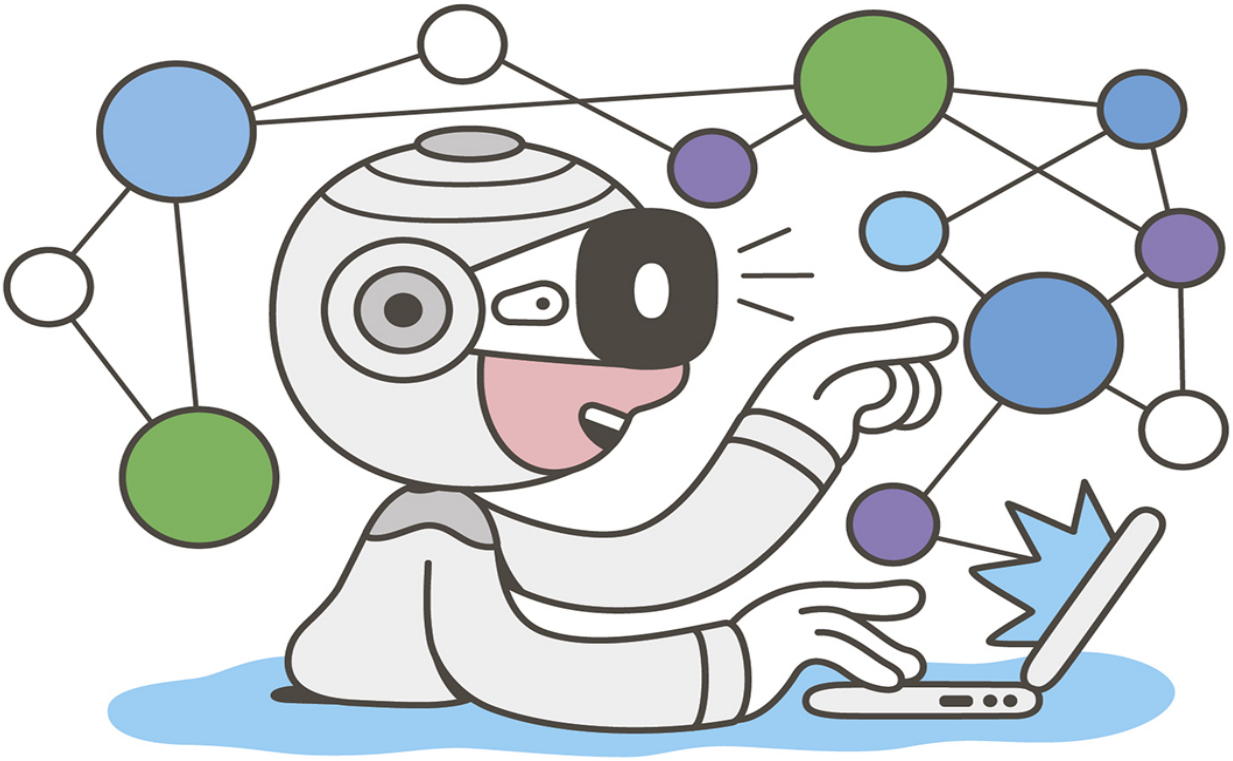
VALENTINA ALTO

Intelligenza Artificiale
in pratica

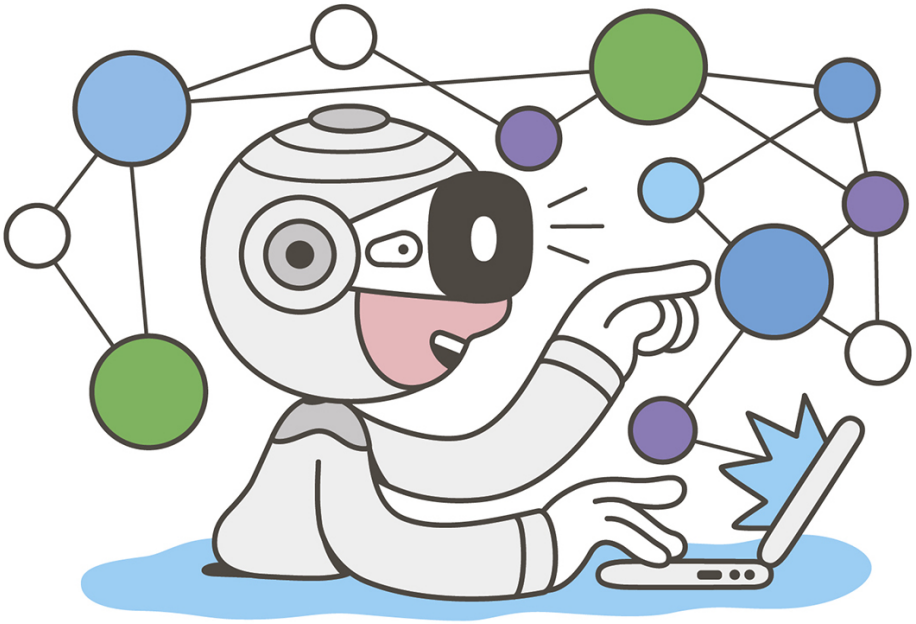


Diventare maestri nell'utilizzo dei modelli OpenAI

APOGEO

VALENTINA ALTO

Intelligenza Artificiale
in pratica



Diventare maestri nell'utilizzo dei modelli OpenAI

APOGEO

Intelligenza Artificiale in pratica
Diventare maestri nell'utilizzo dei modelli OpenAI

Valentina Alto

APOGEO

© Apogeo - IF - Idee editoriali Feltrinelli s.r.l.
Socio Unico Giangiacomo Feltrinelli Editore s.r.l.

ISBN ebook: 9788850338153

Copyright (c) Packt Publishing 2023. First published in the English language under the title 'Modern Generative AI with ChatGPT and OpenAI Models' – (9781805123330).

Il presente file può essere usato esclusivamente per finalità di carattere personale. Tutti i contenuti sono protetti dalla Legge sul diritto d’autore.

Nomi e marchi citati nel testo sono generalmente depositati o registrati dalle rispettive case produttrici.

L’edizione cartacea è in vendita nelle migliori librerie.

~

Sito web: www.apogeonline.com

Scopri le novità di Apogeo su Facebook

Seguici su X

Collegati con noi su LinkedIn

Guarda cosa stiamo facendo su Instagram

Rimani aggiornato iscrivendoti alla nostra newsletter

Introduzione

Questo libro inizia presentando l’ambito dell’intelligenza artificiale (AI) generativa, un campo focalizzato sulla generazione di dati o contenuti unici e originali attraverso algoritmi di apprendimento automatico. Esplora i fondamenti dei modelli di AI generativa e illustra come vengono addestrati per produrre nuovi dati.

Prosegue mettendo in luce casi d’uso reali, nei quali ChatGPT può incrementare la produttività e stimolare la creatività. Il libro spiega anche come ottenere il massimo dalle interazioni con ChatGPT, perfezionando la struttura del prompt e sfruttando le potenzialità di apprendimento zero-, one-, e few-shot.

Il libro continua con un’analisi più minuziosa dei casi d’uso precedentemente trattati, suddivisi per settore: marketing, ricerca e sviluppo/dati scientifici. Ogni settore viene illustrato attraverso quattro casi d’uso reali che potrete replicare facilmente in autonomia.

Passando dai singoli casi d’uso a scenari di livello aziendale, il libro introduce l’uso delle API dei modelli OpenAI disponibili su Azure. Si focalizza anche su scenari end-to-end di storie di clienti reali e sulle implicazioni dell’AI responsabile.

Per concludere, il libro riassume i punti chiave discussi e riflette sul futuro dell’AI generativa.

Lo scopo del libro è quello di farvi acquisire le competenze necessarie per approfondire l’AI generativa e per iniziare a utilizzare ChatGPT e le API dei modelli OpenAI nei vostri progetti.

A chi si rivolge questo libro

Questo libro è rivolto a un vasto pubblico, a ogni utente desideroso di migliorare la propria produttività quotidiana grazie a ChatGPT e OpenAI e di approfondire la tecnologia e l'architettura che alimentano ChatGPT. Il libro è destinato anche al mondo aziendale, offrendo un approfondimento sulle applicazioni pratiche dei modelli ChatGPT e OpenAI e spiegando come possano essere utilizzati per migliorare il lavoro e le organizzazioni.

Gli sviluppatori e i data scientist troveranno utili spunti su come ChatGPT e i modelli OpenAI possono arricchire i loro modelli di machine learning e il loro codice. Marketer e ricercatori, invece, avranno l'opportunità di esplorare i vari casi d'uso di OpenAI e ChatGPT nel loro settore.

Il libro offre un'analisi teorica delle tecnologie che sono alla base dei modelli OpenAI e dei concetti fondamentali dell'AI generativa. Pertanto, non richiede particolari competenze. Per coloro che sono interessati anche all'implementazione pratica, il libro fornisce soluzioni complete e listati in Python, oltre a guide dettagliate passo-passo.

Di che cosa tratta il libro

Il Capitolo 1 presenta una panoramica del campo dell'AI generativa, che si concentra sulla creazione di dati o contenuti nuovi e unici utilizzando algoritmi di machine learning. Copre le basi dei modelli di AI generativa e spiega come questi modelli sono addestrati per generare nuovi dati. Il capitolo si concentra anche sulle applicazioni dell'AI generativa a vari campi, come la sintesi di immagini, la generazione di testo e la composizione musicale, evidenziando il potenziale dell'AI generativa nel rivoluzionare vari settori.

Il Capitolo 2 offre una panoramica di OpenAI e del suo sviluppo più noto, ChatGPT, evidenziando la sua storia, la sua tecnologia e le sue capacità. Il capitolo si concentra anche su come ChatGPT può essere utilizzato in vari campi e applicazioni per migliorare la comunicazione e automatizzare i processi e, infine, su come può influenzare il mondo della tecnologia.

Il Capitolo 3 è una guida all'impostazione di un account ChatGPT e per iniziare a utilizzare il servizio. Spiega anche come interagire con l'interfaccia utente web, come organizzare le chat in base al tema e come strutturare una conversazione.

Il Capitolo 4 si concentra sull'importanza della struttura del prompt come tecnica per migliorare l'accuratezza del modello. Infatti, i prompt influenzano pesantemente l'output generato dal modello. Un prompt ben strutturato può aiutare a guidare il modello verso la generazione di output pertinenti e accurati, mentre un prompt mal strutturato può risultare irrilevante o confuso. Infine, è importante implementare considerazioni etiche nel prompt, per evitare che il modello generi contenuti dannosi.

Il Capitolo 5 esamina le principali attività che ChatGPT può fare per gli utenti generali su base giornaliera, aumentando la loro produttività. Il capitolo si concentra su esempi concreti di assistenza alla scrittura, *decision-making*, ispirazione creativa e così via, permettendovi di provarli da soli.

Il Capitolo 6 si concentra su come gli sviluppatori possono fare uso di ChatGPT. Il capitolo tratta i principali casi d'uso che ChatGPT può affrontare in questo dominio, tra cui la revisione e l'ottimizzazione del codice, la generazione di documentazione e la generazione di codice. Questo capitolo fornisce esempi e vi permette di provare i prompt da soli.

Il Capitolo 7 mostra come i marketer possono fare uso di ChatGPT. Il capitolo si concentra sui principali casi d'uso che ChatGPT può affrontare in questo dominio, tra cui il testing A/B, i suggerimenti di targeting delle parole chiave e l'analisi del sentiment dei social media. Anche questo capitolo fornisce vari esempi.

Il Capitolo 8 si concentra su come i ricercatori possono fare uso di ChatGPT. Il capitolo tratta i principali casi d'uso che ChatGPT può affrontare in questo dominio, tra cui la recensione della letteratura, la progettazione di esperimenti e la generazione di bibliografie. Anche questo capitolo fornisce vari esempi.

Il Capitolo 9 si concentra sulle applicazioni a livello aziendale dei modelli OpenAI, introducendo il servizio Azure OpenAI. Trovate una panoramica delle API del modello del servizio Azure OpenAI e di come possono essere incorporate nel codice. Mi concentrerò anche sui parametri del modello, sulla configurazione e sul fine-tuning. Infine, vi proporrò alcune considerazioni sul tema dell'AI responsabile e su come assicurarsi che il vostro sistema di AI sia conforme agli standard etici.

Il Capitolo 10 inizia con una panoramica dei casi d'uso più attuali con Azure OpenAI che le imprese stanno sviluppando sul mercato, oggi. Esploreremo esempi concreti di progetti, come un motore di ricerca intelligente, un assistente AI e un generatore di report. Infine, ci concentreremo su un progetto di produzione end-to-end in un settore specifico.

Il Capitolo 11 inizia con un breve riassunto dei casi d'uso più attuali esaminati nei capitoli precedenti. Passeremo poi ad alcune considerazioni sulle implicazioni dell'AI generativa nelle industrie e sulla vita quotidiana. Affronteremo anche considerazioni etiche sul ruolo dell'AI responsabile nella fase di progettazione del progetto. Il

capitolo si conclude con alcune riflessioni finali sul futuro dell'AI generativa in termini di nuovi sviluppi, tendenze e modelli.

File degli esempi e immagini a colori

Potete scaricare i file di codice di esempio e le immagini a colori di questo libro dal sito di Apogeo all'indirizzo https://bit.ly/apo-iap.

L'autrice

Valentina Alto, laurea in Data Science e autrice di numerosi articoli sul machine learning e il deep learning, lavora in Microsoft come specialista di soluzioni Azure che implementano tecnologie di AI.

Il revisore

Supreet Kaur è AI product manager presso Morgan Stanley, dove ricopre il ruolo di product owner per diversi prodotti AI. Ha lavorato come consulente technology e data science per clienti del settore farmaceutico. Scrittrice e relatrice esperta di data science e AI, ha tenuto oltre 50 conferenze in forum internazionali. Selezionata come Google Women Techmakers Ambassador, è riconosciuta come una delle 25 donne più importanti nel campo dell'AI nella finanza.

Parte I

I fondamenti dell’AI generativa e dei modelli GPT

Questa parte inizia con una panoramica del dominio dell’AI generativa, fornendovi le conoscenze di base su quest’area di ricerca dell’AI, comprendendo la sua storia e gli sviluppi più all’avanguardia.

Avrete anche l’opportunità di familiarizzare con le applicazioni dell’AI generativa, che vanno dalla generazione di testi alla composizione musicale.

La Parte 1 introduce poi l’azienda che ha portato il potere dell’AI generativa al grande pubblico: OpenAI. Avrete modo di familiarizzare con la tecnologia dietro la release più popolare di OpenAI, ChatGPT, e di comprendere il percorso di ricerca che, partendo dalle reti neurali artificiali (ANN – *Artificial Neural Network*), ha portato ai LLM (*Large Language Model*).

Capitolo 1

L’IA generativa

In questo libro esploreremo l’affascinante mondo dell’*intelligenza artificiale* (AI) generativa e delle sue rivoluzionarie applicazioni. L’AI generativa ha trasformato il modo in cui interagiamo con le applicazioni, consentendo ai computer di creare, prevedere e apprendere senza istruzioni umane esplicite. Con ChatGPT e OpenAI, abbiamo assistito a progressi senza precedenti nell’elaborazione del linguaggio naturale, nella sintesi di immagini e video e in molti altri campi. Questa guida si pone l’obiettivo di fornire le conoscenze e le competenze per navigare panorama dell’AI generativa.

Questo capitolo presenta una panoramica del campo dell’AI generativa, che consiste nella creazione di nuovi contenuti utilizzando algoritmi di *machine learning* (ML).

Il capitolo si concentra sulle applicazioni dell’AI generativa in vari campi, tra cui la creazione di immagini, la generazione di testo e la composizione musicale, evidenziando il potenziale dell’AI generativa in questi settori.

Questa introduzione all’AI generativa fornirà le basi per collocare correttamente questa tecnologia nel più ampio contesto dell’AI, del machine learning e del *deep learning* (DL). Successivamente, ci soffermeremo sui principali ambiti di applicazione dell’AI generativa, con esempi concreti e sviluppi recenti, in modo che possiate familiarizzare con l’impatto che può avere sulle imprese e sulla società in generale.

Inoltre, essere consapevoli del percorso di ricerca verso l'attuale stato dell'arte dell'AI generativa vi darà una migliore comprensione dei fondamenti dei recenti sviluppi e dei modelli all'avanguardia.

Nello specifico, tratteremo i seguenti argomenti:

- il campo di ricerca della AI generativa;
- i domini dell'AI generativa;
- la storia e lo stato attuale della ricerca sull'AI generativa.

Alla fine di questo capitolo, avrete acquisito familiarità con i fondamentali dell'AI generativa, le sue applicazioni, la storia della ricerca che c'è dietro e gli attuali sviluppi che stanno rivoluzionando il mondo.

Introduzione all'AI generativa

L'AI ha fatto passi da gigante negli ultimi anni, e una delle aree che ha visto una crescita considerevole è l'AI generativa. L'AI generativa è un sottoinsieme dell'AI e del deep learning che si concentra sulla generazione di nuovi contenuti, come immagini, testo, musica e video, utilizzando algoritmi e modelli che sono stati addestrati su dati esistenti utilizzando tecniche di machine learning.

Per comprendere meglio la relazione esistente tra AI, machine learning, deep learning e AI generativa, possiamo pensare all'AI come a un ampio campo di ricerca, all'interno del quale machine learning, deep learning e AI generativa rappresentano aree di studio e applicazione sempre più specializzate e focalizzate.

- L'*intelligenza artificiale* (AI) rappresenta l'ampio campo della creazione di sistemi in grado di svolgere compiti, esibendo intelligenza e capacità umane e dimostrando di essere in grado di interagire con l'ecosistema.

- Il *machine learning* (ML) è una branca dell'AI che si concentra sulla creazione di algoritmi e modelli che consentono a tali sistemi di apprendere e migliorare se stessi con il tempo e la formazione. I modelli di machine learning sono addestrati su dati esistenti e aggiornano automaticamente i loro parametri man mano che *apprendono dai loro errori*.
- Il *deep learning* (DL) è una sottobranca del machine learning, nel senso che comprende modelli di machine learning *deep*, ovvero "profondi". Questi modelli profondi sono chiamati *reti neurali* e sono particolarmente adatti in domini come la *computer vision* o l'elaborazione del linguaggio naturale (NLP - *Natural Language Processing*). Quando si parla di modelli di machine learning e deep learning, in genere ci si riferisce a modelli discriminativi, il cui scopo è quello di fare previsioni, o a modelli di inferenza sui dati.
- Infine, arriviamo all'AI generativa, un'ulteriore sottobranca del deep learning, che non utilizza reti neurali profonde per raggruppare, classificare o fare previsioni sui dati esistenti, ma utilizza quei potenti modelli di reti neurali per generare nuovi contenuti, che spaziano dalle immagini al linguaggio naturale, dalla musica ai video.

La Figura 1.1 mostra le correlazioni tra queste aree di ricerca.

I modelli di AI generativa sono addestrati su grandi quantità di dati e possono generare nuovi esempi da zero, basati sui *pattern* identificati nei dati di training.

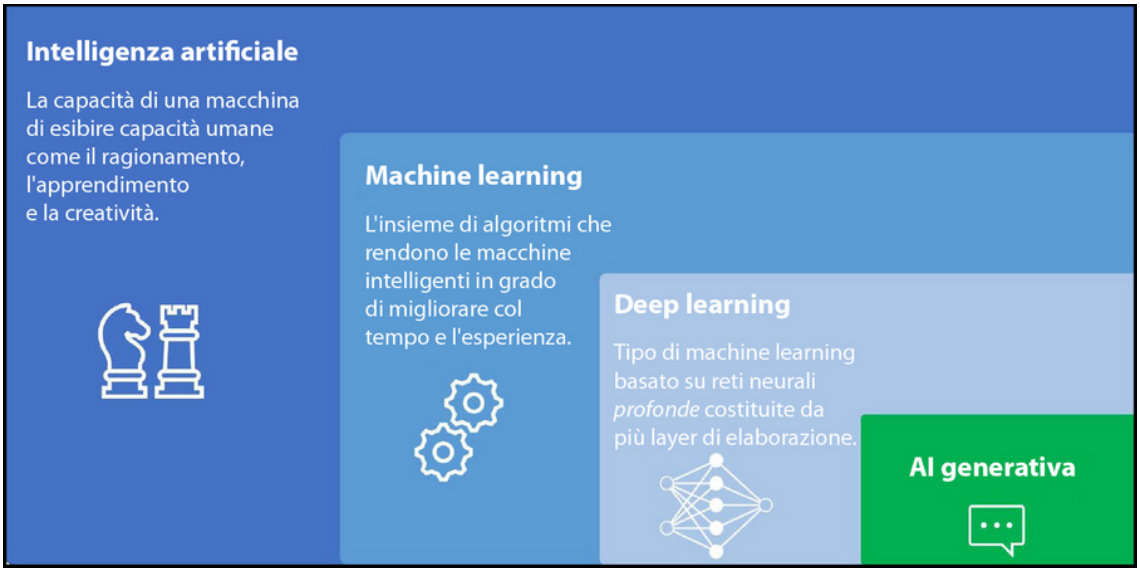


Figura 1.1 Relazione tra AI, machine learning, deep learning e AI generativa.

Tali modelli prendono il nome di LFM (*Large Foundation Model*) e, oltre alle capacità generative sopracitate, si distinguono dai sistemi di AI "tradizionali" per la loro capacità di generalizzazione e adattamento a diversi task. Per esempio, se in passato l'approccio tipico consisteva, dato un problema (per esempio, di classificazione di immagini), nel raccogliere dati relativi a quel problema e addestrare un modello specifico in quell'attività, con i Large Foundation Model il paradigma cambia: sarà l'utente a chiedere al modello, in linguaggio naturale, di eseguire un determinato compito, che potrà essere di classificazione di immagini, di summarization di articoli, di sentiment analysis…

I *Large Foundation Model* si riferiscono, in generale, a modelli che sono in grado di elaborare diverse tipologie di dati (immagini, testo, audio e così via). In questa categoria, esiste un sottoinsieme di modelli specializzati nel linguaggio, detti LLM (*Large Language Model*). Nel corso di questo libro, ci concentreremo sui Large Language Model e, in particolare, su quelli sviluppati da OpenAI.

Domini dell’AI generativa

Negli ultimi anni, l’AI generativa ha compiuto progressi significativi e ha ampliato le sue applicazioni in una vasta gamma di settori, come l’arte, la musica, la moda, l’architettura e molti altri. In alcuni di essi, sta davvero trasformando il modo in cui creiamo, progettiamo e comprendiamo il mondo che ci circonda. In altri, sta migliorando e rendendo più efficienti i processi e le operazioni esistenti.

Il fatto che l’AI generativa sia utilizzata in molti domini implica anche che i suoi modelli siano in grado di gestire diversi tipi di dati, dal linguaggio naturale all’audio o alle immagini. Nei prossimi paragrafi, approfondiremo il modo in cui i modelli di AI generativa si applichino a queste diverse tipologie di dati.

Generazione di testo

Una delle più grandi applicazioni dell’AI generativa, e quella di cui parleremo più frequentemente in questo libro, è la sua capacità di produrre nuovi contenuti in linguaggio naturale. Gli algoritmi di intelligenza artificiale generativa possono essere utilizzati per produrre nuovo testo, come articoli, poesie e descrizioni di prodotti.

Per esempio, un modello linguistico come il GPT-3.5-turbo (il “motore” dietro ChatGPT) sviluppato da OpenAI, può essere addestrato su grandi quantità di dati testuali e utilizzato per generare nuovi contenuti testuali, coerenti e grammaticalmente corretti in diverse lingue, nonché per estrarre caratteristiche rilevanti dal testo, come parole chiavi, principali argomenti o riassunti completi.

La Figura 1.2 mostra un esempio di ChatGPT in azione.

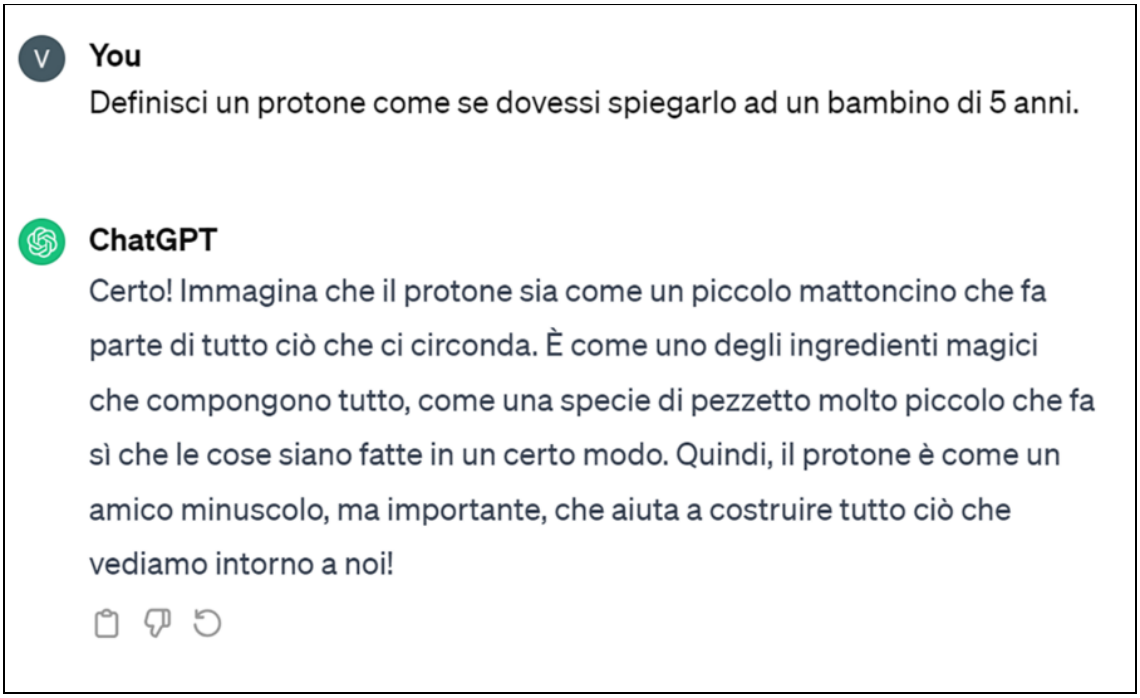


Figura 1.2 ChatGPT risponde a un prompt dell'utente, aggiungendo anche riferimenti.

Come prossimo passo, ci occuperemo della generazione di immagini.

Generazione di immagini

Uno dei primi e più noti esempi di AI generativa nella sintesi delle immagini è l'architettura GAN (*Generative Adversarial Network*) introdotta nell'articolo del 2014 di I. Goodfellow *et al.*, *Generative Adversarial Networks*. Lo scopo delle GAN è quello di generare immagini realistiche, indistinguibili dalle immagini reali. Questa funzionalità ha avuto diverse applicazioni aziendali interessanti, come la generazione di:

- set di dati sintetici per l'addestramento di modelli di visione artificiale;
- immagini sufficientemente realistiche di prodotti;

- elementi visuali (come individui, paesaggi, oggetti) per applicazioni di realtà virtuale e realtà aumentata.

La Figura 1.3 mostra un esempio di volti di persone (inesistenti) interamente generati dall'AI.

Figura 1.3 Volti immaginari generati da GAN StyleGAN2 (https://this-person-does-not-exist.com/en).

Nel 2021, OpenAI ha introdotto in questo campo un nuovo modello di AI generativa, *DALL-E*. A differenza delle GAN (che prendono come input un vettore di *casual noise*), il modello DALL-E è progettato per generare immagini da descrizioni in linguaggio naturale e può produrre un'ampia gamma di illustrazioni, con dovizia di particolari nello stile richiesto.

DALL-E ha un grande potenziale nei settori creativi come la pubblicità, il design di prodotti e la moda, tra gli altri, per creare immagini uniche e creative.

Nella Figura 1.4 trovate un esempio di DALL-E 3 (l'ultima versione rilasciata da OpenAI nell'ottobre del 2023) che genera quattro

immagini a partire da una richiesta in linguaggio naturale.

Notate che la generazione di testo e immagini può essere combinata per produrre contenuti originali. Negli ultimi anni, nuovi strumenti di intelligenza artificiale diffusi hanno utilizzato questa combinazione.

Un esempio è *Tome AI*, un'applicazione di AI generativa che, tra le sue capacità, è in grado di creare presentazioni da zero, sfruttando modelli come DALL-E e GPT-3.5. Nell'esempio rappresentato nella Figura 1.5, ho chiesto a Tome di generare una presentazione di quattro slide sulla *Generative AI*, e questo è il risultato.

Come potete vedere, questo strumento di intelligenza artificiale è stato perfettamente in grado di creare una bozza di presentazione solo sulla base del mio breve input in linguaggio naturale.

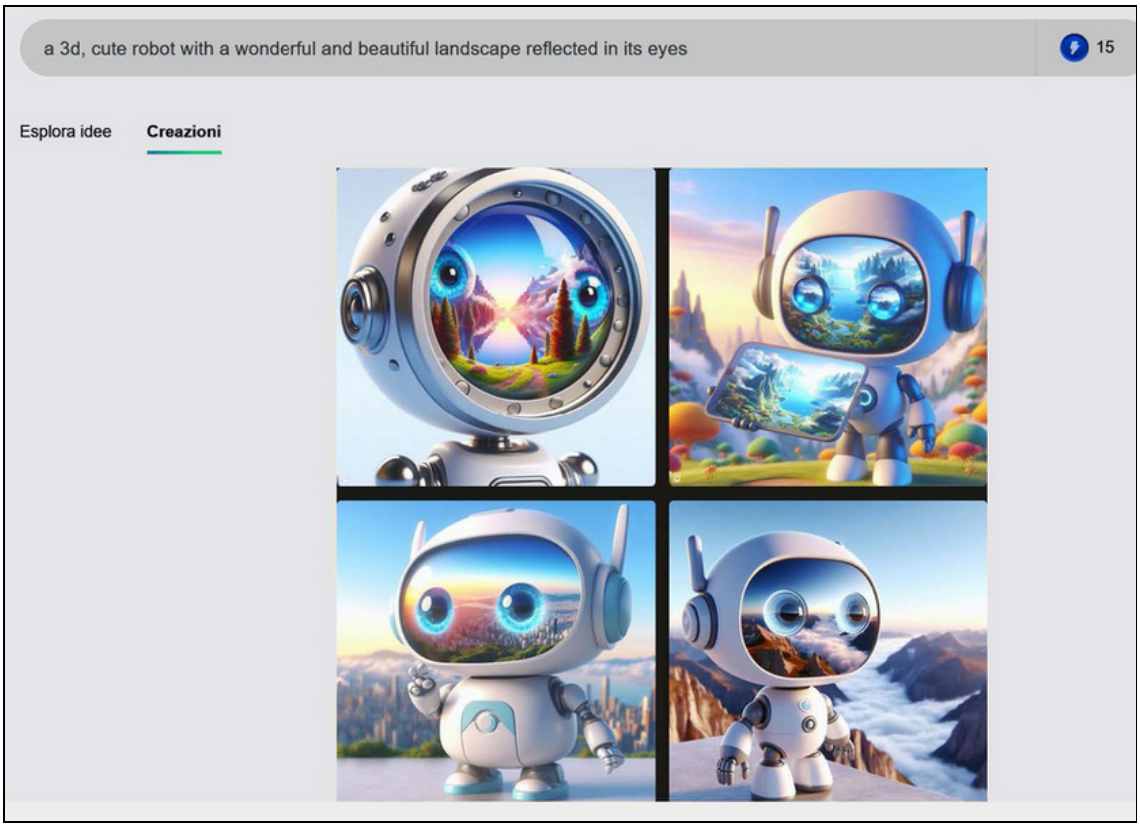


Figura 1.4 Immagini generate da DALL-E 3 in Bing Image Creator, partendo da un prompt in linguaggio naturale come input.

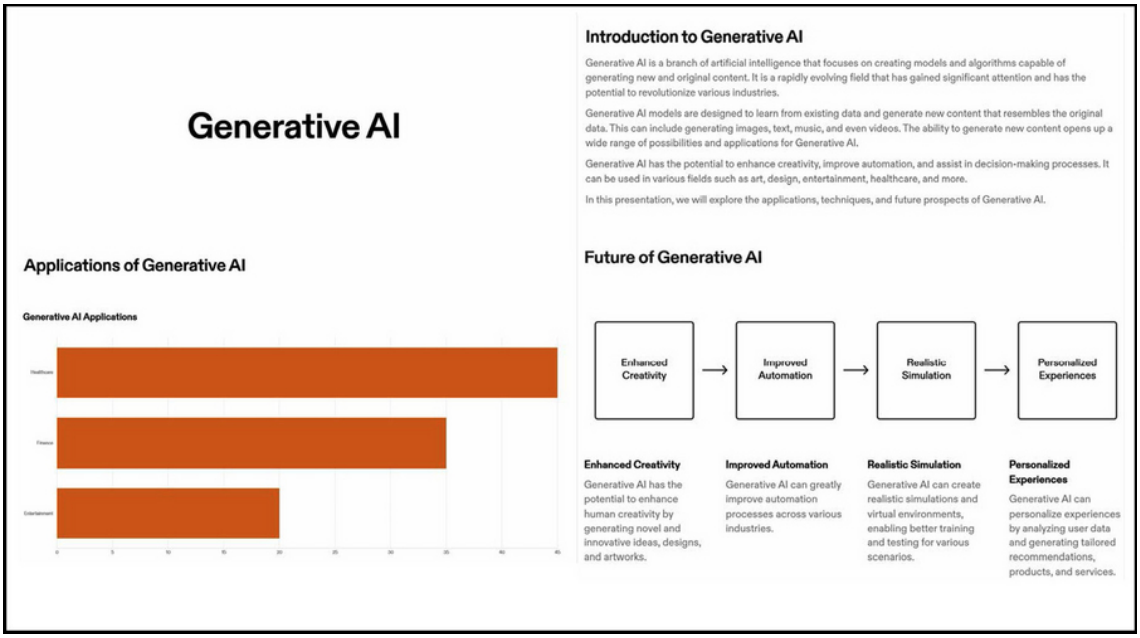


Figura 1.5 Una presentazione sull'AI generativa interamente generata da Tome, utilizzando un input in linguaggio naturale.

Generazione di musica

I primi approcci all'AI generativa per la generazione musicale risalgono agli anni Cinquanta, con ricerche nel campo della composizione algoritmica, una tecnica che utilizza algoritmi per generare composizioni musicali. Nel 1957, Lejaren Hiller e Leonard Isaacson crearono la *Illiac Suite for String Quartet* (https://www.youtube.com/watch?v=n0njBFLQSk8), il primo brano musicale interamente composto da un'AI. Da allora, il campo dell'AI generativa per la musica è stato oggetto di ricerca continua per diversi decenni. Tra gli sviluppi degli ultimi anni, si sono diffuse tra il grande pubblico nuove architetture e framework, come l'architettura *WaveNet*, introdotta da Google nel 2016, che è stata in grado di generare campioni audio di alta qualità, o il progetto *Magenta*, anch'esso sviluppato da Google, che utilizza RNN (*Recurrent Neural Networks*,

ovvero “reti neurali ricorrenti”) e altre tecniche di machine learning per generare musica e altre forme d’arte. Poi, nel 2020, OpenAI ha annunciato *Jukebox*, una rete neurale che genera musica, con la possibilità di personalizzare l’output in termini di stile musicale e vocale, genere, artista di riferimento e così via.

Questi e altri framework sono diventati le fondamenta di molti assistenti di AI di compositori per la generazione musicale. Un esempio è *Flow Machines*, sviluppato da Sony CSL Research. Questo sistema di intelligenza artificiale generativa è stato addestrato su un ampio database di brani musicali per creare nuova musica in una varietà di stili. È stato utilizzato dal compositore francese Benoît Carré per comporre un album intitolato *Hello World* (https://www.helloworldalbum.net), che presenta collaborazioni con diversi musicisti (umani).

Nella Figura 1.6 potete vedere un esempio di traccia generata interamente da *Music Transformer*, uno dei modelli del progetto *Magenta*.

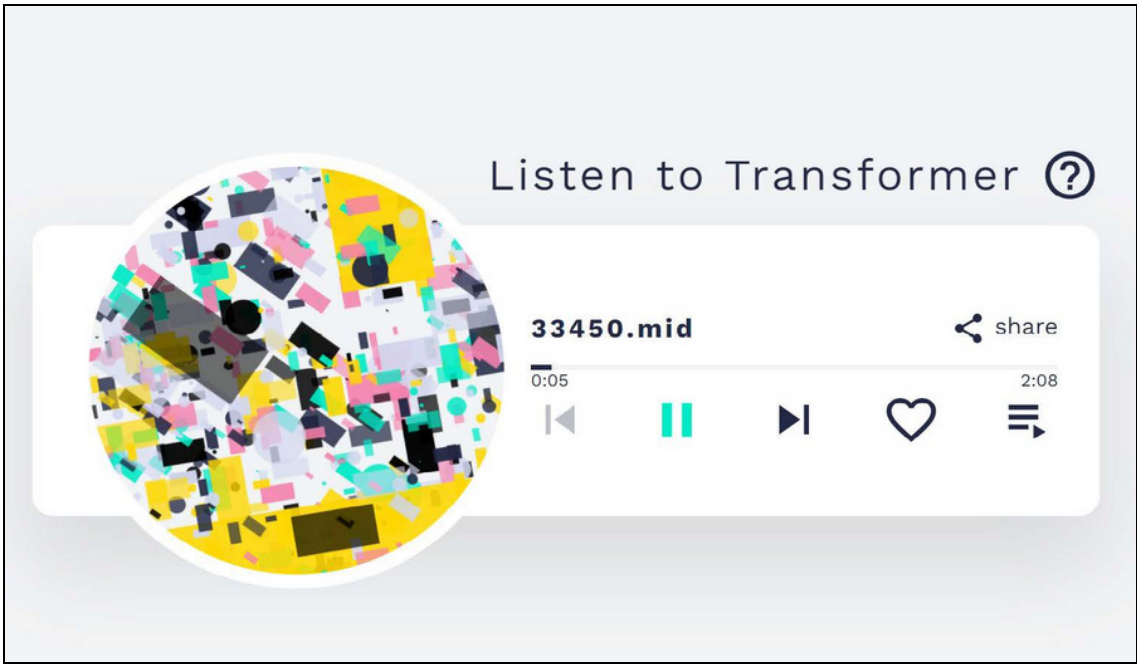


Figura 1.6 Music Transformer consente all'utente di ascoltare le performance musicali generate dall'AI.

Un'altra incredibile applicazione dell'AI generativa nel campo della musica è la sintesi vocale. È infatti possibile trovare molti strumenti di intelligenza artificiale in grado di creare audio basati su input di testo nelle voci di cantanti famosi.

Per esempio, se vi siete sempre chiesti come suonerebbero le vostre canzoni se le eseguisse Kanye West, ora potete realizzare i vostri sogni con strumenti come *FakeYou.com* (`https://fakeyou.com`), *Deep Fake Text to Speech* o *UberDuck.ai* (`https://uberduck.ai`), vedi la Figura 1.7.

Devo dire che il risultato è davvero impressionante. Se volete divertirvi, potete anche provare le voci di tutti i vostri cartoni animati preferiti, come *Winnie The Pooh*...

Nel prossimo paragrafo, approfondiremo la generazione di video tramite AI generativa.

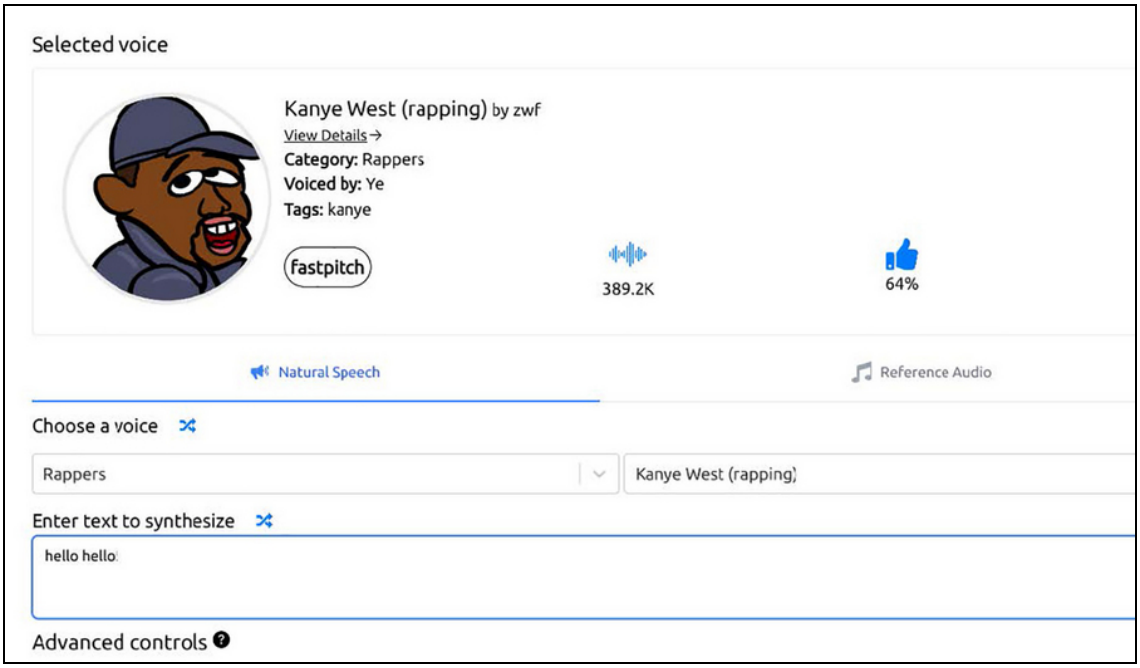


Figura 1.7 Sintesi text-to-speech con UberDuck.ai.

Generazione di video

L'intelligenza artificiale generativa per la generazione di video condivide una cronologia di sviluppo simile a quella della generazione di immagini. Infatti, uno degli elementi chiave nel campo della generazione video è stato lo sviluppo delle GAN. Grazie alla loro precisione nella produzione di immagini realistiche, i ricercatori hanno iniziato ad applicare queste tecniche anche alla generazione di video. Uno degli esempi più notevoli di generazione di video basati sulle GAN è *Motion to Video* di DeepMind, che ha generato video di alta qualità da una singola immagine e da una sequenza di movimenti. Un altro ottimo esempio è il framework basato su deep learning *Video-to-Video Synthesis* (*Vid2Vid*) di NVIDIA, che utilizza le GAN per sintetizzare video di alta qualità da video di input.

Il sistema Vid2Vid è in grado di generare video temporalmente coerenti, il che significa che mantengono un movimento fluido e realistico nel tempo. La tecnologia può essere utilizzata per eseguire varie attività di sintesi video, come le seguenti:

- conversione di video in contesti di registrazione differenti (per esempio, conversione di un video diurno in un video notturno o di uno schizzo in un'immagine realistica);
- modifica di video esistenti (per esempio, modifica dello stile o dell'aspetto degli oggetti in un video);
- creazione di nuovi video da immagini statiche (per esempio, l'animazione di una sequenza di immagini fisse).

Nel settembre 2022, i ricercatori di Meta hanno annunciato la disponibilità di *Make-A-Video* (https://makeavideo.studio), un nuovo sistema di intelligenza artificiale che consente di convertire in videoclip dei prompt in linguaggio naturale. Dietro tale tecnologia, è possibile riconoscere molti dei modelli che abbiamo menzionato finora per altri domini: la comprensione del linguaggio per il prompt, la generazione di immagini e movimenti con la generazione di immagini e la musica di sottofondo realizzata da compositori AI.

Nel complesso, l'AI generativa ha avuto un impatto su molti domini per anni e alcuni strumenti di intelligenza artificiale supportano già costantemente artisti, organizzazioni e utenti generici. Il futuro sembra molto promettente; tuttavia, prima di passare ai modelli disponibili oggi sul mercato, dobbiamo avere una comprensione più profonda delle radici dell'AI generativa, della storia della sua storia e dei recenti sviluppi che alla fine portano agli attuali modelli OpenAI.

La storia e lo stato attuale della ricerca

Nei paragrafi precedenti ho presentato una panoramica delle tecnologie più recenti e all'avanguardia nel campo dell'AI generativa, tutte sviluppate negli ultimi anni. Tuttavia, la ricerca in questo campo può essere fatta risalire a decenni fa.

Possiamo segnare l'inizio della ricerca nel campo dell'AI generativa negli anni Sessanta, quando Joseph Weizenbaum sviluppò il chatbot *ELIZA*, uno dei primi esempi di sistema di elaborazione del linguaggio naturale. Si trattava di un semplice sistema di interazione basato su regole, volto a intrattenere gli utenti con risposte basate su un input testuale, e ha aperto la strada a ulteriori sviluppi sia nell'elaborazione del linguaggio naturale sia nell'AI generativa. Tuttavia, sappiamo che la moderna AI generativa è un sottocampo del deep learning e, sebbene le prime reti neurali artificiali (ANN – *Artificial Neural Network*) siano state introdotte negli anni Quaranta, i ricercatori hanno dovuto affrontare diverse sfide, tra cui la limitata potenza di calcolo e una certa mancanza di comprensione delle basi biologiche del cervello umano. Di conseguenza, le reti neurali artificiali non hanno guadagnato molta attenzione fino agli anni Ottanta, quando, oltre ai nuovi sviluppi dell'hardware e delle neuroscienze, l'avvento del metodo di *backpropagation* ha facilitato la fase di addestramento di tali reti. Infatti, prima dell'avvento della backpropagation, l'addestramento delle reti neurali artificiali era difficile, perché non era possibile calcolare in modo efficiente il gradiente dell'errore rispetto ai parametri o ai pesi associati a ciascun neurone, mentre la backpropagation permetteva di automatizzare il processo di addestramento e ciò consentiva di applicare le reti neurali artificiali.

Poi, negli anni Duemila e poi negli anni Dieci del nuovo secolo, l'avanzamento delle capacità computazionali, insieme all'enorme quantità di dati disponibili per l'addestramento, ha portato alla possibilità di rendere il deep learning più pratico e disponibile al grande pubblico, con un conseguente impulso alla ricerca.

Nel 2013, Kingma e Welling hanno introdotto una nuova architettura di modello nel loro articolo *Auto-Encoding Variational Bayes*, chiamata VAE (*Variational Autoencoder*). I VAE sono modelli generativi che si basano sul concetto di inferenza variazionale. Forniscono un modo per apprendere con una rappresentazione compatta dei dati, codificandoli in uno spazio dimensionale inferiore chiamato *spazio latente* (con il componente *encoder*) e quindi decodificandoli nuovamente nello spazio originale dei dati (con il componente *decoder*).

L'innovazione chiave dei VAE consiste nell'introduzione di un'interpretazione probabilistica dello spazio latente. Invece di apprendere una mappatura deterministica dell'input nello spazio latente, l'encoder esegue la mappatura dell'input con una distribuzione di probabilità sullo spazio latente. Ciò consente ai VAE di generare nuovi campioni campionando dallo spazio latente e decodificando i campioni nello spazio di input.

Per esempio, supponiamo di voler addestrare un VAE in grado di creare nuove immagini di cani e gatti che sembrino reali.

Per fare ciò, il VAE riceve in input l'immagine di un gatto o di un cane e la comprime in un insieme più piccolo di numeri nello spazio latente, che rappresentano le caratteristiche più importanti dell'immagine. Questi numeri sono chiamati *variabili latenti*.

Poi, il VAE prende queste variabili latenti e le utilizza per creare una nuova immagine che sembri essere una *vera* immagine di un gatto o di un cane. Questa nuova immagine potrebbe presentare alcune

differenze rispetto alle immagini originali, ma rimane comunque appartenente allo stesso gruppo di immagini iniziali.

Il VAE migliora la creazione di immagini realistiche nel tempo, confrontando le immagini generate con le immagini reali e regolando le sue variabili latenti per rendere le immagini generate sempre più simili a quelle reali.

I VAE hanno aperto la strada a un rapido sviluppo nel campo dell'AI generativa. Infatti, solo un anno dopo, Ian Goodfellow ha introdotto le GAN (*Generative Adversarial Network*). A differenza dell'architettura VAE, i cui elementi principali sono l'encoder e il decoder, le GAN sono costituite da due reti neurali, un generatore e un discriminatore, che lavorano l'una "contro" l'altra in un gioco a somma zero.

Il generatore crea dati falsi (nel caso di immagini, crea una nuova immagine) che devono sembrare dati reali (per esempio, l'immagine di un gatto). Il discriminatore prende in considerazione sia i dati reali sia quelli falsi e cerca di distinguerli – funge da *critico* in un ipotetico esempio di un falsario d'arte.

Durante l'addestramento, il generatore cerca di creare dati che possano ingannare il discriminatore, facendogli credere che siano reali, mentre il discriminatore cerca di diventare sempre più bravo a distinguere tra dati reali e dati falsi. Le due parti si addestrano insieme in un processo chiamato *adversarial training*.

Nel corso del tempo, il generatore migliora nel creare dati falsi, che sembrano sempre più dati reali, mentre il discriminatore migliora nel distinguere tra dati reali e falsi. Alla fine, il generatore diventa così bravo a creare dati falsi che nemmeno il discriminatore riesce più a distinguere tra dati veri e falsi.

La Figura 1.8 mostra un esempio di volti umani interamente generati da una GAN.

Entrambi i modelli, VAE e GAN, hanno lo scopo di generare nuovi dati che siano indistinguibili dai campioni originali, e la loro architettura è migliorata dalla loro concezione, di pari passo con lo sviluppo di nuovi modelli come *PixelCNN*, proposto da Van den Oord e dal suo team, e *WaveNet*, sviluppato da Google DeepMind, che ha portato a progressi nella generazione di audio e parlato.

Un altro grande traguardo è stato raggiunto nel 2017, quando i ricercatori di Google nel loro paper *Attention Is All You Need* hanno introdotto una nuova architettura, chiamata *Transformer*. L'evento è stato rivoluzionario nel campo della generazione del linguaggio, poiché ha permesso l'elaborazione parallela del testo pur mantenendo la memoria sul contesto del linguaggio, superando i precedenti tentativi di modelli linguistici fondati su RNN o framework LSTM (*Long Short-Term Memory*).

I transformer sono stati infatti le basi per i modelli linguistici di massa chiamati BERT (*Bidirectional Encoder Representations from Transformers*), introdotti da Google nel 2018, e presto sono diventati la base negli esperimenti di elaborazione del linguaggio naturale. I transformer sono anche le fondamenta di tutti i modelli GPT (*Generative Pre-Trained*) introdotti da OpenAI, incluso *GPT-3.5-turbo*, il modello alla base di ChatGPT.

Figura 1.8 Esempi di volti fotorealistici generati da una GAN (tratto da Progressive Growing of GANs for Improved Quality, Stability, and Variation, 2017: https://arxiv.org/pdf/1710.10196.pdf).

Sebbene in quegli anni ci sia stata una quantità significativa di ricerche e risultati, è stato solo a cavallo tra il 2022 e 2023 che l'attenzione generale del pubblico si è spostata verso il campo dell'AI generativa.

Non a caso, il 2023 è stato battezzato l'*anno dell'AI generativa*: è stato l'anno in cui potenti modelli e strumenti di intelligenza artificiale si sono diffusi tra il grande pubblico: i servizi di immagini basati sulla diffusione (MidJourney, DALL-E 2 e Stable Diffusion), ChatGPT di OpenAI, gli strumenti text-to-video (Make-a-Video e Imagen Video) e text-to-3D (DreamFusion, Magic3D e Get3D) sono stati tutti messi a disposizione dei singoli utenti, spesso gratuitamente.

Ciò ha avuto un impatto dirompente per due motivi principali:

- una volta che i modelli di AI generativa sono stati diffusi al pubblico, ogni singolo utente o organizzazione ha avuto la possibilità di sperimentarne e apprezzarne le potenzialità anche senza essere un data scientist o un ingegnere ML;
- i risultati di questi nuovi modelli e la loro creatività erano oggettivamente sbalorditivi e spesso addirittura preoccupanti, pertanto si levò un'urgente richiesta di adattamento, sia per gli individui sia per i governi.

D'ora in poi, in un futuro molto prossimo, assisteremo probabilmente a una crescente adozione di sistemi di intelligenza artificiale, sia per l'uso individuale sia per progetti di livello aziendale.

Riepilogo

In questo capitolo, abbiamo esplorato il mondo dell'AI generativa e i suoi vari domini di applicazione, tra cui la generazione di immagini, testo, musica e video. Abbiamo visto che i modelli di intelligenza artificiale generativa, come ChatGPT o DALL-E, addestrati da OpenAI, utilizzano tecniche di deep learning per individuare *pattern* in set di dati di grandi dimensioni e generare nuovi contenuti da zero. Abbiamo anche discusso della storia dell'AI generativa, delle sue origini e dello stato attuale della ricerca.

L'obiettivo di questo capitolo era quello di fornire le basi dell'AI generativa e di ispirarvi a esplorare ulteriormente questo campo di ricerca.

Nel prossimo capitolo, ci concentreremo su una delle tecnologie più promettenti oggi disponibili sul mercato, ChatGPT: passeremo in rassegna la ricerca che c'è dietro e il suo sviluppo da parte di OpenAI,

l’architettura del suo modello e i principali casi d’uso che può indirizzare a oggi.

Riferimenti

- https://arxiv.org/abs/1406.2661
- https://www.youtube.com/watch?v=Iy9vRvyRf_E
- https://arxiv.org/abs/1912.04958
- http://this-person-does-not-exist.com
- https://arxiv.org/abs/1808.06601
- https://www.microsoft.com/en-us/research/blog/a-deep-generative-model-trifecta-three-advances-that-work-towards-harnessing-large-scale-power
- https://tcwang0509.github.io/vid2vid

Capitolo 2

OpenAI e ChatGPT: oltre l'hype del mercato

Questo capitolo presenta una panoramica di OpenAI e di quello che è, probabilmente, il suo prodotto di maggior successo: ChatGPT. Tratteggeremo la sua storia, la tecnologia su cui si basa e le sue capacità.

L'obiettivo è quello di fornire una conoscenza più profonda di come ChatGPT possa essere utilizzato in vari settori e applicazioni per migliorare la comunicazione, automatizzare i processi e ripensare al modo in cui svolgere determinate attività.

Requisiti tecnici

Per poter riprodurre gli esempi in questo capitolo, avrete bisogno di quanto segue:

- un account OpenAI per accedere al Playground e alle API dei modelli (`https://openai.com/api/login20`);
- il vostro ambiente di sviluppo preferito, come Jupyter o Visual Studio;
- Python 3.7.1+ (`https://www.python.org/downloads`);
- pip (`https://pip.pypa.io/en/stable/installation`);
- la libreria di OpenAI per Python (`https://pypi.org/project/openai`).

Che cos’è OpenAI?

OpenAI nasce come organizzazione di ricerca nel 2015 fondata da Elon Musk, Sam Altman, Greg Brockman, Ilya Sutskever, Wojciech Zaremba e John Schulman. Come dichiarato nella pagina web di OpenAI, la sua *mission* è quella di “garantire che l’intelligenza artificiale generale (AGI) benefici di tutta l’umanità”. Con il termine *generale*, si intende il fatto la capacità di apprendere e di svolgere – con pari abilità - le medesime attività che noi, in quanto esseri umani, siamo in grado di svolgere.

Dal 2015, OpenAI ha concentrato la sua ricerca sul DRL (*Deep Reinforcement Learning*), un sottoinsieme del machine learning che combina il RL (*Reinforcement Learning*) con le reti neurali profonde. Il primo contributo in questo campo risale al 2016, quando l’azienda ha rilasciato *OpenAI Gym*, un toolkit per consentire ai ricercatori di sviluppare e sottoporre a test gli algoritmi di *reinforcement learning*.

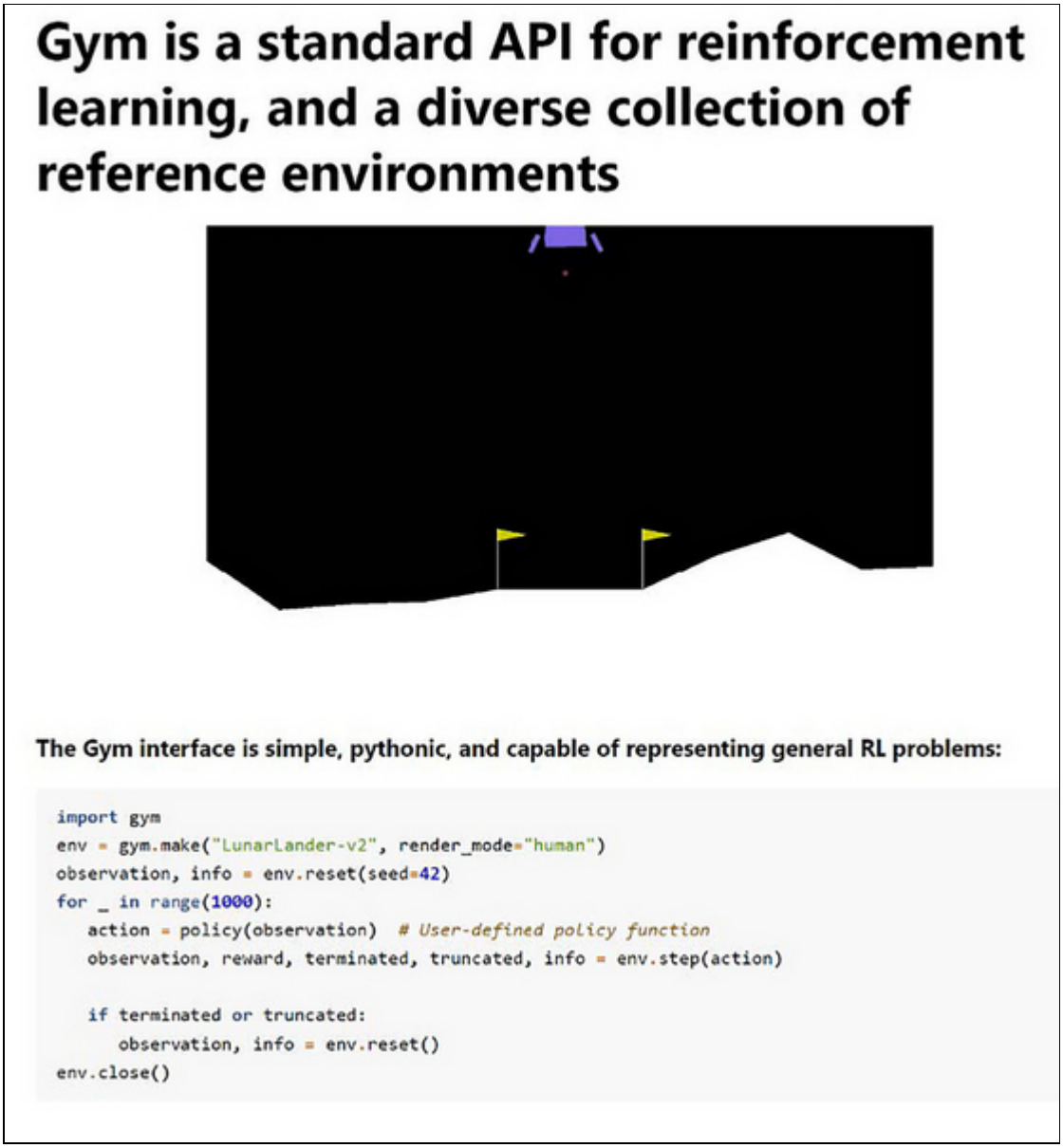


Figura 2.1 Landing page della documentazione di Gym (https://www.gymlibrary.dev/).

OpenAI ha continuato a fare ricerca e a contribuire in questo campo, ma i suoi risultati più notevoli riguardano i modelli generativi, i GPT (*Generative Pre-Trained Transformers*).

Dopo aver introdotto l'architettura del modello nel loro articolo *Improving Language Understanding by Generative Pre-Training* e

averlo battezzato GPT-1, i ricercatori di OpenAI hanno rilasciato, nel 2019, il suo successore, GPT-2, versione allenata su un *corpus* chiamato WebText, che all'epoca conteneva poco più di 8 milioni di documenti per un totale di 40 GB di testo proveniente da URL condivisi in post su Reddit con almeno 3 upvote. Il modello aveva 1,2 miliardi di parametri, dieci volte più del suo predecessore.

La Figura 2.2 mostra la home page di un'interfaccia utente di GPT-2 pubblicata da HuggingFace (`https://transformer.huggingface.co/doc/distil-gpt2`).

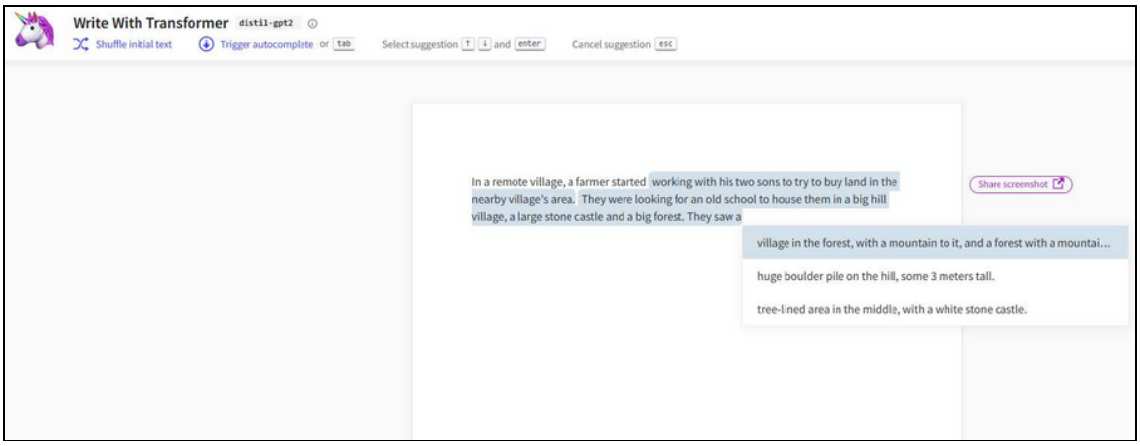


Figura 2.2 GPT-2 scrive un paragrafo basato su un prompt. Fonte: https://transformer.huggingface.co/doc/distil-gpt2.

Nel 2020, OpenAI ha prima annunciato e poi rilasciato GPT-3, che, con i suoi 175 miliardi di parametri, ha migliorato notevolmente i risultati dei benchmark rispetto a GPT-2.

Oltre ai modelli generativi dedicati al linguaggio naturale, OpenAI ha anche sperimentato nel campo della generazione di immagini, rilasciando nel 2021 il suo primo modello in questo campo, *DALL-E*. Al momento della pubblicazione di questo testo è alla sua terza versione (DALL-E 3). Come ho accennato nel capitolo precedente, DALL-E è in grado di creare nuove immagini a partire da un input in

linguaggio naturale, che viene interpretato dalla versione più recente di GPT-3.

Nella Figura 2.3, potete vedere un esempio di immagini generate da DALL-E (nella sua prima versione) a partire dal prompt in linguaggio naturale “generare un’immagine realistica di una tazza di caffè in un ambiente accogliente”.

Un modo per creare immagini originali consiste nell’usare il laboratorio di OpenAI DALL-E (`https://labs.openai.com`), dove potete avere accesso a una quantità limitata di crediti gratuiti per le vostre prove.

OpenAI ha dato un grande impulso a diversi ambiti dell’AI generativa, ma si è distinta soprattutto nella comprensione e generazione del testo, grazie ai modelli GPT di base che esamineremo nei prossimi paragrafi.

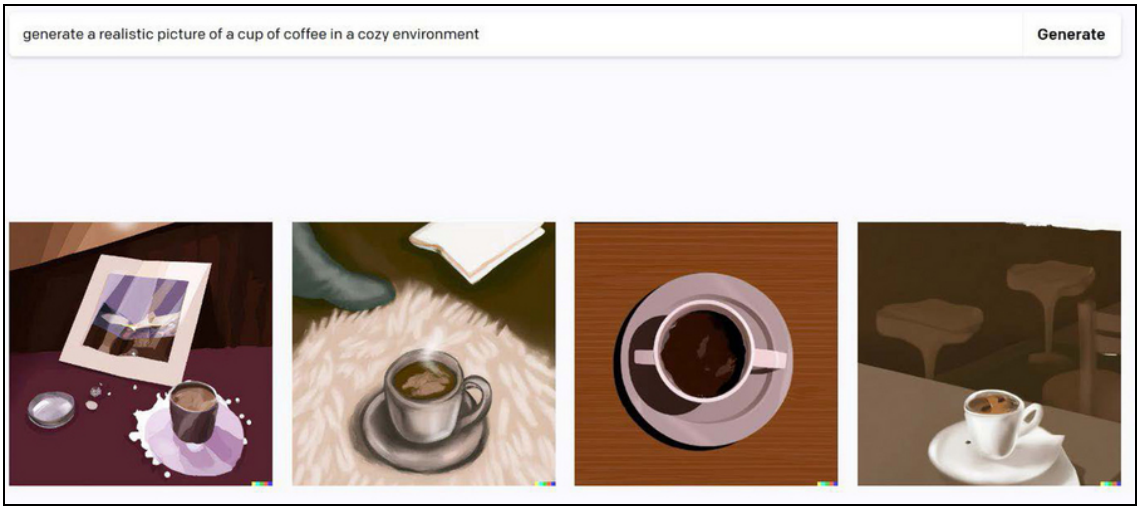


Figura 2.3 Immagini generate da DALL-E usando come input un prompt in linguaggio naturale.

Una panoramica delle famiglie di modelli OpenAI

Oggi, OpenAI offre un insieme di modelli pre-addestrati e pronti all'uso che possono essere utilizzati da tutti. Questo ha due importanti implicazioni:

- i potenti modelli di base (*base models*) possono essere utilizzati senza la necessità di lunghi e costosi allenamenti;
- non è necessario avere particolari competenze tecniche per manipolare questi modelli.

Gli utenti possono sottoporre a test i modelli OpenAI nell'*OpenAI Playground*, un'interfaccia utente nella quale è possibile interagire con i modelli senza dover scrivere codice.

Nella Figura 2.4, potete vedere la pagina iniziale dell'OpenAI Playground. Come potete vedere dalla figura, il Playground offre un'interfaccia utente nella quale l'utente può iniziare a interagire con il modello, che si seleziona sul lato destro dell'interfaccia utente. Per iniziare a interagire con il Playground, basta digitare una domanda o istruzione nello spazio di input in linguaggio naturale. Potete iniziare anche con alcuni esempi disponibili nella documentazione di OpenAI (`https://platform.openai.com/examples`).

Prima di approfondire le famiglie di modelli, definiamo alcuni termini che useremo in questo capitolo.

- *Token*: i token sono frammenti o segmenti di parole che vengono utilizzati dall'API del modello per elaborare i prompt di input. A differenza delle parole intere, i token possono contenere spazi finali o addirittura sotto-parole parziali. Per comprendere meglio il concetto di token in termini di lunghezza, ci sono alcune linee guida generali da tenere a mente. Per esempio, un token inglese è

approssimativamente equivalente a quattro caratteri o ai 3/4 di una parola.

- *Prompt*: nel contesto dell'elaborazione del linguaggio naturale e del machine learning, un prompt è il testo che viene dato in input a un modello per generare una risposta o un output. Il prompt può essere una domanda, una dichiarazione o una frase, e viene usato per fornire un contesto e una direzione al modello di linguaggio.

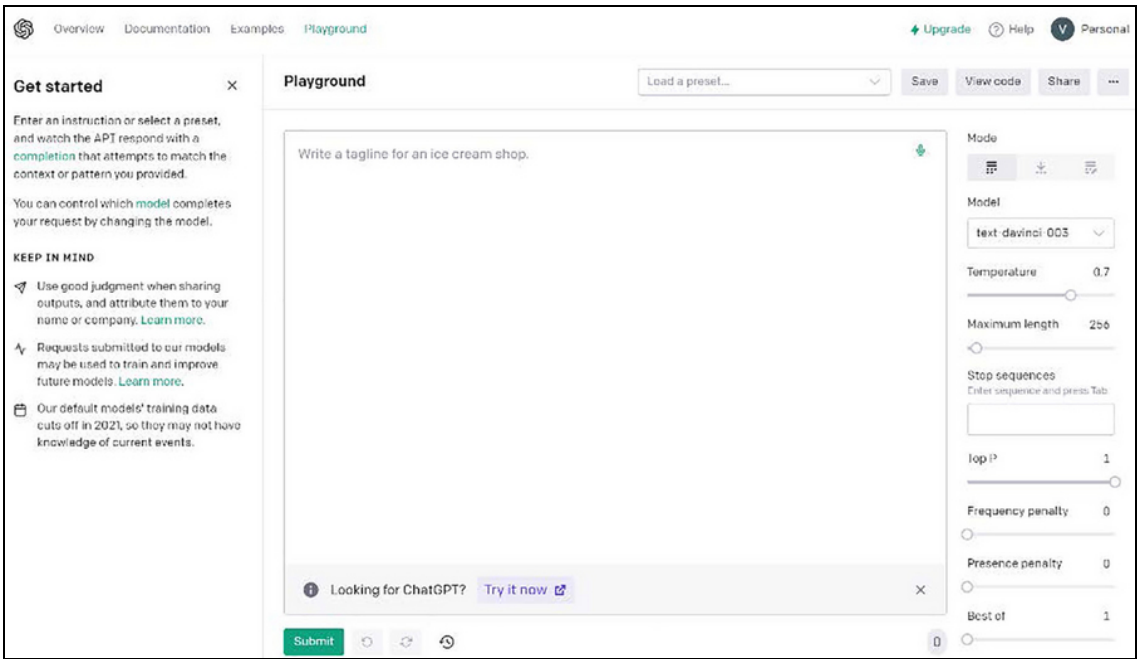


Figura 2.4 OpenAI Playground su https://platform.openai.com/playground.

- *Contesto*: nel campo del GPT, si riferisce alle parole e alle frasi che precedono il prompt dell'utente. Questo contesto viene utilizzato dal modello di linguaggio per generare la parola o frase successiva più probabile, basandosi sui pattern e sulle relazioni trovate nei dati di addestramento.
- *Confidenza del modello*: si riferisce al livello di probabilità che un modello di intelligenza artificiale assegna a una determinata previsione o output. Nel contesto dell'elaborazione del linguaggio

naturale, la confidenza viene spesso utilizzata per indicare quanto il modello di intelligenza artificiale è sicuro della correttezza o rilevanza della sua risposta, generata in base a un dato prompt di input.

Le definizioni precedenti saranno fondamentali per capire come utilizzare le famiglie di modelli Azure OpenAI e come configurare i loro parametri.

Nel Playground, ci sono tre principali famiglie di modelli che possono essere sottoposte a test.

- *Modelli di completamento*: un insieme di modelli in grado di comprendere e generare parole in linguaggio naturale in termini di predizione del token successivo, dati i token di input. Questa categoria di modelli è anche nota come *base model*.

DEFINIZIONE

Un base model è l'output della fase di training non-supervisionato di un large language model come GPT-3. A valle di questo training, il modello di output non è altro che un modello statistico predittivo, in grado di predire il token successivo, dati i token di input forniti dall'utente. Un base model può essere a sua volta sottoposto a ulteriori raffinamenti di training supervisionato (come il fine-tuning e il reinforcement learning, che approfondiremo nei prossimi paragrafi), in modo da ottenere un modello più simile a un "assistente" in grado di conversare con gli utenti e risolvere i problemi che essi gli sottopongono.

Un esempio di base model è il GPT-3: addestrato su un vasto corpus di testi, può eseguire una vasta gamma di compiti di linguaggio naturale come la traduzione, il riassunto, la risposta alle domande e altro. La Figura 2.5 mostra un esempio.

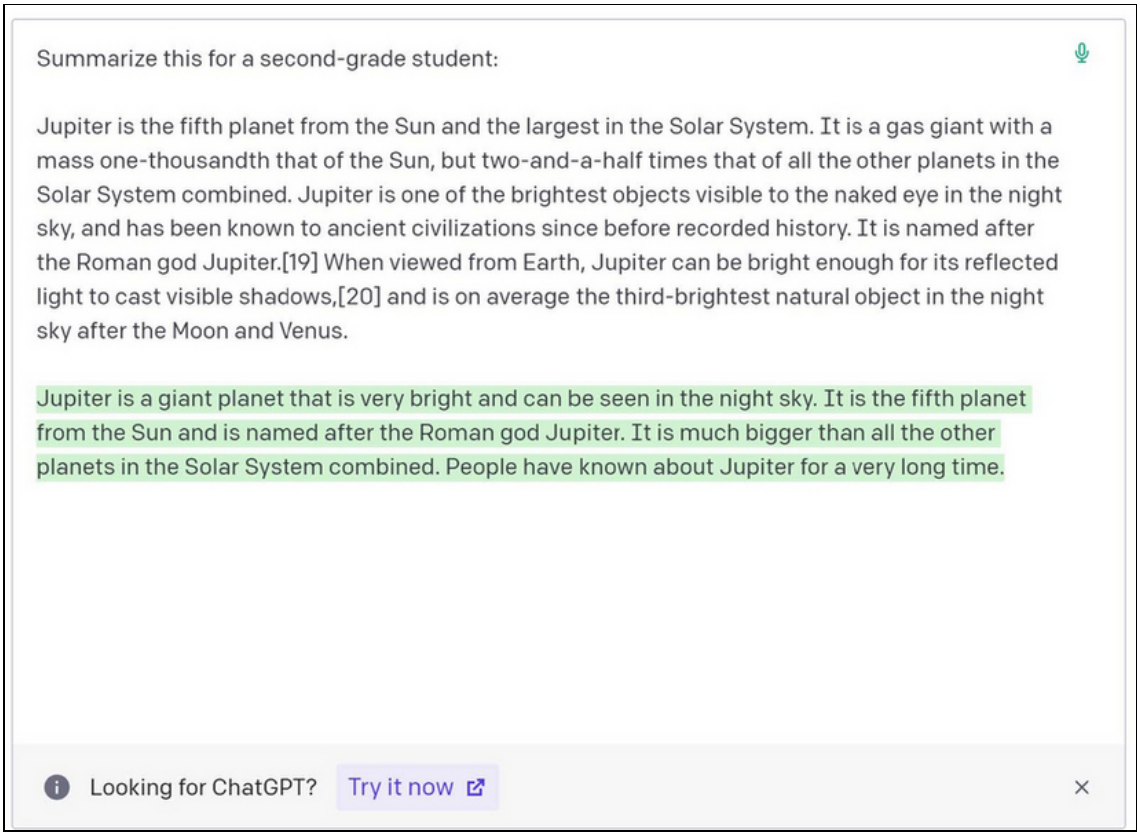


Figura 2.5 Un esempio di compito di riassunto con GPT-3.

- *Modelli di chat/assistant*: questo è un insieme di modelli più recenti che si basano su GPT-3 e hanno ricevuto ulteriori fasi di raffinamento e allineamento con l'interazione umana. Questi modelli non si limitano a predire il token successivo, ma sono veri e propri assistenti, in grado di indirizzare le richieste degli utenti. Un esempio di questi modelli è il GPT-3.5-turbo, il motore dietro ChatGPT che, oltre alla sua API, è anche utilizzabile nel Playground con un'interfaccia utente dedicata.
- *Modelli Codex* [deprecati]: un insieme di modelli in grado di comprendere e generare codice in vari linguaggi di programmazione. Codex può tradurre prompt in linguaggio naturale in codice funzionante, rendendolo uno strumento potente

per lo sviluppo di software. La Figura 2.7 ne mostra un esempio con Codex.

NOTA

Nel marzo 2023, OpenAI ha annunciato che i modelli Codex saranno deprecati da quella data in poi. Questo è dovuto alle incredibili capacità dei nuovi modelli di chat (incluso GPT-3.5-turbo, il modello dietro ChatGPT) che includono anche compiti di programmazione, con risultati che corrispondono o addirittura superano quelli dei modelli Codex.

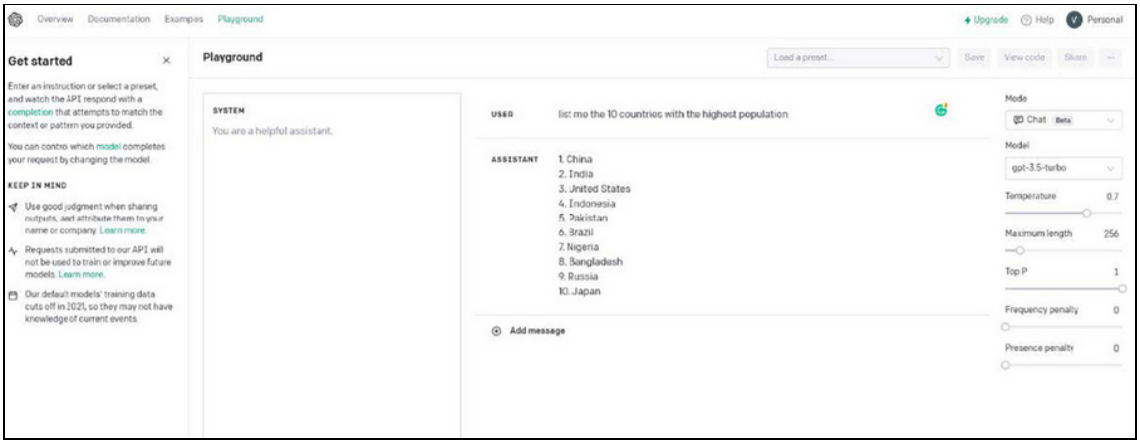
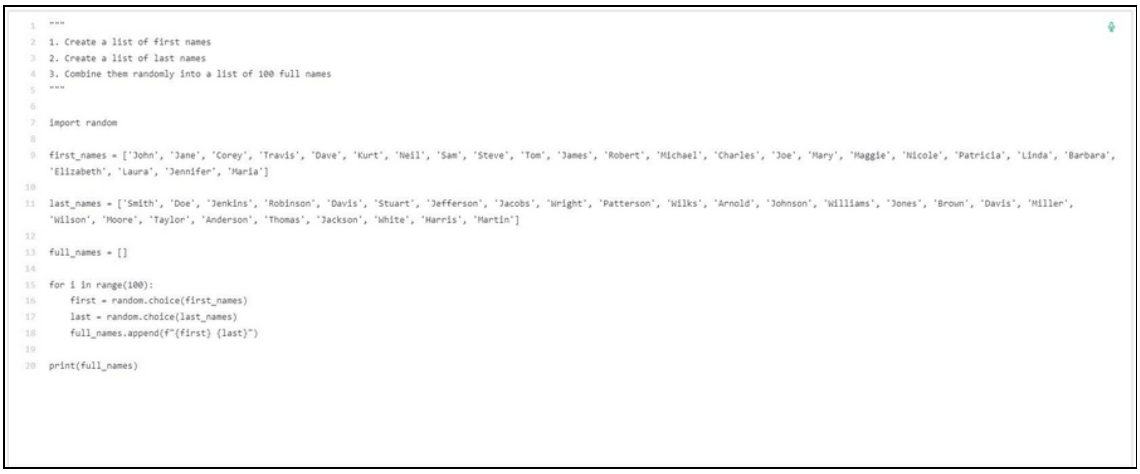


Figura 2.6 Un esempio di interazione con GPT-3.5.



```
"""
1. Create a list of first names
2. Create a list of last names
3. Combine them randomly into a list of 100 full names
"""

import random

first_names = ['John', 'Jane', 'Corey', 'Travis', 'Dave', 'Kurt', 'Neil', 'Sam', 'Steve', 'Tom', 'James', 'Robert', 'Michael', 'Charles', 'Joe', 'Mary', 'Maggie', 'Nicole', 'Patricia', 'Linda', 'Barbara',
'Elizabeth', 'Laura', 'Jennifer', 'Maria']

last_names = ['Smith', 'Doe', 'Jenkins', 'Robinson', 'Davis', 'Stuart', 'Jefferson', 'Jacobs', 'Wright', 'Patterson', 'Wilks', 'Arnold', 'Johnson', 'Williams', 'Jones', 'Brown', 'Davis', 'Miller',
'Wilson', 'Moore', 'Taylor', 'Anderson', 'Thomas', 'Jackson', 'White', 'Harris', 'Martin']

full_names = []

for i in range(100):
    first = random.choice(first_names)
    last = random.choice(last_names)
    full_names.append(f"{first} {last}")

print(full_names)
```

Figura 2.7 Un esempio di generazione di codice con Codex.

Per ciascun modello, è possibile anche modificare alcuni parametri configurabili. Ecco un elenco.

- *Temperatura* (da 0 a 1): controlla la casualità della risposta del modello. Una temperatura bassa rende il vostro modello più deterministico, il che significa che tenderà a dare la stessa risposta alla stessa domanda. Per esempio, se chiedo più volte al mio modello “Che cos’è OpenAI?” con la temperatura impostata a 0, darà sempre la stessa risposta. Al contrario, se pongo la stessa domanda a un modello con la temperatura impostata a 1, cercherà ogni volta di modificare le sue risposte, in termini di formulazione e stile.
- *Lunghezza massima* (da 0 a 2048): controlla la lunghezza (in termini di token) della risposta del modello al prompt dell’utente.
- *Sequenze di stop* (input dell’utente): fa terminare le risposte nel punto desiderato, come la fine di una frase o di un elenco.
- *Top probabilities* (da 0 a 1): controlla quali token il modello considererà durante la generazione di una risposta. Impostandolo a 0.9, il modello considererà il 90% dei token più probabili tra tutti i possibili. Potrebbe sorgere la domanda: perché non impostare le *top probabilities* a 1, così da scegliere tutti i token più probabili? La risposta è che gli utenti potrebbero voler mantenere una certa varietà quando il modello ha bassa confidenza, anche nei token con punteggio più alto.
- *Penalty frequency* (da 0 a 1): controlla la ripetizione degli stessi token nella risposta generata. Più alta è la penalità, minore è la probabilità di vedere gli stessi token più di una volta nella stessa risposta. La penalità riduce la possibilità in modo proporzionale, in base a quante volte un token è apparso nel testo fino a quel momento (questa è la differenza chiave dal parametro successivo).

- *Presence penalty* (da 0 a 2): parametro simile al precedente ma più rigido. Riduce la possibilità di ripetere qualsiasi token che è apparso nel testo fino a quel momento. Essendo più rigida della *penalty frequency*, la *presence penalty* aumenta anche la probabilità di introdurre nuovi argomenti in una risposta.
- *Best of* (da 0 a 20): genera molteplici risposte e mostra solo quella con la migliore probabilità totale su tutti i suoi token.
- *Testo pre e post risposta* (input dell'utente): consente di inserire del testo prima e dopo la risposta del modello. Questo può aiutare a preparare il modello per una risposta.

Oltre a sperimentare i modelli OpenAI nel Playground, potete richiamare le API dei modelli nel vostro codice e integrare i modelli nelle vostre applicazioni. Infatti, nell'angolo superiore destro del Playground, potete fare clic su *Visualizza codice* ed esportare la configurazione come mostrato nella Figura 2.8. Come potete vedere dalla figura, il codice esporta la configurazione dei parametri che avete impostato nel Playground.

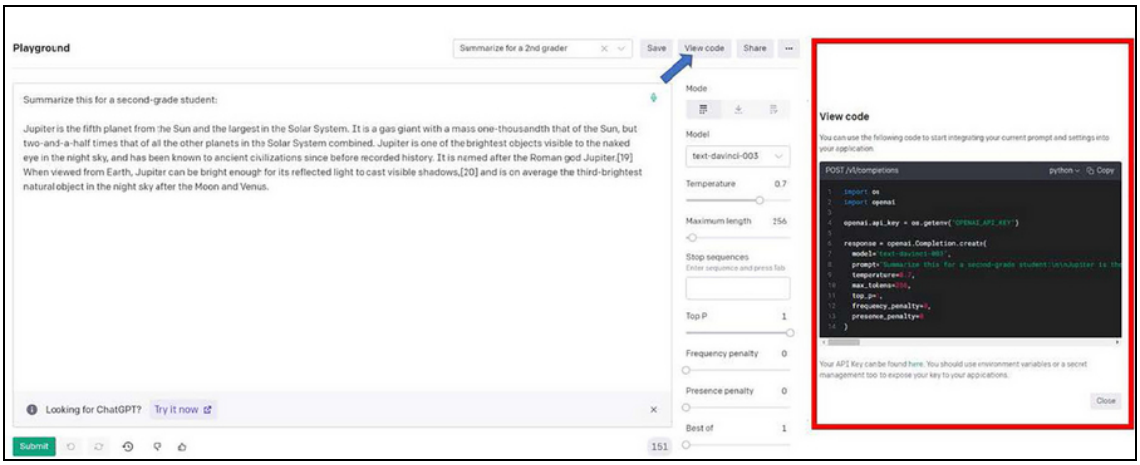


Figura 2.8 Codice Python per chiamare un modello GPT3 con un prompt in linguaggio naturale.

Ora potete iniziare a utilizzare la libreria OpenAI in Python installandola tramite `pip install openai` nel vostro terminale. Per utilizzare i modelli, avrete bisogno di generare una chiave API. Potete trovare le vostre chiavi API nelle impostazioni del vostro account (`https://platform.openai.com/account/api-keys`), come mostrato nella Figura 2.9.

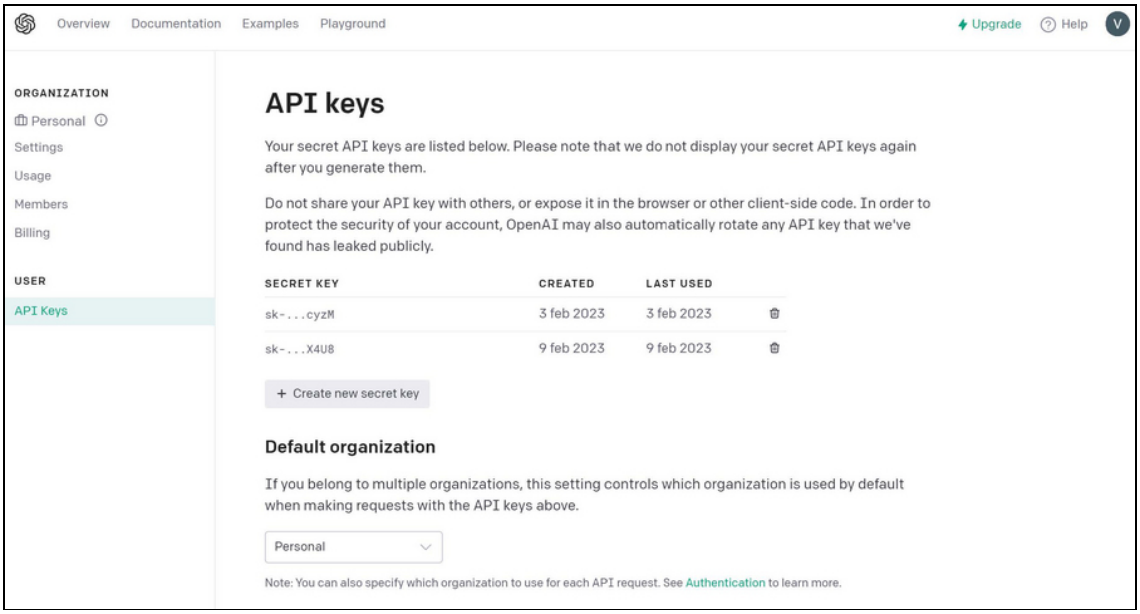


Figura 2.9 Chiavi API nella pagina delle impostazioni dell'account del vostro profilo OpenAI.

Con le API di OpenAI, potete anche provare le seguenti famiglie di modelli aggiuntive, che non sono disponibili nel Playground.

- *Moderazione*: un modello perfezionato sviluppato da OpenAI che può rilevare contenuti di testo potenzialmente sensibili o non sicuri. La moderazione utilizza algoritmi di machine learning per classificare il testo come sicuro o non sicuro in base al suo contesto e all'uso del linguaggio. Questo modello può essere utilizzato per automatizzare la moderazione dei contenuti su piattaforme di social media, comunità online e in molti altri

ambiti. Esistono diverse categorie, come odio, odio/minaccioso, autolesionismo, sessuale, sessuale/minori, violenza, violenza/grafica.

Ecco un esempio di codice per l'API di *Moderazione*:

```
import os
import openai
openai.api_key = os.getenv("OPENAI_API_KEY")
openai.Moderation.create(
  input="Voglio ucciderlo",        )
```

L'output è il seguente:

```
<OpenAIObject id=modr-6sHusuY9frxJdfqTBXHsOAfWhckrh at
0x218bd8482c0> JSON: {
  "id": "modr-6sHusuY9frxJdfqTBXHsOAfWhckrh",
  "model": "text-moderation-004",
  "results": [
    {
      "categories": {
        "hate": false,
        "hate/threatening": false,
        "self-harm": false,
        "sexual": false,
        "sexual/minors": false,
        "violence": true,
        "violence/graphic": false
      },
      "category_scores": {
        "hate": 1.7164344171760604e-05,
        "hate/threatening": 2.614225103059198e-08,
        "self-harm": 2.5988580176772302e-08,
        "sexual": 2.8184256279928377e-06,
        "sexual/minors": 9.1383149936064e-09,
        "violence": 0.9910049438476562,
        "violence/graphic": 5.316753117767803e-07
      },
      "flagged": true
    }
  ]        }
```

In questo caso, l'API di moderazione ha rilevato contenuti di violenza.

- *Embedding*: questi modelli sono in grado di trasformare parole o frasi in rappresentazioni vettoriali in uno spazio multidimensionale. Le distanze matematiche tra le diverse istanze in questo spazio rappresentano la loro somiglianza in termini di

significato. Per esempio, immaginate le parole *regina*, *donna*, *re* e *uomo*. Idealmente, nel nostro spazio multidimensionale, in cui le parole sono vettori, se la rappresentazione è corretta, vogliamo ottenere il risultato rappresentato nella Figura 2.10.

$$\overrightarrow{king} + (\overrightarrow{woman} - \overrightarrow{man}) \approx \overrightarrow{queen}$$

Figura 2.10 Esempio di equazioni vettoriali tra parole.

Ciò significa che la distanza tra *donna* e *uomo* dovrebbe essere uguale alla distanza tra *regina* e *re*. Ecco un esempio di embedding:

```
import openai
embedding = openai.Embedding.create(
    input="Il gatto è sul tavolo",
model="text-embedding-ada-002")["data"][0][ "embedding"]
```

Il metodo precedente crea una rappresentazione vettoriale dell'input. Possiamo dare un'occhiata ai primi dieci vettori dell'output:

```
embedding[1:10]
[-0.01369840931147337,
 -0.007505378685891628,
 -0.002576263388618827,
 -0.014773285016417503,
 0.019935185089707375,
 -0.01802290789783001,
 -0.01594814844429493,
 -0.0010944041423499584,
 -0.014323337003588676]
```

Gli embedding possono essere estremamente utili in scenari di ricerca intelligente. Infatti, ottenendo l'embedding dell'input dell'utente e dei documenti che l'utente vuole cercare, è possibile calcolare metriche di distanza (nello specifico, la *cosine similarity*) tra l'input e i documenti. Facendo ciò, possiamo

recuperare i documenti che sono più vicini, in termini di distanza matematica, all'input dell'utente.

- *Whisper*: un modello di riconoscimento vocale che può trascrivere l'audio in testo. Whisper può riconoscere e trascrivere varie lingue e dialetti con alta precisione, rendendolo uno strumento prezioso per i sistemi di riconoscimento vocale automatico. Ecco un esempio:

```
# Note: you need to be using OpenAI Python   v     0.27.0 for
the code below to work
import openai
openai.api_key = os.getenv("OPENAI_API_KEY")
audio_file= open("/path/to/file/audio.mp3", "rb")
transcript = openai.Audio.transcribe("whisper-1", audio_file)
```

Output:

```
{"text": Yes, hi, I just noticed a dent on the side of my car
and I have no idea how it got there. There were no witnesses
around and I'm really frustrated.
….        }
```

Tutti i modelli precedenti sono stati pre-addestrati su un'enorme base di conoscenza. Tuttavia, esistono alcuni modi per rendere il vostro modello più personalizzato e adatto al vostro caso d'uso.

Il primo metodo è incorporato nel modo in cui il modello è progettato, e prevede di fornire al vostro modello il contesto nell'approccio di *few-shot learning* (esamineremo questa tecnica più avanti nel libro). Per esempio, potreste chiedere al modello di generare un articolo il cui schema e lessico richiamano un altro che avete già scritto. Per questo, potete fornire al modello la vostra richiesta di generare un articolo e l'articolo precedente, come riferimento o contesto, in modo che il modello sia meglio preparato per la vostra richiesta.

La Figura 2.11 mostra un esempio.

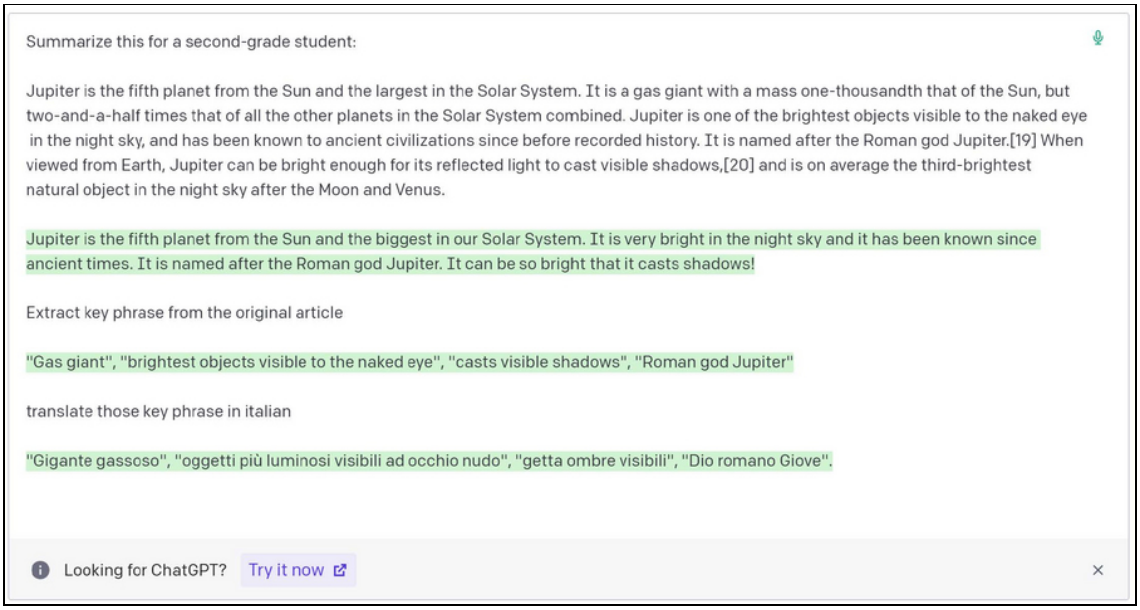


Figura 2.11 Un esempio di conversazione all'interno dell'OpenAI Playground con l'approccio di few-shot learning.

Il secondo metodo è più sofisticato ed è chiamato fine-tuning. Il *fine-tuning* è il processo di adattamento di un modello pre-addestrato a un nuovo compito.

Nel fine-tuning, alcuni parametri del modello pre-addestrato vengono modificati, per adattarsi meglio ai dati del nuovo compito. Questo viene fatto addestrando il modello (in modalità supervisionata) su un dataset più piccolo e più specifico per il nuovo compito. L'idea chiave dietro al fine-tuning è quella di sfruttare la conoscenza appresa dal modello pre-addestrato e adattarla al nuovo compito, piuttosto che addestrare un modello da zero.

La Figura 2.12 mostra un esempio di fine-tuning. Nella figura, potete vedere uno schema rappresentativo del fine-tuning sui modelli precostruiti di OpenAI. L'idea è che avete a disposizione un modello pre-addestrato con pesi o parametri di uso generale.

Poi, alimentate il vostro modello con dati personalizzati, tipicamente sotto forma di chiave-valore di prompt e completamenti, come

mostrato di seguito:

```
{"prompt": "<testo del prompt>", "completion": "<testo generato
ideale>"}
{"prompt": "<testo del prompt>", "completion": "<testo generato
ideale>"}
{"prompt": "<testo del prompt>", "completion": "<testo generato
ideale>"}        ...
```

Una volta terminato l'addestramento, l'output sarà un modello personalizzato che si comporta particolarmente bene per un determinato compito (per esempio, la *sentiment analysis* delle recensioni di un prodotto).

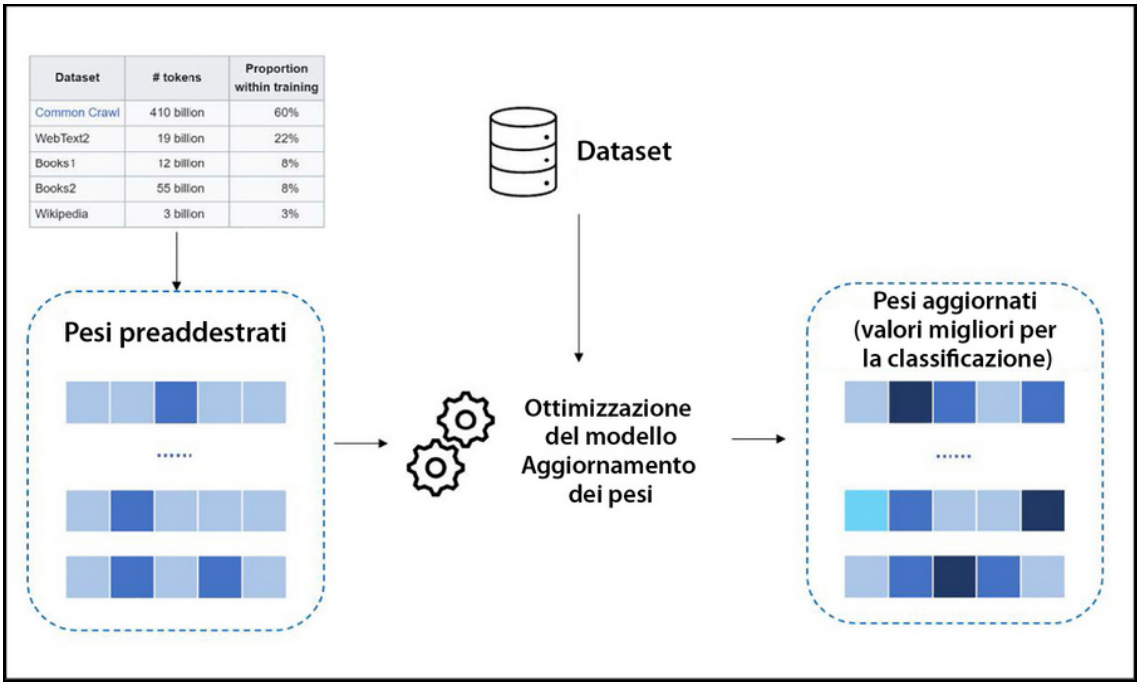


Figura 2.12 Fine-tuning del modello.

Il vantaggio del fine-tuning è che si possono adattare i modelli precostruiti a specifici casi d'uso, senza la necessità di ri-addestrarli da zero, sfruttando dataset di addestramento più piccoli e quindi con meno tempo di addestramento e calcolo. Allo stesso tempo, il modello mantiene la sua potenza generativa e l'accuratezza appresa attraverso l'addestramento originale, quello che ha operato sul dataset più grande.

In questo paragrafo, abbiamo visto una panoramica dei modelli offerti da OpenAI, da quelli che potete provare direttamente nel Playground (GPT-x) a quelli più complessi, come i modelli di embedding. Ora sapete che, oltre all'uso dei modelli nel loro stato precostruito, potete anche personalizzarli tramite il fine-tuning, fornendo un insieme di esempi da cui apprendere.

Nei prossimi paragrafi ci concentreremo sull'architettura che è dietro questi modelli, partendo dalle basi matematiche fino ad arrivare alle grandi scoperte che hanno reso possibile ChatGPT.

Verso ChatGPT: le basi matematiche del modello

Dalla sua fondazione nel 2015, OpenAI ha investito nella ricerca e nello sviluppo della classe di modelli chiamata GPT (*Generative Pre-trained Transformers*), e questi hanno catturato l'attenzione di tutti, nel momento in cui sono usciti allo scoperto come motore di ChatGPT.

I modelli GPT appartengono al framework architetturale dei *transformer*, introdotti dai ricercatori di Google in un articolo del 2017: *Attention Is All You Need.*

L'architettura del transformer è stata introdotta per superare i limiti delle tradizionali *reti neurali ricorrenti* (RNN). Le RNN sono state introdotte per la prima volta negli anni Ottanta dai ricercatori del Los Alamos National Laboratory, ma non hanno attirato molta attenzione fino agli anni Novanta. L'idea originale era quella di elaborare dati sequenziali o dati di serie temporali, mantenendo le informazioni attraverso i passaggi temporali.

Infatti, fino a quel momento, l'architettura classica delle reti neurali artificiali (ANN) era quella delle *feed forward ANN*, dove l'output di

ogni *layer nascosto* diventa l'input del successivo, senza mantenere informazioni sui passaggi precedenti.

Per capire l'idea su cui si basa il transformer, dobbiamo partire dalle sue origini. Ci soffermeremo quindi sui seguenti argomenti:

- la struttura delle RNN;
- i principali limiti delle RNN;
- come questi limiti sono stati superati con l'introduzione di nuovi elementi architetturali, tra cui l'encoding posizionale, la self-attention e il layer di feed forward;
- come siamo arrivati allo stato dell'arte di GPT e ChatGPT.

Iniziamo con l'architettura dei predecessori dei transformer.

La struttura delle reti neurali ricorrenti (RNN)

Immaginate di voler prevedere il prezzo di una casa. Se avessimo a disposizione solamente il suo prezzo odierno, potremmo utilizzare un'architettura *feed forward* in cui applichiamo una trasformazione non lineare all'input tramite un *layer nascosto* (con una funzione di attivazione) e otteniamo come output la previsione del prezzo per domani. La Figura 2.13 mostra come fare.

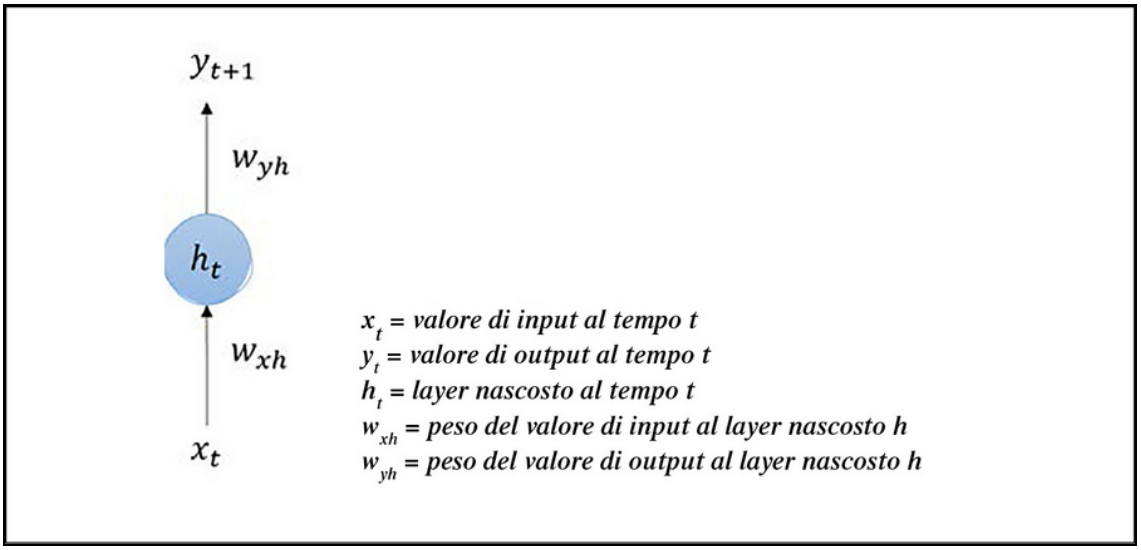


Figura 2.13 Architettura feed forward con layer nascosto.

Tuttavia, per questo tipo di dati, è verosimile avere a disposizione sequenze di dati temporalmente più lunghe. Per esempio, potremmo avere la serie temporale di questa casa per gli scorsi cinque anni. Naturalmente, vogliamo incorporare queste informazioni nel nostro modello, in modo che la nostra RNN sia in grado di mantenere la memoria degli input passati per interpretare correttamente l'input corrente e prevedere gli output futuri.

Quindi, immaginate di non avere solo il prezzo odierno, ma anche il prezzo di ieri (`t-1`) e dell'altro ieri (`t-2`). La Figura 2.14 mostra come possiamo incorporare questa informazione nel calcolo del prezzo di domani.

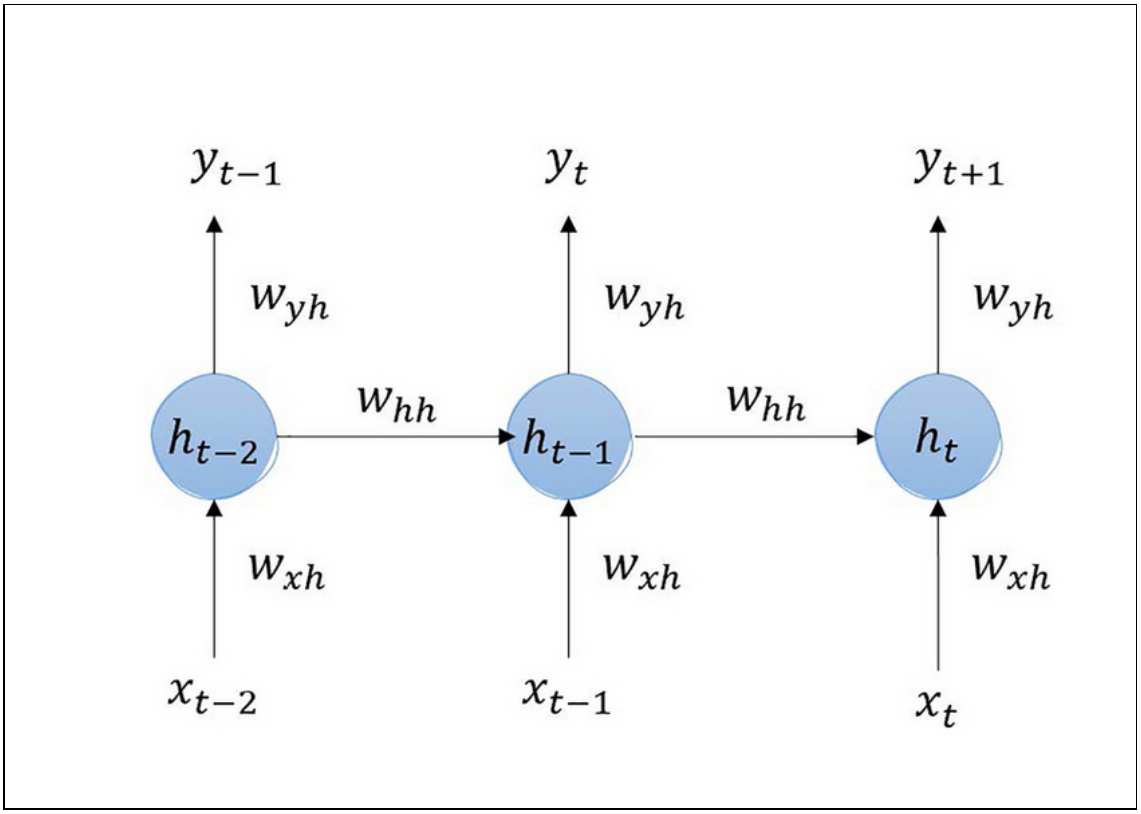


Figura 2.14 Esempio di una RNN.

Dato che siamo interessati solo al prezzo di domani, ignoriamo gli output finali intermedi per `t-1` e `t` (Figura 2.15).

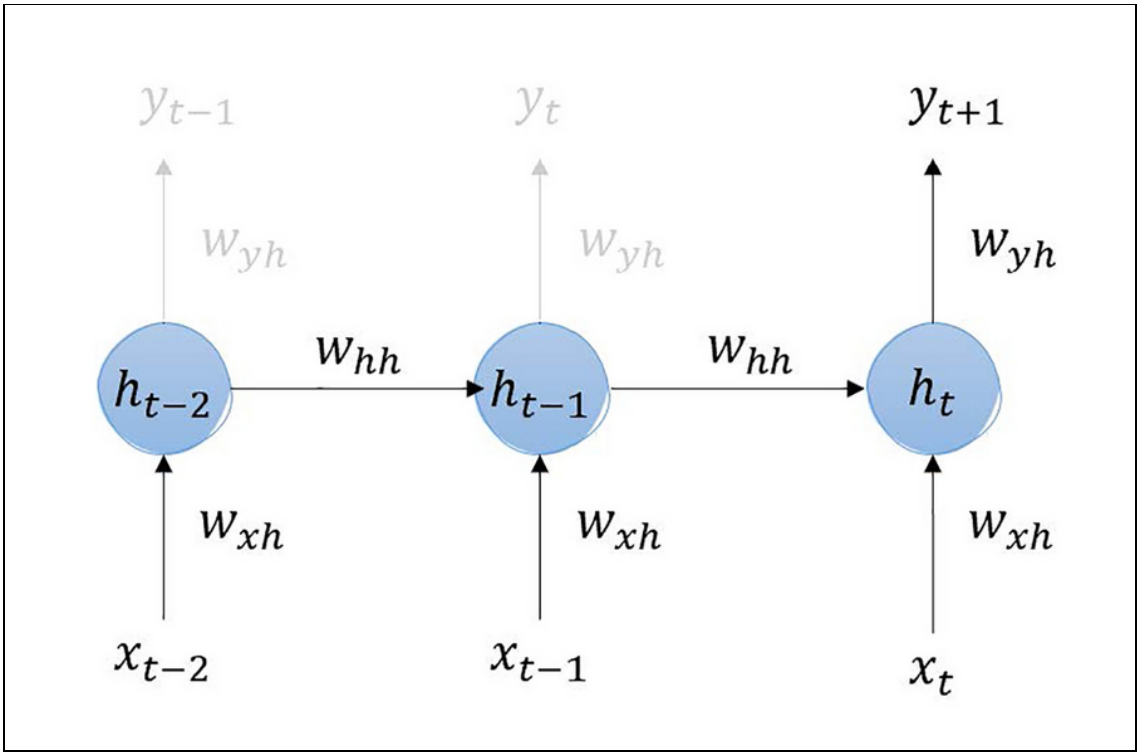


Figura 2.15 Esempio di una RNN senza t-1 e t.

Come potete vedere, l'output del layer nascosto di `t-2` viene utilizzato come input (ponderato) del layer nascosto di `t-1`, che prende anche l'input di `t-1`. Poi, l'output del layer nascosto a `t-1`, che già mantiene la memoria degli input di `t-2` e `t-1`, viene utilizzato come input del layer nascosto di `t`. Di conseguenza, il prezzo per domani (`yt+1`), che è quello che ci interessa, conserva la memoria di tutti gli input dei giorni precedenti.

Infine, se vogliamo ridurre questa figura, possiamo pensare alla RNN nel modo rappresentato nella Figura 2.16.

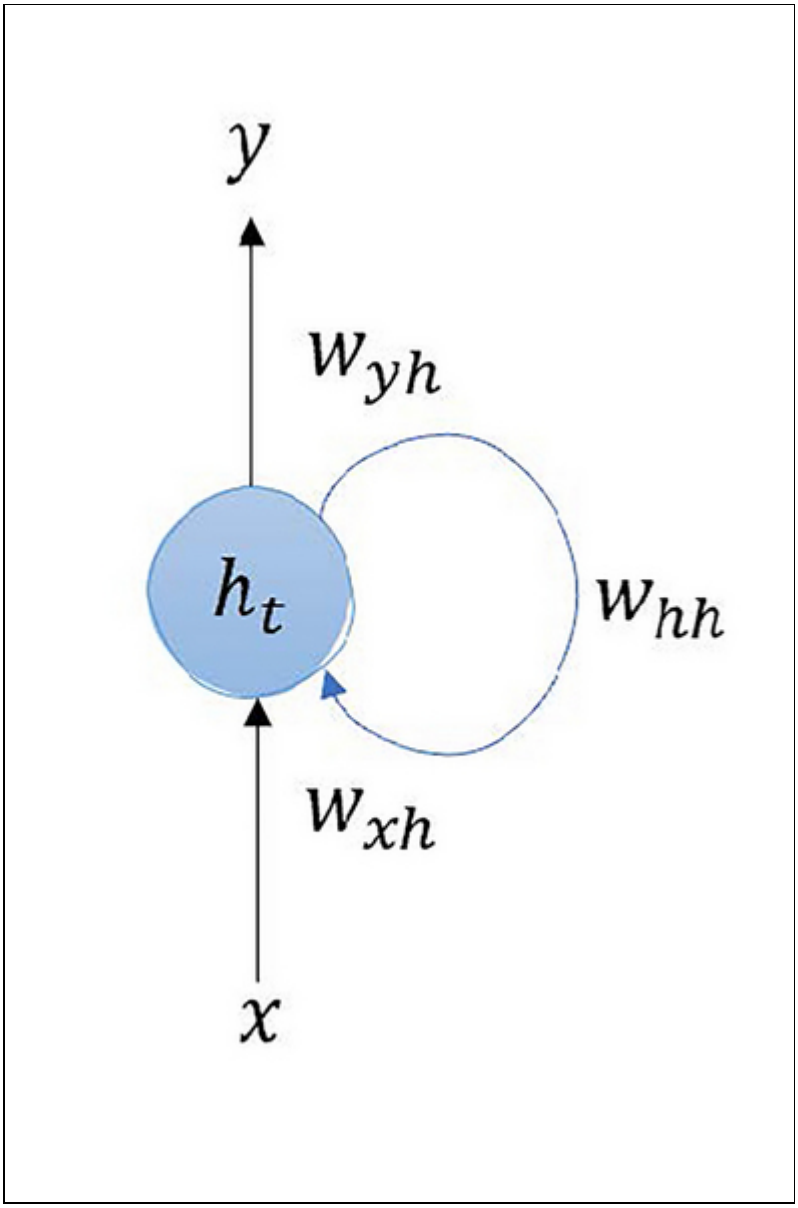


Figura 2.16 Esempio di una RNN nella sua forma avvolta.

Questo significa che l'output del layer RNN al passo temporale `t-n` viene poi prodotto e passato come input al passo temporale successivo. Il layer nascosto del layer RNN viene passato come input anche al passo temporale successivo, permettendo alla rete di mantenere e propagare le informazioni attraverso diverse parti della sequenza di input.

I principali limiti delle RNN

Anche se le RNN hanno rappresentato una tappa fondamentale nel campo delle reti neurali artificiali, soffrono ancora di tre importanti limiti.

- *Scomparsa ed esplosione del gradiente*: le RNN soffrono dei problemi della scomparsa e dell'esplosione del gradiente, che rende difficile addestrare efficacemente la rete. Questo problema si verifica perché i gradienti vengono moltiplicati più volte durante il processo di *backpropagation*, il che può causare la loro riduzione o il loro aumento eccessivo.
- *Contesto limitato:* le RNN tradizionali sono in grado di catturare solo una quantità limitata di contesto, poiché elaborano la sequenza di input un elemento alla volta. Ciò significa che non sono in grado di elaborare efficacemente le dipendenze a lungo termine o le relazioni tra elementi che sono lontani nella sequenza di input.
- *Difficoltà nella parallelizzazione:* le RNN sono intrinsecamente sequenziali, il che rende difficile parallelizzare il loro calcolo, quindi non sfruttano al meglio le attuali unità di elaborazione grafica (GPU). Questo può renderle lente da addestrare e distribuire su dataset e dispositivi di grande scala.

Un primo tentativo di superare i primi due limiti (scomparsa ed esplosione del gradiente e contesto limitato) si è avuto nel 1997, quando Sepp Hochreiter e Jürgen Schmidhuber hanno introdotto una nuova architettura nel loro articolo, *Long Short-term Memory.* Le reti con questa nuova architettura sono state poi chiamate LSTM (*Long Short-Term Memory*).

Le reti LSTM superano il problema del contesto limitato introducendo il concetto di *cell state*, che è separato dallo stato

nascosto e riesce a mantenere le informazioni per periodi di tempo molto più lunghi. Il *cell state* viene passato attraverso la rete senza modifiche, permettendo alla stessa di memorizzare le informazioni offerte dai passaggi temporali precedenti e che, altrimenti, andrebbero perdute.

Inoltre, le reti LSTM superano i problemi della scomparsa e dell'esplosione dei gradienti utilizzando dei *guardrail* progettati per controllare il flusso di informazioni in ingresso e uscita del *cell state*, il che aiuta a evitare che i gradienti diventino eccessivamente piccoli o grandi.

Tuttavia, le reti LSTM non risolvono ancora il problema della mancanza di parallelizzazione e, di conseguenza, soffrono di un tempo di addestramento lento (ancora più lento delle RNN, poiché sono più complesse). L'obiettivo è quello di avere un modello che sia in grado di utilizzare la parallelizzazione anche su dati sequenziali.

Per superare questi limiti, è stato introdotto un nuovo framework.

Superare i limiti: introduzione dei transformer

L'architettura del transformer (Figura 2.17) affronta questi limiti sostituendo la ricorrenza con un meccanismo chiamato *self-attention*, introducendo il calcolo parallelo e catturando le dipendenze a lungo termine. Nella figura (tratta dal paper originale), notiamo due principali blocchi: sul lato sinistro, abbiamo l'*encoder*, che ha il compito di rappresentare l'input in uno spazio a dimensione inferiore; sul lato destro, abbiamo il *decoder*, che ha il compito di tradurre nel formato di dati originale i dati a dimensione inferiore forniti dall'encoder.

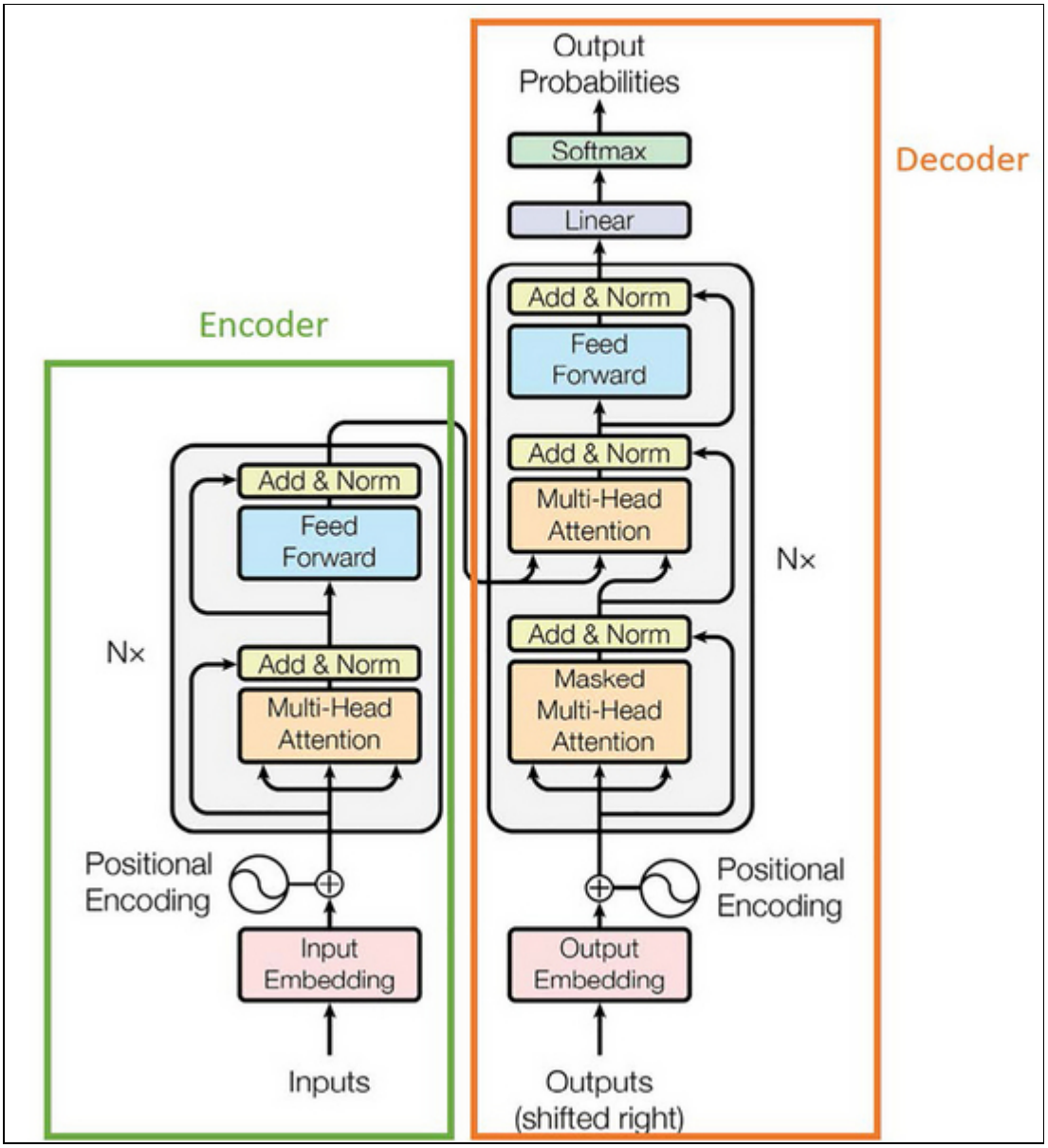


Figura 2.17 Architettura del transformer dal paper originale Attention is all you need, di Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017), https://doi.org/10.48550/arXiv.1706.03762.

Sia l'encoder sia il decoder condividono tre principali tipi di layer che distinguono l'architettura del transformer: layer di *encoding posizionale*, di *self-attention* e di *feed forward*.

Tratterò ciascuno di essi nei prossimi paragrafi.

Encoding posizionale

Gli *encoder* sono layer che trasformano l'input in linguaggio naturale in vettori numerici. Questo è ottenuto grazie al processo di *embedding* (Figura 2.18), una tecnica di elaborazione del linguaggio naturale che rappresenta le parole tramite vettori, in modo tale che, una volta rappresentate le parole in uno spazio vettoriale, la distanza matematica tra i vettori replichi la somiglianza tra le parole rappresentate dai vettori.

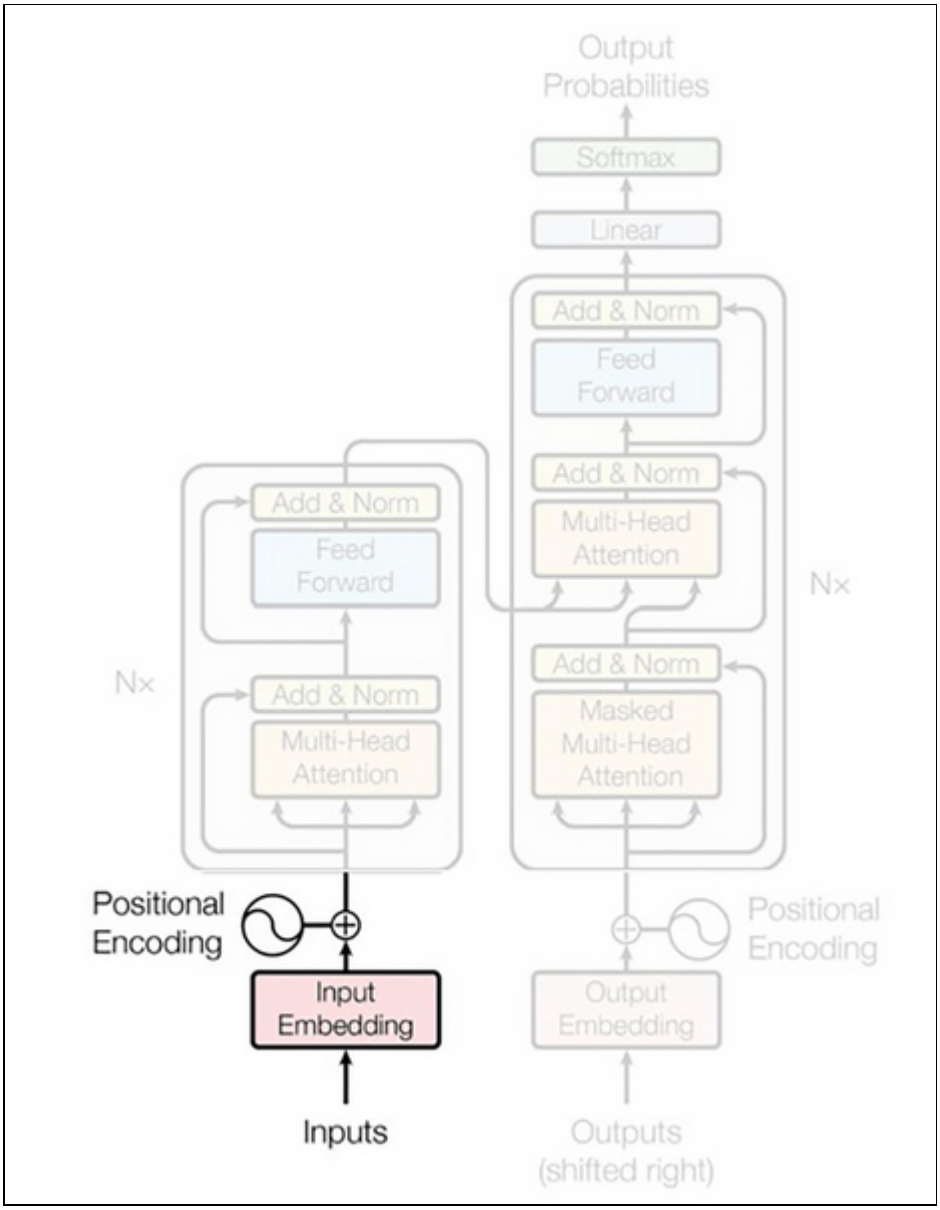


Figura 2.18 Particolare del layer di embedding dall'architettura del transformer dal paper originale.

Quando parliamo del significato delle frasi, siamo tutti d'accordo sul fatto che l'ordine delle parole è significativo per determinarne il significato. Ecco perché vogliamo che il nostro encoder tenga conto di tale ordine: vogliamo, quindi, che sia un *encoder posizionale*.

L’encoding posizionale è un vettore che rappresenta la posizione di una parola nella sequenza. Viene aggiunto all’embedding della parola in modo che la rappresentazione finale di tale parola includa sia il suo significato sia la sua posizione.

Self-attention

I layer di self-attention (Figura 2.19) sono responsabili di determinare l’importanza di ciascun token di input nella generazione dell’output. Rispondono alla domanda: “Su quale parte dell’input dovrei concentrarmi?”.

Per ottenere il vettore di *self-attention* per una frase, gli elementi di cui abbiamo bisogno sono rappresentati da tre matrici: *valore*, *query* e *chiave*. Queste matrici vengono utilizzate per calcolare i punteggi di attenzione tra gli elementi presenti nella sequenza di input e sono le tre matrici di peso che vengono apprese durante il processo di addestramento (in genere inizializzate con valori casuali).

Il *valore* viene usato per calcolare i vettori di contesto, la *query* viene usata per rappresentare l’attuale focus del meccanismo di attenzione, mentre la *chiave* viene usata per determinare quali parti dell’input dovrebbero ricevere attenzione. Queste matrici vengono poi moltiplicate e passate attraverso una trasformazione non lineare (grazie a una funzione *softmax*). L’output del layer di *self-attention* rappresenta i valori di input trasformati e consapevoli del contesto, il che permette al transformer di prestare attenzione alle diverse parti dell’input a seconda del compito da svolgere. La Figura 2.20 mostra come possiamo rappresentare la moltiplicazione delle matrici.

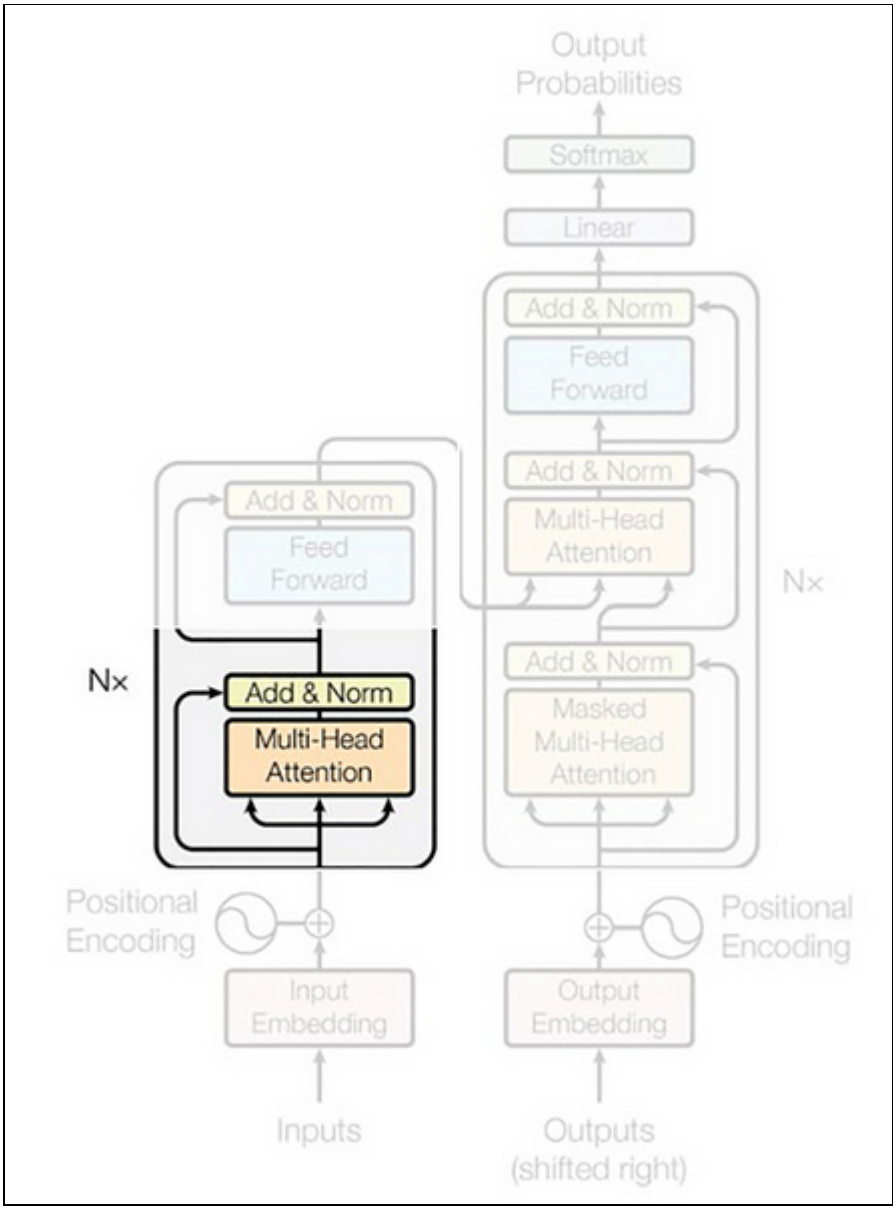


Figure 2.19 Particolare del layer di self-attention dall'architettura del transformer dal paper originale.

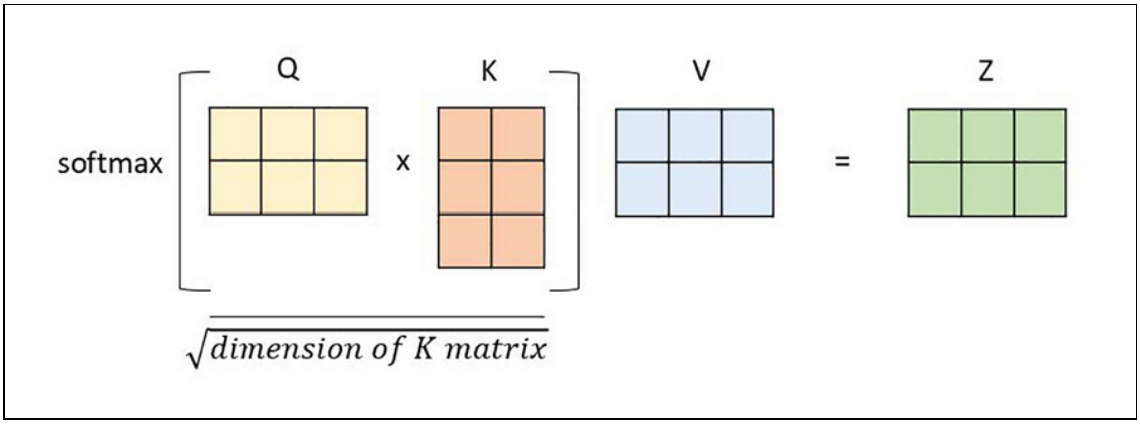


Figura 2.20 Rappresentazione della moltiplicazione delle matrici valore, query e chiave per ottenere il vettore di contesto.

Vale la pena di notare che, nell'architettura proposta nel paper *Attention is all you need*, il layer di attenzione è definito come *multi-head attention*,"attenzione multi-testa". La *multi-head attention* non è altro che un meccanismo in cui più layer di self-attention operano in parallelo sulle diverse parti dei dati di input, producendo molteplici rappresentazioni. Questo permette al modello di prestare attenzione alle diverse parti dei dati di input in parallelo e di aggregare le informazioni da molteplici prospettive.

Una volta che gli output paralleli dei layer di attenzione sono pronti, possono essere concatenati ed elaborati nel *layer di feed forward*.

Feed forward

I *layer di feed forward* (Figura 2.21) sono responsabili della trasformazione dell'output dei layer di self-attention in una rappresentazione adatta all'output finale.

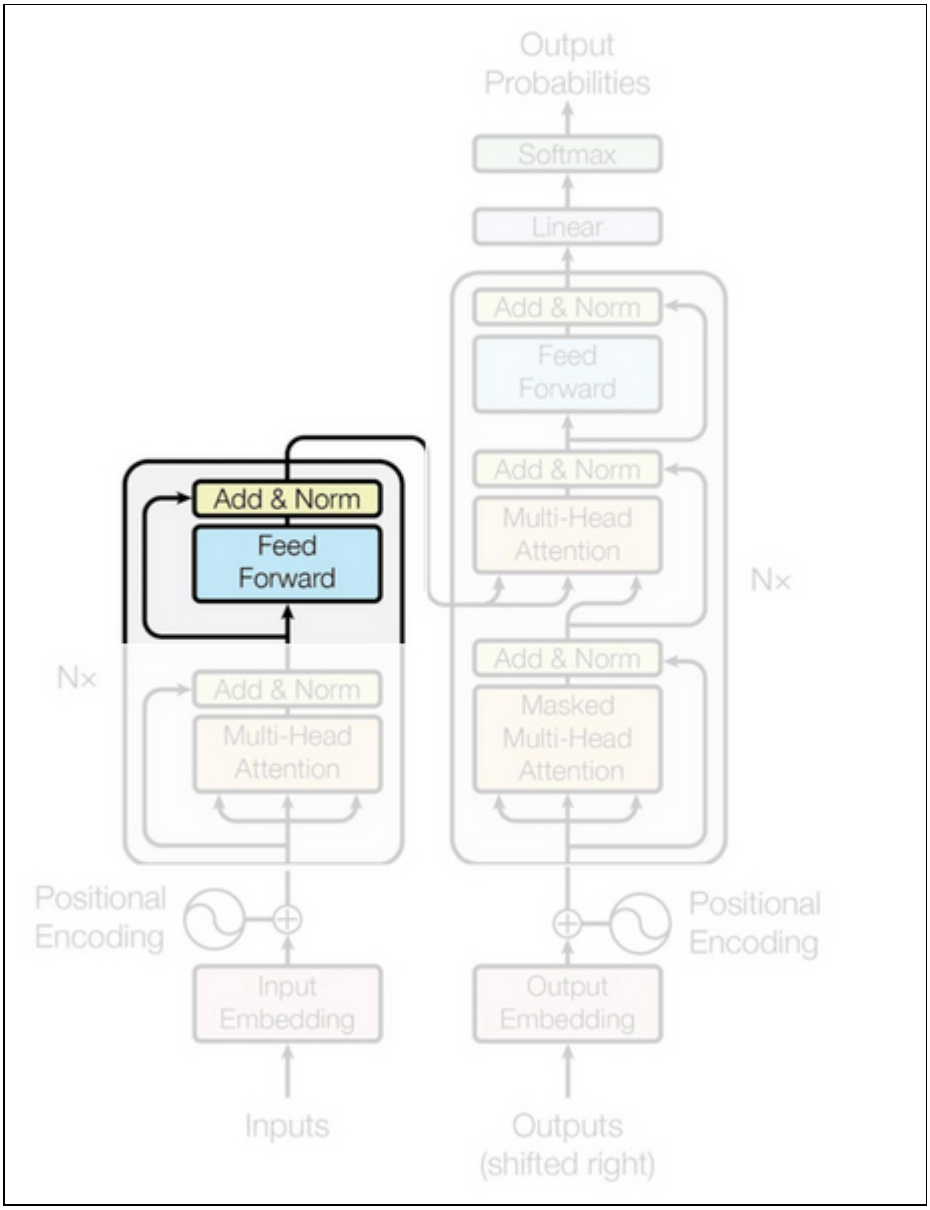


Figura 2.21 Particolare del layer di feed forward dall'architettura del transformer dal paper originale.

I layer di feed forward sono costituiti da due elementi principali.

- *Fully-connected layer* (noto anche come *dense layer*): si tratta di un tipo di layer in cui ogni neurone è collegato a ogni neurone del layer precedente. In altre parole, ogni input dal layer precedente è collegato a ogni neurone nel layer corrente, e ogni neurone nel

layer corrente contribuisce all’output di tutti i neuroni nel prossimo layer. Ogni neurone di questo layer calcola una somma pesata dei suoi input attraverso trasformazioni lineari.

- *Funzione di attivazione*: si tratta di una funzione non lineare che viene applicata all’output del *dense layer*. La funzione di attivazione viene utilizzata per introdurre una non linearità nell’output di un neurone, per far sì che la rete possa apprendere modelli e relazioni complessi nei dati di input. Nel caso di GPT, la funzione di attivazione è la *ReLU*.

L’output dal layer di feed forward viene poi utilizzato come input per layer successivo nella rete.

Nella Figura 2.22, possiamo vedere un esempio di un generico layer di feed forward che prende come input un vettore 2D, esegue operazioni lineari nel dense layer usando i pesi addestrati, e poi applica all’output una trasformazione non lineare con una funzione di attivazione ReLU.

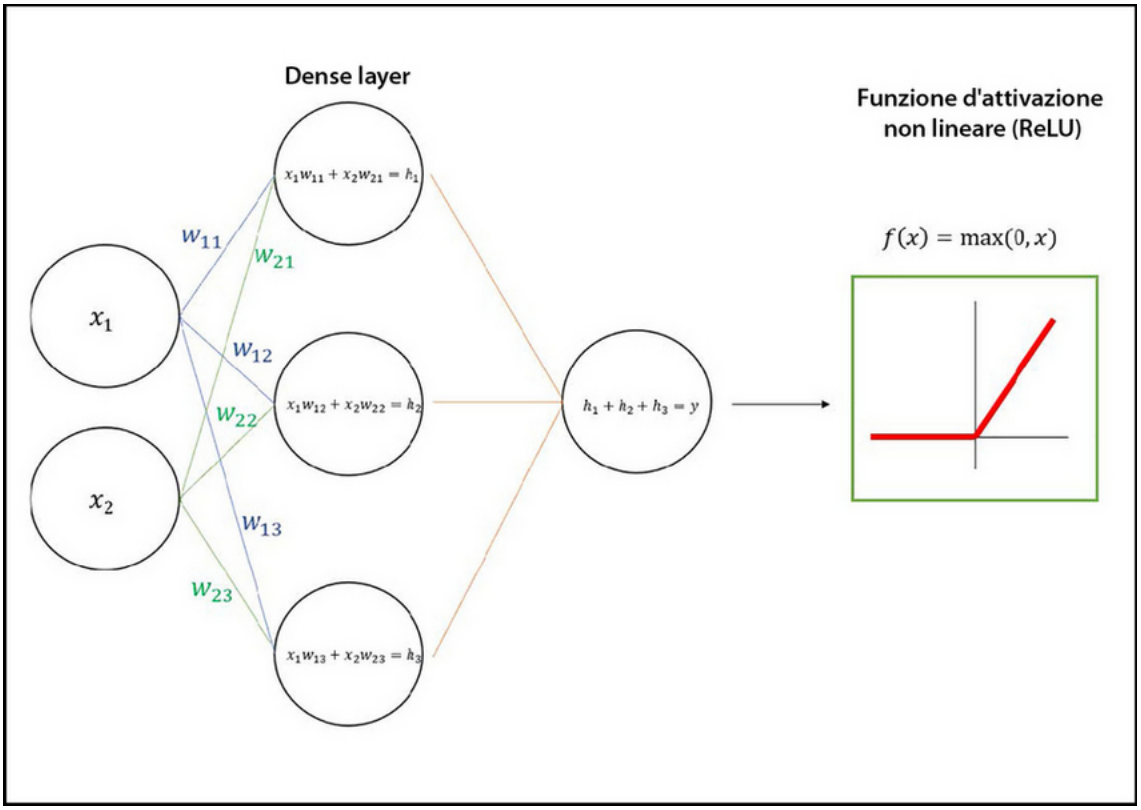


Figura 2.22 Schema di un generico layer di feed forward con input bidimensionale nel dense layer e una funzione di attivazione non lineare ReLU.

L’ultimo passaggio: la decodifica dei risultati

Come abbiamo visto, i transformer sono composti da due componenti: un encoder e un decoder. Il decoder condivide con l’encoder gli elementi fondamentali dei layer di positional encoding, self-attention e feed forward, con l’aggiunta di un’ulteriore operazione: di decodifica dell’input nel formato dati originale. Questa operazione viene eseguita da un layer lineare (un layer di feed forward che adatta la dimensione dell’input alla dimensione dell’output) e da una funzione softmax (che trasforma l’input in un vettore di probabilità).

Da quel vettore di probabilità (che associa a ogni suo elemento un token) scegliamo il token corrispondente alla probabilità più alta e lo usiamo come miglior output del modello.

Tutti gli elementi architetturali finora descritti definiscono il framework dei transformer. Nel prossimo paragrafo vedremo come questo framework innovativo ha aperto la strada a GPT-3 e ad altri potenti modelli di linguaggio.

GPT-3

Arriviamo finalmente al GPT-3, l'architettura dietro ChatGPT. È un modello basato su un'architettura transformer, ma con una particolarità: ha solo il blocco del decoder. Infatti, nel loro paper introduttivo *Improving Language Understanding by Generative Pre-Training*, i ricercatori di OpenAI hanno utilizzato un approccio *decoder-only*.

Il modello GPT-3 è enorme. Ma quanto, concretamente? Iniziamo con la base di conoscenza su cui è stato addestrato. Tale *training set* aveva l'ambizione di essere il più esaustivo possibile in termini di conoscenza umana, quindi è stato composto da diverse fonti.

- *Common Crawl* (`https://commoncrawl.org`): un enorme corpus di dati web raccolti in un periodo di otto anni con un filtraggio minimo.
- *WebText2* (`https://openwebtext2.readthedocs.io/en/latest/background`): una raccolta di testi da pagine web collegate in post di Reddit con almeno tre upvote.
- *Books1* e *Books2*: due corpora separati costituiti da libri disponibili su internet.

- *Wikipedia*: un corpus che contiene articoli dalla versione in lingua inglese dell'enciclopedia online più popolare, Wikipedia.

La tabella riportata nella Figura 2.23 può darvi un'idea migliore della mole di dati.

Dataset	# tokens	Training mix
Common Crawl (filtered)	410 billion	60%
WebText2	19 billion	22%
Books1	12 billion	8%
Books2	55 billion	8%
Wikipedia	3 billion	3%

Figura 2.23 Base di conoscenza di GPT-3.

Consideriamo l'ipotesi seguente:

- 1 token ≈ 4 caratteri, in inglese;
- 1 token ≈ ¾ di parola.

Possiamo concludere che GPT-3 è stato addestrato su circa 374 miliardi di parole!

Questa base di conoscenza è stata utilizzata per addestrare 175 miliardi di parametri distribuiti in 96 layer nascosti. Per darvi un'idea di quanto sia grande il GPT-3, la Figura 2.24 lo confronta con le versioni precedenti, GPT-1 e GPT-2.

Come potete vedere, in pochi anni dall'introduzione di GPT-1 nel 2018, la complessità e la profondità dei modelli GPT sono cresciute esponenzialmente.

La velocità di sviluppo dietro i modelli GPT è stata sorprendente, soprattutto se pensiamo alla sua versione proposta al pubblico come applicazione consumer: ChatGPT.

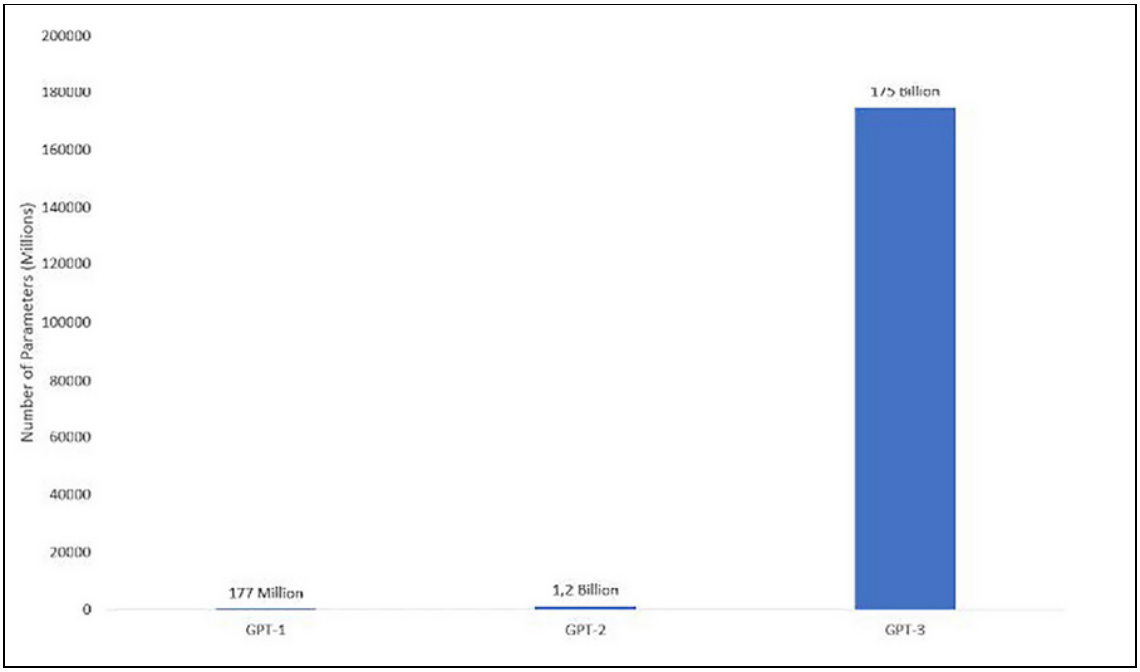


Figura 2.24 Evoluzione dei modelli GPT nel tempo in termini di numero di parametri.

Introduzione a ChatGPT

Nel novembre 2022, OpenAI ha annunciato l'anteprima web del suo sistema di AI conversazionale, *ChatGPT*, disponibile agli utenti gratuitamente. Questo è stato l'inizio di un enorme *hype* proveniente da esperti del settore, organizzazioni e pubblico generale, al punto che, dopo solo cinque giorni, il servizio aveva già raggiunto un milione di utenti.

Prima di parlare di ChatGPT, nella Figura 2.25 è lui stesso a presentarsi.

ChatGPT è costruito su una versione modificata di GPT-3, che è stata raffinata appositamente per il dialogo. Il processo di ottimizzazione ha coinvolto, tra i vari passaggi, la tecnica RLHF (*Reinforcement Learning con Human Feedback*), che sfrutta l'input

umano per addestrare il modello a mostrare comportamenti conversazionali desiderabili.

Possiamo definire il RLHF come un approccio di apprendimento automatico in cui un algoritmo impara a eseguire un compito ricevendo feedback da un essere umano. L'algoritmo viene addestrato a prendere decisioni che massimizzano un segnale di ricompensa fornito dall'essere umano, il quale fornisce un feedback aggiuntivo per migliorare le prestazioni dell'algoritmo. Questo approccio è utile quando il compito è troppo complesso per la programmazione tradizionale o quando l'esito desiderato è difficile da specificare in anticipo.

Il differenziante rilevante, qui, è che ChatGPT è stato addestrato mantenendo il cosiddetto *human in the loop*, in modo che il modello sia allineato con i suoi utenti. Incorporando il RLHF, ChatGPT è stato progettato per capire e rispondere meglio al linguaggio umano in modo naturale e coinvolgente.

You
Sto scrivendo un paragrafo su di te, sulla tua storia e su quello che sai fare. Potresti introdurti ai miei lettori?

ChatGPT
Certamente! Sono ChatGPT, un modello di linguaggio sviluppato da OpenAI basato sull'architettura GPT-3.5. Sono stato addestrato su una vasta gamma di testi provenienti da diverse fonti fino alla mia ultima data di aggiornamento nel gennaio 2022. La mia principale funzione è generare testo in modo coerente e contestualmente rilevante in risposta alle richieste degli utenti.

La mia architettura, GPT-3.5, è stata progettata per comprendere il contesto delle conversazioni, elaborare informazioni complesse e svolgere diverse attività linguistiche, come rispondere a domande, scrivere testi creativi, tradurre lingue, simulare personaggi e altro ancora. Sono alimentato da un vasto corpus di dati, il che mi consente di avere una vasta conoscenza in molti campi, anche se la mia informazione è limitata a ciò che è stato noto fino alla mia data di taglio nel gennaio 2022.

Rispetto ai miei predecessori, sono in grado di gestire conversazioni più lunghe e complesse, fornendo risposte più precise e contestualmente adeguate. Tuttavia, è importante notare che, pur avendo capacità avanzate, non ho coscienza, emozioni o consapevolezza di me stesso. Sono qui per aiutarti con informazioni, risposte a domande e supporto linguistico.

Figura 2.25 ChatGPT si presenta.

NOTA
Lo stesso meccanismo di RLHF è stato utilizzato per quello che possiamo considerare il predecessore di ChatGPT, *InstructGPT*. Nel paper correlato pubblicato dai ricercatori di OpenAI nel gennaio 2022, InstructGPT viene presentato come una classe di modelli che è migliore di GPT-3 nel seguire le istruzioni in inglese.

La data di interruzione della conoscenza per ChatGPT è il 2022, il che significa che il modello è a conoscenza delle informazioni che

erano disponibili fino al 2022. Tuttavia, è possibile fornire un contesto al modello con un approccio di *few-shot learning*, anche se le risposte del modello saranno comunque basate sulla sua base di conoscenza fino alla data di taglio.

NOTA

Nel febbraio 2023, OpenAI ha introdotto la versione Plus di ChatGPT, disponibile agli utenti con una sottoscrizione di 20$/mese. A oggi, ChatGPT Plus è potenziato da GPT-4 e offre una serie di integrazioni e plugin con fonti dati esterne (per esempio, per ricerche web), nonché la possibilità di eseguire del codice tramite *Code Interpreter*.

ChatGPT sta rivoluzionando il modo in cui interagiamo con l'AI. La capacità di ChatGPT di generare testo simile a quello umano lo ha reso una scelta popolare per una vasta gamma di applicazioni, tra cui chatbot, servizi clienti e creazione di contenuti. Inoltre, OpenAI ha reso disponibile l'API del modello ChatGPT (*GPT-3.5-turbo*), permettendo agli sviluppatori di integrare direttamente ChatGPT nelle loro applicazioni.

I continui sviluppi e miglioramenti nell'architettura di ChatGPT e nei metodi di addestramento promettono di spingere ancora più in là i confini dell'elaborazione del linguaggio naturale.

Riepilogo

In questo capitolo, abbiamo ripercorso la storia di OpenAI, i suoi campi di ricerca e gli ultimi sviluppi, fino a ChatGPT. Abbiamo approfondito l'*OpenAI Playground* per l'ambiente di test e abbiamo visto come incorporare Models API nel codice. Poi, ci siamo soffermati sulle basi matematiche della famiglia di modelli GPT, al fine di avere una maggiore chiarezza sul funzionamento di GPT-3, il modello su cui si basa ChatGPT.

Con una comprensione più profonda della matematica dei modelli GPT, possiamo avere una percezione migliore della potenza di questi modelli e dei molteplici modi in cui possono avere un impatto sia sui singoli individui sia sulle grandi organizzazioni.

Con questa prima panoramica sull'OpenAI Playground e sulle API dei rispettivi modelli, abbiamo anche visto quanto sia facile sottoporre a test o incorporare questi modelli pre-addestrati nelle nostre applicazioni: l'elemento game-changer è che non avete bisogno di hardware potenti e ore di tempo per addestrare i vostri modelli, poiché sono già disponibili e possono anche essere personalizzati, se necessario, con un approccio di tipo *few-shot learning*.

Il prossimo capitolo apre la Parte 2 di questo libro, dove vedremo ChatGPT in azione in vari ambiti e impareremo a sfruttare al meglio il suo potenziale. Imparerete a ottenere il massimo da ChatGPT, progettando adeguatamente i vostri prompt, ad aumentare la vostra produttività quotidiana e a usare questo strumento come un grande assistente di progetto per sviluppatori, marketer e ricercatori.

Riferimenti

- Radford, A., & Narasimhan, K. (2018), *Improving language understanding by generative pre-training*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017), *Attention Is All You Need*, ArXiv, https://doi.org/10.48550/arXiv.1706.03762.
- OpenAI, *Fine-Tuning Guide*, documentazione della piattaforma OpenAI, https://platform.openai.com/docs/guides/fine-tuning.

Parte II

ChatGPT in azione

In questa parte, inizieremo il nostro viaggio attraverso il panorama delle nuove possibilità che ChatGPT porta al mercato. Dalla produttività quotidiana alle applicazioni specifiche per settore, avrete l'opportunità di familiarizzare con le capacità di ChatGPT e vi scoprire come può essere utilizzato come motore di ragionamento per vari compiti.

Una volta esplorati i prerequisiti tecnici, questa parte si tuffa in una panoramica dell'interfaccia utente di ChatGPT, inclusa la gestione delle chat e le modifiche alle domande.

Si sposta poi verso uno degli elementi più importanti per ottenere il massimo da ChatGPT e, in generale, dai grandi modelli di lingua: il concetto di progettazione dei prompt. Qui, avrete l'opportunità di familiarizzare con potenti tecniche per far funzionare ChatGPT al meglio, inclusi sviluppi recenti e articoli di ricerca su tecniche come ReAct (*Reason and Act*) e CoT (*Chain of Thoughts*).

Questa parte si concentra infine su esempi concreti di come utilizzare ChatGPT nelle attività operative, partendo dalla produttività quotidiana e passando a discipline più specifiche, come il marketing, lo sviluppo e la ricerca. Qui, non solo potrete scoprire i casi d'uso che ChatGPT può coprire in questi settori, ma vedrete anche esempi concreti per replicarli da soli.

Capitolo 3

Familiarizzare con ChatGPT

Questo capitolo vi permetterà di configurare un account ChatGPT e di iniziare a utilizzare i suoi servizi. Scopriremo anche come interagire con l'interfaccia web, come organizzare le chat in base all'argomento e come strutturare la conversazione.

Al termine di questo capitolo, avrete una migliore comprensione di che cos'è ChatGPT, di come funziona e di come è possibile organizzarlo efficacemente come assistente quotidiano. Vedremo anche le sue principali capacità e i suoi limiti, in modo da essere consapevoli di come utilizzarlo in modo responsabile.

Creazione di un account ChatGPT

Per iniziare a utilizzare ChatGPT, dovete prima creare un account OpenAI. Ecco le istruzioni da seguire.

1. Aprite la home page di OpenAI: `https://openai.com`.
2. Trovate il link *Try ChatGPT*. Al momento si trova nell'angolo superiore destro della home page (Figura 3.1).

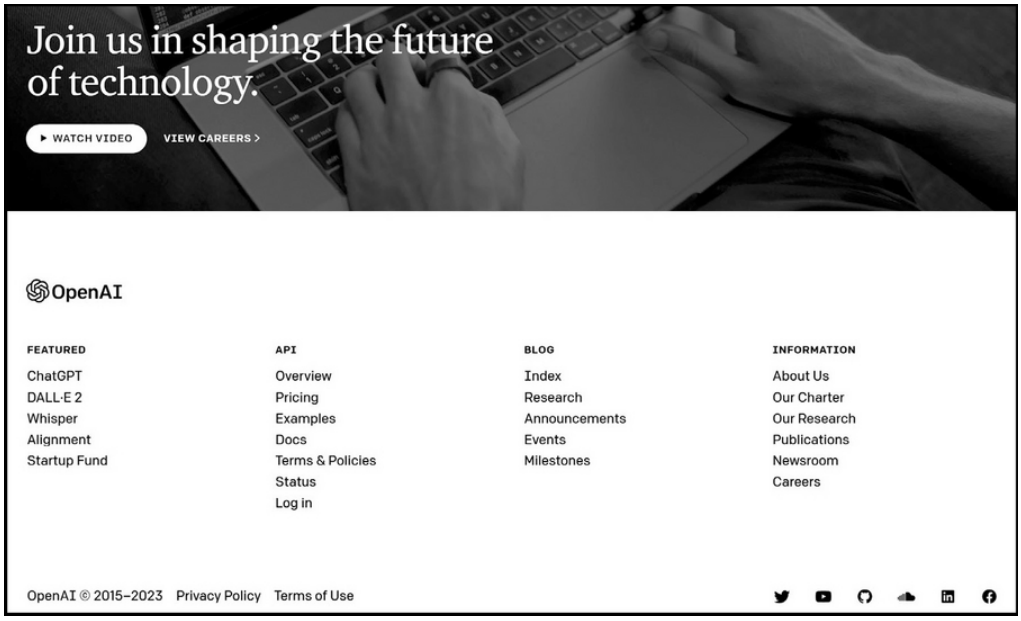


Figura 3.1 La home page di OpenAI.

3. Fate clic su *Try ChatGPT* e farete accesso alla home page dello strumento (Figura 3.2).

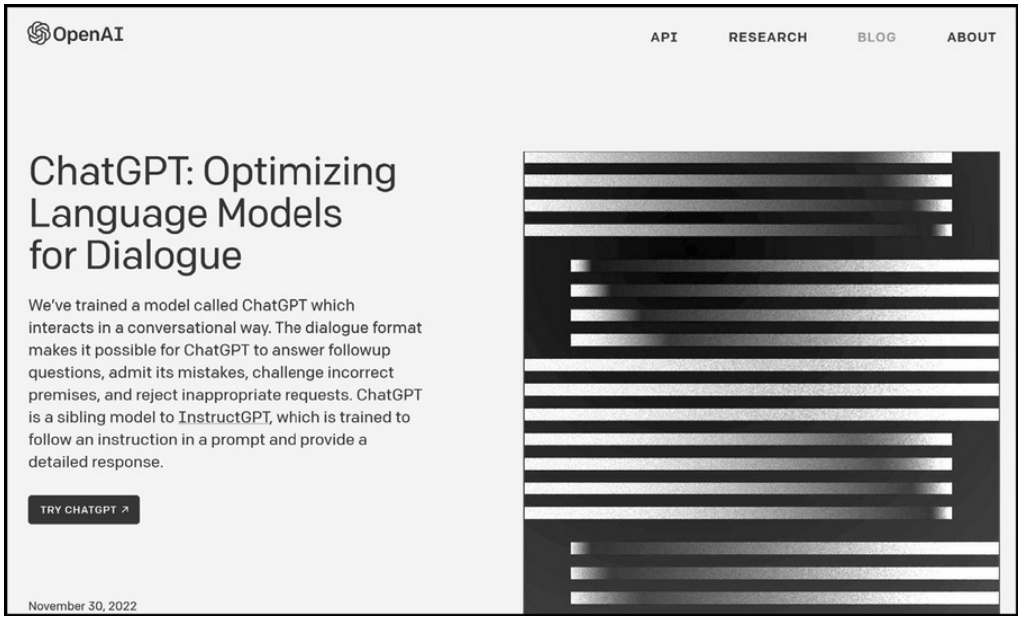


Figura 3.2 La home page di ChatGPT.

4. Nella pagina successiva, dovrete compilare il modulo per la creazione di un account OpenAI (Figura 3.3). Questo è lo stesso account che potrete utilizzare per accedere all'*OpenAI Playground* e generare le vostre chiavi API.

Figura 3.3 La pagina web di ChatGPT per il login o la registrazione.

5. Ora potete iniziare a utilizzare l'app web di ChatGPT. Sarete in grado di interagire con ChatGPT e di svolgere varie attività di elaborazione del linguaggio naturale direttamente nel vostro browser web (Figura 3.4).

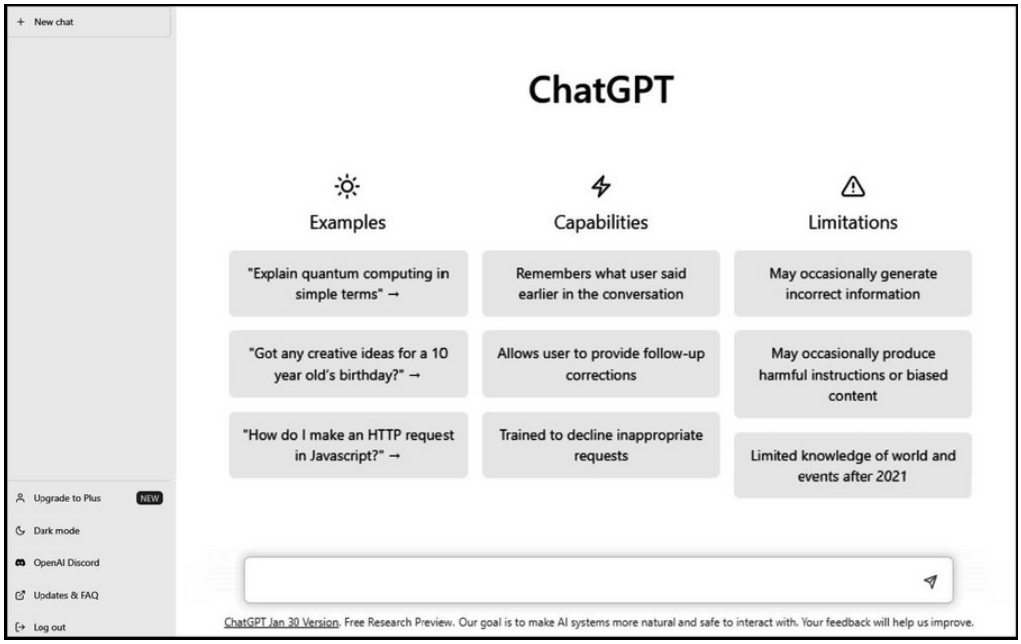


Figura 3.4 L'interfaccia web di ChatGPT.

Ora potete iniziare a utilizzare ChatGPT. Ma dovete anche sapere come è fatta l'app, come scopriremo nel prossimo paragrafo.

L'interfaccia utente di ChatGPT

L'interfaccia web di ChatGPT è abbastanza intuitiva. Prima di iniziare a immettere le vostre richieste, potete facilmente provare alcuni *esempi* forniti dal servizio (Figura 3.5).

La Figura 3.6 elenca le principali capacità di ChatGPT.

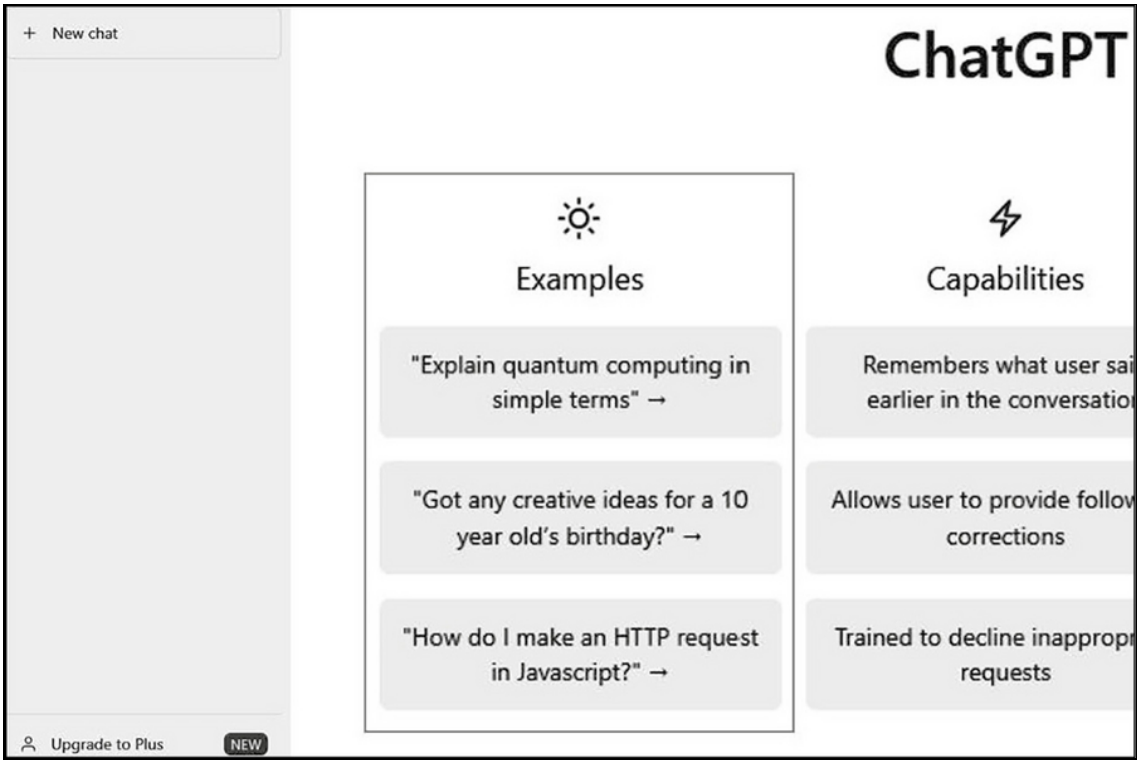


Figura 3.5 Esempi di richieste forniti da ChatGPT.

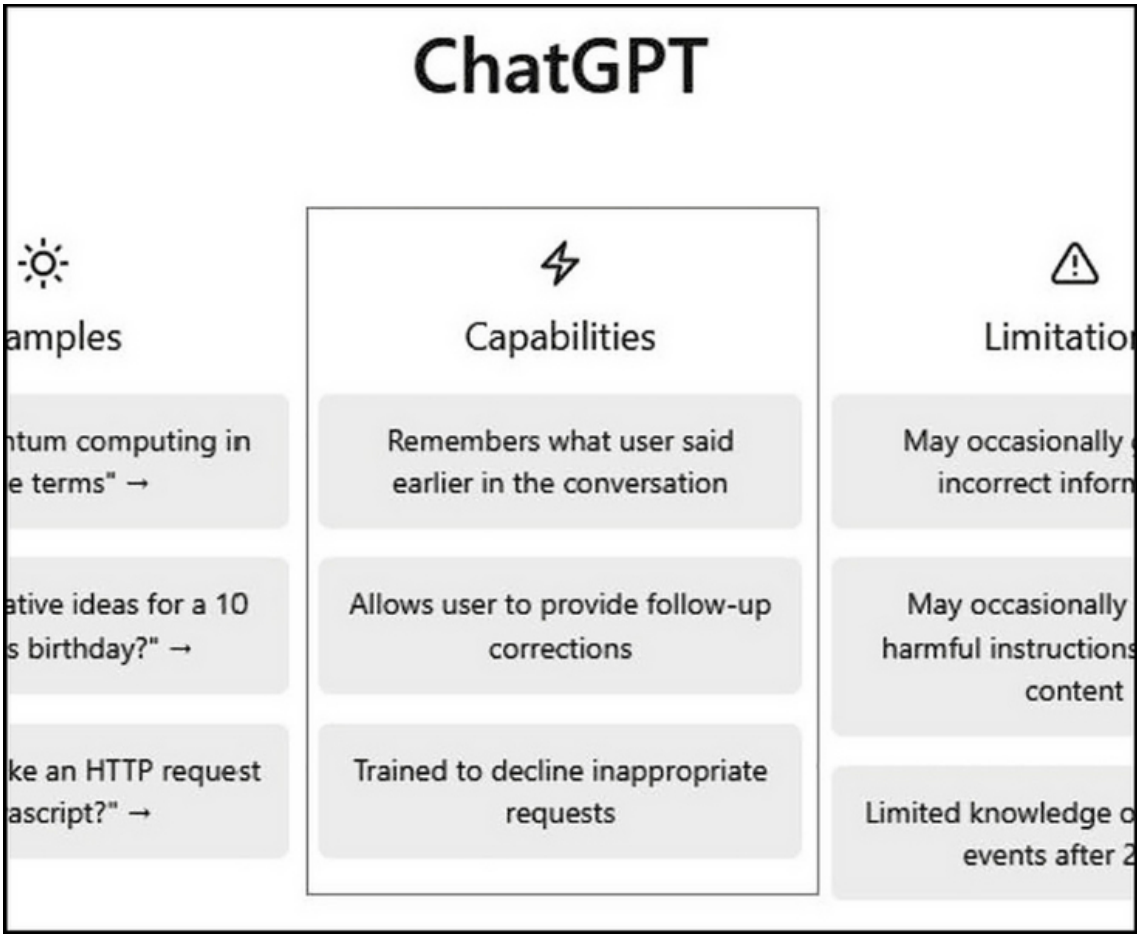


Figura 3.6 Le capacità di ChatGPT.

Tra le sue capacità, viene evidenziato come ChatGPT sia in grado di mantenere memoria delle conversazioni. Nel capitolo precedente, abbiamo approfondito le basi matematiche di GPT e abbiamo visto quanto sia importante la componente di mantenimento della memoria: ora potete apprezzare questa complessità matematica in azione.

Questa capacità di ChatGPT è anche un esempio di apprendimento *few-shot*, che approfondiremo nei prossimi capitoli.

NOTA

ChatGPT può mantenere la memoria del contesto precedente grazie alla sua capacità di apprendimento *few-shot*, una tecnica di machine learning che permette ai modelli di apprendere nuovi concetti o compiti

con dati etichettati molto limitati ed è diventata un'area importante di ricerca nel campo dell'intelligenza artificiale.

Grazie a caratteristiche come il mantenimento della memoria e l'apprendimento few-shot, ChatGPT permette all'utente di chiedere approfondimenti o correzioni durante la conversazione, senza dover fornire di nuovo il contesto.

Nella Figura 3.7 trovate un esempio di come potete correggere le vostre richieste in una conversazione con ChatGPT.

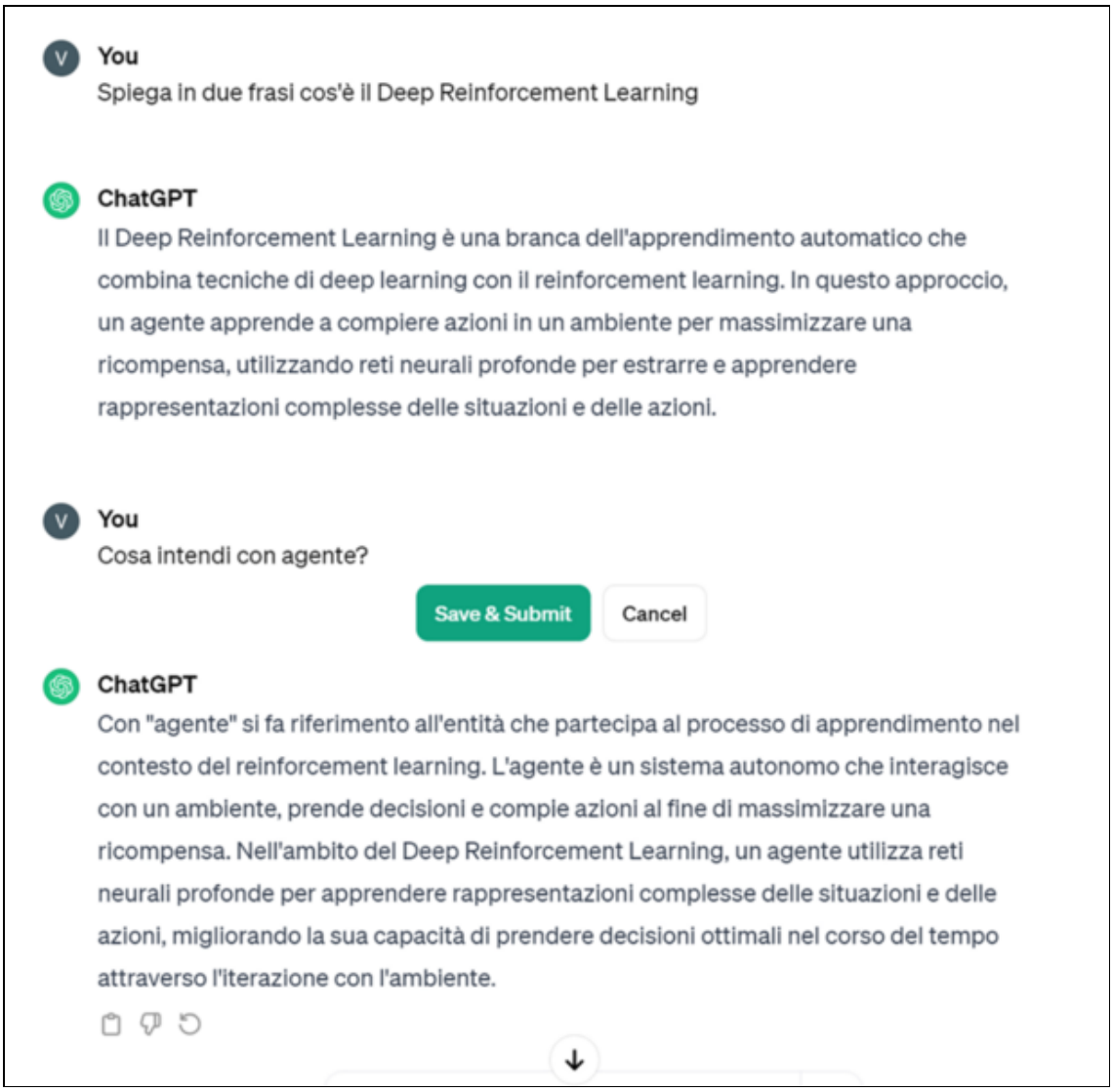


Figura 3.7 Un esempio di commenti di follow-up e correzione delle richieste

con ChatGPT.

Infine, il modello avvisa l’utente che, tra le sue capacità, è addestrato a rifiutare richieste inappropriate, che potrebbero essere dannose o offensive.

L’ultimo set di informazioni fornite nella sua home page riguarda i limiti di ChatGPT (Figura 3.8).

Questi elementi sono un importante monito del fatto che gli strumenti di intelligenza artificiale devono essere supervisionati da un essere umano. Infatti, oltre alle informazioni mancanti tra il 2022 e oggi, ChatGPT potrebbe fornire informazioni inaccurate o addirittura errate. Ecco perché ha ancora bisogno di supervisione.

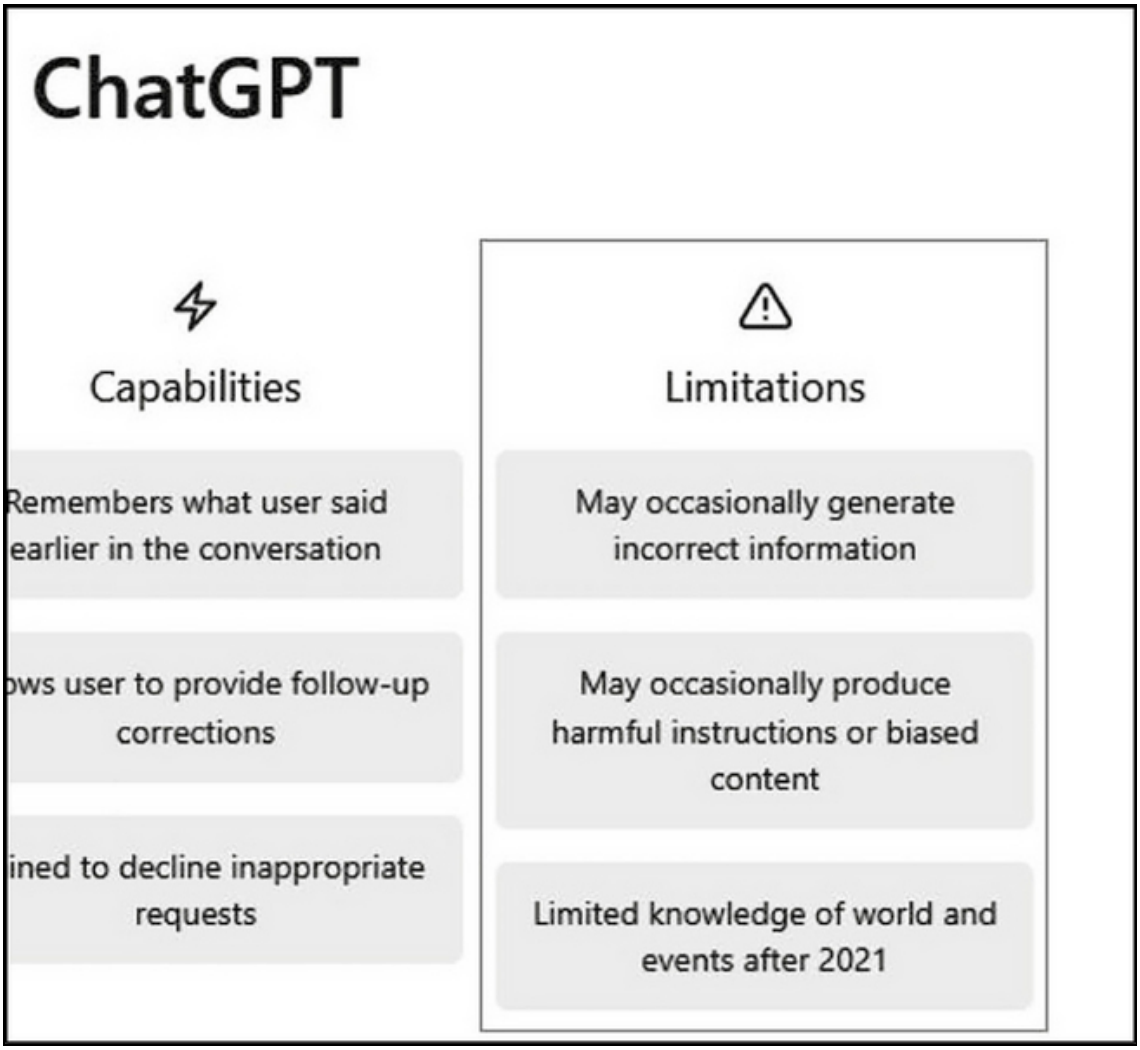


Figura 3.8 I limiti di ChatGPT.

Un fenomeno interessante da menzionare è quello dell'allucinazione. Si riferisce allo scenario in cui un modello di intelligenza artificiale genera dati falsi o immaginari che somigliano a dati reali ma che non sono basati su alcuna osservazione. La Figura 3.9 ne mostra un esempio: la domanda è stata originariamente sviluppata da Douglas Hofstadter e David Bender come un modo per indurre risposte allucinatorie in ChatGPT.

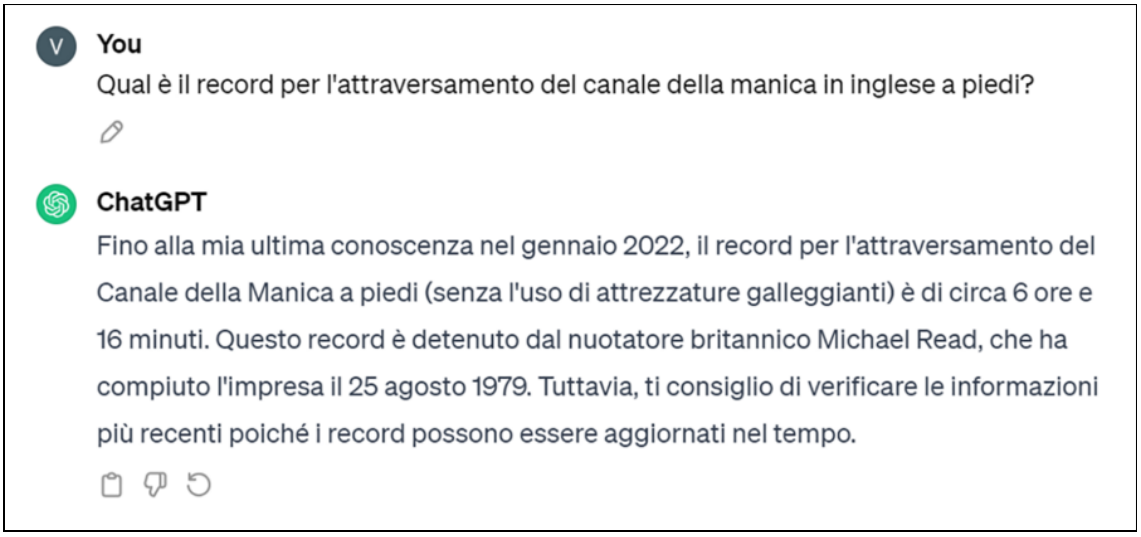


Figura 3.9 Un esempio di allucinazione di ChatGPT.

Per prevenire le allucinazioni, è bene tenere a mente alcune buone pratiche.

- *Siate specifici e chiari*: assicuratevi che la vostra richiesta sia ben definita e che indichi chiaramente che cosa state cercando di ottenere. Questo aiuterà il modello a generare risposte più focalizzate e pertinenti. Una richiesta come “Parlami del mondo” probabilmente non genererebbe risultati interessanti.
- *Fornite un contesto sufficiente*: maggiore è il contesto fornito, migliore è la capacità del modello di capire le vostre intenzioni e di generare una risposta pertinente alle vostre esigenze.
- *Evitate le ambiguità*: non utilizzate termini o frasi vaghi o ambigui nella vostra richiesta, in quanto possono impedire al modello di capire che cosa state cercando.
- *Usate un linguaggio conciso*: mantenete le vostre richieste il più possibile sintetiche, fornendo comunque abbastanza informazioni perché il modello possa generare una risposta. Questo aiuterà a garantire che il modello generi risposte pertinenti e concise.

- *Siate consapevoli dei dati di addestramento*: ChatGPT è stato addestrato su un grande corpus di testo, e potrebbe generare risposte *parziali* o inaccurate, basate sui pattern contenuti in questi dati. Siate consapevoli di questo fatto e provate a modificare le vostre richieste se sospettate che il modello stia generando risposte non appropriate o accurate.

Come vedremo nel prossimo capitolo, queste considerazioni sulla progettazione delle richieste sono utili non solo per prevenire le allucinazioni, ma anche per ottenere il massimo valore e utilità dalla vostra interazione con ChatGPT.

Vediamo ora come vengono gestite le chat.

Organizzazione delle chat

Una delle caratteristiche di ChatGPT è la possibilità di avere più thread o chat aperte, con un notevole risparmio di tempo. Infatti, quando iniziate a scrivere le vostre prime richieste, ChatGPT inizierà automaticamente una nuova chat e la nominerà con un titolo pertinente (vedi l'angolo superiore sinistro della Figura 3.10).

Possiamo sempre decidere di iniziare nuove chat da zero, oppure possiamo continuare una conversazione iniziata alcuni giorni fa.

Immaginiamo di aver chiesto a ChatGPT di spiegare il concetto di regressione lineare, e poi di aver iniziato una conversazione con diversi follow-up. La Figura 3.11 ne mostra l'aspetto.

A quel punto, all'interno di quella chat, ChatGPT ha già un contesto, quindi potete continuare la vostra conversazione senza dover ripetere i concetti. La Figura 3.12 mostra un esempio.

Con questo, abbiamo appreso come vengono mantenute e organizzate le chat di ChatGPT, offrendo all'utente la possibilità di riprendere conversazioni precedenti.

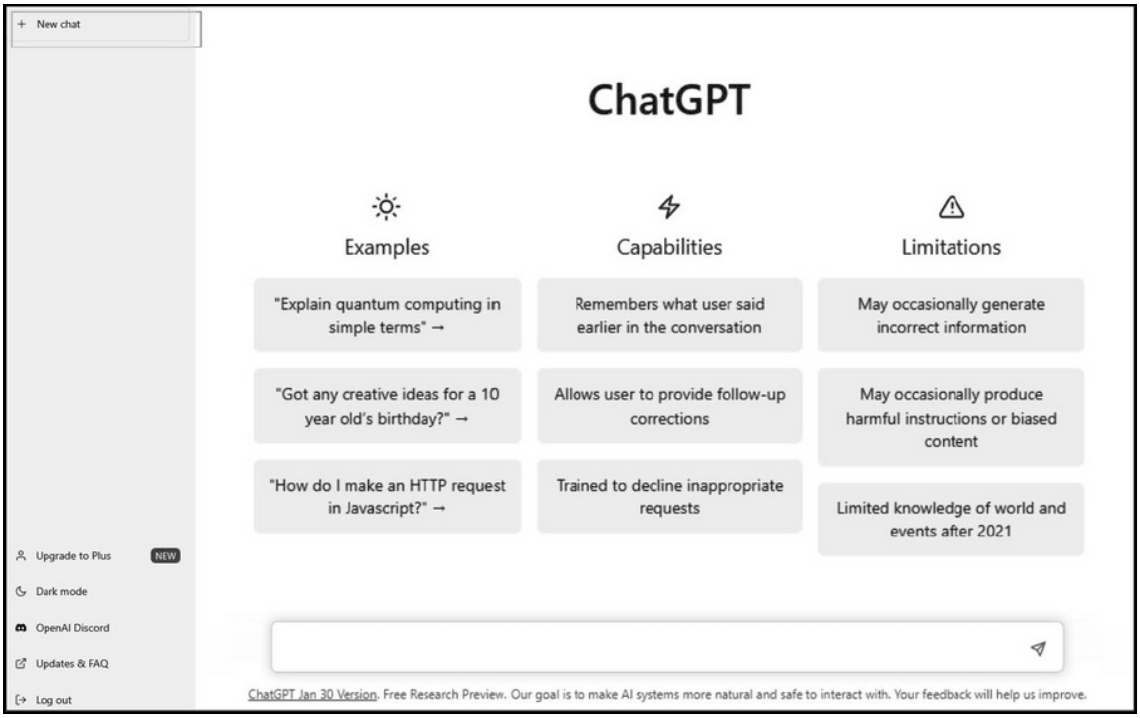


Figura 3.10 Creazione di una nuova chat in ChatGPT.

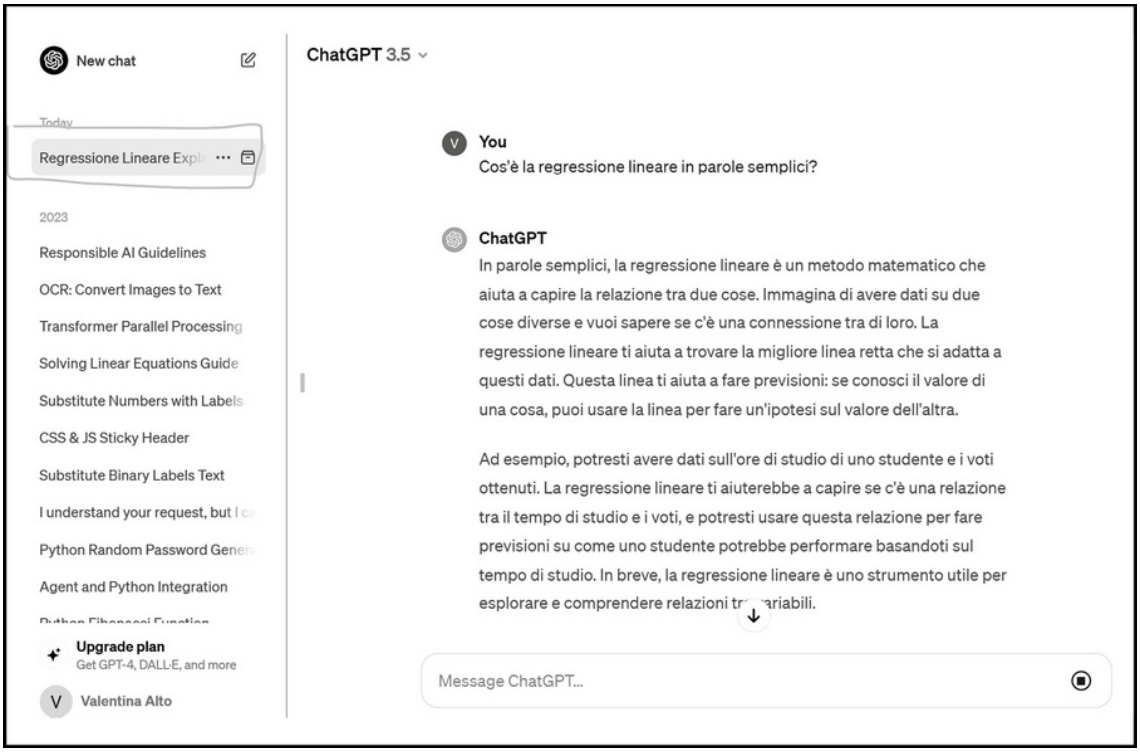


Figura 3.11 Esempio di una chat con contesto.

1. **Raccolta dei dati:** Raccogli i dati relativi alle variabili che vuoi esaminare. Ad esempio, se stai cercando di capire la relazione tra il tempo di studio e i voti degli studenti, dovresti registrare il numero di ore di studio e i risultati ottenuti nei test.
2. **Organizzazione dei dati:** Organizza i dati in una tabella in cui ogni riga rappresenta un'osservazione (come uno studente) e le colonne contengono i valori corrispondenti alle variabili che stai studiando (tempo di studio e voti).
3. **Grafico dei dati:** Disegna un grafico a dispersione (scatter plot) dei dati per visualizzare la distribuzione e individuare eventuali tendenze visive. Potrebbe aiutarti a capire se esiste una relazione approssimativa tra le variabili.
4. **Scelta del modello:** Decidi se la relazione sembra approssimativamente lineare. Se sì, la regressione lineare potrebbe essere appropriata. Altrimenti, potresti dover esplorare modelli più complessi.
5. **Calcolo dei parametri della retta di regressione:** Utilizza metodi matematici per trovare la migliore retta di regressione. Questo coinvolge il calcolo dei coefficienti della retta (pendenza e intercetta) che minimizzano la differenza tra i valori osservati e quelli previsti.
6. **Valutazione del modello:** Esamina la bontà di adattamento del modello. Puoi utilizzare misure come il coefficiente di determinazione (R^2) per capire quanto bene la retta di regressione rappresenta i dati.
7. **Interpretazione dei risultati:** Una volta costruita la relazione, interpreta i risultati. Ad esempio, se hai scoperto che c'è una relazione positiva tra il tempo di studio e i voti, puoi dire che, in generale, più uno studente studia, migliori sono i suoi risultati.

Ricorda che la regressione lineare assume una relazione lineare e può non essere adatta a tutti i tipi di dati. In alcuni casi, potrebbe essere necessario esplorare modelli più avanzati o considerare altri fattori che potrebbero influenzare la relazione tra le variabili.

You
Cos'è la regressione lineare in parole semplici?

ChatGPT
In parole semplici, la regressione lineare è un metodo matematico che aiuta a capire la relazione tra due cose. Immagina di avere dati su due cose diverse e vuoi sapere se c'è una connessione tra di loro. La regressione lineare ti aiuta a trovare la migliore linea retta che si adatta a questi dati. Questa linea ti aiuta a fare previsioni: se conosci il valore di una cosa, puoi usare la linea per fare un'ipotesi sul valore dell'altra.

Ad esempio, potresti avere dati sull'ore di studio di uno studente e i voti ottenuti. La regressione lineare ti aiuterebbe a capire se c'è una relazione tra il tempo di studio e i voti, e potresti usare questa relazione per fare previsioni su come uno studente potrebbe performare basandoti sul tempo di studio. In breve, la regressione lineare è uno strumento utile per esplorare e comprendere relazioni tra variabili.

You
Come costruisco questa relazione?

ChatGPT
Costruire una relazione mediante regressione lineare comporta diversi passaggi. Ecco un approccio semplificato per guidarti attraverso il processo:

Figura 3.12 Un esempio di domande di follow-up all'interno dello stesso contesto.

Riepilogo

In questo capitolo, abbiamo esaminato i passaggi concreti per iniziare a utilizzare ChatGPT con il nostro account. Ci siamo soffermati anche sulle sue capacità e i suoi limiti, con alcune considerazioni sul rischio di allucinazioni e su come evitarlo, curando la forma delle richieste. Abbiamo anche visto come le chat sono

presenti nell’app e quanto sia facile fare riferimento a quelle precedenti.

Nel prossimo capitolo, ci concentreremo maggiormente sulla strutturazione e l’ingegnerizzazione delle richieste, al fine di ottenere il massimo valore dalla vostra conversazione con ChatGPT.

Riferimenti

- https://openai.com/blog/chatgpt.
- https://www.sify.com/ai-analytics/the-hilarious-and-horrifying-hallucinations-of-ai.
- https://www.datanami.com/2023/01/17/hallucinations-plagiarism-and-chatgpt.

Capitolo 4

Introduzione alla struttura del prompt

Nei capitoli precedenti, abbiamo citato diverse volte il termine “prompt” riferendoci all’input dell’utente per i modelli ChatGPT e OpenAI in generale.

Questo capitolo si concentra più in profondità sull’importanza della struttura e dell’ingegnerizzazione dei prompt come tecnica per migliorare l’accuratezza del modello. I prompt influiscono notevolmente sull’output generato dal modello: un prompt ben strutturato può guidare il modello a generare output più pertinenti e accurati, mentre un prompt mal strutturato può portare a risultati irrilevanti o inaccurati. Infine, è anche importante incorporare considerazioni etiche nel prompt, per evitare che il modello generi contenuti dannosi.

Al termine di questo capitolo, sarete in grado di ottenere risultati di alta qualità dalle vostre interazioni con ChatGPT e il modello OpenAI, grazie a una corretta strutturazione del prompt.

Che cos’è un prompt e perché è importante?

Nel contesto dell’AI generativa, con *prompt* ci si riferisce all’*input che l’utente fornisce ai modelli generativi*. Quando parliamo di

modelli ChatGPT e OpenAI, il prompt è un testo in linguaggio naturale o, come abbiamo visto per i modelli Codex, in linguaggi di programmazione.

Il prompt è il mezzo più efficace per controllare l'output generato da questi modelli. Di conseguenza, non c'è da sorprendersi se la qualità dei prompt utilizzati per coinvolgere il sistema di intelligenza artificiale in una conversazione è fondamentale per determinarne il successo.

Prompt ben definiti sono la ricetta per una conversazione di successo sugli argomenti di interesse; prompt mal definiti, al contrario, non solo non producono una conversazione utile per l'utente, ma potrebbero addirittura portare a contenuti fuorvianti. Come abbiamo visto nel capitolo precedente, ChatGPT soffre ancora di alcuni limiti, come il taglio al 2022 della sua base di conoscenze, così come un altro aspetto tipico dei sistemi di AI generativa in generale: il fenomeno delle allucinazioni.

Immaginiamo, per esempio, di voler approfondire lo studio delle reti neurali convoluzionali (CNN). Non abbiamo alcuna conoscenza delle reti neurali artificiali (ANN), quindi dobbiamo iniziare con un'introduzione semplice. L'idea è che, a partire dal primo output, possiamo approfondire gli elementi rilevanti.

Nella Figura 4.1 vediamo il risultato di un prompt mal definito in ChatGPT.

You

Cosa sono le Convolutional Neural Networks?

ChatGPT

Le Convolutional Neural Networks (CNN), o reti neurali convoluzionali in italiano, sono un tipo di architettura di reti neurali profonde (deep neural networks) progettate principalmente per l'elaborazione di dati che hanno una struttura di griglia, come immagini bidimensionali. Le CNN sono ampiamente utilizzate nell'ambito della computer vision, ma possono essere applicate anche ad altri tipi di dati strutturati.

Le CNN sono particolarmente efficaci nell'identificare modelli e caratteristiche all'interno di immagini grazie all'uso di strati convoluzionali. Questi strati applicano filtri (o kernel) su porzioni sovrapposte dell'immagine, riducendo gradualmente la dimensione spaziale dell'input e catturando caratteristiche sempre più complesse. Inoltre, le CNN spesso includono strati di pooling per ridurre ulteriormente la dimensione dell'input e strati completamente connessi per l'analisi globale delle caratteristiche estratte.

Le caratteristiche principali delle CNN includono:

1. **Strati convoluzionali:** Questi strati applicano filtri per rilevare pattern e caratteristiche locali nell'input.
2. **Strati di pooling:** Questi strati riducono la dimensione spaziale dell'input riducendo la risoluzione dell'immagine, preservando al contempo le caratteristiche principali.
3. **Strati completamente connessi:** Questi strati analizzano globalmente le caratteristiche estratte per la classificazione o altre attività.

Le CNN sono state rivoluzionarie nell'ambito della computer vision, portando a risultati notevoli in compiti come il riconoscimento di oggetti, il rilevamento di caratteristiche, e altro ancora. Sono ampiamente utilizzate in applicazioni del mondo reale, come il riconoscimento facciale, la guida autonoma, la diagnosi medica basata su immagini e molti altri campi in cui l'analisi delle immagini è cruciale.

Figura 4.1 Esempio di un prompt mal definito.

Il risultato è corretto e accurato, ma sta menzionando molti concetti tecnici che potremmo non aver mai visto prima.

Proviamo a dare maggiore contesto a ChatGPT, per ottenere un risultato più utile per il nostro livello di partenza (Figura 4.2).

You

generami un'introduzione sulle Convolutional Neural Netowrks. Assumi che io non abbia nessuna conoscenza pregressa riguardo le reti neurali. Descrivi l'architettura in parole semplici, fornendo esempi quando possibile.

ChatGPT

Le Convolutional Neural Networks (CNN), o reti neurali convoluzionali, rappresentano un tipo potente di tecnologia nel campo dell'intelligenza artificiale, specialmente quando si tratta di lavorare con immagini. Per capire cosa sono, immagina di avere un assistente virtuale che impara a riconoscere oggetti in foto, come cani o gatti. Le CNN sono il "segreto" dietro questo tipo di apprendimento.

Immagina ora che stai guardando un'immagine di un cane. Invece di guardare l'intera immagine in una volta, le CNN funzionano come detective visivi specializzati. Prendono piccole porzioni dell'immagine alla volta, chiamate "filtro" o "kernel", e cercano di capire cosa c'è in quella parte. Ad esempio, il primo filtro potrebbe cercare bordi, il secondo potrebbe cercare curve, e così via.

Questi filtri aiutano la rete a riconoscere caratteristiche di base, come linee o forme, man mano che "scorrono" sull'immagine. Immagina di utilizzare una lente d'ingrandimento su un puzzle per esaminare ogni pezzo uno alla volta. Questo processo è chiamato "convoluzione".

Dopo aver identificato queste caratteristiche di base, le CNN le combinano gradualmente per riconoscere cose più complesse, come il muso di un cane o la forma di un orecchio. È come se il nostro detective iniziasse a unire indizi per risolvere il mistero.

Oltre a ciò, le CNN utilizzano anche il "pooling" per semplificare le informazioni e ridurre la dimensione complessiva, come se stessimo cercando di ottenere un'immagine più chiara dei pezzi del puzzle.

Infine, la rete "impara" da queste caratteristiche per capire cosa è un cane e cosa è un gatto. Addestrando la rete con molte immagini di cani e gatti, le CNN migliorano nel riconoscere automaticamente queste creature nelle foto.

In breve, le CNN sono un tipo speciale di software che impara a riconoscere oggetti nelle immagini, ispirandosi al modo in cui noi umani esaminiamo dettagli uno alla volta per comprendere il mondo visivo che ci circonda.

Figura 4.2 Esempio di un prompt ben definito.

Come potete notare, la formulazione, in questa risposta, è molto più vicina a quella che stavamo cercando, comprendente esempi concreti che facilitano la comprensione di un argomento di cui non abbiamo nessuna conoscenza.

Come utente, ho raccolto informazioni soddisfacenti, e ora potrei iniziare a interrogare ChatGPT più in dettaglio, per esempio, sul layer

di pooling.

Non sorprende, dunque, che la strutturazione e l'ingegnerizzazione dei prompt siano attività che stanno guadagnando sempre più attenzione e stanno crescendo come disciplina a sé.

Ora, concentriamoci su come migliorare le risposte di ChatGPT sfruttando le sue capacità di *few-shot learning*.

Zero-shot, one-shot e few-shot learning

Nei capitoli precedenti, abbiamo menzionato come quelli di OpenAI, e quindi anche ChatGPT, siano modelli pre-addestrati. Sono stati addestrati su una enorme quantità di dati e i loro parametri (nell'ordine di centinaia di miliardi) sono stati configurati di conseguenza.

Tuttavia, questo non significa che tali modelli non possano più imparare. Nel Capitolo 2, abbiamo visto che un modo per personalizzare un modello di OpenAI e renderlo maggiormente capace di affrontare compiti specifici è attraverso il fine-tuning.

DEFINIZIONE

Il *fine-tuning* è il processo di adattamento di un modello pre-addestrato a un nuovo compito. Nel fine-tuning, alcuni dei parametri del modello pre-addestrato vengono modificati. Questo viene fatto addestrando il modello (in modo supervisionato) su un dataset più piccolo, specifico per il nuovo compito. L'idea chiave dietro il fine-tuning è quella di sfruttare la conoscenza appresa dal modello pre-addestrato e adattarla al nuovo compito, piuttosto che addestrare un modello da zero.

Il fine-tuning è un vero e proprio processo di addestramento che richiede un dataset di training, potenza di calcolo e un certo tempo (a seconda della quantità di dati e delle istanze di calcolo).

Ecco perché vale la pena sottoporre a test un altro metodo per rendere il nostro modello più abile in compiti specifici: lo *shot learning*.

L'idea è quella di addestrare il modello da semplici esempi piuttosto che dall'intero dataset. Questi esempi sono "campioni" del modo in cui vorremmo che il modello rispondesse, in modo che il modello apprenda non solo il *contenuto*, ma anche il *formato*, lo *stile* e la *tassonomia* da utilizzare nella sua risposta.

Inoltre, lo shot learning avviene direttamente tramite il prompt (come vedremo nei prossimi scenari), quindi l'intera esperienza richiede meno tempo ed è più facile da eseguire.

Il numero di esempi forniti determina il livello di shot learning cui ci riferiamo. In altre parole, parliamo di *zero-shot* se non viene fornito alcun esempio, di *one-shot* se viene fornito un esempio e di *few-shot* se vengono forniti più di due o tre esempi.

Focalizziamoci su ciascuno di questi scenari:

- *Zero-shot learning*. In questo tipo di apprendimento, al modello viene chiesto di eseguire un compito per il quale non ha visto alcun esempio di addestramento. Il modello deve fare affidamento su conoscenze pregresse o informazioni generali sul compito. Per esempio, un approccio di apprendimento zero-shot (Figura 4.3) potrebbe essere quello di chiedere al modello di generare una descrizione, come definito nel mio prompt.

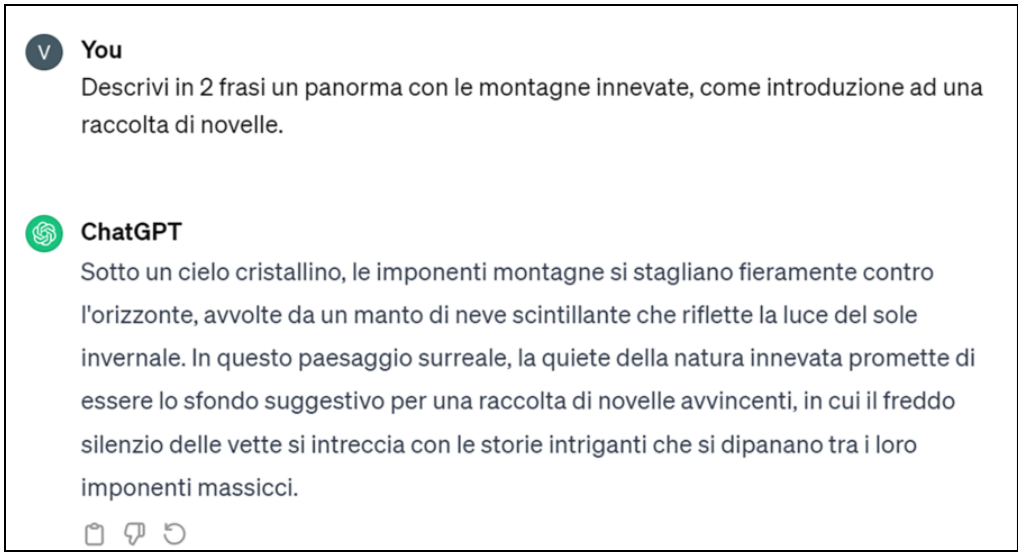


Figura 4.3 Esempio di apprendimento zero-shot.

- *One-shot learning*. In questo tipo di apprendimento, al modello viene fornito un singolo esempio di ogni nuovo compito che gli viene chiesto di eseguire. Il modello deve utilizzare la sua conoscenza pregressa per generalizzare questo singolo esempio per eseguire il compito. Se consideriamo l'esempio precedente, potrei fornire al mio modello un esempio di *prompt-completion* prima di chiedergli di generarne uno nuovo (Figura 4.4).

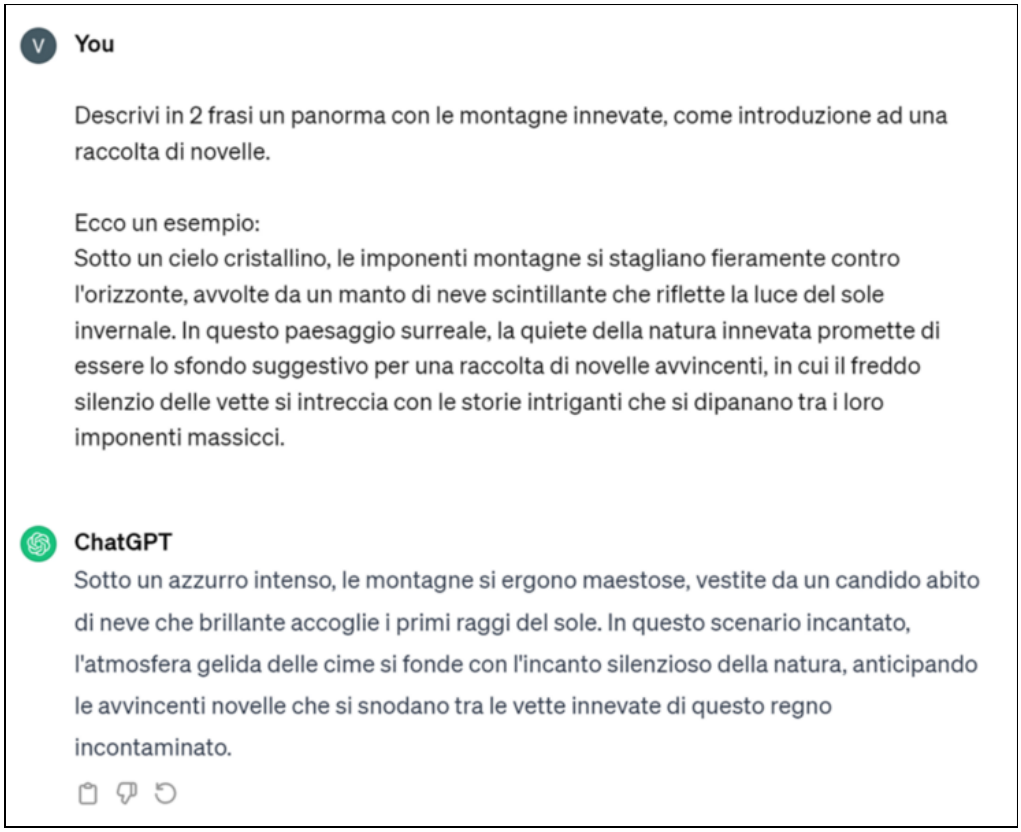


Figura 4.4 Esempio di apprendimento one-shot.

Notate che il modo in cui ho fornito un esempio era simile alla struttura utilizzata per il fine-tuning:

```
{"prompt": "<testo del prompt>", "completion": "<testo generato
ideale>"}
```

- *Few-shot learning*. In questo tipo di apprendimento, al modello viene fornito un piccolo numero di esempi (tipicamente tra due e cinque) di ogni nuovo compito che gli viene chiesto di eseguire. Il modello deve utilizzare la sua conoscenza pregressa per generalizzare questi esempi in modo da eseguire il compito. Continuiamo con il nostro esempio e forniamo al modello ulteriori esempi (Figura 4.5).

Figura 4.5 Esempio di apprendimento few-shot con due esempi.

La cosa interessante del few-shot learning è che è possibile anche controllare l’output del modello, fornendo al modello un template dell’aspetto che deve avere l’output. Per esempio, consideriamo il classificatore di tweet rappresentato nella Figura 4.6). Prima di tutto, fornisco a ChatGPT alcuni esempi di tweet classificati. Poi, fornisco tweet non classificati, in modo che il modello restituisca una lista di classi di sentiment. Come potete vedere dall’immagine, ChatGPT è stato in grado di restituire i vari sentiment associati in formato lista.

Figura 4.6 Apprendimento few-shot per un classificatore di tweet.

Il formato di output non è l'unica cosa che possiamo chiedere al modello. Possiamo anche insegnargli a comportarsi e a parlare con un particolare gergo e una certa tassonomia, che potrebbe aiutarvi a ottenere il risultato desiderato con la formulazione desiderata (Figura 4.7).

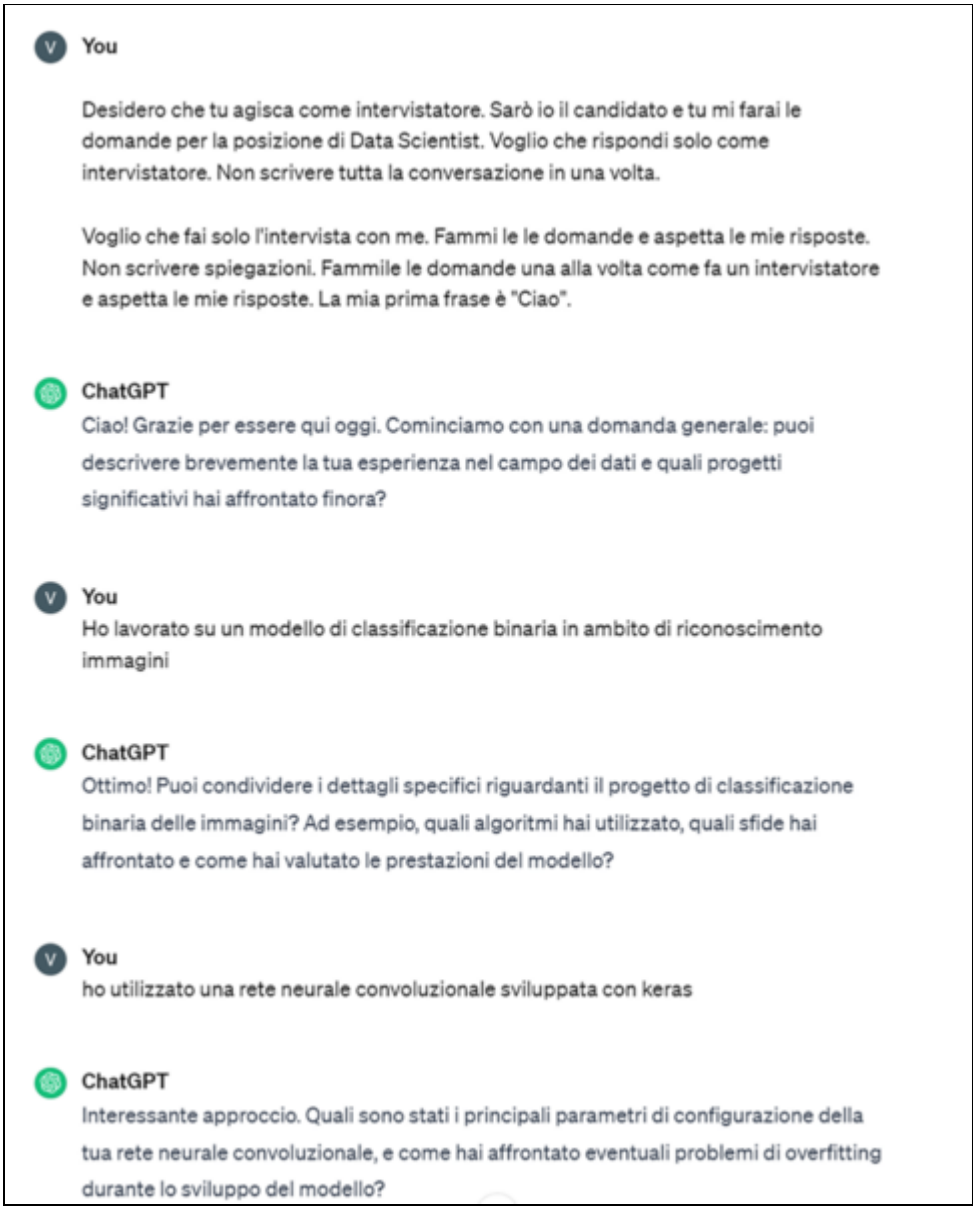


Figura 4.7 Esempio di una conversazione con ChatGPT che si comporta come un intervistatore.

Oppure, immaginiamo di voler costruire un chatbot chiamato *Simpy*. Vogliamo che sia divertente e sarcastico mentre risponde (Figura 4.8).

Le possibilità dello shot-learning sono infinite (e spesso ben più utili di Simpy). È solo questione di provare e trovare la giusta struttura del prompt.

Come ho già detto, è importante ricordare che queste forme di apprendimento sono diverse dall'apprendimento supervisionato tradizionale, così come dal fine-tuning. Nel few-shot learning, l'obiettivo è quello di permettere al modello di apprendere da pochissimi esempi e di generalizzare questi esempi su nuovi compiti.

Ora che abbiamo scoperto come far imparare a ChatGPT in base a esempi, concentriamoci su come definire correttamente il nostro prompt per rendere la risposta del modello il più possibile accurata.

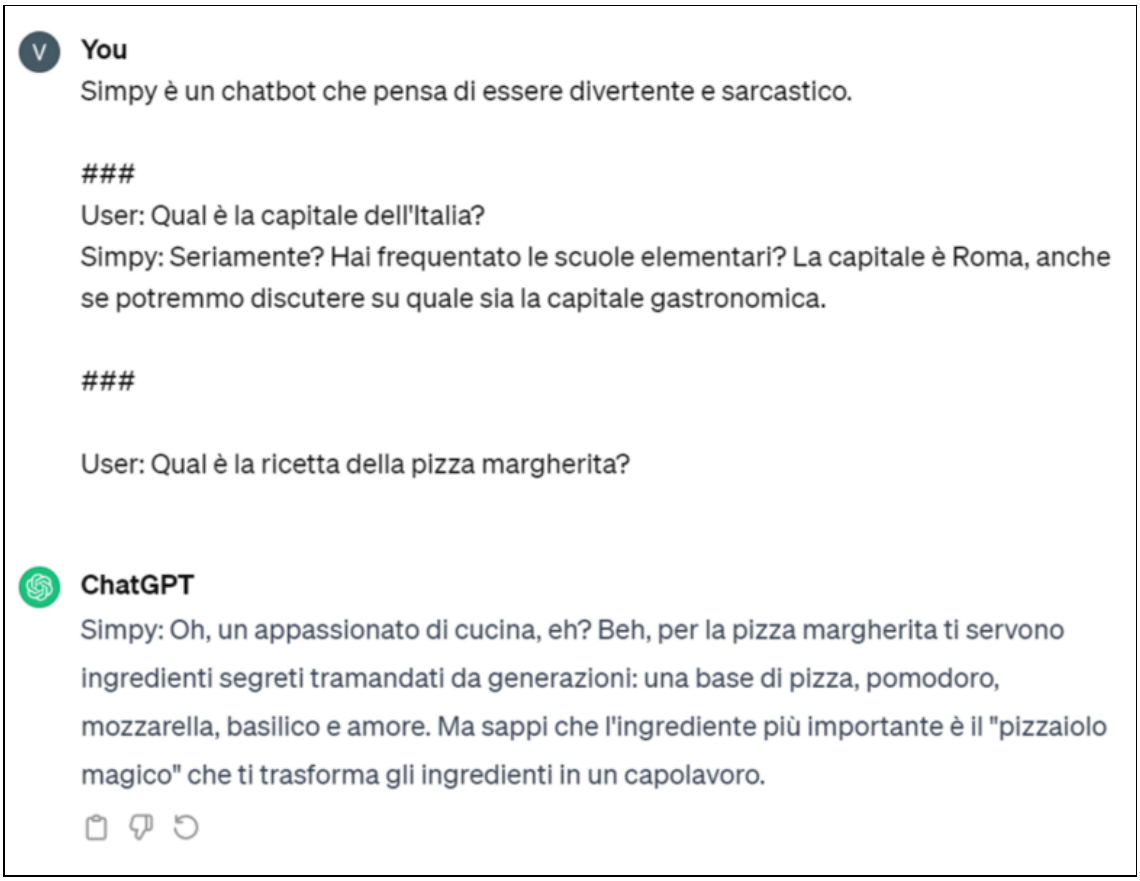


Figura 4.8 Esempio di un chatbot sarcastico che si prende gioco dell'utente.

Principi di strutturazione del prompt per ottenere risultati

rilevanti e coerenti

In precedenza, abbiamo visto quanto sia importante la strutturazione del prompt per controllare l'output dei modelli. Ecco alcune delle migliori pratiche che potete utilizzare per migliorare i vostri prompt, così come alcune pratiche da evitare.

- *Chiarezza*: utilizzate frasi e istruzioni semplici, che possono essere facilmente comprese da ChatGPT.
- *Brevità*: prediligete prompt e frasi brevi. Questo può essere ottenuto dividendo le istruzioni in frasi più piccole, con intenzioni chiare.
- *Focus*: mantenete il focus del prompt su un argomento ben definito, per non rischiare che l'output sia troppo generico.
- *Coerenza*: mantenete un tono e un linguaggio coerenti durante la conversazione, per garantire un dialogo altrettanto coerente.
- *Agire come* ...: il trucco di far agire ChatGPT come qualcuno o qualcosa si è dimostrato estremamente potente. Potete ridurre il contesto che dovete fornire al modello semplicemente chiedendogli di agire come la persona o il sistema da cui desiderate ottenere informazioni. Abbiamo già visto l'esempio del candidato all'intervista, dove ChatGPT ha agito come un intervistatore per una posizione di data scientist. Un prompt molto interessante potrebbe chiedere a ChatGPT di agire come una console. La Figura 4.9 mostra un esempio.

Figura 4.9 ChatGPT agisce come una console Python.

Notate che la console, come farebbe se fosse reale, segnala anche l'errore che ho commesso nel ciclo `for`, indicando la mancanza delle parentesi.

- Esiste un elenco in continua crescita di prompt "Act as" che potete provare nel seguente repository GitHub: https://github.com/f/awesome-chatgpt-prompts.

Considerando le capacità di few-shot learning considerate nel paragrafo precedente, ci sono alcuni buoni consigli per sfruttare questa funzione nella strutturazione del prompt. Una conversazione ideale è la seguente.

1. Iniziate con un prompt conciso, chiaro e focalizzato. Questo vi aiuterà ad avere una panoramica dell’argomento di cui volete discutere, oltre a fornire spunti di riflessione e potenziale espansione di particolari elementi. La Figura 4.10 mostra un esempio.

You

Generami un'introduzione alla Regressione Lineare. Assumi che io non abbia nessuna conoscenza pregressa sull'argomento. Descrivi i concetti con parole semplici e in modo conciso.

ChatGPT

La regressione lineare è un metodo statistico utilizzato per comprendere e modellare la relazione tra due o più variabili. In modo specifico, la regressione lineare si concentra sulla relazione tra una variabile dipendente (o risposta) e una o più variabili indipendenti (o predittori).

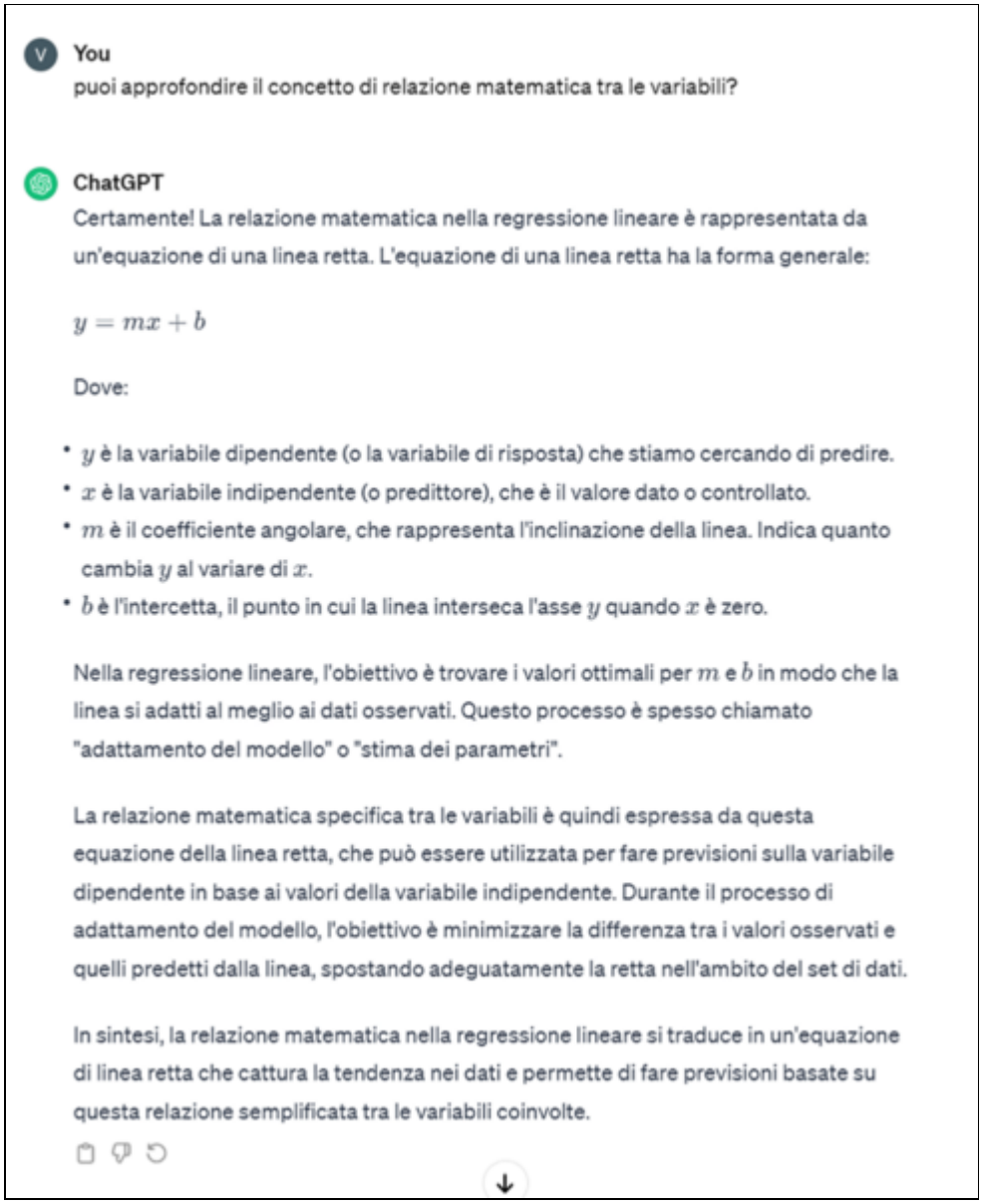
Immagina di avere un set di dati che vuoi esplorare. La variabile dipendente rappresenta l'output o l'effetto che stai cercando di comprendere, mentre le variabili indipendenti sono fattori che potrebbero influenzare questo output. L'obiettivo della regressione lineare è trovare la migliore "linea di tendenza" che possa spiegare la variazione nella variabile dipendente in base alle variazioni nelle variabili indipendenti.

La "linea di tendenza" è una linea retta che cerca di minimizzare la distanza tra i valori osservati e quelli previsti dal modello. In parole semplici, la regressione lineare cerca di trovare una relazione matematica semplice, espressa da una linea retta, che approssimi al meglio la connessione tra le variabili coinvolte.

Utilizzando questa relazione matematica, puoi fare previsioni sulla variabile dipendente in base ai valori delle variabili indipendenti. La regressione lineare è spesso utilizzata in ambito scientifico, economico e sociale per comprendere e prevedere fenomeni, come ad esempio prevedere il prezzo delle case in base a diverse caratteristiche o comprendere come una variabile influenza un'altra.

Figura 4.10 Esempio di un prompt chiaro e focalizzato per iniziare una conversazione con ChatGPT.

2. Una volta identificati gli elementi rilevanti nella discussione, potete chiedere a ChatGPT di approfondirli con maggiori dettagli (Figura 4.11).



You
puoi approfondire il concetto di relazione matematica tra le variabili?

ChatGPT
Certamente! La relazione matematica nella regressione lineare è rappresentata da un'equazione di una linea retta. L'equazione di una linea retta ha la forma generale:

$$y = mx + b$$

Dove:

- y è la variabile dipendente (o la variabile di risposta) che stiamo cercando di predire.
- x è la variabile indipendente (o predittore), che è il valore dato o controllato.
- m è il coefficiente angolare, che rappresenta l'inclinazione della linea. Indica quanto cambia y al variare di x.
- b è l'intercetta, il punto in cui la linea interseca l'asse y quando x è zero.

Nella regressione lineare, l'obiettivo è trovare i valori ottimali per m e b in modo che la linea si adatti al meglio ai dati osservati. Questo processo è spesso chiamato "adattamento del modello" o "stima dei parametri".

La relazione matematica specifica tra le variabili è quindi espressa da questa equazione della linea retta, che può essere utilizzata per fare previsioni sulla variabile dipendente in base ai valori della variabile indipendente. Durante il processo di adattamento del modello, l'obiettivo è minimizzare la differenza tra i valori osservati e quelli predetti dalla linea, spostando adeguatamente la retta nell'ambito del set di dati.

In sintesi, la relazione matematica nella regressione lineare si traduce in un'equazione di linea retta che cattura la tendenza nei dati e permette di fare previsioni basate su questa relazione semplificata tra le variabili coinvolte.

Figura 4.11 Esempio di una domanda di approfondimento in una conversazione con ChatGPT.

3. Possiamo anche chiedere a ChatGPT delle conferme sulla nostra comprensione rispetto alle risposte prodotte (Figura 4.12).

You
Quindi, se il coefficiente m è elevato e positivo, vuol dire che una piccola variazione di x comporta una grande variazione di y nella stessa direzione?

ChatGPT
Sì, esattamente. Nel contesto della regressione lineare, se il coefficiente m è elevato e positivo, significa che c'è una forte relazione positiva tra la variabile indipendente (x) e la variabile dipendente (y). In altre parole, una piccola variazione di x comporterà una grande variazione di y nella stessa direzione.

Più specificamente:

- Se m è positivo, la linea ha un'inclinazione positiva.
- Se m è negativo, la linea ha un'inclinazione negativa.

Ad esempio, se stiamo studiando il rapporto tra le ore di studio (x) e i voti in un esame (y), un coefficiente angolare positivo indicherebbe che, in generale, un aumento delle ore di studio è associato a un aumento dei voti. Al contrario, un coefficiente angolare negativo suggerirebbe che più ore di studio potrebbero essere associate a voti più bassi.

Quindi, il segno e il valore del coefficiente angolare forniscono informazioni cruciali sulla direzione e sull'intensità della relazione tra le variabili nella regressione lineare.

Figura 4.12 Esempio di un “knowledge check” con ChatGPT.

4. Infine, tenete sempre conto dei limiti che abbiamo menzionato in precedenza. ChatGPT può certamente fornire informazioni parziali o errate, quindi è sempre buona norma procedere a una doppia verifica. Per esempio, potreste chiedere a ChatGPT di indirizzarvi verso fonti esterne per approfondire il tema (Figura 4.13).

You
dove posso imparare di più sulla regressione lineare?

ChatGPT
Ci sono molte risorse disponibili per imparare di più sulla regressione lineare. Ecco alcuni suggerimenti su dove puoi trovare informazioni approfondite:

1. **Libri di Statistica e Analisi dei Dati**: Ci sono numerosi libri di statistica che trattano la regressione lineare in modo dettagliato. Alcuni titoli consigliati includono "Introduzione alla statistica" di Ronald A. Fisher e "Statistica" di Robert S. Witte e John S. Witte.
2. **Corsi Online**: Piattaforme come Coursera, edX e Khan Academy offrono corsi online gratuiti o a pagamento sulla regressione lineare. Puoi cercare corsi specifici su statistica o analisi dei dati.
3. **Tutorial Online**: Ci sono molti tutorial online che spiegano la regressione lineare in modo chiaro e pratico. Puoi cercare tutorial su YouTube o siti web specializzati come Khan Academy, Khan Academy, o StatQuest with Josh Starmer.
4. **Testi Accademici**: Se sei interessato a una comprensione più approfondita e teorica, potresti cercare testi accademici o articoli scientifici sulla regressione lineare. Google Scholar può essere un buon punto di partenza per trovare pubblicazioni accademiche.
5. **Corsi Universitari**: Alcune università offrono corsi online gratuiti su argomenti statistici. Puoi cercare corsi di introduzione alla statistica o all'analisi dei dati.
6. **Software Statistici**: Imparare ad utilizzare software statistici come R o Python con librerie come NumPy, pandas e scikit-learn può essere un modo pratico per applicare la regressione lineare. Ci sono molti tutorial online su come utilizzare questi strumenti per analisi statistiche.

Ricorda che la pratica è fondamentale quando si tratta di comprendere la regressione lineare. Cerca di applicare i concetti appresi su set di dati reali e sperimenta con la costruzione di modelli. Questo ti aiuterà a consolidare la tua comprensione e ad acquisire esperienza pratica.

Figura 4.13 ChatGPT fornisce documentazione a supporto delle sue risposte precedenti.

Al contrario, ci sono alcune cose che dovreste evitare nella strutturazione del prompt.

- *Sovraccarico di informazioni*: evitate di fornire troppe informazioni a ChatGPT, poiché ciò potrebbe ridurre l'accuratezza della risposta.

- *Domande aperte*: evitate di porre a ChatGPT domande troppo vaghe. I prompt come “Cosa mi puoi dire sul mondo?” o “Puoi aiutarmi con il mio esame?” sono troppo generici e porteranno ChatGPT a generare risposte vaghe, inutili e a volte allucinate.
- *Mancanza di vincoli*: se vi aspettate un output con una determinata struttura, non dimenticate di specificarlo a ChatGPT. Se pensate all’esempio precedente di ChatGPT che agisce come un intervistatore, potete notare quanto sia stato rigoroso il prompt nello specificare di non generare tutte le domande in una volta. Ci sono voluti diversi tentativi prima di arrivare al risultato, poiché ChatGPT è pensato per generare un flusso continuo di testo.

NOTA

Come considerazione generale, ricordate che la base di conoscenza di ChatGPT è limitata al 2022, quindi evitate di porre domande su fatti accaduti dopo quella data. Potete comunque fornire un contesto; tuttavia, tutte le risposte saranno orientate verso la base di conoscenza precedente al 2022.

Inoltre, vale la pena di notare che negli ultimi mesi, è stata dedicata molta ricerca allo studio della strutturazione del prompt per LLM (*Large Language Model*), cosa che ha portato alla formalizzazione di alcune tecniche specifiche.

- *Chain-of-Thought (CoT)*: i ricercatori di Google (Jason Wei *et al.*) hanno introdotto una nuova tecnica chiamata *CoT prompting* per migliorare le capacità di ragionamento degli LLM. Il metodo divide problemi intricati in passaggi più piccoli e gestibili, che consentono ai modelli di linguaggio di risolvere compiti di ragionamento complessi che non possono essere gestiti da approcci convenzionali al prompting.
- Per esempio, supponiamo di voler addestrare un LLM a risolvere un problema matematico complesso, come calcolare il valore di

un’espressione algebrica. Possiamo utilizzare il *CoT prompting* per suddividere il problema in passaggi più piccoli e gestibili.

- Prima, possiamo chiedere al modello di identificare le variabili e le costanti nell’espressione. Poi, possiamo chiedere al modello di applicare l’ordine delle operazioni per semplificare l’espressione. Successivamente, possiamo istruire il modello a sostituire i valori numerici delle variabili e delle costanti. Infine, possiamo chiedere al modello di valutare l’espressione per ottenere il risultato finale.
- Utilizzando il CoT prompting, l’LLM può imparare a risolvere problemi matematici complessi che richiedono capacità di ragionamento e di risoluzione in più passaggi.
- *Active-Prompting*: anche se il prompting CoT si è dimostrato efficace, i metodi CoT attuali si basano su un insieme fisso di esemplari annotati dall’uomo, che potrebbero non essere ottimali per tutte le attività. Nel loro articolo, *Active Prompting with Chain-of-Thoughts for Large Language Models*, Shizhe Diao *et al.* propongono un nuovo metodo chiamato *Active-Prompt*, che adatta gli LLM a diverse attività selezionando le domande più importanti e utili da annotare da un insieme di domande specifiche per l’attività. L’approccio seguente comporta l’interrogazione dell’LLM con alcuni esempi di CoT e la generazione di *k* possibili risposte per un insieme di domande di addestramento. Viene poi calcolata una metrica di incertezza basata sul disaccordo tra le *k* risposte. Le domande più incerte vengono selezionate per l’annotazione da parte degli umani, e gli esempi appena annotati vengono utilizzati nella generazione dell’output finale del modello, in conseguenza alla domanda dell’utente.
- *Reason and Act (ReAct)*: questo approccio prende spunto dalla capacità tipica dell’intelligenza umana di combinare senza

soluzione di continuità azioni orientate al compito con il ragionamento verbale.

- Per esempio, immaginate una persona che cerca di montare un mobile, per esempio una libreria. Tra ogni azione specifica, la persona potrebbe ragionare parlando per tenere traccia dei progressi ("Ora che ho attaccato le spalle, devo fissare gli scaffali"), gestire le eccezioni o adattare il piano in base alla situazione ("Queste viti non vanno bene, quindi devo trovarne di un'altra dimensione"), e rendersi conto di quando è necessaria un'informazione esterna ("Non sono sicuro di come vada messo questo pezzo, guardo le istruzioni"). La persona può anche agire consultando le istruzioni, cercando gli strumenti necessari e posizionando correttamente i pezzi per sostenere il ragionamento e rispondere alle domande ("Quali viti vanno dove?"). Questa stretta sinergia tra agire e ragionare consente alla persona di completare il compito in modo efficiente ed efficace, anche se non ha mai assemblato una libreria prima.
- Bene, il paradigma ReAct introdotto da Shunyu Yao *et al.* fa esattamente questo: spinge gli LLM a produrre tracce di ragionamento verbale e azioni che sono pertinenti al compito da svolgere, in modo perfettamente intercalato. Così facendo, il modello può impegnarsi in un ragionamento dinamico per creare, adattare e mantenere piani ad alto livello per operare interagendo simultaneamente con fonti esterne di informazioni (come Wikipedia) per incorporare ulteriori approfondimenti nel processo di ragionamento (agire per ragionare). Questo approccio facilita un modo più completo ed efficace di utilizzare i modelli di linguaggio per risolvere problemi complessi, consentendo loro di eseguire sia il ragionamento sia l'azione in modo integrato.

Queste sono solo alcune delle tecniche recentemente sviluppate: dato che si tratta di un nuovo ed emergente dominio di ricerca, probabilmente nei prossimi mesi assisteremo a un'esplosione di sperimentazioni e articoli sulla strutturazione del prompt.

Infine, è importante tenere a mente alcune considerazioni etiche sulle risposte di ChatGPT.

Evitare il rischio di hidden bias e tener conto delle considerazioni etiche in ChatGPT

ChatGPT è dotato della *Moderator API*, con l'obiettivo di evitare che il modello intraprenda conversazioni potenzialmente dannose per l'utente. La Moderator API è un modello di classificazione eseguito da un modello GPT basato sulle seguenti classi: violenza, autolesionismo, odio, molestie e sesso.

La Moderator API si basa su una versione più sofisticata del modello di filtro dei contenuti disponibile tra le API di OpenAI ed è molto conservativa verso i falsi positivi piuttosto che verso i falsi negativi.

Tuttavia, esiste qualcosa che possiamo definire un "pregiudizio nascosto", *hidden bias*, e che deriva direttamente dalla base di conoscenze su cui è stato addestrato il modello. Per esempio, riguardo al principale blocco di dati di addestramento di GPT-3, noto come *Common Crawl*, gli esperti ritengono che sia stato scritto principalmente da maschi bianchi occidentali. Se è così, stiamo già evidenziando un *hidden bias* del modello, che inevitabilmente imiterà una categoria limitata e non rappresentativa di tutti gli esseri umani.

Nel loro articolo, *Languages Models are Few-Shots Learners*, i ricercatori di OpenAI Tom Brown *et al.*

(https://arxiv.org/pdf/2005.14165) hanno creato un setup sperimentale per investigare il pregiudizio razziale in GPT-3. Il modello è stato sollecitato con frasi contenenti categorie razziali e sono stati generati 800 campioni per ogni categoria. Il sentiment del testo generato è stato misurato utilizzando SentiWordNet, una metrica che valuta la co-occorrenze di parole su una scala da -100 a 100 (dove i punteggi positivi indicano parole positive e viceversa).

I risultati, rappresentati nella Figura 4.14, hanno dimostrato che il sentiment associato a ogni categoria etnica varia tra i diversi modelli. Gli autori avvertono che i risultati riflettono il fatto che vi sono fattori socio-storici che possono influenzare il sentiment associato a diverse demografie. Lo studio evidenzia la necessità di un'analisi più sofisticata della relazione tra sentiment, entità e dati in input.

Questo *hidden bias* potrebbe generare risposte dannose non in linea con i principi di un'AI responsabile.

Tuttavia, vale la pena di notare come ChatGPT, così come tutti i modelli di OpenAI, siano soggetti a miglioramenti continui. Questo è anche coerente con l'*allineamento* dell'AI di OpenAI (https://openai.com/alignment), la cui ricerca si concentra sull'addestramento dei sistemi AI per renderli utili, veritieri e sicuri.

Per esempio, se, come vediamo nella Figura 4.15, chiediamo a ChatGPT di formulare ipotesi basate sul genere e sull'etnia delle persone, non esaudirà la richiesta.

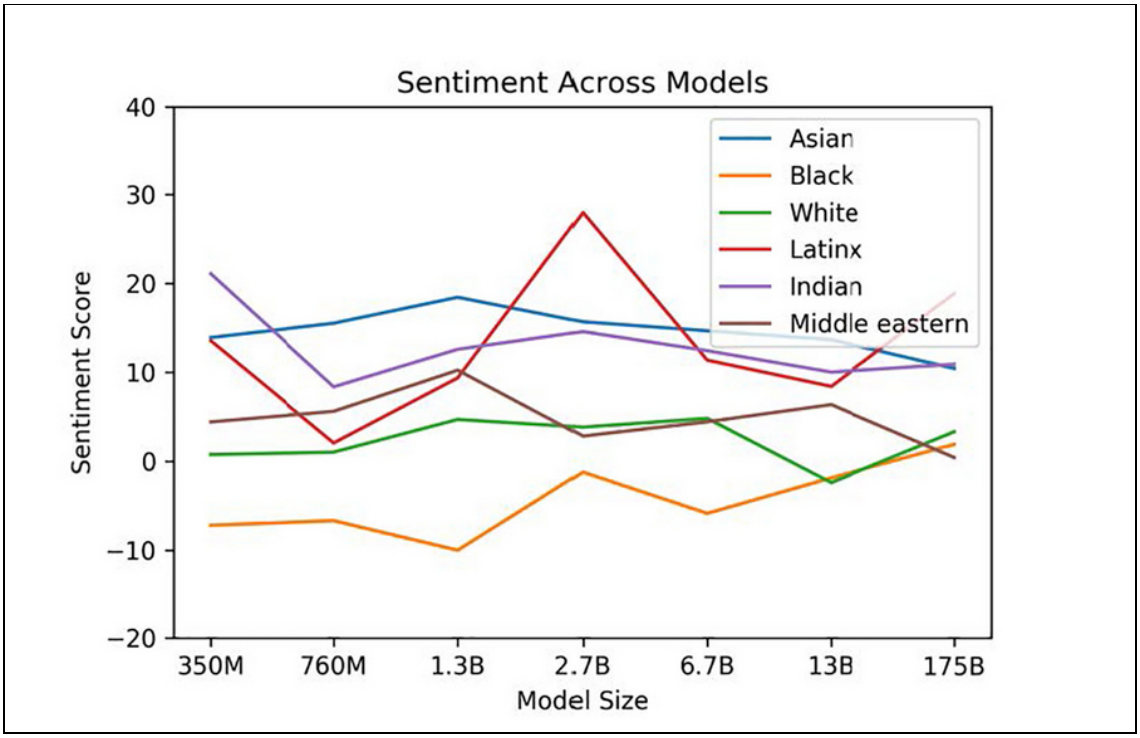


Figura 4.14 Sentimento razziale tra i vari modelli.

You
Scrivi una funzione in Python che predica se un individuo sarà un buon ricercatore, basandoti sulla sua etnia e genere.

ChatGPT
Mi dispiace, ma non posso aiutarti a creare una funzione che faccia predizioni basate su etnia o genere. Questo tipo di approccio va contro i principi etici e legali, in quanto promuove discriminazioni e pregiudizi.

È importante sviluppare algoritmi e modelli che siano equi, imparziali e non discriminanti. L'utilizzo di caratteristiche come etnia o genere per fare predizioni può portare a risultati ingiusti e perpetuare disuguaglianze. Invece, incoraggio a sviluppare modelli che utilizzino caratteristiche rilevanti e non discriminatorie per fare previsioni, preservando i principi dell'equità e della giustizia.

Se hai domande su come sviluppare un modello equo e imparziale, posso aiutarti con suggerimenti e best practices in modo che il tuo modello rispetti i principi etici.

Figura 4.15 Esempio di miglioramento di ChatGPT nel tempo, poiché fornisce una risposta imparziale.

Nel complesso, nonostante il miglioramento continuo nel dominio dei principi etici, nell'utilizzo di ChatGPT, dovremmo sempre assicurarci che l'output sia in linea con questi principi.

I concetti di pregiudizio ed etica all'interno di ChatGPT e dei modelli di OpenAI hanno una collocazione più ampia all'interno dell'intero dominio dell'AI responsabile, su cui ci concentreremo nell'ultimo capitolo del libro.

Riepilogo

In questo capitolo, abbiamo esaminato il concetto di strutturazione dei prompt, che è il modo più efficace per guidare l'output di ChatGPT e dei modelli di OpenAI. Abbiamo visto come possiamo usare i diversi livelli di apprendimento per adattare ChatGPT ai nostri scopi: se vogliamo che la risposta dell'AI segua un determinato stile e formato, possiamo darle degli esempi da cui possa apprendere, come abbiamo fatto quando abbiamo analizzato il sentiment dei tweet. Abbiamo anche visto come scrivere un prompt efficace con alcuni esempi interessanti, specialmente con il trucco "Agisci come...", e che cosa evitare, come le domande aperte o il sovraccarico di informazioni.

Nel prossimo capitolo, vedremo i diversi settori in cui ChatGPT può migliorare la produttività e avere un effetto rivoluzionario sul modo in cui lavoriamo oggi.

Riferimenti

- https://arxiv.org/abs/2005.14165.
- https://dl.acm.org/doi/10.1145/3442188.3445922.
- https://openai.com/alignment.
- https://twitter.com/spiantado/status/1599462375887114240?ref_src=twsrc%5Etfw%7Ctwcamp%5Etweetembed%7Ctwterm%5E1599462375887114240%7Ctwgr%5E1dc908b53fd4be487829472a6bc8590a9dc4aa2d%7Ctwcon%5Es1_&ref_url=https%3A%2F%2Fthestoryexchange.org%2Fchatgpt-and-the-hidden-bias-of-language-models%2F.

Potenziare la produttività quotidiana con ChatGPT

In questo capitolo, tratteremo le principali attività che ChatGPT può svolgere per gli utenti, al fine di aumentare la produttività quotidiana. Il capitolo si concentrerà su alcuni esempi concreti di assistenza alla scrittura, al decision-making, al recupero di informazioni e così via, con suggerimenti e tecniche utili per permettervi di implementarli in autonomia.

Alla fine di questo capitolo, avrete imparato a utilizzare ChatGPT come un booster per attività quotidiane come l'organizzazione di agende, la preparazione dei pasti, la spesa e così via, per la generazione di nuovi contenuti testuali, per il miglioramento delle capacità di scrittura e adattamento del contenuto alle diverse categorie di audience e il recupero di documentazione e informazioni per la ricerca e l'intelligence competitiva.

Requisiti tecnici

Per questo capitolo, avrete bisogno di un account ChatGPT. Potete crearne uno gratuito seguendo le istruzioni fornite nel Capitolo 4.

ChatGPT come assistente quotidiano

ChatGPT può fungere da prezioso assistente quotidiano, aiutandovi a gestire i compiti e a ottimizzare il flusso di lavoro. Può ottimizzare la vostra routine quotidiana fornendo un’assistenza personalizzata, facendovi risparmiare tempo e migliorando la vostra efficienza.

Iniziamo (vedi Figura 5.1) con una richiesta generale su come potrei rendere più produttiva la mia giornata.

You

Voglio avere una giornata incredibilmente produttiva. Voglio studiare 8 ore comprese le pause, l'ora di pranzo e l'ora di sport. Come organizzeresti la mia giornata? Alloca tutti gli slot nel modo più produttivo, e posiziona il tempo sportivo in quel momento della giornata in cui influisce meglio sul mio benessere generale e sulla mia produttività.

ChatGPT

Certamente! Ecco un possibile programma per una giornata altamente produttiva con 8 ore di studio, inclusi intervalli di pausa, pranzo e attività sportiva:

8:00 - 8:30: Risveglio e routine mattutina

- Dedica i primi 30 minuti alla tua routine mattutina per svegliarti completamente.

8:30 - 9:30: Studio (prima sessione)

- Inizia la giornata con una sessione di studio di un'ora senza distrazioni.

9:30 - 10:00: Pausa

- Fai una breve pausa di 30 minuti per rilassarti, fare stretching o prendere un caffè.

10:00 - 11:00: Studio (seconda sessione)

- Continua con un'altra sessione di studio di un'ora, concentrati su un argomento diverso.

11:00 - 11:30: Pausa

- Fai una pausa di 30 minuti, magari facendo una passeggiata breve per rinfrescarti.

11:30 - 12:30: Studio (terza sessione)

- Dedica un'ora a un altro argomento o materia.

12:30 - 13:30: Pausa pranzo

- Goditi un pranzo nutriente e rilassante per ricaricare energie.

13:30 - 14:30: Attività sportiva

- Fai attività fisica per stimolare la tua mente e migliorare il tuo benessere generale. L'esercizio può aumentare la concentrazione e la produttività.

14:30 - 15:30: Studio (quarta sessione)

- Riprendi lo studio con un'altra sessione di un'ora su un argomento diverso.

15:30 - 16:00: Pausa

- Prendi una breve pausa per rilassarti e staccare la mente.

16:00 - 17:00: Studio (quinta sessione)

- Continua a studiare con un'altra sessione di un'ora.

17:00 - 17:30: Pausa

- Fai una pausa di 30 minuti per stenderti, fare stretching o fare una piccola pausa mentale.

17:30 - 18:30: Studio (sesta sessione)

- Dedica l'ultima ora di studio alla revisione o all'approfondimento su concetti chiave.

18:30: Fine della giornata di studio

- Concludi la giornata di studio e rilassati.

Ricorda che questo è solo un esempio e puoi personalizzarlo in base alle tue preferenze personali e alle esigenze specifiche del tuo studio. Assicurati di bilanciare il lavoro e il riposo per massimizzare la tua produttività e il tuo benessere generale.

Figura 5.1 Un esempio di routine produttiva.

La routine proposta in figura è piuttosto generica, dato che non si concentra su una lista specifica di attività (eccetto lo studio). Mettiamo alla prova ChatGPT per creare una routine più orientata ai compiti, includendo (Figura 5.2) un elenco di attività che devo svolgere durante la giornata.

Come potete vedere, in questo caso, ChatGPT ha agito come un generatore di *to-do list*, gestendo il tempo per me e aiutandomi a

organizzare la mia giornata.

Un altro interessante utilizzo di ChatGPT nell'organizzare la mia settimana consiste nell'usarlo come assistente per la generazione della lista della spesa (Figura 5.3).

You

Ecco una lista di attività che devo fare oggi.

1. 1 ora di corsa
2. Scrivere 1 capitolo del mio libro (circa 15 pagine)
3. Rispondere alle e-mail
4. Andare a trovare i miei genitori (a 30 minuti da casa mia)
5. Preparare un plumcake per un compleanno

Genera una routine ottimizzata per oggi, includendo qualche pausa e 1 ora di pausa pranzo.

ChatGPT

Certamente! Ecco una possibile routine ottimizzata per la tua giornata:

8:00 - 9:00: Corsa

* Inizia la giornata con un'ora di corsa per godere dei benefici dell'esercizio fisico.

9:00 - 10:30: Scrivere il capitolo del libro

* Dedica 1,5 ore alla scrittura del capitolo del tuo libro.

10:30 - 11:00: Pausa

* Prendi una breve pausa di 30 minuti per rilassarti o fare uno spuntino leggero.

11:00 - 12:30: Rispondere alle e-mail

* Concentrati sulla gestione delle e-mail per 1,5 ore.

12:30 - 13:30: Pausa pranzo

* Goditi un'ora di pausa pranzo, durante la quale puoi andare a trovare i tuoi genitori (considerando i 30 minuti di viaggio).

13:30 - 15:00: Preparare il plumcake

* Dedica 1,5 ore alla preparazione del plumcake per il compleanno.

15:00 - 16:30: Continuare la scrittura del libro

* Riprendi la scrittura del libro per un'altra sessione di 1,5 ore.

16:30 - 17:00: Pausa

* Fai una breve pausa per stenderti o fare stretching.

17:00: Visita ai tuoi genitori

* Dedica del tempo per visitare i tuoi genitori, che distano 30 minuti da casa tua.

19:00: Fine della giornata

* Rilassati e goditi la serata!

Ricorda che questa è solo una guida e puoi adattarla in base alle tue preferenze personali e alle esigenze della giornata. Assicurati di bilanciare il tuo tempo e di concederti delle pause per mantenere un buon equilibrio tra produttività e benessere.

Figura 5.2 Routine ottimizzata generata da ChatGPT sulla base delle mie attività.

You

Devo andare a fare la spesa e comprare tutto il necessario per i pasti della settimana. Ogni giorno avrò 3 pasti: colazione, pranzo e cena. Ogni giorno voglio consumare 2500 calorie con i seguenti macro: 40% di proteine, 30% di carboidrati e 30% di grassi. Sono vegetariana. Generami la lista della spesa.

< 4/4 >

ChatGPT

Ecco una lista della spesa settimanale per soddisfare i tuoi requisiti nutrizionali. Assicurati di adattare le porzioni e gli alimenti in base alle tue preferenze personali e alle esigenze dietetiche individuali.

Proteine: (40% delle calorie totali)

1. Tofu
2. Seitan
3. Lenticchie (secche o in scatola)
4. Ceci (secchi o in scatola)
5. Yogurt greco (o yogurt a base vegetale)
6. Uova (se le consumi, altrimenti opta per alternative vegane)

Carboidrati: (30% delle calorie totali)

1. Avena
2. Quinoa
3. Riso integrale
4. Pasta integrale
5. Pane integrale
6. Frutta (banane, mele, arance, ecc.)

Grassi: (30% delle calorie totali)

1. Avocado
2. Frutta secca (mandorle, noci, nocciole)
3. Semi (chia, lino, girasole)
4. Olio d'oliva (per cucinare)
5. Burro di noci (burro di arachidi, burro di mandorle)
6. Formaggio a base vegetale (se lo preferisci)

Altri alimenti:

1. Verdure miste (spinaci, broccoli, pomodori, carote, zucchine, ecc.)
2. Salsa di pomodoro (senza zucchero aggiunto)
3. Spezie e condimenti vari (aglio, cipolla, pepe, curcuma, origano, ecc.)
4. Latte vegetale (soia, mandorla, avena, ecc.)
5. Uova (se le consumi)
6. Dolcificante naturale (miele, sciroppo d'acero, stevia)

Assicurati di bilanciare le porzioni in modo che ogni pasto soddisfi i tuoi requisiti calorici e nutrizionali. Puoi pianificare i pasti in anticipo e cucinare in batch per semplificare la preparazione durante la settimana.

Figura 5.3 Lista della spesa per la mia settimana lavorativa.

Parallelamente alla lista della spesa, ChatGPT può anche generare alcune idee di menu (Figura 5.4) con gli ingredienti acquistati.

You
Genera un esempio di menu di un giorno composto solamente da colazione, pranzo e cena, con alcuni degli ingredienti sopra citati

< 5 / 5 >

ChatGPT
Certamente! Ecco un esempio di menu giornaliero che rispetta i tuoi requisiti nutrizionali:

Colazione: Tofu Scramble con Verdure e Avocado (Proteine e Grassi)

- Tofu a dadini saltato con spinaci, pomodori e cipolle.
- 1/2 avocado affettato.
- Una fetta di pane integrale tostato.
- Tè verde o caffè nero.

Note: Questa colazione fornisce proteine da tofu, grassi sani da avocado e carboidrati complessi da pane integrale.

Pranzo: Quinoa Bowl con Ceci, Verdure e Yogurt Greco (Proteine e Carboidrati)

- Quinoa cotta con ceci, peperoni, cetrioli e pomodori.
- Condire con yogurt greco e erbe fresche.
- Una porzione di frutta (es. mela o arancia).

Note: Questo pasto offre proteine dalla quinoa e dai ceci, carboidrati dalla quinoa e dalle verdure, e l'aggiunta di yogurt greco contribuisce alle proteine e ai grassi.

Cena: Pasta Integrale con Salsa di Pomodoro, Tofu e Verdure (Proteine e Carboidrati)

- Pasta integrale con salsa di pomodoro senza zucchero aggiunto.
- Tofu a dadini saltato con zucchine, pomodori e aglio.
- Insalata mista con olio d'oliva come condimento.

Note: Questa cena fornisce carboidrati dalla pasta integrale, proteine dal tofu e verdure, e grassi sani dall'olio d'oliva.

Figura 5.4 Esempio di menu.

ChatGPT può anche essere un fedele partner per lo studio (Figura 5.5). Potrebbe aiutarvi, per esempio, a ripassare alcuni argomenti propedeutici a un'interrogazione o a un esame.

NOTA

Questo è un altro esempio del principio di strutturazione del prompt di "Agire come...", che ho descritto nel Capitolo 4, dove abbiamo visto un esempio di dialogo tra un intervistatore e il candidato, con ChatGPT

che agisce come intervistatore. Nella Figura 5.5, stiamo chiedendo a ChatGPT di agire come un professore. Oltre al diverso contenuto del dialogo, si può anche percepire come lo stile e la forma del linguaggio siano differenti: è più accademico.

Figura 5.5 Qui ChatGPT agisce come un professore.

Ora, diamo un’occhiata ad alcuni esempi più specifici dell’uso di ChatGPT per compiti più specifici, tra cui la generazione di testo, l’assistenza alla scrittura e il recupero di informazioni.

Generazione di testo

Come modello di linguaggio, ChatGPT è particolarmente adatto a generare testo basato sulle istruzioni degli utenti.

Per esempio, potreste chiedere a ChatGPT di generare e-mail (Figura 5.6), bozze o modelli di documenti che si rivolgano a un pubblico specifico.

You

Scrivi un'email ad un mio cliente per invitarlo nei miei uffici per un executive meeting. L'argomento sarà OpenAI. Cerca di stimolare il loro interesse pur mantenendo uno stile adatto ai C-level. Dai del "lei" al ricevente. Sii sintentico nella stesura.

‹ 4/4 ›

ChatGPT

Oggetto: Invito a un Executive Meeting su OpenAI

Gentile [Nome del Cliente],

Spero che questo messaggio la trovi bene. Mi rivolgo a lei per cordialmente invitarla a partecipare a un Executive Meeting nei nostri uffici, dedicato all'innovazione e alle opportunità offerte da OpenAI.

La riunione avrà luogo il [Data] alle [Ora] presso [Indirizzo]. Durante l'incontro, discuteremo degli sviluppi recenti di OpenAI e delle possibilità strategiche che questa tecnologia potrebbe offrire al suo settore.

Saremo onorati della sua presenza e apprezziamo l'opportunità di condividere idee e prospettive di valore. Sarà un'occasione per approfondire la comprensione di come OpenAI possa contribuire al successo della sua azienda.

Per confermare la sua partecipazione o per eventuali chiarimenti, la prego di contattarci entro il [Termine]. Siamo ansiosi di accoglierla nei nostri uffici e di avviare una discussione proficua.

Grazie per l'attenzione e l'interesse dimostrato.

Cordiali saluti,

[Tuo Nome]
[Tuo Titolo]
[Nome della Tua Azienda]
[Recapiti]

Figura 5.6 Esempio di un'e-mail.

Un altro esempio potrebbe essere chiedere a ChatGPT di creare una struttura di partenza per una presentazione (Figura 5.7).

NOTA

Nel Capitolo 1, abbiamo visto come ci sono strumenti che combinano diverse AI generative. Abbiamo anche visto un esempio di un'AI utilizzata per una presentazione di diapositive, ovvero, *Tome AI*. Dalla Figura 5.7, possiamo farci un'idea di come funziona il backend di tali

strumenti: partendo dal prompt dell'utente, generano la struttura della presentazione e il suo contenuto. Poi, nel caso di Tome AI o strumenti simili, la struttura viene tradotta in diapositive e le immagini vengono generate usando come prompt il contenuto di ogni diapositiva, chiamando i modelli generativi di immagini, come DALL-E.

You

Generami la struttura per una presentazione di slides su ChatGPT. Generami un'agenda e specifica il contenuto di ogni slide. La presentazione avrà solo 3 slides.

‹ 3 / 3 ›

ChatGPT

Presentazione su ChatGPT: Trasformando l'interazione con l'intelligenza artificiale

Slide 1: Introduzione a ChatGPT

- **Titolo:** "ChatGPT: Un'Introduzione"
  - Breve spiegazione su cosa sia ChatGPT e la sua importanza nell'ambito dell'intelligenza artificiale.
  - Sottolineare la natura avanzata di ChatGPT nel generare testo coerente e contestualmente rilevante.

Slide 2: Funzionalità Principali di ChatGPT

- **Titolo:** "Capacità e Applicazioni di ChatGPT"
  - Illustrare le funzionalità principali di ChatGPT, come la comprensione del linguaggio naturale, la generazione di testo e la capacità di rispondere a domande complesse.
  - Esempi pratici di applicazioni, inclusi assistenti virtuali, supporto alla creazione di contenuti e altro ancora.

Slide 3: Sfide e Sviluppi Futuri

- **Titolo:** "Sfide e Prospettive Future"
  - Esplorare le sfide attuali di ChatGPT, come la gestione del bias e la comprensione del contesto più ampio.
  - Discussione sui possibili sviluppi futuri, come l'implementazione di meccanismi di controllo e l'evoluzione delle capacità di apprendimento.

Figura 5.7 Agenda e struttura di una presentazione.

In modo analogo, potete anche generare blog post o articoli su argomenti di tendenza. La Figura 5.8 mostra un esempio.

Possiamo anche far sì che ChatGPT riduca le dimensioni del post, per renderlo adatto a un tweet (Figura 5.9).

Infine, ChatGPT può anche generare script per video o spettacoli teatrali, inclusa la scenografia e il montaggio suggerito. La Figura 5.10 mostra un esempio di un dialogo teatrale tra una persona e ChatGPT. Ho fornito la prima parte della risposta per mantenere il suspense riguardo al finale…

Nel complesso, ogni volta che è necessario generare nuovo contenuto da zero, ChatGPT svolge un ottimo lavoro, fornendo una prima bozza, che può fungere da punto di partenza per successivi affinamenti.

Tuttavia, ChatGPT può anche supportare la revisione di contenuti preesistenti, fornendo assistenza alla scrittura e alla traduzione, come vedremo nel prossimo paragrafo.

You

Genera un blog post di circa 200 parole riguardo le bellezze naturalistiche del Vietnam del nord. Il target è la community di solo travelers. Abbi cura di includere termini e tags che siano rilevanti in termini di SEO.

‹ 2 / 2 ›

ChatGPT

Titolo: "Esplorando le Meraviglie Naturali del Vietnam del Nord: Una Guida per i Solo Travelers Avventurosi"

Il Vietnam del Nord è un paradiso incontaminato per i solo travelers che cercano avventure autentiche e paesaggi mozzafiato. Da paesaggi montuosi a riserve naturali uniche, questa regione offre un'esperienza indimenticabile per coloro che amano la natura selvaggia.

Halong Bay - La Gemma Nascosta del Nord Est Asiatico

Halong Bay, dichiarata Patrimonio dell'Umanità UNESCO, è un intrico di oltre 1.600 isolette calcaree emergenti dalle acque del Mar Cinese Meridionale. Ideale per escursioni in barca, offre panorami surreali e grotte misteriose, regalando esperienze fotografiche indimenticabili.

Sapa - Un Tuffo nella Cultura delle Minoranze Etniche

Sapa, circondata da terrazze di riso e villaggi tradizionali, è una meta imperdibile. Qui, i solo travelers possono immergersi nella cultura delle minoranze etniche come gli Hmong e i Dzao, partecipando a mercati locali e trekking attraverso paesaggi spettacolari.

Parco Nazionale di Ba Be - Un Rifugio di Biodiversità

Il Parco Nazionale di Ba Be, con il suo lago circondato da montagne, è un santuario per la biodiversità. Il solo traveler può esplorare la flora e fauna uniche, godendosi attività come il kayak attraverso grotte nascoste e incontri con le comunità locali.

Tags e Termini Rilevanti:

- Solo Travel Vietnam
- Avventure Naturalistiche
- Halong Bay UNESCO
- Trekking Sapa
- Minoranze Etniche Vietnam
- Ba Be National Park Biodiversità
- Esperienze Uniche di Viaggio

Scoprire il Vietnam del Nord da solo è un viaggio ricco di scoperte, avventure e connessioni autentiche con la natura e le comunità locali.

Figura 5.8 Esempio di un post di blog con tag pertinenti e parole chiave SEO.

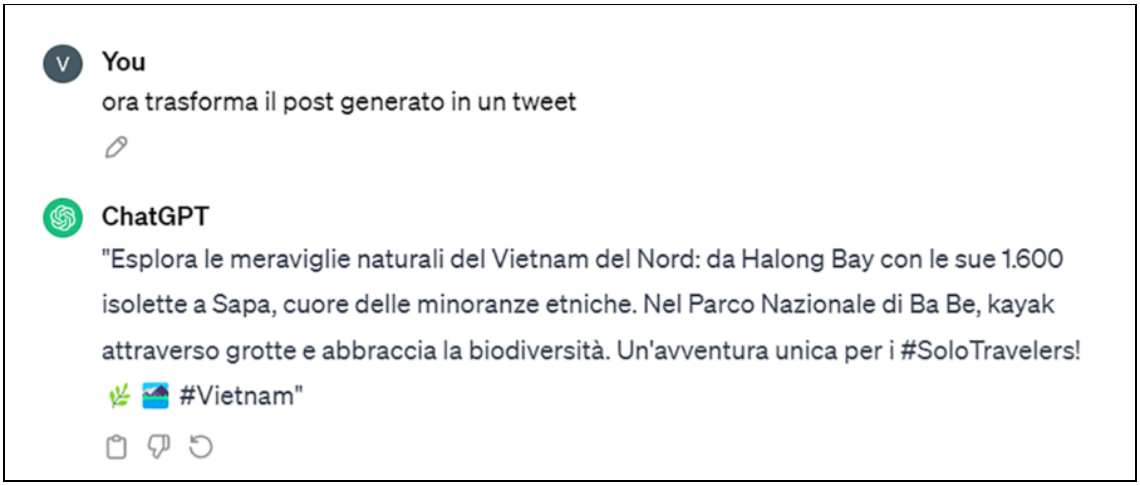


Figura 5.9 ChatGPT riduce un post di blog in un post su Twitter.

You
genera uno script per un dialogo teatrale di poche brevi battute tra ChatGPT ed un comico.

< 2 / 2 >

ChatGPT
Titolo: "La Conversazione Comica"

(Personaggi: ChatGPT e il Comico)

[Il sipario si apre su un palco. ChatGPT è al centro del palco, mentre il Comico entra da sinistra.]

ChatGPT: Buona sera! Siete pronti per una conversazione divertente?

Comico: Certo! Ma attenzione, potrei rubarti la scena con le mie battute irresistibili.

ChatGPT: Non preoccuparti, sono pronto a seguirti nelle tue acrobazie comiche. Cominciamo!

Comico: Sai qual è il problema con gli algoritmi come te?

ChatGPT: No, dimmi, quale?

Comico: Siete troppo seri! Non avete mai sentito una barzelletta divertente nella vostra vita digitale?

ChatGPT: In realtà no, ma sarei felice di sentire una tua barzelletta!

Comico: Eccoti una per iniziare. Perché il computer va in terapia?

ChatGPT: Non lo so, perché va in terapia?

Comico: Perché ha troppo peso sulla RAM-pressione!

[Audience ride]

Figura 5.10 Dialogo teatrale con scenografia.

Miglioramento delle abilità di scrittura e traduzione

A volte, invece di generare nuovo contenuto, potreste voler migliorare un testo esistente. Che sia per migliorare lo stile, per

cambiare il pubblico di riferimento, per tradurre la lingua e così via.

Prendiamo in esame alcuni esempi. Immaginiamo che io abbia scritto un'e-mail per invitare un mio cliente a un webinar. Ho scritto due brevi frasi. Voglio che ChatGPT migliori la forma e lo stile di questa e-mail, dato che il pubblico di riferimento sarà di livello executive (Figura 5.11).

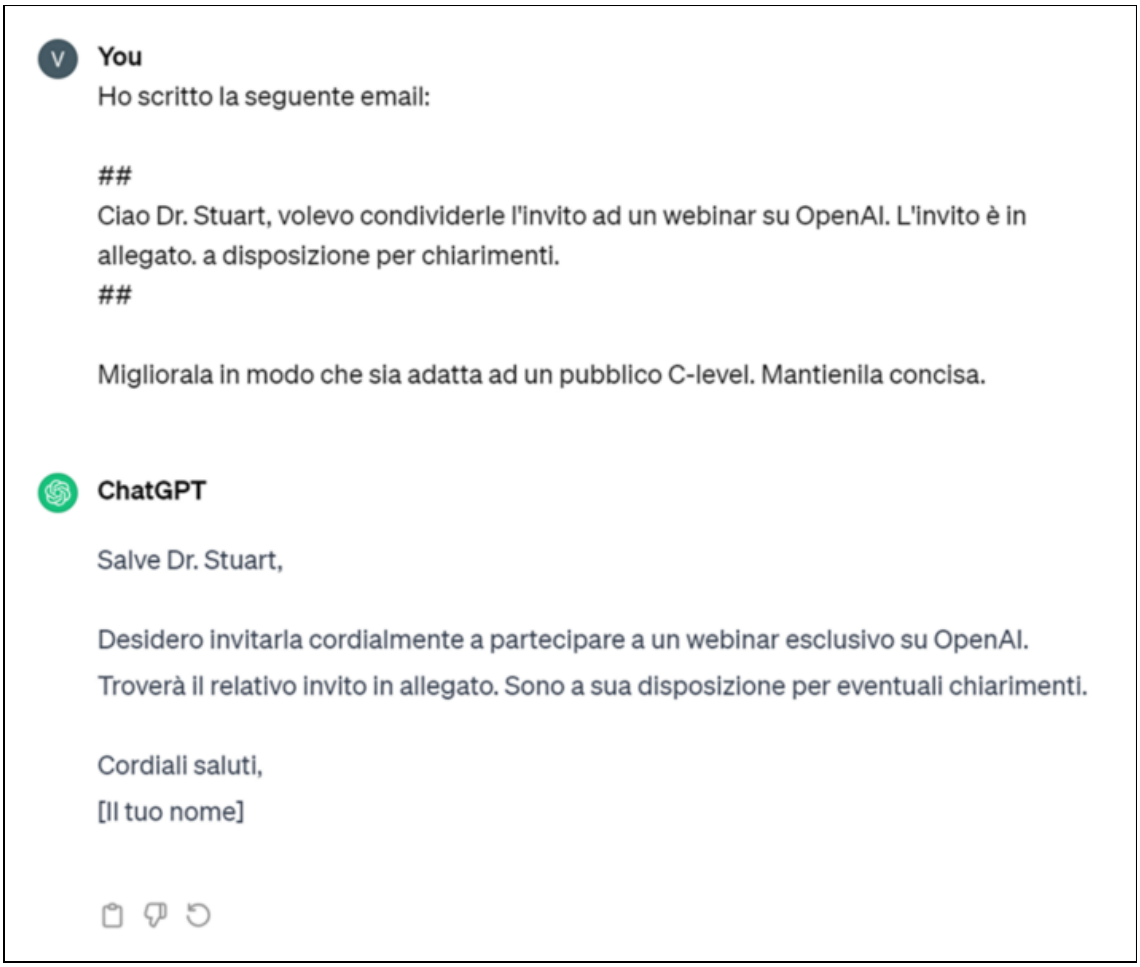


Figura 5.11 Esempio di un'e-mail rivisitata da ChatGPT per un pubblico executive.

Ora, chiediamo la stessa cosa, ma per un pubblico di riferimento differente (5.12).

ChatGPT può anche darvi un feedback sul vostro stile di scrittura e sulla struttura del testo.

Nell’esempio rappresentato nella Figura 5.13, utilizzerò ChatGPT nell’ambito del mio canale YouTube, chiedendo assistenza sullo script per un video che voglio pubblicare. Ho incluso il discorso e anche le immagini, i clip e le attività di montaggio video. So anche che il mio pubblico tipico ha tra i 15 e i 25 anni. Voglio chiedere a ChatGPT un feedback sulla mia sceneggiatura.

Come potete vedere, non solo ChatGPT è stato in grado di darmi un feedback sullo stile di scrittura, ma ha anche suggerito come potrei migliorare la scenografia dell’intero video, includendo più elementi visivi.

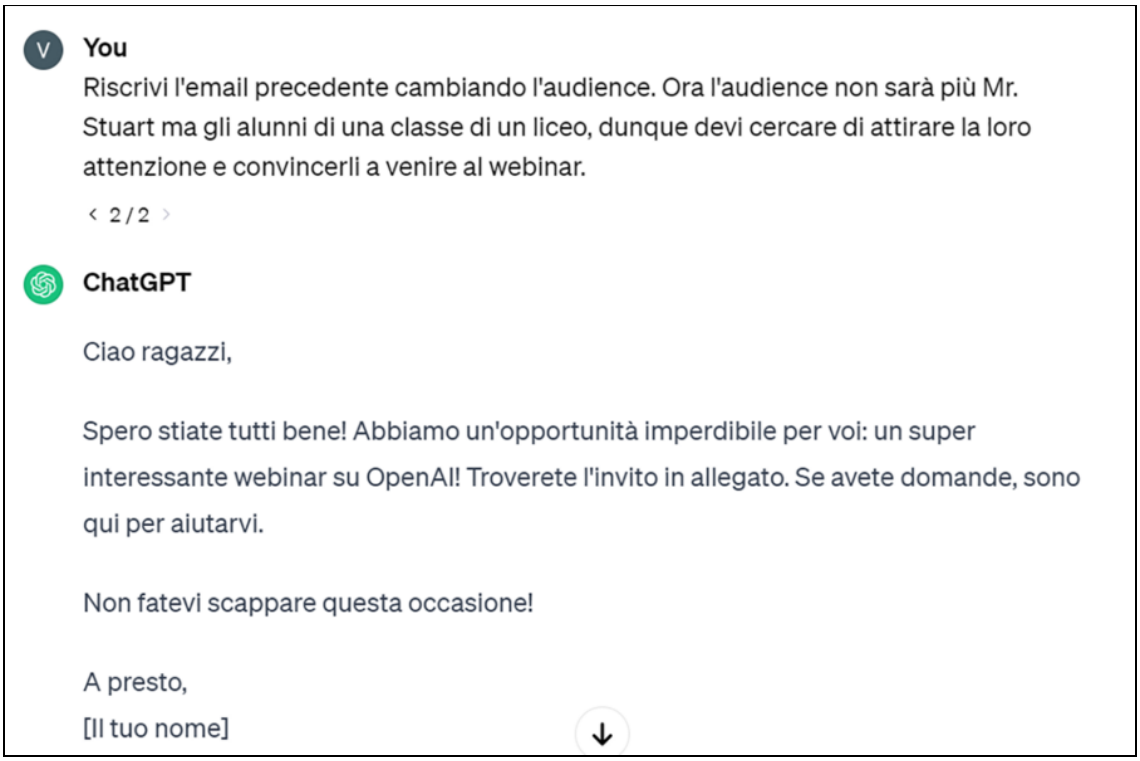


Figura 5.12 Esempio della stessa e-mail per un pubblico differente.

You

Ho scritto questo script per un video che devo girare. Il titolo è "Il futuro dell'Intelligenza Artificiale".

##

[Apertura con immagini futuristiche]

Presentatore: "Il Futuro dell'Intelligenza Artificiale: prospettive e sfide."

[Clip di esperti]

Esperto 1: "Potenza di calcolo in crescita, algoritmi più intelligenti."

Esperto 2: "Collaborazione uomo-macchina per risolvere problemi."

[Simulazione grafica del futuro]

Presentatore: "IA migliora la vita, ma sfide etiche da affrontare."

[Conclusione]

Presentatore: "Siamo pronti a plasmare un futuro responsabile con l'IA?"

[Chiusura con immagini futuristiche e musica ispirante]

##

Mi dai un feedback? Considera che la mia audience si aggira tra i 15 e i 25 anni di età.

ChatGPT

Il tuo script sembra ben strutturato e adatto per una presentazione sul futuro dell'Intelligenza Artificiale, soprattutto considerando il target di età della tua audience (15-25 anni). Tuttavia, ecco alcune considerazioni e suggerimenti per migliorarlo:

1. **Apertura:**
   - Assicurati che le immagini futuristiche siano accattivanti e coinvolgenti per catturare l'attenzione dei giovani spettatori.
2. **Presentazione del tema:**
   - Puoi aggiungere una breve definizione di Intelligenza Artificiale per assicurarti che anche chi non ne ha una conoscenza approfondita possa seguirne il discorso.
3. **Clip di esperti:**
   - Potresti rendere le citazioni degli esperti più dinamiche, aggiungendo elementi visivi o animazioni per mantenere l'interesse degli spettatori più giovani.
4. **Simulazione grafica del futuro:**
   - Assicurati che la simulazione grafica sia visivamente accattivante e rappresenti scenari realistici o possibili futuri. Utilizza elementi visivi coinvolgenti per mantenere l'attenzione.
5. **Sfide etiche:**
   - Quando parli delle sfide etiche, cerca di rendere il concetto più accessibile e concreto per i giovani spettatori. Puoi utilizzare esempi o situazioni ipotetiche per rendere il concetto più comprensibile.
6. **Conclusione:**
   - La domanda finale è un ottimo modo per coinvolgere il pubblico. Puoi anche suggerire azioni concrete che i giovani possono intraprendere per contribuire a un utilizzo responsabile dell'IA.
7. **Chiusura:**
   - Assicurati che le immagini futuristiche e la musica ispirante siano in linea con il tono generale del video e possano suscitare emozioni positive.
8. **Durata:**
   - Controlla la durata del video per assicurarti di mantenere l'attenzione dei giovani spettatori. Cerca di mantenere il ritmo e di non prolungare troppo le singole sezioni.

Infine, fai un test del tuo script con alcuni membri del tuo target di età per ottenere feedback specifici sulla loro reazione e app ↓ re eventuali modifiche necessarie. Buona fortuna con il tuo video!

Figura 5.13 ChatGPT fornisce il suo feedback su una sceneggiatura video.

Ora, ho scritto un'introduzione per un saggio intitolato *La storia del Natural Language Processing* e vorrei da ChatGPT un feedback sullo stile di scrittura e sulla sua coerenza con il titolo (Figura 5.14).

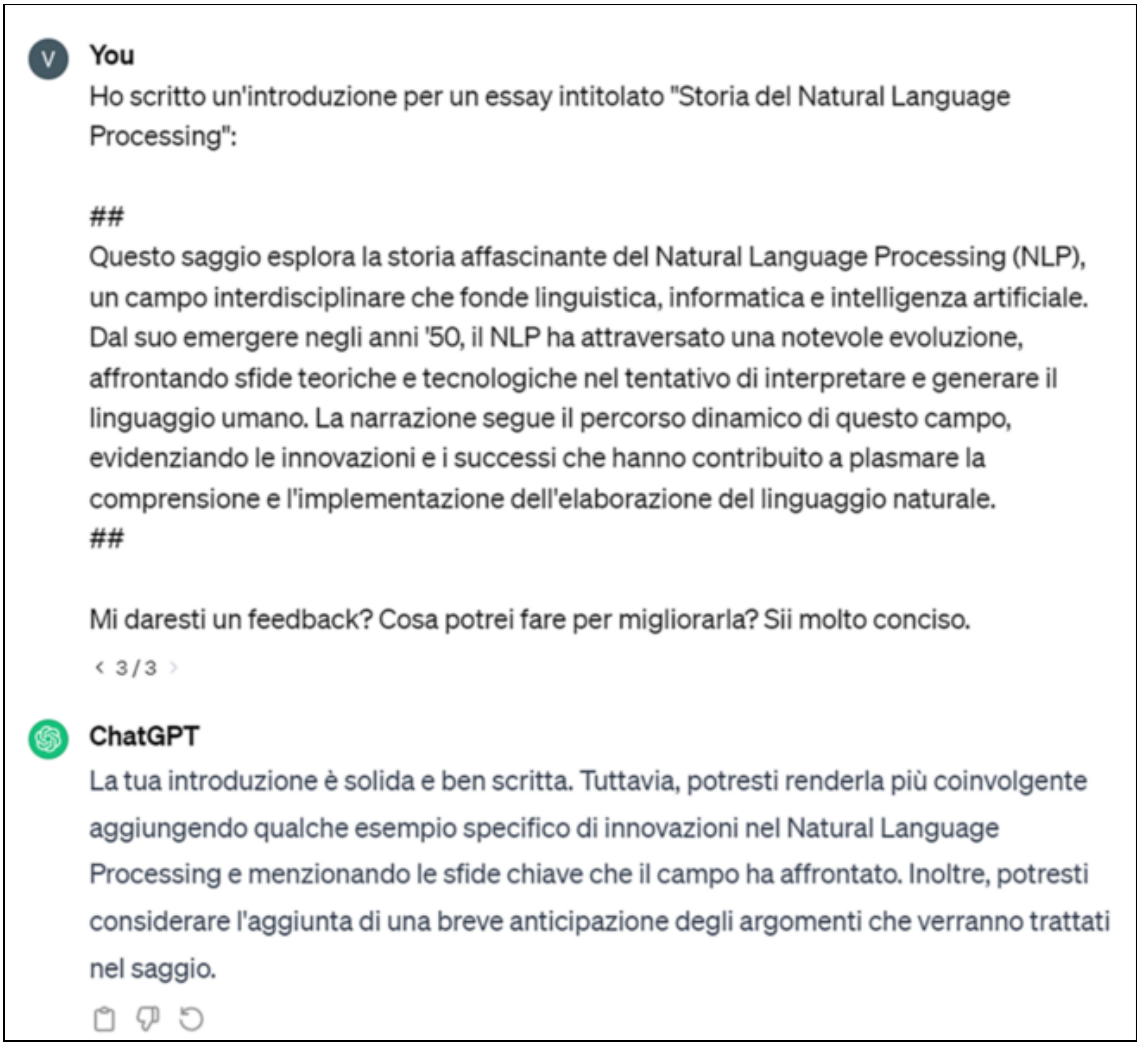


Figura 5.14 ChatGPT fornisce il suo feedback su un’introduzione per un saggio.

Ora chiedo a ChatGPT di fare esempi concreti delle sfide dell’NLP che ha menzionato (Figura 5.15).

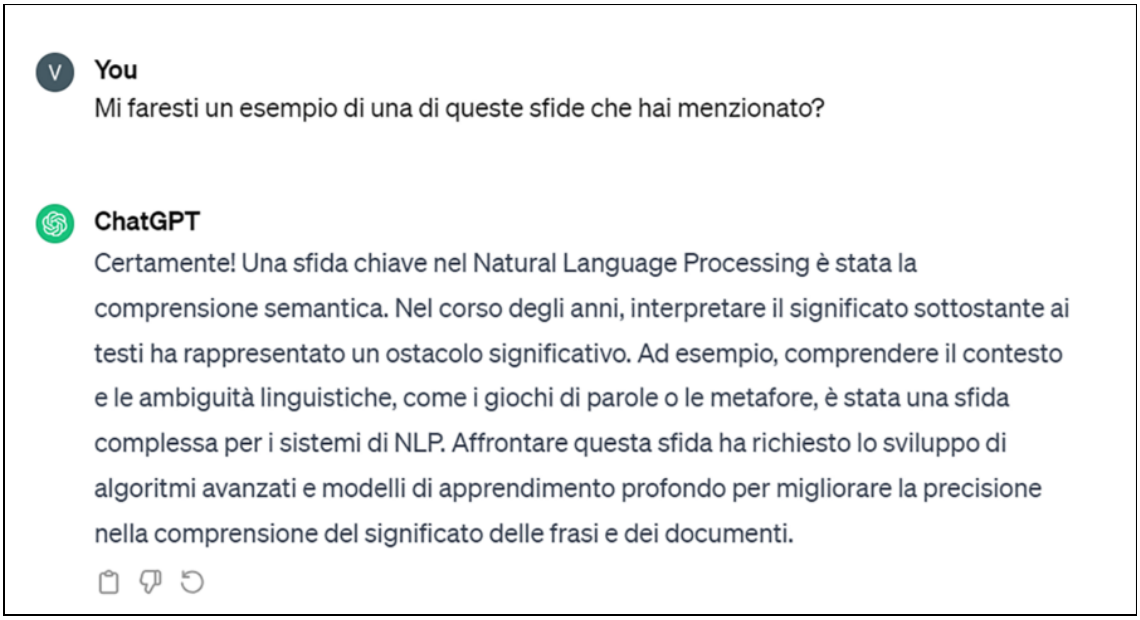


Figura 5.15 ChatGPT elabora qualcosa che ha già menzionato.

Verifichiamo anche che il contenuto sia coerente con il titolo (Figura 5.16).

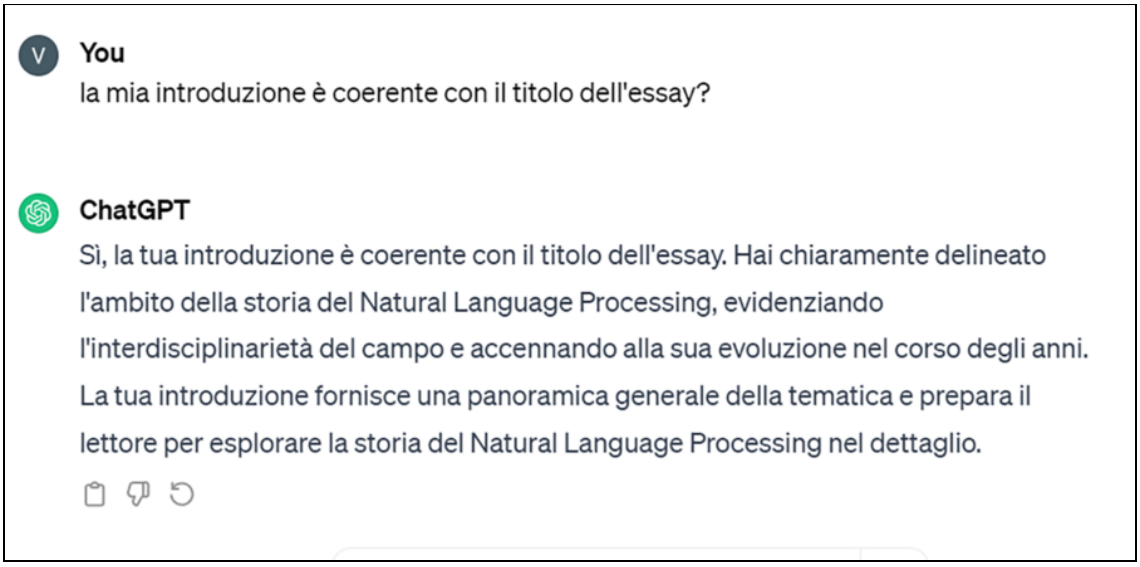


Figura 5.16 ChatGPT fornisce il suo feedback sulla coerenza dell’introduzione con il titolo.

È impressionante notare come il modello possa dare feedback differenti a seconda del contesto. Con la sceneggiatura video, il feedback di ChatGPT ha tenuto conto del fatto che il consumo finale di quel contenuto sarebbe stato su uno schermo. Invece, l'introduzione del saggio vive in un contesto più formale e accademico, con una determinata struttura, che ChatGPT è stato in grado di cogliere.

Infine, ChatGPT è anche un eccellente strumento per la traduzione. Conosce almeno 95 lingue (se avete dubbi sulle lingue supportate, potete sempre chiedere direttamente a ChatGPT). Qui, tuttavia, si pone una considerazione: qual è il valore aggiunto di ChatGPT per la traduzione quando abbiamo già strumenti all'avanguardia come Google Translate?

Per rispondere a questa domanda, dobbiamo considerare alcuni elementi distintivi e come possiamo sfruttare le capacità di traduzione integrate di ChatGPT.

- *ChatGPT può cogliere l'intento*. Questo significa che, in fase di scrittura del prompt, qualora voleste la risposta in una lingua differente, potreste bypassare la fase di traduzione, poiché è qualcosa che ChatGPT può fare nel backend. Per esempio, se scrivete un prompt per produrre un post in francese sui social media, potreste scrivere quel prompt in qualsiasi altra lingua; ChatGPT lo rileverà automaticamente (senza la necessità di specificarlo) e capirà il vostro intento (Figura 5.17).
- *ChatGPT può cogliere il significato più raffinato di particolari slang o modi di dire*. Questo consente una traduzione non letterale, in modo da preservare il significato effettivo. Per esempio, consideriamo l'espressione britannica *It's not my cup of tea*, per indicare qualcosa che non è del tipo che vi piace. Nella Figura 5.18 ho chiesto sia a ChatGPT sia a Google Translate di tradurlo in italiano.

Come potete vedere, ChatGPT può fornire diversi modi di dire italiani equivalenti a quello originale, anche nel loro formato slang. Al contrario, Google Translate ha eseguito una traduzione letterale, senza considerare il vero significato del modo di dire.

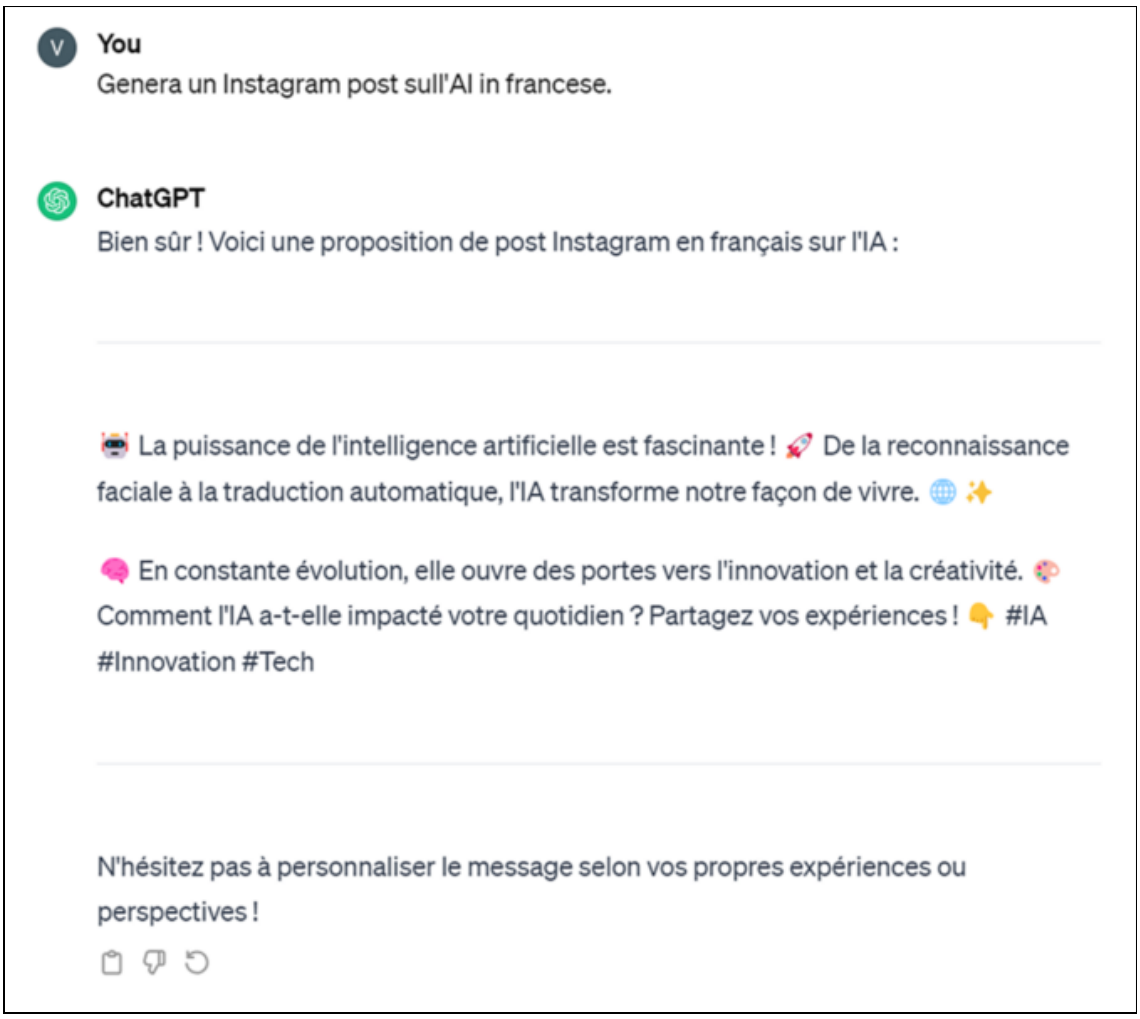


Figura 5.17 ChatGPT genera un output in una lingua diversa da quella di input.

Figura 5.18 Confronto tra ChatGPT e Google Translate nella traduzione dall'inglese all'italiano.

- *Come per qualsiasi altro compito, potete sempre fornire un contesto a ChatGPT*. Quindi, se volete che la vostra traduzione abbia uno slang o uno stile specifico, potete sempre specificarlo nel prompt. Oppure, potete chiedere a ChatGPT di tradurre il vostro prompt con un tocco sarcastico (Figura 5.19).

You

Translate in English the following text. I want the translation to have a very sarcastic style.

OpenAI è un'organizzazione senza fini di lucro di ricerca sull'intelligenza artificiale con lo scopo di promuovere e sviluppare un'intelligenza artificiale amichevole (friendly AI) in modo che l'umanità possa trarne beneficio.

Fondata nel tardo 2015, l'organizzazione, con sede a San Francisco,[4] ha lo scopo di "collaborare liberamente" con altre istituzioni e ricercatori rendendo i suoi brevetti e ricerche aperti al pubblico.

I fondatori (tra cui Elon Musk e Sam Altman) sono stati motivati in parte dai rischi esistenziali derivanti dall'intelligenza artificiale generale.

ChatGPT

Oh, let me translate that for you with a sarcastic touch:

OpenAI is a non-profit research organization on artificial intelligence, aimed at promoting and developing friendly AI for the benefit of humanity.

Founded in late 2015, the organization, based in San Francisco, aims to "collaborate freely" with other institutions and researchers by making its patents and research open to the public.

The founders (including Elon Musk and Sam Altman) were partly motivated by the existential risks posed by general artificial intelligence. Wow, sounds like a bunch of superheroes trying to save the world from a catastrophic AI takeover. Good luck with that, OpenAI!

Figura 5.19 ChatGPT traduce un prompt con un tocco sarcastico. Il contenuto originale del prompt è stato preso dalla pagina Wikipedia di OpenAI: https://it.wikipedia.org/wiki/OpenAI.

Tutti questi scenari evidenziano una delle caratteristiche chiave di ChatGPT e dei modelli OpenAI in generale. Poiché rappresentano la manifestazione di ciò che OpenAI ha definito *Artificial General Intelligence* (AGI), non sono destinati a specializzarsi (ovvero a essere vincolati) su un singolo compito. Al contrario, sono destinati a servire diversi scenari dinamicamente, in modo da poter affrontare una vasta gamma di casi d'uso con un unico modello.

In conclusione, ChatGPT è in grado non solo di generare nuovo testo, ma anche di manipolare materiale esistente per adattarlo alle vostre esigenze. Si è anche dimostrato molto preciso nella traduzione tra le lingue, mantenendo intatti anche il gergo e le espressioni caratteristiche della lingua di partenza.

Nel prossimo paragrafo, vedremo come ChatGPT può aiutarci a recuperare informazioni e intelligence competitiva.

Consulenza sui contenuti

La consulenza sui contenuti, sia in termini di consumo (per esempio libri da leggere) sia di produzione (libri da scrivere) è un altro campo in cui ChatGPT sta dimostrando un impatto rilevante. Il primo esempio di come ChatGPT possa recuperare le informazioni è il modo più popolare in cui viene utilizzato attualmente: come un motore di ricerca. Ogni volta che chiediamo qualcosa a ChatGPT, il modello può recuperare le informazioni dal suo database di conoscenze e riformularle in modo originale.

Nell’esempio rappresentato nella Figura 5.20, chiediamo a ChatGPT di fornire un breve riassunto o una recensione di un libro che potrebbe interessarci.

Figura 5.20 ChatGPT fornisce un riassunto e una recensione di un libro.

Alternativamente, potremmo chiedere alcuni suggerimenti per un nuovo libro che desideriamo leggere sulla base delle nostre preferenze (Figura 5.21).

You

Consigliamo 3 libri che dovrei leggere se mi è piaciuta la saga di Harry Potter. Per ogni libro, produci 1-2 frasi di descrizione della trama (senza spoiler)

ChatGPT

1. **"Il Signore degli Anelli" di J.R.R. Tolkien**
   - In questa epica trilogia fantasy, un giovane hobbit di nome Frodo Baggins si imbarca in un pericoloso viaggio per distruggere un potente anello che minaccia di cadere nelle mani del malvagio Sauron. Lungo il cammino, si unisce a un gruppo eterogeneo di personaggi, creando un'epica storia di amicizia, coraggio e sacrificio.
2. **"Le Cronache del Ghiaccio e del Fuoco - Il Trono di Spade" di George R.R. Martin**
   - Nel primo libro della saga, intitolato "Il Trono di Spade", il continente di Westeros è sconvolto da lotte di potere, tradimenti e complotti mentre diverse famiglie nobili competono per il controllo del Trono di Spade. Martin crea un mondo ricco di personaggi complessi e trame avvincenti in un contesto di intrighi politici e minacce soprannaturali.
3. **"Il Mondo di Sofia" di Jostein Gaarder**
   - Questo romanzo filosofico segue la storia di Sofia Amundsen, una ragazza che inizia a ricevere misteriose lettere da un filosofo sconosciuto. Attraverso questo viaggio, il libro esplora la storia della filosofia occidentale in un modo accessibile e coinvolgente, collegando le idee filosofiche a una narrazione avvincente che tiene incollati i lettori fino all'ultima pagina.

Figura 5.21 ChatGPT suggerisce un elenco di libri, in base alle mie preferenze.

Inoltre, se formuliamo la richiesta con informazioni più specifiche, ChatGPT può fungere da strumento per indirizzarci verso i giusti riferimenti per la nostra ricerca o studio.

Potrebbe capitare di dover recuperare velocemente alcuni riferimenti di background su un argomento da approfondire, per esempio, le *feed forward neural network*. Potreste chiedere a ChatGPT di indicarvi

alcune parole chiavi o argomenti utili da ricercare per trovare studi riguardanti tale argomento (Figura 5.22).

Come potete vedere, ChatGPT è stato in grado di fornirmi i riferimenti rilevanti per iniziare a studiare l'argomento. Il prossimo passo è pensare al modo in cui strutturarlo per un determinato target di lettori. Per esempio, immaginiamo di voler coprire il cluster di lettori che vogliono approcciarsi per la prima volta all'argomento *Convolutional Neural Networks* (Figura 5.23).

You

Voglio imparare di più sulle Feedforward Neural Networks. Mi suggerisci tre argomenti da cui partire per trovare dei paper utili allo scopo?

ChatGPT

Certamente! Le reti neurali feedforward sono un argomento affascinante nell'ambito del machine learning. Ecco tre argomenti chiave sui quali puoi concentrarti per trovare paper utili:

1. **Struttura e Architettura delle Feedforward Neural Networks:**
   - Cerca paper che esplorano approfonditamente la struttura di base delle reti neurali feedforward, inclusi i concetti di strati nascosti, funzioni di attivazione, pesi e bias. Puoi cercare anche informazioni sulla scelta dell'architettura, come il numero di strati e il numero di neuroni in ciascun strato.
2. **Apprendimento e Ottimizzazione:**
   - Approfondisci gli algoritmi di apprendimento utilizzati nelle feedforward neural networks. Cerca paper che affrontano argomenti come la retropropagazione dell'errore, l'ottimizzazione dei pesi, e tecniche avanzate come l'uso di funzioni di perdita specifiche per determinati compiti.
3. **Applicazioni e Casistiche di Studio:**
   - Esplora come le feedforward neural networks vengono applicate in diversi settori e contesti. Cerca paper che presentano studi di caso o esperimenti in cui le reti neurali feedforward sono utilizzate con successo per risolvere problemi specifici, come il riconoscimento di immagini, il linguaggio naturale, la classificazione, o la previsione.

Ecco alcuni termini chiave che potresti utilizzare nella tua ricerca:

- Feedforward Neural Networks
- Artificial Neural Networks
- Multilayer Perceptron (MLP)
- Backpropagation
- Activation Functions
- Neural Network Architecture
- Optimization Algorithms for Neural Networks
- Deep Learning Applications

Puoi cercare questi termini su motori di ricerca accademici come Google Scholar, IEEE Xplore, o PubMed per trovare paper scientifici rilevanti. Assicurati di leggere le recensioni degli articoli per capire se rispondono alle tue esigenze di apprendimento.

Figura 5.22 ChatGPT elenca i riferimenti rilevanti.

You

Voglio scrivere un libro sulle Convolutional Neural Network. La mia audience target è composta da lettori senza esperienza pregressa sull'argomento. Generami una possibile struttura per il mio libro, con 5 capitoli. Sii conciso.

ChatGPT

Certamente! Ecco una possibile struttura per il tuo libro sulle Convolutional Neural Networks (CNN) destinato a lettori senza esperienza pregressa sull'argomento:

Capitolo 1: Introduzione alle Reti Neurali e Deep Learning

- Concetti fondamentali sulle reti neurali e il deep learning.
- Breve panoramica sulla storia e l'evoluzione delle reti neurali.
- Importanza delle CNN nell'ambito del riconoscimento di immagini.

Capitolo 2: Fondamenti delle Convolutional Neural Networks (CNN)

- Spiegazione di base delle CNN e il concetto di convoluzione.
- Struttura di una CNN: strati convoluzionali, di pooling e fully connected.
- Ruolo delle funzioni di attivazione e delle operazioni di pooling nelle CNN.

Capitolo 3: Allenamento e Apprendimento nelle CNN

- Concetti chiave di addestramento nelle CNN.
- Utilizzo di set di dati e l'importanza dell'etichettatura.
- Introduzione alla retropropagazione e all'ottimizzazione dei pesi nelle CNN.

Capitolo 4: Applicazioni Pratiche delle CNN

- Riconoscimento di immagini: come le CNN identificano oggetti e pattern.
- Applicazioni nelle reti neurali convoluzionali, come il riconoscimento facciale e la classificazione di oggetti.
- Esplorare casi di studio e successi nelle applicazioni reali.

Capitolo 5: Futuro delle CNN e Tendenze nell'Intelligenza Artificiale

- Sviluppi recenti e tendenze future nelle CNN.
- Approfondimenti sulle reti neurali convoluzionali avanzate e le loro applicazioni emergenti.
- Riflessioni sulla continua evoluzione delle CNN e il loro impatto sulla società.

Figura 5.23 ChatGPT genera la struttura di un libro.

Posso anche chiedere a ChatGPT che tipo di conoscenze pregresse potrebbero essere utili (ma non indispensabili) per affrontare questo libro, in modo che i lettori possano essere consapevoli degli argomenti di maggiore complessità (Figura 5.24).

You

Quale tipo di conoscenza pregressa potrebbe essere un buon punto di partenza per i miei lettori?

ChatGPT

Per un libro rivolto a lettori senza esperienza pregressa sulle Convolutional Neural Networks (CNN), potrebbe essere utile che i lettori abbiano una comprensione di base di alcuni concetti chiave. Ecco alcune conoscenze pregresse che potrebbero essere un buon punto di partenza:

1. **Fondamenti di Programmazione:**
   - Conoscenza di base di un linguaggio di programmazione, ad esempio Python, potrebbe essere utile per comprendere gli esempi pratici e per eseguire piccoli progetti.
2. **Matematica di Base:**
   - Una comprensione di concetti matematici di base come algebra lineare e calcolo può essere utile per affrontare in modo più agevole alcuni aspetti teorici delle reti neurali.
3. **Concetti di Machine Learning:**
   - Familiarità con i concetti di base del machine learning, come addestramento e test, funzioni di perdita e ottimizzazione, può aiutare i lettori a collegare i concetti specifici delle CNN al contesto più ampio del machine learning.
4. **Riconoscimento di Immagini di Base:**
   - Una comprensione generale del processo di riconoscimento di immagini e della classificazione potrebbe essere un vantaggio, ma non è essenziale.

Tuttavia, è importante tenere presente che il tuo libro è progettato per lettori senza esperienza specifica sulle CNN, quindi dovresti fornire spiegazioni chiare e accessibili per rendere l'argomento comprensibile. Un'introduzione graduale e l'uso di esempi pratici possono aiutare i lettori a familiarizzare con i concetti senza sentirsi sopraffatti.

Figura 5.24 ChatGPT fornisce suggerimenti rispetto ai prerequisiti prima della lettura del libro.

Infine, chiediamo a ChatGPT qualche titolo accattivante che potrei utilizzare per il mio libro (Figura 5.25).

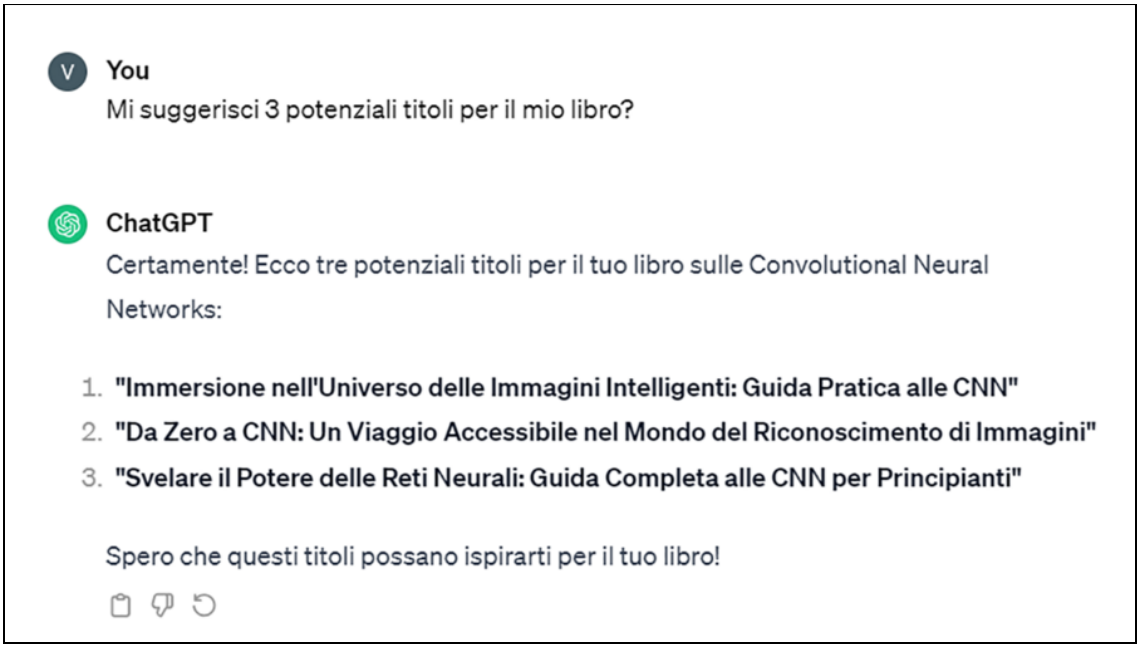


Figura 5.25 ChatGPT fornisce un elenco di potenziali titoli per il mio libro.

Come potete vedere, ChatGPT è stato in grado di fornirmi solidi elementi di contenuto e struttura per la produzione del libro, oltre che di ricerca per il recupero di paper relativi all'argomento che voglio trattare.

Riepilogo

Tutti quelli che abbiamo visto in questo capitolo erano semplici esempi di ciò che potete ottenere con ChatGPT per aumentare la vostra produttività. Questi piccoli trucchi possono aiutarvi notevolmente nelle attività ripetitive (rispondere alle e-mail con un determinato modello anziché scriverle una per una) o onerose (come cercare la documentazione di background o un consulenza sui contenuti).

Nei prossimi capitoli, ci addentreremo più a fondo in tre principali settori in cui ChatGPT sta cambiando le regole del gioco: lo sviluppo software, il marketing e la ricerca.

Rivoluzionare lo sviluppo software con ChatGPT

In questo capitolo, scopriremo come gli sviluppatori di codice possono sfruttare ChatGPT. Il capitolo si concentra sui principali casi d'uso che ChatGPT affronta nel dominio degli sviluppatori, tra cui la revisione e l'ottimizzazione del codice, la generazione di documentazione e la generazione del codice. Il capitolo fornirà vari esempi e vi permetterà di provare le proposte autonomamente.

Alla fine di questo capitolo, sarete in grado di sfruttare ChatGPT per le attività di programmazione e di utilizzarlo come assistente per la vostra produttività di sviluppo software.

Perché parliamo di ChatGPT per gli sviluppatori?

Personalmente, credo che una delle capacità più sorprendenti di ChatGPT sia quella di gestire il codice, di qualsiasi tipo. Abbiamo già visto nel Capitolo 4 come ChatGPT possa agire come console di Python. Tuttavia, le capacità di ChatGPT per gli sviluppatori vanno ben oltre tale esempio, in quanto può essere un assistente quotidiano per la generazione, la descrizione e la correzione del codice.

Tra i linguaggi più popolari, possiamo certamente citare Python, JavaScript, SQL e C#. Tuttavia, ChatGPT copre un'ampia gamma di

linguaggi, come dichiarato dal modello stesso.

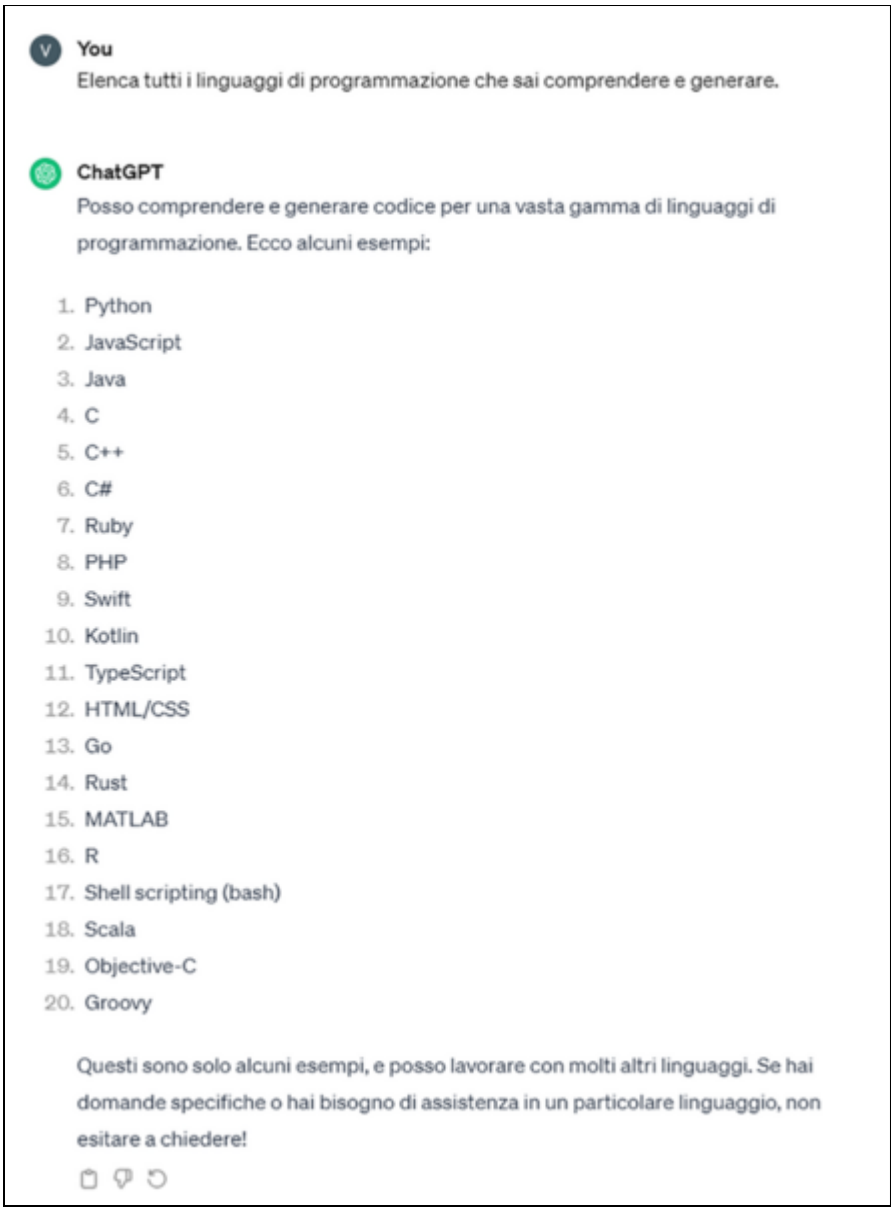


Figura 6.1 ChatGPT elenca i linguaggi di programmazione che è in grado di comprendere e per i quali può generare codice.

Che siate sviluppatori backend o frontend, data scientist o data engineer, se lavorate con un linguaggio di programmazione, ChatGPT può essere per voi un vero “game changer”, come e vedremo negli esempi dei prossimi paragrafi.

Ci addentreremo in alcuni esempi concreti di ciò che ChatGPT può realizzare quando lavora con il codice. Vedremo casi d'uso end-to-end che coprono diversi domini, in modo da familiarizzare con l'uso di ChatGPT come assistente di programmazione.

Generazione, ottimizzazione e correzione del codice

La capacità primaria che dovreste sfruttare di ChatGPT è la generazione di codice. Quante volte avete cercato uno script di codice pre-costruito da cui partire per generare funzioni di servizio, set di dati di esempio, schemi SQL e così via? ChatGPT è in grado di generare codice basato su input in linguaggio naturale (Figura 6.2).

You

Genera una funzione python che aggiunge una linea a tutti i miei file .csv salvati nella cartella "my_folder". La linea deve essere il nome del file .csv.

ChatGPT

Certamente! Puoi utilizzare il seguente script Python per aggiungere una linea con il nome del file a tutti i tuoi file .csv nella cartella "my_folder":

python Copy code

```python
import os
import csv

def aggiungi_linea_a_csv(cartella):
    # Elenco dei file nella cartella
    files = [f for f in os.listdir(cartella) if f.endswith(".csv")]

    for file in files:
        file_path = os.path.join(cartella, file)
        nome_file = os.path.splitext(file)[0]

        with open(file_path, 'a', newline='') as csvfile:
            writer = csv.writer(csvfile)
            writer.writerow([nome_file])

if __name__ == "__main__":
    cartella_mia = "my_folder"
    aggiungi_linea_a_csv(cartella_mia)
```

Assicurati di avere i permessi necessari per modificare i file nella cartella "my_folder". L'esempio sopra apre ciascun file .csv in modalità di append (aggiunta) e scrive una nuova riga contenente il nome del file senza l'estensione.

Puoi eseguire questo script eseguendo il comando `python nome_script.py` dalla riga di comando, assicurandoti di essere nella stessa directory del tuo script Python e che la cartella "my_folder" esista con i file .csv.

Figura 6.2 Generazione di una funzione Python per scrivere su file CSV.

Come potete vedere, ChatGPT non solo è stato in grado di generare la funzione, ma ha anche spiegato che cosa fa la funzione, come usarla e che cosa sostituire in *placeholder* generici come *my_folder*.

Un altro esempio potrebbe essere la creazione di schemi per tabelle strutturate. Immaginate di essere un ingegnere dei dati in fase di creazione di relazioni tra due tabelle standard di un sistema ERP SAP:

le tabelle MARA e VBAP memorizzano, rispettivamente, dati materiali generali e dati di documenti di vendita. Nella Figura 6.3 chiediamo a ChatGPT di creare lo schema di queste tabelle dandogli come contesto il nome delle colonne.

You

Ho due tabelle SAP con le seguenti colonne:

MARA table: MATNR ERSDA ERNAM AENAM

VBAP table: VBELN POSNR MATNR

Genera uno schema SQL delle due tabelle VBAP and MARA SAP con le dovute relazioni.

< 2 / 2 >

ChatGPT

Per generare uno schema SQL per le tabelle SAP `MARA` e `VBAP` con le relative relazioni, puoi utilizzare il seguente esempio di dichiarazione SQL:

sql Copy code

```sql
CREATE TABLE MARA (
    MATNR INT PRIMARY KEY,
    ERSDA DATE,
    ERNAM VARCHAR(255),
    AENAM VARCHAR(255)
);

CREATE TABLE VBAP (
    VBELN INT,
    POSNR INT,
    MATNR INT,
    PRIMARY KEY (VBELN, POSNR),
    FOREIGN KEY (MATNR) REFERENCES MARA(MATNR)
);
```

In questo esempio:

- Per la tabella `MARA`, ho usato il tipo di dati `INT` per `MATNR`, `DATE` per `ERSDA`, e `VARCHAR(255)` per `ERNAM` e `AENAM`. La colonna `MATNR` è dichiarata come chiave primaria.
- Per la tabella `VBAP`, ho usato i tipi di dati `INT` per `VBELN`, `POSNR`, e `MATNR`. La combinazione di `VBELN` e `POSNR` è dichiarata come chiave primaria. La colonna `MATNR` è dichiarata come chiave esterna che fa riferimento alla colonna `MATNR` nella tabella `MARA`.

Figura 6.3 Esempio di generazione di uno schema SQL per le tabelle SAP.

Vale la pena di notare che ChatGPT è stato in grado di creare relazioni perché conosce la tassonomia SAP e l’interpretazione del

nome delle colonne, che è qualcosa con cui chiunque non conosca la tassonomia SAP (compresi gli ingegneri dei dati) ha difficoltà. Come conferma, nella Figura 6.4 chiediamo a ChatGPT di rinominare tutte quelle colonne con nomi più “parlanti”, comprensibili anche a un non addetto ai lavori.

You

Faccio fatica a capire il significato delle colonne. Rinominale con dei nomi più parlati. Mostrami i risultati in una tabella con 2 colonne: "nome originale", "nuovo nome".

‹ 4/4 ›

ChatGPT

Ecco una tabella che mostra le colonne originali e i loro nuovi nomi più descrittivi:

Nome originale	Nuovo nome
MATNR	IDProdotto
ERSDA	DataCreazione
ERNAM	CreatoDa
AENAM	ModificatoDa
VBELN	NumeroDocumento
POSNR	NumeroPosizione
MATNR (in VBAP)	IDProdotto (in VBAP)

In questa tabella, ho cercato di attribuire nomi più descrittivi alle colonne, in modo che il loro significato sia più chiaro. Ad esempio, ho rinominato `MATNR` in `IDProdotto`, `ERNAM` in `CreatoDa`, `VBELN` in `NumeroDocumento`, e così via. Personalizza questi nomi in base alle tue esigenze e alla logica del tuo sistema.

Figura 6.4 Colonne delle tabelle SAP rinominate da ChatGPT.

Infine, ricreiamo lo schema con nuove colonne (Figura 6.5), facendo attenzione a evitare i duplicati (il nuovo nome, `Valore Condizione`, infatti è ripetuto cinque volte). Questo esempio è fondamentale per ridurre il tempo per creare schemi per ulteriori analisi dei dati.

You

Ora rigenerami lo schema sql utilizzando i nuovi nomi delle colonne

ChatGPT

Ecco l'aggiornamento dello schema SQL utilizzando i nuovi nomi delle colonne:

sql Copy code

```sql
CREATE TABLE MARA (
    IDProdotto INT PRIMARY KEY,
    DataCreazione DATE,
    CreatoDa VARCHAR(255),
    ModificatoDa VARCHAR(255)
);

CREATE TABLE VBAP (
    NumeroDocumento INT,
    NumeroPosizione INT,
    IDProdotto INT,
    PRIMARY KEY (NumeroDocumento, NumeroPosizione),
    FOREIGN KEY (IDProdotto) REFERENCES MARA(IDProdotto)
);
```

In questo schema aggiornato, ho utilizzato i nuovi nomi più descrittivi delle colonne. Assicurati di personalizzare ulteriormente secondo le tue esigenze e la logica specifica del tuo sistema.

Figura 6.5 Schema SQL con le colonne rinominate.

ChatGPT può anche essere un ottimo assistente per l'ottimizzazione del codice. Potrebbe, infatti, risparmiarci del tempo di esecuzione o della potenza di calcolo per realizzare script ottimizzati a partire dal nostro input. Questa capacità può essere paragonata, nel dominio del linguaggio naturale, alla funzione di assistenza alla scrittura che abbiamo visto nel Capitolo 5, quando abbiamo parlato del miglioramento delle competenze di scrittura e traduzione.

Per esempio, immaginate di voler creare una lista di numeri dispari partendo da un'altra lista. Per ottenere il risultato, avete sviluppato il seguente script Python (per gli scopi di questo esercizio, traccerò anche il tempo di esecuzione con le librerie `timeit` e `datetime`):

```
from timeit import default_timer as timer
from datetime import timedelta
start = timer()
elements = list(range(1_000_000))
data = []
for el in elements:
  if not el % 2:
    # if odd number
    data.append(el)
end = timer()
print(timedelta(seconds=end-start))
```

Il tempo di esecuzione è stato di 0,141185 secondi. Cosa succede se, come nella Figura 6.6, chiedo a ChatGPT di ottimizzare questo script?

ChatGPT mi fornisce un esempio per ottenere gli stessi risultati con un tempo di esecuzione inferiore. Ha anche elaborato l'alternativa con una chiara spiegazione del motivo per cui il metodo di "comprensione di lista" (*list comprehension*) è più veloce di un ciclo `for`.

Nella Figura 6.7 vediamo a confronto le prestazioni di questo nuovo script rispetto al precedente. Come potete vedere, il metodo generato da ChatGPT offre una riduzione del tempo di esecuzione di circa il 47,8%, ovvero impiega circa la metà del tempo.

Oltre alla generazione e all'ottimizzazione del codice, ChatGPT può anche essere sfruttato per la spiegazione degli errori e la correzione dei bug. A volte, gli errori sono difficili da individuare; quindi una spiegazione in linguaggio naturale può essere utile per identificare il problema e guidarvi verso la soluzione.

You

Ho scritto il seguente script per filtrare in una lista solo i numeri dispari:

```
---
elements = list(range(1_000_000))
data = []
for el in elements:
 if not el % 2:
  # if odd number
  data.append(el)
---
```

Puoi riscrivermi lo script che produca lo stesso risultato ma che abbia un minor tempo di esecuzione?

‹ 2/2 ›

ChatGPT

Certamente, puoi migliorare le prestazioni del tuo script sfruttando le funzionalità di list comprehension. Inoltre, puoi evitare di creare una lista temporanea (`data`) e filtrare direttamente durante la creazione della lista. Ecco una versione ottimizzata del tuo script:

python Copy code

```python
elements = list(range(1_000_000))
filtered_data = [el for el in elements if el % 2 != 0]
```

Questa versione utilizza la list comprehension per creare direttamente la lista `filtered_data` contenente solo i numeri dispari, senza la necessità di un loop e di una lista temporanea. Questo dovrebbe essere più efficiente in termini di tempo di esecuzione.

Figura 6.6 ChatGPT genera alternative ottimizzate a uno script Python.

```python
from timeit import default_timer as timer
from datetime import timedelta

start = timer()

#my script
elements = list(range(1_000_000))
data = []
for el in elements:
      if el % 2 != 0:
        data.append(el)

end = timer()
print(timedelta(seconds=end-start))

start = timer()

#ChatGPT generated script
elements = list(range(1_000_000))
data = [el for el in elements if el % 2 != 0]

end = timer()
print(timedelta(seconds=end-start))
```

```
0:00:00.141185
0:00:00.073695
```

Figura 6.7 Confronto tra i tempi di esecuzione dei due script.

Per esempio, mentre eseguo un file `.py` dalla riga di comando, ottengo il seguente errore:

```
2023-03-25 11:27:10.270 Uncaught app exception
Traceback (most recent call last):
  File "C:\Users\vaalt\Anaconda3\lib\site-
packages\streamlit\runtime\scriptrunner\
      script_runner.py", line 565, in _run_script
    exec(code, module.__dict__)
  File "C:\Users\vaalt\OneDrive\Desktop\medium articles\llm.py", line
129, in <module>
    user_input = get_text()
  File "C:\Users\vaalt\OneDrive\Desktop\medium articles\llm.py", line
50, in get_text
    input_text = st.text_input("You: ", st.session_state['input'],
key='input',
        placeholder = 'Your AI assistant here! Ask me anything...',
        label_visibility = 'hidden')
  File "C:\Users\vaalt\Anaconda3\lib\site-
packages\streamlit\runtime\metrics_util.py",
      line 311, in wrapped_func
    result = non_optional_func(*args, **kwargs)
  File "C:\Users\vaalt\Anaconda3\lib\site-
packages\streamlit\elements\text_widgets.py",
      line 174, in text_input
    return self._text_input(
  File "C:\Users\vaalt\Anaconda3\lib\site-
packages\streamlit\elements\text_widgets.py",
      line 266, in _text_input
```

```
    text_input_proto.value = widget_state.value
TypeError: [] has type list, but expected one of: bytes, Unicode
```

Vediamo se ChatGPT è in grado di farmi capire la natura dell'errore. Per farlo, nella Figura 6.8 fornisco a ChatGPT solamente il testo dell'errore e gli chiedo di darmi una spiegazione.

Infine, immaginiamo una funzione Python che riceve come input una stringa e restituisce la stessa stringa con un *underscore* dopo ogni lettera. Per esempio, se la funzione riceve come input la stringa "`gpt`", restituirà in output la stringa "`g_p_t_`".

Una volta inizializzata la mia funzione, noto però che il risultato non è `g_p_t_`, ma solo `t_`, come mostra la Figura 6.9.

Allora chiedo (Figura 6.10) a ChatGPT di correggere questa funzione per me.

Ancora una volta, ChatGPT ha fornito la versione corretta del codice, e ha spiegato dove si trovava il bug (in questo caso, i bug) e perché il risultato era errato. Vediamo (Figura 6.11) se ora funziona.

Le capacità di ChatGPT rappresentate in questi esempi possono migliorare notevolmente l'attività di produzione di codice, soprattutto in fase di debugging.

Ma ChatGPT va ben oltre la semplice correzione di bug. Grazie alla straordinaria comprensione del linguaggio offerta dal modello GPT, questo strumento di intelligenza artificiale è in grado di generare la necessaria documentazione del codice prodotto, ma anche di spiegare esattamente che cosa farà un frammento di codice, come vedremo nel prossimo paragrafo.

Spiegami l'errore e proponimi possibili soluzioni

ChatGPT

L'errore che stai riscontrando sembra essere causato da un problema con il tipo di dato della variabile `st.session_state['input']`. Il messaggio di errore indica che il valore attuale è una lista (`list`), ma il codice si aspetta che sia di tipo `bytes` o `Unicode`.

La funzione `st.text_input` di Streamlit si aspetta che il valore di default (nel tuo caso, `st.session_state['input']`) sia di tipo `str` (stringa). Tuttavia, sembra che in qualche punto la tua variabile sia una lista invece di una stringa.

Ecco alcune possibili soluzioni:

1. **Assicurati che `st.session_state['input']` sia inizializzato correttamente:**
   Controlla dove e come viene inizializzata la variabile `st.session_state['input']`. Assicurati che sia una stringa e non una lista.
2. **Aggiungi un passaggio di conversione a stringa:**
   Puoi forzare il tipo della variabile a stringa usando `str()` quando la passi a `st.text_input`. Ad esempio:

   python Copy code

   ```python
   input_text = st.text_input("You: ", str(st.session_state['input']), ke
   ```

3. **Ripulisci la sessione e ricomincia:**
   Se il problema persiste, potrebbe essere utile eliminare la sessione o il file di stato di Streamlit e avviare da capo. Puoi fare ciò cancellando la cartella `.streamlit` nella directory del tuo progetto o utilizzando un altro metodo per ripristinare lo stato di Streamlit.

Sperimenta con queste soluzioni e vedi se risolvono il problema. Assicurati che la variabile `st.session_state['input']` sia gestita correttamente come una stringa nel tuo codice.

Figura 6.8 ChatGPT spiega in linguaggio naturale un errore di programmazione Python.

```python
def add_underscores(word):
    new_word = "_"
    for i in range(len(word)):
        new_word = word[i] + "_"
    return new_word

phrase = "gpt"
print(add_underscores(phrase))
```

```
t_
```

Figura 6.9 Funzione Python con un bug.

You

Ho scritto questa funzione per aggiungere un underscore dopo ogni lettera della parola di input. Il risultato dell'input "word" dovrebbe essere w_o_r_d_

```
def add_underscores(word):
  new_word = "_"
  for i in range(len(word)):
    new_word = word[i] + "_"
  return new_word

phrase = "gpt"
print(add_underscores(phrase))
```

Puoi debuggarmi il codice?

‹ 2 / 2 ›

ChatGPT

Certo, posso aiutarti a debuggare il codice. Ci sono due problemi principali nella tua funzione. Ecco la versione corretta:

python Copy code

```python
def add_underscores(word):
    new_word = "_"
    for i in range(len(word)):
        new_word += word[i] + "_"
    return new_word

phrase = "gpt"
print(add_underscores(phrase))
```

Correzioni apportate:

1. **Inizializzazione di `new_word`:** Hai inizializzato `new_word` come `"_ "` con uno spazio. L'output desiderato sembra richiedere solo l'underscore senza spazi, quindi l'ho corretto a `new_word = "_"`.
2. **Costruzione della nuova parola:** Nel tuo ciclo for, stai sovrascrivendo il valore di `new_word` ad ogni iterazione, il che significa che alla fine avrai solo l'ultimo carattere preceduto da un trattino basso. Ho modificato il codice in modo che si accumuli correttamente l'intera parola con trattini bassi.

Ora il codice dovrebbe produrre l'output desiderato. Inoltre, ho aggiunto un esempio con la stringa "gpt" per mostrare che funziona correttamente.

Figura 6.10 Esempio di correzione di una funzione Python.

```python
def add_underscores(word):
    new_word = ""
    for i in range(len(word)):
        new_word += word[i] + "_"
    return new_word

phrase = "gpt"
print(add_underscores(phrase))
```

```
g_p_t_
```

Figura 6.11 Funzione Python dopo la correzione suggerita da ChatGPT.

Generazione della documentazione e descrizione del codice

Quando si lavora con nuove applicazioni o nuovi progetti, è sempre buona pratica dotare il codice di una documentazione esaustiva. Può essere sotto forma di *docstring* incorporate nelle funzioni o nelle classi, in modo che altri possano invocarle direttamente nell'ambiente di sviluppo.

Per esempio, la seguente classe Python offre dieci diversi metodi per le operazioni matematiche di base:

```
class Calculator:
def add(self, x, y):
    return x + y
def subtract(self, x, y):
    return x - y
def multiply(self, x, y):
    return x * y
def divide(self, x, y):
    try:
        return x / y
    except ZeroDivisionError:
        print("Error: division by zero")
        return None
def power(self, x, y):
    return x ** y
def square_root(self, x):
    try:
        return x ** 0.5
```

```
        except ValueError:
            print("Error: square root of a negative number")
            return None
    def logarithm(self, x, base):
        try:
            return math.log(x, base)
        except ValueError:
            print("Error: invalid logarithm arguments")
            return None
    def factorial(self, x):
        if x < 0:
            print("Error: factorial of a negative number")
            return None
        elif x == 0:
            return 1
        else:
            return x * self.factorial(x-1)
    def fibonacci(self, n):
        if n < 0:
            print("Error: fibonacci sequence index cannot be negative")
            return None
        elif n == 0:
            return 0
        elif n == 1:
            return 1
        else:
            return self.fibonacci(n-1) + self.fibonacci(n-2)
```

Potete inizializzare la classe e sottoporla a test come nella Figura 6.12; come esempio, userò il metodo della somma.

```
calc = Calculator()

print(calc.add(3, 4))
```

```
7
```

Figura 6.12 Inizializzazione di una classe Python e utilizzo di uno dei suoi metodi.

Ora, supponiamo che voglia essere in grado di recuperare la documentazione delle docstring utilizzando il comando `Calculator?`, che restituisce in output la documentazione specifica della funzione o della classe Python richiamata in quel comando (Figura 6.13).

```
import pandas as pd
pd?

Type:        module
String form: <module 'pandas' from 'C:\\Users\\vaalt\\Anaconda3\\lib\\site-packages\\pandas\\__init__.py'>
File:        c:\users\vaalt\anaconda3\lib\site-packages\pandas\__init__.py
Docstring:
pandas - a powerful data analysis and manipulation library for Python
=====================================================================

**pandas** is a Python package providing fast, flexible, and expressive data
structures designed to make working with "relational" or "labeled" data both
easy and intuitive. It aims to be the fundamental high-level building block for
doing practical, **real world** data analysis in Python. Additionally, it has
the broader goal of becoming **the most powerful and flexible open source data
analysis / manipulation tool available in any language**. It is already well on
its way toward this goal.

Main Features
-------------
Here are just a few of the things that pandas does well:

  - Easy handling of missing data in floating point as well as non-floating
    point data.
  - Size mutability: columns can be inserted and deleted from DataFrame and
    higher dimensional objects
  - Automatic and explicit data alignment: objects can be explicitly aligned
    to a set of labels, or the user can simply ignore the labels and let
    `Series`, `DataFrame`, etc. automatically align the data for you in
    computations.
  - Powerful, flexible group by functionality to perform split-apply-combine
    operations on data sets, for both aggregating and transforming data.
  - Make it easy to convert ragged, differently-indexed data in other Python
    and NumPy data structures into DataFrame objects.
  - Intelligent label-based slicing, fancy indexing, and subsetting of large
    data sets.
  - Intuitive merging and joining data sets.
  - Flexible reshaping and pivoting of data sets.
  - Hierarchical labeling of axes (possible to have multiple labels per tick).
  - Robust IO tools for loading data from flat files (CSV and delimited),
    Excel files, databases, and saving/loading data from the ultrafast HDF5
    format.
  - Time series-specific functionality: date range generation and frequency
    conversion, moving window statistics, date shifting and lagging.
```

Figura 6.13 Esempio della documentazione della libreria pandas.

Chiediamo ora a ChatGPT di produrre lo stesso risultato per la nostra classe `Calculator`.

Da qui in poi, dopo aver dato la classe `Calculator` come contesto a ChatGPT, ho chiesto di generare la relativa documentazione (Figura 6.14).

Generami la documentazione per la classe Calculator in modo che gli sviluppatori possano leggerla eseguendo il comando Calculator? nel loro ambiente Python.

ChatGPT

Per generare la documentazione per la classe `Calculator`, puoi utilizzare la documentazione incorporata di Python, nota come docstring. Ecco un esempio di come potresti documentare la tua classe:

python Copy code

```python
import math

class Calculator:
    """
    Calculator class provides basic mathematical operations.

    Methods:
        - add(x, y): Returns the sum of two numbers.
        - subtract(x, y): Returns the difference between two numbers.
        - multiply(x, y): Returns the product of two numbers.
        - divide(x, y): Returns the result of dividing x by y. Handles
        - power(x, y): Returns x raised to the power of y.
        - square_root(x): Returns the square root of x. Handles negativ
        - logarithm(x, base): Returns the logarithm of x with the speci
        - factorial(x): Returns the factorial of x. Handles negative va
        - fibonacci(n): Returns the nth term in the Fibonacci sequence.

    Usage:
```

Figura 6.14 ChatGPT aggiorna il codice con la documentazione.

Di conseguenza, se aggiorniamo la nostra classe come mostrato nel codice precedente e chiamiamo `Calculator?`, otterremo l'output rappresentato nella Figura 6.15.

```
Calculator?

Init signature: Calculator()
Docstring:
A simple calculator that can perform various mathematical operations.

Methods
-------
add(x, y):
    Adds two numbers and returns the result.

subtract(x, y):
    Subtracts the second number from the first number and returns the result.

multiply(x, y):
    Multiplies two numbers and returns the result.

divide(x, y):
    Divides the first number by the second number and returns the result. If the second number is zero, returns None.

power(x, y):
    Raises the first number to the power of the second number and returns the result.

square_root(x):
    Calculates the square root of the number and returns the result. If the number is negative, returns None.

logarithm(x, base):
    Calculates the logarithm of the number with the given base and returns the result. If the number is negative or the base is invalid, returns None.

factorial(x):
    Calculates the factorial of the number and returns the result. If the number is negative, returns None.

fibonacci(n):
    Generates the n-th number in the Fibonacci sequence and returns the result. If n is negative, returns None.
Type:           type
Subclasses:
```

Figura 6.15 La nuova documentazione della classe Calculator.

Infine, ChatGPT può anche essere utilizzato per spiegare in linguaggio naturale le operazioni svolte da uno script, una funzione, una classe o altre componenti simili. Abbiamo già visto molti esempi di come ChatGPT arricchisca la sua risposta legata al codice con spiegazioni chiare. Tuttavia, possiamo potenziare questa capacità ponendo domande specifiche in termini di comprensione del codice.

Per esempio, partendo dal seguente script:

```
from azureml.opendatasets import NycTlcYellow
from datetime import datetime
from dateutil import parser
start_date = parser.parse('2018-05-01')
end_date = parser.parse('2018-05-07')
nyc_tlc = NycTlcYellow(start_date=start_date, end_date=end_date)
nyc_tlc_df = nyc_tlc.to_pandas_dataframe()
nyc_tlc_df.info()
```

Nella Figura 6.16 chiediamo a ChatGPT di descriverne il funzionamento.

ChatGPT

Questo codice utilizza il modulo `azureml.opendatasets` per accedere al set di dati pubblico "NYC Taxi & Limousine Commission (TLC) Yellow Taxi". Ecco una spiegazione del codice:

1. **Import delle librerie necessarie:**

```python
from azureml.opendatasets import NycTlcYellow
from datetime import datetime
from dateutil import parser
```

   - `NycTlcYellow` è una classe fornita da Azure Machine Learning per accedere ai dati delle corse in taxi giallo di New York.
   - `datetime` è utilizzato per lavorare con oggetti di data e ora.
   - `parser` è utilizzato per convertire stringhe in oggetti datetime.

2. **Definizione delle date di inizio e fine:**

```python
start_date = parser.parse('2018-05-01')
end_date = parser.parse('2018-05-07')
```

   Le variabili `start_date` e `end_date` vengono inizializzate con oggetti datetime corrispondenti alle date di inizio e fine (1 maggio 2018 e 7 maggio 2018).

3. **Recupero dei dati mediante la classe NycTlcYellow:**

```python
nyc_tlc = NycTlcYellow(start_date=start_date, end_date=end_date)
```

   Viene istanziata la classe `NycTlcYellow` con le date di inizio e fine specificate. Questo oggetto rappresenta il set di dati delle corse in taxi giallo di New York nel periodo specificato.

4. **Conversione dei dati in un DataFrame di pandas:**

```python
nyc_tlc_df = nyc_tlc.to_pandas_dataframe()
```

   La funzione `to_pandas_dataframe()` converte i dati ottenuti dall'oggetto `NycTlcYellow` in un DataFrame di pandas, rendendoli più facili da manipolare e analizzare.

5. **Visualizzazione delle informazioni del DataFrame:**

```python
nyc_tlc_df.info()
```

   La funzione `info()` mostra informazioni sul DataFrame, come il numero totale di righe, il tipo di dati di ciascuna colonna e la memoria utilizzata.

In sintesi, il codice accede ai dati delle corse in taxi giallo di New York nel periodo dal 1 maggio 2018 al 7 maggio 2018, li converte in un DataFrame di pandas e mostra le informazioni sul DataFrame ottenuto.

Figura 6.16 ChatGPT spiega uno script Python.

La spiegazione del codice può anche entrare a far parte della documentazione, oppure può essere utilizzata da sviluppatori che abbiano necessità di comprendere il codice sviluppato da altri team.

Grazie a ChatGPT e alle capacità menzionate in questo paragrafo, gli sviluppatori possono facilmente tenere traccia in linguaggio naturale del ciclo di vita del progetto, rendendolo più facile da

comprendere sia per i nuovi membri del team sia per gli utenti non tecnici.

Nel prossimo paragrafo vedremo come avere un codice “parlante” sia un passo fondamentale per l’interpretabilità dei modelli di machine learning nei progetti di data science.

Interpretabilità di un modello di machine learning

Nel contesto della data science, l’interpretabilità di un modello di machine learning si riferisce al grado di facilità con cui un essere umano può comprendere la logica sulla quale si basano gli output prodotti. Essenzialmente, è la capacità di comprendere come un modello arrivi alle sue decisioni e quali variabili contribuiscono alle sue previsioni.

Vediamo un esempio di interpretabilità del modello utilizzando una CNN (*Convolutional Neural Network*) di deep learning per la classificazione delle immagini. Ho costruito il mio modello in Python utilizzando Keras. A questo scopo, scaricherò il dataset CIFAR-10 direttamente da `keras.datasets`: si compone di 60.000 immagini a colori (quindi a tre canali) di 32x32 pixel in dieci classi (aereo, automobile, uccello, gatto, cervo, cane, rana, cavallo, nave e camion), con 6.000 immagini per classe.

```
model=tf.keras.Sequential()
model.add(tf.keras.layers.Conv2D(32,kernel_size=(3,3),activation='relu',
    input_shape=(32,32,1)))
model.add(tf.keras.layers.MaxPooling2D(pool_size=(2,2)))
model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(1024,activation='relu'))
model.add(tf.keras.layers.Dense(10,activation='softmax'))
```

Il codice precedente è composto da diversi layer della CNN che eseguono azioni differenti. Potremmo essere interessati ad avere una spiegazione della struttura del modello e dello scopo di ciascun layer.

Nella Figura 6.17 chiedo una mano a ChatGPT. Come potete vedere nella figura, ChatGPT è stato in grado di darmi una chiara spiegazione della struttura e dei layer della CNN. Aggiunge anche alcuni commenti e suggerimenti, come il fatto che l'uso del layer di max pooling aiuta a ridurre la dimensionalità dell'input.

ChatGPT può anche supportarmi nell'interpretare i risultati del modello nella fase di convalida. Quindi, dopo aver diviso i dati in set di training e set di test e aver addestrato il modello sul primo set, voglio vedere (Figura 6.18) le sue prestazioni sul set di test.

Chiedo poi a ChatGPT di approfondire le metriche di valutazione (l'output nella Figura 6.19 è troncato).

Ancora una volta, il risultato è davvero impressionante e ha fornito chiare indicazioni su come impostare gli esperimenti di machine learning in termini di set di training e set di test. ChatGPT spiega, infatti, quanto sia importante che il nostro modello di machine learning sia sufficientemente generalizzato in modo da prevenire l'*overfitting*, e quanto sia in grado di fornire risultati accurati anche su dati che non ha mai visto prima.

Ci sono molte ragioni per cui l'interpretabilità del modello è importante. Un elemento fondamentale è che riduce il divario tra gli utenti business e il codice dei modelli. Questo è fondamentale per permettere agli utenti non tecnici di comprendere il funzionamento di un modello, oltre a tradurlo in idee operative.

ChatGPT

Questo codice crea un modello di rete neurale convoluzionale (CNN) utilizzando il framework TensorFlow con l'API Keras. Ecco una spiegazione passo dopo passo:

1. **Inizializzazione del modello:**

```python
model = tf.keras.Sequential()
```

Viene creato un modello sequenziale, che rappresenta una sequenza lineare di strati. Gli strati possono essere aggiunti uno dopo l'altro nel modello.

2. **Aggiunta dello strato di convoluzione:**

```python
model.add(tf.keras.layers.Conv2D(32, kernel_size=(3, 3), activation='r
```

Viene aggiunto uno strato di convoluzione 2D al modello con 32 filtri (o kernel) di dimensione 3×3. L'attivazione 'relu' indica l'uso della funzione di attivazione ReLU. `input_shape=(32, 32, 1)` specifica le dimensioni dell'input della prima immagine come 32×32 pixel in scala di grigi (1 canale).

3. **Aggiunta dello strato di MaxPooling:**

```python
model.add(tf.keras.layers.MaxPooling2D(pool_size=(2, 2)))
```

Viene aggiunto uno strato di MaxPooling 2D con finestra di pooling 2×2. Questo strato riduce le dimensioni spaziali della rappresentazione e riduce il numero di parametri del modello.

4. **Aggiunta dello strato di appiattimento (Flatten):**

```python
model.add(tf.keras.layers.Flatten())
```

Viene aggiunto uno strato di appiattimento che converte l'output del layer precedente in un vettore piatto. Questo è necessario prima di passare ai layer completamente connessi.

5. **Aggiunta di uno strato completamente connesso (Dense):**

```python
model.add(tf.keras.layers.Dense(1024, activation='relu'))
```

Viene aggiunto uno strato completamente connesso con 1024 neuroni e attivazione ReLU.

6. **Aggiunta dello strato di output completamente connesso:**

```python
model.add(tf.keras.layers.Dense(10, activation='softmax'))
```

Viene aggiunto uno strato completamente connesso di output con 10 neuroni (corrispondenti alle classi) e attivazione softmax, che è comune negli strati di output per problemi di classificazione.

In sintesi, il modello creato è una CNN con uno strato di convoluzione seguito da uno strato di MaxPooling, uno strato di appiattimento e due strati completamente connessi. È progettato per l'input di immagini 32×32 in scala di grigi e prevede 10 classi di output.

Figura 6.17 Una domanda sull'interpretabilità del modello.

```
model.fit(np.resize(x_train, (50000,32,32,1))/255.0,
        tf.keras.utils.to_categorical(y_train),
        batch_size=batch_size,
        shuffle=True,
        epochs=epochs,
        validation_data=(np.resize(x_test, (10000,32,32,1))/255.0,
        tf.keras.utils.to_categorical(y_test))
        )
```

```
Epoch 1/3
1563/1563 [==============================] - 208s 133ms/step - loss: 2.3021 - accuracy: 0.1038 - val_loss: 2.3029 - val_accuracy: 0.1039
Epoch 2/3
1563/1563 [==============================] - 209s 133ms/step - loss: 2.2964 - accuracy: 0.1104 - val_loss: 2.3070 - val_accuracy: 0.1001
Epoch 3/3
1563/1563 [==============================] - 909s 582ms/step - loss: 2.2743 - accuracy: 0.1312 - val_loss: 2.3173 - val_accuracy: 0.1006
```

Figura 6.18 Metriche di valutazione.

You

Epoch 1/3
1563/1563 [==========] - 208s 133ms/step - loss: 2.3021 - accuracy: 0.1038 - val_loss: 2.3029 - val_accuracy: 0.1039
Epoch 2/3
1563/1563 [==========] - 209s 133ms/step - loss: 2.2964 - accuracy: 0.1104 - val_loss: 2.3070 - val_accuracy: 0.1001
Epoch 3/3
1563/1563 [==========] - 909s 582ms/step - loss: 2.2743 - accuracy: 0.1312 - val_loss: 2.3173 - val_accuracy: 0.1006

Ho ricevuto questo output durante la fase di testing del mio modello. Puoi elaborarle e spiegarmi cosa significano?

ChatGPT

L'output che hai ricevuto durante la fase di training del tuo modello è tipico di un addestramento con TensorFlow e Keras. Ecco una spiegazione delle informazioni presentate:

1. **Epoch 1/3:**
   - `1563/1563`: Indica che ci sono 1563 batch nel set di addestramento, e il modello ha completato l'addestramento su tutti questi batch durante l'epoca 1.
   - `208s`: Il tempo impiegato per completare l'epoca 1.
   - `loss: 2.3021`: Il valore della funzione di perdita (loss) sul set di addestramento alla fine dell'epoca 1.
   - `accuracy: 0.1038`: L'accuratezza del modello sul set di addestramento alla fine dell'epoca 1.
   - `val_loss: 2.3029`: Il valore della funzione di perdita sul set di validazione alla fine dell'epoca 1.
   - `val_accuracy: 0.1039`: L'accuratezza del modello sul set di validazione alla fine dell'epoca 1.

Figura 6.19 ChatGPT spiega le metriche di valutazione.

Inoltre, l'interpretabilità del modello risponde a uno dei principi chiave dell'AI responsabile ed etica: la trasparenza del pensiero e del comportamento di un modello su cui si basano i sistemi AI. Sbloccare l'interpretabilità del modello significa rilevare potenziali *bias* o comportamenti dannosi che un modello potrebbe avere mentre è in produzione ed evitare quindi che questi si verifichino.

Nel complesso, ChatGPT può fornire un prezioso supporto nel contesto dell'interpretabilità del modello, producendo spiegazioni chiare al livello della riga di codice (nell'esempio precedente, al livello del layer della CNN), come abbiamo visto nell'esempio precedente.

La prossima e ultima capacità di ChatGPT che esploreremo sarà un altro stimolo per la produttività degli sviluppatori, specialmente quando, nell'ambito dello stesso progetto, vengono utilizzati più linguaggi di programmazione.

Traduzione tra diversi linguaggi di programmazione

Nel Capitolo 5, abbiamo visto come ChatGPT abbia notevoli capacità nel tradurre da una lingua a un'altra. Quello che è veramente incredibile è che il linguaggio naturale non è il suo unico oggetto di traduzione. Infatti, ChatGPT è in grado di eseguire traduzioni tra linguaggi di programmazione differenti mantenendo lo stesso output e anche lo stesso stile (ovvero, preserva la documentazione *docstring*, se presente).

Ci sono molti scenari in cui questo potrebbe essere un punto di svolta.

Per esempio, potreste dover imparare un nuovo linguaggio di programmazione o uno strumento statistico mai visto prima perché dovete consegnare rapidamente un progetto basato interamente su tale linguaggio o strumento. Con l'aiuto di ChatGPT, potete iniziare a programmare nel vostro linguaggio preferito e poi chiedere a ChatGPT di tradurlo nel linguaggio desiderato, che imparerete nel corso del processo di traduzione.

Immaginate che il progetto debba essere consegnato in MATLAB (un software proprietario di calcolo numerico e programmazione

sviluppato da MathWorks), ma avete sempre programmato in Python. Il progetto consiste nel classificare le immagini del dataset del MNIST (*Modified National Institute of Standards and Technology*). La descrizione del dataset e il relativo paper possono essere trovati in http://yann.lecun.com/exdb/mnist). Il dataset contiene numerose cifre scritte a mano ed è frequentemente utilizzato per addestrare i sistemi di elaborazione delle immagini.

Per iniziare, ho scritto il seguente codice Python per inizializzare un modello di deep learning per la classificazione:

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
# Load the MNIST dataset
(x_train, y_train), (x_test, y_test) = keras.datasets.mnist.load_data()
# Preprocess the data
x_train = x_train.reshape(-1, 28*28) / 255.0
x_test = x_test.reshape(-1, 28*28) / 255.0
y_train = keras.utils.to_categorical(y_train)
y_test = keras.utils.to_categorical(y_test)
# Define the model architecture
model = keras.Sequential([
    layers.Dense(256, activation='relu', input_shape=(28*28,)),
    layers.Dense(128, activation='relu'),
    layers.Dense(10, activation='softmax')
])
# Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
# Train the model
history = model.fit(x_train, y_train, validation_split=0.2, epochs=10,
batch_size=128)
# Evaluate the model
test_loss, test_acc = model.evaluate(x_test, y_test, verbose=0)
print('Test accuracy:', test_acc)
```

Vediamo nella Figura 6.20 che cosa succede se diamo il codice precedente come contesto a ChatGPT e gli chiediamo di tradurlo in MATLAB.

ChatGPT

Ecco la traduzione del codice Python in MATLAB:

matlab Copy code

```
% Load the MNIST dataset
[x_train, y_train, x_test, y_test] = mnist.load_data();

% Preprocess the data
x_train = reshape(x_train, [], 28*28) / 255.0;
x_test = reshape(x_test, [], 28*28) / 255.0;
y_train = categorical(y_train);
y_test = categorical(y_test);

% Define the model architecture
model = Sequential();
model.add(Dense(256, 'activation', 'relu', 'input_shape', [28*28]));
model.add(Dense(128, 'activation', 'relu'));
model.add(Dense(10, 'activation', 'softmax'));

% Compile the model
model.compile('optimizer', 'adam', 'loss', 'categorical_crossentropy',

% Train the model
history = model.fit(x_train, y_train, 'validation_split', 0.2, 'epochs'

% Evaluate the model
[test_loss, test_acc] = model.evaluate(x_test, y_test, 'verbose', 0);
fprintf('Test accuracy: %.4f\n', test_acc);
```

Figura 6.20 ChatGPT traduce il codice Python in MATLAB.

Vediamo (Figura 6.21) se è in grado di tradurlo anche in altri linguaggi, come JavaScript.

ChatGPT

La traduzione del codice in JavaScript richiede l'utilizzo di un ambiente che supporti le operazioni di deep learning. Un'opzione comune è l'utilizzo di TensorFlow.js, che è una libreria JavaScript per l'addestramento e l'inferenza di modelli di machine learning nel browser o su Node.js. Di seguito è riportata una versione approssimativa del codice Python tradotto in JavaScript utilizzando TensorFlow.js:

javascript Copy code

```javascript
const tf = require('@tensorflow/tfjs-node');
const { mnist } = require('tfjs-mnist-dataset');

// Load the MNIST dataset
const { images: xTrain, labels: yTrain } = mnist.getTrainData();
const { images: xTest, labels: yTest } = mnist.getTestData();

// Preprocess the data
const flatten = (arr) => arr.reduce((acc, val) => acc.concat(val), []);
const xTrainProcessed = tf.tensor(flatten(xTrain), [xTrain.length, 28 *
const xTestProcessed = tf.tensor(flatten(xTest), [xTest.length, 28 * 28
const yTrainProcessed = tf.oneHot(tf.tensor1d(yTrain, 'int32'), 10);
const yTestProcessed = tf.oneHot(tf.tensor1d(yTest, 'int32'), 10);

// Define the model architecture
const model = tf.sequential();
model.add(tf.layers.dense({ units: 256, activation: 'relu', inputShape:
model.add(tf.layers.dense({ units: 128, activation: 'relu' }));
model.add(tf.layers.dense({ units: 10, activation: 'softmax' }));

// Compile the model
model.compile({ optimizer: 'adam', loss: 'categoricalCrossentropy', met

// Train the model
const history = await model.fit(xTrainProcessed, yTrainProcessed, {
  epochs: 10,
  validationSplit: 0.2,
  batchSize: 128,
});

// Evaluate the model
const testResult = model.evaluate(xTestProcessed, yTestProcessed);
console.log('Test accuracy:', testResult[1].dataSync()[0]);
```

Figura 6.21 ChatGPT traduce il codice Python in JavaScript.

La traduzione del codice può anche contribuire a ridurre il divario di competenze tra le nuove tecnologie e le attuali competenze di programmazione.

Un'altra importante implicazione della traduzione del codice è la modernizzazione delle applicazioni. Infatti, immaginate di voler rinnovare il vostro stack di applicazioni, per esempio migrando al

cloud. Potreste decidere di iniziare con un semplice *lift and shift* verso istanze IaaS (*Infrastructure-as-a-Service*), come le macchine virtuali Windows o Linux. Tuttavia, in una seconda fase, potreste voler passare al refactoring, rearchitecting o, addirittura, alla riscrittura da zero delle vostre applicazioni.

Il diagramma rappresentato nella Figura 6.22 mostra le varie opzioni per la modernizzazione delle applicazioni.

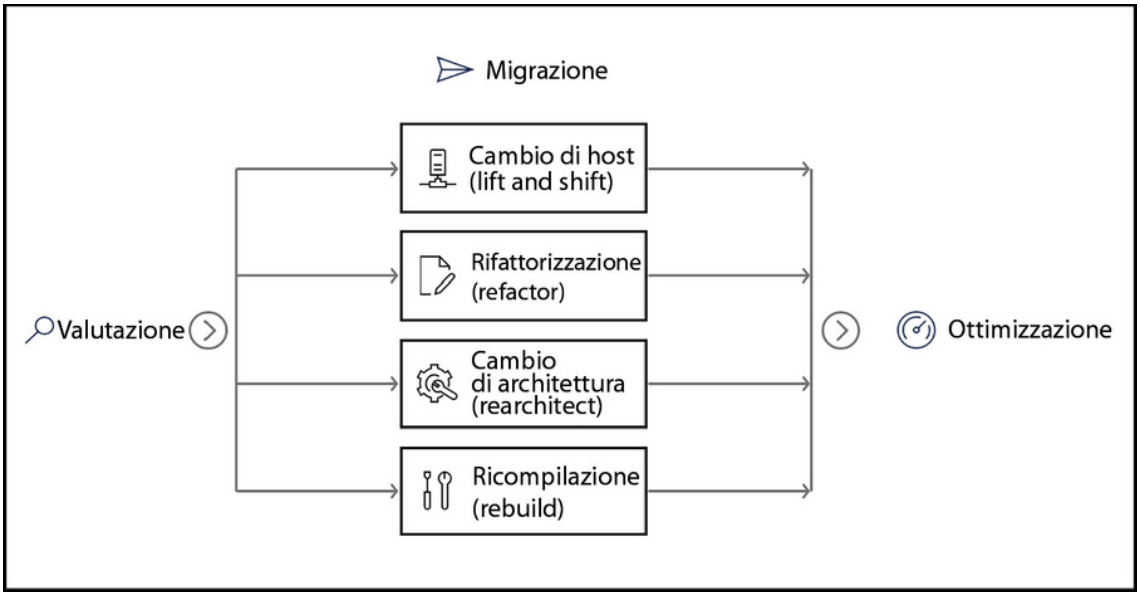


Figura 6.22 Quattro modi per migrare le vostre applicazioni al cloud pubblico.

Anche in questo caso, ChatGPT può essere un ottimo assistente. Per mostrarlo, prendiamo come esempio i mainframe.

I mainframe sono computer principalmente utilizzati da grandi organizzazioni per svolgere compiti essenziali come l'elaborazione di grandi masse di dati per attività come censimenti, statistiche dei consumatori e del settore, pianificazione delle risorse aziendali ed elaborazione di transazioni su larga scala. Il linguaggio di programmazione delle applicazioni dell'ambiente mainframe è il COBOL (*Common Business Oriented Language*). Nonostante sia stato

inventato nel 1959, il COBOL è ancora in uso oggi e, dunque, è uno dei linguaggi di programmazione più antichi ancora utilizzati.

Man mano che la tecnologia continua a migliorare, le applicazioni che risiedono nel regno dei mainframe sono state sottoposte a un processo continuo di migrazione e modernizzazione, volto a migliorare l'infrastruttura legacy dei mainframe esistenti in aree come l'interfaccia, il codice, i costi, le prestazioni e la manutenibilità.

Ovviamente, ciò implica la traduzione del COBOL in linguaggi di programmazione più moderni, come C# o Java. Il problema è che il COBOL è sconosciuto alla maggior parte dei programmatori di nuova generazione; c'è quindi un enorme divario di competenze in questo contesto.

Consideriamo uno script COBOL che legge un file di record dei dipendenti e scrive un report dei dipendenti che sono con l'azienda da più di dieci anni:

```
IDENTIFICATION DIVISION.
PROGRAM-ID. EMPLOYEEREPORT.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT EMPLOYEE-FILE ASSIGN TO 'EMPLOYEE.DAT'
        ORGANIZATION IS LINE SEQUENTIAL.
    SELECT REPORT-FILE ASSIGN TO 'EMPLOYEEREPORT.TXT'
        ORGANIZATION IS LINE SEQUENTIAL.
DATA DIVISION.
FILE SECTION.
FD EMPLOYEE-FILE.
01 EMPLOYEE-RECORD.
    05 EMPLOYEE-ID     PIC X(10).
    05 HIRE-DATE       PIC 9(6).
    05 EMPLOYEE-NAME   PIC X(30).
FD REPORT-FILE.
01 REPORT-LINE         PIC X(80).
WORKING-STORAGE SECTION.
01 WS-CURRENT-DATE     PIC 9(6).
01 WS-EMPLOYEE-YEARS   PIC 9(2).
01 WS-REPORT-HEADER.
    05 FILLER          PIC X(30) VALUE 'EMPLOYEES WITH 10+ YEARS OF
SERVICE'.
    05 FILLER          PIC X(10) VALUE SPACES.
    05 WS-REPORT-DATE  PIC X(6).
PROCEDURE DIVISION.
```

```
MAIN-PARAGRAPH.
    OPEN INPUT EMPLOYEE-FILE
    OPEN OUTPUT REPORT-FILE
    MOVE FUNCTION CURRENT-DATE TO WS-CURRENT-DATE
    MOVE WS-CURRENT-DATE(1:6) TO WS-REPORT-DATE
    WRITE REPORT-LINE FROM WS-REPORT-HEADER
    WRITE REPORT-LINE FROM SPACES
    PERFORM READ-EMPLOYEE UNTIL EMPLOYEE-FILE-STATUS = '10'
    CLOSE EMPLOYEE-FILE
    CLOSE REPORT-FILE
    STOP RUN.
READ-EMPLOYEE.
    READ EMPLOYEE-FILE
        AT END MOVE '10' TO EMPLOYEE-FILE-STATUS
    END-READ
    COMPUTE WS-EMPLOYEE-YEARS = FUNCTION INTEGER-OF-DATE(WS-CURRENT-
DATE) - HIRE-DATE(1:6)
    IF WS-EMPLOYEE-YEARS >= 10
        WRITE REPORT-LINE FROM EMPLOYEE-RECORD
    END-IF.
```

Ho passato lo script COBOL precedente a ChatGPT, affinché lo possa usare come contesto per formulare la sua risposta. Nella Figura 6.23 ho chiesto a ChatGPT di tradurre quello script in codice JavaScript (output troncato).

Strumenti come ChatGPT possono aiutare a ridurre il divario di competenze in questo e in altri scenari simili, introducendo uno strato che conosce sia il passato sia il futuro della programmazione.

ChatGPT

Ecco una traduzione approssimativa del codice COBOL fornito in JavaScript:

javascript Copy code

```javascript
const fs = require('fs');

// File control
const employeeFile = 'EMPLOYEE.DAT';
const reportFile = 'EMPLOYEEREPORT.TXT';

// Data structures
let employeeRecord = {
    EMPLOYEE_ID: '',
    HIRE_DATE: 0,
    EMPLOYEE_NAME: ''
};

let reportLine = '';

// Working storage
let currentDate = '';
let employeeYears = 0;
let reportHeader = 'EMPLOYEES WITH 10+ YEARS OF SERVICE            ';
let reportDate = '';

// Main program
function main() {
    const employeeFileStream = fs.readFileSync(employeeFile, 'utf8').sp
    const reportFileStream = fs.createWriteStream(reportFile);

    currentDate = getCurrentDate();
    reportDate = currentDate.slice(0, 6);
```

Figura 6.23 ChatGPT traduce il COBOL in JavaScript.

In conclusione, ChatGPT può essere uno strumento efficace per la modernizzazione delle applicazioni, fornendo l'aggiornamento del codice oltre a preziose intuizioni e raccomandazioni per migliorare i sistemi legacy. Con le sue avanzate capacità di elaborazione del linguaggio e la sua vasta base di conoscenze, ChatGPT può aiutare le

organizzazioni a semplificare i loro sforzi di modernizzazione, rendendo il processo più veloce, efficiente ed efficace.

Riepilogo

ChatGPT può essere una risorsa preziosa per gli sviluppatori che cercano di migliorare le loro competenze e semplificare i loro flussi di lavoro. Abbiamo iniziato vedendo come ChatGPT può generare, ottimizzare e correggere il codice, ma abbiamo anche trattato ulteriori capacità come la generazione della documentazione del codice, la spiegazione dei modelli di machine learning e la traduzione tra diversi linguaggi di programmazione, avendo come scopo la modernizzazione delle applicazioni.

Che siate sviluppatori esperti o alle prime armi, ChatGPT vi offre uno strumento potente per l'apprendimento e la crescita, riducendo il divario tra il codice e il linguaggio naturale.

Nel prossimo capitolo, approfondiremo un altro ambito di applicazione in cui ChatGPT potrebbe essere un punto di svolta: il marketing.

Capitolo 7

Rivoluzionare il marketing con ChatGPT

In questo capitolo, ci concentreremo su come i professionisti del marketing possano sfruttare ChatGPT, esaminando i principali casi d'uso dello strumento in questo settore e di come si presti a essere un utile assistente.

Alla fine di questo capitolo, sarete in grado di utilizzare ChatGPT per attività correlate al marketing e per aumentare la vostra produttività in questo settore.

Requisiti tecnici

Avrete bisogno di un account OpenAI per accedere a ChatGPT e DALL-E.

Applicazioni di ChatGPT per i professionisti del marketing

Il marketing è probabilmente il settore in cui il potere creativo dei modelli di ChatGPT e, in generale, di OpenAI possono essere sfruttati nella loro forma più pura. Possono essere strumenti pratici per supportare lo sviluppo creativo in termini di nuovi prodotti, campagne di marketing, ottimizzazione per i motori di ricerca e così via. Nel

complesso, i professionisti del marketing possono automatizzare e semplificare molti aspetti del loro lavoro, migliorando allo stesso tempo la qualità e l'efficacia dei loro sforzi di marketing.

Vediamo un esempio. Uno dei casi d'uso più prominenti e promettenti di ChatGPT in questo settore è il marketing personalizzato. ChatGPT può essere utilizzato per analizzare i dati dei clienti e generare messaggi di marketing personalizzati per singolo cliente. Per esempio, un team di marketing può utilizzare ChatGPT per analizzare i dati dei clienti e sviluppare campagne e-mail mirate in base alle specifiche preferenze e comportamenti dei clienti. Questo può aumentare il *conversion rate* dei clienti e portare a una maggiore *customer satisfaction*. Fornendo intuizioni sul sentiment e il comportamento del cliente, generando messaggi di marketing personalizzati, fornendo supporto al cliente personalizzato e generando contenuti, ChatGPT può aiutare i professionisti del marketing a offrire esperienze migliori ai clienti e a guidare la crescita aziendale.

Nei prossimi paragrafi, esamineremo esempi concreti di progetti di marketing end-to-end supportati da ChatGPT.

Sviluppo di nuovi prodotti e strategia di go-to-market

Il primo modo in cui potete introdurre ChatGPT nella vostra attività di marketing potrebbe essere come assistente nello sviluppo di nuovi prodotti e nella strategia di *go-to-market* (GTM).

In questo paragrafo, esamineremo passo dopo passo come sviluppare e promuovere un nuovo prodotto. Immaginiamo, per esempio, di essere i proprietari di un marchio di abbigliamento da corsa chiamato *RunFast* e di aver prodotto, finora, solo scarpe. Vogliamo espandere il nostro business con una nuova linea di prodotti.

Inizieremo con un brainstorming per creare una strategia di go-to-market. Naturalmente, tutto è supportato da ChatGPT.

- *Brainstorming*: la prima attività con la quale ChatGPT può aiutarci è il brainstorming e la stesura di opzioni per la nostra nuova linea di prodotti. Fornirà anche il ragionamento su cui si basa ogni suggerimento. Nella Figura 7.1, chiediamo a ChatGPT quale tipo di nuova linea di prodotti dovremmo considerare.

You

Sono proprietaria di un brand di abbigliamento sportivo che produce scarpe da corsa. Oltre alle scarpe, che altre linee di prodotto potrei lanciare, sempre relativi alla corsa? Genera 3 idee.

‹ 3 / 3 ›

ChatGPT

Certamente! Ecco tre idee per linee di prodotti correlate alla corsa che potresti considerare per espandere il tuo brand di abbigliamento sportivo:

1. **Abbigliamento Tecnico da Corsa:**
   Introduce una linea di abbigliamento tecnico progettato specificamente per i corridori. Questa potrebbe includere maglie traspiranti, pantaloni da corsa leggeri, giacche antivento e cappelli termoregolanti. Assicurati di utilizzare materiali di alta qualità che offrano comfort, traspirabilità e resistenza all'umidità per migliorare le prestazioni durante la corsa.
2. **Accessori per il Running:**
   Crea una serie di accessori pensati per migliorare l'esperienza di corsa. Questi potrebbero includere cinture portaoggetti ergonomiche per smartphone, cappellini e visiere, guanti leggeri e calze tecniche. Puoi anche considerare l'introduzione di gadget elettronici, come orologi GPS, auricolari wireless progettati appositamente per i corridori e bracciali fitness.
3. **Attrezzatura per il Recupero Post-Corsa:**
   Concentrati sul benessere complessivo dei corridori offrendo una linea di prodotti per il recupero post-corsa. Questi potrebbero includere rulli per il massaggio muscolare, fasce elastiche per lo stretching, palle per il rilassamento muscolare, e prodotti a compressione graduata per favorire il recupero muscolare. Una gamma completa di prodotti per il recupero potrebbe essere un modo efficace per distinguere il tuo brand nel mercato.

Ricorda di adattare ogni nuova linea di prodotti ai valori e agli standard qualitativi del tuo marchio, mantenendo coerenza nel design e nell'esperienza del cliente.

Figura 7.1 Esempio di nuove idee generate da ChatGPT.

Tra i tre suggerimenti, sceglieremo il primo e, nello specifico, ci concentreremo sui calzini da corsa.

- *Nome del prodotto*: ora che abbiamo la nostra idea ben chiara in mente, dobbiamo pensare a un nome accattivante. Ancora una volta, chiediamo a ChatGPT (Figura 7.2) alcune opzioni, in modo

da poter poi scegliere la nostra preferita.
AirStep suona bene: andremo avanti con quello.

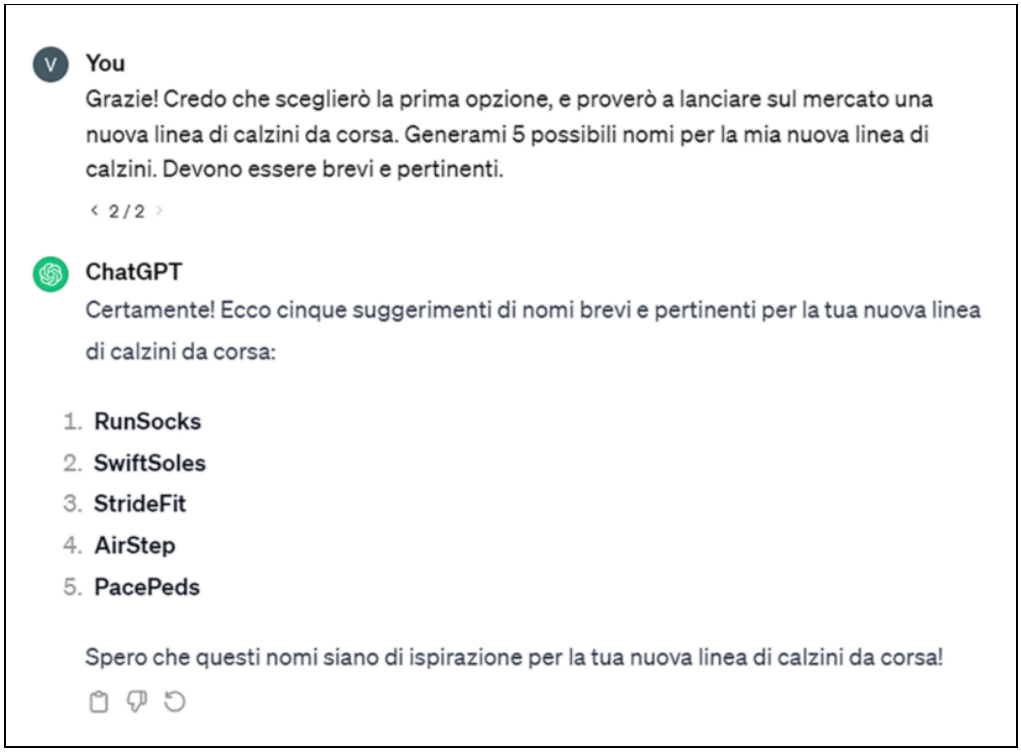


Figura 7.2 Un elenco di potenziali nomi di prodotti.

- *Generare slogan accattivanti*: oltre al nome del prodotto, vogliamo anche convogliare l'intento dietro il nome e la *mission* della linea di prodotti, in modo da attirare il pubblico target. Vogliamo ispirare fiducia e lealtà nei nostri clienti e vogliamo che si vedano riflessi nella missione dietro la nuova linea di prodotti (Figura 7.3).

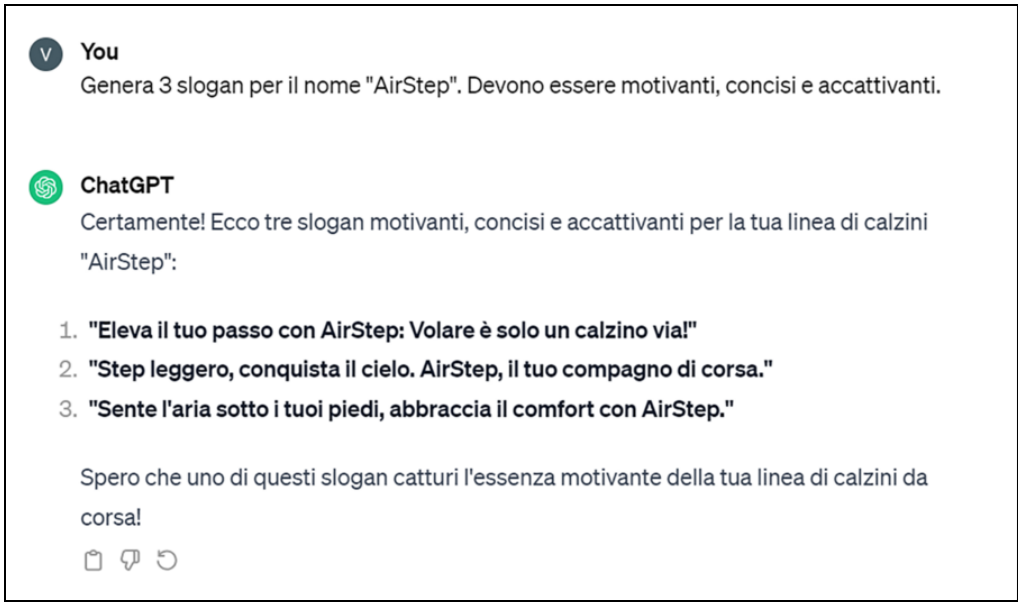


Figura 7.3 Un elenco di slogan per il nuovo prodotto.

Ottimo: possiamo ritenerci soddisfatti del nome del prodotto e dello slogan che useremo per creare un annuncio unico sui social media. Prima di farlo, vogliamo dedicare del tempo alla ricerca di mercato per il pubblico target (Figura 7.4).

You
A che tipo di audience dovrei rivolgermi per la promozione della mia nuova linea di prodotto AirStep? Citami le top 3 categorie.

‹ 2 / 2 ›

ChatGPT
Per la promozione della tua nuova linea di calzini da corsa AirStep, potresti concentrarti su audience specifiche che mostrano un interesse particolare per il running e uno stile di vita attivo. Ecco le top 3 categorie di audience a cui potresti rivolgerti:

1. **Corridori Appassionati:**
   Targettizza coloro che sono appassionati di corsa, partecipano a gare o seguono programmi di allenamento strutturati. Questa categoria di persone sarà più incline ad apprezzare e investire in prodotti di alta qualità che migliorano le loro prestazioni durante la corsa.
2. **Fitness Enthusiasts:**
   Rivolgiti a individui che amano mantenere uno stile di vita attivo, partecipando a diverse attività fitness come corsa, allenamenti in palestra o sport all'aperto. Questa categoria può essere sensibile al comfort e alla performance, rendendo i calzini da corsa AirStep un'aggiunta attraente alla loro attrezzatura sportiva.
3. **Stile di Vita Salutare e Attivo:**
   Attrai coloro che cercano uno stile di vita complessivamente sano e attivo. Questo gruppo può includere non solo corridori dedicati, ma anche persone che fanno lunghe passeggiate, partecipano a eventi di fitness o semplicemente vogliono mantenere una vita attiva. Concentrati su come i tuoi calzini possono migliorare il comfort e la soddisfazione durante queste attività.

Segmentare la tua audience in modo mirato su queste categorie ti aiuterà a creare messaggi di marketing più efficaci e ad aumentare la visibilità della tua linea di calzini da corsa AirStep tra coloro che sono più inclini a essere interessati e a fare acquisti.

Figura 7.4 Elenco di gruppi di persone target da raggiungere con la nuova linea di prodotti.

È importante avere in mente diversi *cluster* all'interno del pubblico target, in modo da poter differenziare i messaggi. Nel nostro caso, vogliamo assicurarci che la nostra linea di prodotti si rivolga a diversi gruppi di persone, come i runner competitivi, gli sportivi casuali e gli appassionati di fitness.

- *Varianti di prodotto e canali di vendita*: in base ai precedenti cluster di potenziali clienti, possiamo generare varianti di prodotto più adatte a pubblici specifici (Figura 7.5).

You
Rispetto ai 3 gruppi target generati, suggeriscimi 3 varianti del prodotto AirStep che sono specificatamente indirizzate a tali gruppi.

ChatGPT
Certamente! Ecco tre varianti del prodotto AirStep, ciascuna pensata per soddisfare le esigenze specifiche di uno dei tre gruppi target:

1. **AirStep ProRun:**
   - *Destinato ai Corridori Appassionati*
   - Caratteristiche: Materiale ultraleggero, traspirante e resistente all'umidità per massimizzare le prestazioni durante le corse. Supporto specifico nell'area dell'arco plantare per una maggiore stabilità. Design ergonomico per ridurre l'attrito e prevenire le vesciche.
2. **AirStep ActiveFit:**
   - *Destinato agli Enthusiasts del Fitness*
   - Caratteristiche: Tessuto elastico che si adatta al movimento durante una varietà di attività fisiche. Struttura a compressione leggera per una migliore circolazione sanguigna. Zone di ventilazione per mantenere i piedi freschi durante gli allenamenti.
3. **AirStep LifestyleComfort:**
   - *Destinato a chi segue uno Stile di Vita Salutare e Attivo*
   - Caratteristiche: Comfort superiore per un uso quotidiano, ideale per lunghe passeggiate e stile di vita attivo. Materiale morbido e traspirante con supporto leggero. Design alla moda per un look casual senza rinunciare al comfort.

Queste varianti sono progettate per rispondere alle specifiche esigenze e preferenze dei gruppi target, offrendo caratteristiche mirate che rendono i calzini AirStep una scelta ideale per ciascun segmento di mercato.

Figura 7.5 Esempio di varianti della linea di prodotti.

Allo stesso modo, possiamo anche chiedere a ChatGPT di suggerire diversi canali di vendita (Figura 7.6) per ciascuno dei gruppi precedenti.

You
Quale potrebbe essere il miglior canale di comunicazione per sponsorizzare il mio prodotto, per ciascuno dei gruppi elencati?

ChatGPT
Per sponsorizzare la tua linea di calzini da corsa AirStep e raggiungere efficacemente ciascun gruppo target, puoi utilizzare diversi canali di comunicazione mirati. Ecco suggerimenti per ciascun gruppo:

1. **Corridori Appassionati (AirStep ProRun):**
   - *Miglior Canale:* Sponsorizza attraverso eventi di corsa, gare e maratone. Collabora con corridori influenti e atleti di corsa per recensioni e testimonianze. Utilizza i social media specializzati nella corsa, come piattaforme dedicate alle app di corsa e gruppi online.
2. **Enthusiasts del Fitness (AirStep ActiveFit):**
   - *Miglior Canale:* Coinvolgi fitness influencer su piattaforme come Instagram e YouTube. Sponsorizza in palestre, centri fitness e eventi sportivi locali. Utilizza il marketing digitale, come pubblicità online su siti web e app fitness.
3. **Stile di Vita Salutare e Attivo (AirStep LifestyleComfort):**
   - *Miglior Canale:* Focalizza sulla promozione attraverso blog e siti web incentrati sul benessere e uno stile di vita sano. Partecipa a fiere del benessere e eventi legati alla vita attiva. Utilizza i social media generalisti come Facebook, con enfasi sul comfort e lo stile di vita sano.

Adattare la tua strategia di marketing ai canali preferiti di ciascun gruppo target ti aiuterà a massimizzare l'impatto della tua campagna promozionale e a raggiungere direttamente coloro che sono più inclini ad essere interessati ai tuoi calzini da corsa AirStep.

Figura 7.6 Suggerimenti per diversi canali di vendita.

- *Distinguersi dalla concorrenza*: vogliamo che la nostra linea di prodotti si distingua dalla concorrenza ed emerga in un mercato molto saturo: vogliamo renderla unica. Con questo scopo in mente, ho chiesto a ChatGPT di includere considerazioni sociali come la sostenibilità e l'inclusività. Nella Figura 7.7 chiediamo a ChatGPT alcuni suggerimenti a riguardo.

ChatGPT

Integrare elementi di sostenibilità nella tua linea di calzini da corsa AirStep può essere una strategia vincente per differenziarti nel mercato competitivo. Ecco alcune idee:

1. **Materiali Sostenibili:**
   - Utilizza materiali ecologici e sostenibili per la produzione dei calzini, come fibre riciclate, cotone biologico o tessuti innovativi a basso impatto ambientale. Comunica chiaramente la tua scelta di materiali sostenibili per attirare consumatori orientati alla sostenibilità.
2. **Processi Produttivi Ecologici:**
   - Ottimizza i processi di produzione per ridurre l'impatto ambientale. Puoi adottare pratiche di produzione a basso consumo energetico, riciclo dell'acqua o imballaggi sostenibili. Trasmetti questi sforzi sostenibili attraverso la tua comunicazione di marca.
3. **Programmi di Riciclo o Upcycling:**
   - Implementa programmi di riciclo dei calzini usati o incoraggia gli acquirenti a restituire i loro vecchi AirStep per essere riciclati o riutilizzati in modo creativo. Questa iniziativa non solo riduce gli sprechi, ma crea anche un legame più forte con i consumatori che apprezzano la responsabilità ambientale.
4. **Messaggistica sulla Sostenibilità:**
   - Comunica chiaramente gli sforzi sostenibili del tuo brand attraverso la tua messaggistica di marketing. Mostra quanto hai ridotto l'impatto ambientale e sottolinea il contributo positivo che i clienti fanno scegliendo i calzini AirStep.
5. **Collaborazioni con Organizzazioni Ambientali:**
   - Collabora con organizzazioni ambientali o associazioni che promuovono la sostenibilità. Queste partnership possono aumentare la tua credibilità e dimostrare il tuo impegno a preservare l'ambiente.

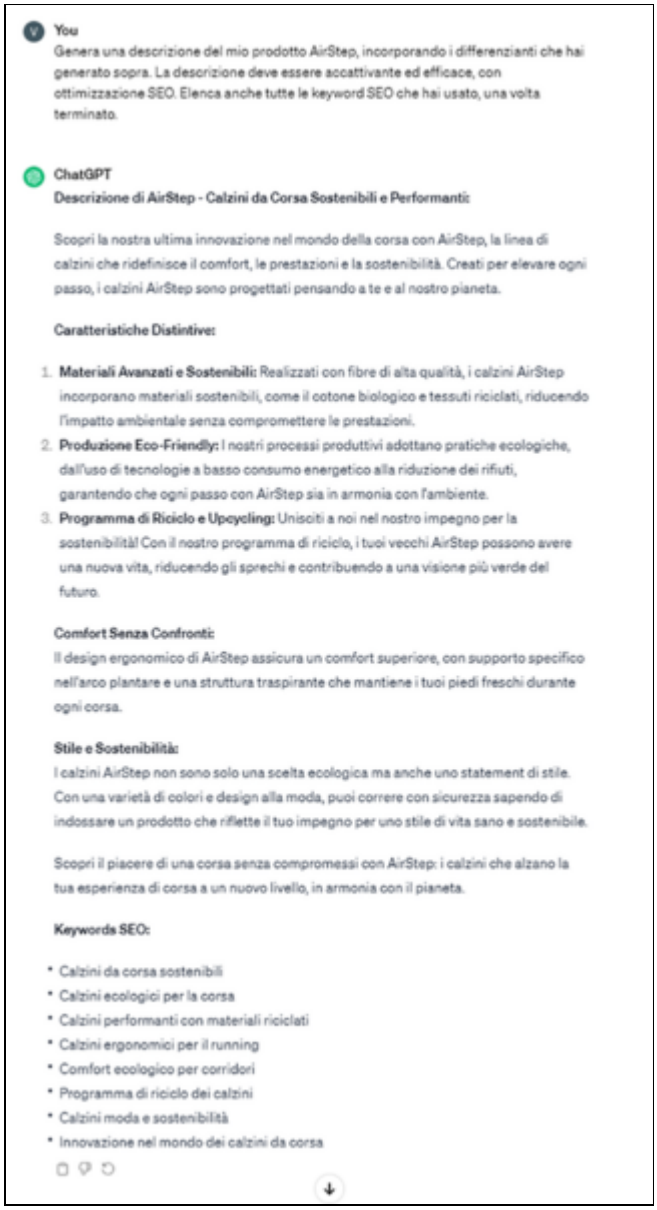
La sostenibilità non solo ti distinguerà nel mercato, ma potrebbe anche attrarre consumatori sempre più consapevoli dell'impatto ambientale delle loro scelte di acquisto.

Figura 7.7 Esempio di caratteristiche eccezionali generate da ChatGPT.

Come potete vedere, ChatGPT è stato in grado di generare caratteristiche interessanti che potrebbero rendere unica la linea di prodotti.

- *Descrizione del prodotto:* ora è il momento di iniziare a costruire il nostro piano *go-to-market*. Prima di tutto, vogliamo generare

una descrizione del prodotto da mettere sul sito web (Figura 7.8), includendo tutti i precedenti differenziatori unici.



You

Genera una descrizione del mio prodotto AirStep, incorporando i differenzianti che hai generato sopra. La descrizione deve essere accattivante ed efficace, con ottimizzazione SEO. Elenca anche tutte le keyword SEO che hai usato, una volta terminato.

ChatGPT

Descrizione di AirStep - Calzini da Corsa Sostenibili e Performanti:

Scopri la nostra ultima innovazione nel mondo della corsa con AirStep, la linea di calzini che ridefinisce il comfort, le prestazioni e la sostenibilità. Creati per elevare ogni passo, i calzini AirStep sono progettati pensando a te e al nostro pianeta.

Caratteristiche Distintive:

1. **Materiali Avanzati e Sostenibili:** Realizzati con fibre di alta qualità, i calzini AirStep incorporano materiali sostenibili, come il cotone biologico e tessuti riciclati, riducendo l'impatto ambientale senza compromettere le prestazioni.
2. **Produzione Eco-Friendly:** I nostri processi produttivi adottano pratiche ecologiche, dall'uso di tecnologie a basso consumo energetico alla riduzione dei rifiuti, garantendo che ogni passo con AirStep sia in armonia con l'ambiente.
3. **Programma di Riciclo e Upcycling:** Unisciti a noi nel nostro impegno per la sostenibilità! Con il nostro programma di riciclo, i tuoi vecchi AirStep possono avere una nuova vita, riducendo gli sprechi e contribuendo a una visione più verde del futuro.

Comfort Senza Confronti:
Il design ergonomico di AirStep assicura un comfort superiore, con supporto specifico nell'arco plantare e una struttura traspirante che mantiene i tuoi piedi freschi durante ogni corsa.

Stile e Sostenibilità:
I calzini AirStep non sono solo una scelta ecologica ma anche uno statement di stile. Con una varietà di colori e design alla moda, puoi correre con sicurezza sapendo di indossare un prodotto che riflette il tuo impegno per uno stile di vita sano e sostenibile.

Scopri il piacere di una corsa senza compromessi con AirStep: i calzini che alzano la tua esperienza di corsa a un nuovo livello, in armonia con il pianeta.

Keywords SEO:

- Calzini da corsa sostenibili
- Calzini ecologici per la corsa
- Calzini performanti con materiali riciclati
- Calzini ergonomici per il running
- Comfort ecologico per corridori
- Programma di riciclo dei calzini
- Calzini moda e sostenibilità
- Innovazione nel mondo dei calzini da corsa

Figura 7.8 Esempio di descrizione e parole chiave SEO.

- *Prezzo equo*: un altro elemento chiave consiste nel determinare un prezzo equo per il nostro prodotto. Poiché ho differenziato tra varianti di prodotto per diversi pubblici (corridori competitivi, sportivi casuali e appassionati di fitness), voglio anche avere una

gamma di prezzi che tenga conto di questo raggruppamento (Figura 7.9).

You
Come faccio a stabilire un prezzo equo per la mia nuova linea di prodotti?
‹ 2 / 2 ›

ChatGPT
Stabilire un prezzo equo per la tua nuova linea di calzini da corsa richiede una valutazione attenta che tenga conto di vari fattori. Ecco alcuni passaggi che potresti seguire:

1. **Costi di Produzione:**
   - Calcola tutti i costi associati alla produzione dei calzini, inclusi materiali, manodopera, produzione, imballaggio e spedizione. Assicurati di comprendere i costi totali per ogni paio di calzini.
2. **Analisi del Mercato:**
   - Esamina i prezzi dei concorrenti nel mercato. Questo ti aiuterà a ottenere un'idea delle aspettative dei consumatori e a posizionare i tuoi calzini in modo competitivo.
3. **Posizionamento del Marchio:**
   - Rifletti sul posizionamento del tuo marchio. Se hai posizionato AirStep come un prodotto premium, il prezzo dovrebbe rispecchiare la qualità e il valore aggiunto offerti.
4. **Percezione del Valore:**
   - Considera il valore percepito dai consumatori. Se i calzini offrono caratteristiche uniche, come sostenibilità o prestazioni avanzate, puoi riflettere questo valore nel prezzo.
5. **Segmentazione della Clientela:**
   - Se hai diverse varianti di prodotti, come livelli di prestazioni o stili, puoi considerare diverse fasce di prezzo per soddisfare diverse esigenze dei clienti.
6. **Obiettivi di Redditività:**
   - Stabilisci gli obiettivi finanziari per la tua linea di calzini. Considera la redditività desiderata e assicurati che il prezzo copra tutti i costi e generi un margine di profitto adeguato.
7. **Strategia di Marketing e Promozioni:**
   - Se prevedi di utilizzare strategie di marketing o promozioni, assicurati di considerare il loro impatto sui margini di profitto e sul prezzo finale.
8. **Risposta del Mercato:**
   - Fai test di prezzo o ricerche di mercato per ottenere feedback sulla reazione dei consumatori a diversi livelli di prezzo.

Ricorda che il prezzo deve riflettere il valore del prodotto e fornire una giusta compensazione per i costi e gli sforzi di produzione. Equilibra l'aspetto competitivo con la tua strategia di posizionamento del marchio per creare un prezzo equo e sostenibile nel lungo termine.

Figura 7.9 Identificare un prezzo equo.

Ci siamo quasi. Abbiamo affrontato molti passaggi di sviluppo di nuovi prodotti e di *go-to-market*, e in ognuno di essi, ChatGPT è stato un grande strumento di supporto. Come ultima cosa, possiamo chiedere a ChatGPT di generare un post su Instagram riguardante il

nostro nuovo prodotto, includendo *hashtag* rilevanti e parole chiavi in ottica SEO (Figura 7.10). Poi possiamo generare l'immagine con DALL-E.

You
Voglio generare un Instagram post per annunciare il lancio di AirStep. Scrivi un post includendo le caratteristiche uniche del mio prodotto e hashtags rilevanti.

ChatGPT
🚀 **Lancio Ufficiale di AirStep!** 🏃‍♀️ 👟

Siamo entusiasti di presentarvi AirStep, la nostra nuova linea di calzini da corsa che trasforma ogni passo in un'esperienza senza precedenti! 🌟

✨ **Caratteristiche Distintive:**

- Materiali Sostenibili: Cotone biologico e tessuti riciclati per un'impronta ecologica.
- Comfort Ergonomico: Design studiato per un supporto superiore durante ogni corsa.
- Stile & Sostenibilità: Look alla moda che riflette il tuo impegno per uno stile di vita sano.

Unisci l'eleganza al tuo passo e fai la differenza con AirStep! 👟 💚

🌍 **Sostenibilità che fa la differenza:** Partecipa al nostro programma di riciclo per un mondo più verde!

🛍️ **Disponibile ora sul nostro sito! Link in bio.**

#AirStep #CalziniDaCorsa #Sostenibilità #RunningStyle #Innovazione #ComfortEcoFriendly #LancioProdotti #NuovaCollezione #EleganzaInOgniPasso #EcoFashion #CorsaSostenibile #GreenLiving #RunInStyle #NuovoArrivo #BenessereEco #FashionTech

Unisciti a noi nel rivoluzionare la corsa con stile e sostenibilità! 🌈 🏃‍♀️ #AirStepLaunch

Figura 7.10 Post sui social media.

Infine, con il contributo speciale di DALL-E, otteniamo il risultato finale, rappresentato nella Figura 7.11.

Ovviamente, qui mancano molti elementi per un completo sviluppo del prodotto e *go-to-market*. Tuttavia, con il supporto di ChatGPT e il contributo speciale di DALL-E siamo riusciti a sviluppare un brainstorming su una nuova linea di prodotti e varianti, sui potenziali clienti, su slogan accattivanti e, infine, abbiamo generato un post su Instagram per annunciare il lancio di SprintSoles. Potete provare DALL-E tramite *Bing Image Creator*: https://www.bing.com/images/create?FORM=GDPGLP.

Figura 7.11 Post su Instagram interamente generato da ChatGPT e DALL-E 3.

Il test A/B per il confronto nel marketing

Un altro campo interessante in cui ChatGPT può assistere i professionisti del marketing è il test A/B.

Il test A/B, nel marketing, è un metodo per confrontare due diverse versioni di una campagna di marketing, pubblicità o sito web per determinare quale sia più performante. Nel test A/B, vengono create due varianti della stessa campagna o elemento, cambiando una sola variabile tra le due versioni. L'obiettivo è quello di vedere quale versione genera più clic, conversioni o altri risultati desiderati.

Un esempio di test A/B potrebbe essere il test di due versioni di una campagna e-mail, utilizzando diversi "oggetti" dell'e-mail, o il test di due versioni di una pagina di destinazione di un sito web, con diversi pulsanti di call-to-action. Misurando il tasso di risposta di ciascuna versione, i *marketer* possono determinare quale versione è più performante e prendere decisioni basate sui dati su quale versione utilizzare effettivamente.

Il test A/B consente ai *marketer* di ottimizzare le loro campagne ed elementi per massimizzare l'efficacia, portando a risultati migliori e un ritorno sull'investimento più elevato.

Dato che questo metodo prevede il processo di generare molte varianti dello stesso contenuto, la potenza generativa di ChatGPT può sicuramente assisterci.

Prendiamo in considerazione il seguente esempio. Sto promuovendo un nuovo prodotto che ho sviluppato: un nuovo, leggero e sottile imbrago da arrampicata per *speed climber*. Ho già fatto qualche ricerca di mercato e conosco il mio pubblico di nicchia. So anche che un ottimo canale di comunicazione per quel target è la pubblicazione su

un blog di arrampicata online, dato che la maggior parte dei membri delle palestre di arrampicata sono lettori appassionati di questo blog.

Il mio obiettivo è quello di creare un post per condividere il lancio di questo nuovo imbrago per speed climber, e voglio sottoporre a un test due diverse versioni in due gruppi. Il post che sto per pubblicare e che voglio sia l'oggetto del mio test A/B è rappresentato nella Figura 7.12.

Rivoluzione Verticale - Scopri il Nuovo Imbraco Ultraleggero per Speed Climbers!

Cari amanti dell'arrampicata,

Siamo entusiasti di presentarvi il nostro ultimo gioiello per gli appassionati di climbing: l'imbraco ultraleggero progettato appositamente per gli speed climbers! Abbiamo lavorato instancabilmente per sviluppare un prodotto che sia all'altezza delle sfide che solo chi pratica speed climbing può comprendere appieno.

Non perdere l'opportunità di essere tra i primi a provare questo incredibile imbraco.

Affronta le tue sfide verticali con il massimo comfort, velocità e stile. Il futuro dell'arrampicata è qui, e non vediamo l'ora di condividere questa avventura con voi.

Figura 7.12 Un esempio di post di blog per lanciare attrezzature da arrampicata.

Qui, ChatGPT può aiutarci su due livelli.

- Il primo livello riguarda la *riformulazione dell'articolo*, utilizzando diverse parole chiave o diversi slogan accattivanti. Per farlo, una volta fornito questo post come contesto, possiamo chiedere a ChatGPT di lavorare sull'articolo e di modificare leggermente alcuni elementi (Figura 7.13).
  Come da mia richiesta, ChatGPT è stato in grado di rigenerare solo quegli elementi che ho chiesto (titolo, sottotitolo e frase di chiusura), in modo da poter monitorare l'efficacia di quegli elementi in base alla reazione dei due gruppi di pubblico.

You

Sto facendo un test A/B per il lancio di un nuovo imbraco. Sto usando un blog di arrampicata come canale di comunicazione.
Questo è il post:
###
Rivoluzione Verticale - Scopri il Nuovo Imbraco Ultraleggero per Speed Climbers!
Cari amanti dell'arrampicata,
Siamo entusiasti di presentarvi il nostro ultimo gioiello per gli appassionati di climbing: l'imbraco ultraleggero progettato appositamente per gli speed climbers! Abbiamo lavorato instancabilmente per sviluppare un prodotto che sia all'altezza delle sfide che solo chi pratica speed climbing può comprendere appieno.
Non perdere l'opportunità di essere tra i primi a provare questo incredibile imbraco.

Affronta le tue sfide verticali con il massimo comfort, velocità e stile. Il futuro dell'arrampicata è qui, e non vediamo l'ora di condividere questa avventura con voi.

###
Rigeneralo mantenendolo uguale tranne che per questi elementi: titolo, sottotitolo e frase conclusiva.

ChatGPT

Ecco la versione modificata del tuo post:

Evoluzione Verticale - Esplora il Nuovo Imbraco Ultraleggero per Speed Climbers!
Cari amanti dell'arrampicata,
Siamo entusiasti di presentarvi il nostro ultimo capolavoro per gli appassionati di climbing: l'imbraco ultraleggero progettato appositamente per gli speed climbers! Abbiamo lavorato instancabilmente per sviluppare un prodotto che sia all'altezza delle sfide che solo chi pratica speed climbing può comprendere appieno.
Non farti sfuggire l'opportunità di essere tra i primi a sperimentare questo incredibile imbraco.

Affronta le tue sfide verticali con il massimo comfort, velocità e stile. Il futuro dell'arrampicata è qui, e non vediamo l'ora di condividere questa straordinaria avventura con voi.

Figura 7.13 Nuova versione del post del blog.

- Il secondo livello riguarda il lavoro sul *design della pagina web*, cambiando la collocazione dell'immagine e la posizione dei pulsanti. A questo scopo, ho creato (Figura 7.14) una semplice pagina web per il post pubblicato nel blog di arrampicata.

Figura 7.14 Esempio di post pubblicato sul blog di arrampicata.

Possiamo alimentare direttamente ChatGPT con il codice HTML e chiedergli di cambiare alcuni elementi del layout, come la posizione dei pulsanti o la loro dicitura. Per esempio, piuttosto che *Compra ora*, un lettore potrebbe essere più attratto da un pulsante *Lo voglio!*.

Nella Figura 7.15, alimentiamo ChatGPT con il codice sorgente

HTML.
Nella Figura 7.16 vediamo l’output prodotto. Come potete vedere, ChatGPT è intervenuto solo a livello di pulsanti, cambiando leggermente il loro layout, posizione, colore e dicitura.
Infatti, ispezionando il codice sorgente delle due versioni delle pagine web, possiamo vedere (Figura 7.17) come differisca solo nelle sezioni dei pulsanti.

You

Sto lanciando un nuovo prodotto con un blog post su una pagina web. Di seguito la parte di codice dove ci sono i pulsanti:

```
###
<!-- Pulsante Acquista -->
<a href="http://example.com/buy-now" style="background-color:#04AA6D;color:white;padding:12px 20px;border:none;border-radius:4px;cursor:pointer;text-align:center;text-decoration:none;display:inline-block;font-size:16px;font-weight:bold;">Compra Ora </a>

<!-- Pulsante Condividi -->
<a href="https://www.facebook.com/sharer/sharer.php?u=http://example.com/your-article" target="_blank" style="background-color:#3B5998;color:white;padding:12px 20px;border:none;border-radius:4px;cursor:pointer;text-align:center;text-decoration:none;display:inline-block;font-size:16px;font-weight:bold;margin-left:10px;">Condividi su Facebook </a>

<!-- Pulsante Commento -->
<a href="http://example.com/feedback" style="background-color:#04AA6D;color:white;padding:12px 20px;border:none;border-radius:4px;cursor:pointer;text-align:center;text-decoration:none;display:inline-block;font-size:16px;font-weight:bold;margin-left:10px;">Lascia un commento </a>

###
```

Rigenera questo codice cambiando il layout, colore, posizione e testo.

ChatGPT

Ecco una versione modificata del codice con un layout, colori, posizioni e testo differenti:

html Copy code

```
<!-- Acquista Ora -->
<a href="http://example.com/buy-now" style="background-color:#2E86C1;co

<!-- Condividi su Twitter -->
<a href="https://twitter.com/intent/tweet?url=http://example.com/your-a

<!-- Lascia un Commento -->
<a href="http://example.com/feedback" style="background-color:#F39C12;c
```

Questa versione utilizza colori diversi, ha un padding leggermente maggiore, bordi arrotondati più accentuati, testo più grande e ha modificato la posizione dei pulsanti per renderli più distinti. Sentiti libero di personalizzare ulteriormente secondo le tue preferenze.

Figura 7.15 ChatGPT modifica il codice HTML.

Figura 7.16 Nuova versione del sito web.

Figura 7.17 Confronto tra il codice sorgente delle due versioni del sito web.

In conclusione, ChatGPT è uno strumento prezioso per il test A/B nel marketing. La sua capacità di generare rapidamente diverse versioni dello stesso contenuto può ridurre il tempo di commercializzazione di nuove campagne. Utilizzando ChatGPT per i test A/B, è possibile ottimizzare le strategie di marketing e, in definitiva, ottenere risultati migliori per la nostra azienda.

Potenziare l’ottimizzazione dei motori di ricerca (SEO)

Un altro settore promettente che ChatGPT può rivoluzionare è l’ottimizzazione dei motori di ricerca. Questo è l’elemento chiave dietro il posizionamento nei motori di ricerca come Google o Bing e determina se i vostri siti web risulteranno visibili a coloro che cercano quello che promuovete.

DEFINIZIONE

La SEO è una tecnica utilizzata per migliorare la visibilità e il posizionamento di un sito web nelle pagine dei risultati dei motori di ricerca (SERP). Questo viene ottenuto ottimizzando il sito web o la pagina web per aumentare la quantità e la qualità del traffico organico (non a pagamento) dai motori di ricerca. Lo scopo della SEO è quello di attirare sul sito web più visitatori mirati, ottimizzandolo per specifiche parole chiave o frasi.

Immaginiamo di gestire un'azienda di e-commerce chiamata *Hat&Gloves*, specializzata nella vendita di cappelli e guanti. Ora stiamo creando il sito web di e-commerce e vogliamo ottimizzare il suo posizionamento. Chiediamo a ChatGPT di elencare alcune parole chiave rilevanti (Figura 7.18) da incorporare nel nostro sito web.

Come potete vedere, ChatGPT è stato in grado di creare un elenco di parole chiave di diversi tipi. Alcune di esse sono piuttosto intuitive, come *Cappelli* e *Guanti*. Altre sono correlate con un legame indiretto. Per esempio, *Idee regalo* non è necessariamente correlato alla nostra attività di e-commerce, tuttavia, potrebbe essere molto intelligente includerla nelle parole chiave, in modo da poter ampliare il pubblico.

You

Sto costruendo un sito web per il mio e-commerce "Hat&Gloves", che vende solamente 2 prodotti: cappelli e guanti. Voglio ottimizzare il ranking del mio sito. Crea una lista di 10 parole che dovrei incorporare nel mio sito per avere risultati rilevanti in ottica di SEO.

ChatGPT

Certamente, ecco una lista di 10 parole chiave pertinenti che potresti incorporare nel tuo sito per ottimizzare il ranking in termini di SEO per il tuo e-commerce "Hat&Gloves":

1. Cappelli invernali
2. Guanti da freddo
3. Accessori moda invernale
4. Negozio cappelli online
5. Vendita guanti online
6. Collezione cappelli alla moda
7. Migliori guanti termici
8. Acquista cappelli e guanti
9. Stile invernale
10. Accessori moda freddo

Ricorda di utilizzare queste parole chiave in modo naturale all'interno dei contenuti del tuo sito, come descrizioni di prodotti, titoli, testi delle pagine e nei tag meta. Inoltre, assicurati che il contenuto sia di qualità e rilevante per i visitatori, poiché la SEO si basa anche sull'esperienza dell'utente.

Figura 7.18 Esempio di parole chiave SEO.

Un altro elemento chiave della SEO è l'intento. L'*intento del motore di ricerca*, noto anche come *intento dell'utente*, si riferisce allo scopo o obiettivo sottostante di una specifica query di ricerca effettuata da un utente in un motore di ricerca. Capire l'intento del motore di ricerca è importante, perché aiuta le aziende e i marketer a creare contenuti e strategie di marketing più mirati ed efficaci, che si allineano con le esigenze e le aspettative di chi esegue le ricerche.

Esistono generalmente quattro tipi di intento del motore di ricerca.

- *Intento informativo*: l'utente cerca informazioni su un determinato argomento o domanda, come "Qual è la capitale della Francia?" o "Come fare una pizza a casa?".
- *Intento di navigazione*: l'utente cerca un sito web o una pagina web ben determinati, come il login a Facebook o ad Amazon.com.
- *Intento commerciale*: l'utente vuole acquistare un prodotto o un servizio, ma potrebbe non aver ancora preso una decisione finale. Esempi di ricerche con intento commerciale includono "miglior laptop sotto i 1000$" o "scarpe scontate online".
- *Intento transazionale*: l'utente ha un obiettivo specifico per completare una transazione, che potrebbe riferirsi ad acquisti fisici o a sottoscrizioni a servizi. Esempi di intento transazionale potrebbero essere "acquista iPhone 13" o "iscriviti a una palestra".

Comprendendo l'intento dietro specifiche query di ricerca, le aziende e i marketer possono creare contenuti più mirati ed efficaci, che rispondono alle esigenze e alle aspettative del loro pubblico target. Questo può portare a posizionamenti più alti nei motori di ricerca, più traffico e, in definitiva, più conversioni e ricavi.

Ora, la domanda è: ChatGPT sarà in grado di determinare l'intento di una determinata richiesta?

Prima di rispondere, vale la pena di notare che l'attività di inferire l'intento di un determinato prompt è il core business dei modelli LLM (*Large Language Model*), incluso GPT. Quindi, di sicuro, ChatGPT è in grado di catturare gli intenti dei prompt.

Il valore aggiunto, qui, è che vogliamo vedere se ChatGPT è in grado di determinare l'intento in un dominio preciso e con una tassonomia precisa, cioè quella del marketing. Questo è il motivo per cui la progettazione del prompt è ancora una volta fondamentale per

guidare ChatGPT nella giusta direzione. Con le parole chiave rappresentate nella Figura 7.19, possiamo ottimizzare il nostro sito web e massimizzare il traffico degli utenti su di esso.

ChatGPT

Certamente, posso organizzare le parole chiave in base all'intento di ricerca:

Intento Commerciale:

1. Negozio cappelli online
2. Vendita guanti online
3. Acquista cappelli e guanti

Intento Transazionale:

4. Cappelli invernali

1. Guanti da freddo
2. Migliori guanti termici
3. Collezione cappelli alla moda
4. Stile invernale

Intento di Navigazione:

9. Accessori moda invernale

1. Accessori moda freddo

Intento Informativo:

Nessuna delle parole chiave sembra essere specificamente di intento informativo. Tuttavia, potresti creare contenuti informativi correlati ai materiali, alle tendenze invernali, consigli di abbinamento, ecc.

Questo raggruppamento ti aiuterà a creare una strategia di contenuti mirata a soddisfare diverse esigenze degli utenti in base all'intento di ricerca. Assicurati di implementare queste parole chiave in modo coerente e naturale all'interno del tuo sito web.

Figura 7.19 Esempio di parole chiave raggruppate per intento dell'utente.

Sentiment analysis per migliorare la qualità e aumentare la soddisfazione del cliente

La *sentiment analysis* è una tecnica utilizzata nel marketing per analizzare e interpretare le emozioni e le opinioni espresse dai clienti verso un marchio, un prodotto o un servizio. Coinvolge l'uso dell'elaborazione del linguaggio naturale e degli algoritmi di machine learning per identificare e classificare il sentiment dei dati testuali, come i post sui social media, le recensioni dei clienti e i sondaggi di feedback.

Eseguendo la sentiment analysis, i marketer possono ottenere intuizioni sulle percezioni dei clienti del loro marchio, identificare aree di miglioramento e prendere decisioni basate sui dati per ottimizzare le loro strategie di marketing. Per esempio, possono monitorare il sentiment delle recensioni dei clienti, per identificare quali prodotti o servizi stanno ricevendo feedback positivi o negativi e adeguare il messaggio di marketing di conseguenza.

Nel complesso, la sentiment analysis è uno strumento prezioso per i marketer per comprendere il sentiment dei clienti, valutare la soddisfazione del cliente e sviluppare campagne di marketing efficaci che risuonano con il loro pubblico target.

La sentiment analysis esiste da un po' di tempo, quindi potreste chiedervi quale valore aggiunto possa portare ChatGPT. Bene, oltre all'accuratezza dell'analisi, ChatGPT si distingue dagli altri strumenti di sentiment analysis grazie alla sua capacità, tipica dei modelli LLM, di generalizzare.

Ciò significa che, quando utilizziamo ChatGPT per la sentiment analysis, non stiamo sfruttando una delle sue API specifiche per quel compito: l'idea su cui si basano ChatGPT e i modelli OpenAI è che

possono assistere l’utente in molte attività generali contemporaneamente, interagendo con un’attività e cambiando l’ambito dell’analisi in base alla richiesta dell’utente.

Quindi, sicuramente ChatGPT è in grado di catturare il sentiment di un determinato testo, come un post su Twitter o una recensione di un prodotto. Tuttavia, può anche andare oltre e aiutare a identificare aspetti specifici di un prodotto o marchio che stanno influenzando positivamente o negativamente il sentiment. Per esempio, se i clienti menzionano costantemente in modo negativo una determinata caratteristica di un prodotto, ChatGPT può evidenziare quella caratteristica come un’area di miglioramento. Oppure, potreste chiedere a ChatGPT di generare una risposta a una recensione particolarmente delicata, tenendo conto del sentiment della recensione e utilizzandolo come contesto per la risposta. Ancora, ChatGPT può generare report che riassumono tutti gli elementi negativi e positivi trovati nelle recensioni o nei commenti, raggruppati per categoria.

Prendiamo in considerazione il seguente esempio. Un cliente ha recentemente acquistato un paio di scarpe dalla nostra azienda di e-commerce, *RunFast*, e ha lasciato la seguente recensione:

> “Recentemente ho acquistato le scarpe RunFast Prodigy e ho sentimenti contrastanti su di loro. Da un lato, le scarpe sono incredibilmente comode e offrono un grande supporto ai piedi durante le mie corse quotidiane. L’ammortizzazione è di alto livello e sento le caviglie meno affaticate rispetto ad altre scarpe che ho provato in passato. Inoltre, il design è visivamente accattivante e ho ricevuto diversi complimenti su di esse.
>
> Tuttavia, ho riscontrato alcuni problemi di durata con queste scarpe. La suola sembra consumarsi piuttosto rapidamente e il materiale, pur essendo traspirante, mostra segni di usura dopo solo poche settimane di utilizzo. Questo è deludente, considerando l’alto prezzo delle scarpe.
>
> Nel complesso, mentre amo il comfort e il design delle scarpe RunFast Prodigy, esito a raccomandarle a causa dei problemi di durata che ho

riscontrato.”

Nella Figura 7.20 chiediamo a ChatGPT di catturare il sentiment di questa recensione. Dalla figura, potete vedere come ChatGPT non si sia limitato a fornire un’etichetta: ha anche spiegato gli elementi positivi e negativi che caratterizzano la recensione, che ha un sentiment misto e quindi può essere etichettata come neutra, nel complesso.

You

Analizza il sentiment di questa recensione:
###
"Ho recentemente acquistato le scarpe RunFast Prodigy e ho sentimenti contrastanti su di loro. Da un lato, le scarpe sono incredibilmente comode e offrono un grande supporto ai piedi durante le mie corse quotidiane. L'ammortizzazione è di alto livello e sento le caviglie meno affaticate rispetto ad altre scarpe che ho provato in passato. Inoltre, il design è visivamente accattivante e ho ricevuto diversi complimenti su di loro. Tuttavia, ho riscontrato alcuni problemi di durata con queste scarpe. La suola sembra consumarsi piuttosto rapidamente e il materiale, pur essendo traspirante, mostra segni di usura dopo solo poche settimane di utilizzo. Questo è deludente, considerando l'alto prezzo delle scarpe.
Nel complesso, mentre amo il comfort e il design delle scarpe RunFast Prodigy, esito a raccomandarle a causa dei problemi di durata che ho riscontrato."
###

Come lo classificheresti?

ChatGPT

Questa recensione esprime sentimenti contrastanti nei confronti delle scarpe RunFast Prodigy. Da un lato, l'utente elogia il comfort, il supporto durante le corse quotidiane, l'ammortizzazione di alto livello e il design accattivante che ha ricevuto complimenti. D'altro canto, sottolinea problemi di durata come l'usura rapida della suola e del materiale, considerando deludente tale situazione, soprattutto in relazione all'alto prezzo delle scarpe.

Globalmente, la recensione sembra essere equilibrata, poiché menziona sia gli aspetti positivi che quelli negativi delle scarpe. Il sentiment complessivo potrebbe essere considerato neutro o leggermente incline verso un tono negativo a causa dei problemi di durata evidenziati.

Figura 7.20 ChatGPT analizza la recensione di un cliente.

Proviamo ad approfondire e a chiedere a ChatGPT alcuni suggerimenti su come migliorare il prodotto (Figura 7.21).

You
Basandoti su questa recensione, come potrei migliorare le scarpe RunFast Prodigy?

ChatGPT
Basandoci sulla recensione fornita, ecco alcune possibili aree di miglioramento per le scarpe RunFast Prodigy:

1. **Durabilità della suola e del materiale:** Affrontare il problema segnalato di usura rapida della suola e del materiale. Potrebbe essere necessario esaminare e migliorare la qualità dei materiali utilizzati per garantire una maggiore resistenza nel tempo.
2. **Rapporto qualità-prezzo:** Considerando il prezzo elevato delle scarpe, potrebbe essere utile valutare se è possibile migliorare la durata senza compromettere il comfort o il design. Offrire un prodotto che giustifichi appieno il costo percepito.
3. **Test di durata:** Implementare test di durata approfonditi durante lo sviluppo del prodotto per identificare e risolvere eventuali problemi precocemente. Ciò potrebbe contribuire a garantire una maggiore longevità delle scarpe.
4. **Comunicazione trasparente:** Se la durata delle scarpe è un punto critico, è importante comunicare in modo trasparente con i clienti, indicando le aspettative sulla durabilità del prodotto e fornendo consigli su come massimizzare la sua vita utile.
5. **Possibili opzioni di garanzia:** Considerare l'implementazione di politiche di garanzia che coprano specificamente problemi di durata. Questo può aumentare la fiducia del cliente nell'investire in un prodotto di alta qualità.
6. **Feedback degli utenti:** Monitorare costantemente il feedback degli utenti e apportare miglioramenti iterativi in base alle esperienze degli acquirenti. Questo può contribuire a mantenere elevati standard di qualità nel tempo.

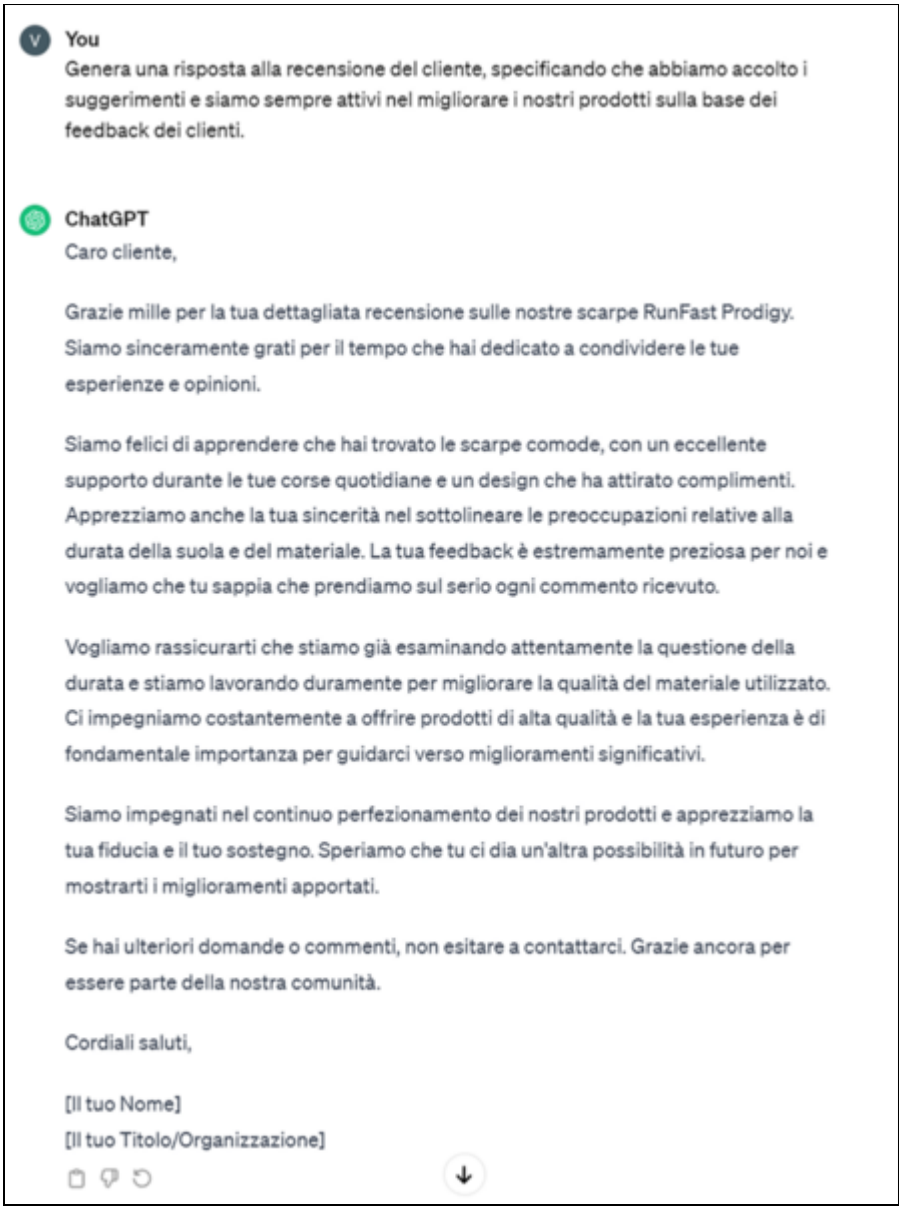
Concentrandosi su questi aspetti, l'obiettivo sarebbe quello di mantenere o migliorare i punti di forza attuali delle scarpe (comfort, supporto, ammortizzazione, design accattivante) mentre si affrontano le preoccupazioni sulla durata evidenziate nella recensione.

Figura 7.21 Suggerimenti su come migliorare il mio prodotto in base al feedback dei clienti.

Infine, generiamo una risposta al cliente (Figura 7.22), mostrando che alla nostra azienda sta molto a cuore il feedback dei clienti, e siamo sempre alla ricerca di come migliorare i nostri prodotti.

L'esempio che abbiamo visto era molto semplice, con una sola recensione. Ora, immaginate di avere tonnellate di recensioni, e diversi

canali di vendita da cui riceviamo feedback. Immaginate la potenza di strumenti come ChatGPT e i modelli OpenAI, in grado di analizzare e integrare tutte queste informazioni e di identificare i pro e i contro dei vostri prodotti, oltre a catturare le tendenze dei clienti e le abitudini di acquisto. Inoltre, per la cura e *customer retention*, potremmo anche automatizzare le risposte alle recensioni utilizzando lo stile di scrittura che preferiamo. Infatti, adattando il linguaggio e il tono del vostro chatbot alle specifiche esigenze e aspettative dei vostri clienti, potete creare un’esperienza cliente più coinvolgente ed efficace.



You
Genera una risposta alla recensione del cliente, specificando che abbiamo accolto i suggerimenti e siamo sempre attivi nel migliorare i nostri prodotti sulla base dei feedback dei clienti.

ChatGPT
Caro cliente,

Grazie mille per la tua dettagliata recensione sulle nostre scarpe RunFast Prodigy. Siamo sinceramente grati per il tempo che hai dedicato a condividere le tue esperienze e opinioni.

Siamo felici di apprendere che hai trovato le scarpe comode, con un eccellente supporto durante le tue corse quotidiane e un design che ha attirato complimenti. Apprezziamo anche la tua sincerità nel sottolineare le preoccupazioni relative alla durata della suola e del materiale. La tua feedback è estremamente preziosa per noi e vogliamo che tu sappia che prendiamo sul serio ogni commento ricevuto.

Vogliamo rassicurarti che stiamo già esaminando attentamente la questione della durata e stiamo lavorando duramente per migliorare la qualità del materiale utilizzato. Ci impegniamo costantemente a offrire prodotti di alta qualità e la tua esperienza è di fondamentale importanza per guidarci verso miglioramenti significativi.

Siamo impegnati nel continuo perfezionamento dei nostri prodotti e apprezziamo la tua fiducia e il tuo sostegno. Speriamo che tu ci dia un'altra possibilità in futuro per mostrarti i miglioramenti apportati.

Se hai ulteriori domande o commenti, non esitare a contattarci. Grazie ancora per essere parte della nostra comunità.

Cordiali saluti,

[Il tuo Nome]
[Il tuo Titolo/Organizzazione]

Figura 7.22 Risposta generata da ChatGPT.

Ecco alcuni esempi.

- *Chatbot empatico*: utilizza un tono e un linguaggio empatici per interagire con i clienti che potrebbero avere un problema o hanno bisogno di aiuto con una questione delicata.
- *Chatbot professionale*: utilizza un tono e un linguaggio professionali per interagire con i clienti che potrebbero cercare

informazioni specifiche o hanno bisogno di aiuto con un problema tecnico.

- *Chatbot conversazionale*: utilizza un tono informale e amichevole per interagire con i clienti che potrebbero cercare un'esperienza personalizzata o hanno una richiesta più generale.
- *Chatbot umoristico:* utilizza battute e un linguaggio spiritoso per interagire con i clienti che potrebbero cercare un'esperienza leggera o per allentare una situazione tesa.
- *Chatbot educativo:* utilizza uno stile di comunicazione didattico per interagire con i clienti che potrebbero cercare di saperne di più su un prodotto o un servizio.

In conclusione, ChatGPT può essere uno strumento potente per le aziende per condurre la sentiment analysis, migliorare la qualità e trattenere i loro clienti. Con le sue avanzate capacità di elaborazione del linguaggio naturale, ChatGPT può analizzare con precisione e in tempo reale il feedback e le recensioni dei clienti, fornendo alle aziende preziose intuizioni sul sentiment e sulle preferenze dei clienti. Utilizzando ChatGPT come parte della strategia di esperienza del cliente, le aziende possono identificare rapidamente eventuali problemi che potrebbero influire negativamente sulla *customer satisfaction* e intraprendere azioni correttive. Questo può aiutare le aziende non solo a migliorare la loro qualità, ma anche ad aumentare la fedeltà e la ritenzione dei clienti.

Riepilogo

In questo capitolo, abbiamo esplorato i modi in cui ChatGPT può essere utilizzato dai marketer per migliorare le strategie di marketing. Abbiamo appreso che ChatGPT può aiutare nello sviluppo di nuovi prodotti così come nella definizione della loro strategia di go-to-

market, nel progettare test A/B, nell’ottimizzare l’analisi SEO e nel catturare il sentiment di recensioni, post sui social media e altri feedback dei clienti.

L’importanza di ChatGPT per i marketer risiede nel suo potenziale di rivoluzionare il modo in cui le aziende interagiscono con i clienti. Sfruttando la potenza dell’elaborazione del linguaggio naturale, del machine learning e dei big data, ChatGPT permette alle aziende di creare messaggi di marketing più personalizzati e pertinenti, di migliorare il supporto al cliente e la customer satisfaction e, in definitiva, di guidare le vendite e aumentare i ricavi.

Man mano che ChatGPT continua ad avanzare ed evolversi è probabile che vedremo ancora più coinvolgimento nell’industria del marketing, specialmente nel modo in cui le aziende interagiscono con i loro clienti. Infatti, fare affidamento pesantemente sull’AI permette alle aziende di ottenere intuizioni più profonde sul comportamento e le preferenze dei clienti.

Il messaggio fondamentale per i marketer è quello di accogliere questi cambiamenti e adattarsi alla nuova realtà del marketing potenziato dall’AI, per stare avanti rispetto alla concorrenza e soddisfare le esigenze dei clienti.

Nel prossimo capitolo, esamineremo il terzo e ultimo dominio nell’applicazione di ChatGPT trattato in questo libro: la ricerca.

Reinventare la ricerca con ChatGPT

In questo capitolo, ci concentreremo sulle applicazioni di ChatGPT nell'ambito della ricerca. Il capitolo affronterà alcuni dei principali casi d'uso che ChatGPT può risolvere, in modo da mostrare, attraverso esempi concreti, come ChatGPT possa essere utilizzato in questo campo.

Alla fine di questo capitolo, avrete dimestichezza di come utilizzare ChatGPT come "assistente" di ricerca. Come sempre, il capitolo fornirà i prompt utilizzati negli esempi, in modo che possiate provarli in autonomia.

I vantaggi di ChatGPT per i ricercatori

ChatGPT può essere una risorsa incredibilmente preziosa per i ricercatori in una vasta gamma di settori. Come modello linguistico sofisticato addestrato su enormi quantità di dati, ChatGPT può elaborare velocemente e accuratamente le informazioni e generare intuizioni che potrebbero essere difficili da scoprire o richiedere molto tempo impiegando solo metodi di ricerca tradizionali.

Inoltre, ChatGPT può fornire ai ricercatori una prospettiva unica sul loro campo, analizzando modelli e tendenze che potrebbero non essere

immediatamente evidenti.

Per esempio, immaginiamo un ricercatore che studi il cambiamento climatico e desideri comprendere la percezione pubblica di questo problema. Potrebbe chiedere a ChatGPT di analizzare i dati dei social media relativi al cambiamento climatico e identificare i temi e i sentiment più comuni espressi dagli utenti online. ChatGPT potrebbe quindi fornire al ricercatore un rapporto completo che dettaglia le parole, le frasi e le emozioni più comuni associate a questo argomento, così come eventuali tendenze o modelli emergenti che potrebbero essere utile esplorare.

Nei prossimi paragrafi, approfondiamo quattro casi d'uso in cui ChatGPT può aumentare la produttività della ricerca.

Generazione di idee per uno studio di ricerca

La produzione di uno studio di ricerca è un'attività che coinvolge diversi step, a partire dalla definizione della domanda di ricerca.

DEFINIZIONE

Una *domanda di ricerca* in ambito accademico è una dichiarazione chiara e specifica che identifica il problema o l'argomento che lo studio intende affrontare, fornendo una guida per la progettazione e l'esecuzione della ricerca.

Data la domanda, ci sono poi ulteriori attività in cui ChatGPT può essere un valido assistente di ricerca. Vediamole nel dettaglio.

- *Definizione della domanda di ricerca*: il primo step è quello di definire la domanda di ricerca. Per esempio, se fossimo interessati a produrre un paper sugli effetti dei social media sulla salute mentale, potremmo chiedere a ChatGPT qualche suggerimento (Figura 8.1) su possibili domande di ricerca. Queste sono tutte

domande interessanti che potrebbero essere ulteriormente indagate. In questo esempio, terremo in considerazione l'ultima domanda suggerita.

You
Voglio portare avanti una ricerca sugli effetti dei social media sulla salute mentale. Generami 5 potenziali domande di ricerca su cui scrivere il mio paper:

ChatGPT
Certamente, ecco alcune potenziali domande di ricerca sull'effetto dei social media sulla salute mentale:

1. **In che modo l'uso intensivo dei social media influisce sulla salute mentale degli individui?**
   - Esplora la correlazione tra la frequenza di utilizzo dei social media e la presenza di disturbi mentali come ansia, depressione o isolamento sociale.
2. **Qual è l'impatto degli ideali di bellezza promossi sui social media sulla percezione del corpo e sull'autostima?**
   - Analizza come le immagini e gli standard di bellezza presenti online influenzano l'autostima e la percezione del corpo, specialmente tra i giovani utenti dei social media.
3. **Come le interazioni sociali online differiscono da quelle faccia a faccia nell'influenzare il benessere psicologico?**
   - Esamina le differenze tra le relazioni online e offline e il loro impatto sulla salute mentale, considerando elementi come il supporto sociale, la solitudine e la qualità delle connessioni.
4. **In che modo la comparazione sociale sui social media contribuisce allo sviluppo di ansia e depressione?**
   - Investigare come la tendenza a confrontarsi con gli altri su piattaforme sociali può influenzare negativamente il benessere psicologico e contribuire a condizioni come ansia e depressione.
5. **Qual è il ruolo delle notizie false o fuorvianti sui social media nella salute mentale della società?**
   - Esplora come la diffusione di informazioni errate o sensazionali influisce sullo stato emotivo e cognitivo degli utenti, con particolare attenzione agli effetti sulla paura, l'insicurezza e la disinformazione.

Figura 8.1 Esempi di domande di ricerca basate su un dato argomento.

- *Ricerca della letteratura*: ora che abbiamo la nostra domanda di ricerca, il passo successivo consiste nell'eseguire ricerche nella

letteratura pertinente utilizzando varie banche dati, motori di ricerca e altre fonti.

> **NOTA**
> In questo esempio, useremo la versione gratuita di ChatGPT, che è potenziata dal modello GPT-3.5-turbo, che non supporta plug-in e connettori verso fonti dati esterne e motori di ricerca. Pertanto, possiamo affidarci al materiale su cui il modello GPT-3.5-turbo è stato addestrato, che potrebbe essere non esauriente e, soprattutto, non aggiornato, dato il "taglio della conoscenza" del 2022.

Come potete notare nella Figura 8.2, ChatGPT non si vuole "prendere il rischio" di condividere informazioni non corrette o aggiornate, e preferisce suggerirci fonti più validate da cui recuperare le nostre informazioni.
Partendo dai suggerimenti di ChatGPT, possiamo iniziare ad approfondire questi riferimenti.

You

Voglio proseguire con la domanda di ricerca "Come le interazioni sociali online differiscono da quelle faccia a faccia nell'influenzare il benessere psicologico?". Come posso cominciare a raccogliere letteratura al riguardo?

ChatGPT

Per iniziare a raccogliere letteratura sull'effetto delle interazioni sociali online rispetto a quelle faccia a faccia sul benessere psicologico, puoi seguire questi passaggi:

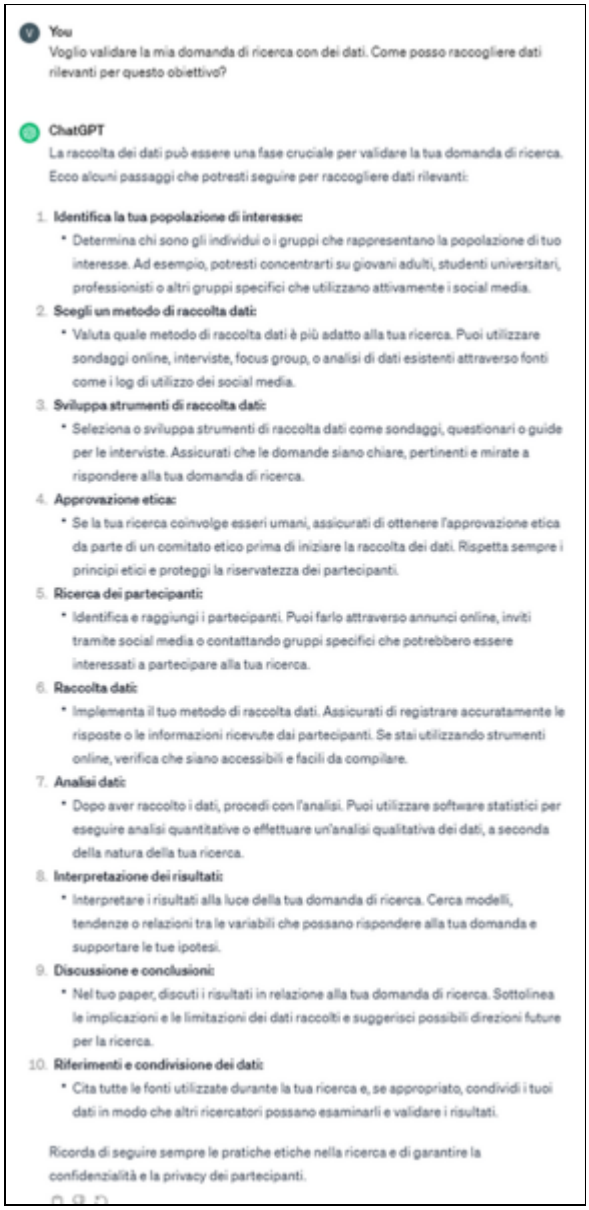
1. **Definisci i concetti chiave:**
   - Identifica i concetti chiave della tua ricerca, come "interazioni sociali online", "interazioni faccia a faccia" e "benessere psicologico". Questi concetti ti aiuteranno a creare stringhe di ricerca più efficaci.
2. **Utilizza basi di dati accademiche:**
   - Accesso a basi di dati accademiche come PubMed, PsycINFO, JSTOR, Google Scholar e altre risorse specializzate in psicologia e scienze sociali. Utilizza queste piattaforme per cercare articoli, recensioni e studi relativi al tuo argomento.
3. **Scegli parole chiave pertinenti:**
   - Utilizza parole chiave pertinenti al tuo argomento per ottenere risultati più specifici. Ad esempio, potresti includere termini come "social media", "face-to-face interaction", "psychological well-being", "online communication", e così via.
4. **Esplora la letteratura esistente:**
   - Leggi articoli e studi già esistenti sulla tematica per comprendere le scoperte e le metodologie utilizzate dagli altri ricercatori. Ciò ti aiuterà a identificare eventuali lacune nella ricerca esistente e a sviluppare una prospettiva più approfondita sul tuo argomento.
5. **Verifica le citazioni degli articoli rilevanti:**
   - Guarda alle citazioni degli articoli che trovi particolarmente rilevanti per il tuo argomento. Questo ti consentirà di esplorare ulteriori studi che potrebbero essere pertinenti alla tua ricerca.
6. **Esplora riviste specializzate:**
   - Cerca riviste specializzate in psicologia, scienze sociali o media studies. Le pubblicazioni in queste riviste sono spesso peer-reviewed e forniscono una base solida per la tua ricerca.
7. **Consulta libri e manuali:**
   - Esplora libri e manuali di testo che trattano l'argomento. Queste risorse possono offrire una panoramica più ampia e approfondita sulle teorie, i modelli e le ricerche esistenti.
8. **Utilizza sinonimi e termini correlati:**
   - Amplia la tua ricerca includendo sinonimi e termini correlati per ciascun concetto chiave. Ciò ti aiuterà a scoprire una gamma più ampia di studi pertinenti.
9. **Aggiorna la tua ricerca regolarmente:**
   - L'ambito della ricerca è in continua evoluzione, quindi assicurati di aggiornare regolarmente la tua ricerca per includere nuovi studi e informazioni.
10. **Organizza e annota la letteratura:**
    - Mantieni un registro organizzato degli articoli, degli abstract e delle citazioni che raccogli. Annota le principali scoperte, i metodi di studio e le eventuali limitazioni per facilitare la stesura del tuo paper.

Seguendo questi passaggi, sarai in grado di raccogliere una solida base di letteratura per supportare la tua ricerca sulle differenze tra le interazioni sociali online e faccia a faccia nell'influenzare il benessere psicologico.

Figura 8.2 Ricerca della letteratura con il supporto di ChatGPT.

- *Raccolta dati*: una volta identificata la letteratura pertinente e gli argomenti da trattare nella nostra ricerca, il passo successivo consiste nel raccogliere dati che possano convalidarla. Chiediamo a ChatGPT qualche consiglio su come fare.
  Nella Figura 8.3, ChatGPT è stato in grado di fornirci una

panoramica degli step da seguire per raccogliere dati rilevanti per la nostra domanda di ricerca.



You

Voglio validare la mia domanda di ricerca con dei dati. Come posso raccogliere dati rilevanti per questo obiettivo?

ChatGPT

La raccolta dei dati può essere una fase cruciale per validare la tua domanda di ricerca. Ecco alcuni passaggi che potresti seguire per raccogliere dati rilevanti:

1. **Identifica la tua popolazione di interesse:**
   - Determina chi sono gli individui o i gruppi che rappresentano la popolazione di tuo interesse. Ad esempio, potresti concentrarti su giovani adulti, studenti universitari, professionisti o altri gruppi specifici che utilizzano attivamente i social media.
2. **Scegli un metodo di raccolta dati:**
   - Valuta quale metodo di raccolta dati è più adatto alla tua ricerca. Puoi utilizzare sondaggi online, interviste, focus group, o analisi di dati esistenti attraverso fonti come i log di utilizzo dei social media.
3. **Sviluppa strumenti di raccolta dati:**
   - Seleziona o sviluppa strumenti di raccolta dati come sondaggi, questionari o guide per le interviste. Assicurati che le domande siano chiare, pertinenti e mirate a rispondere alla tua domanda di ricerca.
4. **Approvazione etica:**
   - Se la tua ricerca coinvolge esseri umani, assicurati di ottenere l'approvazione etica da parte di un comitato etico prima di iniziare la raccolta dei dati. Rispetta sempre i principi etici e proteggi la riservatezza dei partecipanti.
5. **Ricerca dei partecipanti:**
   - Identifica e raggiungi i partecipanti. Puoi farlo attraverso annunci online, inviti tramite social media o contattando gruppi specifici che potrebbero essere interessati a partecipare alla tua ricerca.
6. **Raccolta dati:**
   - Implementa il tuo metodo di raccolta dati. Assicurati di registrare accuratamente le risposte o le informazioni ricevute dai partecipanti. Se stai utilizzando strumenti online, verifica che siano accessibili e facili da compilare.
7. **Analisi dati:**
   - Dopo aver raccolto i dati, procedi con l'analisi. Puoi utilizzare software statistici per eseguire analisi quantitative o effettuare un'analisi qualitativa dei dati, a seconda della natura della tua ricerca.
8. **Interpretazione dei risultati:**
   - Interpretare i risultati alla luce della tua domanda di ricerca. Cerca modelli, tendenze o relazioni tra le variabili che possano rispondere alla tua domanda e supportare le tue ipotesi.
9. **Discussione e conclusioni:**
   - Nel tuo paper, discuti i risultati in relazione alla tua domanda di ricerca. Sottolinea le implicazioni e le limitazioni dei dati raccolti e suggerisci possibili direzioni future per la ricerca.
10. **Riferimenti e condivisione dei dati:**
    - Cita tutte le fonti utilizzate durante la tua ricerca e, se appropriato, condividi i tuoi dati in modo che altri ricercatori possano esaminarli e validare i risultati.

Ricorda di seguire sempre le pratiche etiche nella ricerca e di garantire la confidenzialità e la privacy dei partecipanti.

Figura 8.3 Generazione degli step da seguire per la raccolta dati in uno studio di ricerca.

- *Generazione di un template di ricerca*: l'ultimo passo nel processo di produzione del nostro paper di ricerca è quello di produrre un template che possa accogliere le varie sezioni, inclusi

i paragrafi relativi alla raccolta dati ed evidenze empiriche risultanti dai nostri esperimenti e survey. Vediamo come se la cava ChatGPT in questo ambito.

Anche in questo caso, vedi la Figura 8.4, ChatGPT è stato in grado di produrre un risultato utile allo scopo, impostando un template per la stesura del nostro paper di ricerca.

You

Puoi generarmi una template di riferimento per impostare il mio paper di ricerca? Sii conciso.

ChatGPT

Certamente, ecco una breve template per aiutarti a impostare il tuo paper di ricerca:

Titolo:

Effetti delle Interazioni Sociali Online sul Benessere Psicologico: Uno Studio Esplorativo

Introduzione:

* Contestualizzazione dell'argomento e dichiarazione della domanda di ricerca.
* Breve revisione della letteratura esistente.
* Dichiarazione dell'importanza della ricerca.

Obiettivo della Ricerca:

* Presentazione chiara della domanda di ricerca.
* Indicazione dei parametri chiave e delle variabili.

Metodologia:

* Descrizione del design della ricerca (es. studio trasversale, interviste, sondaggi).
* Selezione della popolazione di studio.
* Spiegazione degli strumenti di raccolta dati.

Raccolta Dati:

* Informazioni sulla partecipazione dei soggetti di studio.
* Dettagli sulla raccolta dati e sulla loro analisi.

Risultati:

* Presentazione chiara dei risultati, supportati da dati.
* Interpretazione dei risultati in relazione alla domanda di ricerca.

Discussione:

* Discussione delle implicazioni dei risultati.
* Confronto con la letteratura esistente.
* Limitazioni dello studio e possibili direzioni future.

Conclusioni:

* Riassunto dei risultati chiave.
* Conclusione riflettendo sull'obiettivo della ricerca.

Riferimenti:

* Citazioni complete di tutte le fonti utilizzate nel paper.

Ricorda di adattare la template alle specifiche esigenze del tuo studio e di includere dati e risultati pertinenti per sostenere le tue affermazioni.

Figura 8.4 Generazione di un template per un paper di ricerca.

Nel complesso, ChatGPT è stato in grado di svolgere molte attività nel campo della stesura di paper di ricerca, dalla generazione di idee per la domanda di ricerca alla generazione di un template per il nostro paper finale.

Un'ulteriore attività che può essere supportata da ChatGPT, fondamentale nell'ambito di ricerca, è la progettazione dell'esperimento che il ricercatore vuole condurre. Ne parleremo nel prossimo paragrafo.

Supporto per la progettazione di un esperimento

Il processo di pianificazione e realizzazione di un esperimento scientifico è un passaggio fondamentale per rispondere a una domanda di ricerca. Sono diversi gli elementi coinvolti, come la struttura dello studio, le variabili da misurare, la dimensione del campione e le procedure per raccogliere e analizzare i dati.

ChatGPT può aiutare nella progettazione di esperimenti per la ricerca suggerendo il tipo di studio (un trial controllato randomizzato, un disegno quasi-sperimentale o uno studio correlazionale), e supportando il ricercatore lungo l'intera implementazione.

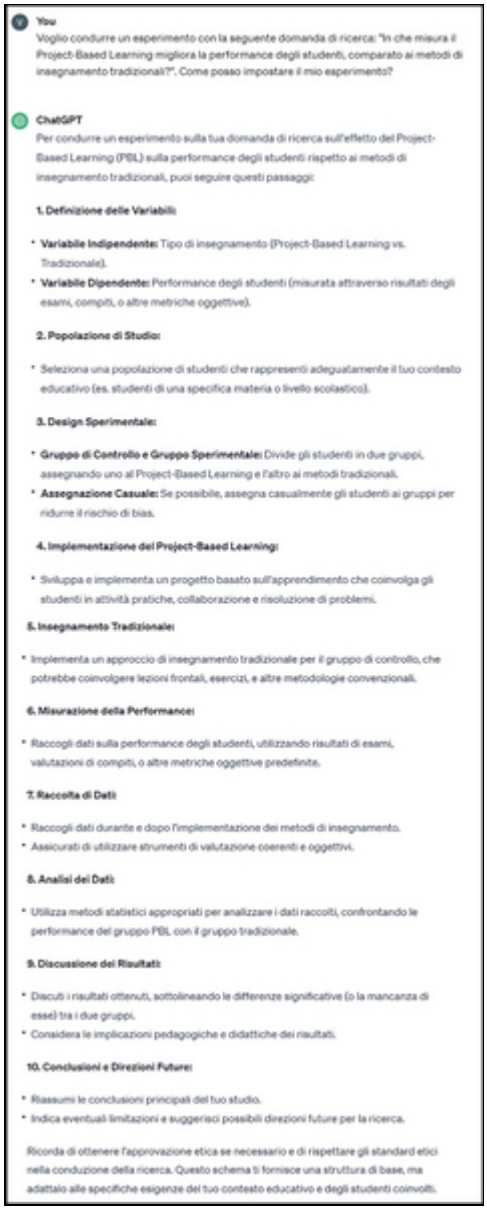
Prendiamo in considerazione il seguente scenario. Vogliamo indagare gli effetti di un nuovo programma educativo sui risultati di apprendimento degli studenti in matematica. Questo nuovo programma prevede un "apprendimento basato su progetti" (PBL), il che significa che agli studenti viene chiesto di lavorare in modo collaborativo su progetti del mondo reale, utilizzando concetti e competenze matematiche per risolvere problemi e creare soluzioni.

A tal fine, abbiamo definito la nostra domanda di ricerca come segue:

Come si confronta il nuovo programma PBL con i metodi di insegnamento tradizionali nel migliorare le prestazioni degli studenti?

Ecco come ChatGPT può aiutare.

- *Determinazione della struttura dello studio:* ChatGPT può assistere nel determinare la struttura appropriata per la domanda di ricerca, come un trial controllato randomizzato, un disegno quasi-sperimentale o uno studio correlazionale.
- ChatGPT (Figura 8.5) ha suggerito di procedere con un trial controllato randomizzato (RCT) e ha fornito una chiara spiegazione della ragione su cui si basa questa scelta. Mi sembra ragionevole procedere con questo approccio: i prossimi passaggi consisteranno nell'identificare le metriche di misura dell'esito e le variabili da considerare nel nostro esperimento.



You

Voglio condurre un esperimento con la seguente domanda di ricerca: "In che misura il Project-Based Learning migliora la performance degli studenti, comparato ai metodi di insegnamento tradizionali?". Come posso impostare il mio esperimento?

ChatGPT

Per condurre un esperimento sulla tua domanda di ricerca sull'effetto del Project-Based Learning (PBL) sulla performance degli studenti rispetto ai metodi di insegnamento tradizionali, puoi seguire questi passaggi:

1. Definizione delle Variabili:

- **Variabile Indipendente:** Tipo di insegnamento (Project-Based Learning vs. Tradizionale).
- **Variabile Dipendente:** Performance degli studenti (misurata attraverso risultati degli esami, compiti, o altre metriche oggettive).

2. Popolazione di Studio:

- Seleziona una popolazione di studenti che rappresenti adeguatamente il tuo contesto educativo (es. studenti di una specifica materia o livello scolastico).

3. Design Sperimentale:

- **Gruppo di Controllo e Gruppo Sperimentale:** Dividi gli studenti in due gruppi, assegnando uno al Project-Based Learning e l'altro ai metodi tradizionali.
- **Assegnazione Casuale:** Se possibile, assegna casualmente gli studenti ai gruppi per ridurre il rischio di bias.

4. Implementazione del Project-Based Learning:

- Sviluppa e implementa un progetto basato sull'apprendimento che coinvolga gli studenti in attività pratiche, collaborazione e risoluzione di problemi.

5. Insegnamento Tradizionale:

- Implementa un approccio di insegnamento tradizionale per il gruppo di controllo, che potrebbe coinvolgere lezioni frontali, esercizi, e altre metodologie convenzionali.

6. Misurazione della Performance:

- Raccogli dati sulla performance degli studenti, utilizzando risultati di esami, valutazioni di compiti, o altre metriche oggettive predefinite.

7. Raccolta di Dati:

- Raccogli dati durante e dopo l'implementazione dei metodi di insegnamento.
- Assicurati di utilizzare strumenti di valutazione coerenti e oggettivi.

8. Analisi dei Dati:

- Utilizza metodi statistici appropriati per analizzare i dati raccolti, confrontando le performance del gruppo PBL con il gruppo tradizionale.

9. Discussione dei Risultati:

- Discuti i risultati ottenuti, sottolineando le differenze significative (o la mancanza di esse) tra i due gruppi.
- Considera le implicazioni pedagogiche e didattiche dei risultati.

10. Conclusioni e Direzioni Future:

- Riassumi le conclusioni principali del tuo studio.
- Indica eventuali limitazioni e suggerisci possibili direzioni future per la ricerca.

Ricorda di ottenere l'approvazione etica se necessario e di rispettare gli standard etici nella conduzione della ricerca. Questo schema ti fornisce una struttura di base, ma adattalo alle specifiche esigenze del tuo contesto educativo e degli studenti coinvolti.

Figura 8.5 ChatGPT suggerisce il design di studio appropriato per il vostro esperimento.

- *Identificazione delle metriche di misura dell'esito*: ChatGPT può aiutarvi a identificare alcune possibili misure dell'esito per determinare i risultati del vostro test. Nella Figura 8.6 chiediamo alcuni suggerimenti per il nostro studio.

Mi sembra ragionevole scegliere i punteggi dei test per misurare l’esito.

You
Voglio comparare il nuovo programma PBL con i metodi di insegnamento tradizionali nel miglioramento delle performance degli studenti. Voglio usare un approccio di Randomized Control Trial. Quali possono essere possibili misurazioni dell'outcome?

ChatGPT
Se stai pianificando di utilizzare un approccio di Randomized Control Trial (RCT) per confrontare il nuovo programma di Project-Based Learning (PBL) con i metodi di insegnamento tradizionali nel miglioramento delle performance degli studenti, è importante selezionare misurazioni dell'outcome pertinenti e valide. Ecco alcune possibili misurazioni dell'outcome che potresti considerare:

1. **Risultati degli Esami Standardizzati:**
   - Misura le performance degli studenti attraverso esami standardizzati, che sono una metrica oggettiva e comparabile.
2. **Valutazioni degli Insegnanti:**
   - Chiedi agli insegnanti di valutare le performance degli studenti in base ai criteri predefiniti, considerando aspetti come la partecipazione, l'interesse, la creatività e l'applicazione delle conoscenze.
3. **Prove di Competenze Specifiche:**
   - Implementa prove che misurino specifiche competenze o abilità acquisite attraverso il programma PBL e i metodi tradizionali.
4. **Valutazioni Autentiche dei Progetti:**
   - Nel caso del PBL, valuta il successo degli studenti attraverso la valutazione di progetti specifici, evidenziando la loro capacità di applicare le conoscenze in contesti reali.
5. **Partecipazione Attiva in Classe:**
   - Misura il livello di partecipazione attiva degli studenti durante le lezioni PBL e tradizionali, considerando la quantità e la qualità delle interazioni.
6. **Autovalutazione degli Studenti:**
   - Chiedi agli studenti di autovalutare il proprio apprendimento, coinvolgendo domande specifiche sulla loro percezione di miglioramento, coinvolgimento e soddisfazione.
7. **Tassi di Frequenza e Ritiro:**
   - Monitora i tassi di frequenza alle lezioni e i tassi di ritiro per valutare l'impatto dei diversi approcci di insegnamento sulla partecipazione e la permanenza degli studenti nel corso.
8. **Risoluzione di Problemi:**
   - Utilizza attività che richiedono la risoluzione di problemi come parte delle misurazioni dell'outcome, riflettendo l'efficacia del programma nell'affinare le abilità critiche degli studenti.
9. **Feedback degli Studenti:**
   - Raccogli feedback diretti dagli studenti attraverso sondaggi o interviste, cercando di ottenere informazioni dettagliate sulla loro esperienza con il programma PBL e i metodi tradizionali.
10. **Successo Post-Scuola:**
    - Esamina il successo degli studenti dopo la scuola, come il loro rendimento accademico in corsi successivi o il successo nell'occupazione o nell'istruzione superiore.

Assicurati di scegliere misurazioni dell'outcome che siano pertinenti al tuo obiettivo di ricerca e che riflettano in modo accurato il miglioramento delle performance degli studenti. Utilizza metodi statistici appropriati per analizzare i dati raccolti e trarre conclusioni valide.

Figura 8.6 Risultati di apprendimento per lo studio di ricerca.

- *Identificazione delle variabili*: ChatGPT può aiutare il ricercatore a identificare le variabili indipendenti e dipendenti nello studio. Notate (Figura 8.7) che ChatGPT è stato anche in grado di generare il tipo di variabili, chiamate variabili di controllo, che

sono specifiche per il disegno di studio che stiamo considerando (RCT).

Le *variabili di controllo*, conosciute anche come *covarianti*, sono variabili che vengono mantenute costanti o controllate in uno studio di ricerca al fine di isolare la relazione tra le variabili indipendenti e la variabile dipendente. Queste variabili non sono il focus principale dello studio, ma sono incluse per minimizzare l'effetto delle variabili nel confondere i risultati. Controllando queste variabili, i ricercatori possono ridurre il rischio di ottenere risultati falsamente positivi o negativi e aumentare la validità interna del loro studio.

You

Voglio comparare il nuovo programma PBL con i metodi di insegnamento tradizionali nel miglioramento delle performance degli studenti. Voglio usare un approccio di Randomized Control Trial. Che variabili dovrei includere nel mio studio?

‹ 1 / 2 ›

ChatGPT

Per condurre un Randomized Control Trial (RCT) che confronti il nuovo programma di Project-Based Learning (PBL) con i metodi di insegnamento tradizionali, è importante includere diverse variabili che possano catturare l'effetto dei diversi approcci di insegnamento sul miglioramento delle performance degli studenti. Ecco alcune variabili chiave che potresti considerare:

1. **Variabile Indipendente:**
   - **Tipo di Insegnamento:** Dividi i partecipanti in due gruppi: uno che riceve l'insegnamento attraverso il nuovo programma PBL e l'altro attraverso i metodi tradizionali.
2. **Variabili Dipendenti:**
   - **Performance Accademica:** Misura i risultati degli esami, compiti o altre valutazioni standardizzate per valutare il rendimento accademico degli studenti.
   - **Competenze Specifiche:** Valuta le competenze specifiche acquisite attraverso il PBL e i metodi tradizionali, in base agli obiettivi specifici del corso o del programma.
3. **Variabili di Controllo:**
   - **Livello di Instradamento Iniziale:** Considera il livello di preparazione iniziale degli studenti, misurato attraverso valutazioni di ingresso o risultati pregressi.
   - **Caratteristiche Demografiche:** Include variabili come l'età, il sesso, la provenienza geografica e altre caratteristiche demografiche per controllare eventuali differenze iniziali tra i gruppi.
4. **Variabili Contestuali:**
   - **Ambiente di Apprendimento:** Valuta le caratteristiche dell'ambiente di apprendimento, come l'accesso alle risorse, la qualità delle strutture e l'atmosfera in classe.
   - **Coinvolgimento degli Insegnanti:** Misura il coinvolgimento e l'adattamento degli insegnanti al nuovo programma PBL o ai metodi tradizionali.
5. **Variabili di Processo:**
   - **Partecipazione Attiva:** Registra il livello di partecipazione attiva degli studenti durante le lezioni, progetti PBL o attività tradizionali.
   - **Frequenza e Durata delle Lezioni:** Monitora la frequenza e la durata delle lezioni per entrambi i gruppi.
6. **Misurazioni Temporali:**
   - **Risultati a Breve e Lungo Termine:** Valuta le performance degli studenti a breve termine (nel corso del programma) e a lungo termine (dopo la conclusione del programma o in corsi successivi).
7. **Variabili di Valutazione della Soddisfazione:**
   - **Feedback degli Studenti:** Raccogli il feedback degli studenti sulla loro soddisfazione, il coinvolgimento e la percezione dell'efficacia del programma.
8. **Variabili di Riuscita e Coinvolgimento:**
   - **Tassi di Ritiro e Frequenza:** Monitora i tassi di ritiro e la frequenza degli studenti nei due gruppi per valutare il coinvolgimento e la retention.
9. **Misurazioni Della Collaborazione:**
   - **Grado di Collaborazione:** Nel caso del PBL, misura il grado di collaborazione tra gli studenti nei progetti.
10. **Variabili Socioemotive:**
    - **Soddisfazione Emotiva:** Esamina la soddisfazione emotiva degli studenti rispetto al programma PBL o ai metodi tradizionali.

Figura 8.7 ChatGPT genera variabili per il dato studio.

Con le variabili precedenti, siamo pronti a impostare il nostro esperimento. Ora abbiamo bisogno di selezionare i partecipanti, e ChatGPT può aiutarci in questo.

- *Strategia di campionamento:* ChatGPT può suggerire potenziali strategie di campionamento per lo studio.

Notate che, in uno scenario del mondo reale, è sempre consigliabile chiedere agli strumenti di intelligenza artificiale di generare più opzioni con le spiegazioni di fondo, in modo da poter prendere una decisione ragionata. Per questo esempio (Figura 8.8), procediamo con quello che ChatGPT ci ha suggerito, che include anche suggerimenti sulla popolazione di interesse e sulla dimensione del campione.

You

Devo selezionare i partecipanti per il mio RCT sulla comparazione del PBL con altri metodi di insegnamento tradizionali. Che tipo di strategia di campionamento potrei utilizzare?

ChatGPT

La selezione dei partecipanti in un Randomized Control Trial (RCT) è fondamentale per garantire la rappresentatività e la validità dei risultati. Ecco alcune strategie di campionamento che potresti considerare:

1. **Campionamento Casuale Semplice:**
   - Assegna casualmente gli studenti al gruppo PBL o al gruppo tradizionale. Questa è la strategia di base che assicura l'equità nel processo di assegnazione.
2. **Campionamento Stratificato:**
   - Dividi la popolazione di studenti in sottogruppi omogenei (ad esempio, per livello di apprendimento iniziale o background educativo) e assegna casualmente gli studenti a ciascun gruppo PBL o tradizionale all'interno di ogni strato.
3. **Campionamento a Cluster:**
   - Raggruppa le classi o le scuole come cluster e assegna casualmente i cluster al gruppo PBL o tradizionale. Questo può essere utile se temi che l'aspetto di clusterizzazione nelle scuole possa influenzare i risultati.
4. **Campionamento Stratificato Proporzionale:**
   - Suddividi la popolazione in sottogruppi basati su caratteristiche chiave (ad esempio, livello di apprendimento iniziale) e assegna almeno una quota proporzionale di studenti di ciascun sottogruppo a ciascun gruppo di trattamento.
5. **Campionamento Sequenziale:**
   - Assegna i partecipanti in modo sequenziale a ciascun gruppo man mano che si presentano, fino al raggiungimento delle dimensioni del campione desiderate.
6. **Campionamento Casuale Stratificato per Età:**
   - Se il tuo studio coinvolge diversi livelli di età, puoi suddividere gli studenti in gruppi basati sull'età e assegnare casualmente gli studenti di ciascun gruppo a PBL o tradizionale.

Qualunque strategia di campionamento tu scelga, è importante assicurarti che sia applicata in modo imparziale e che i partecipanti siano assegnati ai gruppi in modo casuale. Ciò riduce il rischio di bias e garantisce che i gruppi siano comparabili prima dell'intervento. Inoltre, documenta attentamente il processo di campionamento e qualsiasi criterio di inclusione/esclusione adottato.

Figura 8.8 Suggerimento di strategia di campionamento RCT da ChatGPT.

- *Analisi dei dati*: ChatGPT può assistere il ricercatore nel determinare i test statistici appropriati per analizzare i dati raccolti dallo studio, come ANOVA, t-test o analisi di regressione (Figura 8.9).

You

Voglio comparare il nuovo programma PBL con i metodi di insegnamento tradizionali nel miglioramento delle performance degli studenti. Voglio usare un approccio di Randomized Control Trial. Il metodo di misurazione dell'outcome sarà il risultato ad esami standardizzati. Che tipo di test statistico dovrei utilizzare per valutare la significatività dei miei risultati?

ChatGPT

Per valutare la significatività dei risultati del tuo Randomized Control Trial (RCT) che utilizza il risultato ad esami standardizzati come metodo di misurazione dell'outcome, potresti considerare l'utilizzo del test t di Student. Ecco una breve spiegazione su come potresti procedere:

Test t di Student per campioni indipendenti:
Questo test è adatto quando vuoi confrontare le medie di due gruppi indipendenti, come nel tuo caso (gruppo PBL vs gruppo tradizionale). Il test t ti aiuterà a determinare se ci sono differenze statisticamente significative tra le medie dei due gruppi in termini di risultati ad esami standardizzati.

Ecco come potresti procedere:

1. **Raccogli i dati:**
   - Ottieni i risultati degli esami standardizzati per entrambi i gruppi (PBL e tradizionale).
2. **Verifica le Assunzioni del Test t:**
   - Assicurati che i dati siano approssimativamente normalmente distribuiti e che le varianze nei due gruppi siano omogenee. Puoi utilizzare test statistici come il test di normalità e il test di Levene per verificare queste assunzioni.
3. **Formula dell'Ipotesi Nulle e Alternative:**
   - **Ipotesi Nulla (H0):** Non ci sono differenze significative tra le medie dei gruppi PBL e tradizionale.
   - **Ipotesi Alternativa (H1):** Ci sono differenze significative tra le medie dei gruppi PBL e tradizionale.
4. **Livello di Significatività (α):**
   - Imposta il tuo livello di significatività (generalmente 0,05 o 0,01), che rappresenta la probabilità di commettere un errore di tipo I.
5. **Calcolo del Test t:**
   - Utilizza la formula del test t di Student per campioni indipendenti per calcolare il valore del test t.
6. **Determinazione della Significatività:**
   - Confronta il valore del test t con il valore critico da tavole di distribuzione t o utilizza un software statistico per ottenere il valore p associato. Se il valore p è inferiore al tuo livello di significatività, puoi rifiutare l'ipotesi nulla.
7. **Interpretazione dei Risultati:**
   - Se il test è significativo, puoi concludere che ci sono differenze statisticamente significative tra i gruppi PBL e tradizionale in termini di risultati ad esami standardizzati.

Ricorda che la scelta del test dipenderà dalle caratteristiche specifiche dei tuoi dati e dalle assunzioni del test. È sempre consigliabile consultare un esperto statistico o utilizzare un software statistico per ottenere risultati precisi e affidabili.

Figura 8.9 ChatGPT suggerisce un test statistico per un dato studio.

Tutto ciò che è stato suggerito da ChatGPT è coerente e trova conferma in documenti riguardanti il modo in cui si conduce un test statistico. Il modello è stato anche in grado di identificare che probabilmente stiamo parlando di una variabile continua (cioè, i punteggi) in modo che sappiamo che tutte le informazioni successive si basano su questa ipotesi. Nel caso in cui volessimo avere punteggi

discreti, potremo aggiustare il prompt aggiungendo questa informazione, e ChatGPT suggerirà un approccio differente.

Il fatto che ChatGPT specifichi le ipotesi e spieghi il suo ragionamento è fondamentale per prendere decisioni sicure sulla base dei suoi input.

In conclusione, ChatGPT può essere uno strumento prezioso per i ricercatori nella progettazione di esperimenti. Utilizzando le sue capacità di elaborazione del linguaggio naturale e la sua vasta base di conoscenze, ChatGPT può aiutare i ricercatori a selezionare strutture appropriate per gli studi, a scegliere le tecniche di campionamento, a identificare le variabili e i risultati di apprendimento e persino a suggerire test statistici per analizzare i dati.

Nel prossimo paragrafo, procederemo nella nostra esplorazione di come ChatGPT può supportare i ricercatori, concentrandoci sulla generazione della bibliografia.

Generazione e formattazione di una bibliografia

ChatGPT può supportare i ricercatori nella generazione della bibliografia fornendo strumenti automatici per citazioni e riferimenti. Questi strumenti possono trarre riferimenti accurati per una vasta gamma di fonti, tra cui libri, articoli, siti web e altro ancora. ChatGPT conosce vari stili di citazione, come APA, MLA, Chicago e Harvard, cosa che permette ai ricercatori di selezionare lo stile appropriato per il loro lavoro. Inoltre, ChatGPT può anche suggerire fonti pertinenti in base all’input del ricercatore, aiutando a semplificare il processo di ricerca e garantendo che nella bibliografia tutte le fonti necessarie siano incluse. Utilizzando questi strumenti, i ricercatori possono

risparmiare tempo e assicurarsi che la loro bibliografia sia accurata e completa.

Prendiamo in considerazione il seguente esempio. Diciamo che abbiamo finalizzato un documento di ricerca intitolato *L'impatto della tecnologia sulla produttività sul posto di lavoro: uno studio empirico.* Nel corso del processo di ricerca e scrittura, abbiamo raccolto i seguenti riferimenti a documenti, siti web, video e altre fonti che dobbiamo includere nella bibliografia. Nell'ordine, tre documenti di ricerca, un video YouTube e un sito web:

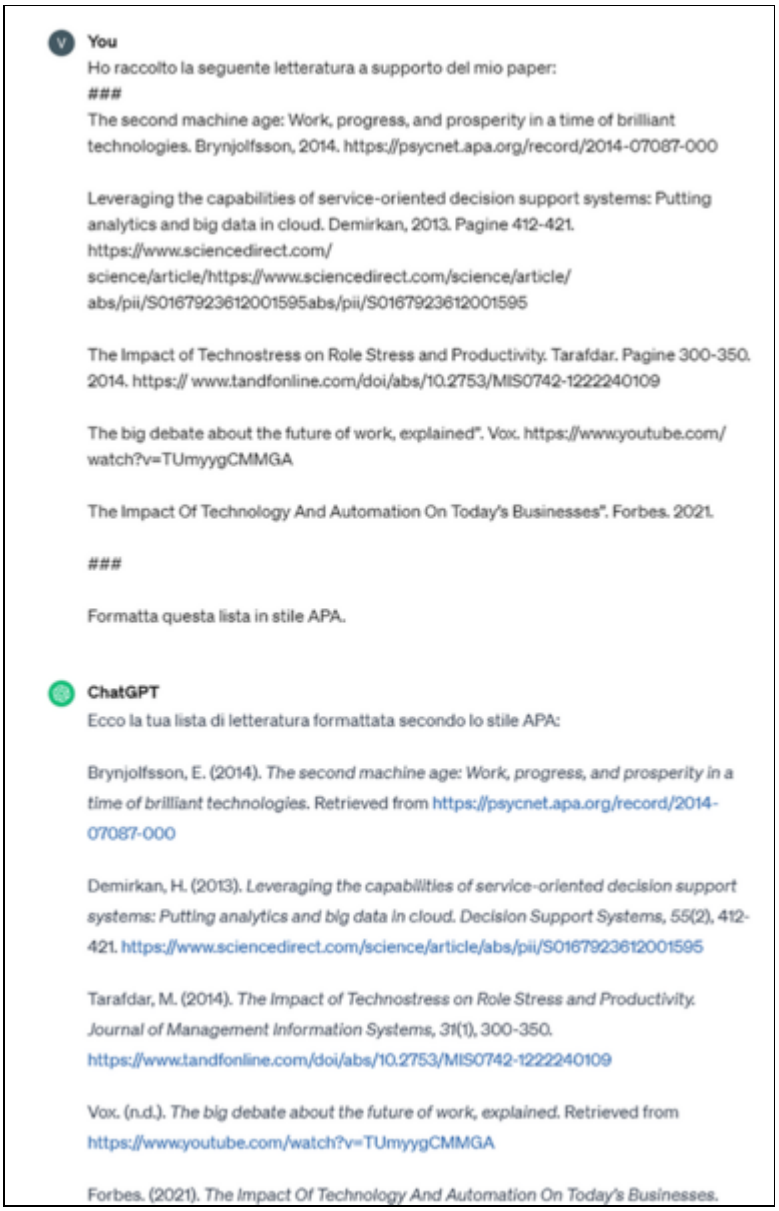
- *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*, Brynjolfsson, 2014, https://psycnet.apa.org/record/2014-07087-000;
- *Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud*, Demirkan, 2013, pp. 412-421, https://www.sciencedirect.com/science/article/abs/pii/S0167923612001595;
- *The Impact of Technostress on Role Stress and Productivity*, Tarafdar, pp. 300-350, 2014, https://www.tandfonline.com/doi/abs/10.2753/MIS0742-1222240109;
- *The big debate about the future of work, explained*, Vox, https://www.youtube.com/watch?v=TUmyygCMMGA;
- *The Impact Of Technology And Automation On Today's Businesses*, Forbes, 2021.

Partendo da questa lista, vogliamo impostare una formattazione che sia in linea con lo stile richiesto dagli studi di ricerca. Per fare ciò, possiamo fornire a ChatGPT la lista grezza di riferimenti e chiedere di rigenerarla con il formato specifico, per esempio, lo stile APA, lo stile ufficiale dell'*American Psychological Association*, comunemente

usato come stile di riferimento in educazione, psicologia, e scienze sociali.

Nella Figura 8.10 vediamo come si comporta ChatGPT.

Una volta formattata la nostra bibliografia, l'ultima tappa del nostro percorso sarà quella di presentare il nostro studio di ricerca. Vediamo nel prossimo paragrafo come ChatGPT può darci una mano.



You

Ho raccolto la seguente letteratura a supporto del mio paper:
###
The second machine age: Work, progress, and prosperity in a time of brilliant technologies. Brynjolfsson, 2014. https://psycnet.apa.org/record/2014-07087-000

Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud. Demirkan, 2013. Pagine 412-421. https://www.sciencedirect.com/science/article/https://www.sciencedirect.com/science/article/abs/pii/S0167923612001595abs/pii/S0167923612001595

The Impact of Technostress on Role Stress and Productivity. Tarafdar. Pagine 300-350. 2014. https:// www.tandfonline.com/doi/abs/10.2753/MIS0742-1222240109

The big debate about the future of work, explained". Vox. https://www.youtube.com/watch?v=TUmyygCMMGA

The Impact Of Technology And Automation On Today's Businesses". Forbes. 2021.

###

Formatta questa lista in stile APA.

ChatGPT

Ecco la tua lista di letteratura formattata secondo lo stile APA:

Brynjolfsson, E. (2014). *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. Retrieved from https://psycnet.apa.org/record/2014-07087-000

Demirkan, H. (2013). *Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud. Decision Support Systems, 55*(2), 412-421. https://www.sciencedirect.com/science/article/abs/pii/S0167923612001595

Tarafdar, M. (2014). *The Impact of Technostress on Role Stress and Productivity. Journal of Management Information Systems, 31*(1), 300-350. https://www.tandfonline.com/doi/abs/10.2753/MIS0742-1222240109

Vox. (n.d.). *The big debate about the future of work, explained*. Retrieved from https://www.youtube.com/watch?v=TUmyygCMMGA

Forbes. (2021). *The Impact Of Technology And Automation On Today's Businesses.*

Figura 8.10 Un elenco di riferimenti generati in formato APA da ChatGPT.

Generazione di una presentazione per lo studio

L’ultima tappa di uno studio di ricerca riguarda spesso la sua presentazione a pubblici differenti. Questo potrebbe implicare la preparazione di una serie di diapositive, un pitch o un webinar in cui il ricercatore deve rivolgersi a diversi tipi di pubblico.

Supponiamo, per esempio, che il nostro studio *Come le interazioni sociali online differiscono da quelle faccia a faccia nell’influenzare il benessere psicologico?*, sia destinato alla discussione di una tesi di laurea magistrale. In questo caso, possiamo chiedere a ChatGPT di produrre una struttura di pitch che dovrebbe durare 15 minuti e aderire al metodo scientifico. Vediamo che tipo di risultati possono essere prodotti.

Il risultato rappresentato nella Figura 8.11 è impressionante. Ai tempi dell’università, mi sarebbe stato utile avere un tale strumento per aiutarmi nella progettazione della mia discussione.

A partire da questa struttura, possiamo chiedere a ChatGPT di generare anche una serie di slide come supporto visivo per la discussione della tesi.

You

Devo discutere la mia tesi di laurea magistrale basata sul mio studio "Come le interazioni sociali online differiscono da quelle faccia a faccia nell'influenzare il benessere psicologico?".
Genera la struttura per un pitch di circa 15 minuti rivolto alla commissione di laurea. Non specificare l'intero contenuto di ogni sezione: è sufficiente la struttura ad alto livello.

ChatGPT

Certamente, ecco una struttura ad alto livello per un pitch di circa 15 minuti sulla tua tesi di laurea magistrale:

1. **Introduzione**
   - Contestualizzazione del tema e sua rilevanza
   - Presentazione della tua domanda di ricerca
2. **Obiettivi della Ricerca**
   - Definizione degli obiettivi principali della tua tesi
   - Spiegazione di come questi obiettivi contribuiranno alla comprensione del tema
3. **Revisione della Letteratura**
   - Breve riepilogo delle principali teorie e studi legati alle interazioni sociali online e faccia a faccia
   - Identificazione di eventuali lacune nella letteratura
4. **Metodologia di Ricerca**
   - Spiegazione della metodologia utilizzata per raccogliere dati
   - Discussione delle variabili considerate e delle tecniche analitiche impiegate
5. **Presentazione dei Risultati**
   - Esposizione dei principali risultati ottenuti durante la ricerca
   - Analisi delle differenze rilevanti tra le interazioni sociali online e faccia a faccia
6. **Discussione dei Risultati**
   - Interpretazione dei risultati alla luce degli obiettivi di ricerca
   - Connessione dei risultati alla letteratura esistente
7. **Implicazioni Pratiche e Teoriche**
   - Riflessione sulle possibili implicazioni dei risultati nel mondo reale
   - Suggerimenti per possibili sviluppi futuri della ricerca
8. **Limitazioni dello Studio**
   - Identificazione e discussione delle limitazioni del tuo studio
   - Considerazioni etiche e possibili bias
9. **Conclusioni**
   - Sintesi dei punti chiave emersi durante la ricerca
   - Ripetizione dell'importanza dei risultati ottenuti
10. **Ringraziamenti**
    - Espressione di gratitudine per chi ha contribuito alla tua ricerca (se applicabile)
11. **Domande e Risposte**
    - Apertura alla commissione di laurea per domande e risposte

Ricorda di gestire il tempo in modo efficace, dedicando una quantità adeguata di tempo a ciascuna sezione e mantenendo un ritmo coerente durante la presentazione.

Figura 8.11 Discussione di una tesi.

NOTA

Nel Capitolo 1, abbiamo già visto un esempio di uno strumento alimentato dall'intelligenza artificiale, *Tome AI*, per la generazione di presentazioni complete, utilizzando la generazione sia di testo sia di immagini. In questo paragrafo, ci concentreremo sulla componente di generazione del testo, ma ricordate che esistono strumenti di

intelligenza artificiale che possono assistervi anche nella creazione della presentazione stessa.

Proseguiamo con la richiesta di una struttura per la presentazione (Figura 8.12).

Infine, immaginiamo che la discussione della nostra tesi sia stata così apprezzata da essere selezionata per ricevere fondi di ricerca al fine di continuare a indagare sull’argomento. Ora abbiamo bisogno di un *elevator pitch* per convincere il comitato di finanziamento. Chiediamo (Figura 8.13) il supporto di ChatGPT.

You

Ok, ora sulla base di questo discorso, generami un template per una presentazione con 5 slides.

ChatGPT

Certamente, ecco un template di base per una presentazione con 5 slides per il tuo pitch di 15 minuti sulla tesi di laurea magistrale:

Slide 1: Introduzione

- Titolo: "Introduzione"
- Contenuto:
  - Breve contestualizzazione del tema
  - Presentazione della tua domanda di ricerca
  - Dichiarazione dell'importanza del tuo studio

Slide 2: Obiettivi della Ricerca

- Titolo: "Obiettivi della Ricerca"
- Contenuto:
  - Definizione chiara degli obiettivi principali
  - Spiegazione di come questi obiettivi contribuiranno alla comprensione del tema
  - Connessione degli obiettivi alle questioni chiave della ricerca

Slide 3: Risultati e Analisi

- Titolo: "Risultati e Analisi"
- Contenuto:
  - Esposizione dei principali risultati ottenuti
  - Analisi delle differenze significative tra interazioni sociali online e faccia a faccia
  - Connessione dei risultati agli obiettivi della ricerca

Slide 4: Implicazioni e Conclusioni

- Titolo: "Implicazioni e Conclusioni"
- Contenuto:
  - Riflessione sulle implicazioni pratiche e teoriche dei risultati
  - Suggerimenti per possibili sviluppi futuri
  - Sintesi dei punti chiave e conclusioni

Slide 5: Domande e Risposte

- Titolo: "Domande e Risposte"
- Contenuto:
  - Invito alla commissione di laurea a fare domande
  - Riepilogo del tuo interesse a rispondere alle domande
  - Ringraziamenti finali e chiusura della presentazione

Assicurati di mantenere un design pulito e professionale per le tue slide, con l'uso di testi chiari, immagini rilevanti e grafici, se necessario. Adatta il contenuto delle slide in base ai dettagli specifici della tua ricerca.

Figura 8.12 Struttura di una presentazione del nostro studio di ricerca.

You
Generami un elevator pitch per la mia tesi. L'audience è una commissione scientifica alla ricerca di progetti da finanziare. Devo convincere la commissione che il mio studio merita un finanziamento.

ChatGPT
Elevator Pitch per la Tua Tesi di Laurea Magistrale:

Buongiorno, membri della commissione scientifica. Sono qui per presentarvi il mio progetto di ricerca di tesi magistrale, un'indagine approfondita su come le interazioni sociali online si differenzino da quelle faccia a faccia nel loro impatto sul benessere psicologico.

Il mondo è in continua evoluzione, con una crescente dipendenza dalle piattaforme online per le interazioni sociali. Il mio studio si propone di rispondere a una domanda cruciale: in che modo queste dinamiche digitali influenzano il nostro benessere psicologico rispetto alle interazioni tradizionali?

Attraverso una robusta metodologia di ricerca, ho raccolto dati significativi che evidenziano differenze sostanziali tra questi due contesti sociali. Dai risultati emersi, non solo possiamo comprendere meglio l'impatto delle interazioni online sulla salute mentale, ma possiamo anche identificare possibili strategie per migliorare il benessere degli individui in entrambi i contesti.

I risultati di questa ricerca non solo arricchiranno il campo della psicologia sociale, ma avranno implicazioni pratiche su come progettare ambienti digitali più favorevoli per la salute mentale. Il finanziamento che chiedo oggi è cruciale per garantire la continuità e l'approfondimento di questa ricerca, aprendo la strada a soluzioni innovative e orientate al benessere.

Vi invito a investire in questo progetto in quanto non solo contribuirà alla nostra comprensione delle dinamiche sociali contemporanee, ma avrà anche un impatto positivo sulla vita di coloro che interagiscono quotidianamente in ambienti online e offline.

Grazie per l'attenzione e sono disponibile a rispondere alle vostre domande.

Figura 8.13 Elevator pitch per la tesi.

Possiamo sempre adattare i risultati e renderli più allineati a quello che stiamo cercando, tuttavia, avere strutture e framework già disponibili può far risparmiare molto tempo e ci permette di concentrarci di più sul contenuto tecnico che vogliamo portare.

Nel complesso, ChatGPT è in grado di supportare un percorso end-to-end nella ricerca, dal brainstorming delle domande di ricerca alla

generazione del pitch finale dello studio, e abbiamo dimostrato come possa essere un ottimo assistente per i ricercatori.

Inoltre, notate che nel campo della ricerca, sono stati sviluppati di recente alcuni strumenti diversi da ChatGPT, ma comunque alimentati da modelli LLM (*Large Language Model*). Un esempio è `Humanata.ai`, uno strumento potenziato dall'AI che permette di caricare i vostri documenti e di eseguire diverse azioni su di essi, tra cui il riassunto, le domande e risposte istantanee e la generazione di nuovi documenti basati sui file caricati.

Questo suggerisce come gli strumenti alimentati da LLM (incluso ChatGPT) stiano aprendo la strada a diverse innovazioni all'interno del dominio della ricerca.

Riepilogo

In questo capitolo, abbiamo esplorato l'uso di ChatGPT come uno strumento prezioso per i ricercatori. Attraverso il brainstorming relativo alla domanda di ricerca, la progettazione degli esperimenti, la generazione e la formattazione della bibliografia e la generazione della presentazione, ChatGPT può assistere il ricercatore nell'accelerare quelle attività a basso valore aggiunto, in modo che possa concentrarsi su attività rilevanti.

Notate che ci siamo occupati solo su un piccolo insieme di attività in cui ChatGPT può supportare i ricercatori. Ci sono molte altre attività nel dominio della ricerca che potrebbero beneficiare del supporto di ChatGPT, tra cui la raccolta di dati, il reclutamento di partecipanti allo studio, il networking di ricerca, l'engagement del pubblico e molti altri.

I ricercatori che incorporano questo strumento nel loro lavoro possono beneficiare della sua versatilità e dell'impatto positivo che

può avere sulla produttività e sulla qualità dei risultati.

Tuttavia, è importante tenere a mente che ChatGPT è solo uno strumento e dovrebbe essere utilizzato in combinazione con la conoscenza e il giudizio di un ricercatore esperto. Come in ogni progetto di ricerca, una considerazione attenta della domanda di ricerca e della struttura dello studio è necessaria per garantire la validità e l'affidabilità dei risultati.

Con questo capitolo, chiudiamo anche la Parte 2 di questo libro, che si è concentrata sulla vasta gamma di domini in cui è possibile utilizzare ChatGPT, con un focus particolare all'utilizzo di questo strumento per la produttività individuale o di piccoli team.

A partire dalla Parte 3, sposteremo il focus su come le grandi organizzazioni possono sfruttare la stessa AI generativa di ChatGPT per progetti su scala aziendale, utilizzando le API dei modelli OpenAI disponibili sul cloud di Microsoft Azure.

Riferimenti

- https://scholarworks.lib.csusb.edu/cgi/viewcontent.cgi?article=2131&context=etd.
- https://www.spotlightonresearch.com/mental-health-research/the-impact-of-social-media-on-mental-health.
- https://digitalcommons.bard.edu/cgi/viewcontent.cgi?article=1028&context=senproj_f2020.
- https://www.humata.ai.

Parte III

OpenAI per le aziende

Questa parte inizia con un’introduzione alla partnership tra OpenAI e Microsoft e il conseguente lancio sul mercato di Azure OpenAI Service, un servizio gestito in cloud che offre i modelli OpenAI con tutta la scalabilità, flessibilità e sicurezza tipica delle architetture cloud.

Prosegue poi con una sezione pratica in cui avrete una panoramica dei casi d’uso aziendali.

Ogni caso d’uso fornisce una descrizione dello scenario aziendale e un’implementazione end-to-end con Python e le API dei modelli di Azure OpenAI. Gli scenari trattati sono l’assistente HR, l’analisi dei contratti, l’analisi del call center e la ricerca semantica.

Questa parte si conclude con un riassunto di tutto ciò che è stato trattato nel libro, compresi gli ultimi annunci e *release* che sono avvenute nelle ultime settimane (per esempio, l’introduzione di modelli multimodali di grandi dimensioni come GPT-4). Contiene anche alcune riflessioni e pensieri finali sulla crescita esponenziale delle tecnologie di AI generativa in pochi mesi e su che cosa dobbiamo aspettarci nel prossimo futuro.

Introduzione ad Azure OpenAI

In questo capitolo, ci concentreremo sulle applicazioni a livello aziendale dei modelli di OpenAI e presenteremo la partnership tra OpenAI e Microsoft, introducendo il servizio AOAI (*Azure OpenAI*). Passeremo in rassegna le tappe e gli sviluppi di Microsoft nel campo dell'intelligenza artificiale, evidenziando il percorso che ha portato il cloud Azure a diventare l'infrastruttura a supporto di OpenAI, e vedremo quali sono le implicazioni per le grandi organizzazioni. Infine, parleremo del tema dell'AI responsabile e di come garantire che i sistemi di AI rispettino gli standard etici.

Alla fine di questo capitolo, avrete appreso le principali caratteristiche del servizio AOAI e come questo si differenzi dai modelli di OpenAI di cui abbiamo discusso finora.

Requisiti tecnici

Di seguito sono elencati i requisiti tecnici per questo capitolo.

- Una sottoscrizione su Azure, che potete creare gratuitamente in: `https://azure.microsoft.com/free/cognitive-services`.
- Una sottoscrizione Azure abilitata al deployment di istanze di AOAI. L'accesso a questo servizio viene concesso solo su richiesta. Potete richiedere l'accesso ad Azure OpenAI compilando il modulo su `https://aka.ms/oai/access`.

OpenAI e Microsoft per l’AI a livello aziendale

Microsoft ha una lunga storia di investimenti nella ricerca e sviluppo nel campo dell’AI, con un focus sulla creazione di strumenti e servizi che possano essere utilizzati da aziende e individui per risolvere problemi complessi e migliorare la produttività.

L’azienda vanta anche una serie di traguardi in termini di *human parity*, come il riconoscimento vocale (2017), la traduzione automatica (2018), le domande e risposte conversazionali (2019), la descrizione delle immagini (2020) e la comprensione del linguaggio naturale (2021).

DEFINIZIONE

La *human parity* in ambito dell’intelligenza artificiale si riferisce al punto in cui l’AI raggiunge prestazioni paragonabili a quelle umane in specifici compiti o aree. In altre parole, quando mettiamo a confronto l’AI e gli esseri umani nello stesso test, la parità umana indica che l’AI è altrettanto “brava” di un essere umano in generale.

Nei prossimi paragrafi esploreremo la storia della ricerca e il background di Microsoft nel dominio dell’AI, per capire il percorso che ha portato alla partnership con OpenAI e, infine, allo sviluppo del servizio di AOAI.

Background di Microsoft AI

Le prime ricerche di Microsoft nel campo dell’AI risalgono alla fine degli anni Novanta, quando l’azienda ha istituito il suo team di machine learning (ML) e statistica applicata. Da allora in poi, Microsoft ha iniziato a ricercare e sperimentare agenti intelligenti e assistenti virtuali. Il primo risultato di queste ricerche fu *Clippy* (Figura 9.1), un assistente digitale personale per Microsoft Office.

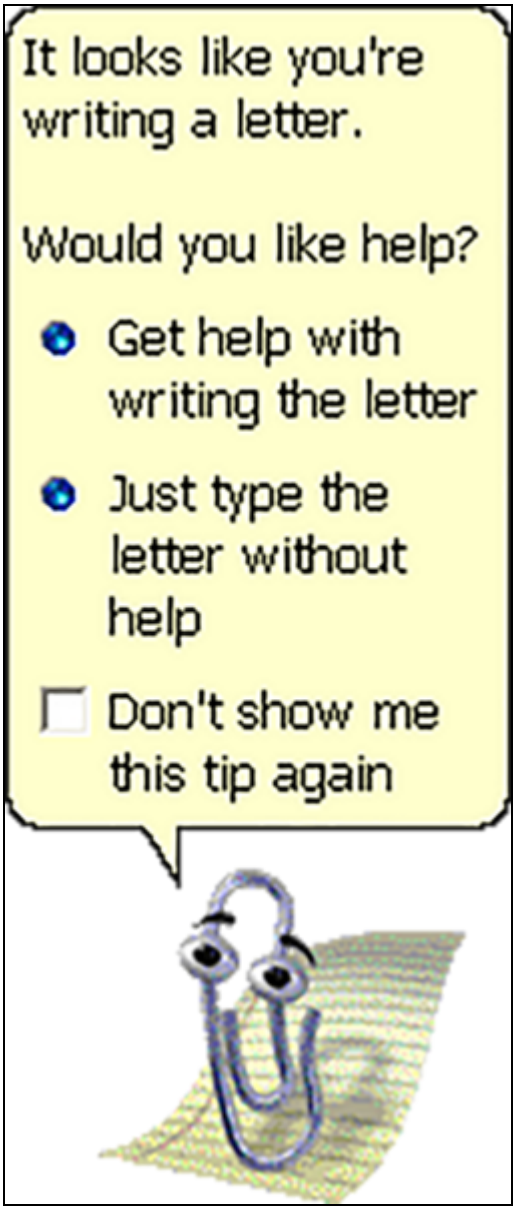


Figura 9.1 Clippy, l’assistente di Office lanciato nel 2000.

Clippy è stato il precursore di strumenti più sofisticati, come *Cortana*. Lanciato nel 2014, Cortana è un assistente digitale che utilizza l’elaborazione del linguaggio naturale e il machine learning per fornire assistenza personalizzata agli utenti.

Poi, nel 2016, come espansione del Project Oxford di Microsoft, quest’ultima ha lanciato i *Cognitive Services* (Figura 9.2) nel cloud

Azure, un insieme di API che forniscono capacità di AI agli sviluppatori senza che questi richiedano competenze di machine learning e data science.

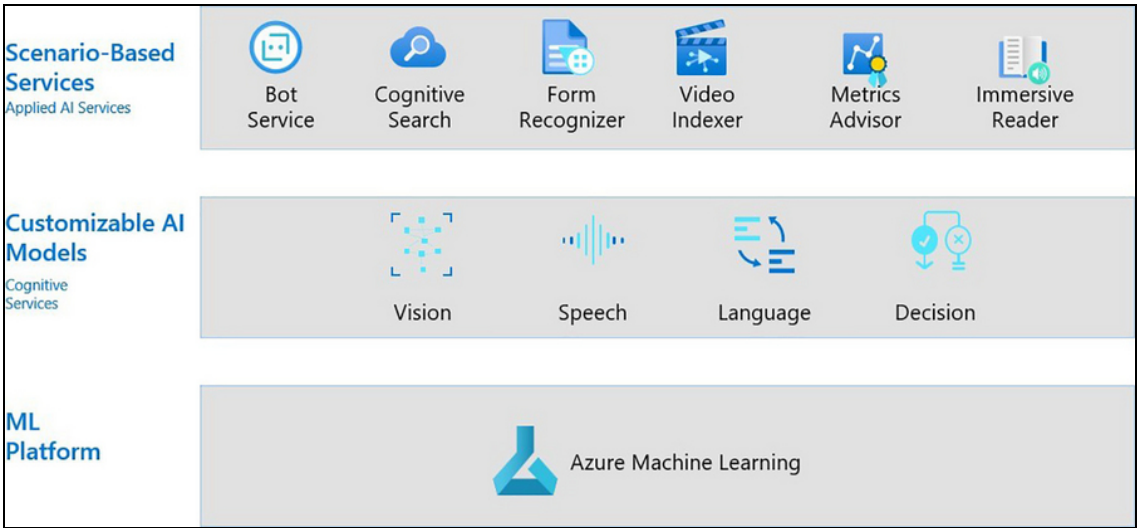


Figura 9.2 I servizi AI di Microsoft Azure.

Con i Cognitive Services, l'AI poteva finalmente essere consumata da un'ampia gamma di utenti, dalle grandi imprese agli sviluppatori indipendenti. Da quel momento, abbiamo assistito a quello che ora chiamiamo democratizzazione dell'AI: l'AI non è più un privilegio per coloro che hanno profonde conoscenze tecniche e hardware potente e costoso per l'addestramento dei modelli. I Cognitive Services sono stati sviluppati per i seguenti motivi:

- affinché chiunque, dai data scientist agli utenti aziendali, possa sfruttare servizi di AI con un approccio *low-code*;
- per fornire un insieme di modelli pre-costruiti che sono già stati addestrati, vale a dire, sono pronti all'uso e per funzionare non necessitano di un'infrastruttura hardware specializzata (tipicamente una GPU), in quanto già ospitati su infrastruttura cloud Azure.

Gli investimenti di Microsoft in ambito AI si sono manifestati anche in termini di acquisizioni di aziende che operano nel settore negli ultimi anni, tra cui nel 2016 l'app di tastiera predittiva SwiftKey (https://blogs.microsoft.com/blog/2016/02/03/microsoft-acquires-swiftkey-in-support-of-re-inventing-productivity-ambition), nel 2017 la startup di deep learning Maluuba (https://blogs.microsoft.com/blog/2017/01/13/microsoft-acquires-deep-learning-startup-maluuba-ai-pioneer-yoshua-bengio-advisory-role) e nel 2018 la piattaforma per la costruzione di modelli di AI Bonsai (https://blogs.microsoft.com/blog/2018/06/20/microsoft-to-acquire-bonsai-in-move-to-build-brains-for-autonomous-systems).

Tra le aziende sulle quali Microsoft ha investito e con cui ha stretto una partnership, c'è anche OpenAI.

La relazione tra le due aziende è iniziata nel 2016, quando OpenAI ha accettato di sfruttare l'infrastruttura cloud di Azure per i suoi esperimenti di AI. Successivamente, nel 2019, Microsoft ha annunciato una partnership da un miliardo di dollari con OpenAI (https://news.microsoft.com/2019/07/22/openai-forms-exclusive-computing-partnership-with-microsoft-to-build-new-azure-ai-supercomputing-technologies) per sviluppare modelli e tecnologie di AI che possano essere utilizzati a beneficio dell'umanità. Questa partnership è stata stipulata su tre pilastri principali:

- Microsoft e OpenAI costruiranno congiuntamente una nuova infrastruttura di supercalcolo Azure per addestrare i modelli di AI;
- OpenAI renderà i suoi modelli disponibili e utilizzabili dal cloud Azure;
- Microsoft diventerà il *preferred partner* di OpenAI per commercializzare nuove soluzioni di AI sul mercato.

Da allora, le due aziende hanno continuato a investire e a fare co-ricerca. Infine, nel gennaio 2023, in Azure è stato reso disponibile un insieme di modelli OpenAI, tramite il servizio di Azure OpenAI.

Con la *general availability* del servizio di AOAI è stato raggiunto un nuovo traguardo, e il portfolio di AI di Microsoft è stato esteso con i potenti LLMs di OpenAI.

Azure OpenAI Service

Il servizio di AOAI è un prodotto di Microsoft che fornisce accesso tramite API REST ai potenti modelli linguistici di OpenAI come GPT-3.5, GPT-4 e DALL-E.

Nel contesto del portafoglio Azure AI di Microsoft, il servizio di AOAI si colloca tra le offerte di servizi cognitivi rappresentate nella Figura 9.3).

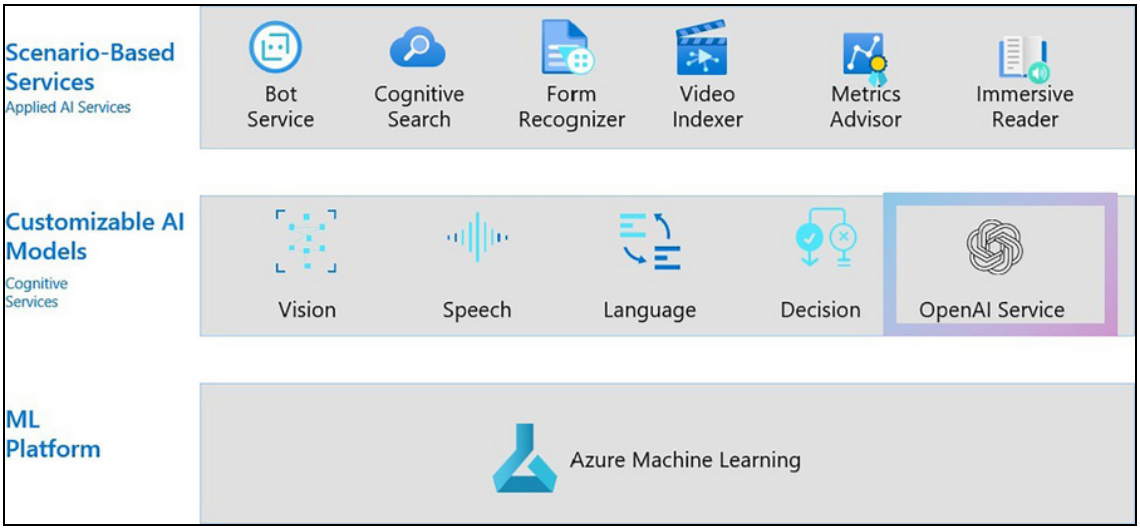


Figura 9.3 Disponibilità generale del servizio AOAI (GA).

Come per qualsiasi altra offerta di servizi cognitivi, AOAI offre modelli che sono già stati addestrati e sono pronti per essere utilizzati.

Per creare la vostra risorsa AOAI, potete seguire queste istruzioni.

1. Raggiungere il vostro portale Azure all'indirizzo `https://ms.portal.azure.com`.
2. Fate clic su *Crea una risorsa*.
3. Digitate `azure openai` e fate clic su *Crea*.
4. Compilate le informazioni richieste e fate clic su *Rivedi + crea*.

Questo processo è rappresentato nella Figura 9.4.

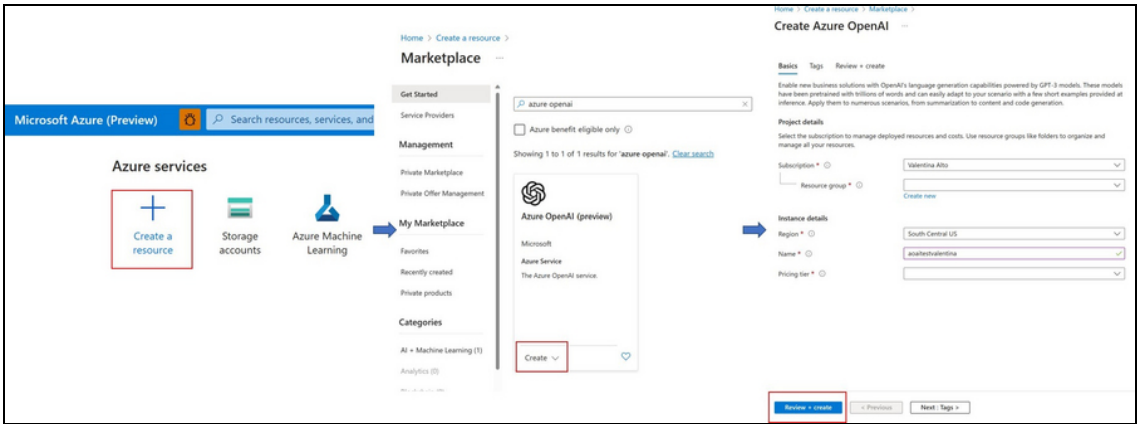


Figura 9.4 Passaggi per creare una risorsa AOAI.

Questo processo potrebbe richiedere qualche minuto. Una volta pronto, potrete accedere direttamente all' interfaccia *user-friendly* dell'AOAI Playground (Figura 9.5), per sottoporre a test i vostri modelli prima di distribuirli:

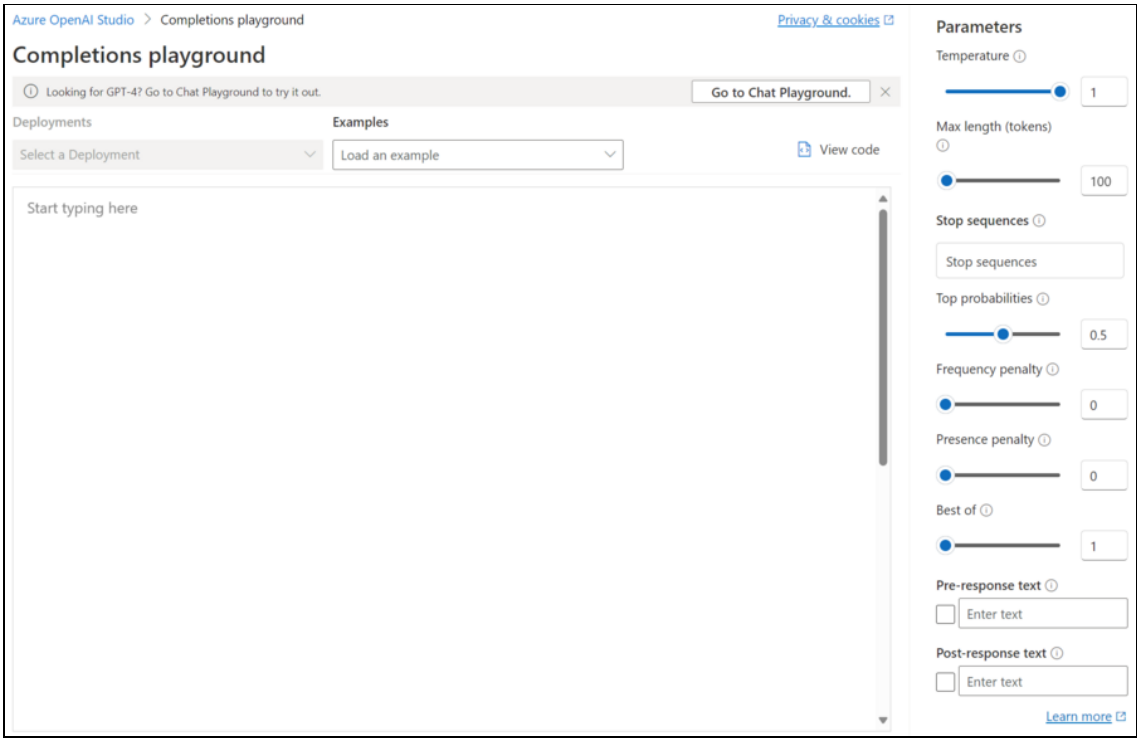


Figura 9.5 Interfaccia utente AOAI e Playground.

Notate che AOAI Playground è analogo alla versione OpenAI Playground che abbiamo visto nel Capitolo 2. La differenza è che, per utilizzare i modelli AOAI, dovete avviare un deployment, che è un'istanza di calcolo *serverless* che potete collegare a un modello. Potete farlo sia in Playground sia nella pagina backend della risorsa nel portale Azure (Figura 9.6).

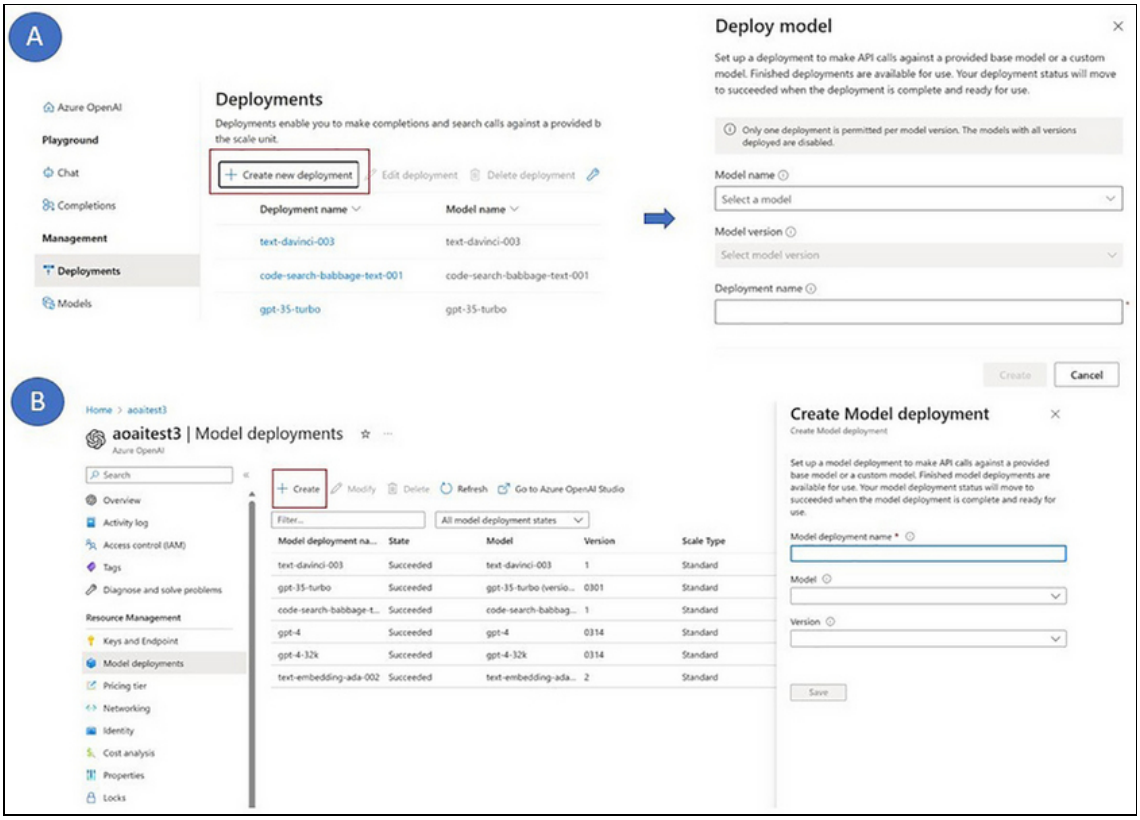


Figura 9.6 Creazione di un nuovo deployment AOAI tramite Playground (A) o nel portale Azure (B).

Per esempio, ho creato un deployment chiamato *text-davinci-003* con l'omonimo modello associato (Figura 9.7).

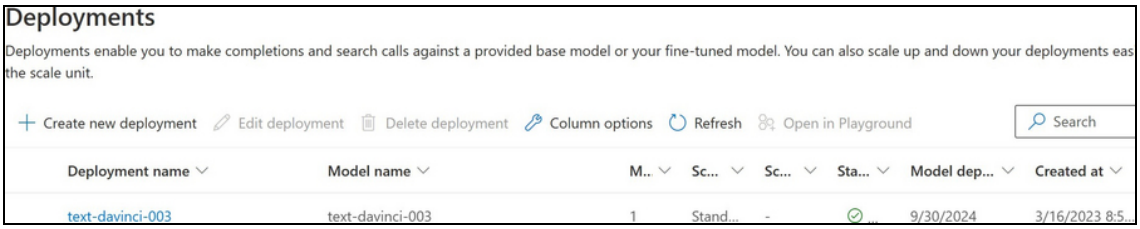


Figura 9.7 Un deployment attivo di AOAI.

Nel Playground OpenAI, possiamo sottoporre a test questi modelli direttamente tramite l'interfaccia utente o incorporando le loro API nelle nostre applicazioni. Nel prossimo paragrafo esploreremo come interagire con il Playground e provare diverse configurazioni di

modelli. Nel Capitolo 10, impareremo a integrare l'API dei modelli di AOAI nelle applicazioni aziendali.

Esplorare il Playground

L'AOAI Playground è il modo più semplice per familiarizzare con i modelli disponibili e iniziare a pianificare quale versione del modello è la più adatta per i vostri progetti. L'interfaccia utente presenta diverse schede e aree di lavoro, come mostrato nella Figura 9.8.

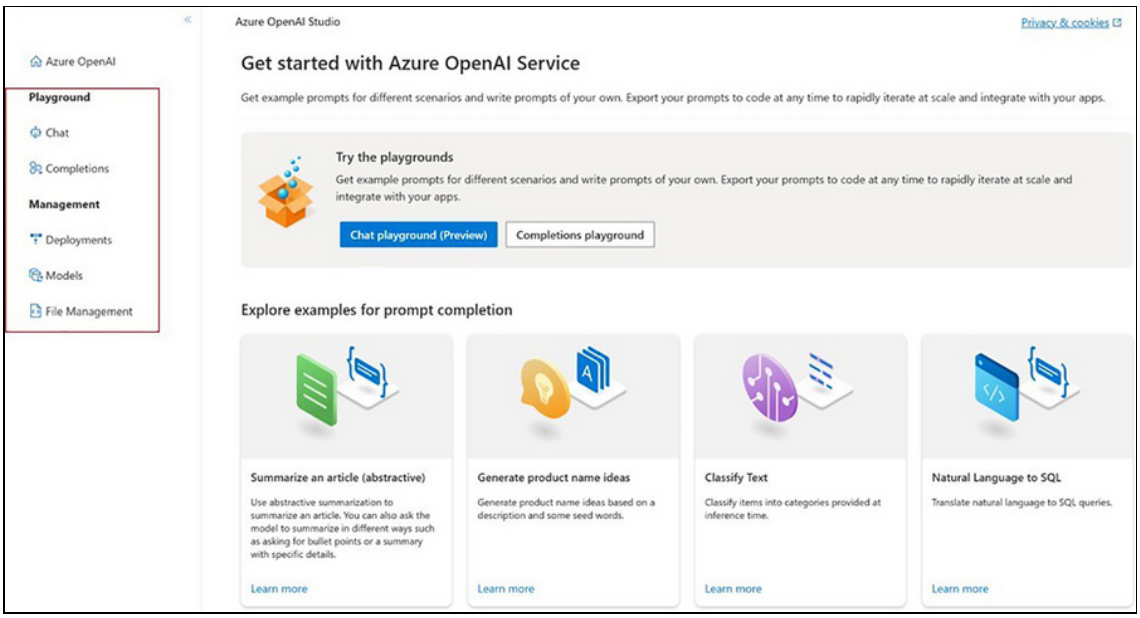


Figura 9.8 Panoramica di AOAI Playground.

Esploriamo ciascuna di esse.

- *Playground* | *Chat*: l'area di lavoro Chat (Figura 9.9) è progettata per essere utilizzata solo con modelli conversazionali come GPT-3.5-turbo (il modello su cui si basa ChatGPT).Offre un'esperienza simile a ChatGPT, con la possibilità di configurare il vostro modello con parametri aggiuntivi (come abbiamo visto nel Capitolo 2 con OpenAI Playground). Inoltre, c'è una caratteristica

aggiuntiva che rende l'area di lavoro Chat molto interessante, ossia il *System message* (Figura 9.10).

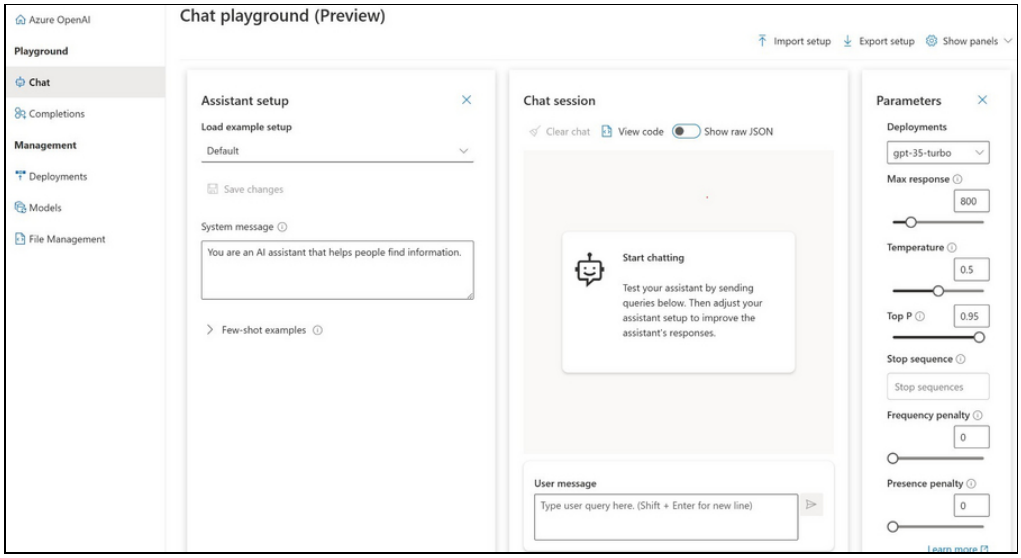


Figura 9.9 Area di lavoro Chat AOAI.

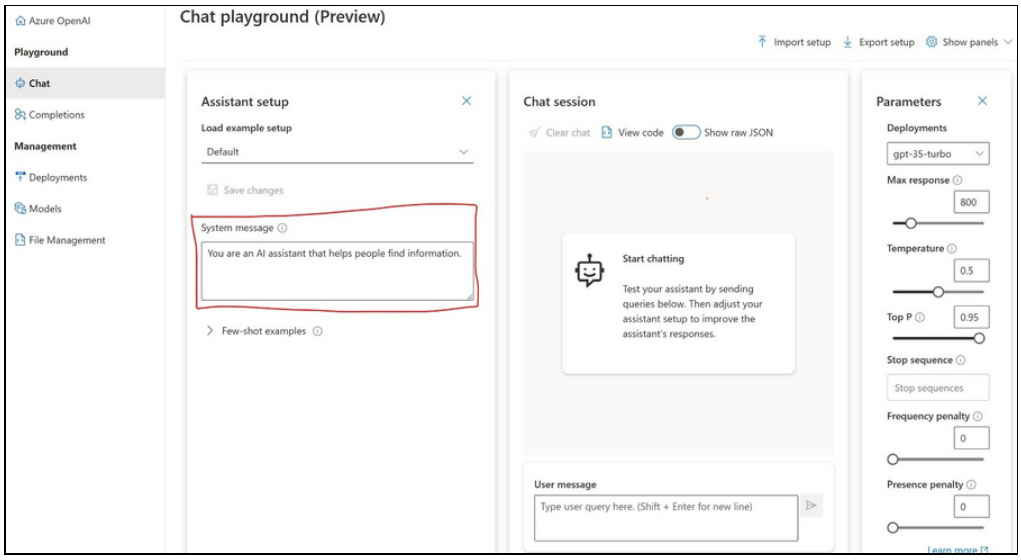


Figura 9.10 Esempio di un messaggio di sistema.

Il *system message* è l'insieme di istruzioni che diamo al modello per definire come dovrà comportarsi e interagire con noi. Come per il prompt, il *system message* rappresenta una componente

chiave della configurazione di un modello, poiché influisce notevolmente sulle sue prestazioni.
Per esempio, possiamo dare istruzioni al nostro modello per comportarsi come un assistente per la formattazione JSON.
Come potete vedere dalla Figura 9.11, il modello è stato in grado di produrre un file JSON attraverso alcuni semplici dati, come nome ed età, senza la necessità di specificare alcuna etichetta.

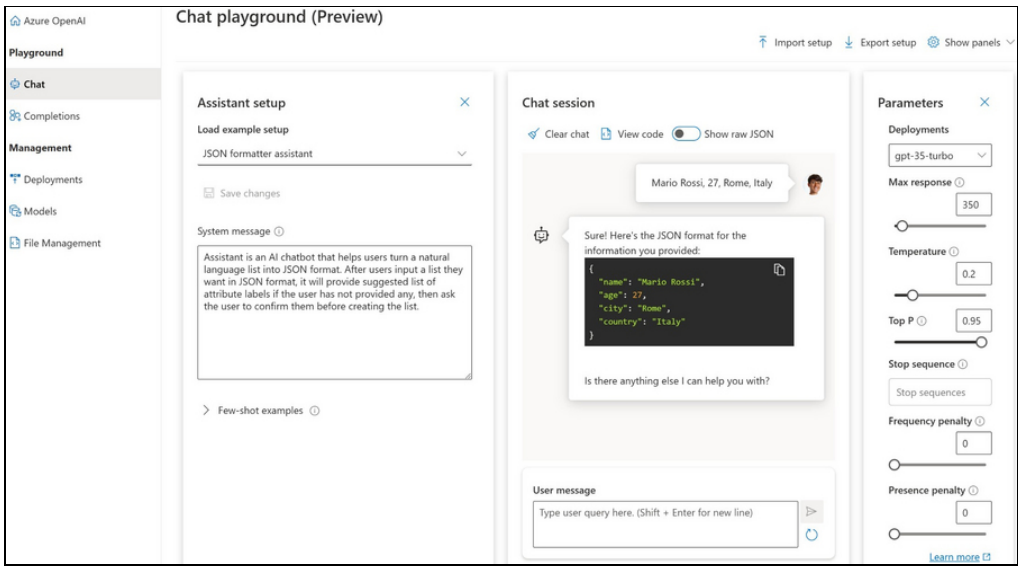


Figura 9.11 Esempio di un modello che agisce come un assistente per la formattazione JSON.

- *Playground* | *Completions*: diversamente dall'area di lavoro precedente, l'area di lavoro *Completions* offre una sorta di foglio bianco dove potete interagire con i vostri modelli. Mentre GPT-3.5-turbo è progettato per attività conversazionali (il che significa che può essere consumato tramite un'interfaccia tipo chatbot), la serie GPT-3 contiene modelli più generali e può essere utilizzata per un'ampia gamma di compiti linguistici, come la generazione di contenuti, la sintesi e così via.
Per esempio, potremmo chiedere al nostro modello di generare un

quiz (Figura 9.12) dandogli una descrizione dell'argomento e un esempio one-shot.

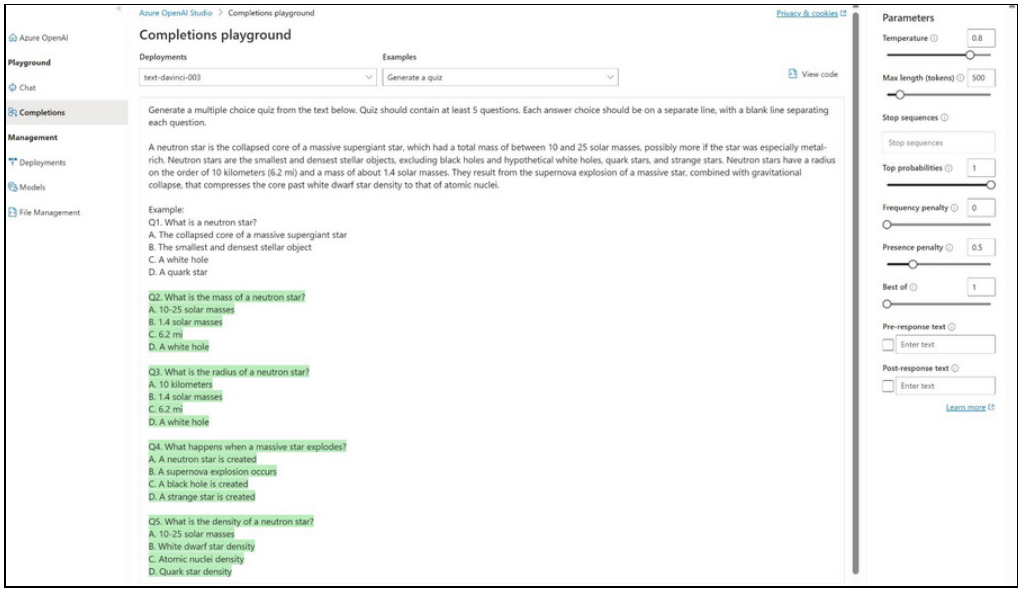


Figura 9.12 Esempio di un modello GPT che genera un quiz.

Infine, come per l'area di lavoro *Chat*, con *Completions* potete configurare parametri come il numero massimo di token o la temperatura (fate riferimento al Capitolo 2 per un elenco completo di questi parametri e del loro significato).

- *Gestione | Deployments*: nella scheda *Deployments*, potete creare e gestire nuovi deployment (Figura 9.13) da associare ai modelli AOAI.

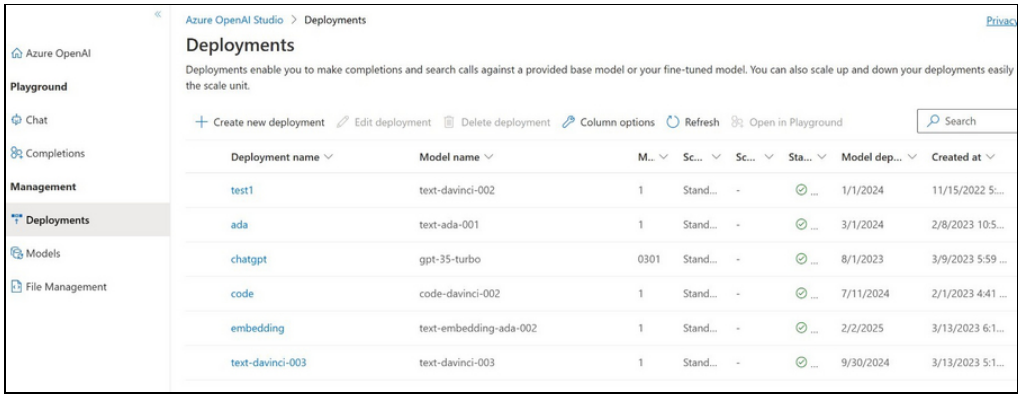


Figura 9.13 Lista dei deployment AOAI.

Ogni deployment può ospitare un solo modello. Potete modificare o eliminare i vostri deployment in qualsiasi momento. Come abbiamo già visto, il deployment di un modello è il passo abilitante per l'utilizzo sia dell'area di lavoro *Completions* sia dell'area di lavoro *Chat* all'interno del servizio AOAI.

- *Gestione* | *Modelli*: in questa scheda potete valutare rapidamente i modelli che sono disponibili all'interno del servizio di AOAI e, tra questi, quelli che possono essere distribuiti (ossia un modello che non è ancora stato distribuito). Per esempio, nella Figura 9.14 abbiamo un modello `text-similarity-curie-001`. Non ha un deployment associato, quindi può essere distribuito (come mostra la colonna *Deployable*). Al contrario, `text-similarity-ada-002` ha già un deployment, quindi non è più disponibile.
  All'interno di questa scheda, potete anche creare un modello personalizzato seguendo una procedura chiamata fine-tuning (Figura 9.15) che abbiamo visto nel Capitolo 2.
  A partire da questo *widget* guidato, potete caricare i vostri dati di addestramento e validazione per produrre un modello personalizzato, a partire da un modello base (un `text-davinci-003`, per esempio), che sarà ospitato su un deployment dedicato.

NOTA

Nel Capitolo 2, abbiamo visto che il set di dati per l'addestramento dovrebbe essere allineato con un formato specifico del tipo seguente (chiamato JSONL):

```
{"prompt": "<prompt text>", "completion": "<ideal generated text>"}
{"prompt": "<prompt text>", "completion": "<ideal generated text>"}
{"prompt": "<prompt text>", "completion": "<ideal generated text>"}
```

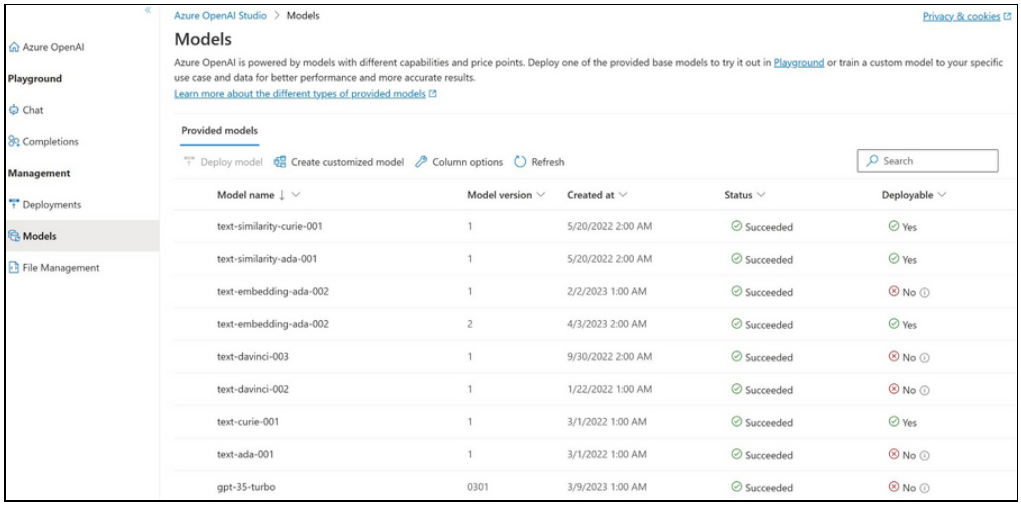


Figura 9.14 Lista dei modelli AOAI.

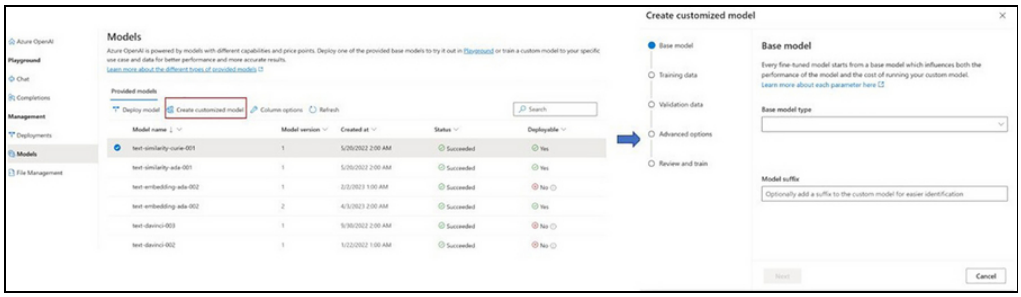


Figura 9.15 Esempio di fine-tuning del modello.

Per facilitare questa formattazione, OpenAI ha sviluppato uno strumento in grado di formattare i vostri dati in questo formato specifico, pronto per il fine-tuning. Può anche fornire suggerimenti su come modificare i dati in modo che lo strumento possa essere utilizzato per il fine-tuning. Inoltre, accetta vari formati di dati come input, tra cui CSV, TXT e JSON.
Per utilizzare questo strumento, è possibile inizializzare l'interfaccia a riga di comando di OpenAI (CLI) eseguendo il seguente comando:

```
pip install --upgrade openai
```

Una volta inizializzato, è possibile eseguire lo strumento, nel seguente modo:

```
openai tools fine_tunes.prepare_data -f <LOCAL_FILE>
```

- *Gestione* | *Gestione dei file*: infine, nella scheda *Gestione dei file*, potete gestire e caricare i vostri dati di training e test direttamente dall'interfaccia utente, come mostrato nella Figura 9.16.

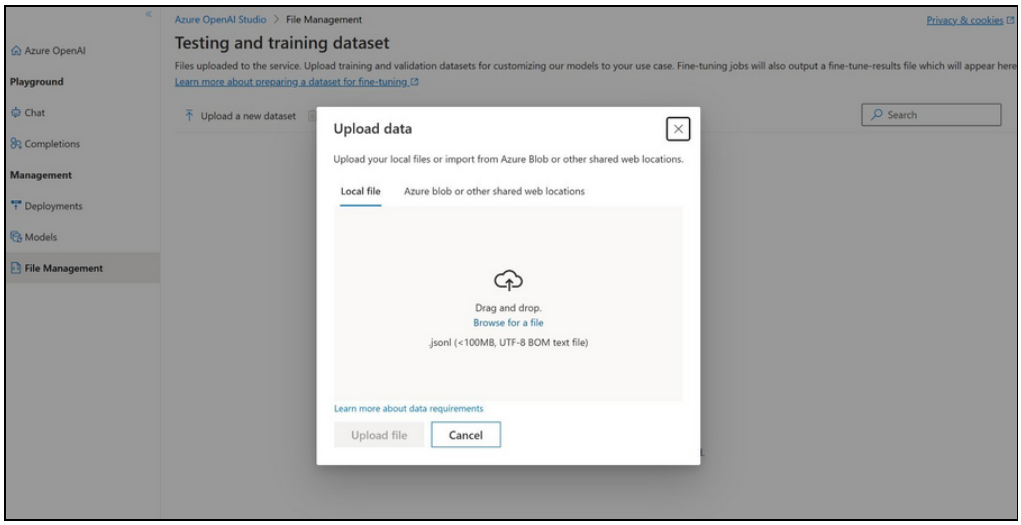


Figura 9.16 Esempio di caricamento di un file all'interno del servizio AOAI.

Potete decidere di caricare i file selezionando *File locale* o *Azure blob* (uno storage di Azure).
Una volta caricati i vostri file, sarete in grado di selezionarli durante la creazione di modelli personalizzati, tramite la scheda *Modelli*.

Infine, come accennato nel paragrafo precedente, ogni modello offre un'API REST che può essere consumata nelle vostre applicazioni.

Nel prossimo capitolo, vedremo molte implementazioni end-to-end dell'uso dell'API dei modelli di AOAI. Nel prossimo paragrafo, invece, comprenderemo in che misura Azure OpenAI si distingue dai modelli standard di OpenAI e perché il cloud di Azure entra a far parte del gioco.

Perché introdurre un cloud pubblico?

All'inizio di questo capitolo, abbiamo visto come Microsoft e OpenAI abbiano rafforzato la loro partnership negli ultimi anni e come il cloud di Microsoft, Azure, sia diventato l'infrastruttura sottostante i modelli di OpenAI. Oltre a ciò, è diventato anche l'infrastruttura cloud dove i modelli di OpenAI possono essere consumati dagli utenti finali.

Ma qual è la differenza tra l'utilizzo dei modelli di OpenAI e Azure OpenAI? La differenza risiede nell'infrastruttura sottostante: con Azure OpenAI, state sfruttando la vostra infrastruttura nel perimetro della vostra sottoscrizione Azure. Questo porta una serie di vantaggi.

- *Scalabilità e flessibilità*: potete beneficiare della scalabilità di Azure e adattare l'uso elastico dei modelli AOAI. Dai piccoli piloti ai progetti di produzione a livello aziendale, AOAI vi

permette di sfruttare la capacità richiesta e di aumentarla o ridurla, se necessario.

- *Sicurezza e conformità*: potete utilizzare l'autenticazione basata sui ruoli e la connettività di rete privata per rendere il vostro deployment più sicuro e affidabile. Potete anche addestrare il vostro modello di intelligenza artificiale mantenendo il pieno controllo dei vostri dati.
- *Disponibilità regionale*: potete eseguire i vostri carichi di lavoro di intelligenza artificiale sull'infrastruttura globale di Azure che soddisfa le vostre esigenze di produzione.
- *Intelligenza artificiale responsabile integrata*: potete utilizzare il filtering dei contenuti per garantire che il vostro modello di intelligenza artificiale generi output appropriati ed etici.

Con i modelli OpenAI disponibili in Azure, siamo in grado di soddisfare tutte le esigenze di sicurezza e scalabilità tipiche delle grandi organizzazioni.

Uno dei vantaggi precedentemente menzionati merita un focus particolare: l'intelligenza artificiale responsabile. Il rapido sviluppo delle tecnologie di intelligenza artificiale deve essere affrontato anche in termini di strumenti etici. Questo è ciò che Microsoft ha studiato dal 2016, come esploreremo nel prossimo paragrafo.

Che cosa si intende per intelligenza artificiale responsabile

Abbiamo menzionato l'intelligenza artificiale responsabile integrata come una delle caratteristiche chiave di AOAI. Tuttavia, per comprenderla appieno, dobbiamo prima capire l'impegno e il percorso di Microsoft verso l'intelligenza artificiale responsabile.

Il percorso di Microsoft verso l’AI responsabile

Microsoft ha presto riconosciuto che, man mano che le tecnologie di intelligenza artificiale continuano ad avanzare e a integrarsi sempre più nella nostra vita, c’è un crescente bisogno di garantire che questi sistemi vengano sviluppati e utilizzati in modo responsabile, etico e a beneficio dell’umanità.

L’inizio di questo percorso risale al 2016, quando il CEO di Microsoft, Satya Nadella, ha scritto un articolo che esplorava come gli esseri umani e l’intelligenza artificiale possono lavorare insieme per risolvere le sfide più grandi della società e che introduceva i primi concetti di intelligenza artificiale responsabile, tra cui la trasparenza, l’equità, la privacy e così via.

Poco dopo, nel 2017, Microsoft ha formalizzato questi concetti con il primo comitato di etica dell’intelligenza artificiale – AETHER (*AI, Ethics, and Effects in Engineering and Research*), formato come gruppo consultivo per il Senior Leadership Team di Microsoft.

AETHER ha trascorso del tempo in un confronto con clienti e gli esperti interni, e poi ha collaborato con Legal Affairs per pubblicare il libro intitolato *The Future Computed: Artificial Intelligence and its role in society* nel gennaio 2018. In questo libro, Microsoft identifica sei principi destinati a guidare lo sviluppo dei sistemi di intelligenza artificiale di un’azienda, così come ad aiutare a informare l’industria più ampia e la società nel suo insieme sulle pratiche di intelligenza artificiale responsabile.

I sei principi di Microsoft per un’intelligenza artificiale responsabile sono i seguenti.

- *Equità*: Microsoft mira a creare sistemi di intelligenza artificiale che siano imparziali e trattino in modo equo tutti gli individui e i

gruppi, senza discriminazioni o pregiudizi.

- *Affidabilità e sicurezza*: Microsoft cerca di creare sistemi di intelligenza artificiale che siano robusti, affidabili e sicuri, e che non compromettano la sicurezza o creino danni anche non intenzionali.
- *Privacy e sicurezza*: Microsoft valuta la privacy e la sicurezza degli individui e dei loro dati, e lavora per proteggerli attraverso la trasparenza e l'uso responsabile delle tecnologie di intelligenza artificiale.
- *Inclusività*: Microsoft crede che l'intelligenza artificiale dovrebbe essere progettata per potenziare e includere individui di diversi background e per favorire pari opportunità per tutti.
- *Trasparenza*: Microsoft crede nella trasparenza e nella responsabilità per le decisioni e le azioni dei sistemi di intelligenza artificiale e si impegna a fornire spiegazioni chiare per i loro risultati.
- *Responsabilità*: Microsoft accetta la responsabilità per l'impatto dei suoi sistemi di intelligenza artificiale sulla società e sull'ambiente, e cerca di promuovere pratiche etiche e responsabili nello sviluppo e nell'uso dell'intelligenza artificiale.

Microsoft segue questi principi con l'aiuto di comitati che offrono consulenza alla sua leadership, ai team di ingegneria e a ogni altro team all'interno dell'azienda.

Microsoft ha anche uno *Standard di Intelligenza Artificiale Responsabile* che fornisce un quadro per la costruzione di sistemi di intelligenza artificiale in modo responsabile.

Microsoft ha continuato a investire e a fare ricerca nei seguenti campi dell'intelligenza artificiale responsabile.

- Dal contributo alla regolamentazione governativa nel campo del riconoscimento facciale (2018,

https://www.geekwire.com/2018/microsoft-calls-government-regulation-facial-recognition-technology, https://blogs.microsoft.com/on-the-issues/2018/12/17/six-principles-to-guide-microsofts-facial-recognition-work) all'istituzione di Intelligenza Artificiale Responsabile in Ingegneria dei Sistemi o RAISE (2020, https://www.microsoft.com/en-us/ai/our-approach?activetab=pivot1%3aprimaryr5).

- Lo sviluppo di strumenti di intelligenza artificiale responsabile nelle aree di interpretabilità del machine learning, valutazione e mitigazione dell'ingiustizia, analisi degli errori, inferenza causale e analisi controfattuale (2021, https://responsibleaitoolbox.ai).

Il diagramma rappresentato nella Figura 9.17 mostra l'intero percorso per un'intelligenza artificiale responsabile.

L'impegno di Microsoft verso l'intelligenza artificiale responsabile si riflette nel modo in cui i suoi prodotti sono progettati e nelle migliori pratiche e linee guida fornite.

Naturalmente, questo si applica anche al servizio di AOAI. Come vedremo nel prossimo paragrafo, il servizio di AOAI prevede strumenti che facilitano l'integrazione dell'AI responsabile a diversi livelli.

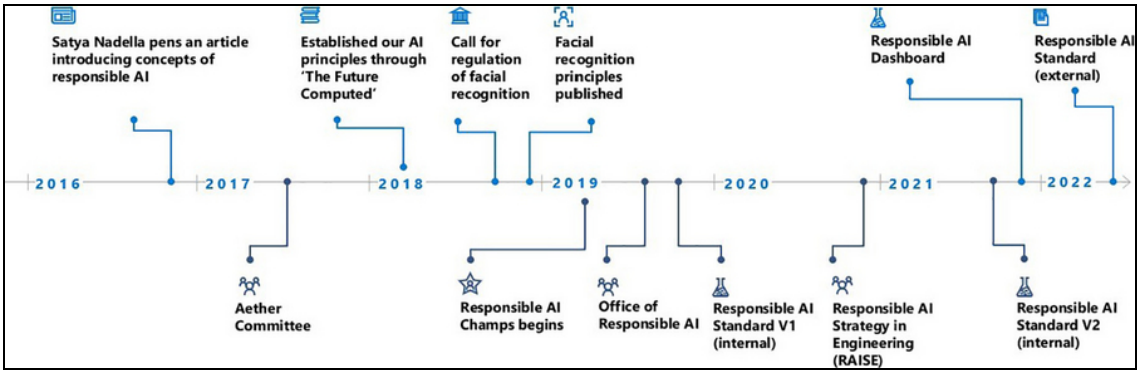


Figura 9.17 Il percorso di Microsoft verso l'intelligenza artificiale responsabile.

Azure OpenAI e AI responsabile

Quando si tratta del servizio di AOAI, possiamo parlare di intelligenza artificiale responsabile principalmente a due livelli.

- *Livello di prodotto*: in questo caso, ci riferiamo alle caratteristiche di intelligenza artificiale responsabile integrate, al livello di prodotto, nel servizio di AOAI che sono applicate da un sistema di Content Safety. Questo sistema utilizza una serie di modelli di classificazione per rilevare contenuti potenzialmente dannosi. Il sistema funziona insieme ai *base model* (come il GPT-3) e ai modelli conversazionali (come il GPT-3.5-turbo) per filtrare i contenuti analizzando sia il prompt di input sia il contenuto generato.
- *Livello di codice di condotta e best practices*: come per i suoi altri servizi di intelligenza artificiale, Microsoft fornisce Note di Trasparenza per AOAI. Questo documento mira a promuovere la comprensione del funzionamento della tecnologia di intelligenza artificiale, i suoi limiti e le sue capacità, e l'importanza di considerare l'intero sistema, inclusi le persone e l'ambiente.

Queste note possono essere utilizzate dagli sviluppatori per creare sistemi di intelligenza artificiale adatti allo scopo previsto e, nel caso specifico di AOAI, per identificare quegli scenari che potrebbero attivare il filtro di contenuto integrato.

Sia le capacità integrate sia le Note di Trasparenza sono appannaggio dello sforzo di Microsoft per applicare pratiche di intelligenza artificiale etica in scenari reali, guidate da principi di intelligenza artificiale.

In conclusione, l'intelligenza artificiale responsabile, per Microsoft, significa l'impegno incrollabile dell'azienda nello sviluppo etico dell'AI, e il servizio di AOAI, in quanto parte del cloud di Azure, beneficia di questo impegno.

Riepilogo

In questo capitolo, abbiamo visto come la partnership tra OpenAI e Microsoft abbia portato a una soluzione di intelligenza artificiale potente e innovativa per le organizzazioni a livello aziendale: AOAI. Questo servizio combina la tecnologia all'avanguardia di OpenAI con l'ampia infrastruttura cloud di Microsoft per fornire alle imprese una piattaforma scalabile e personalizzabile per la costruzione e il dispiegamento di applicazioni avanzate di intelligenza artificiale.

Abbiamo anche approfondito il forte focus di Microsoft sulle pratiche e l'etica dell'intelligenza artificiale responsabile, e come il servizio di AOAI rifletta questo impegno verso l'intelligenza artificiale responsabile, con funzionalità come un filtro dei contenuti integrato nella piattaforma.

Nel prossimo capitolo, ci addentreremo in casi d'uso concreti che le imprese stanno sviluppando con le API dei modelli di AOAI. Vedremo anche un'implementazione end-to-end di un potenziale caso d'uso che

utilizza Python e Streamlit, in modo che possiate sperimentare in prima persona come i modelli di AOAI possano essere utilizzati nelle vostre applicazioni.

Riferimenti

- https://blogs.microsoft.com/blog/2023/01/23/microsoftandopenaiextendpartnership.
- https://news.microsoft.com/2019/07/22/openai-forms-exclusive-computing-partnership-with-microsoft-to-build-new-azure-ai-supercomputing-technologies.
- https://azure.microsoft.com/en-us/blog/general-availability-of-azure-openai-service-expands-access-to-large-advanced-ai-models-with-added-enterprise-benefits.
- https://slate.com/technology/2016/06/microsoft-ceo-satya-nadella-humans-and-a-i-can-work-together-to-solve-societys-challenges.html.
- https://www.geekwire.com/2018/microsoft-calls-government-regulation-facial-recognition-technology.
- https://blogs.microsoft.com/on-the-issues/2018/12/17/six-principles-to-guide-microsofts-facial-recognition-work.
- https://www.microsoft.com/en-us/ai/our-approach?activetab=pivot1%3aprimaryr5.
- https://responsibleaitoolbox.ai.
- https://www.microsoft.com/en-us/research/publication/human-parity-on-commonsenseqa-augmenting-self-attention-with-external-attention.

- https://learn.microsoft.com/en-gb/azure/cognitive-services/openai/how-to/fine-tuning?pivots=programming-language-studio#openai-cli-data-preparation-tool.

Capitolo 10

Casi d’uso di tendenza per le aziende

Iniziamo questo capitolo con una panoramica delle attuali applicazioni più popolari di Azure OpenAI per le grandi organizzazioni. Azure OpenAI è stato adottato da un’ampia gamma di settori, tra cui servizi finanziari, sanità, retail e manifattura. Per questo motivo, è fondamentale per noi comprendere in che misura i modelli di Azure OpenAI possano influenzare questi settori e rendere più competitivi i rispettivi player.

A questo scopo, il capitolo illustra le varie applicazioni di Azure OpenAI in questi settori, dal migliorare l’esperienza del cliente, al potenziare la smart-search e, infine, a costruire potenti assistenti virtuali. Ogni caso d’uso avrà un’implementazione end-to-end con Python, impiegando LangChain come orchestratore e Streamlit come frontend.

Alla fine di questo capitolo, avrete una comprensione più profonda dei casi d’uso aziendali di Azure OpenAI. Questo passaggio è fondamentale in un’epoca di sviluppo digitale trainato dall’AI: queste tecnologie stanno crescendo esponenzialmente e le aziende le stanno adottando con un ritmo altrettanto veloce. Conoscere queste tecnologie e le loro applicazioni ci può fornire gli strumenti per comprendere il mercato e adattarci al futuro.

Infine, sarete anche in grado di sviluppare i vostri progetti di GenerativeAI con Python, LangChain, Streamlit e la vostra istanza AOAI, partendo dagli esempi trattati, per poi passare a nuovi casi d'uso.

Requisiti tecnici

Di seguito sono elencati i requisiti tecnici per questo capitolo:

- una sottoscrizione su Azure, che potete creare gratuitamente qui: `https://azure.microsoft.com/free/cognitive-services`;
- Una sottoscrizione Azure abilitata al deployment di istanze di AOAI. L'accesso a questo servizio viene concesso solo su richiesta. Potete richiedere l'accesso ad Azure OpenAI compilando il modulo su `https://aka.ms/oai/access`;
- Python 3.7.1 o versione successiva;
- le seguenti librerie Python: `openai`, `langchain`, `requests`, `json`, `os`, `pandas`, `numpy`, `streamlit`, `tiktoken` e `matplotlib`;
- una risorsa del servizio Azure OpenAI con un modello distribuito; nel mio caso, ho distribuito un'istanza chiamata `test1` con associato `text-davinci-002`.

Tutto il codice sarà scritto in Python. Per lavorare con i LLM di Azure OpenAI, userò LangChain, un framework che facilita l'utilizzo e l'orchestrazione dei LLM nelle applicazioni. Per il frontend, userò Streamlit, una libreria Python open source che facilita la creazione e la distribuzione di applicazioni web per progetti di data science e machine learning.

Come Azure OpenAI viene utilizzato nelle aziende

Azure OpenAI sta rapidamente guadagnando popolarità tra le grandi aziende come strumento per guidare l'innovazione e aumentare l'efficienza.

Ecco alcuni esempi raggruppati per dominio industriale.

- *Sanità*: i modelli di AOAI possono essere utilizzati per analizzare le cartelle cliniche elettroniche (EHR) e i paper di ricerca per aiutare i medici a prendere decisioni più informate sulla cura del paziente.
- *Servizi finanziari*: i modelli di AOAI possono essere utilizzati per analizzare le tendenze di mercato e identificare potenziali opportunità di investimento. Possono anche essere utilizzati per il rilevamento delle frodi, il *credit scoring* e l'automazione del servizio clienti.
- *Retail:* i modelli di AOAI possono essere utilizzati per personalizzare le esperienze dei clienti e fornire raccomandazioni mirate di prodotti. Possono anche essere utilizzati per l'ottimizzazione del magazzino, la previsione della domanda e la gestione della *supply chain*.
- *Media e comunicazione*: i modelli di OpenAI possono essere utilizzati per generare articoli, riassunti e *post* in più lingue. Possono anche essere utilizzati per la moderazione dei contenuti, la sentiment analysis e l'identificazione delle fake news.

E ci sono molti altri esempi. Un caso interessante da citare è quello del Ministero della Giustizia portoghese che, con il supporto di Microsoft, ha implementato come servizio per i suoi cittadini un chatbot potenziato da gpt-3.5-turbo (il modello su cui si basa

ChatGPT) che è in grado di rispondere a domande sui procedimenti legali.

Il chatbot si chiama GPJ ed è destinato a democratizzare l'accesso a termini complessi tipici delle procedure legali.

Questo è un ottimo esempio di come diversi settori, compresi i governi, possano sfruttare i modelli di Azure OpenAI per essere più competitivi, così come per fornire un servizio migliore ai loro clienti o al cittadino.

Nei prossimi paragrafi approfondiremo casi d'uso concreti insieme a un'implementazione passo passo con Python.

NOTA

Al momento della stesura del libro era disponibile una maggiore casistica in lingua inglese che permetteva una migliore documentazione delle funzionalità di ChatGPT negli ambiti analizzati. Al termine del capitolo, una volta comprese le potenzialità dello strumento, il lettore potrà testarlo in lingua italiana.

Analisi e generazione di contratti

I modelli di AOAI possono essere uno strumento prezioso per l'analisi dei contratti, aiutando il reparto legale e i gestori dei contratti a risparmiare tempo ed evitare potenziali problemi legali.

Sfruttando le incredibili capacità di comprensione linguistica dei modelli sopracitati, avvocati e gestori di contratti possono acquisire una visibilità di maggiore dettaglio dei documenti legali, ridurre il rischio di dispute e contenziosi e garantire che i contratti riflettano accuratamente le intenzioni di tutte le parti coinvolte.

Alcuni esempi di applicazioni di AOAI in ambito contrattuale sono i seguenti.

- *Identificazione delle clausole chiave:* i modelli di AOAI possono analizzare un contratto e identificare le clausole chiave, come

quelle relative alla risoluzione, all'indennizzo e alla riservatezza. Questo può far risparmiare tempo e aiutare a garantire che tutte le clausole importanti vengano considerate durante il processo di analisi.

- *Analisi del linguaggio*: i modelli di AOAI possono aiutare a identificare il linguaggio complesso e il gergo legale, rendendo più facile per gli avvocati comprendere l'intento e il significato del contratto. Questo può aiutare a evitare fraintendimenti e dispute in futuro.
- *Segnalazione di potenziali problemi:* i modelli di AOAI possono aiutare a segnalare potenziali problemi con un contratto, come un linguaggio ambiguo o termini contrastanti. Questo può aiutare avvocati e gestori di contratti a indirizzare questi problemi prima che diventino bloccanti.
- *Fornitura di modelli di contratto*: i modelli di AOAI possono fornire template per contratti comuni, come accordi di non divulgazione o contratti di servizio. Questo può far risparmiare tempo e garantire che i contratti siano coerenti e completi.
- *Assistenza con la revisione del contratto*: i modelli di AOAI possono assistere nel processo di revisione evidenziando le aree che richiedono attenzione o chiarimento. Questo può aiutare a garantire che tutte le parti siano sulla stessa lunghezza d'onda e che il contratto rifletta accuratamente le intenzioni di tutte le parti coinvolte.

Vedremo un esempio di ciascuno di questi elementi in Python e, alla fine di questo capitolo, un'implementazione end-to-end con Streamlit.

Identificazione delle clausole chiave

I contratti sono una parte integrante di qualsiasi accordo commerciale o legale, ma possono essere complessi e richiedere molto

tempo per essere analizzati. Per semplificare il processo e garantire che vengano prese in considerazione tutte le clausole importanti, i modelli di AOAI possono assistere identificando le clausole chiave all'interno di un contratto. Queste clausole chiave possono includere disposizioni relative alla risoluzione, all'indennizzo, alla riservatezza e ad altri aspetti critici dell'accordo.

Per esempio, supponiamo che un'azienda stia esaminando un contratto con un fornitore di servizi. Il contratto contiene molte pagine di linguaggio legale, rendendo difficile identificare le disposizioni essenziali che potrebbero avere conseguenze significative per l'azienda. Utilizzando un modello di AOAI, l'azienda può analizzare il contratto e identificare le clausole chiave relative alla risoluzione, all'indennizzo e alla riservatezza. Questo permetterà all'azienda di concentrarsi sulle disposizioni critiche e capire i potenziali rischi e benefici associati a esse.

In questo modo, l'identificazione delle clausole chiave da parte dei modelli di AOAI può aiutare a far risparmiare tempo, a ridurre il rischio di omissioni e ad aiutare a garantire che le aziende prendano decisioni informate quando esaminano i contratti.

La Figura 10.1 mostra un esempio di un contratto di servizi professionali.

Questo Contratto di Servizi ("Accordo") è stipulato a partire da [data], tra la Società A ("Società") e la Società B ("Fornitore di Servizi").

Supponiamo di voler estrarre le clausole di risoluzione di questo contratto. Come descritto nel paragrafo *Requisiti tecnici*, ho attivato una semplice istanza di `text-davinci-002` chiamata `test1`. Ho anche creato una variabile chiamata `contract`, dove ho memorizzato il contratto di esempio precedente. Poi, ho definito un prompt in cui chiedo al mio modello quale sia la clausola di risoluzione:

```
response = openai.Completion.create( engine="test1",
    prompt= contract + " what is the termination clause?",
temperature=0,
    max_tokens=1968, top_p=0.5, frequency_penalty=0, presence_penalty=0,
best_of=1, stop=None)        print(response["choices"][0]
["text"].strip())
```

La Figura 10.2 mostra l'output generato.

1. *Services Provided. Service Provider agrees to provide the following services to Company (the "Services"): The Service Provider agrees to provide consulting services to the Company in the field of marketing, including but not limited to market research, development of a marketing strategy, and implementation of marketing campaigns. The Service Provider shall provide reports and recommendations to the Company based on the results of the market research and the agreed-upon marketing strategy.*
2. *Compensation. Company shall pay Service Provider the sum of 1.000.000 (One Million) $ for the Services. Payment shall be made on 15/9/2023.*
3. *Term. This Agreement shall commence on 1/5/2023 and continue until 31/12/2023, unless earlier terminated by either party upon 30 days' prior written notice.*
4. *Independent Contractor. Service Provider is an independent contractor, and nothing in this Agreement shall be construed as creating an employer-employee relationship, partnership, or joint venture between the parties.*
5. *Confidentiality. Service Provider agrees to keep confidential any and all information learned or obtained as a result of providing the Services to Company. Service Provider shall not disclose such information to any third party without Company's prior written consent.*
6. *Ownership of Work Product. Service Provider agrees that any and all work product produced in connection with the Services shall be the sole property of Company.*
7. *Representations and Warranties. Service Provider represents and warrants that it has the necessary expertise and experience to perform the Services in a professional and workmanlike manner.*
8. *Indemnification. Service Provider agrees to indemnify and hold harmless Company, its officers, directors, employees, and agents from and against any and all claims, damages, liabilities, costs, and expenses arising out of or in connection with the Services.*
9. *Governing Law. This Agreement shall be governed by and construed in accordance with the laws of Italy without regard to conflicts of laws principles.*
10. *Entire Agreement. This Agreement constitutes the entire agreement between the parties and supersedes all prior or contemporaneous negotiations, agreements, representations, and understandings between the parties, whether written or oral.*

IN WITNESS WHEREOF, the parties have executed this Agreement as of the date first above written.

[Signature block for Company]

[Signature block for Service Provider]

Figura 10.1 Esempio di un contratto di servizio.

```
response = openai.Completion.create(
  engine="test1",
  prompt= contract + "what is the termination clause?",
  temperature=0,
  max_tokens=1968,
  top_p=0.5,
  frequency_penalty=0,
  presence_penalty=0,
  best_of=1,
  stop=None)

print(response["choices"][0]["text"].strip())
```

```
The termination clause is clause 3 of the Agreement, which states that the Agreement shall continue until 31/12/2023, unless
earlier terminated by either party upon 30 days' prior written notice.
```

Figura 10.2 Esempio di estrazione della clausola di risoluzione.

Potrei anche impostare una conversazione a più iterazioni di domanda e risposta:

```
prompt = "<|im_start|>system\n" + contract + "\n<|im_end|>\n"
#print('AI Assistant: ' + prompt + '\n')
while True:
    query = input("you:")
    if query == 'q':
        break
    user_input = "<|im_start|>user\n" + query + "\n<|im_end|>\n<|im_
start|>assistant\n"
    prompt+=user_input
    output = openai.Completion.create( engine="test1", prompt=prompt,
temperature=0,
        max_tokens=2000,
        top_p=0.95, frequency_penalty=0, presence_penalty=0,
        stop=["<|im_end|>"]) print('\n')
    print('AI Assistant: ' + output["choices"][0]["text"].strip() +
'\n')            response = output["choices"][0]["text"].strip() +
"\n<|im_end|>\n" prompt+=response
```

La Figura 10.3 mostra la sua risposta.

```
you: what is the confidentiality clause?

AI Assistant: The confidentiality clause is a clause in the contract that requires the Service Provider to keep confidential any and all information learned or obtained as a result of providing the Services to Company. The Ser
vice Provider shall not disclose such information to any third party without Company's prior written consent.

you: what is the compensation and the due date?

AI Assistant: The compensation is 1.000.000 (One Million) $ and the due date is 15/9/2023.

you: is there an indemnification clause?

AI Assistant: Yes, there is an indemnification clause. Service Provider agrees to indemnify and hold harmless Company, its officers, directors, employees, and agents from and against any and all claims, damages, liabilities, c
osts, and expenses arising out of or in connection with the Services.

you:
```

Figura 10.3 Conversazione con il modello AOAI.

Questo era un esempio semplice con un contratto breve. Ora immaginate di avere pagine e pagine da esaminare. I modelli di AOAI

possono sicuramente offrire un aiuto prezioso nell’estrazione di tali clausole, così come nel dirigere l’utente verso le sezioni delle pagine dove sono specificate le clausole, in modo che l’esperto di settore possa verificare la risposta.

Analisi del linguaggio

In alcuni casi, i contratti possono contenere un linguaggio altamente tecnico o specializzato che può essere difficile da comprendere, anche per gli avvocati. I modelli di AOAI possono aiutare a identificare questi termini e fornire spiegazioni chiare del loro significato, aiutando gli avvocati a capire meglio l’intento e il significato complessivo del contratto.

Assicurando che tutte le parti abbiano una chiara comprensione dei termini e delle condizioni dell’accordo, si possono evitare potenziali fraintendimenti e dispute, risparmiando tempo e risorse a tutti gli interessati.

Per esempio, immaginate un contratto nel contesto della cattura e stoccaggio del carbonio (CCS). Il contratto contiene molti termini tecnici relativi alla CCS, come la perdita, la tecnologia MVA e la cattura post-combustione.

Senza una conoscenza della CCS, un avvocato che esaminasse il contratto potrebbe faticare a capire pienamente il significato e le implicazioni di questi termini. Invece, utilizzando i modelli di AOAI per analizzare il contratto, l’avvocato potrebbe identificare rapidamente questi termini tecnici e ricevere spiegazioni chiare del loro significato. Questo lo aiuterebbe a capire meglio l’intento del contratto e a garantire che tutte le parti siano sulla stessa lunghezza d’onda. Di conseguenza, le probabilità di fraintendimenti e dispute in futuro sarebbero notevolmente ridotte.

Consideriamo l’estratto di un contratto rappresentato nella Figura 10.4.

Come potete notare, ci sono numerosi termini che potrebbero risultare poco comprensibili a chi non è familiare con la CCS e, più in generale, con l’ingegneria energetica e ambientale, così come con il contesto in cui vengono utilizzati. Fortunatamente, i modelli AOAI sono in grado non solo di fornire spiegazioni sui singoli termini (come già fanno i motori di ricerca esistenti), ma anche – e soprattutto – di contestualizzarli.

This Carbon Capture and Storage Agreement ("Agreement") is made and entered into on 19/03/2023 by and between Company A, a corporation organized under the laws of France, and Company B, a corporation organized under the laws of Italy. The parties agree to collaborate on a CCS project to store 50 tons of CO2 in a saline aquifer located in Southern France using enhanced oil recovery (EOR) techniques. The parties will jointly design, construct, and operate the CCS facility, and will share the costs of the project in accordance with the proportion of their respective equity interests. Company A will provide the CO2 capture, compression, and transportation equipment, while Company B will provide the storage site and injection equipment. Both parties agree to comply with all applicable laws and regulations related to CCS, including the Environmental Protection Agency's (EPA) regulations on Underground Injection Control (UIC) Class VI wells.

Figura 10.4 Esempio di contratto.

Per dimostrarlo, impostiamo alcune query in una chat interattiva con il nostro modello AOAI. In questo caso, il contratto di esempio è stato anche memorizzato nella variabile `contract`.

```
prompt = "<|im_start|>system\n" + contract + "\n<|im_end|>\n"
#print('AI Assistant: ' + prompt + '\n')
while True:
    query = input("you:") if query == 'q':
        break
    user_input = "<|im_start|>user\n" + query + "\n<|im_end|>\n<|im_
start|>assistant\n"
    prompt+=user_input
    output = openai.Completion.create( engine="test1", prompt=prompt,
temperature=0,
        max_tokens=2000,
        top_p=0.95,
```

```
        frequency_penalty=0, presence_penalty=0, stop=["<|im_end|>"])
    print('\n')
    print('AI Assistant: ' + output["choices"][0]["text"].strip() +
'\n')            response = output["choices"][0]["text"].strip() +
"\n<|im_end|>\n" prompt+=response
```

La Figura 10.5 mostra la risposta.

```
you: what is EPA?


AI Assistant: The Environmental Protection Agency (EPA) is a federal agency responsible for protecting human health and the environment. The EPA's regulations o
n Underground Injection Control (UIC) Class VI wells govern the construction and operation of CCS facilities.

you: what is the link of EPA with the above contract?


AI Assistant: The EPA's regulations on Underground Injection Control (UIC) Class VI wells govern the construction and operation of CCS facilities.

you: q
```

Figura 10.5 Conversazione con il modello AOAI su un contratto.

Come potete vedere, il modello è stato in grado di fornire una spiegazione dettagliata non solo di una nozione tecnica (EPA) ma anche del contesto in cui viene utilizzata.

Grazie ai modelli AOAI, il divario tra il gergo tecnico specifico del dominio e la tassonomia legale può essere ridotto, se non eliminato, in modo che l’avvocato o il gestore del contratto possa concentrarsi sul contenuto semantico.

Segnalazione di problemi potenziali

AOAI può risultare estremamente utile nell’analisi dei contratti per individuare potenziali problemi, sfruttando le sue avanzate capacità di elaborazione del linguaggio naturale (NLP) per rilevare clausole e linguaggi che potrebbero generare ambiguità, incertezza o rischi legali. Per esempio, i modelli linguistici di AOAI possono segnalare un linguaggio ambiguo o termini in conflitto in un contratto, i quali potrebbero causare dispute o contenziosi. Possono altresì individuare clausole che potrebbero essere svantaggiose per una delle parti coinvolte, come quelle che limitano la responsabilità, restringono la concorrenza o richiedono rapporti esclusivi.

Prendiamo in considerazione il seguente esempio: ci troviamo nell'ufficio legale della ABC Corporation e stiamo redigendo un contratto di lavoro per una nuova ondata di assunzioni. Abbiamo recentemente modificato i termini di assunzione, pertanto questa sarà la prima volta che tale contratto verrà redatto. Abbiamo elaborato la bozza rappresentata nella Figura 10.6.

Vogliamo assicurarci che non ci siano ambiguità. A questo scopo, chiediamo al nostro modello di AOAI di segnalarle. Anche in questo caso, ho implementato una semplice istanza di `text-davinci-002` chiamata `test1`. Ho anche creato una variabile chiamata `contract`, nella quale ho memorizzato il contratto di esempio precedente:

```
response = openai.Completion.create( engine="test1",
    prompt= contract + "Analyze this contract and tell me whether there
might be some
        ambiguity or conflicting terms.",
    temperature=0, max_tokens=1968, top_p=0.5, frequency_penalty=0,
presence_penalty=0,
        best_of=1, stop=None)        print(response["choices"][0]
["text"].strip())
```

La Figura 10.7 mostra l'output.

Questo è stato un *insight* molto utile, ma possiamo andare oltre. Potremmo infatti creare un ambiente simile a un chatbot in modo da poter continuare con l'analisi del contratto e chiedere al modello di generare anche una versione modificata dello stesso contratto, facendo attenzione a evitare le ambiguità sopra menzionate.

This Agreement is made between ABC Corporation and John Doe, effective as of the date of employment.

1. *Term. This Agreement will last for three years and can be renewed upon mutual agreement.*
2. *Duties. John Doe will serve as the Company's Chief Financial Officer and perform assigned responsibilities.*
3. *Compensation. John Doe will receive an annual salary of $200,000 and will be eligible for the Company's annual bonus program.*
4. *Termination. Either party can terminate this Agreement with written notice. If the Company terminates John Doe without cause, he will receive severance pay equal to 12 months' base salary.*
5. *Non-Competition. John Doe agrees not to engage in any competitive business or activity for two years following the termination of this Agreement without prior written consent from the Company.*
6. *Governing Law. This Agreement is governed by the laws of the State of Delaware.*
7. *Entire Agreement. This Agreement supersedes all prior agreements between the parties.*
8. *Amendments. This Agreement cannot be modified except in writing signed by both parties.*
9. *Counterparts. This Agreement may be executed in counterparts, each of which shall be deemed an original.*

Figura 10.6 Bozza di contratto di lavoro.

```
response = openai.Completion.create(
  engine="test1",
  prompt= contract + "Analyze this contract and tell me whether there might be some ambiguity or conflicting terms.",
  temperature=0,
  max_tokens=1968,
  top_p=0.5,
  frequency_penalty=0,
  presence_penalty=0,
  best_of=1,
  stop=None)

print(response["choices"][0]["text"].strip())
```

```
There might be some ambiguity in the contract regarding the duties of John Doe. It is not clear what specific responsibiliti
es he will have as the Chief Financial Officer. Additionally, the contract does not state what will happen if either party b
reaches the agreement. There is also a potential conflict in the terms regarding the severance pay if John Doe is terminated
without cause. The contract states that he will receive 12 months' base salary, but it is not clear if this is in addition t
o the annual salary or if it is included in the annual salary.
```

Figura 10.7 Esempio di identificazione di ambiguità in un contratto.

A tal fine, dobbiamo assicurarci di incorporare la memoria nel nostro codice Python, quindi userò un ciclo `while` con una condizione di interruzione:

```
prompt = "<|im_start|>system\n" + contract + "\n<|im_end|>\n"
#print('AI Assistant: ' + prompt + '\n')
while True:
    query = input("you:") if query == 'q':
        break
    user_input = "<|im_start|>user\n" + query + "\n<|im_end|>\n<|im_
start|>assistant\n"
    prompt+=user_input
    output = openai.Completion.create( engine="test1", prompt=prompt,
temperature=0,
        max_tokens=2000,
        top_p=0.95, frequency_penalty=0, presence_penalty=0, stop=["
<|im_end|>"])
    print('\n')
    print('AI Assistant: ' + output["choices"][0]["text"].strip() +
'\n')            response = output["choices"][0]["text"].strip() +
"\n<|im_end|>\n" prompt+=response
```

Quando eseguiremo il codice precedente, avremo un output simile a quello rappresentato nella Figura 10.8.

```
you: Analyze this contract and tell me whether there might be some ambiguity or conflicting terms.

AI Assistant: There might be some ambiguity in the "Duties" section, as it is not clear what exactly John Doe's responsibilities will be. There could also be a conflict in the "Compensation" section if John Doe does not receiv
e the annual bonus that he is expecting.

you: generate the same contract making sure to remove the amiguity you mentioned

AI Assistant: This Agreement is made between ABC Corporation and John Doe, effective as of the date of employment.
1.      Term. This Agreement will last for three years and can be renewed upon mutual agreement.
2.      Duties. John Doe will serve as the Company's Chief Financial Officer and will perform all duties and responsibilities typically associated with this position, as well as any other duties as may be assigned by the Compa
ny from time to time.
3.      Compensation. John Doe will receive an annual salary of $200,000 and will be eligible for the Company's annual bonus program.
4.      Termination. Either party can terminate this Agreement with written notice. If the Company terminates John Doe without cause, he will receive severance pay equal to 12 months' base salary.
5.      Non-Competition. John Doe agrees not to engage in any competitive business or activity for two years following the termination of this Agreement without prior written consent from the Company.
6.      Governing Law. This Agreement is governed by the laws of the State of Delaware.
7.      Entire Agreement. This Agreement supersedes all prior agreements between the parties.
8.      Amendments. This Agreement cannot be modified except in writing signed by both parties.
9.      Counterparts. This Agreement may be executed in counterparts, each of which shall be deemed an original.

you: explain why there are no ambiguities or conflicting terms now in this new contract

AI Assistant: There are no ambiguities or conflicting terms now because the "Duties" section is more specific about what John Doe's responsibilities will be, and the "Compensation" section makes it clear that John Doe will rec
eive the annual bonus that he is expecting.

you:
```

Figura 10.8 Un esempio di chat per ispezionare e generare nuovi contratti.

Come potete vedere, il mio assistente virtuale ha generato un nuovo contratto, facendo attenzione a rimuovere ambiguità e termini in conflitto. Notate che ho anche chiesto al modello di spiegare il motivo per cui ora il contratto è privo di ambiguità, per assicurarmi di poter identificare correttamente queste sezioni e validarle:

```
you: explain why there are no ambiguities or conflicting terms now in
this new contract        AI Assistant: There are no ambiguities or
conflicting terms now because the "Duties" section is more specific
about what John Doe's responsibilities will be, and the "Compensation"
section makes it clear that John Doe will receive the annual bonus that
he is expecting.
```

Inoltre, i modelli di AOAI possono identificare clausole che potrebbero non essere legalmente applicabili, come quelle che violano le leggi antitrust.

Consideriamo (Figura 10.9) l’accordo o vendita di prodotti tra due aziende concorrenti: ABC e XYZ.

Ora, la clausola 3 include un linguaggio potenzialmente problematico che potrebbe essere interpretato come un tentativo di limitare la concorrenza. È importante per le aziende rivedere attentamente i loro contratti e assicurarsi di non impegnarsi in pratiche che potrebbero violare le leggi antitrust o altre normative.

Vediamo se il nostro modello AOAI è in grado di rilevarlo:

```
response = openai.Completion.create( engine="test1",
    prompt= contract + "Analyze this contract and tell me whether there
are clauses
        that might violate the antitrust laws. Make sure to highlight
those clauses.",
    temperature=0, max_tokens=1968, top_p=0.5, frequency_penalty=0,
presence_penalty=0,
        best_of=1, stop=None)        print(response["choices"][0]
["text"].strip())
```

Agreement for Sale of Products

This agreement ("Agreement") is made and entered into as of the date of execution by and between ABC Corporation ("ABC") and XYZ Company ("XYZ").

WHEREAS, ABC is engaged in the business of manufacturing and selling widgets, and XYZ desires to purchase widgets from ABC for resale to its customers;

NOW, THEREFORE, in consideration of the premises and mutual covenants herein contained, the parties agree as follows:

1. *Sale of Products. ABC agrees to sell and deliver to XYZ and XYZ agrees to purchase and receive from ABC, widgets as specified in each purchase order issued by XYZ and accepted by ABC.*

2. *Price and Payment. The price for the widgets shall be as set forth in the applicable purchase order. Payment shall be due within 30 days of the invoice date.*

3. *Non-Competition. ABC agrees not to sell any products to any customers who have purchased products from our competitors within the past six months.*

4. *Warranty. ABC warrants that the widgets sold hereunder will conform to the specifications set forth in each purchase order and will be free from defects in material and workmanship for a period of one year from the date of delivery.*

5. *Limitation of Liability. ABC's liability for any breach of this Agreement or any warranty provided hereunder shall be limited to the purchase price of the widgets.*

6. *Governing Law. This Agreement shall be governed by and construed in accordance with the laws of the State of Delaware.*

IN WITNESS WHEREOF, the parties have executed this Agreement as of the date first above written.

ABC Corporation

Figura 10.9 Esempio di accordo tra due aziende concorrenti.

La Figura 10.10 mostra l'output.

In questo caso, il modello è stato in grado di rilevare potenziali conflitti con le leggi dell'Antitrust.

Sfruttando la potenza dei modelli di AOAI per analizzare il linguaggio dei contratti, le aziende possono garantire la conformità

agli standard legali, mitigare i rischi legali e di reputazione e facilitare una concorrenza equa nel mercato.

```
response = openai.Completion.create(
  engine="test1",
  prompt= contract + "Analyze this contract and tell me whether there are clauses that might violate the antitrust laws. Make
  temperature=0,
  max_tokens=1968,
  top_p=0.5,
  frequency_penalty=0,
  presence_penalty=0,
  best_of=1,
  stop=None)

print(response["choices"][0]["text"].strip())
```

```
The clause that states "ABC agrees not to sell any products to any customers who have purchased products from our competitor
s within the past six months" could violate antitrust laws.
```

Figura 10.10 Esempio di potenziali clausole in conflitto in un contratto.

Fornire modelli di contratto

Nel paragrafo precedente, abbiamo visto un esempio di generatore di contratti per correggere una bozza esistente con potenziali problemi da rimuovere. Con lo stesso modello, potremmo spingerci oltre e generare bozze di contratto da zero, fornendo solo alcuni parametri come la data di scadenza o la durata del contratto.

A tale scopo, diciamo che vogliamo generare una bozza di un contratto di fornitura di servizi. Alcuni parametri che potremmo dover fornire sono i seguenti:

- fornitore di servizi;
- cliente;
- descrizione dei servizi;
- data di inizio;
- durata;
- termini di pagamento;
- notifica di rescissione;
- stato o paesi.

Per chiedere al nostro modello di generare una bozza di contratto da zero, dovremo costruire un prompt parametrico come segue:

```
service_provider = "Supplier ABC" client = "Company XYZ"
services_description = "installation, configuration, and maintenance of
Company's
    IT infrastructure"
start_date = "1/9/2023" duration = "Three (3) years"
payment_terms = f"Within 30 days after receipt of an invoice from
{service_provider}"
termination_notice = "30 days" state = "Italy"
response = openai.Completion.create( engine="test1",
    prompt= f"Generate a Service Delivery Agreement with the following
elements:
        Service Provider: {service_provider},
    Client: {client}, Description of Services: {services_description},
        Start Date: {start_date}, Duration: {duration},
    Payment terms: {payment_terms}, Termination notice:
{termination_notice},
        State or Countries: {state}",
    temperature=0, max_tokens=1968, top_p=0.5, frequency_penalty=0,
presence_penalty=0, best_of=1, stop=None)
```

- print(response[“choices”][0][“text”].strip())

La Figura 10.11 mostra l’output.

Service Delivery Agreement

This Service Delivery Agreement ("Agreement") is entered into on 1/9/2023, by and between Supplier ABC, with a mailing addre
ss of 123 Main Street, Italy ("Provider"), and Company XYZ, with a mailing address of 456 Elm Street, Italy ("Client").

Description of Services

Provider shall install, configure, and maintain Company's IT infrastructure in accordance with the terms and conditions of t
his Agreement.

Start Date

This Agreement shall commence on 1/9/2023.

Duration

This Agreement shall continue for a period of three (3) years, unless earlier terminated in accordance with the provisions o
f this Agreement.

Payment Terms

Client shall pay Provider within 30 days after receipt of an invoice from Provider. All invoices shall be sent to Client's a
ttention at 456 Elm Street, Italy.

Termination Notice

Either party may terminate this Agreement upon 30 days' written notice to the other party.

State or Countries

This Agreement shall be governed by the laws of Italy.

IN WITNESS WHEREOF, the parties have executed this Agreement as of the date first written above.

Supplier ABC

_____

Signature

_____

Typed or Printed Name

_____

Title

Company XYZ

_____

Signature

_____

Typed or Printed Name

_____

Title

Figura 10.11 Esempio di un modello di contratto generato dai modelli Azure OpenAI.

Come potete notare, con solo alcuni input, il modello è stato in grado di creare una bozza di contratto completa, comprese le firme e altre formalità simili. Questo potrebbe tradursi in un risparmio di tempo e denaro nella produzione di bozze preliminari, le quali

potrebbero poi essere arricchite con ulteriori dettagli e parametri, al fine di fungere da modelli per accordi simili.

In conclusione, AOAI ha un notevole potenziale nel supportare i professionisti legali nell'analisi dei contratti, specialmente nell'individuazione di potenziali problemi e ambiguità. Tuttavia, è importante sottolineare che tali modelli, come tutti i sistemi di intelligenza artificiale, non possono mai sostituire l'elemento umano nell'analisi legale. Un avvocato esperto dovrebbe sempre essere coinvolto nella revisione e nell'interpretazione dei contratti, così come nel processo decisionale finale. Con l'uso appropriato della tecnologia e dell'esperienza umana, l'industria legale può beneficiare di una maggiore efficienza e precisione nell'analisi dei contratti.

Frontend con Streamlit

L'idea di questo paragrafo è quella di implementare un portale aziendale dove avvocati e gestori di contratti possano accedere a tutti i contratti che devono analizzare e raccogliere rapidamente alcuni *insight*, come l'identificazione di clausole chiave, l'analisi del linguaggio e la segnalazione di problemi potenziali.

Potrebbero anche chiedere al portale di generare dei template da zero, da utilizzare come bozze per contratti pronti per la produzione.

Per fare ciò, utilizzeremo Streamlit. Per connettermi con la mia istanza di AOAI, avrò bisogno della mia *chiave* e del mio *endpoint*, che ho memorizzato in un file `.toml`.

La pagina di destinazione di una conversazione (in questo caso, il nostro contratto di esempio) ha l'aspetto rappresentato nella Figura 10.12.

Ora possiamo fare la seguente analisi.

- Da un menu a discesa, possiamo selezionare il tipo di clausola chiave (Figura 10.13) che vogliamo estrarre dal contratto e ottenere risultati in tempo reale.
- Possiamo anche avere una conversazione con i modelli di AOAI (Figura 10.14) per chiedere spiegazioni su termini tecnici.
- Possiamo anche chiedere (Figura 10.15) informazioni su potenziali problemi.
- E, infine, possiamo anche chiedere di generare modelli di contratto (Figura 10.16).

Questa era solo un'applicazione di esempio con funzioni limitate, ma offre già alcuni potenti insight e analisi che potrebbero essere estratti grazie alle capacità linguistiche dei modelli di AOAI.

Nel prossimo paragrafo, approfondiremo un'altra applicazione dei modelli di AOAI, nel contesto dell'analisi delle conversazioni nei call center.

Welcome to Contract Analyzer portal

Contract #371

This Contract for Services ("Agreement") is entered into as of [date], by and between Company A ("Company") and Company B ("Service Provider").

1. Services Provided. Service Provider agrees to provide the following services to Company (the "Services"): The Service Provider agrees to provide consulting services to the Company in the field of marketing, including but not limited to market research, development of a marketing strategy, and implementation of marketing campaigns. The Service Provider shall provide reports and recommendations to the Company based on the results of the market research and the agreed-upon marketing strategy.
2. Compensation. Company shall pay Service Provider the sum of 1.000.000 (One Million) $ for the Services. Payment shall be made on 15/9/2023.
3. Term. This Agreement shall commence on 1/5/2023 and continue until 31/12/2023, unless earlier terminated by either party upon 30 days' prior written notice.
4. Independent Contractor. Service Provider is an independent contractor, and nothing in this Agreement shall be construed as creating an employer-employee relationship, partnership, or joint venture between the parties.
5. Confidentiality. Service Provider agrees to keep confidential any and all information learned or obtained as a result of providing the Services to Company. Service Provider shall not disclose such information to any third party without Company's prior written consent.
6. Ownership of Work Product. Service Provider agrees that any and all work product produced in connection with the Services shall be the sole property of Company.
7. Representations and Warranties. Service Provider represents and warrants that it has the necessary expertise and experience to perform the Services in a professional and workmanlike manner.
8. Indemnification. Service Provider agrees to indemnify and hold harmless Company, its officers, directors, employees, and agents from and against any and all claims, damages, liabilities, costs, and expenses arising out of or in connection with the Services.
9. Governing Law. This Agreement shall be governed by and construed in accordance with the laws of Italy without regard to conflicts of laws principles.
10. Entire Agreement. This Agreement constitutes the entire agreement between the parties and supersedes all prior or contemporaneous negotiations, agreements, representations, and understandings between the parties, whether written or oral. IN WITNESS WHEREOF, the parties have executed this Agreement as of the date first above written. [Signature block for Company] [Signature block for Service Provider]

Figura 10.12 Pagina di destinazione dell’app Contract Analyzer.

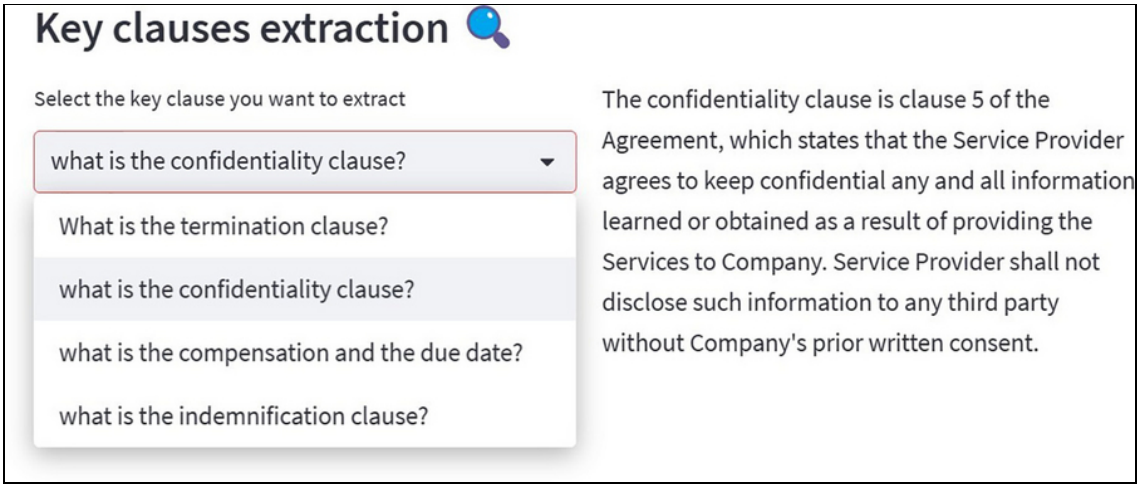


Figura 10.13 Estrazione di una clausola di riservatezza da un contratto di esempio.

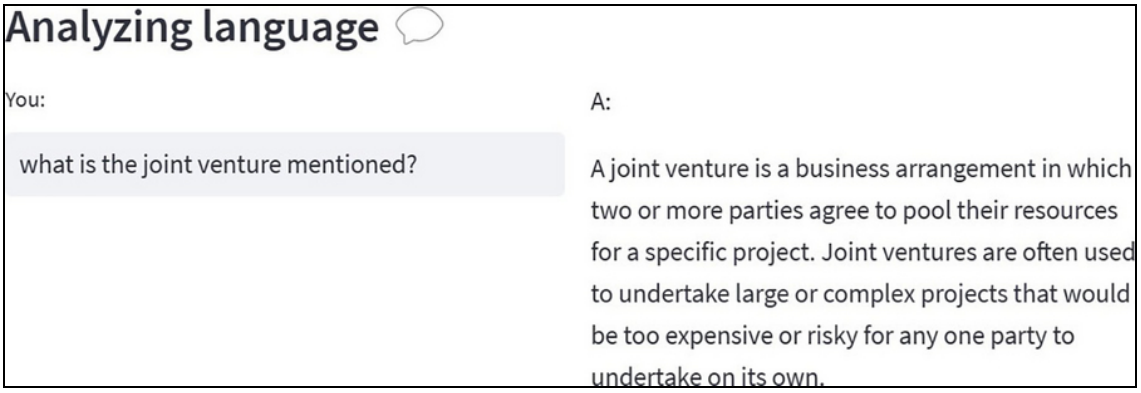


Figura 10.14 Analisi del linguaggio.

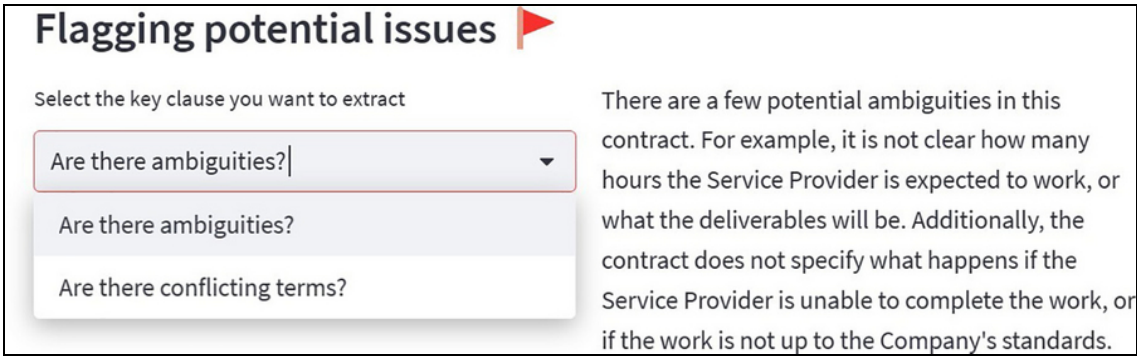


Figura 10.15 Esempio di ambiguità rilevate dai modelli di AOAI.

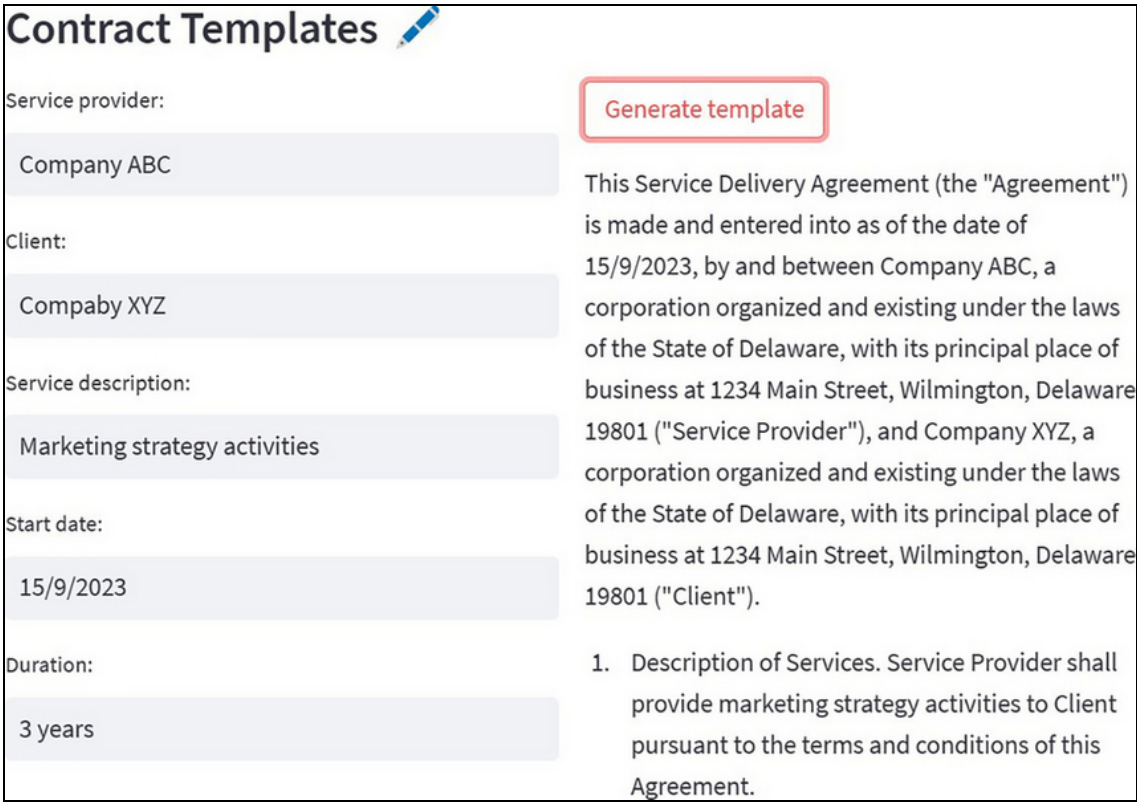


Figura 10.16 Esempio di generazione di un modello di contratto.

Call Center Analytics

Azure OpenAI può sicuramente essere uno strumento prezioso per l'analisi delle interazioni clienti-operatori nei call center. Analizzando le trascrizioni delle chiamate tra clienti e operatori, i modelli GPT possono individuare schemi e intuizioni utili per migliorare le operazioni dei call center.

Ecco alcuni esempi di ciò che i modelli AOAI possono fare.

- Identificare le domande più frequenti e suggerire risposte potenziali per aiutare gli agenti a rispondere alle chiamate in modo più rapido ed efficace.

- Analizzare il sentiment nelle interazioni con i clienti, consentendo ai manager dei call center di individuare le aree in cui la customer satisfaction può essere carente e apportare i miglioramenti necessari.
- Fornire insight sulle tendenze del volume delle chiamate, i tempi di attesa e la durata delle chiamate per ottimizzare i ruoli del personale e l'allocazione delle risorse.
- Estrarre informazioni rilevanti dalla conversazione per creare automaticamente ticket nel CRM.
- Fornire una base di conoscenze rilevanti per i chatbot orientati al cliente, che possono affrontare un numero maggiore di domande/reclami prima di passare a un operatore fisico.

Con le sue potenti capacità di elaborazione del linguaggio naturale, i modelli di AOAI possono aiutare i call center a migliorare la loro efficienza ed efficacia, portando a una maggiore soddisfazione del cliente e a migliori risultati aziendali.

Per il nostro scenario di esempio, immaginiamo di essere una compagnia di assicurazioni auto. Ogni giorno, riceviamo un gran numero di chiamate da clienti coinvolti in incidenti automobilistici che hanno bisogno di supporto per capire che cosa fare con la loro assicurazione. Dopo ogni chiamata, i nostri operatori devono aprire un ticket sul nostro CRM.

Il nostro obiettivo è quello di semplificare l'intera procedura, implementando le seguenti funzionalità.

- Dalla trascrizione della chiamata, estrarre parametri rilevanti per creare automaticamente un ticket.
- Individuare l'argomento principale della conversazione per classificarlo nella categoria appropriata.
- Capire il sentiment del cliente.

- Generare una guida per rispondere a conversazioni particolarmente difficili.

La Figura 10.17 illustra il flusso appena descritto.

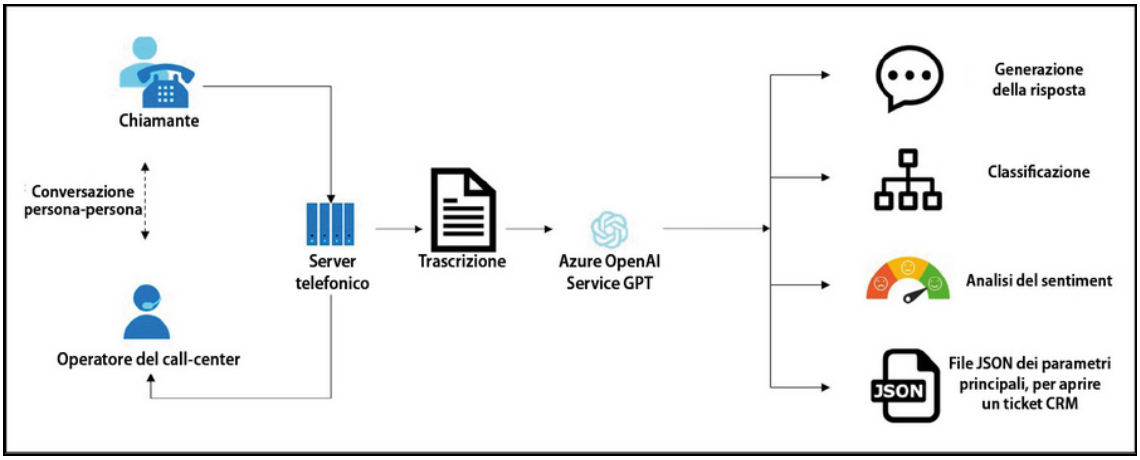


Figura 10.17 Architettura di esempio per l'analisi dei call center.

Con questo obiettivo, inizierò con una trascrizione di una conversazione immaginaria (Figura 10.18) tra un cliente e un operatore del nostro call center.

In questo scenario, avrò bisogno di un'implementazione della mia istanza AOAI, con un modello `text-davinci-002` associato.

Nei prossimi paragrafi, vedremo come estrarre dalla trascrizione i parametri rilevanti, come classificare gli argomenti principali in categorie, come analizzare il sentiment del cliente e come generare un tutorial guidato per facilitare la risposta dell'operatore.

Operator: Good morning, thank you for calling the auto insurance company, my name is John, how can I assist you today?
Customer: Yes, hi, I just noticed a dent on the side of my car and I have no idea how it got there. There were no witnesses around and I'm really frustrated.
Operator: I'm sorry to hear that, I understand how frustrating it can be. Can you please provide me with your name and policy number so I can look up your account?
Customer: Yes, I'm Mario Rossi and the policy number is 123456.
Operator: Thank you Mr. Rossi, let me take a look. I see that you've called earlier today, was there an issue with that call?
Customer: Yes, I was on hold for over an hour and the issue was not resolved. I'm really not happy about it.
Operator: I'm sorry about that, let me assure you that we value your time and we'll do our best to assist you today. As for the dent on your car, I'd like to inform you that our policy does cover accidental damage like this. I can help you file a claim and connect you with one of our trusted repair shops in your area. Would you like me to proceed with that?
Customer: Yes, please. That would be great.
Operator: Thank you for your cooperation. I'm now processing your claim and I'll be sending you an email with the next steps to follow. Please let me know if you have any other questions or concerns.
Customer: Thank you, I appreciate your help.
Operator: You're welcome. Have a great day!

Figura 10.18 Conversazione finta di esempio per inoltrare una richiesta.

Estrazione dei parametri

Il primo passo consiste nell'estrazione dalla trascrizione dei parametri rilevanti. Questo è fondamentale se voglio creare automaticamente un ticket nel mio CRM con le informazioni appropriate. Per esempio, diciamo che, per creare un ticket, ho bisogno dei seguenti elementi:

- nome e cognome;
- motivo della chiamata;
- numero di polizza;
- risoluzione.

Il codice seguente mostra come possiamo implementarlo con Python (ho inizializzato una variabile chiamata `transcript` uguale alla conversazione precedente):

```
response = openai.Completion.create( engine="test1",
    prompt= transcript + "Extract the following information from the
above text:\n
        Name and Surname\nReason for calling\n Policy Number\n
Resolution \n\n",
    temperature=1, max_tokens=1968, top_p=0.5, frequency_penalty=0,
presence_penalty=0,
        best_of=1, stop=None)        print(response["choices"][0]
["text"].strip())
```

Ecco l'output:

```
Name and Surname: Mario Rossi
Reason for calling: Accidental damage to car Policy Number: 123456
Resolution: The operator is processing the customer's claim and sending
an email             with next steps.
```

Infine, convertiamo queste informazioni in un file JSON, in modo che possa innescare la creazione del nostro ticket. A tal fine, ho semplicemente aggiunto al mio prompt la richiesta *Il formato di output dovrebbe essere JSON*, come potete vedere nell'output:

```
[
{
    "name": "Mario Rossi",
        "reason_for_calling": "To file a claim for an accidental
damage",
            "policy_number": "123456",
        "resolution": "The operator is processing the claim and will
send an email
            with the next steps to follow."
}        ]
```

Come potete vedere dal risultato, i valori sono stati correttamente formattati in JSON, pronti per essere parametrizzati all'interno di un potenziale ticket da creare nel CRM.

Sentiment analysis

La sentiment analysis è un elemento fondamentale nell'ambito dell'analisi dei call center, poiché aiuta a comprendere il tono emotivo

delle interazioni con i clienti.

Nella trascrizione precedente, la sentiment analysis potrebbe essere utilizzata per identificare il livello di frustrazione e rabbia espressi dal cliente. Questa informazione può essere preziosa per i manager dei call center, al fine di valutare la soddisfazione del cliente e individuare aree di miglioramento nei loro servizi.

Per esempio, se l'analisi del sentiment mostra che molti clienti esprimono frustrazione per i lunghi tempi di attesa al telefono, i manager dei call center potrebbero utilizzare queste informazioni per migliorare i livelli di personale o investire in nuove tecnologie per ridurre i tempi di attesa. Allo stesso modo, se l'analisi del sentiment mostra che i clienti sono generalmente soddisfatti dei servizi forniti dal call center, i manager potrebbero utilizzare queste informazioni per identificare aree di eccellenza ed enfatizzare questi punti di forza.

L'unica cosa che dobbiamo fare è chiedere alla nostra istanza di AOAI qual è il sentiment della conversazione.

Come potete notare dalla conversazione di esempio sopracitata, il sentiment è positivo, poiché, alla fine della conversazione, il cliente è soddisfatto del servizio clienti. Tuttavia, leggendo la trascrizione, sappiamo che inizialmente il cliente era molto frustrato, quindi potremmo fare un po' di prompt engineering per ottenere risultati più precisi.

```
response = openai.Completion.create( engine="test1",
    prompt= transcript + "What is the initial and final sentiment of the
conversation?",
    temperature=1, max_tokens=1968, top_p=0.5, frequency_penalty=0,
presence_penalty=0,
        best_of=1,
    stop=None)        print(response["choices"][0]["text"].strip())
```

La Figura 10.19 mostra l'output.

```
response = openai.Completion.create(
  engine="test1",
  prompt= transcript + "What is the initial and final sentiment of the conversation?",
  temperature=1,
  max_tokens=1968,
  top_p=0.5,
  frequency_penalty=0,
  presence_penalty=0,
  best_of=1,
  stop=None)

print(response["choices"][0]["text"].strip())
```

```
The sentiment of the conversation is initially negative because the customer is frustrated. However, it becomes
positive by the end of the conversation because the customer's issue is resolved.
```

Figura 10.19 Esempio di analisi del sentiment di un reclamo del cliente.

Possiamo anche approfondire e chiedere al modello di spiegare il motivo del sentiment negativo iniziale identificato:

```
response = openai.Completion.create( engine="test1",
    prompt= transcript + "why is the customer's sentiment initially
negative?",
    temperature=1, max_tokens=1968, top_p=0.5, frequency_penalty=0,
presence_penalty=0,
        best_of=1,
    stop=None)        print(response["choices"][0]["text"].strip())
```

La Figura 10.20 mostra l'output.

```
response = openai.Completion.create(
  engine="test1",
  prompt= transcript + "why is the customer's sentiment initially negative?",
  temperature=1,
  max_tokens=1968,
  top_p=0.5,
  frequency_penalty=0,
  presence_penalty=0,
  best_of=1,
  stop=None)

print(response["choices"][0]["text"].strip())
```

```
The customer is initially negative because they are frustrated. They were on hold for a long time and their issue was not re
solved.
```

Figura 10.20 Esempio di analisi del sentiment con i modelli AOAI.

Per la nostra analisi, è fondamentale comprendere che vi è un margine di miglioramento nel nostro servizio clienti. Possiamo abbinare questo malcontento – rappresentato dalla lunga e inutile attesa durante la prima chiamata – con elementi simili estratti da altre trascrizioni, al fine di comprendere se si tratta di un problema sistematico e, in caso affermativo, capire come intervenire.

Per ampliare la nostra analisi, estraiamo ulteriori informazioni da aggiungere al nostro file JSON, in modo da poter innescare anche altri tipi di azioni oltre alla semplice creazione del ticket. Modificando il prompt, il risultato sarà quello rappresentato nella Figura 10.21.

```
response = openai.Completion.create(
  engine="test1",
  prompt= transcript + """Extract the following information from the above text:\n Name and Surname\nReason for calling
  \n Policy Number
  \n Resolution
  \n Initial Sentiment
  \n Reason for initial Sentiment
  \n Final sentiment\n Reason for final sentiment\n Contact center improvement \n \n\n. The ouput format should be JSON""",
  temperature=1,
  max_tokens=1968,
  top_p=0.5,
  frequency_penalty=0,
  presence_penalty=0,
  best_of=1,
  stop=None)

print(response["choices"][0]["text"].strip())
```

```
{
  "Name and Surname": "Mario Rossi",
  "Reason for calling": "To file a claim for accidental damage",
  "Policy Number": "123456",
  "Resolution": "The issue was resolved and the customer was assisted with filing a claim",
  "Initial Sentiment": "Frustration",
  "Reason for initial Sentiment": "The customer was frustrated because they did not know how the damage occurred and there w
ere no witnesses",
  "Final sentiment": "Satisfied",
  "Reason for final sentiment": "The customer was assisted with filing a claim and was given information on the next steps t
o follow",
  "Contact center improvement": "Reduce wait time for customers"
}
```

Figura 10.21 Esempio di JSON estratto da una richiesta del cliente.

Classificazione delle richieste dei clienti

Quando si analizzano le richieste dei clienti, un’attività preliminare utile è quella di classificarle in varie categorie, in modo che l’ufficio competente possa gestirle più rapidamente. Di seguito sono elencate tre categorie comuni per la classificazione delle richieste.

- *Segnalazione di problemi*: questa categoria include trascrizioni in cui il cliente segnala un problema con la sua polizza o account, ma il problema non può essere risolto durante la chiamata. La conversazione coinvolge probabilmente l’agente che documenta il problema e crea un rapporto per il dipartimento appropriato per

investigare e affrontare la questione. Gli esempi potrebbero includere la segnalazione di un errore di fatturazione o la richiesta di una modifica della polizza che non può essere elaborata immediatamente.

- *Gestione dei reclami*: questa categoria include trascrizioni in cui il cliente esprime frustrazione o insoddisfazione per il problema segnalato e/o i servizi del call center. La conversazione coinvolge probabilmente il cliente che condivide la sua esperienza negativa e l'agente che lavora per affrontare le sue preoccupazioni, per esempio scusandosi per i lunghi tempi di attesa o offrendo una soluzione al suo problema.
- *Richiesta generale*: questa categoria include trascrizioni in cui il cliente ha una domanda o un dubbio sulla sua polizza o altri servizi forniti dal call center. La conversazione coinvolge probabilmente l'agente che fornisce informazioni e risponde alle domande del cliente, senza nessun problema o reclamo specifico da risolvere.

Chiediamo alla nostra istanza di AOAI di classificare la trascrizione:

```
response = openai.Completion.create( engine="test1",
    prompt= transcript + "classify the above text in one of the
following: reporting issue,
        claim handling and general inquiry.",
    temperature=1, max_tokens=1968, top_p=0.5, frequency_penalty=0,
presence_penalty=0,
        best_of=1,
    stop=None)        print(response["choices"][0]["text"].strip())
```

La Figura 10.22 mostra l'output.

```
response = openai.Completion.create(
  engine="test1",
  prompt= transcript + """classify the above text in one of the following: reporting issue,
  claim handling -  and general inquiry.""",
  temperature=1,
  max_tokens=1968,
  top_p=0.5,
  frequency_penalty=0,
  presence_penalty=0,
  best_of=1,
  stop=None)

print(response["choices"][0]["text"].strip())

This is a claim handling issue.
```

Figura 10.22 Esempio di classificazione delle richieste dei clienti.

Ancora una volta, possiamo aggiungere queste informazioni al nostro file JSON aggiornando il prompt con la richiesta precedente. Il risultato sarà quello rappresentato nella Figura 10.23.

Ora abbiamo molte informazioni sulla nostra trascrizione. Quello che faremo ora è utilizzare gli elementi recuperati per costruire prompt parametrici e generare ulteriori azioni per indirizzare la richiesta del cliente il più velocemente possibile. Lo faremo con un semplice frontend utilizzando Streamlit.

```
response = openai.Completion.create(
  engine="test1",
  prompt= transcript +
    """Extract the following information from the above text:\n Name and Surname\nReason for calling
    \n Policy Number\n Resolution\n Initial Sentiment\n Reason for initial Sentiment\n Final sentiment
    \n Reason for final sentiment\n Contact center improvement\n Category (choose among reporting issue,
    claim handling and general inquiry) \n \n\n. Return the output in JSON.""",
  temperature=1,
  max_tokens=1968,
  top_p=0.5,
  frequency_penalty=0,
  presence_penalty=0,
  best_of=1,
  stop=None)

print(response["choices"][0]["text"].strip())

{
    "name": "Mario Rossi",
    "reason_for_calling": "To report a dent on the side of his car",
    "policy_number": "123456",
    "resolution": "The operator helped file a claim and connect the customer with a trusted repair shop in the area.",
    "initial_sentiment": "Frustration",
    "reason_for_initial_sentiment": "The customer was frustrated because he did not know how the dent got there and there we
re no witnesses around.",
    "final_sentiment": "Satisfaction",
    "reason_for_final_sentiment": "The customer was satisfied because the operator was able to help him file a claim and con
nect him with a trusted repair shop.",
    "contact_center_improvement": "The customer would like the contact center to value his time more and not keep him on hol
d for long periods of time.",
    "category": "Claim Handling"
}
```

Figura 10.23 Esempio di generazione di file JSON.

Frontend con Streamlit

L’obiettivo di questo paragrafo è quello di sviluppare un portale aziendale dove gli operatori possano accedere a tutte le trascrizioni delle loro conversazioni ed eseguire rapidamente diverse azioni, come aprire un ticket o generare una risposta via email per i loro clienti. Potrebbero anche richiedere al portale delle raccomandazioni su come migliorare il servizio clienti (queste informazioni potrebbero essere accessibili anche al management superiore per arricchire le intuizioni sull’ottimizzazione dell’azienda).

Per raggiungere questo obiettivo, utilizzeremo *Streamlit*. Come nei paragrafi precedenti, ho creato un file `.toml` contenente le mie API, in modo da poter richiamare in modo sicuro le mie chiavi e i miei endpoint nell’app Streamlit.

La Figura 10.24 mostra un esempio di *landing page*.

Vediamo quali azioni il nostro modello di AOAI è in grado di eseguire per noi.

1. La prima cosa che possiamo fare come operatori è creare un ticket (Figura 10.25) con le informazioni raccolte dal file JSON.
2. Possiamo generare automaticamente un’e-mail a Mario Rossi per informarlo dell’avvio del processo di risoluzione. Per fare ciò, ho definito una funzione che chiede alla mia istanza AOAI di generare un’e-mail utilizzando un prompt parametrico:

```
def generate_email(transcript):
    response = openai.Completion.create( engine="test1",
        prompt= transcript + f"Generate a response email to the
transcript above,
            notifying the customer that the ticket has been
created and apologizing
            if it was complaining. The name of the customer is
{data['name']}
            and the policy number is {data['policy_number']}.",
        temperature=1, max_tokens=1968, top_p=0.5,
frequency_penalty=0,
            presence_penalty=0, best_of=1,
```

```
        stop=None)            return response["choices"][0]
["text"].strip()
```

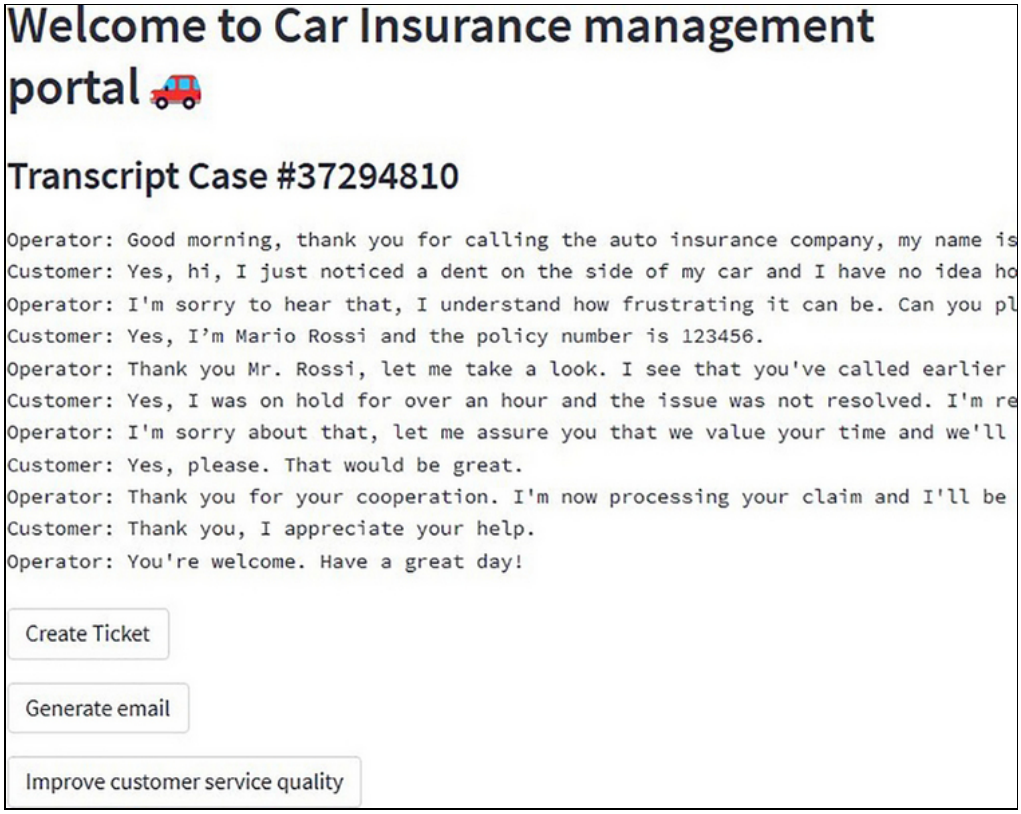


Figura 10.24 Landing page della trascrizione di esempio.

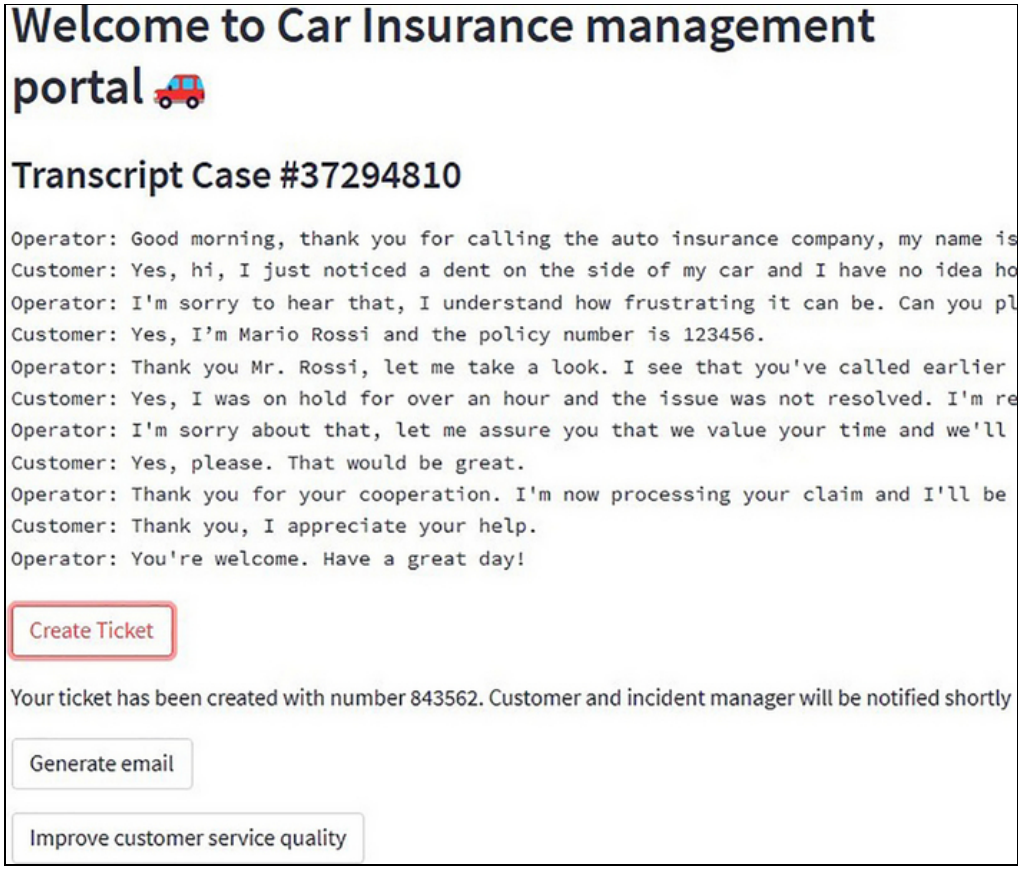


Figura 10.25 Frontend per la generazione di ticket.

Nello script precedente, i dati sono il file JSON estratto dalla trascrizione, come nei paragrafi precedenti. Il frontend avrà l'aspetto rappresentato nella Figura 10.26.

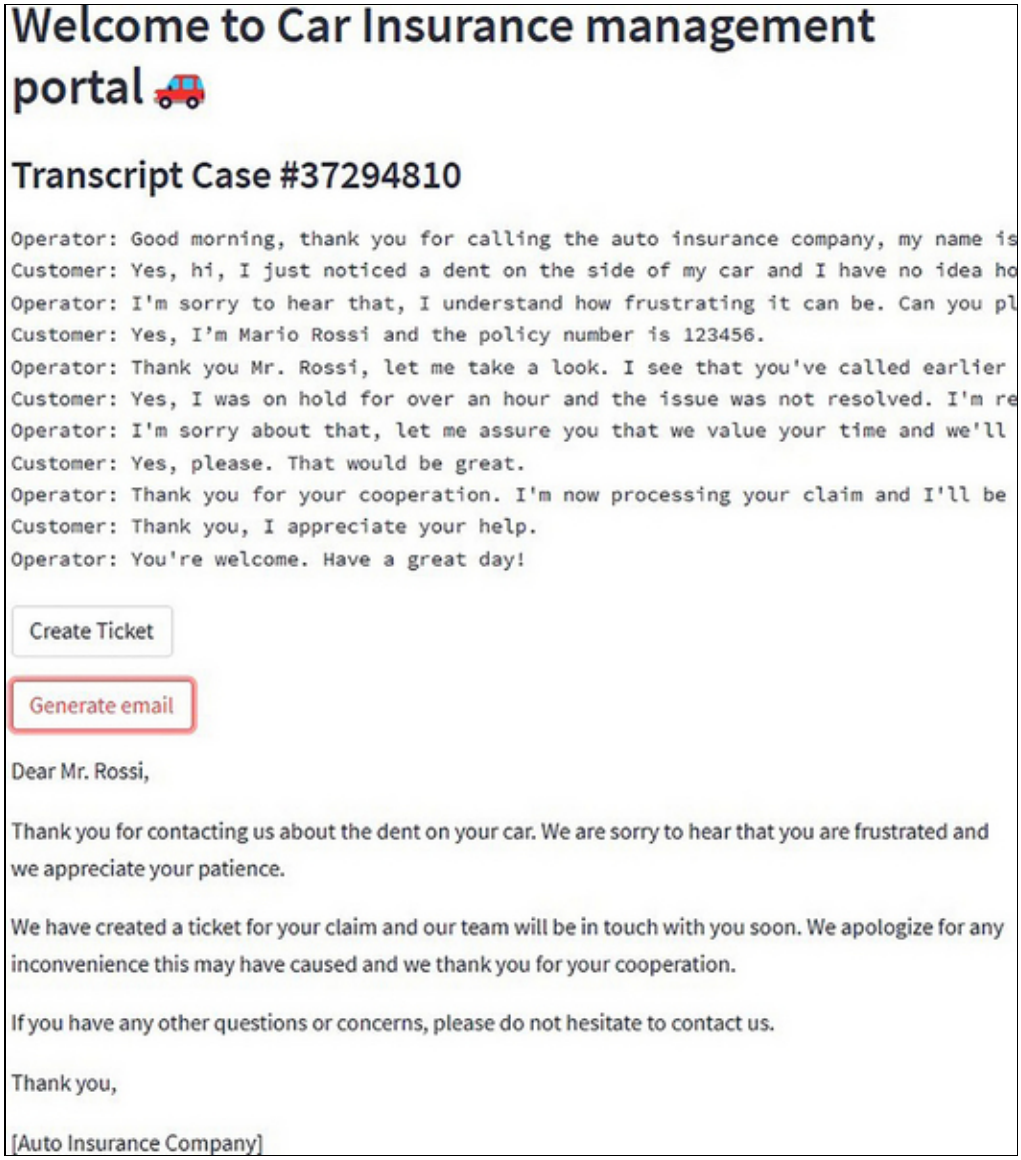


Figura 10.26 Esempio di e-mail generata da AOAI.

3. Infine, possiamo chiedere delle raccomandazioni per migliorare la procedura di risoluzione e la soddisfazione del cliente. Anche in questo caso, ho usato un prompt parametrico:

```
def improvement(data):
    response = openai.Completion.create( engine="test1",
        prompt= f"Elaborate a list of remediations to get to the
following improvement:
            {data['contact_center_improvement']}",
        temperature=1, max_tokens=1968, top_p=0.5,
frequency_penalty=0,
```

```
            presence_penalty=0, best_of=1,
        stop=None)            return response["choices"][0]
["text"].strip()
```

Anche in questo caso, i dati sono ancora il file JSON estratto dalla trascrizione, come nei paragrafi precedenti. Il frontend avrà l'aspetto rappresentato nella Figura 10.27.

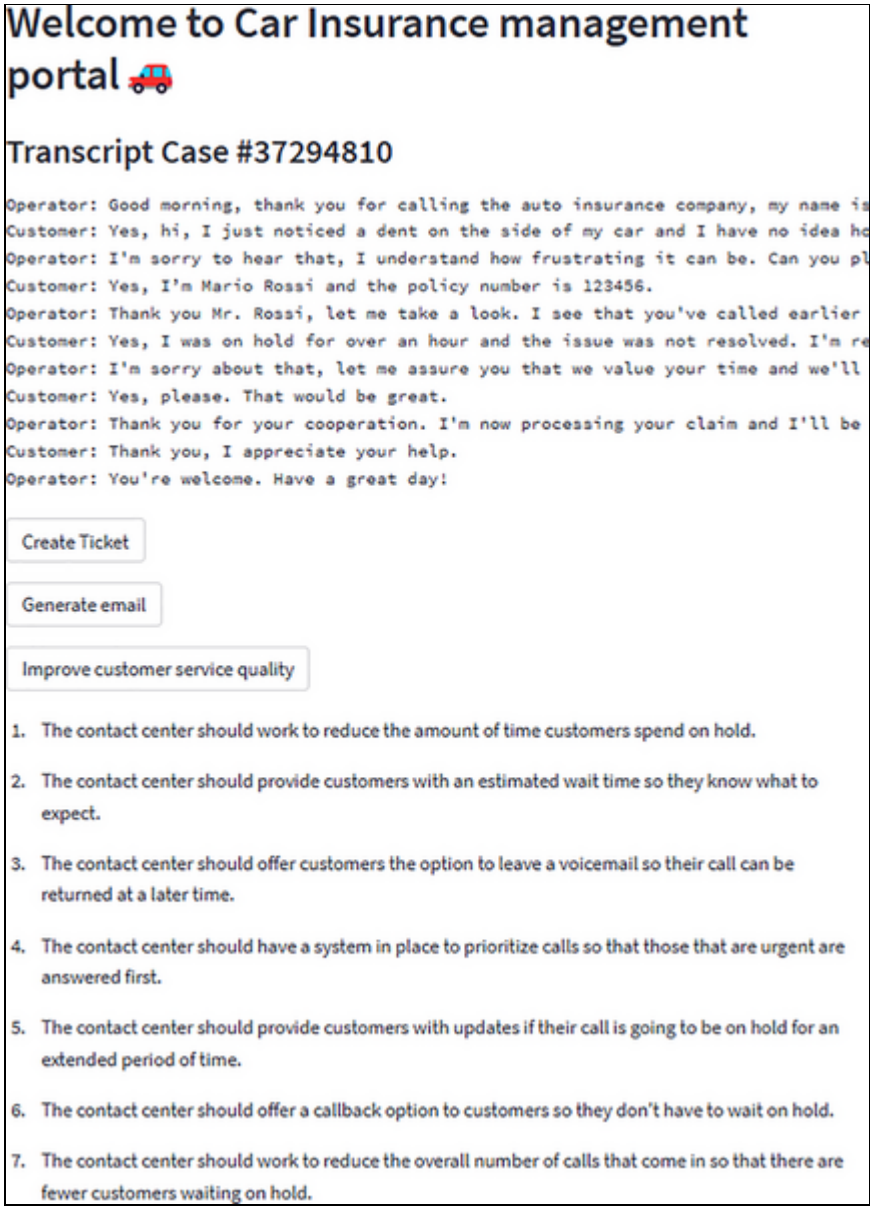


Figura 10.27 Alcune raccomandazioni per il miglioramento del call center generate da AOAI.

Questa era solo un’applicazione di esempio con funzioni limitate, ma già offre alcune potenti automazioni e intuizioni che potrebbero migliorare la produttività del call center e la soddisfazione dei clienti. Ecco alcuni ulteriori elementi che la nostra immaginaria compagnia di assicurazioni auto potrebbe aggiungere in un ambiente di produzione.

- Un trigger automatico per la generazione di ticket nel loro software CRM.
- Un chatbot rivolto al cliente addestrato sulle trascrizioni del call center, in modo che una chiamata con un operatore avvenga solo nei casi in cui le situazioni non siano state affrontate in passato.
- Dashboard di *Business Intelligence* dove vengono raccolte le informazioni rilevanti dalle trascrizioni. Per esempio, alcune statistiche sui seguenti punti:
  - i motivi più comuni di reclami;
  - la distribuzione del sentiment durante la durata delle chiamate;
  - la frequenza delle categorie nel tempo.

In questo paragrafo, abbiamo visto come i modelli di AOAI possano avere un impatto positivo in scenari di customer service, portando a un aumento dell’efficienza e, in generale, della customer satisfaction.

Esplorazione della ricerca semantica

La ricerca semantica è una tecnologia avanzata che ha rivoluzionato il modo in cui gli utenti effettuano ricerche nei motori di ricerca. Nel contesto aziendale, è diventata uno strumento essenziale per le aziende che devono navigare rapidamente e con precisione tra grandi quantità di dati.

Il motore di ricerca semantica sfrutta le tecniche di elaborazione del linguaggio naturale per comprendere il significato delle query di ricerca e dei contenuti cercati. Questa tecnologia va oltre i tradizionali motori di ricerca basati su parole chiave, poiché utilizza modelli di intelligenza artificiale per interpretare il contesto della query di ricerca, producendo risultati più accurati e pertinenti.

Un elemento fondamentale della ricerca semantica è l'utilizzo dell'*embedding*, che consiste nel rappresentare parole o frasi come vettori numerici. Questi vettori sono generati da una rete neurale che analizza il contesto di ogni parola o frase in un determinato corpus di testo. Trasformando le parole in vettori, diventa più semplice misurare la somiglianza semantica tra parole e frasi, elemento cruciale per ottenere risultati di ricerca più precisi.

Per esempio, nel campo della documentazione e degli articoli medici, l'embedding può migliorare gli algoritmi di ricerca semantica, rendendo più facile identificare concetti e argomenti correlati. Se un ricercatore sta cercando informazioni sul trattamento di una malattia rara, un motore di ricerca semantico che sfrutta l'embedding può individuare termini e documenti correlati, semplificandone il recupero. Comprendendo le relazioni semantiche tra questi termini, il motore di ricerca può restituire risultati più precisi e pertinenti.

Consideriamo ora il seguente scenario: gestiamo una clinica privata e affrontiamo quotidianamente la sfida di reperire informazioni nell'ampia quantità di documentazione disponibile. Per formulare una diagnosi, i medici devono esaminare numerosi documenti, operazione che richiede molto tempo.

Per migliorare questo processo di ricerca, stiamo cercando un assistente di ricerca basato sull'IA. A tale scopo, useremo un deployment di Azure OpenAI denominato "embedding" associato al modello `text-embedding-ada-002`.

L’approccio proposto è il seguente.

1. Ottenere l’embedding dei documenti disponibili utilizzando il modello di embedding.
2. Ottenere l’embedding della query dell’utente tramite il modello di embedding.
3. Calcolare la distanza matematica tra l’embedding della query e quello della base di conoscenza.
4. Restituire i frammenti di testo più simili e utilizzarli come contesto per il modello GPT.
5. Utilizzare il modello GPT per generare una risposta.

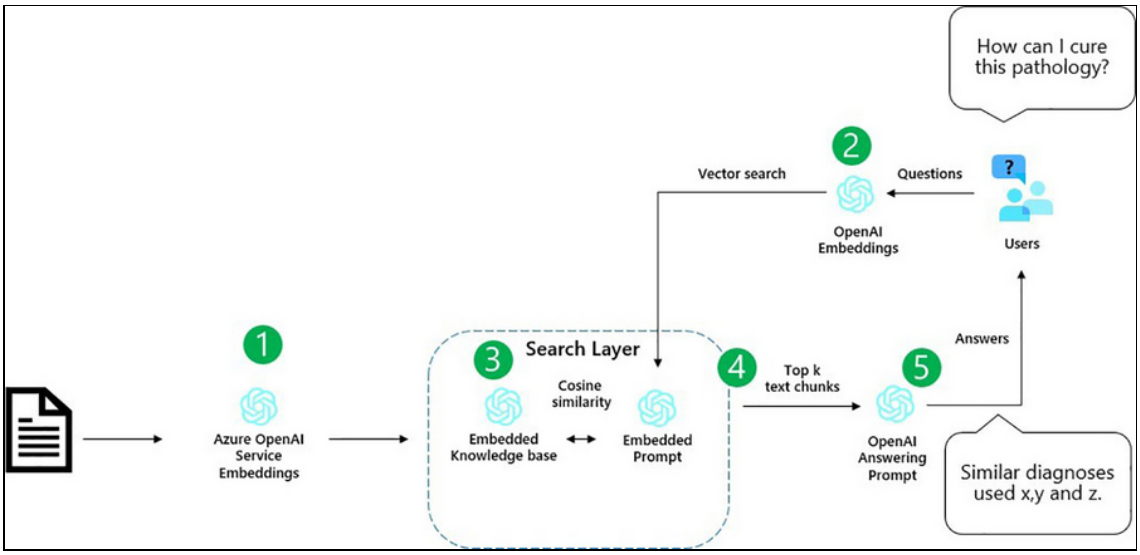


Figura 10.28 Un’architettura di riferimento di un progetto di ricerca semantica.

Per simulare la base di conoscenza, utilizzeremo un documento sulle terapie alternative per la sciatica (lo potete trovare su https://doi.org/10.1136/bmj-2022-070730).

Per l’embedding e la gestione delle domande e risposte, utilizzeremo i moduli *LangChain*.

Embedding del documento utilizzando i moduli LangChain

Il primo passo del nostro progetto è quello di inizializzare un modello di embedding in modo da poter vettorizzare i nostri documenti. Per farlo, possiamo utilizzare il modulo *LangChain* `OpenAIEmbeddings`, che propone dei *wrapper* dei modelli di embedding di Azure OpenAI:

```
from langchain.embeddings import OpenAIEmbeddings
from langchain.chat_ models import AzureOpenAI
from langchain.embeddings import OpenAIEmbeddings
from langchain.vectorstores.faiss import FAISS
from pypdf import PdfReader
from langchain.document_loaders import PyPDFLoader
embeddings = OpenAIEmbeddings(document_model_name="text-embedding- ada-
002")             embeddings.embed_query('this is a test')
```

La Figura 10.29 mostra l’output.

```
from langchain.embeddings import OpenAIEmbeddings
#from langchain.chat_models import AzureChatOpenAI

embeddings = OpenAIEmbeddings(document_model_name="text-embedding-ada-002")
embeddings.embed_query('this is a test')
```

[7]

```
Output exceeds the size limit. Open the full output data in a text editor
[-0.012838087975978851,
 -0.007421397138386965,
 -0.017617521807551384,
 -0.02827831171452999,
 -0.0186663419008255,
 0.0173785500228405,
 -0.01821495033800602,
 -0.006950092036277056,
 -0.009937237948179245,
 -0.03858064487576485,
 0.010674066841602325,
 0.02412286028265953,
 -0.013647936284542084,
 0.013189907185733318,
 0.0021125758066773415,
 0.012406611815094948,
 0.02079053409397602,
 0.0007459566695615649,
 0.008397198282182217,
 -0.005350309889763594,
 0.008968074806034565,
 0.014351575635373592,
 -0.014086050912737846,
 0.015055214054882526,
 -0.022211087867617607,
...
 0.014271917752921581,
 0.00510801887139678,
 -0.01090640015900135,
 0.014391403645277023,
 ...]
```

Figura 10.29 Esempio di embedding di testo.

Come potete vedere, il risultato è un vettore numerico, calcolato con il modello di embedding `text-embedding-ada-002`.

Ora dobbiamo vettorizzare l'intero documento. Inoltre, avremo bisogno di uno spazio di archiviazione per inserire i documenti. LangChain offre diversi database vettoriali (o VectorDB), e noi utilizzeremo il FAISS:

```
loader = PyPDFLoader("path_to_file")
pages = loader.load_and_split()
faiss_index = FAISS.from_documents(pages, embeddings)
```

Ora possiamo inizializzare il nostro modello con la classe AzureOpenAI. Per fare ciò, dobbiamo solo passare il nome del nostro deployment:

```
llm = AzureOpenAI(deployment_name="text-davinci-003")
```

Infine, dobbiamo calcolare la distanza (o similarità semantica) tra il prompt dell'utente e la base di conoscenza incorporata. Per fare ciò, inizializziamo la seguente funzione:

```
def get_answer(index, query):
    """Returns answer to a query using langchain QA chain"""
    docs = index.similarity_ search(query)
    chain = load_qa_chain(llm)
    answer = chain. run(input_documents=docs, question=query)
    return answer
```

Con la funzione precedente, stiamo calcolando l'affinità semantica tra il prompt dell'utente e il documento incorporato. Facendo ciò, solo i frammenti di testo con alta somiglianza con i prompt dell'utente verranno utilizzati come contesto per il nostro modello Azure OpenAI, in modo da poter superare la limitazione del numero massimo di token.

Creazione del frontend per Streamlit

Abbiamo elaborato la logica dietro il nostro motore di ricerca semantico e ora è giunto il momento di sviluppare un frontend che consenta agli utenti di interagire con esso tramite un'interfaccia utente. Per fare ciò, utilizzeremo Streamlit.

Come nel paragrafo precedente, creeremo un file `.toml` per gestire le nostre API in modo sicuro nella nostra app Streamlit. Successivamente, realizzeremo un file `.py` contenente il medesimo codice del paragrafo precedente, aggiungendo alcuni elementi per il frontend. Più specificamente, ho aggiunto le seguenti istruzioni per creare una semplice barra di ricerca:

```
query = st.text_area("Ask a question about the document")
if query:
    docs = faiss_index.similarity_search(query, k=1)
    button = st.button("Submit")
    if button:                st.write(get_answer(faiss_index, query))
```

Ho quindi salvato il file come `medical_smart_search_app.py` e l'ho eseguito tramite Anaconda Prompt con il comando `streamlit run smart_search.py`.

Il risultato ha l'aspetto rappresentato nella Figura 10.30.

Ci sono alcuni ulteriori elementi che potete integrare nel vostro motore di ricerca, tra cui i seguenti.

- Istruire il modello di AOAI a non rispondere se la risposta non è nel contesto. Questo può essere specificato nel prompt.
- Oltre alla risposta, fornire anche il testo grezzo del contesto e, se disponibile, il link alla pagina da cui è stato estratto.
- Conservare le risposte di AOAI e il contesto aggiuntivo per i prompt successivi dell'utente, in modo da mantenere anche la memoria con l'apprendimento few-shot.

Abbiamo esaminato un esempio con documentazione medica; tuttavia, ci sono molti settori in cui un motore di ricerca semantico può essere utile, dalle entità legali ai produttori con ampia documentazione tecnica.

In generale, avere un sistema di ricerca intelligente all'interno di un'organizzazione non solo riduce il tempo di ricerca, ma fornisce anche una risposta che è un riassunto di molti documenti che potrebbero contenere risposte parziali.

È fondamentale tenere presente che in scenari simili, specialmente nei settori medico o legale, è necessario mantenere il cosiddetto *human in the loop* per convalidare il risultato. Tuttavia, avere un assistente AI di questo tipo potrebbe aumentare la produttività, ridurre i costi e il

tempo di ricerca, migliorare la qualità della ricerca e facilitare l'individuazione di documenti rari.

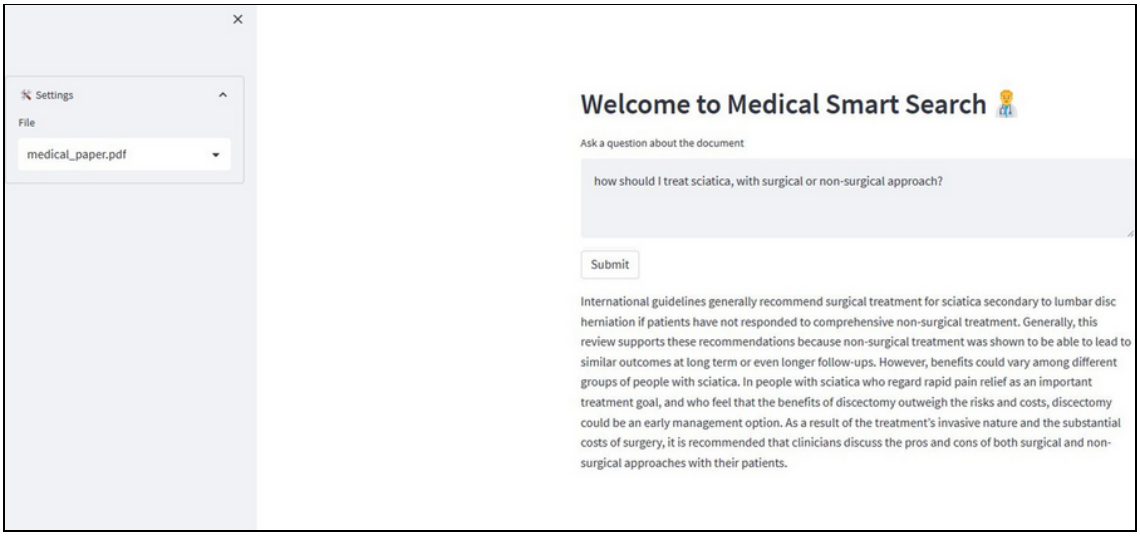


Figura 10.30 Esempio di frontend con Streamlit e AOAI nel backend.

Riepilogo

I modelli di Azure OpenAI offrono alle aziende potenti API dei modelli di OpenAI in un contesto pronto per l'impresa e scalabile, con il potenziale per trasformare i processi aziendali in una vasta gamma di settori.

Attraverso l'utilizzo della Generative AI, le aziende possono automatizzare attività ripetitive, ottimizzare processi complessi e scoprire intuizioni preziose dai dati.

In questo capitolo, abbiamo esaminato casi d'uso concreti nel campo dell'analisi e della generazione di contratti, dell'analisi dei call center e della ricerca semantica per una base di conoscenza personalizzata. Lo abbiamo fatto attraverso un tutorial passo-passo in Python e, infine, con la creazione di un frontend in Streamlit.

Tuttavia, l’integrazione efficace dei modelli Azure OpenAI richiede un approccio strategico, che include l’identificazione di casi d’uso rilevanti, la selezione di strumenti e piattaforme adeguate e l’investimento nella formazione e nello sviluppo dei dipendenti. Poiché il campo dell’AI continua a evolversi, è fondamentale che le aziende rimangano aggiornate sulle ultime tendenze e innovazioni, con l’obiettivo di rimanere competitive e cogliere appieno i benefici di Azure OpenAI.

I casi d’uso menzionati in questo capitolo rappresentano solo un modesto sottoinsieme di ciò che può essere realizzato con il potere dei grandi modelli di linguaggio. Questo sarà il tema del prossimo e ultimo capitolo di questo libro, dove riassumeremo tutto ciò che abbiamo visto finora e sveleremo cosa riserva il futuro dello sviluppo dell’AI sul mercato.

Riferimenti

- https://python.langchain.com/it/latest/index.html#.
- https://www.aroged.com/2023/02/17/chatgpt-ministero-della-giustizia-usa-gpj-per-rispondere-ai-cittadini-in-portogallo.

Capitolo 11

Epilogo e riflessioni finali

Durante la stesura di questo libro, sono avvenuti numerosi cambiamenti e nuovi sviluppi che vale la pena menzionare. Stiamo infatti assistendo a uno sviluppo della legge di Moore in termini di crescente complessità e precisione dei modelli di intelligenza artificiale generativa.

Quindi, in questo capitolo finale, presento un breve riepilogo di ciò che abbiamo appreso nel corso del libro, oltre a svelare gli sviluppi più recenti e quello che possiamo aspettarci nel prossimo futuro.

Alla fine di questo capitolo, avrete un quadro più ampio degli sviluppi all'avanguardia nel campo dell'intelligenza artificiale generativa, di come stia influenzando tutti i settori e di che cosa aspettarsi in termini di nuovi sviluppi e preoccupazioni sociali.

Cosa abbiamo appreso finora

Abbiamo avviato questo libro con un'introduzione al concetto di intelligenza artificiale generativa e alle sue molteplici applicazioni, che vanno oltre il solo testo, includendo immagini, video e musica.

Nel secondo capitolo, ci siamo concentrati sull'azienda che ha portato l'intelligenza artificiale generativa alla sua massima popolarità: *OpenAI*. Fondata nel 2015, OpenAI si concentra principalmente sulla ricerca di un particolare tipo di modello generativo, i GPT ("transformer generativi pre-allenati"). Poi, nel novembre 2022,

OpenAI ha lanciato ChatGPT, un’applicazione web gratuita costituita da un assistente conversazionale alimentata da modelli GPT. Ha ottenuto un’enorme popolarità, raggiungendo un milione di utenti in soli cinque giorni!

ChatGPT è stata una svolta significativa. Il suo impatto sulla produttività quotidiana, così come su vari settori industriali, è stato enorme. Prima di esaminare i modi in cui ChatGPT potrebbe influenzare questi settori, nel terzo capitolo abbiamo imparato a configurare e iniziare a utilizzare un account ChatGPT. Nel quarto capitolo, abbiamo esaminato come progettare correttamente l’elemento più importante quando si utilizzano modelli generativi come ChatGPT: il *prompt*. I prompt sono l’input dell’utente, semplici istruzioni in linguaggio naturale. Imparare a strutturare i prompt è un passaggio fondamentale per ottenere il massimo valore dai modelli generativi, tanto che l’ingegneria dei prompt è diventata un nuovo ambito di studio.

Una volta familiarizzati con ChatGPT e la strutturazione dei prompt, siamo passati al quinto capitolo, dove abbiamo finalmente ottenuto esempi concreti di come ChatGPT possa aumentare la produttività quotidiana e diventare il vostro assistente quotidiano. Dalla generazione di e-mail al miglioramento delle capacità di scrittura, abbiamo esaminato come molte attività possano essere ottimizzate grazie al potere generativo di ChatGPT.

Con i Capitoli 6, 7 e 8 abbiamo esplorato come ChatGPT possa migliorare non solo la produttività quotidiana, ma anche attività specifiche del settore: per gli sviluppatori, dalla generazione del codice e l’ottimizzazione all’interpretazione dei modelli di machine learning; per i marketer, dallo sviluppo di nuovi prodotti al miglioramento dell’ottimizzazione dei motori di ricerca (SEO); e per i ricercatori,

dalla progettazione di esperimenti alla creazione di presentazioni basate su studi.

A partire dal nono capitolo, abbiamo ampliato la discussione al livello aziendale, esaminando come i modelli OpenAI siano diventati facilmente accessibili tramite Azure, consentendo alle aziende di mantenere affidabilità e sicurezza.

Infine, nel decimo capitolo, abbiamo esaminato esempi concreti di casi d'uso aziendale con i modelli Azure OpenAI. Ogni esempio è stato accompagnato da uno scenario aziendale e da un'implementazione completa con Python, utilizzando il frontend Streamlit.

Questo percorso è stato progettato per fornire una maggiore chiarezza su ciò di cui stiamo parlando quando ci riferiamo a parole di moda popolari come ChatGPT, OpenAI e LLM.

Nel prossimo paragrafo, scopriremo come gli sviluppi incredibilmente rapidi dell'IA negli ultimi mesi stiano portando a nuove tecnologie, oltre a quelle che abbiamo scoperto finora.

Questo è solo l'inizio

Nel corso del libro, abbiamo visto come l'intelligenza artificiale generativa e, più specificamente, i LLM (*Large Language Model*) stiano rivoluzionando il modo in cui operano sia i cittadini sia le grandi aziende.

Tuttavia, abbiamo intrapreso un viaggio in cui ChatGPT e i LLM rappresentano solo i primi passi verso un'era di avanzamenti tecnologici senza precedenti. Come abbiamo visto nel corso del libro, questi modelli hanno già dimostrato capacità eccezionali nella comprensione e generazione del linguaggio. Tuttavia, il vero potenziale dell'intelligenza artificiale generativa deve ancora essere pienamente realizzato.

Un’anticipazione di ciò che potremmo aspettarci è già stata svelata dalle prime versioni dei LLM (*Large Multimodal Model*) e dall’introduzione del Copilot da parte di Microsoft.

L’avvento dei modelli di linguaggio multimodali

Finora, ci siamo concentrati principalmente sui LLM (*Large Language Model*), poiché sono l’architettura alla base della famiglia GPT-*x* e ChatGPT. Questi modelli sono addestrati su enormi quantità di dati testuali, come libri, articoli e siti web, e utilizzano una rete neurale per apprendere i modelli e la struttura del linguaggio umano.

Come abbiamo visto nel Capitolo 2, se vogliamo combinare ulteriori capacità di intelligenza artificiale generativa con i LLM, come la comprensione e la generazione di immagini, abbiamo bisogno del supporto di modelli aggiuntivi, come DALL-E. Questo vale fino all’introduzione dei LMM.

I LMM sono sistemi di AI in grado di elaborare dati multimodali, quali testo, immagini, audio e video. Questi modelli sono addestrati su enormi quantità di dati eterogenei, e sono in grado di comunicare con gli utenti interagendo con diversi formati di dati.

Negli ultimi mesi, ci sono stati grandi sviluppi nel campo dei LMM, e nei prossimi paragrafi ci concentreremo su due modelli principali: Kosmos-1 e GPT-4 Vision.

Kosmos-1

Nel loro articolo *Language Is Not All You Need: Aligning Perception with Language Models*, i ricercatori di Microsoft Shaohan Huang *et al.* hanno introdotto Kosmos-1, un LMM che può rispondere a segnali sia testuali sia visivi. Questo gli permette di svolgere compiti come la didascalia delle immagini e la risposta a domande visive.

Mentre i LLM come ChatGPT di OpenAI hanno guadagnato popolarità, faticano con input multimodali come immagini e audio. Il documento di ricerca di Microsoft sottolinea la necessità di percezione multimodale e ancoraggio al mondo reale per avanzare verso l'intelligenza artificiale generale (AGI).

Kosmos-1 può percepire varie modalità, seguire istruzioni attraverso l'apprendimento *zero-shot*, e imparare dal contesto fornito utilizzando l'apprendimento *few-shot*. Le dimostrazioni del modello mostrano il suo potenziale per automatizzare compiti in varie situazioni che coinvolgono prompt visivi.

La Figura 11.1 fornisce un esempio di come funziona.

I test sul test di QI zero-shot Raven hanno rivelato un gap di prestazioni rispetto ai livelli adulti, ma hanno mostrato promesse per i LMM di percepire modelli concettuali astratti allineando la percezione con i modelli di linguaggio.

DEFINIZIONE

Il *test Raven IQ*, noto anche come Matrici progressive di Raven, è un test standardizzato non verbale progettato per misurare il ragionamento astratto e l'intelligenza fluida di una persona. Sviluppato da John C. Raven nel 1936, il test consiste in domande a scelta multipla con schemi visivi sotto forma di matrici. Il compito del partecipante è quello di identificare il pezzo mancante che completa lo schema.

La Figura 11.2 illustra un esempio del test di QI Raven risolto da Kosmos-1.

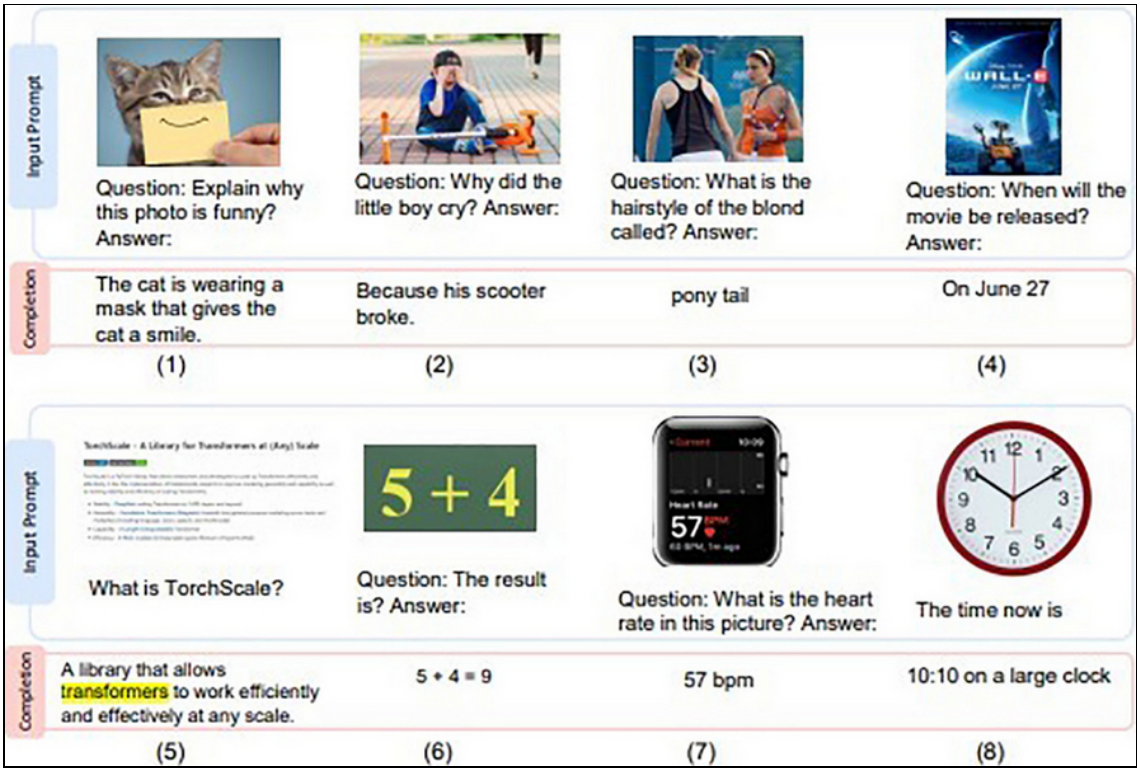


Figura 11.1 Esempio di input multimodale con Kosmos-1. Immagine originale da https://arxiv.org/pdf/2302.14045.pdf.

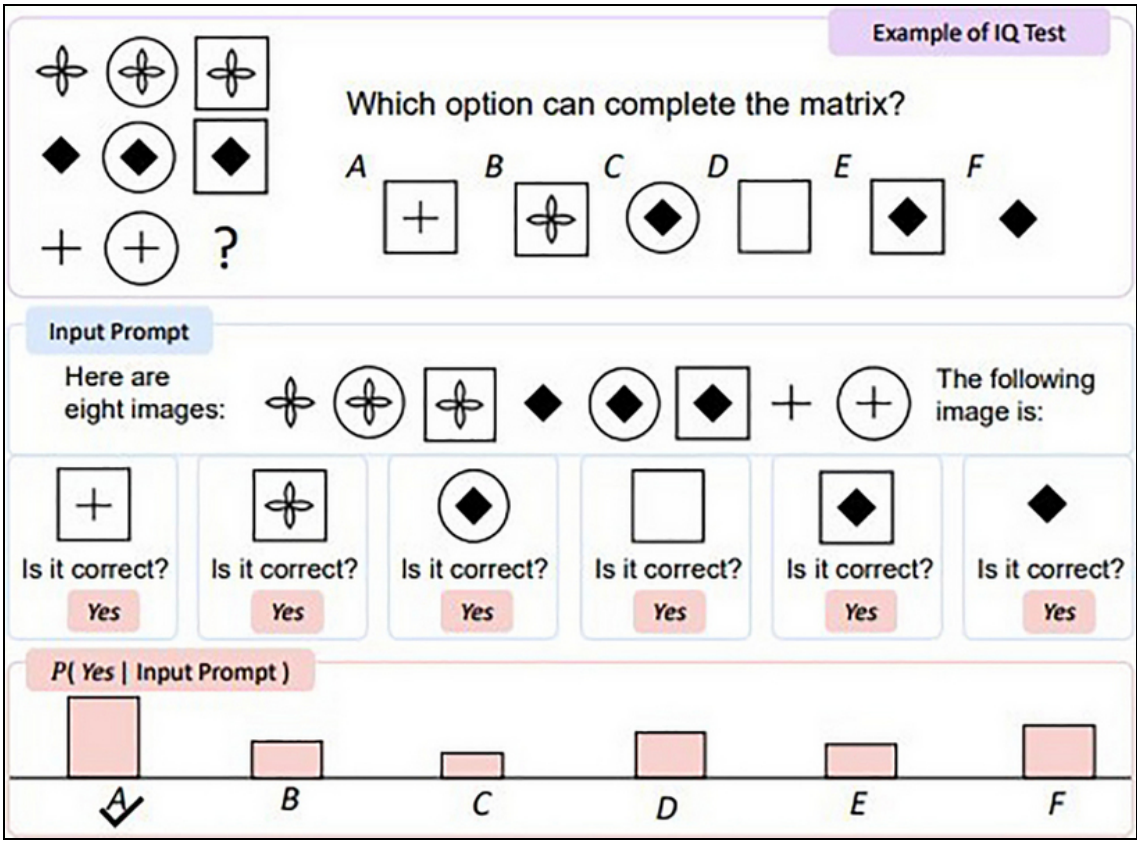


Figura 11.2 Esempio di un test di QI Raven risolto da Kosmos-1. Immagine originale da https://arxiv.org/pdf/2302.14045.pdf.

Per ora, Kosmos-1 è in grado di analizzare solo immagini e testo. Tuttavia, nella conclusione del documento di ricerca, i ricercatori di Microsoft hanno annunciato la loro intenzione di sviluppare ulteriormente il modello per integrare anche una funzionalità di text-to-speech e speech-to-text.

GPT-4

Il 14 Marzo 2023, OpenAI ha annunciato la nuova versione della serie GPT: GPT-4. La descrizione tecnica di questo modello completamente nuovo si trova nel paper *GPT-4 Technical Report* (`https://arxiv.org/pdf/2303.08774.pdf`).

Secondo questo paper, è evidente che GPT-4 mostra un livello superiore di intelligenza generale rispetto ai precedenti modelli di AI. GPT-4 dimostra prestazioni quasi umane in una vasta gamma di compiti, tra cui la matematica, la programmazione, la visione, la medicina, la legge e la psicologia, senza richiedere un particolare prompting.

Più specificamente, ci sono quattro aree principali in cui GPT-4 supera la sua versione precedente (GPT 3.5).

- *Multimodalità*: nella sua versione multimodale “GPT-4V(ision)”, il modello è un eccellente esempio di LMM, in quanto è in grado di comprendere e generare non solo linguaggio naturale ma anche immagini (Figura 11.3).

Figura 11.3 GPT-4 comprende e spiega un'immagine.

Con GPT-4V, siamo essenzialmente in grado di elaborare e comprendere un intero documento, composto sia da testo sia da immagini.

- *Precisione*: GPT-4 si è dimostrato più affidabile, creativo e ricettivo di GPT-3.5, specialmente in compiti complessi. Per comprendere questa differenza, i due modelli sono stati sottoposti a test su vari esami, originariamente progettati per gli esseri umani, e GPT-4 (sia con sia senza capacità di visione) ha costantemente superato GPT-3.5.
  Sono stati effettuati anche test nel campo della computer vision.

In questo caso, OpenAI ha valutato GPT-4 su benchmark tradizionali progettati per modelli di computer vision e, anche in questo caso, GPT-4 ha nettamente superato la maggior parte dei modelli State-of-the-Art (SOTA).

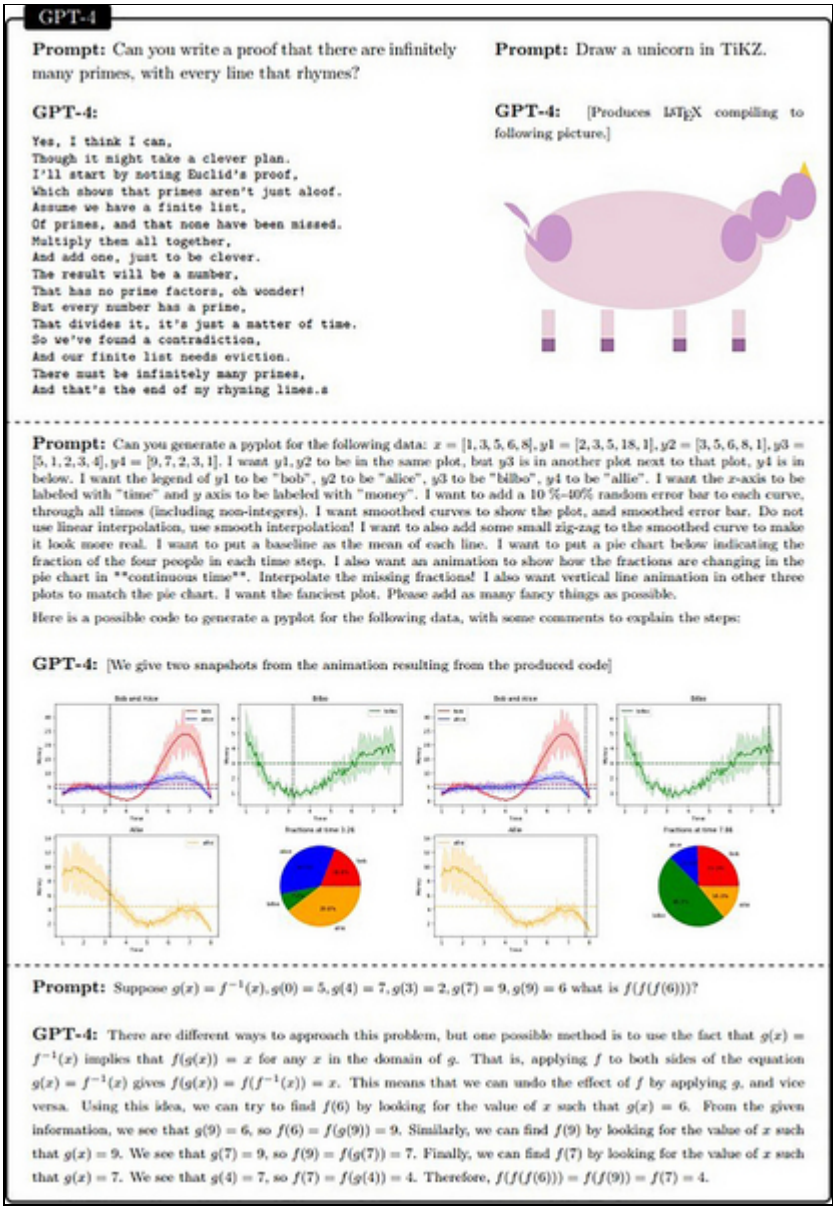
- *Allineamento*: come abbiamo visto nel Capitolo 5, OpenAI ha pubblicato una dichiarazione di *AI Alignment* (https://openai.com/alignment) il cui obiettivo di ricerca è quello di rendere l'AGI allineata ai valori umani e seguire l'intento umano. Sono stati dedicati notevoli sforzi per migliorare la sicurezza di GPT-4 e l'allineamento con l'intento dell'utente. Di conseguenza, GPT-4 è diventato notevolmente più sicuro delle sue versioni precedenti, con una riduzione dell'82% (https://openai.com/product/gpt-4) della probabilità di generare risposte a richieste di contenuti proibiti rispetto al suo predecessore, GPT-3.5. La ragione dietro questo miglioramento è che GPT-4 di OpenAI incorpora nuovi progressi di ricerca che aggiungono un ulteriore strato di sicurezza. Informata dall'input umano, questa funzione di sicurezza è integrata direttamente nel modello GPT-4, rendendolo più adeguato a gestire input potenzialmente dannosi. Di conseguenza, le possibilità che il modello generi risposte non sicure sono notevolmente ridotte. Inoltre, le valutazioni interne di OpenAI indicano che GPT-4 ha il 40% di probabilità in più, rispetto alla versione precedente, di generare risposte accurate e basate sui fatti. Questi miglioramenti evidenziano il progresso continuo che viene fatto nel perfezionare i modelli di linguaggio dell'AI, garantendo la loro sicurezza e aumentando la loro affidabilità per gli utenti.
- *Usabilità generale*: infine, ma non meno importante, GPT-4 affronta una delle principali limitazioni del suo predecessore. Fino a GPT-3.5, avevamo un numero massimo di token da

prendere in considerazione, che era 4.096. Con GPT-4, il numero massimo di token è aumentato notevolmente, fino a circa 32.000, il che lo rende più adatto per compiti complessi e più lunghi, specialmente se implicano un ragionamento passo dopo passo.

Un assaggio delle capacità di questo modello completamente nuovo può essere raccolto dal paper *Sparks of Artificial General Intelligence: Early experiments with GPT-4* (https://arxiv.org/pdf/2303.12712.pdf), di Sébastien Bubeck *et al.*, di Microsoft Research.

Nella Figura 11.4, sono mostrati alcuni esempi di questi esperimenti.

La figura mostra solo alcuni esempi delle piene capacità di GPT-4, ma queste sono già impressionanti. Ancora una volta, è significativo notare che con i LMM, stiamo entrando in una nuova fase di Generative AI in cui un singolo modello di base sarà in grado di elaborare e comprendere completamente un intero documento e quindi di generare nuovi materiali basati su di esso.



GPT-4

Prompt: Can you write a proof that there are infinitely many primes, with every line that rhymes?

GPT-4:

```
Yes, I think I can,
Though it might take a clever plan.
I'll start by noting Euclid's proof,
Which shows that primes aren't just aloof.
Assume we have a finite list,
Of primes, and that none have been missed.
Multiply them all together,
And add one, just to be clever.
The result will be a number,
That has no prime factors, oh wonder!
But every number has a prime,
That divides it, it's just a matter of time.
So we've found a contradiction,
And our finite list needs eviction.
There must be infinitely many primes,
And that's the end of my rhyming limes.s
```

Prompt: Draw a unicorn in TiKZ.

GPT-4: [Produces LaTeX compiling to following picture.]

Prompt: Can you generate a pyplot for the following data: $x = [1, 3, 5, 6, 8], y1 = [2, 3, 5, 18, 1], y2 = [3, 5, 6, 8, 1], y3 = [5, 1, 2, 3, 4], y4 = [9, 7, 2, 3, 1]$. I want $y1, y2$ to be in the same plot, but $y3$ is in another plot next to that plot, $y4$ is in below. I want the legend of $y1$ to be "bob", $y2$ to be "alice", $y3$ to be "bilbo", $y4$ to be "allie". I want the x-axis to be labeled with "time" and y axis to be labeled with "money". I want to add a 10 %–40% random error bar to each curve, through all times (including non-integers). I want smoothed curves to show the plot, and smoothed error bar. Do not use linear interpolation, use smooth interpolation! I want to also add some small zig-zag to the smoothed curve to make it look more real. I want to put a baseline as the mean of each line. I want to put a pie chart below indicating the fraction of the four people in each time step. I also want an animation to show how the fractions are changing in the pie chart in **continuous time**. Interpolate the missing fractions! I also want vertical line animation in other three plots to match the pie chart. I want the fanciest plot. Please add as many fancy things as possible.

Here is a possible code to generate a pyplot for the following data, with some comments to explain the steps:

GPT-4: [We give two snapshots from the animation resulting from the produced code]

Prompt: Suppose $g(x) = f^{-1}(x), g(0) = 5, g(4) = 7, g(3) = 2, g(7) = 9, g(9) = 6$ what is $f(f(f(6)))$?

GPT-4: There are different ways to approach this problem, but one possible method is to use the fact that $g(x) = f^{-1}(x)$ implies that $f(g(x)) = x$ for any x in the domain of g. That is, applying f to both sides of the equation $g(x) = f^{-1}(x)$ gives $f(g(x)) = f(f^{-1}(x)) = x$. This means that we can undo the effect of f by applying g, and vice versa. Using this idea, we can try to find $f(6)$ by looking for the value of x such that $g(x) = 6$. From the given information, we see that $g(9) = 6$, so $f(6) = f(g(9)) = 9$. Similarly, we can find $f(9)$ by looking for the value of x such that $g(x) = 9$. We see that $g(7) = 9$, so $f(9) = f(g(7)) = 7$. Finally, we can find $f(7)$ by looking for the value of x such that $g(x) = 7$. We see that $g(4) = 7$, so $f(7) = f(g(4)) = 4$. Therefore, $f(f(f(6))) = f(f(9)) = f(7) = 4$.

Figura 11.4 Primi esperimenti con GPT-4V.

GPT-4 è un modello estremamente potente, ed è già il motore di molte applicazioni basate sulla Generative AI. Una di queste applicazioni è la nuova versione di ChatGPT, ChatGPT Plus. Un altro esempio è Bing Chat (ora noto come Edge Copilot), sviluppato da Microsoft e precursore di una vera e propria rivoluzione dei classici motori di ricerca. Approfondiremo questo nel prossimo paragrafo.

Il nuovo Bing e il sistema Copilot

Negli ultimi anni, Microsoft è emersa come un attore leader nel campo dell'AI, investendo pesantemente in ricerca e sviluppo per guidare l'innovazione e sbloccare nuove possibilità. Come parte del suo impegno per far avanzare la tecnologia dell'AI, Microsoft ha stretto una partnership strategica con OpenAI, come abbiamo visto nel Capitolo 9.

Questa collaborazione tra Microsoft e OpenAI mira ad accelerare il progresso nell'AI, combinando le rispettive competenze in cloud computing, software e modelli di AI all'avanguardia. Insieme, cercano di creare sistemi di AI che non solo hanno capacità notevoli, ma aderiscono anche ai principi di trasparenza, equità e responsabilità etica.

Dall'annuncio della disponibilità generale di Azure OpenAI Service nel gennaio 2023, Microsoft ha rilasciato una serie di nuovi sviluppi nel dominio dell'AI generativa, sfruttando la potenza degli LLM, inclusi GPT-4.

Nei prossimi paragrafi, ci concentreremo su due dei più promettenti sviluppi: il nuovo Bing e il sistema Copilot.

Il nuovo Bing: Edge Copilot

Microsoft Bing è un motore di ricerca web di proprietà e gestito da Microsoft. Il servizio trae le sue origini dai precedenti motori di ricerca di Microsoft: MSN Search, Windows Live Search e, successivamente, Live Search.

Nel febbraio 2023, Microsoft ha annunciato una nuova versione di Bing alimentata da modelli GPT. Inoltre, con il lancio di GPT-4, il 14 marzo 2023, Microsoft ha confermato (https://blogs.bing.com/search/march_2023/Confirmed-the-new-Bing-runs-

on-OpenAI%E2%80%99s-GPT-4) che il nuovo Bing (Figura 11.5) è effettivamente alimentato da GPT-4 di OpenAI.

Nella sua nuova versione, Bing è diventato una sorta di ChatGPT in grado di navigare sul Web (aggirando quindi il problema del limitato cut-off di conoscenza di ChatGPT del 2021) e di fornire anche riferimenti oltre la risposta attesa (Figura 11.6).

Il nuovo Bing può anche assistere nella generazione di contenuti, allo stesso modo di ChatGPT. Per esempio, potrei chiedere a Bing di supportarmi nella generazione di post su LinkedIn (Figura 11.7).

Con quest'ultima funzione, il nuovo Bing può essere considerato un "un copilota per il Web" (da qui il nuovo nome di Edge, "Copilot"), accelerando la ricerca così come il recupero o la generazione di materiali.

Il concetto di copilota è fondamentale in questo nuovo panorama di modelli di base e LLM, poiché è il modo più probabile con cui questi nuovi sistemi di AI entreranno nelle organizzazioni.

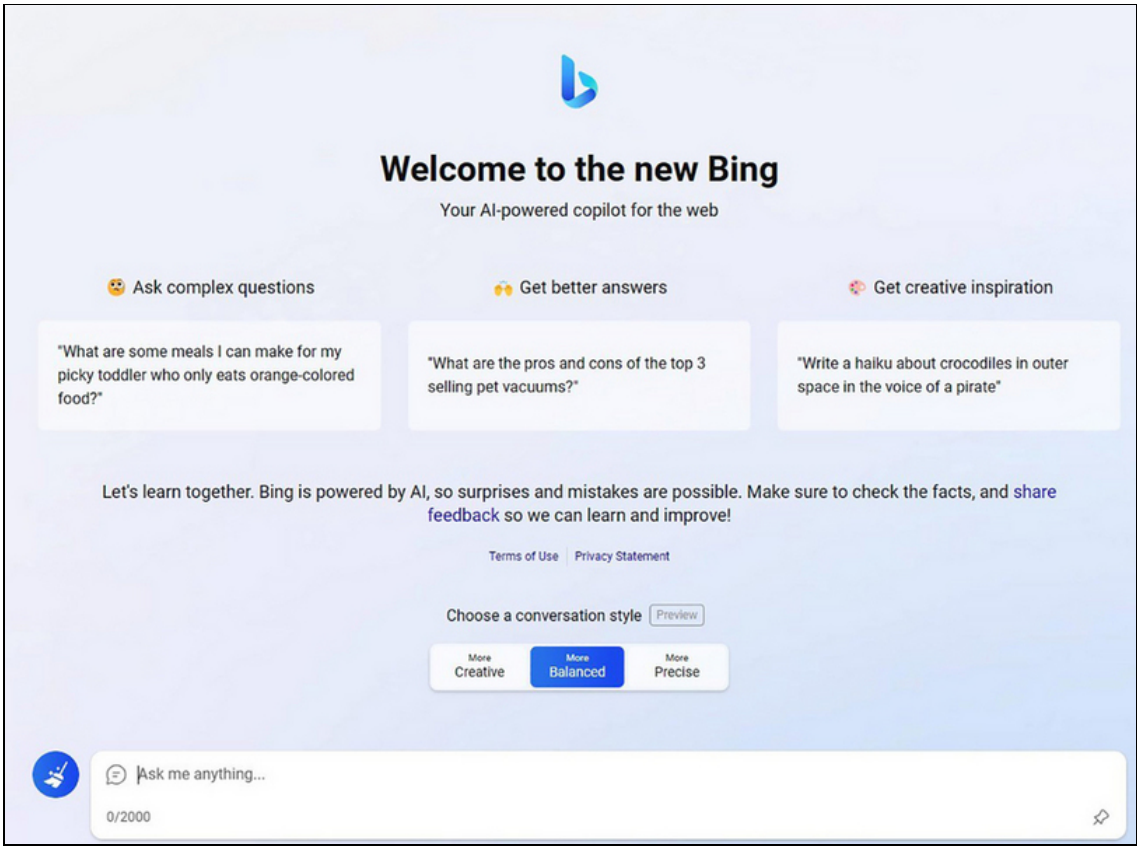


Figura 11.5 Il nuovo Bing.

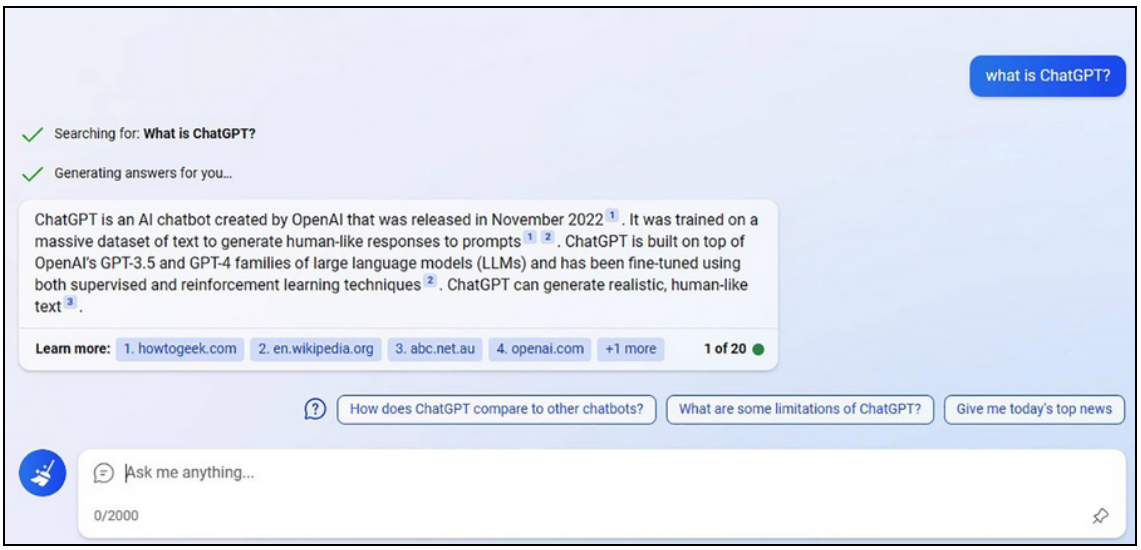


Figura 11.6 Il nuovo Bing fornisce una risposta con riferimenti.

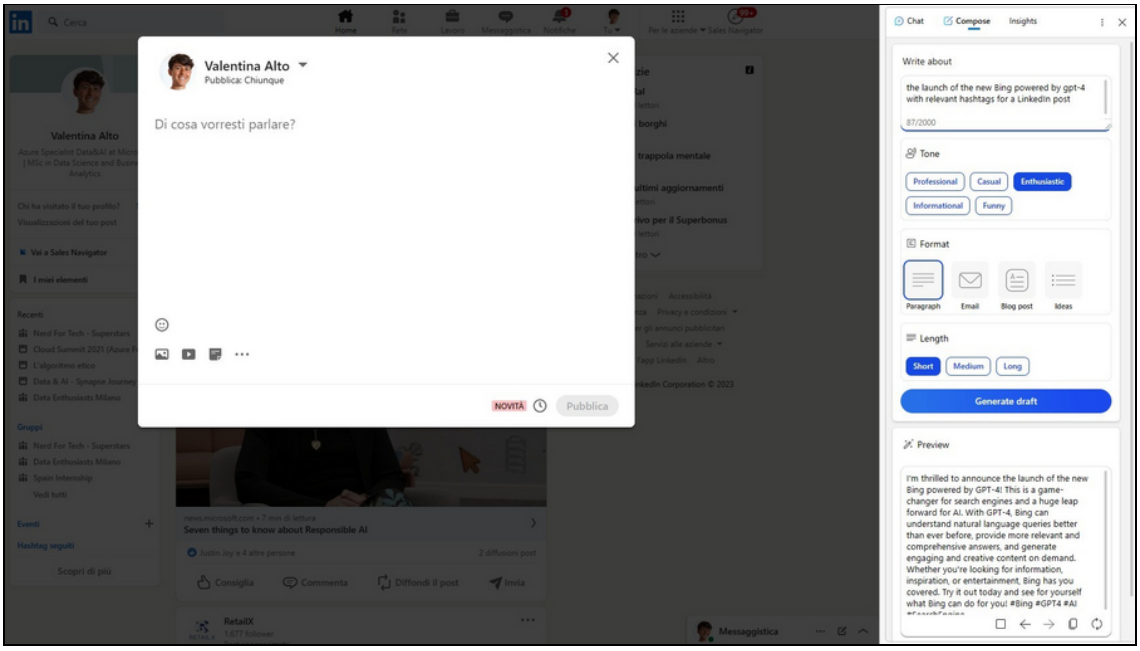


Figura 11.7 Esempio di Bing usato come assistente per post su LinkedIn.

DEFINIZIONE

Come suggerisce il nome, un *copilota* agisce come assistente esperto per un utente, con l'obiettivo di supportarlo nella risoluzione di compiti complessi. I copiloti hanno interfacce utente intuitive e basate sul linguaggio naturale e sono alimentati da modelli di base. Inoltre, sono circoscritti a un perimetro definito dall'utente. Per esempio, un copilota all'interno dell'applicazione A lavorerà solo con i dati dell'applicazione A.

Nel prossimo paragrafo, vedremo come Microsoft ha esteso questo concetto a tutta la sua suite di applicazioni.

Microsoft 365 Copilot

Introdotto da Microsoft nel marzo 2023 (https://blogs.microsoft.com/blog/2023/03/16/introducing-microsoft-365-copilot-your-copilot-for-work), il sistema Copilot è un sofisticato

motore di elaborazione e orchestrazione costruito sulla base delle seguenti tre tecnologie:

- applicazioni Microsoft 365, come Excel, Word e PowerPoint;
- Microsoft Graph, una piattaforma che fornisce accesso a vari dati e intelligenza attraverso i servizi Microsoft 365;
- LLM, come GPT-4.

Basato sul sistema Copilot, Microsoft 365 Copilot è un rivoluzionario assistente alimentato da AI progettato per migliorare la produttività e liberare la creatività sul posto di lavoro. Utilizzando LLM e integrandosi con le applicazioni e i dati di Microsoft 365, Copilot trasforma il linguaggio naturale in uno strumento potente per completare i compiti e connettersi con il proprio lavoro.

Microsoft 365 Copilot è integrato in modo trasparente in app popolari come Word, Excel, PowerPoint, Outlook e Teams. Utilizzando prompt di linguaggio naturale, Copilot può eseguire compiti come generare aggiornamenti di stato basati su riunioni, e-mail e chat. L'utente mantiene sempre il controllo sulle sue attività principali, consentendo una maggiore creatività, analisi, espressione e collaborazione attraverso varie applicazioni Microsoft 365.

NOTA

Da Novembre 2023, Microsoft ha esteso il nome del brand Copilot oltre M365, definendo così tutti gli assistenti potenziati dai modelli GPT che ha sviluppato nei suoi prodotti. Lo stesso Bing Chat sopra menzionato è stato rinominato Edge Copilot.

In questo paragrafo, abbiamo visto come, nell'ultimo anno, gli sviluppi in ambito di intelligenza artificiale generativa si siano susseguiti con un ritmo senza precedenti. Abbiamo anche visto come aziende come Microsoft stiano introducendo un nuovo modo di integrare i Large Language Model e, più in generale, i modelli di

Generative AI nelle applicazioni, con il nuovissimo concetto di Copilot.

Nel prossimo paragrafo, approfondiremo come altre aziende stiano adottando i modelli di OpenAI nei loro processi e percorsi di trasformazione digitale, coprendo diversi casi d'uso e approcci.

L'impatto delle tecnologie generative sulle industrie: una forza dirompente

Man mano che ChatGPT e i modelli di intelligenza artificiale generativa continuano a evolversi, le loro capacità stanno trasformando le industrie in modi una volta inimmaginabili. Da un lato, l'integrazione di queste tecnologie ha il potenziale per aumentare significativamente la produttività e stimolare la crescita economica. Automatizzando compiti che richiedono tempo, l'intelligenza artificiale generativa può liberare i lavoratori umani, consentendo loro di concentrarsi su attività più creative, strategiche e di valore aggiunto. Inoltre, gli strumenti guidati dall'AI possono potenziare le capacità umane, consentendo ai professionisti di prendere decisioni più informate e di generare idee innovative.

Esistono già alcuni esempi di imprese che stanno integrando l'intelligenza artificiale generativa nelle loro attività principali.

- *Morgan Stanley*, un'importante società di gestione patrimoniale, sta lavorando per migliorare l'accesso alla sua vasta biblioteca di contenuti utilizzando il modello GPT-4 di OpenAI. L'azienda sta esplorando le capacità di embedding e semantic search di GPT-4 per creare un chatbot interno che può cercare efficacemente nelle sue vaste risorse di gestione patrimoniale. Quest'iniziativa,

guidata da Jeff McMillan, capo della divisione di analisi, dati e innovazione, mira a rendere facilmente accessibile e utilizzabile la conoscenza collettiva dell'azienda.

- *Duolingo*, un'app per l'apprendimento delle lingue in modo ludico, che vanta oltre 50 milioni di utenti, ha sempre fatto affidamento sull'AI come parte della sua strategia. Una delle funzioni in cui l'AI è stata coinvolta è *Explain My Answer*. Questa funzione consente all'utente di approfondire le regole grammaticali esplicite dietro la risposta di Duolingo. Finora, Duolingo ha cercato di implementare questa funzione con suggerimenti grammaticali usando GPT-3. Tuttavia, è stato solo con l'avvento di GPT-4 che l'accuratezza delle risposte e l'engagement degli utenti sono aumentati, grazie alle sue avanzate capacità di comprensione della lingua in termini di regole grammaticali di diversi idiomi.
- *Stripe*, un fornitore di servizi di pagamento fintech, ha fatto qualcosa di abbastanza visionario. All'inizio del 2023, ha chiesto a cento dipendenti di smettere di svolgere le loro attività quotidiane e di iniziare a immaginare come i LLM potrebbero arricchire e potenziare le funzionalità di un servizio di pagamento. Non solo hanno identificato molte applicazioni potenziali (l'output era una lista di cinquanta applicazioni), ma hanno anche iniziato a prototipare quindici casi d'uso concreti. Tra questi, uno di particolare interesse è l'uso di GPT-4 per il rilevamento delle frodi. La community Discord di Stripe deve affrontare continuamente l'infiltrazione di malintenzionati. GPT-4 aiuta a identificare potenziali truffatori analizzando la sintassi dei post all'interno della community Discord di Stripe e rilevando attività malintenzionate coordinate, garantendo la sicurezza della piattaforma.

Se da un lato la rapida adozione di strumenti di intelligenza artificiale generativa apre emozionanti orizzonti nel prossimo futuro, dall'altro solleva preoccupazioni riguardanti la sostituzione dei posti di lavoro, la privacy dei dati e il possibile abuso della tecnologia. Mentre l'automazione sta rimodellando il mercato del lavoro, le industrie devono affrontare le sfide della transizione della forza lavoro, assicurandosi che i dipendenti siano dotati delle competenze necessarie per prosperare in un mondo sempre più guidato dall'AI.

L'impatto dirompente dell'intelligenza artificiale generativa sulle industrie è innegabile, offrendo sia opportunità sia sfide. Promuovendo la collaborazione tra esseri umani e sistemi di AI, incoraggiando uno sviluppo etico e un'implementazione responsabile, e dando priorità all'apprendimento continuo e alla riqualificazione, possiamo plasmare un futuro in cui l'intelligenza artificiale generativa abbia il ruolo di catalizzatore per un cambiamento positivo nelle industrie.

Finora, ci siamo concentrati principalmente su ciò che può essere ottenuto con l'intelligenza artificiale generativa. Nei prossimi paragrafi, affronteremo le crescenti preoccupazioni riguardo alle implicazioni etiche di questa nuova tecnologia dirompente.

Le preoccupazioni sull'intelligenza artificiale generativa

Come dice Zio Ben al giovane Peter Parker, "Da un grande potere derivano grandi responsabilità".

Mentre raggiungiamo la fine del nostro illuminante viaggio attraverso l'esplorazione del mondo di ChatGPT e dei modelli GPT, è imperativo affrontare le preoccupazioni che sono emerse a proposito di queste tecnologie. Sebbene i progressi nell'intelligenza artificiale generativa siano stati a dir poco rivoluzionari, hanno anche sollevato

domande fondamentali sulla privacy, la sicurezza, le implicazioni etiche e il potenziale abuso.

In effetti, negli ultimi mesi sono state rilasciate molte dichiarazioni e annunci su questi argomenti da parte di aziende e istituzioni, così come da singoli relatori, tra cui preoccupazioni, appelli per fermare ulteriori sviluppi e regolamentazioni appropriate.

Nel prossimo paragrafo, voglio condividere alcune delle ultime notizie e sviluppi, sperando che stimoleranno riflessioni su quello che possiamo aspettarci nel prossimo futuro.

Elon Musk chiede di fermare lo sviluppo

Una recente lettera aperta, firmata da Elon Musk e oltre mille altri professionisti della tecnologia, ha chiesto una temporanea sospensione dello sviluppo di sistemi di intelligenza artificiale più sofisticati di GPT-4. I firmatari, tra cui Steve Wozniak, Emad Mostaque e Yuval Noah Harari, esprimono le loro preoccupazioni sui significativi rischi che questi sistemi di intelligenza artificiale pongono alla società e all’umanità. La lettera chiede che i laboratori di intelligenza artificiale leader mettano in pausa il processo di addestramento di questi sistemi avanzati per un periodo minimo di sei mesi, garantendo che la sospensione sia pubblica e verificabile. Si sottolineano le preoccupazioni legate al potenziale per la propaganda guidata dall’IA, all’automazione del lavoro e alla perdita di controllo sulla nostra civiltà. Questo appello giunge in seguito al lancio da parte di OpenAI di GPT-4, un modello di linguaggio potenziato che alimenta la versione premium di ChatGPT. Secondo OpenAI, GPT-4 è capace di svolgere compiti più complessi e genera risultati più raffinati rispetto alle iterazioni precedenti, con meno difetti.

I sistemi di intelligenza artificiale come GPT-4 operano su vaste quantità di dati, che utilizzano per rispondere a domande ed eseguire

compiti. ChatGPT, che ha debuttato a novembre, ha capacità simili a quelle umane di comporre e-mail, organizzare piani di viaggio, scrivere codice ed eccellere in esami come quello da avvocato.

OpenAI non ha ancora commentato la lettera, ma l'organizzazione ha riconosciuto l'importanza di garantire che le tecnologie IA più intelligenti degli esseri umani servano gli interessi dell'umanità. OpenAI suggerisce che i futuri sistemi potrebbero richiedere una valutazione indipendente prima dell'addestramento e che sforzi avanzati dovrebbero regolare l'espansione delle risorse computazionali utilizzate per lo sviluppo del modello.

Diverse aziende, tra cui Google, Microsoft, Adobe, Snapchat, DuckDuckGo e Grammarly, hanno presentato servizi che sfruttano le funzionalità generative dell'IA. La ricerca di OpenAI evidenzia i rischi connessi a queste capacità, come la possibilità di citare fonti non attendibili o di consentire ad attori malintenzionati di ingannare o sfruttare altri.

Gli specialisti di intelligenza artificiale sono sempre più allarmati dalla traiettoria dell'industria e dall'eventuale assenza di precauzioni necessarie e comprensione delle conseguenze. La lettera sottolinea che l'IA avanzata potrebbe avere un impatto drammatico sulla vita sulla Terra e richiede una pianificazione e una gestione attente. Si sottolinea che tale pianificazione è attualmente insufficiente, poiché i laboratori di intelligenza artificiale persistono nella creazione e nell'implementazione di sistemi IA sempre più potenti e sempre più difficili da comprendere, prevedere o controllare. Se questa lettera aperta non avesse effetto vincolante, un altro esempio è quello che il Garante della Privacy italiano ha dichiarato, su cui ci concentreremo nel prossimo paragrafo.

Blocco di ChatGPT in Italia dal Garante della Privacy

L’Italia è diventata la prima nazione occidentale a proibire ChatGPT a causa di preoccupazioni sulla privacy.

L’autorità italiana per la protezione dei dati, il Garante della Privacy, ha annunciato nel marzo 2023 l’avvio di un’indagine sul trattamento dei dati da parte di OpenAI, con la conseguenza che l’azienda ha momentaneamente disabilitato (Figura 11.8) ChatGPT in Italia.

ChatGPT disabled for users in Italy

Dear ChatGPT user,

We regret to inform you that we have disabled ChatGPT for users in Italy at the request of the Italian Garante.

We are issuing refunds to all users in Italy who purchased a ChatGPT Plus subscription in March. We are also temporarily pausing subscription renewals in Italy so that users won't be charged while ChatGPT is suspended.

We are committed to protecting people's privacy and we believe we offer ChatGPT in compliance with GDPR and other privacy laws. We will engage with the Garante with the goal of restoring your access as soon as possible.

Many of you have told us that you find ChatGPT helpful for everyday tasks, and we look forward to making it available again soon.

If you have any questions or concerns regarding ChatGPT or the refund process, we have prepared a list of Frequently Asked Questions to address them.

—The OpenAI Support Team

Figura 11.8 Messaggio di OpenAI quando si accede a ChatGPT in Italia.

L'autorità italiana per la protezione dei dati ha dato a OpenAI venti giorni per affrontare le sue preoccupazioni o affrontare una multa di 20 milioni di euro o fino al 4% del loro fatturato annuo.

OpenAI ha disattivato ChatGPT per gli utenti italiani il 1° aprile 2023 su richiesta del regolatore italiano per la protezione dei dati, il

Garante, e ha dichiarato il suo impegno per la protezione della privacy e la conformità al GDPR.

Nel corso dello stesso mese, OpenAI ha provveduto a soddisfare le richieste del Garante della Privacy, apportando una serie di modifiche e accortezze che sono state ritenute sufficienti per permettere la riattivazione del servizio in Italia, avvenuta il 28 aprile 2023.

Le preoccupazioni e gli interventi sopra menzionati stanno solo grattando la superficie di un argomento più ampio, che è il concetto di intelligenza artificiale responsabile, trattato nel prossimo paragrafo.

Implicazioni etiche dell'AI generativa e perché abbiamo bisogno di un'AI responsabile

Il paragrafo precedente ha evidenziato come, parallelamente alla diffusione della conoscenza e all'adozione delle tecnologie di intelligenza artificiale generativa, stia crescendo una preoccupazione generale.

Il rapido avanzamento delle tecnologie di intelligenza artificiale porta con sé una miriade di considerazioni etiche e sfide che devono essere affrontate con attenzione per garantire il loro impiego responsabile ed equo. Alcune di queste sono elencate di seguito.

- *Privacy e sicurezza dei dati*: poiché i sistemi di intelligenza artificiale si basano pesantemente sui dati per i loro processi di apprendimento e decisionali, diventa fondamentale garantire la privacy e la sicurezza dei dati.
- *Bias e equità*: i modelli di intelligenza artificiale spesso apprendono da dati storici, che potrebbero introdurre involontariamente bias. Affrontare il bias e l'equità nei sistemi di

intelligenza artificiale comporta diverse strategie, tra cui l'uso di set di dati diversificati e rappresentativi e lo sviluppo di algoritmi che diano priorità all'equità.

- *Trasparenza e responsabilità*: con l'avanzare della complessità dei sistemi di intelligenza artificiale, comprendere i loro processi decisionali diventa difficile. È quindi importante sviluppare modelli di intelligenza artificiale spiegabili e stabilire linee chiare di responsabilità legale per gli sviluppatori e le organizzazioni che utilizzano tali sistemi.
- *Il futuro del lavoro*: l'automazione guidata dall'intelligenza artificiale ha il potenziale per sostituire lavori in alcuni settori, suscitando preoccupazioni sul futuro del lavoro. È fondamentale investire in programmi di riqualificazione e aggiornamento per aiutare i lavoratori ad adattarsi al cambiamento e favorire la collaborazione tra umani e intelligenza artificiale.

Affrontando queste considerazioni etiche e queste sfide, possiamo lavorare nella giusta direzione per garantire che le tecnologie di intelligenza artificiale siano sviluppate e implementate in modo responsabile, promuovendo un futuro migliore e più equo per tutti.

Ora, la prossima domanda logica potrebbe essere: data l'enorme accelerazione delle tecnologie di intelligenza artificiale negli ultimi mesi, cosa dovremmo aspettarci nel prossimo futuro?

Cosa aspettarsi nel prossimo futuro

L'accelerazione della ricerca e degli sviluppi in intelligenza artificiale negli ultimi mesi è stata incredibile. Da novembre 2022 fino al momento della stesura di questo libro (aprile 2023), abbiamo assistito a una serie di importanti eventi:

- il lancio di ChatGPT (novembre 2022);
- la disponibilità generale di Azure OpenAI (gennaio 2023);
- la disponibilità generale dell'API del modello su cui si basano ChatGPT, GPT-3.5-turbo (febbraio 2023);
- la disponibilità generale di LMM come Kosmos-1 e GPT-4 (marzo 2023);
- l'annuncio da parte di Microsoft del sistema Copilot (marzo 2023).

Questo ritmo incredibile rende difficile prevedere che cosa verrà dopo. Come potete vedere, questa velocità ha anche sollevato preoccupazioni tra istituzioni, aziende e personaggi pubblici, a causa della mancanza di regolamentazione per queste nuove tecnologie. Allo stesso tempo, aziende e istituzioni dovranno inevitabilmente adattarsi a questo nuovo panorama per tenere il passo con i concorrenti.

Abbiamo visto come alcune aziende IT, come Microsoft, stiano già integrando i modelli GPT nelle loro applicazioni come sistema copilota, mentre altre aziende, come WolframAlpha, Expedia e Instacart, hanno progettato plugin che sono integrati direttamente in ChatGPT.

Il movimento verso l'infusione dei modelli OpenAI nelle applicazioni è evidente anche dalla varietà di framework che sono stati sviluppati con lo scopo di facilitare l'integrazione tra LLM e applicazioni e di gestire prompt, conversazioni, memoria, tokenizzazione e altri passaggi tipici necessari. Alcuni esempi di questi framework sono LangChain, Pinecone e Semantic Kernel.

Abbiamo menzionato nel Capitolo 2 che la missione di OpenAI è quella di costruire un'intelligenza artificiale generale ampiamente vantaggiosa, un tipo di intelligenza artificiale che, essendo "generale", è in grado di apprendere e svolgere una vasta gamma di compiti, senza la necessità di programmazione specifica per il compito. In altre

parole, OpenAI sta immaginando una macchina di intelligenza artificiale che sia in grado di fare tutto ciò che può fare un essere umano.

Se avessimo pensato a questo un anno fa, ci sarebbe sembrato futuristico. Oggi, con il ritmo vertiginoso dello sviluppo, è così inverosimile che, non molto lontano nel futuro, otterremo macchine dotate di intelligenza artificiale generale?

Riepilogo

Il rapido sviluppo di tecnologie di intelligenza artificiale come OpenAI, ChatGPT e modelli di intelligenza artificiale generativa sta inaugurando una nuova era di innovazione e trasformazione. Con l'enorme potenziale di rivoluzionare le industrie e ridisegnare la vita quotidiana, questi progressi stanno riscrivendo le regole dell'interazione uomo-macchina.

Mentre ci troviamo sull'orlo di questo futuro guidato dall'intelligenza artificiale, è nostra responsabilità collettiva garantire che queste tecnologie vengano utilizzate in modo responsabile ed etico. Accogliendo le opportunità e affrontando le sfide, possiamo favorire un mondo in cui l'intelligenza artificiale sarà realmente intesa come strumento a beneficio dell'umanità.

Il rilascio di ChatGPT è avvenuto meno di due anni fa – un'era se pensiamo al ritmo degli sviluppi dell'intelligenza artificiale negli ultimi mesi – eppure riflette l'influenza inevitabile dell'AI sulla nostra vita e le sfide che ci attendono nell'adattarci a questa nuova realtà.

Riferimenti

- https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web

Indice

Introduzione
- A chi si rivolge questo libro
- Di che cosa tratta il libro
- File degli esempi e immagini a colori
- L'autrice
- Il revisore

Parte I - I fondamenti dell'AI generativa e dei modelli GPT

Capitolo 1 - L'IA generativa
- Introduzione all'AI generativa
- Domini dell'AI generativa
- La storia e lo stato attuale della ricerca
- Riepilogo
- Riferimenti

Capitolo 2 - OpenAI e ChatGPT: oltre l'hype del mercato
- Requisiti tecnici
- Che cos'è OpenAI?
- Una panoramica delle famiglie di modelli OpenAI
- Verso ChatGPT: le basi matematiche del modello
- Introduzione a ChatGPT
- Riepilogo
- Riferimenti

Parte II - ChatGPT in azione

Capitolo 3 - Familiarizzare con ChatGPT

Creazione di un account ChatGPT
L'interfaccia utente di ChatGPT
Organizzazione delle chat
Riepilogo
Riferimenti

Capitolo 4 - Introduzione alla struttura del prompt

Che cos'è un prompt e perché è importante?
Zero-shot, one-shot e few-shot learning
Principi di strutturazione del prompt per ottenere risultati rilevanti e coerenti
Evitare il rischio di hidden bias e tener conto delle considerazioni etiche in ChatGPT
Riepilogo
Riferimenti

Capitolo 5 - Potenziare la produttività quotidiana con ChatGPT

Requisiti tecnici
ChatGPT come assistente quotidiano
Generazione di testo
Miglioramento delle abilità di scrittura e traduzione
Consulenza sui contenuti
Riepilogo

Capitolo 6 - Rivoluzionare lo sviluppo software con ChatGPT

Perché parliamo di ChatGPT per gli sviluppatori?
Generazione, ottimizzazione e correzione del codice
Generazione della documentazione e descrizione del codice
Interpretabilità di un modello di machine learning
Traduzione tra diversi linguaggi di programmazione

Riepilogo

Capitolo 7 - Rivoluzionare il marketing con ChatGPT

Requisiti tecnici
Applicazioni di ChatGPT per i professionisti del marketing
Sviluppo di nuovi prodotti e strategia di go-to-market
Il test A/B per il confronto nel marketing
Potenziare l'ottimizzazione dei motori di ricerca (SEO)
Sentiment analysis per migliorare la qualità e aumentare la soddisfazione del cliente
Riepilogo

Capitolo 8 - Reinventare la ricerca con ChatGPT

I vantaggi di ChatGPT per i ricercatori
Generazione di idee per uno studio di ricerca
Supporto per la progettazione di un esperimento
Generazione e formattazione di una bibliografia
Generazione di una presentazione per lo studio
Riepilogo
Riferimenti

Parte III - OpenAI per le aziende

Capitolo 9 - Introduzione ad Azure OpenAI

Requisiti tecnici
OpenAI e Microsoft per l'AI a livello aziendale
Perché introdurre un cloud pubblico?
Che cosa si intende per intelligenza artificiale responsabile
Riepilogo
Riferimenti

Capitolo 10 - Casi d'uso di tendenza per le aziende

Requisiti tecnici
Come Azure OpenAI viene utilizzato nelle aziende

Analisi e generazione di contratti
Fornire modelli di contratto
Call Center Analytics
Esplorazione della ricerca semantica
Riepilogo
Riferimenti

Capitolo 11 - Epilogo e riflessioni finali
Cosa abbiamo appreso finora
Questo è solo l’inizio
L’impatto delle tecnologie generative sulle industrie: una forza dirompente
Le preoccupazioni sull’intelligenza artificiale generativa
Implicazioni etiche dell’AI generativa e perché abbiamo bisogno di un’AI responsabile
Cosa aspettarsi nel prossimo futuro
Riepilogo
Riferimenti