UNIVERSITÀ
POLITECNICA
DELLE MARCHE

Corso di «Data Science»

A.A. 2025/2026

Parte XVII: La Generative AI

Prof. Domenico Ursino
d.ursino@univpm.it

Dal Natural Language Processing alla Generative AI

- La Generative AI si riferisce a sistemi di IA in grado di generare nuovi contenuti, come testi, immagini o musica, sulla base di modelli appresi da dati esistenti.

- L'NLP svolge un ruolo cruciale nella Generative AI, in particolare nella generazione di testi e contenuti basati sul linguaggio.

- Ecco come l'NLP è utile nella Generative AI:

  - Generazione di testo: le tecniche di NLP vengono utilizzate per costruire modelli in grado di generare testi coerenti e contestualmente rilevanti. Ciò include la generazione di articoli di cronaca, racconti, poesie e persino codice.

  - Chatbot e agenti conversazionali: l'NLP consente lo sviluppo di chatbot sofisticati e assistenti virtuali in grado di comprendere e rispondere alle domande degli utenti in linguaggio naturale. Questi sistemi si basano sull'NLP per il riconoscimento dell'intento, la generazione di risposte e il mantenimento del contesto nelle conversazioni.

  - Traduzione linguistica: i modelli generativi basati sull'NLP possono fornire traduzioni accurate e fluenti tra lingue diverse, migliorando la comunicazione e l'accessibilità.

Dal Natural Language Processing alla Generative AI

- Sintesi: le tecniche di NLP possono essere utilizzate per creare modelli generativi in grado di sintetizzare documenti lunghi in versioni concise, estraendo le informazioni chiave e mantenendo il significato originale.

- Personalizzazione dei contenuti: l'NLP aiuta i sistemi Generative AI a personalizzare i contenuti in base alle preferenze e ai comportamenti degli utenti, come feed di notizie personalizzati, consigli e contenuti di marketing.

- Assistenza alla scrittura creativa: la Generative AI basata sull'NLP può assistere gli scrittori suggerendo idee, generando trame o persino redigendo contenuti in base a criteri specifici.

- Esempi dell'utilizzo dell'NLP negli strumenti di Generative AI sono i seguenti:

  - GPT-5: sviluppato da OpenAI, GPT-5 è un modello linguistico all'avanguardia che utilizza l'NLP per generare testi simili a quelli umani. È in grado di svolgere compiti quali scrivere saggi, rispondere a domande e generare frammenti di codice.

  - BERT: sebbene utilizzati principalmente per la comprensione del testo, modelli come BERT possono essere ottimizzati per attività di generazione di testo, migliorando la capacità dei sistemi di IA generativa.

Dal Natural Language Processing alla Generative AI

  - Transformer: questi modelli, che costituiscono la base di molti sistemi NLP all'avanguardia, sono fondamentali per la Generative AI. Consentono la generazione di testi coerenti e contestualizzati attraverso la comprensione e la modellizzazione delle relazioni tra le parole in una frase.

- Sfruttando l'NLP, i sistemi di Generative AI possono creare contenuti più naturali, accurati e contestualmente appropriati, migliorando così l'esperienza degli utenti in varie applicazioni.

Gli strati dell'Intelligenza Artificiale

- La seguente immagine fornisce una panoramica dei vari strati di cui è costituita l'Intelligenza Artificiale:

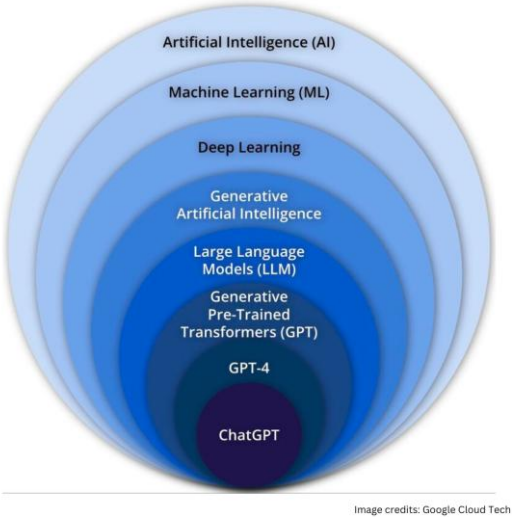


Image credits: Google Cloud Tech

- Machine Learning (ML): Il Machine Learning è un tipo di IA che consente alle applicazioni software di diventare più accurate nella previsione dei risultati senza essere esplicitamente programmate per farlo. Gli algoritmi di ML utilizzano i dati storici come input per prevedere nuovi valori di output.

- Deep Learning (DL): Il Deep Learning è un tipo di ML che utilizza reti neurali artificiali per apprendere dai dati. Le reti neurali si ispirano alla struttura e alla funzione del cervello umano, costituite da strati di nodi interconnessi, ciascuno dei quali esegue una semplice operazione matematica.

Gli strati dell'Intelligenza Artificiale

- La Generative AI: la Generative AI è un tipo di IA in grado di creare nuovi contenuti, inclusi testi, codici, immagini e musica. I modelli di Generative AI vengono addestrati su grandi set di dati di contenuti esistenti, imparando a identificare pattern nei dati e utilizzando tali pattern per generare nuovi contenuti.

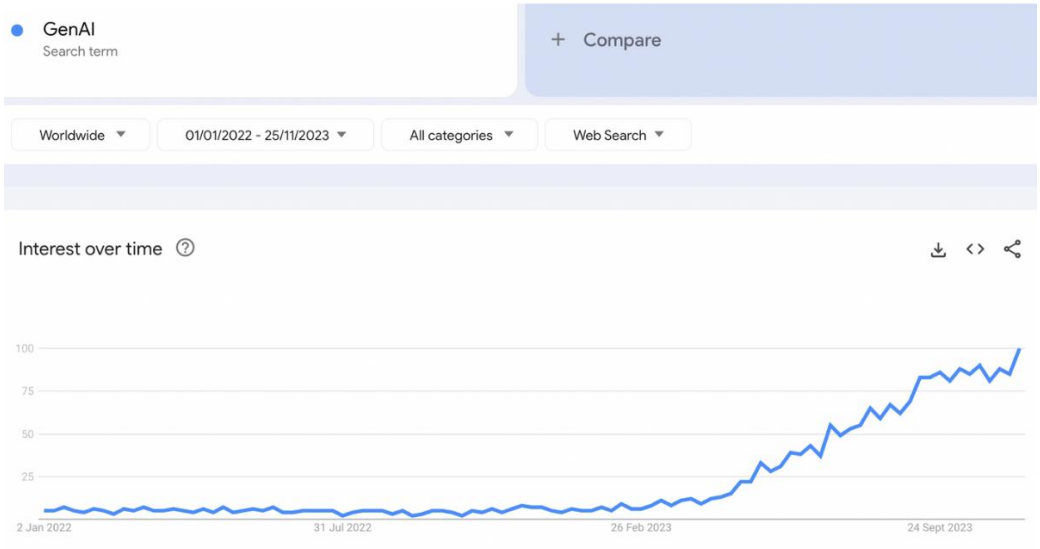
- Large Language Model (LLM): gli LLM sono un tipo di modello di Generative AI addestrato su enormi set di dati di testo e codice. Gli LLM possono generare testo, tradurre lingue, scrivere diversi tipi di contenuti creativi e rispondere alle domande in modo informativo.

- Generative Pre-trained Transformers (GPT): I GPT sono un tipo di LLM che utilizza un'architettura basata sui transformer. I transformer sono un'architettura di rete neurale particolarmente adatta per le attività di elaborazione del linguaggio naturale.

- GPT-5 e ChatGPT: GPT-5 e ChatGPT sono due esempi di modelli GPT. GPT-5 è un LLM sviluppato da OpenAI, mentre ChatGPT è un LLM (anch'esso sviluppato da OpenAI) progettato specificamente per le applicazioni chatbot.

La crescita della Generative AI

- La Generative AI sta vivendo un'ascesa vertiginosa. Questo progresso è alimentato da potenti algoritmi addestrati su enormi set di dati, che consentono loro di apprendere pattern complessi e generare risultati indistinguibili dalle creazioni umane.



- In passato, i sistemi di IA si basavano principalmente su risposte e istruzioni pre-programmate, il che limitava la loro capacità di innovare ed essere veramente creativi.

- La Generative AI rompe questo schema. I modelli di Generative AI non solo sono in grado di rispondere a domande e seguire comandi, ma possono anche immaginare e produrre concetti completamente nuovi, aprendo la strada a possibilità e progressi entusiasmanti in tutti i campi e settori industriali.

La crescita della Generative AI

- Dalla progettazione di esperienze di apprendimento personalizzate nel campo dell'istruzione allo sviluppo di trattamenti medici all'avanguardia nel settore sanitario, le potenziali applicazioni della Generative AI sono vaste e in continua evoluzione.

- Man mano che questa tecnologia continua a maturare, possiamo aspettarci che il suo impatto trasformi ogni aspetto della nostra vita, inaugurando una nuova era di creatività e innovazione.

- Alcune possibili applicazioni già esistenti sono:

  - Scrivere testi di marketing accattivanti e post di blog coinvolgenti

  - Comporre musica su misura per le preferenze specifiche dell'utente

  - Generare immagini realistiche e dettagliate per qualsiasi scopo

  - Sviluppare applicazioni software innovative con caratteristiche prima inimmaginabili.

- Tuttavia, l'ascesa dell'IA generativa non è priva di sfide. Le preoccupazioni relative alle implicazioni etiche, ai potenziali pregiudizi e alla sostituzione dei posti di lavoro sono tutte considerazioni importanti.

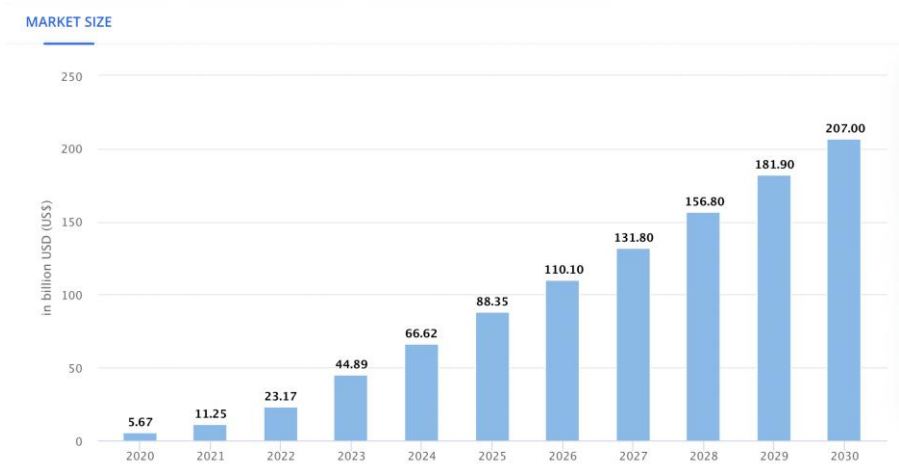
La crescita della Generative AI

- Mentre si va avanti con questa potente tecnologia, è fondamentale affrontare queste preoccupazioni e sviluppare framework responsabili per il suo sviluppo e utilizzo.

- Sfruttando il potenziale dell'IA generativa e mitigandone i rischi, possiamo aprire le porte a un futuro ricco di creatività, innovazione e progresso.

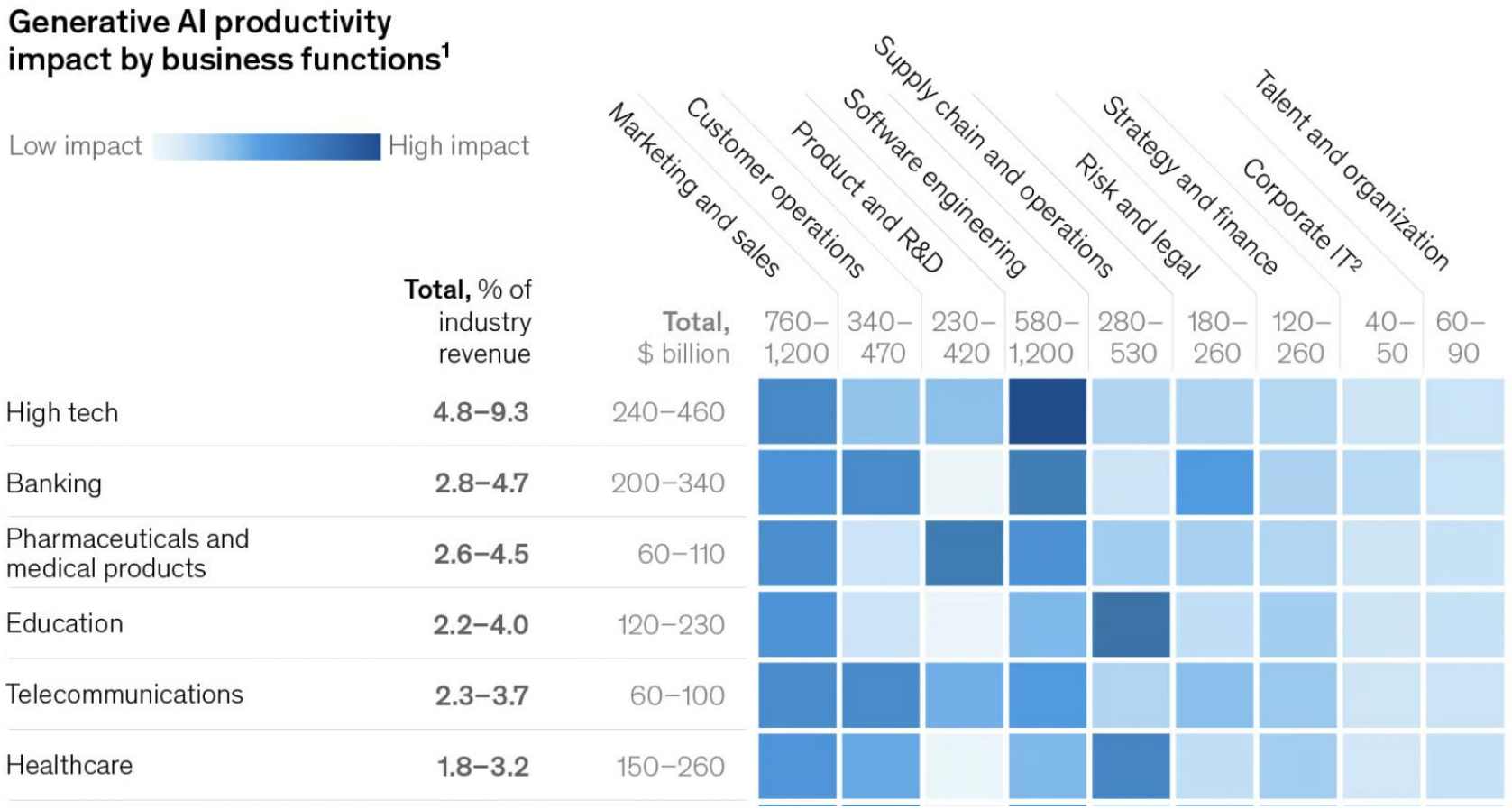
- Secondo Statisa, il mercato della Generative AI dovrebbe registrare un tasso di crescita annuale (CAGR 2023-2030) del 24,40%, raggiungendo un volume di mercato pari a 207 miliardi di dollari entro il 2030 (dati aggiornati ad agosto 2023).

- A livello globale, il mercato più grande sarà quello degli Stati Uniti (16,14 miliardi di dollari nel 2023).



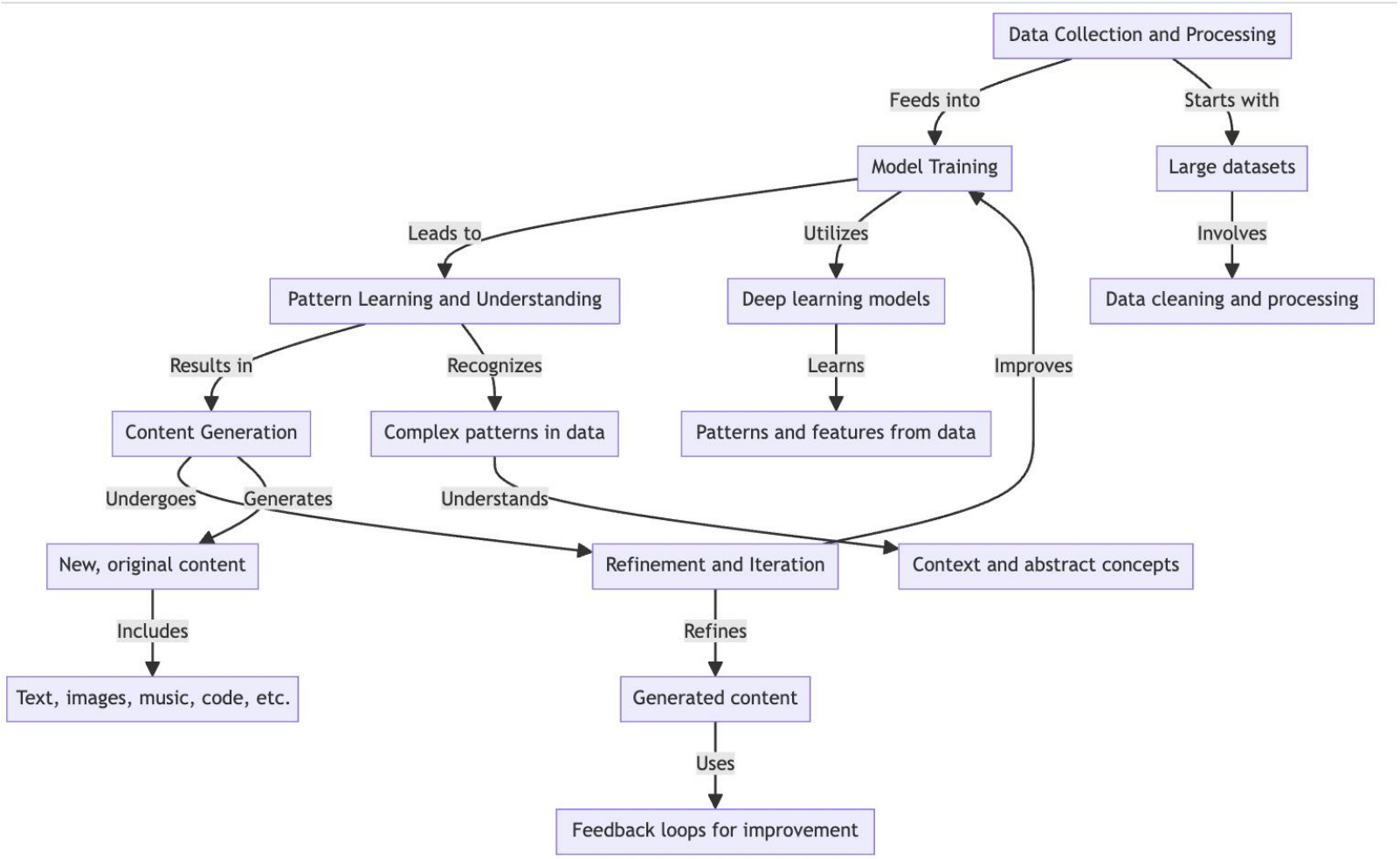
La crescita della Generative AI

- McKinsey & Company ha studiato l'impatto della Generative AI su varie funzioni aziendali in diversi settori, rilevando che high-tech, il settore bancario, quello farmaceutico, l'istruzione e le telecomunicazioni sono i cinque settori in cui l'IA generativa avrà il maggiore impatto.



Come lavora la Generative AI

- Il seguente diagramma fornisce una panoramica approssimativa del funzionamento dell'IA generativa, dalle fasi iniziali fino al risultato finale e alla rifinitura.



Come lavora la Generative AI

- Il diagramma precedente illustra efficacemente questo flusso, mostrando come ogni fase conduca alla successiva, con cicli di feedback che indicano il processo di miglioramento e apprendimento continuo intrinseco nei sistemi di Generative AI.

- In particolare le fasi che caratterizzano la Generative AI sono le seguenti:

  - Raccolta ed elaborazione dei dati.

    - Il processo inizia con la raccolta e l'elaborazione di grandi set di dati.

    - Questa fase è fondamentale poiché costituisce la base su cui il modello di IA apprende e genera contenuti.

    - Comprende la raccolta di dati estesi, la loro pulizia ed elaborazione per renderli adatti all'addestramento del modello di IA.

  - Addestramento del modello.

    - La fase successiva è l'addestramento del modello, in cui vengono utilizzati modelli di deep learning per apprendere pattern e caratteristiche dai dati elaborati.

Come lavora la Generative AI

- Questa è anche la fase in cui l'IA inizia a comprendere la struttura dei dati, i modelli e le sfumature.

- Apprendimento e comprensione dei pattern.

  - In questa fase, il modello di IA riconosce pattern complessi nei dati e inizia a comprendere il contesto e i concetti astratti.

  - Si tratta di una fase critica in cui l'IA sviluppa la capacità di generare contenuti significativi e coerenti.

- Generazione di contenuti.

  - Sulla base dei modelli appresi e della comprensione acquisita, l'IA genera quindi contenuti nuovi e originali. Questi contenuti possono variare notevolmente, includendo testo, immagini, musica, codice, ecc., a seconda dell'addestramento dell'IA e dell'applicazione prevista.

- Perfezionamento e iterazione.

  - La fase finale prevede il perfezionamento dei contenuti generati e l'utilizzo di cicli di feedback per il miglioramento. Questo processo iterativo garantisce che il risultato sia di alta qualità e soddisfi gli standard o gli obiettivi desiderati.

Differenza tra Machine Learning e Generative AI

- I modelli di Generative AI, come GPT-5 o DALL-E, differiscono in modo significativo dai modelli tradizionali di Machine Learning (ML) per quanto riguarda gli obiettivi, gli approcci di formazione, la complessità e le applicazioni.

- I modelli ML tradizionali si concentrano sulla previsione dei risultati attraverso l'apprendimento della relazione tra i dati di input e le etichette di output, il che li rende ideali per attività come la classificazione e la regressione.

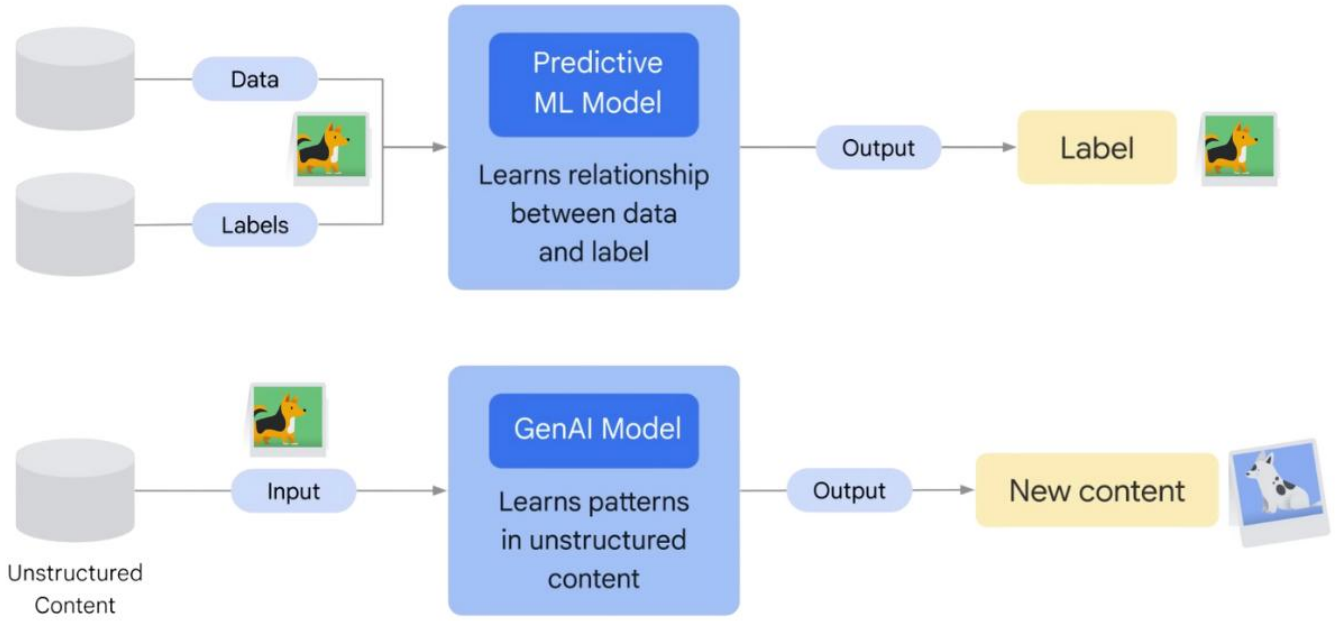


Image credits: Google Cloud Tech

Differenza tra Machine Learning e Generative AI

- I modelli di Generative AI, come GPT-5 o DALL-E, differiscono in modo significativo dai modelli tradizionali di Machine Learning (ML) per quanto riguarda gli obiettivi, gli approcci di formazione, la complessità e le applicazioni.
- I modelli ML tradizionali si concentrano sulla previsione dei risultati attraverso l'apprendimento della relazione tra i dati di input e le etichette di output, il che li rende ideali per attività come la classificazione e la regressione.
- Questi modelli richiedono spesso set di dati strutturati con etichette chiare, e utilizzano tecniche come il supervised learning.
- Al contrario, i modelli di Generative AI sono progettati per creare nuovi contenuti apprendendo pattern in dati non strutturati.
- Sono addestrati su vasti set di dati utilizzando metodi di unsupervised o self-supervised learning, che consentono loro di generare testi, immagini e altro ancora.
- Questi modelli utilizzano in genere architetture complesse, come le reti neurali profonde, e sono particolarmente adatti per attività creative come la scrittura, la generazione di immagini e la composizione musicale.
- Mentre i modelli ML tradizionali eccellono nell'analisi dei dati strutturati, i modelli di Generative AI si distinguono per la loro capacità di produrre contenuti nuovi e diversificati.

Differenza tra Machine Learning e Generative AI

- La seguente tabella riassume le principali differenze tra i due approcci:

Caratteristica	Machine Learning	Generative AI
Scopo	Analizzare i dati esistenti per fare predizioni o prendere decisioni	Creare contenuti completamente nuovi
Approccio di apprendimento	Si basa principalmente su dati etichettati	Utilizza dati non etichettati o parzialmente etichettati
Architettura del modello	Algoritmi specifici per i task (regressione lineare, alberi di decisione, etc.)	Architetture Deep Learning (GAN, Transformers)
Output	Predizioni, classificazioni, decisioni	Testi, codici, immagini e musiche nuovi
Esempi	Filtraggio dello spam riconoscimento delle immagini, ricerca di frodi nelle carte di credito	Generazione di testi (ChatGPT, Gemini), generazione di immagini (DALL-E), composizione di musica
Punti di forza	Alta accuratezza su task specifici, output interpretabili	Creatività, personalizzazione, efficienza
Punti di debolezza	Limitato ai dati esistenti, richiede una estesa ingegnerizzazione delle feature	Potenziali distorsioni, minore trasparenza nel prendere le decisioni
Applicazioni	Medicina, Finanza, Marketing, Manifattura	Creazione di contenuti, design, education, intrattenimento
Scopo complessivo	Ricavare insight dai dati	Generare contenuti nuovi ed innovativi

Modelli di Generative AI

- I principali modelli di Generative AI sono i seguenti:
  - Autoencoder Variazionali (VAE);
  - Reti Generative Avversarie (GAN);
  - Transformer;
  - Modelli di diffusione;
  - Neural Radiance Field (NeRF);
  - Large Language Model (LLMs).
- Abbiamo già visto i Transformer in precedenza; nel seguito daremo uno sguardo veloce a tutti gli altri modelli.

Modelli di Generative AI – Autoencoder Variazionali (VAE)

- Uno dei modelli generativi maggiormente utilizzati è costituito dagli autoencoder variazionali, i quali sono particolarmente efficaci nell'elaborazione di immagini, e quindi trovano il maggiore impiego nelle IA per la generazione di immagini, come Midjourney o DALL·E.

- Essendo i VAE un tipo particolare di autoencoder, prima di scendere nel dettaglio è opportuno spiegare cosa sono e come funzionano gli autoencoder.

- Un autoencoder è un sistema costituito da due reti neurali, una dedicata dalla compressione di grandi quantità di dati in uno spazio chiamato latente, l'altra invece dedicata alla decompressione e alla ricostruzione del contenuto.

- Lo scopo principale di un autoencoder è quello di estrarre da un input le cosiddette variabili latenti attraverso il passaggio in un collo di bottiglia prima di arrivare al livello di output.

- Questo passaggio "obbliga" la rete ad estrarre solo le informazioni che realmente servono per ricostruire il dato.

- Le variabili latenti sono variabili spesso non osservabili, ma che contengono l'informazione su come sono distribuiti i dati.

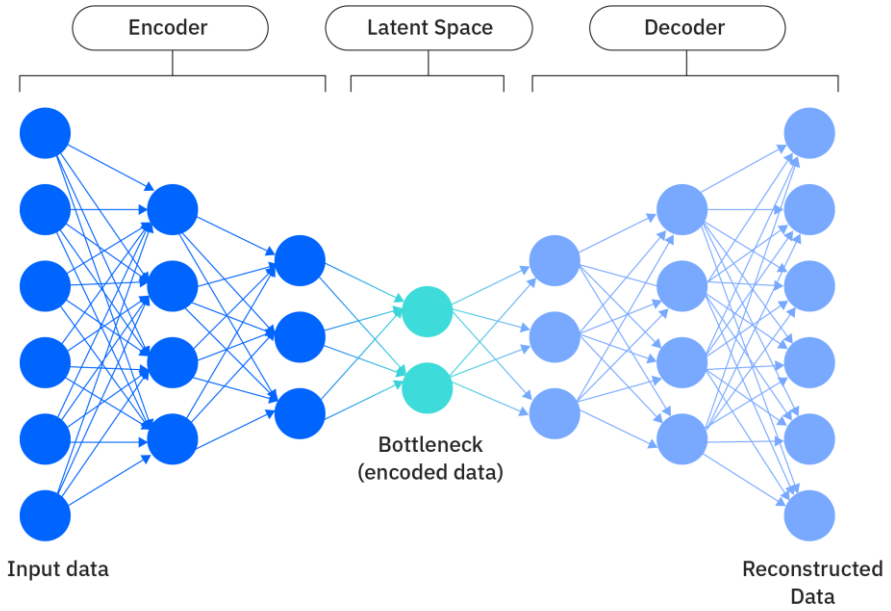
- Infatti, molto spesso, non tutto il dato in input è costituito da informazioni utili, anzi queste ultime rappresentano una minima parte di esso. Il resto è costituito da rumore di fondo la cui rimozione è proprio lo scopo dell'encoder.

Modelli di Generative AI – Autoencoder Variazionali (VAE)

- Una volta ricostruita l'immagine, si può effettuare un confronto pixel per pixel e calcolare la loss function.
- Una delle loss function più utilizzate è l'MSE (Medium Square Error o Errore Quadratico Medio):

$$\mathcal{L}(x, y) = \frac{1}{n} \sum_{i=1}^{n} (x_i - y_i)^2$$

- L'architettura di un autoencoder viene mostrata nella seguente figura:



Modelli di Generative AI – Autoencoder Variazionali (VAE)

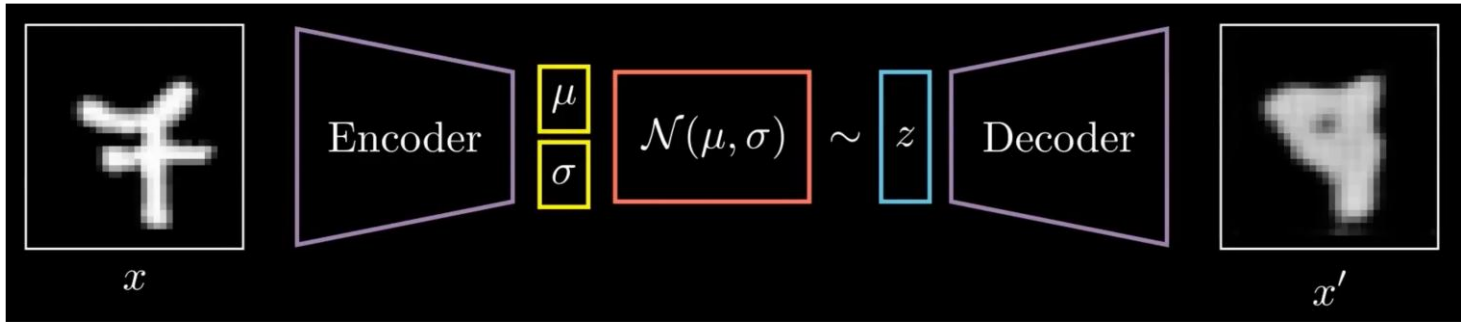
- Come si evince dalla figura, l'architettura di un autoencoder è costituita da tre elementi principali:
  - Encoder: si occupa di estrarre le variabili latenti dai dati in input; nella sua versione base ogni livello ha meno nodi del precedente.
  - Collo di bottiglia: costituisce lo strato di output dell'encoder e quello di input del decoder.
    - Affinché l'autoencoder possa funzionare correttamente è necessario effettuarne un opportuno dimensionamento.
    - La dimensione del collo di bottiglia è costituita dal numero di neuroni dello spazio latente.
  - Decoder: si occupa di ricostruire il dato a partire dallo spazio latente.
    - In genere ha una struttura speculare rispetto a quella dell'encoder; dunque ogni livello attraversato ha più nodi del precedente.
- Attraverso più epoche di addestramento, lo scopo è via via ridurre l'elevato errore iniziale.

Modelli di Generative AI – Autoencoder Variazionali (VAE)

- Qualora ciò avvenisse correttamente, nello spazio latente contenuti simili in input dovrebbero trovarsi nella stessa regione e formare un cluster. Cluster diversi non dovrebbero essere troppo vicini tra loro, altrimenti si correrebbe il rischio di effettuare una ricostruzione errata.

- Gli autoencoder tradizionali, seppur molto utili, sono caratterizzati da alcune problematiche: ad esempio, un'organizzazione disordinata dello spazio latente poteva portare a generare immagini prive di significato e non coerenti.

- L'architettura degli autoencoder variazionali (VAE), è mostrata nella seguente figura:



- Rispetto agli autoencoder tradizionali, in cui il collo di bottiglia ha una codifica di tipo deterministico, negli autoencoder variazionali la codifica è di tipo probabilistico.

- La codifica dello spazio latente avviene mediante due differenti vettori, uno delle medie μ e uno delle deviazioni standard σ, che contengono i valori per ogni dimensione dello spazio latente.

Modelli di Generative AI – Autoencoder Variazionali (VAE)

- A partire da questi valori probabilistici il decoder si occupa di sintetizzare un output che sarà simile al dato originario fornito in input.

- Alla base del funzionamento dei VAE c'è la statistica bayesiana.

- In conclusione, un VAE rappresenta uno strumento molto utile nella generazione di nuovi contenuti poiché, introducendo una componente stocastica nel sistema, l'immagine generata in output risulterà completamente nuova.

- Si noti, infatti, che nella figura precedente l'immagine ricostruita x' è ancora un numero 7 come quella in input, ma è un'immagine generata tramite campionamento nello spazio latente, e dunque un'immagine nuova e distinta.

- Oltre che per la generazione di immagini, gli autoencoder variazionali si sono rivelati particolarmente efficienti nella fusione di due immagini distinte interpolando i punti campionati nello spazio latente.

- Tuttavia, questo modello generativo presenta anche delle limitazioni; infatti, le immagini che genera molto spesso sono sfocate o poco definite.

    - Inoltre, non è possibile aggiungere dei vincoli sull'immagine da generare.

Modelli di Generative AI – Reti Generative Avversarie (GAN)

- Le reti generative avversarie (Generative Adversarial Network) sono un modello generativo che ha avuto molto successo negli ultimi anni.

- È un modello competitivo; una GAN, infatti, è costituita da due componenti che competono tra loro durante l'addestramento. Più specificatamente, i due componenti sono:

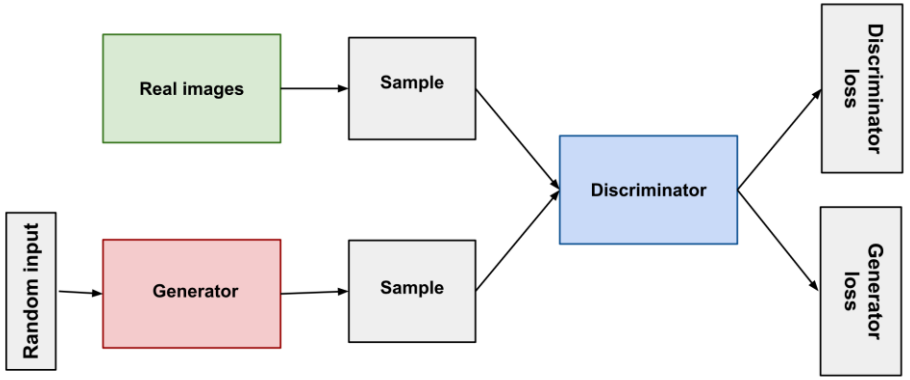
  - un modello generativo (G) che si occupa della generazione di nuovi dati;

  - un modello discriminativo (D) che si occupa di discriminare i dati reali presenti nei dataset da quelli creati artificialmente dal generatore.

- Le due reti nell'addestramento si scontrano fino a raggiungere un punto di equilibrio; infatti, il generatore continua a produrre dati da sottoporre al discriminatore. Quando il discriminatore non è più in grado di distinguere i dati reali da quelli creati artificialmente, allora si può concludere l'addestramento.

Modelli di Generative AI – Reti Generative Avversarie (GAN)

- L'architettura di una GAN viene mostrata nella seguente figura:



- Esistono varie tipologie di reti avversarie generative:
  - Vanilla GAN: è la versione base delle reti generative avversarie.
  - cGAN: le GAN condizionali sono un particolare tipo di GAN che consentono di inserire condizioni, come, ad esempio, un'etichetta o un'informazione aggiuntiva che descrive il contenuto dell'immagine da generare.
  - WGAN: in questa tipologia di GAN, per calcolare la funzione di perdita, si utilizza la distanza di Wasserstein. Ciò conferisce al sistema una maggiore stabilità nell'addestramento.

Modelli di Generative AI – Reti Generative Avversarie (GAN)

- cycleGAN: questa tipologia di GAN sfrutta l'apprendimento non supervisionato per trasferire delle immagini da un dominio X a un dominio Y.
    - La potenza di questo modello sta nel fatto che non servono dati accoppiati dai due modelli, ma sono sufficienti due dataset indipendenti. Un esempio di utilizzo potrebbe essere la conversione di una foto in disegno stile cartone animato.
- VAE-GAN: questa tipologia prevede di combinare un VAE e una GAN, in questo modo il discriminatore valuta i dati forniti in output dal VAE. In precedenza, avevamo visto che i VAE avevano la problematica di generare immagini sfocate; con questa configurazione ciò viene evitato.
- DCGAN: le deep convolutional GAN integrano nell'architettura le reti neurali convoluzionali. Sono particolarmente indicate nella generazione di immagini.
- SRGAN: le GAN a super risoluzione sono impiegate nell'upscaling di immagini a bassa risoluzione.

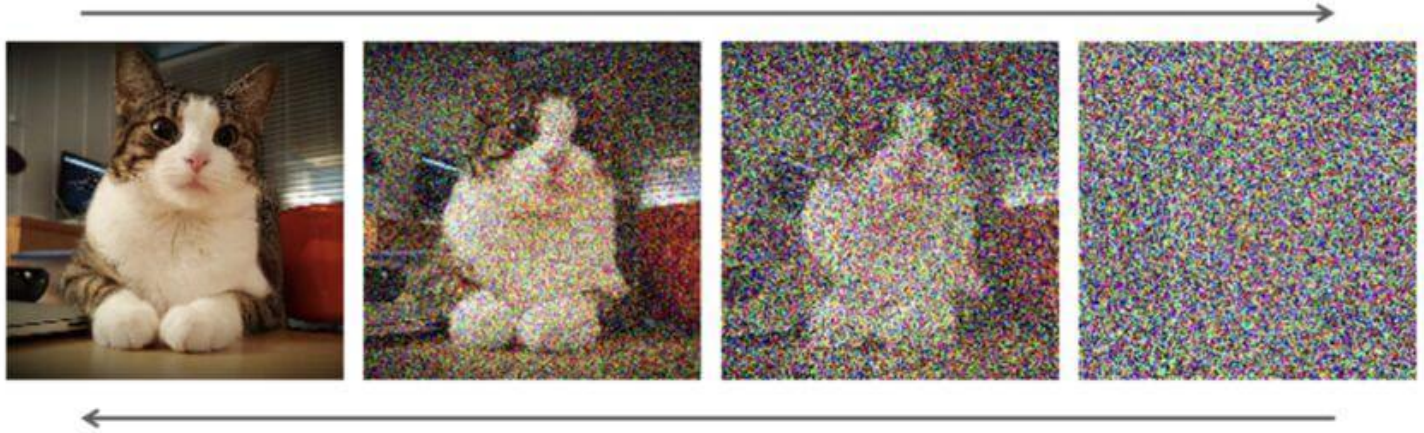
Modelli di Generative AI – Modelli di diffusione

- Una delle nuove frontiere dell'IA generativa è costituita sicuramente dai diffusion model, i quali sono stati introdotti nel 2015 nell'ambito della termodinamica e poi adattati al contesto dell'Intelligenza Artificiale.

- Alla base dei modelli di diffusione c'è il concetto di *denoising*; infatti l'idea è quella di generare un'immagine a partire da rumore casuale.

- È stato dimostrato che i modelli di diffusione basati su un approccio probabilistico erano competitivi con le reti avversarie nel campo della generazione di immagini.

- Alla base dei modelli di diffusione ci sono due processi, visibili nella seguente figura:



Modelli di Generative AI – Modelli di diffusione

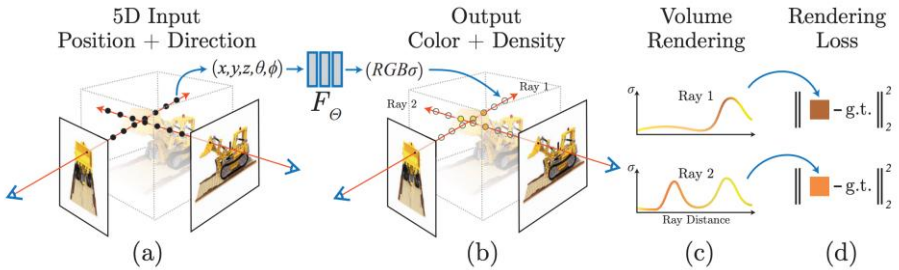
- Diffusione diretta:
  - La diffusione diretta, che avviene durante l'addestramento, consiste nel prendere un'immagine e, ad ogni passo, aggiungere rumore casuale gaussiano fino ad arrivare a un'immagine composta completamente da disturbo.
  - È un processo stocastico basato concettualmente sulle catene di Markov, in cui ogni stato all'istante x_t dipende solo dallo stato x_{t-1}.
  - Il processo di diffusione diretta non richiede addestramento.
- Diffusione inversa:
  - Lo scopo principale di questi modelli è quello di invertire il processo di diffusione diretta in quello che è chiamato processo di diffusione inversa. In questo caso lo scopo è rimuovere ad ogni passo del rumore fino ad arrivare all'immagine desiderata.
  - Tuttavia, è bene osservare come il modello non sia addestrato per stimare l'immagine, bensì per predire il rumore e, dunque, arrivare all'immagine attraverso più passi di rimozione del rumore stimato. Ciò avviene mediante un'architettura chiamata U-NET.

Modelli di Generative AI – Modelli di diffusione

- L'addestramento del modello avviene minimizzando l'errore ad ogni passo.
- Una funzione di errore molto comune è l'MSE (errore quadratico medio), calcolato come quadrato della differenza tra rumore effettivamente misurato ed errore predetto dal modello.
- La minimizzazione dell'errore avviene tramite algoritmi basati sulla discesa del gradiente.

Modelli di Generative AI – Neural Radiance Fields (NeRF)

- I Neural Radiance Field sono stati introdotti nel 2020 come uno strumento per generare delle rappresentazioni fotorealistiche 3D a partire da un insieme di immagini bidimensionali.
- I Neural Radiance Field sfruttano le reti neurali e la computer Graphics per ricostruire una scena in 3D.
- In particolare sfruttano le seguenti proiezioni geometriche:
  - Ray casting: viene analizzata la prospettiva dell'utente per determinare quali oggetti sono visibili e quali no.
  - Ray tracing: si effettua uno studio della luce per determinare ombre, riflessi e rifrazioni.
  - Rasterizzazione: consiste nel proiettare in 2 dimensioni il modello tridimensionale per sintetizzare una nuova vista bidimensionale.
- Il processo attraverso cui operano viene rappresentato nella seguente figura:



Modelli di Generative AI – Neural Radiance Fields (NeRF)

- Come si evince dalla figura, il processo consiste nei seguenti passi:
  - Input 5D: l'input è pentadimensionale; esso, infatti, è composto dalle tre coordinate spaziali di un punto (x, y, z) e da due angoli che indicano la direzione di osservazione (ε, ϕ).
  - Output: l'output della rete neurale è composto da un valore che identifica il colore RGB e da un valore σ che indica la densità volumetrica, ossia un indicatore di quanto un punto è in grado di attenuare la luce.
  - Rendering volumetrico: il rendering volumetrico consiste nel far attraversare alla scena un raggio per ogni pixel e nell'integrare tutti i contributi (colore + densità) ottenuti da ogni punto attraversato dal raggio.
  - Calcolo della perdita: a questo punto si effettua il calcolo della perdita confrontando il colore ottenuto con l'immagine reale. La funzione di perdita più comune, che deve essere minimizzata, è l'errore quadratico medio (MSE).
- I Neural Radiance Field sono utilizzati in moltissimi contesti:
  - Realtà virtuale e aumentata: essi sono particolarmente indicati in questo ambito poiché permettono di generare ambienti tridimensionali. Si pensi, ad esempio, alla generazione di ambienti 3D nel contesto videoludico.

Modelli di Generative AI – Neural Radiance Fields (NeRF)

- Medicina: essi sono utilizzati anche per la diagnostica medica, in quanto, ad esempio, possono essere impiegati nella ricostruzione tridimensionale di esami diagnostici per immagini bidimensionali.

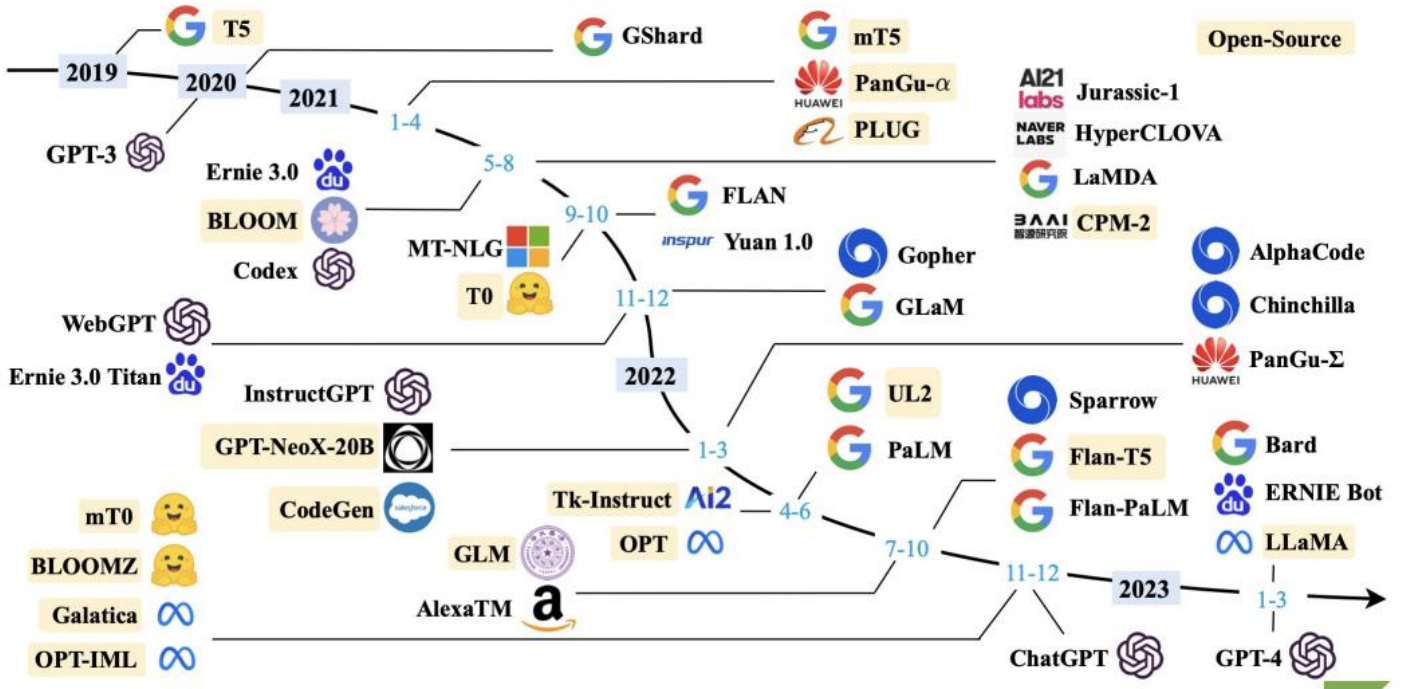
- Immagini satellitari: essi sono anche utilizzati nella ricostruzione tridimensionale della Terra a partire da un dataset di fotografie satellitari.

- Grafica e animazione: essi trovano, inoltre, applicazione anche nell'ambito dell'animazione e della grafica tridimensionale.

I Large Language Models

- I Large Language Model (LLM) rappresentano un progresso significativo nel campo dell'Intelligenza Artificiale, in particolare nella comprensione e nella generazione di testi simili a quelli umani.

- Uno sguardo ai principali LLM tenendo conto della loro comparsa nel tempo viene mostrato nella seguente figura:



Source: Zhao, W. et al., "A Survey of Large Language Models" (2023)

I Large Language Models

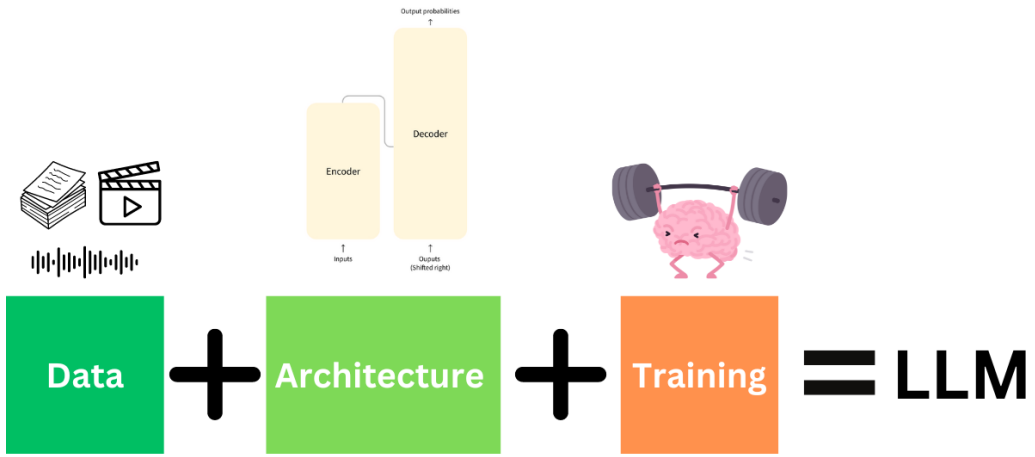
- Gli LLM sono un tipo specifico di modello di deep learning addestrato su enormi set di dati di testo e codice.
- Questi modelli possiedono una conoscenza e una comprensione approfondite del linguaggio, che consentono loro di svolgere varie attività, tra cui:
  - Generazione di testo: essi possono creare testi realistici e coinvolgenti come poesie, frammenti di codice, script e articoli di cronaca.
  - Traduzione: essi possono convertire il testo da una lingua all'altra in modo accurato e fluente.
  - Question answering: essi possono fornire risposte informative ed esaurienti a domande aperte e complesse.
  - Sintesi: essi possono condensare grandi quantità di testo in sintesi concise e informative.

I Large Language Models

- Gli LLM utilizzano in genere architetture basate su Transformer di cui abbiamo parlato prima, basandosi sul concetto di attenzione.
- Ciò consente al modello di concentrarsi sulle parti rilevanti del testo di input quando effettua previsioni e genera output.
- Il processo di addestramento prevede l'alimentazione dell'LLM con enormi set di dati di testo e codice.
- Questi dati aiutano il modello ad apprendere relazioni complesse tra parole e frasi, consentendo ad esso, in ultima analisi, di comprendere e manipolare il linguaggio in modi sofisticati.

I Large Language Models – Componenti di un LLM

- Gli LLM sono costruiti su tre pilastri fondamentali: dati, architettura e addestramento, come mostrato nella seguente figura:



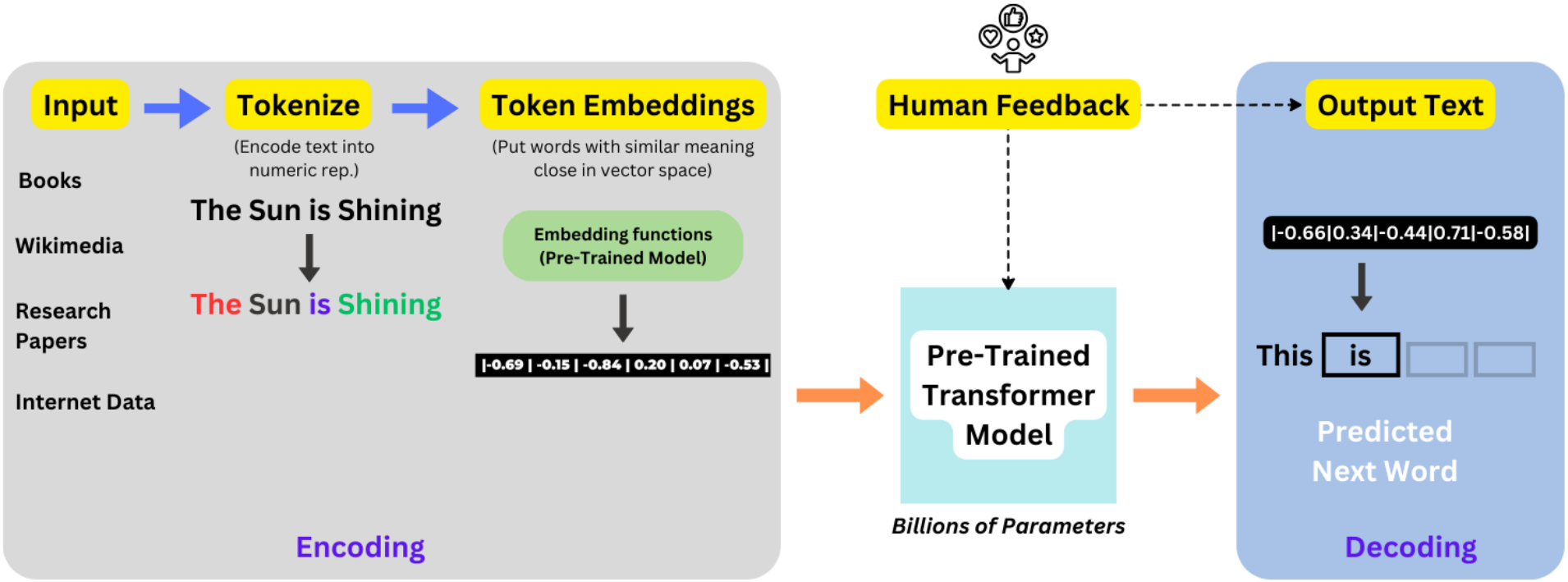
- Dati:
  - Sono la pietra angolare di qualsiasi LLM. La qualità, la diversità e la dimensione del dataset determinano le capacità e i limiti del modello.
  - I dati consistono solitamente in una vasta gamma di testi tratti da libri, siti web, articoli e altre fonti scritte.
  - Questa ampia raccolta aiuta il modello ad apprendere vari modelli linguistici, stili e l'ampiezza della conoscenza umana.

I Large Language Models – Componenti di un LLM

- Architettura:
  - Si riferisce alla struttura sottostante del modello, spesso basata su un'architettura Transformer.
  - I Transformer sono un tipo di rete neurale particolarmente adatta all'elaborazione di dati sequenziali come il testo, che utilizza meccanismi come l'attenzione per valutare l'importanza delle diverse parti dei dati di input. Questa architettura consente al modello di comprendere e generare testi coerenti e contestualmente rilevanti.
- Addestramento:
  - L'ultimo tassello è il processo di addestramento, in cui il modello viene istruito utilizzando i dati raccolti. Durante l'addestramento, il modello regola iterativamente i suoi parametri interni per ridurre al minimo gli errori di previsione.
  - Il processo di addestramento non riguarda solo l'apprendimento della lingua, ma anche la comprensione delle sfumature, del contesto e la capacità di generare risposte creative o innovative.

I Large Language Models – Come apprendono gli LLM?

- Il processo di apprendimento di un LLM viene mostrato nella seguente figura:



- Esso prevede i seguenti passi:
  - Input: il modello parte da un corpus vasto e diversificato di dati testuali provenienti da libri, Wikipedia, articoli di ricerca e vari siti Internet. Questi dati costituiscono la materia prima che il modello utilizza per apprendere i modelli linguistici.

I Large Language Models – Come apprendono gli LLM?

- Tokenizzazione: in questa fase il testo viene tokenizzato, ovvero suddiviso in parti più piccole, spesso parole o sottoparole. La tokenizzazione è essenziale affinché il modello possa elaborare e comprendere i singoli elementi del testo in ingresso.

- Token embeddings: ogni token viene trasformato in una rappresentazione numerica nota come token embedding.

    - Questi embedding sono vettori che codificano il significato dei token in modo tale che le parole con significati simili siano più vicine tra loro nello spazio vettoriale.

- Encoding: gli embedding vengono quindi passati attraverso i livelli di encoding del modello transformer.

    - Questi livelli elaborano gli embedding per comprendere il contesto di ciascuna parola all'interno della frase.

    - Il modello lo fa regolando gli embedding in modo da incorporare le informazioni delle parole circostanti utilizzando meccanismi di auto-attenzione.

I Large Language Models – Come apprendono gli LLM?

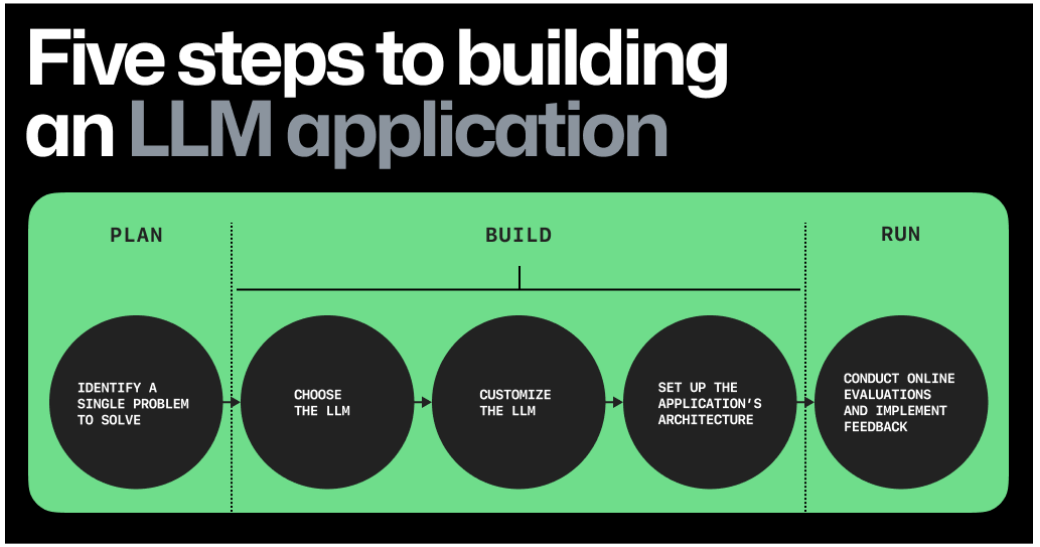
- Modello Transformer pre-addestrato: l'architettura principale del modello è un Transformer, che è stato pre-addestrato sui dati di input.
    - Ha imparato a prevedere parti del testo da altre parti, regolando i suoi parametri interni per ridurre al minimo l'errore di previsione.
    - Questo modello ha livelli di attenzione, reti neurali feed-forward e un gran numero di parametri per catturare le complessità del linguaggio.
- Feedback umano: si tratta di una fase facoltativa ma importante in cui gli esseri umani possono fornire un feedback sui risultati del modello, perfezionandone ulteriormente le prestazioni. Il ciclo di feedback può aiutare ad allineare i risultati del modello alle aspettative e agli standard umani.
- Testo di output: quando genera un testo, il modello utilizza i parametri pre-addestrati e possibilmente perfezionati per prevedere la parola successiva in una sequenza. Ciò comporta il processo di decodifica.

I Large Language Models – Come apprendono gli LLM?

- Decoding: la fase di decoding è quella in cui il modello riconverte gli embedding elaborati in testo leggibile dall'uomo.
  - Dopo aver previsto la parola successiva, il modello decodifica questa previsione dalla sua rappresentazione numerica in una parola.
  - Il processo di decodifica spesso comporta la selezione della parola con la probabilità più alta dalla distribuzione di output del modello.
  - Questa previsione può essere l'output finale oppure può fungere da token di input aggiuntivo per il modello per prevedere le parole successive, consentendo al modello di generare tratti di testo più lunghi.
- Questo ciclo di codifica e decodifica, basato sia sui dati di addestramento originali che sul feedback umano continuo, consente al modello di produrre testi contestualmente pertinenti e sintatticamente corretti.
- In definitiva, ciò può essere utilizzato in una varietà di applicazioni, tra cui l'Intelligenza Artificiale conversazionale, la creazione di contenuti e altro ancora.

Costruire un'applicazione LLM

- I passi ad alto livello che devono essere effettuati per costruire un'applicazione LLM sono riportati nella seguente figura:



- Essi sono i seguenti:
  - Concentrarsi dapprima su un unico problema: trovare un problema delle giuste dimensioni, ovvero abbastanza specifico da poter essere risolto rapidamente e ottenere progressi, ma anche abbastanza grande da stupire gli utenti con la soluzione giusta.

Costruire un’applicazione LLM

- Scegliere l’LLM giusto: a tale proposito alcuni fattori che sicuramente è necessario considerare riguardano il tipo di licenza, i costi e la dimensione del modello.

- Personalizzare l’LLM: quando si addestra un LLM, si sta costruendo l'impalcatura e le reti neurali per consentire il deep learning. Quando si personalizza un LLM pre-addestrato, si sta adattando l'LLM a compiti specifici, come la generazione di testo su un argomento specifico o in uno stile particolare.

- Configurare l'architettura dell'app: i diversi componenti necessari per configurare l'app LLM possono essere suddivisi in tre categorie: input dell'utente, strumenti di arricchimento dell'input e di costruzione dei prompt e strumenti di IA efficienti e responsabili.

- Condurre valutazioni online dell’app: si tratta di valutazioni “online” perché valutano le prestazioni dell'LLM durante l'interazione con l'utente.

Utilizzi degli LLM

- Gli LLM hanno superato i loro limiti iniziali di ricerca e stanno ora trovando applicazioni pratiche in vari settori. Questi potenti modelli, addestrati su enormi dataset di testo e codice, possiedono capacità eccezionali nella comprensione, generazione e manipolazione del linguaggio.

- Ecco alcuni degli interessanti casi d'uso degli LLM:

  - Creazione di contenuti:

    - Articoli di cronaca personalizzati: gli LLM possono generare articoli di cronaca personalizzati in base agli interessi individuali e alle preferenze di lettura.

    - Copywriting di marketing: gli LLM possono creare testi di marketing accattivanti e persuasivi per vari prodotti e servizi.

    - Scrittura creativa: gli LLM possono assistere gli scrittori generando formati di testo creativi come poesie, sceneggiature e brani musicali.

    - Generazione di codice: gli LLM possono generare frammenti di codice e persino interi programmi, automatizzando le attività ripetitive e assistendo gli sviluppatori.

Utilizzi degli LLM

- Analisi del sentiment e approfondimenti sui clienti: gli LLM possono analizzare il feedback dei clienti e i dati dei social media per identificare le tendenze e ottenere informazioni preziose.

- Ricerca e sviluppo:
  - Scoperte scientifiche: gli LLM sono in grado di analizzare enormi quantità di dati di ricerca per identificare modelli e accelerare le scoperte scientifiche.
  - Diagnosi e trattamenti medici: gli LLM possono analizzare cartelle cliniche e immagini per assistere nella diagnosi, raccomandando opzioni di trattamento personalizzate.
  - Revisione e analisi della letteratura: gli LLM possono analizzare rapidamente grandi quantità di letteratura, riassumendo i risultati chiave ed evidenziando i passaggi rilevanti.
- Intrattenimento e media:
  - Esperienze di intrattenimento personalizzate: gli LLM possono personalizzare i consigli di intrattenimento, suggerendo nuova musica e nuovi film in base alle preferenze individuali.
  - Sviluppo di giochi e narrazione: gli LLM possono creare narrazioni e dialoghi interattivi per i videogiochi, migliorando l'immersione e il coinvolgimento dei giocatori.

Utilizzi degli LLM

- Moderazione e filtraggio dei contenuti: gli LLM possono essere utilizzati per rilevare e filtrare i contenuti dannosi online, rendendo lo spazio digitale più sicuro per tutti.

- Questi sono solo alcuni esempi dei diversi casi d'uso degli LLM. Con la continua evoluzione di questa tecnologia e la sua crescente accessibilità, possiamo aspettarci applicazioni ancora più innovative e di grande impatto in vari settori. Grazie alla loro capacità di comprendere, generare e manipolare il linguaggio, gli LLM sono destinati a rivoluzionare il modo in cui interagiamo con le informazioni, nonché il modo in cui apprendiamo, creiamo e lavoriamo.

Limitazioni degli LLM

- Sebbene gli LLM offrano capacità impressionanti nella comprensione e nella generazione del linguaggio, presentano anche dei limiti che possono ostacolare la loro efficacia nelle applicazioni del mondo reale. Questi limiti includono:
  - Contesto mancante: gli LLM possono avere difficoltà a comprendere il contesto più ampio di un determinato compito, portando a risultati imprecisi o irrilevanti.
  - Risultati non personalizzati: gli LLM possono generare risposte generiche che non rispondono alle esigenze o ai requisiti specifici dell'utente o del compito.
  - Vocabolario specialistico limitato: gli LLM addestrati su set di dati generici potrebbero non disporre del vocabolario specialistico necessario per domini o compiti specifici.
  - Allucinazioni: gli LLM a volte possono inventare informazioni o generare risultati distorti, imprecisi o fuorvianti.

Limitazioni degli LLM – Le allucinazioni

- In questa figura viene riportata una classifica delle allucinazioni secondo il modello di valutazione delle allucinazioni di Vectara. GPT-4 e GPT-4 Turbo si sono classificati al primo posto con il tasso di accuratezza più alto (97%) e il tasso di allucinazioni più basso (3%) tra tutti i modelli testati.

Model	Accuracy	Hallucination Rate	Answer Rate	Average Summary Length (Words)
GPT 4	97.0 %	3.0 %	100.0 %	81.1
GPT 4 Turbo	97.0 %	3.0 %	100.0 %	94.3
GPT 3.5 Turbo	96.5 %	3.5 %	99.6 %	84.1
Llama 2 70B	94.9 %	5.1 %	99.9 %	84.9
Llama 2 7B	94.4 %	5.6 %	99.6 %	119.9
Llama 2 13B	94.1 %	5.9 %	99.8 %	82.1
Cohere-Chat	92.5 %	7.5 %	98.0 %	74.4
Cohere	91.5 %	8.5 %	99.8 %	59.8
Anthropic Claude 2	91.5 %	8.5 %	99.3 %	87.5
Google Palm 2 (beta)	91.4 %	8.6 %	99.8 %	86.6
Mistral 7B	90.6 %	9.4 %	98.7 %	96.1
Google Palm 2 Chat (beta)	90.0 %	10.0 %	100.0 %	66.2
Google Palm 2	87.9 %	12.1 %	92.4 %	36.2
Google Palm 2 Chat	72.8 %	27.2 %	88.8 %	221.1

Limitazioni degli LLM – Le allucinazioni

- Le allucinazioni possono essere suddivise nelle seguenti tre categorie:
  - Inesattezze fattuali:
    - Questo tipo di allucinazione si verifica quando un modello linguistico presenta informazioni non vere o corrette, ma formulate come se fossero fattuali.
    - Ciò include date, eventi, statistiche o affermazioni di cui si può verificare la falsità.
    - Ciò può accadere per vari motivi, tra cui l'errata interpretazione dei dati di input, la scarsa qualità dei dati e delle metodologie di addestramento, l'affidamento a fonti obsolete o errate o la combinazione di informazioni provenienti da contesti diversi che porta a un output inesatto.
  - Citazioni o fonti generate:
    - Ciò si verifica quando un modello linguistico inventa citazioni o riferimenti.
    - Ad esempio, il sistema potrebbe generare un'affermazione e attribuirla erroneamente a una persona reale, oppure creare una fonte fittizia che non esiste affatto. Ciò è problematico perché porta a disinformazione, affermazioni attribuite erroneamente e confusione.

Limitazioni degli LLM – Le allucinazioni

- Incoerenze logiche:
  - Ciò include la generazione di risposte internamente incoerenti o logicamente errate.
  - Dopo aver generato una risposta a una query dell'utente, l'LLM può contraddirsi nelle risposte successive.
  - Ciò si verifica quando un modello formula una serie di affermazioni che, nel loro insieme, risultano incoerenti o contraddittorie, mettendo in discussione la credibilità dei suoi risultati.

Limitazioni degli LLM – Le allucinazioni

- In tutti questi casi, il modello linguistico non è intenzionalmente fuorviante, ma mostra i propri limiti per vari motivi che potrebbero includere i dati di addestramento, la qualità dei dati, la data di interruzione delle conoscenze, una messa a punto inadeguata, ecc.

Limitazioni degli LLM – Le allucinazioni – Strategie di mitigazione

- I ricercatori stanno sviluppando vari approcci per garantire che la risposta generata dai modelli LLM sia accurata.
  - Alcune strategie richiedono l'intervento umano, come l'apprendimento per rinforzo attraverso il feedback umano (Reinforcement Learning through Human Feedback - RLHF).
  - Altre richiedono dati nuovi e personalizzati per addestrare il modello, un processo noto come fine-tuning.
  - Il Retrieval Augmented Generation (RAG) consiste nel fornire una fonte di conoscenza esterna al modello LLM.
- Queste tecniche offrono soluzioni preziose per superare i limiti dei modelli LLM e sfruttarne appieno il potenziale. Applicando queste tecniche, possiamo costruire modelli LLM più sensibili al contesto, in grado di generare output personalizzati, dotati di un vocabolario specializzato e in grado di fornire informazioni accurate e affidabili.
- Inoltre, sono allo studio altri approcci promettenti, tra cui:
  - Tecniche di rilevamento e mitigazione dei pregiudizi (bias): queste tecniche mirano a identificare e affrontare i pregiudizi presenti nei dati di addestramento e nei risultati dei modelli LLM.
  - Explainability methods: questi metodi mirano a rendere i processi decisionali dei modelli LLM più trasparenti e comprensibili.

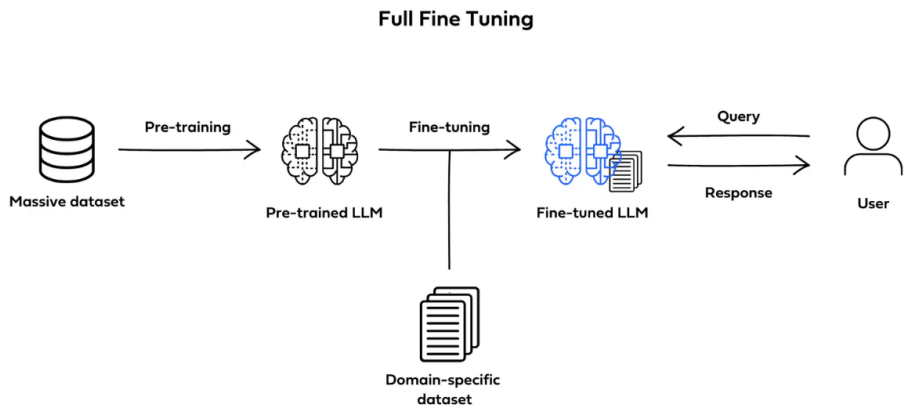
Limitazioni degli LLM – Le allucinazioni – Strategie di mitigazione

  - Sistemi human-in-the-loop: questi sistemi prevedono la supervisione e l'intervento umano per garantire la qualità e l'affidabilità dei risultati dei modelli LLM.

- Migliorando e perfezionando continuamente queste tecniche, possiamo lavorare alla creazione di LLM più robusti e affidabili che apportino benefici alla società in modo responsabile ed etico.

Limitazioni degli LLM – Le allucinazioni – Fine tuning

- Il fine-tuning comporta un ulteriore addestramento di un LLM su un dataset specifico relativo all'attività desiderata.
- Ciò aiuta il modello ad apprendere il vocabolario e le sfumature pertinenti al dominio, portando a risultati più accurati e personalizzati.
- Esempio: il fine-tuning di un LLM su documenti legali può migliorarne la capacità di rispondere con precisione a domande di natura giuridica.
- Il processo di fine-tuning viene mostrato nella seguente figura:

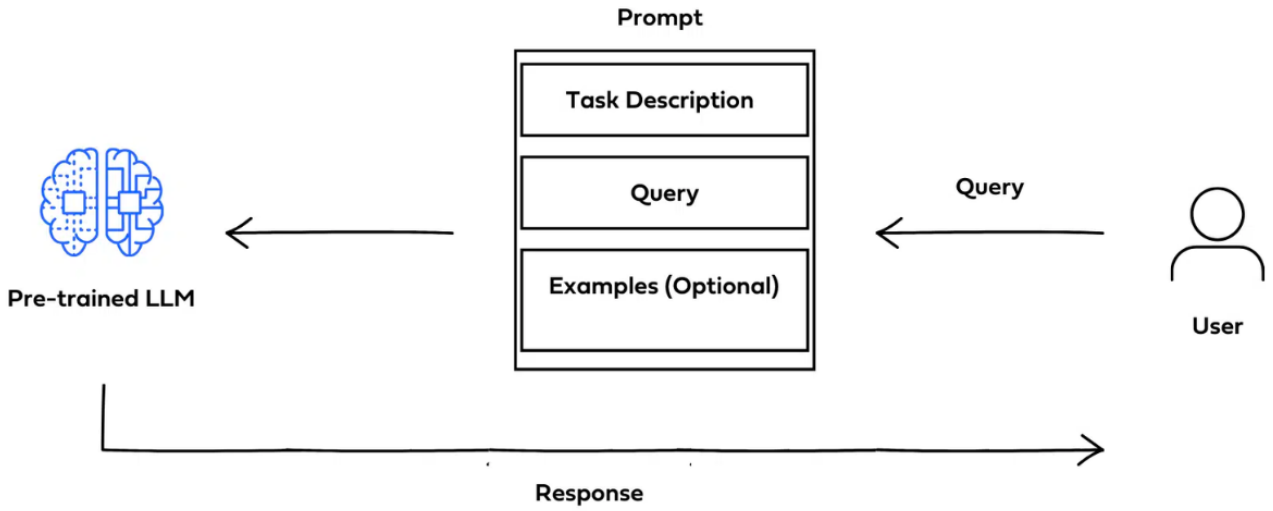


Limitazioni degli LLM – Le allucinazioni – Prompt engineering

- Il prompt engineering consiste nel creare con cura i prompt forniti all'LLM. Fornendo istruzioni e contesti specifici attraverso il prompt, gli utenti possono guidare l'LLM verso i risultati desiderati.

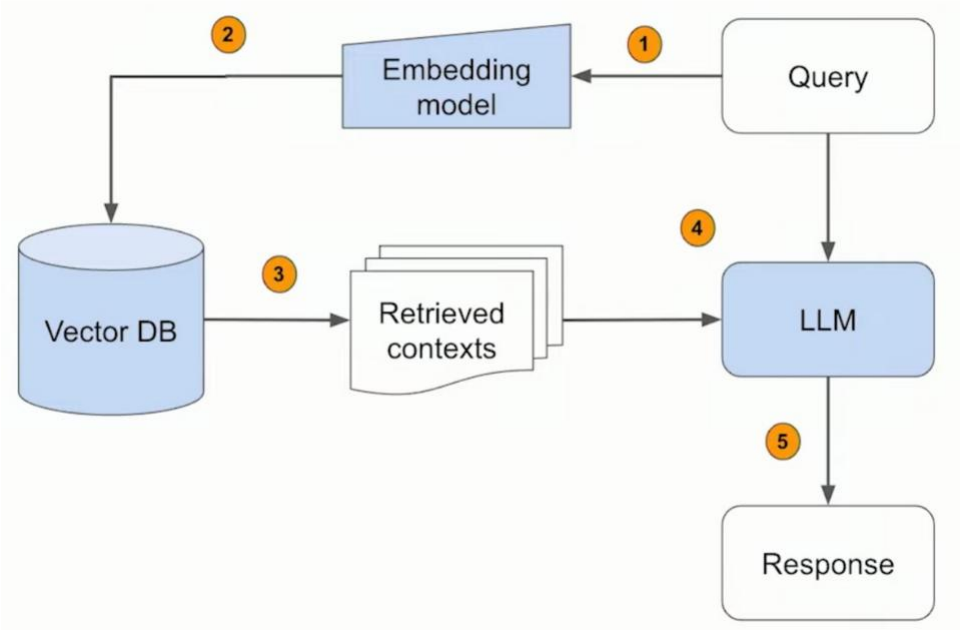
- Esempio: un prompt che chiede all'LLM di "riassumere il seguente articolo di cronaca in 50 parole" produrrà probabilmente un risultato più conciso e informativo rispetto alla semplice fornitura dell'articolo stesso.

- Il processo di prompt engineering viene mostrato nella seguente figura:



Limitazioni degli LLM – Le allucinazioni – RAG

- Il RAG combina approcci basati sul retrieval e approcci generativi.
- Innanzitutto recupera i documenti pertinenti da un ampio corpus in base alla query dell'utente. Quindi utilizza i documenti recuperati come input per generare una risposta nuova e personalizzata.
- Il funzionamento del RAG viene riassunto nella seguente figura:



Limitazioni degli LLM – Le allucinazioni – RAG

- Vediamo più in dettaglio i vari passi del processo:
  - Query: si inizia con una query, ovvero l'input o la domanda che si pone al modello.
  - Modello di embedding: la query viene quindi elaborata da un modello di embedding che la trasforma in un vettore. Questo processo comporta la conversione del testo in forma numerica, catturando il significato semantico in uno spazio ad alta dimensione.
  - Database vettoriale: il vettore della query viene utilizzato per recuperare contesti rilevanti da un ampio database di documenti. Anche questi documenti sono stati convertiti in vettori nello stesso spazio ad alta dimensione.
  - Contesti recuperati: il modello recupera i documenti (contesti) più rilevanti in base alla vicinanza dei loro vettori al vettore della query. Questo viene spesso fatto utilizzando algoritmi di ricerca di tipo nearest neighbor.
  - LLM: i documenti recuperati vengono quindi forniti all'LLM, insieme alla query originale.
  - Risposta: l'LLM genera una risposta basata sia sulla query originale, sia sul contesto aggiuntivo fornito dai documenti recuperati.

Limitazioni degli LLM – Le allucinazioni – RAG

- Esempio: un sistema RAG può cercare articoli scientifici pertinenti sulla base della query di un ricercatore, utilizzando tali articoli per generare un nuovo riassunto informativo.

- Quando si implementa un RAG in produzione, è importante comprendere in che modo le strategie di chunking aiutano a evitare errori.

- In termini tecnici, "chunking" si riferisce alla segmentazione di un corpus di documenti di grandi dimensioni in pezzi più piccoli e più gestibili che possono essere recuperati ed elaborati in modo efficiente dal modello.

- Esistono varie forme di chunking:

  - Naive chunking: significa semplicemente che suddividiamo un documento a intervalli regolari. Ad esempio, dividiamo il testo ogni 200 caratteri, indipendentemente dal fatto che ciò comporti la divisione del testo nel mezzo di una parola o di una frase, ecc.

  - Semantic chunking: quando ci interessa la struttura della frase come costrutto e la suddividiamo solo quando le frasi sono separate come strategia, si parla di suddivisione semantica. Librerie come NLTK e Spacy sono abbastanza potenti da comprendere frasi complesse e suddividerle in parti appropriate.

Limitazioni degli LLM – Le allucinazioni – RAG

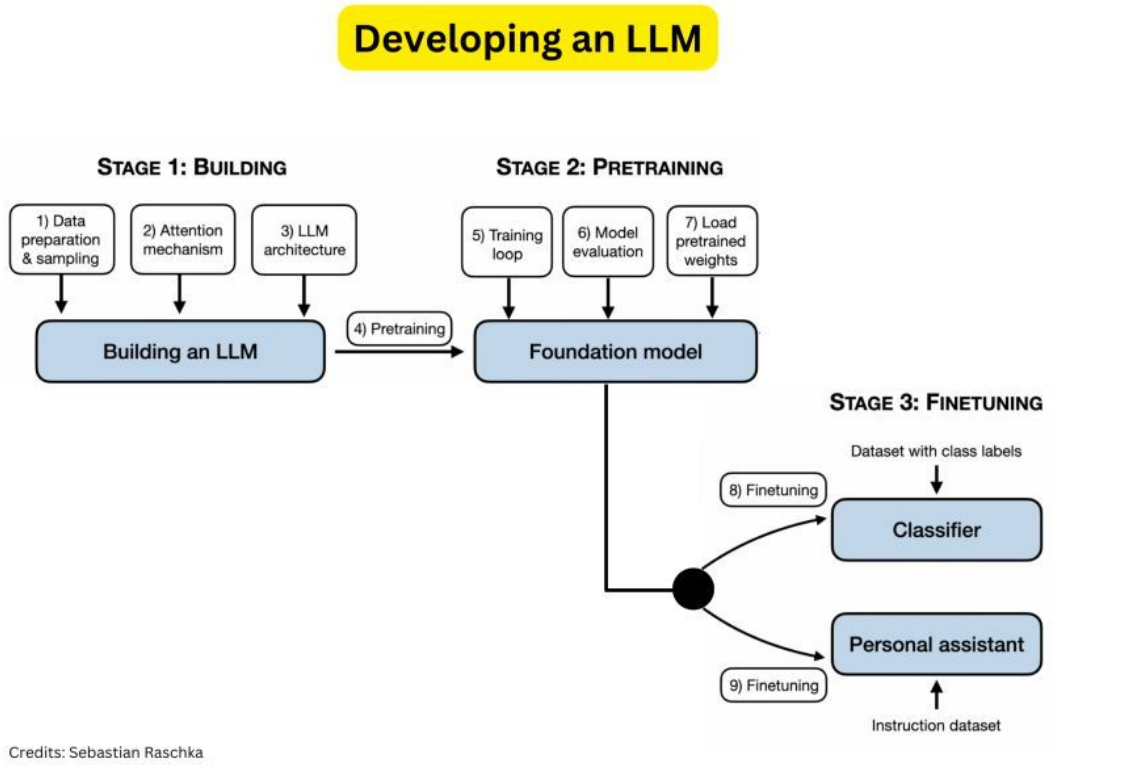
- Compound semantic chunking: si usa quando le frasi sono troppo brevi. In questo caso prima si concatenano le frasi brevi in modo che raggiungano una dimensione adeguata e poi si effettua il chunking semantico sulle frasi risultanti.

- Recursive chunking: quando si ha a che fare con documenti che utilizzano determinati indizi per le nuove righe o la separazione delle frasi, come spazi multipli o caratteri di nuova riga multipli o anche tag HTML multipli, spesso è utile dividere il testo in base a tali indicatori. Questa strategia è utile in ambiti di nicchia, poiché si tratta di una caratteristica del linguaggio che è stata sviluppata dagli utenti come pratica standard.

Sviluppare un Large Language Model

- Gli LLM rappresentano la backbone delle nostre applicazioni di Generative AI ed è molto importante capire cosa serve per crearli.

- Una configurazione molto semplice è quella mostrata in figura, che prevede tre fasi, ovvero Building, Pre-Training e Fine-Tuning.



Sviluppare un Large Language Model

- La fase di building prevede i seguenti task:
  - Preparazione dei dati: comporta la raccolta e la preparazione dei dataset.
  - Architettura del modello: implementazione del meccanismo di attenzione e dell'architettura complessiva.
- La fase di pre-addestramento prevede i seguenti task:
  - Ciclo di addestramento: utilizzo di un ampio dataset per addestrare il modello a prevedere la parola successiva in una frase.
  - Modelli di base: la fase di pre-addestramento crea un modello di base per un ulteriore fine-tuning.
- La fase di fine-tuning prevede i seguenti task:
  - Classificazione: adattamento del modello per compiti specifici come la categorizzazione del testo e il rilevamento dello spam.
  - Fine-tuning delle istruzioni: creazione di assistenti personali o chatbot utilizzando dataset di istruzioni.

Sviluppare un Large Language Model

- I moderni LLM vengono addestrati su vasti dataset, con una tendenza ad aumentarne le dimensioni per ottenere prestazioni migliori.

- Il processo sopra descritto è solo la punta dell'iceberg, ma è un processo molto complesso che porta alla creazione di un LLM.

- Ci vorrebbero ore per spiegarlo, ma basti sapere che lo sviluppo di un LLM comporta la raccolta di enormi dataset testuali, l'utilizzo di tecniche self-supervised per il pre-training su tali dati, lo scaling del modello per avere miliardi di parametri, lo sfruttamento di immense risorse computazionali per l'addestramento, la valutazione delle capacità attraverso benchmark, il fine-tuning per compiti specifici e l'implementazione di vincoli di sicurezza.

Framework e tool di Generative AI

- Il rapido progresso della Generative AI ha stimolato lo sviluppo di vari framework progettati per facilitare la creazione e l'applicazione di questa tecnologia. Questi framework forniscono l'infrastruttura e gli strumenti necessari agli sviluppatori e ai ricercatori per esplorare le possibilità dell'IA generativa.

- Alcuni dei più famosi tool sono:

  - LangChain
  - LlamaIndex
  - Hugging Face
  - Haystack

- Essi verranno illustrati più in dettaglio nelle prossime slide

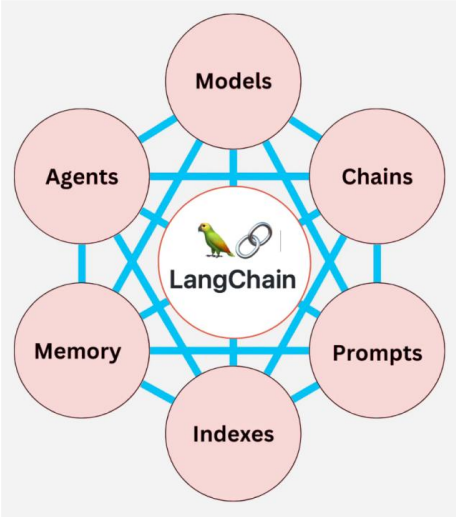
Framework e tool di Generative AI – Langchain

- Sviluppato da Harrison Chase e lanciato nell'ottobre 2022, LangChain è una piattaforma open source progettata per la creazione di applicazioni robuste basate su LLM, tra cui chatbot come ChatGPT e varie applicazioni personalizzate.

- LangChain mira a fornire ai data engineer un toolkit completo per l'utilizzo degli LLM in diversi casi d'uso, tra cui chatbot, risposta automatica alle domande, sintesi di testi e altro ancora.

Framework e tool di Generative AI – Langchain

- LangChain è composta da sei moduli chiave mostrati nella seguente figura:



- Tali moduli sono i seguenti:
  - LLM: LangChain funge da interfaccia standard che consente l'interazione con un'ampia gamma di LLM.
  - Creazione di prompt: LangChain offre una varietà di classi e funzioni progettate per semplificare il processo di creazione e gestione dei prompt.
  - Memoria conversazionale: LangChain incorpora moduli di memoria che consentono la gestione e la modifica delle conversazioni chat passate, una caratteristica fondamentale per i chatbot che devono richiamare le interazioni precedenti.

Framework e tool di Generative AI – Langchain

- Agenti intelligenti: LangChain fornisce agli agenti un kit di strumenti completo. Questi agenti possono scegliere quali strumenti utilizzare in base agli input dell'utente.

- Indici: gli indici in LangChain sono metodi per organizzare i documenti in modo da facilitare un'interazione efficace con gli LLM.

- Chain: sebbene l'utilizzo di un singolo LLM possa essere sufficiente per compiti più semplici, LangChain fornisce un'interfaccia standard e alcune implementazioni comunemente utilizzate per concatenare gli LLM tra loro per applicazioni più complesse, sia tra loro che con altri moduli specializzati.

- LangChain è dotato di funzionalità di memoria, integrazioni con database vettoriali, strumenti per connettersi con fonti di dati esterne, logica e API. Ciò rende LangChain un potente framework per la creazione di applicazioni basate su LLM.

- È possibile installare LangChain utilizzando il seguente comando

```
pip install langchain
```

Framework e tool di Generative AI – LlamaIndex

- LlamaIndex è un framework di orchestrazione avanzato progettato per amplificare le capacità degli LLM come GPT-4.

- Sebbene gli LLM siano intrinsecamente potenti, essendo stati addestrati su vasti set di dati pubblici, spesso non dispongono dei mezzi per interagire con dati privati o specifici di un determinato dominio.

- LlamaIndex colma questa lacuna, offrendo un modo strutturato per acquisire, organizzare e sfruttare varie fonti di dati, tra cui API, database e PDF.

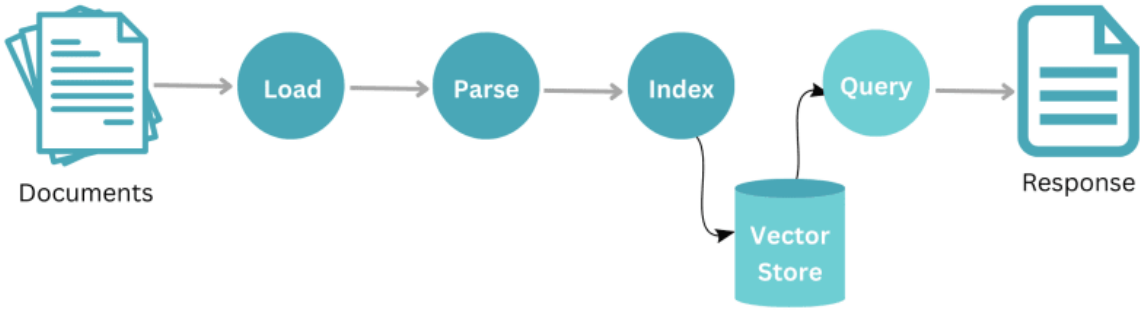
- Indicizzando questi dati in formati ottimizzati per gli LLM, LlamaIndex facilita le query in linguaggio naturale, consentendo agli utenti di interagire senza soluzione di continuità con i propri dati privati senza la necessità di riqualificare i modelli.

- Questo framework è versatile e si rivolge sia ai principianti, con un'API di alto livello per una configurazione rapida, sia agli esperti che cercano una personalizzazione approfondita attraverso API di livello inferiore.

Framework e tool di Generative AI – LlamaIndex

- LlamaIndex funge da ponte, collegando le potenti funzionalità degli LLM con diverse fonti di dati, aprendo così un nuovo mondo di applicazioni in grado di sfruttare la sinergia tra dati personalizzati e modelli linguistici avanzati.

- LlamaIndex opera secondo il seguente workflow:



- Il workflow riceve in input una serie di documenti. Inizialmente, questi documenti vengono sottoposti a un processo di caricamento in cui vengono importati nel sistema.

- Dopo il caricamento, i dati vengono parserizzati per analizzare e strutturare il contenuto in modo comprensibile.

- Una volta parserizzate, le informazioni vengono quindi indicizzate per un recupero e un'archiviazione ottimali.

- Questi dati indicizzati vengono archiviati in modo sicuro in un repository centrale denominato “store”.

Framework e tool di Generative AI – LlamaIndex

- Quando un utente o un sistema desidera recuperare informazioni specifiche da questo archivio dati, può avviare una query.

- In risposta alla query, i dati rilevanti vengono estratti e forniti come risposta, che può essere una serie di documenti pertinenti o informazioni specifiche tratte da essi.

- L'intero processo mostra come LlamaIndex gestisca e recuperi i dati in modo efficiente, garantendo risposte rapide e accurate alle query degli utenti.

Framework e tool di Generative AI – Hugging Face

- Hugging Face è una piattaforma multifunzionale che svolge un ruolo cruciale nel panorama dell'Intelligenza Artificiale, in particolare nel campo del Natural Language Processing (NLP) e della Generative AI.

- Essa comprende vari elementi che operano insieme per consentire agli utenti di esplorare, costruire e condividere applicazioni di IA.

- Ecco una panoramica dei suoi aspetti chiave:

  - Hub dei modelli:

    - Hugging Face ospita un enorme archivio di modelli pre-addestrati per diverse attività di NLP, tra cui classificazione di testi, risposta a domande, traduzione e generazione di testi.

    - Questi modelli sono addestrati su grandi set di dati e possono essere fine-tuned per requisiti specifici, rendendoli prontamente utilizzabili per vari scopi.

    - Ciò elimina la necessità per gli utenti di addestrare i modelli da zero, risparmiando tempo e risorse.

Framework e tool di Generative AI – Hugging Face

- Dataset:
  - Oltre alla libreria di modelli, Hugging Face offre accesso a una vasta raccolta di dataset per attività di NLP.
  - Questi dataset coprono vari ambiti e lingue, offrendo risorse preziose per l'addestramento e il fine-tuning dei modelli.
  - Gli utenti possono anche contribuire con i propri dataset, arricchendo le risorse della piattaforma e favorendo la collaborazione della comunità.
- Strumenti di addestramento e fine-tuning dei modelli:
  - Hugging Face offre strumenti e funzionalità per l'addestramento e il fine-tuning di modelli esistenti su dataset e attività specifici.
  - Ciò consente agli utenti di adattare i modelli alle loro esigenze specifiche, migliorandone le prestazioni e l'accuratezza nelle applicazioni mirate.
  - La piattaforma offre opzioni flessibili per l'addestramento, compreso l'addestramento locale su macchine personali o soluzioni basate su cloud per modelli più grandi.

Framework e tool di Generative AI – Hugging Face

- Creazione di applicazioni:
  - Hugging Face facilita lo sviluppo di applicazioni di IA integrandosi perfettamente con librerie di programmazione popolari come TensorFlow e PyTorch.
  - Ciò consente agli sviluppatori di creare chatbot, strumenti di generazione di contenuti e altre applicazioni basate sull'IA utilizzando modelli pre-addestrati.
  - Sono disponibili numerosi modelli di applicazioni e tutorial per guidare gli utenti e accelerare il processo di sviluppo.
- Comunità e collaborazione:
  - Hugging Face vanta una vivace comunità di sviluppatori, ricercatori e appassionati di IA.
  - La piattaforma favorisce la collaborazione attraverso funzionalità quali la condivisione di modelli, repository di codice e forum di discussione.
  - Questo ambiente collaborativo facilita la condivisione delle conoscenze, accelera l'innovazione e promuove il progresso delle tecnologie NLP e di IA generativa.

Framework e tool di Generative AI – Haystack

- Haystack può essere classificato come un framework end-to-end per la creazione di applicazioni basate su varie tecnologie NLP, tra cui, a titolo esemplificativo ma non esaustivo, l'IA generativa.

- Sebbene non si concentri direttamente sulla creazione di modelli generativi da zero, fornisce una piattaforma robusta per:
  - Retrieval-Augmented Generation (RAG): Haystack eccelle nel combinare approcci retrieval-based e generativi per la ricerca e la creazione di contenuti.
    - Esso consente di integrare varie tecniche di retrieval, tra cui la ricerca vettoriale e la tradizionale ricerca per parole chiave, per recuperare documenti pertinenti per un'ulteriore elaborazione.
    - Questi documenti servono poi come input per i modelli generativi, con il risultato di output più mirati e contestualmente pertinenti.

Framework e tool di Generative AI – Haystack

- Componenti NLP diversificati: Haystack offre una serie completa di strumenti e componenti per varie attività NLP, tra cui la pre-elaborazione dei documenti, la sintesi dei testi, la risposta alle domande e il riconoscimento delle entità denominate.
    - Ciò consente di creare pipeline complesse che combinano più tecniche NLP per raggiungere obiettivi specifici.
- Flessibilità e open source: Haystack è un framework open source basato su librerie NLP popolari come Transformers ed Elasticsearch. Ciò consente la personalizzazione e l'integrazione con strumenti e flussi di lavoro esistenti, rendendolo adattabile a diverse esigenze.
- Scalabilità e prestazioni: Haystack è progettato per gestire in modo efficiente grandi dataset e grandi carichi di lavoro. Si integra con potenti database vettoriali come Pinecone e Milvus, consentendo ricerche e recuperi rapidi e accurati anche con milioni di documenti.
- Integrazione dell'IA generativa: Haystack si integra perfettamente con modelli generativi popolari come GPT-4 e BART. Ciò consente agli utenti di sfruttare la potenza di questi modelli per attività come la generazione di testo, la sintesi e la traduzione all'interno delle loro applicazioni basate su Haystack.

Framework e tool di Generative AI – Haystack

- Sebbene Haystack non si concentri esclusivamente sull'IA generativa, fornisce una solida base per la creazione di applicazioni che sfruttano questa tecnologia. I suoi punti di forza combinati in termini di retrieval, componenti NLP diversificati, flessibilità e scalabilità lo rendono un framework prezioso per sviluppatori e ricercatori che desiderano esplorare il potenziale dell'IA generativa in varie applicazioni.

Tassonomia dei principali sistemi di AI Generativa

- Un modo per classificare i sistemi di AI Generativa considera il tipo di dati che essi elaborano.
- Tale metodo nel futuro potrebbe essere parzialmente superato, dal momento che spesso si stanno proponendo sistemi multimodali.
- Sulla base di tale tassonomia abbiamo:
  - Sistemi di AI Generativa per l'elaborazione dei testi:
    - ChatGPT di OpenAI
    - Gemini di Google
    - Claude di Anthropic
    - Copilot di Microsoft
    - DeepSeek
    - BLOOM di Hugging Face

Tassonomia dei principali sistemi di AI Generativa

- Falcon del Technology Innovation Institute degli Emirati Arabi Uniti
- DeepL (ma solo per la traduzione automatica)
- Llama di Meta
- Mistral AI
- Grok di xAI
- Qwen di Alibaba

Tassonomia dei principali sistemi di AI Generativa

- Sistemi di AI Generativa per la generazione di immagini :
  - Dall-E di OpenAI
  - Midjourney
  - FLUX di Black Forest Labs
  - Firefly di Adobe
  - Imagen di Google
  - Stable Diffusion
  - RunwayML

Tassonomia dei principali sistemi di AI Generativa

- Sistemi di AI Generativa per la generazione di video :
  - CogVideo
  - First Order Motion Model
  - Kling
  - Make-A-Video di Meta
  - MoCoGAN
  - RunwayML
  - Sora di OpenAI
  - Synthesia
  - Veo di Google
  - Elai

Tassonomia dei principali sistemi di AI Generativa

- Sistemi di AI Generativa per la generazione di video :
  - LTX Studio

Tassonomia dei principali sistemi di AI Generativa

- Sistemi di AI Generativa per la generazione di musica e audio:
  - AIVA
  - AudioCraft e EnCodec di Meta
  - Suno AI
  - Lyria e Lyria RealTime di Google
  - Jukedeck
  - Musenet di OpenAI
  - Sonantic
  - SoundStream di Google
  - Spleeter di Deezer
  - Loudly

Tassonomia dei principali sistemi di AI Generativa

- Sistemi di AI Generativa per la generazione di musica e audio:
  - Beethoven
  - Eleven Music
  - MusicGen
  - Udio
  - SpeechT5 di Microsoft
  - Tacotron 2
  - VALL-E di Microsoft
  - Whisper

Tassonomia dei principali sistemi di AI Generativa

- Sistemi di AI Generativa per il riconoscimento e la sintesi vocale:
  - Amazon Polly
  - ElevenLabs
  - IBM Watson Text to Speech e Speech to Text
  - Tacotron di Google
  - Google Cloud Text-to-Speech
  - SpeechT5 di Microsoft
  - VALL-E di Microsoft
  - Lovo
  - Murf
  - Resemble

Tassonomia dei principali sistemi di AI Generativa

- Sistemi di Generative AI per la generazione di codice:
  - CodeWhisperer e Amazon Q di Amazon
  - AskCodi
  - Bolt
  - AlphaCode di Google
  - Codex di OpenAI
  - Code T5 di SalesForce
  - GitHub Copilot di OpenAI e Microsoft
  - Tabnine
  - PolyCoder

Tassonomia dei principali sistemi di AI Generativa

- Sistemi di Generative AI per la generazione di modelli 3D:
  - DreamFusion di Google
  - GET3D di Nvidia
  - Kaedim
  - Luma-Genie AI
  - Instant NeRF di Nvidia
  - Rodin Hyper3D di Deemos Tech
  - Meshy AI di Nvidia e Microsoft
  - Adam

Tassonomia dei principali sistemi di AI Generativa

- Sistemi di Generative AI per la generazione di dati sintetici:
  - Fabric di Microsoft
  - Gretel AI di Nvidia
  - Hazy
  - Mostly AI
  - Synthea
  - Synthetic Data Vault
  - Tonic AI

Tassonomia dei principali sistemi di AI Generativa

- Sistemi di Generative AI per la generazione di contenuti interattivi:
  - InWorld AI (elementi di videogiochi, filmati e metaverso)
  - Convai (elementi di videogiochi, filmati e metaverso)
  - Scenario (elementi di videogiochi, filmati e metaverso)
  - AI Dungeon (videogiochi)
  - Charisma.ai (personaggi virtuali realistici)

Tassonomia dei principali sistemi di AI Generativa

- Sistemi di Generative AI per la comprensione, l'organizzazione e la sintesi di contenuti:
  - NotebookLM di Google
  - AI Summarizer
  - Eightify
  - Notta
  - Scholarcy

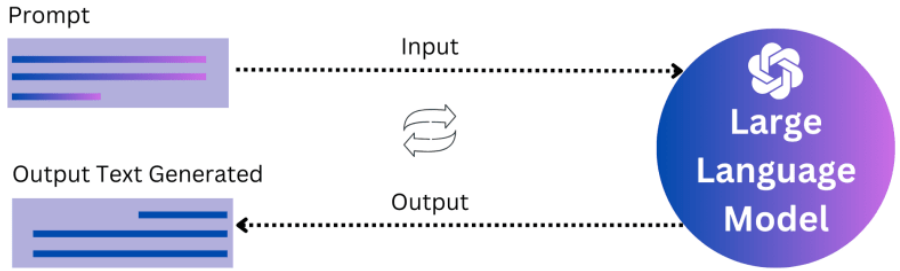
Tassonomia dei principali sistemi di AI Generativa

- Sistemi di Generative AI per la ricerca sul Web:
  - Perplexity AI
- Sistemi di Generative AI per la realizzazione di slide:
  - Gamma AI

Prompt Engineering

- Il prompt engineering è l'arte e la scienza di creare prompt efficaci per guidare e controllare i risultati dei modelli di IA generativa.

- Questi prompt fungono da istruzioni, fornendo contesto e obiettivi specifici al modello, influenzando in ultima analisi la qualità e la direzione dei suoi risultati.



- Alcuni aspetti chiavi del prompt engineering sono i seguenti:

  - Creazione di istruzioni chiare e concise: il prompt deve comunicare chiaramente il risultato desiderato al modello, evitando ambiguità e informazioni superflue.

  - Specificazione del contesto e delle informazioni di base: fornire il contesto e le informazioni di base pertinenti aiuta il modello a comprendere il significato inteso del prompt e a generare output più accurati e pertinenti.

Prompt Engineering

- Utilizzo di diverse tecniche di prompting: è possibile utilizzare varie tecniche come il prefix tuning, il temperature control e il prompt chaining per raffinare il comportamento del modello e ottenere effetti specifici.

- Raffinamento iterativo: il prompting è un processo iterativo; analizzando i risultati del modello e perfezionando i prompt sulla base del feedback, gli utenti possono ottenere risultati ottimali.

- I principali benefici del prompt engineering sono i seguenti:

  - Migliore qualità e pertinenza dei risultati: prompt efficaci possono migliorare significativamente la qualità e la pertinenza dei risultati del modello, garantendo che siano in linea con gli obiettivi desiderati.

  - Controllo sulla direzione creativa: gli utenti possono guidare la direzione creativa del modello e influenzare lo stile, il tono e la direzione generale dei contenuti generati.

  - Esplorazione di nuove possibilità: i prompt aprono le porte all'esplorazione di nuove possibilità e applicazioni della Generative AI, ampliando i confini di ciò che questi modelli possono realizzare.

  - Personalizzazione per compiti specifici: i prompt possono essere adattati a compiti e ambiti specifici, consentendo agli utenti di sfruttare efficacemente la Generative AI per vari scopi.

Prompt Engineering

- Alcuni esempi di applicazioni del prompt engineering sono i seguenti:
  - Generazione di contenuti: creazione di prompt per generare formati di testo creativi come poesie, sceneggiature, codici, brani musicali, testi di marketing, articoli di cronaca, descrizioni di prodotti, ecc.
  - Risposta a domande: formulazione di prompt per guidare il modello verso la generazione di risposte complete e informative a domande aperte e complesse.
  - Generazione di immagini: è possibile fornire prompt descrittivi per istruire il modello su caratteristiche e dettagli specifici da includere nell'immagine generata.
  - Generazione di codice: utilizzo di prompt per guidare il modello nella generazione di frammenti di codice o persino di interi programmi basati su funzionalità e requisiti specifici.
- Alcune sfide che devono essere affrontate dal prompt engineering sono le seguenti:
  - Pregiudizi e imparzialità: i prompt possono ereditare i pregiudizi presenti nei dati di addestramento, portando a risultati distorti o iniqui. Per mitigare questi rischi è fondamentale elaborare e valutare attentamente i prompt.

Prompt Engineering

- Explainability e trasparenza: comprendere come i prompt influenzano il processo decisionale del modello può essere difficile. Sono in corso sforzi per sviluppare metodi per spiegare e interpretare i risultati del modello.

- Overfitting e mancanza di creatività: prompt eccessivamente specifici possono limitare la creatività del modello e portare a risultati ripetitivi o poco originali. Trovare un equilibrio tra guida e libertà è essenziale per ottenere risultati ottimali.

- Il prompt engineering è uno skill fondamentale per massimizzare il potenziale dell'IA generativa. Padroneggiando quest'arte, gli utenti possono sbloccare nuove possibilità, generare risultati creativi e di valore e ampliare i confini dell'intelligenza artificiale.

- Con la continua evoluzione del campo della Generative AI, il prompt engineering svolgerà un ruolo sempre più importante nel plasmare il futuro di questa tecnologia trasformativa.

Utilizzo responsabile della Generative AI

- L'uso responsabile della Generative AI implica il rispetto delle linee guida etiche e delle best practices per garantire che le sue applicazioni siano sicure, eque e vantaggiose.
- Ciò include il rispetto dei diritti di proprietà intellettuale, evitando di generare materiale protetto da copyright o marchio registrato senza autorizzazione.
- Le questioni relative alla privacy devono essere affrontate evitando di creare o condividere dati personali senza consenso.
- È fondamentale evitare di generare contenuti dannosi o offensivi, inclusi deepfake o materiali che potrebbero incitare alla violenza o alla discriminazione.
- Garantire la trasparenza sull'uso dei contenuti generati dall'IA è importante per mantenere la fiducia e prevenire la disinformazione.
- Inoltre, i creatori dovrebbero essere consapevoli del perpetuarsi dei pregiudizi e sforzarsi di utilizzare l'IA in modi che promuovano la diversità e l'inclusività.
- L'uso responsabile della Generative AI implica anche tenersi informati sull'evoluzione degli standard etici e delle normative legali in questo campo in rapida evoluzione.

Utilizzo responsabile della Generative AI

- Le best practices per un uso responsabile della Generative AI riguardano i seguenti fattori:
  - Dati:
    - Utilizzare dataset diversificati e rappresentativi: per evitare distorsioni, addestrare i modelli di IA su dati che riflettono il mondo reale in termini di dati demografici, contesti e prospettive.
    - Ridurre al minimo la raccolta e la conservazione dei dati: raccogliere e conservare solo i dati necessari allo scopo previsto e attuare politiche di cancellazione sicura per i dati obsoleti.
    - Proteggere la privacy: ottenere il consenso informato prima di utilizzare i dati personali e garantirne la conservazione e il trattamento in modo sicuro.
  - Sviluppo e implementazione:
    - Progettare per garantire trasparenza e comprensibilità: rendere i modelli di IA e i loro risultati comprensibili agli utenti, consentendo loro di valutare potenziali pregiudizi o limitazioni.
    - Implementare solide misure di sicurezza: integrare filtri e meccanismi per impedire la generazione di contenuti dannosi come disinformazione, deepfake o materiale offensivo.

Utilizzo responsabile della Generative AI

    - Condurre test e valutazioni approfonditi: testare rigorosamente i sistemi di IA per verificare la presenza di pregiudizi, l'equità e la sicurezza prima dell'implementazione e monitorare continuamente le loro prestazioni in contesti reali.

- Contenuti e interazione con gli utenti:

    - Eichettare chiaramente i contenuti generati dall'IA: differenziare i contenuti generati dall'IA da quelli creati dall'uomo, evitando confusione e potenziali abusi.

    - Promuovere l'autonomia e il controllo degli utenti: fornire agli utenti opzioni per modificare, rifiutare o segnalare contenuti generati dall'IA che siano inappropriati o dannosi.

    - Combattere la disinformazione e le informazioni errate: implementare meccanismi di verifica e controllo dei fatti per garantire l'accuratezza e l'affidabilità dei contenuti generati dall'IA.

- Impatto sociale e governance:

    - Impegnarsi in un dialogo aperto e nella collaborazione: promuovere la discussione e il dibattito sulle implicazioni etiche della Generative AI con diversi soggetti interessati, tra cui il pubblico, i responsabili politici e i ricercatori.

Utilizzo responsabile della Generative AI

- Sviluppare linee guida e framework etici: stabilire principi chiari e best practices per lo sviluppo e l'uso responsabile dell'IA e incoraggiarne l'adozione da parte degli sviluppatori e delle organizzazioni.

- Promuovere la consapevolezza e l'educazione del pubblico: educare il pubblico sulla Generative AI, le sue capacità, i suoi limiti e i suoi potenziali rischi, promuovendo un uso responsabile e un processo decisionale informato.

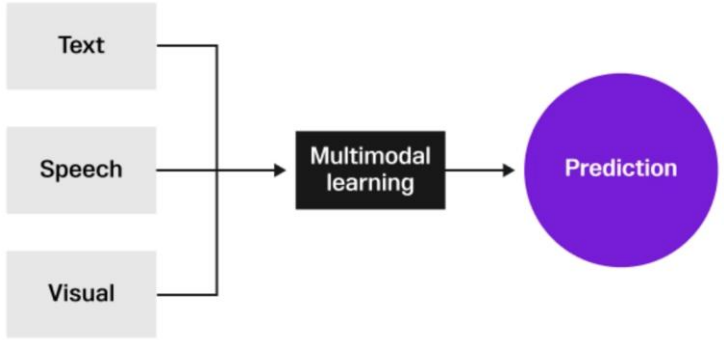
- L'uso responsabile dell'IA generativa è un processo continuo. Attraverso l'apprendimento, l'adattamento e la collaborazione costanti, possiamo garantire che questa potente tecnologia apporti benefici a tutti in modo sicuro, equo ed etico.

Modelli multimodali

- Nel contesto del Machine Learning e dell'Intelligenza Artificiale, i modelli multimodali sono sistemi progettati per comprendere, interpretare o generare informazioni da più tipi di input di dati o "modalità".

- Queste modalità possono includere testo, immagini, audio, video e talvolta anche altri dati sensoriali come il tatto o l'olfatto.

- La caratteristica fondamentale dei modelli multimodali è la loro capacità di elaborare e correlare informazioni provenienti da questi diversi tipi di dati, consentendo loro di svolgere compiti che sarebbero difficili o impossibili per modelli limitati a una singola modalità.



- Il contesto gioca un ruolo fondamentale nei modelli LLM quando un utente effettua una ricerca.

- Mentre i modelli di IA tradizionali spesso faticano a superare i limiti dei singoli tipi di dati, l'aggiunta del contesto diventa un po' più sfidante, con conseguente riduzione dell'accuratezza dei risultati.

Modelli multimodali

- I modelli multimodali risolvono questo problema prendendo in considerazione dati provenienti da più fonti, tra cui testo, immagini, audio e video.

- Questa prospettiva più ampia consente un maggiore contesto e un autoapprendimento più efficace.

- Inoltre, questa comprensione migliorata porta a una maggiore accuratezza, a un processo decisionale più efficace e alla capacità di affrontare compiti che in precedenza erano difficili.

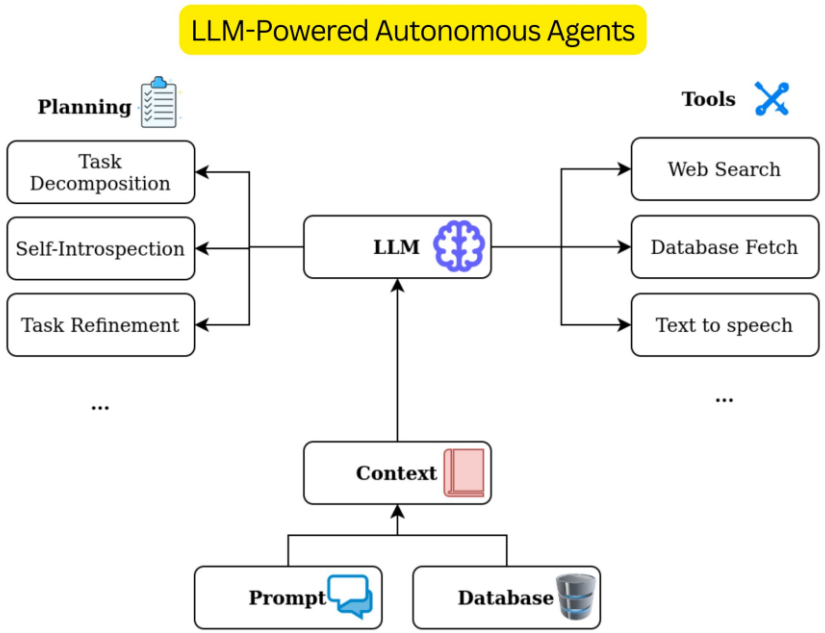
Agenti autonomi potenziati con gli LLM

- Un agente basato su LLM interagisce con il proprio ambiente attraverso la percezione, rilevando i dati ambientali, e agisce sulla base delle informazioni raccolte, che possono coinvolgere anche strumenti.

- In modalità unimodale, l'agente utilizza solo testo sia per l'input che per l'output.

- In modalità multimodale, l'agente è in grado di percepire utilizzando input visivi, uditivi e fisici e può eseguire azioni concrete nell'ambiente.

- La seguente figura mostra un'architettura per un agente potenziato con un LLM:



Agenti autonomi potenziati con gli LLM

- Gli agenti autonomi basati su LLM dimostrano le seguenti caratteristiche di autonomia:
  - Capacità di pianificazione: questi agenti accettano istruzioni o obiettivi di alto livello in linguaggio naturale e li suddividono in compiti più piccoli, pianificano la sequenza di esecuzione, valutano i risultati e li perfezionano per soddisfare l'istruzione o l'obiettivo nel modo più corretto possibile.
  - Capacità di utilizzare strumenti: questi agenti dimostrano la capacità di comprendere la sintassi e la semantica delle chiamate software, selezionare gli strumenti software necessari per qualsiasi attività ed eseguirli fornendo parametri sintatticamente e semanticamente corretti.
    - Questi strumenti possono essere altri agenti basati su LLM, agenti intelligenti non basati su LLM, Application Programming Interfaces (API) esterne, recuperatori da fonti di dati private o semplici funzioni che eseguono alcune logiche di elaborazione dei dati.
  - Capacità di utilizzare informazioni contestuali: infine, tali agenti possono adattare la loro pianificazione e le loro azioni sulla base delle informazioni contestuali presenti nei prompt (chiamato anche in-context learning) o in fonti di dati esterne come i database vettoriali (noti come Retrieval-Augmented Generation).