testimonianze storiche che rivelano l'importanza della memoria e del rispetto per i defunti in culture che valorizzavano profondamente la famiglia e la religione.

Considerazioni

I modelli dietro l'interfaccia di ChatGPT 4 e ChatGPT 4o sono software focalizzati sul risultato e sulla pubblicazione di un output, sono sistemi che raramente reagiscono all'utente con una risposta vuota (al netto dei limiti imposti per contenuti sensibili).

Questo comporta un rischio reale di risposte che sono non pertinenti, inaccurate, o completamente slegate dalla realtà o dal contesto della domanda, in alcuni casi vengono prodotte risposte verosimili ma non aderenti alla realtà. Questo è ancora più evidente

nell'attività precedente di interpretazione delle epigrafi in cui un elemento mancante e non visibile può essere "compensato" dal software introducendo però una traduzione e un contenuto non veritiero e diverso dal testo dell'epigrafe stessa. Diventa così fondamentale il controllo umano sul risultato per verificare una corretta aderenza al contenuto proposto.