Large Language Models e Chatbot

Prof.ssa Tiziana D’Alessandro

Prof. Cesare Davide Pace

MEMBER OF

EUROPEAN UNIVERSITY OF TECHNOLOGY

ARTIFICIAL INTELLIGENCE AND DATA ANALYSIS | lab

26/02/2026

00 — Topics

- Cosa sono i Large Language Models e come funzionano 01
- Generazione ed integrazione di Contenuti Multimediali 02
- ChatBot e paradigma RAG 03
- Modelli LLM e Prompt Engineering 04
- Strumenti 05

01

Cosa sono i Large Language Models e come funzionano

Differenza tra Chatbot ed LLM

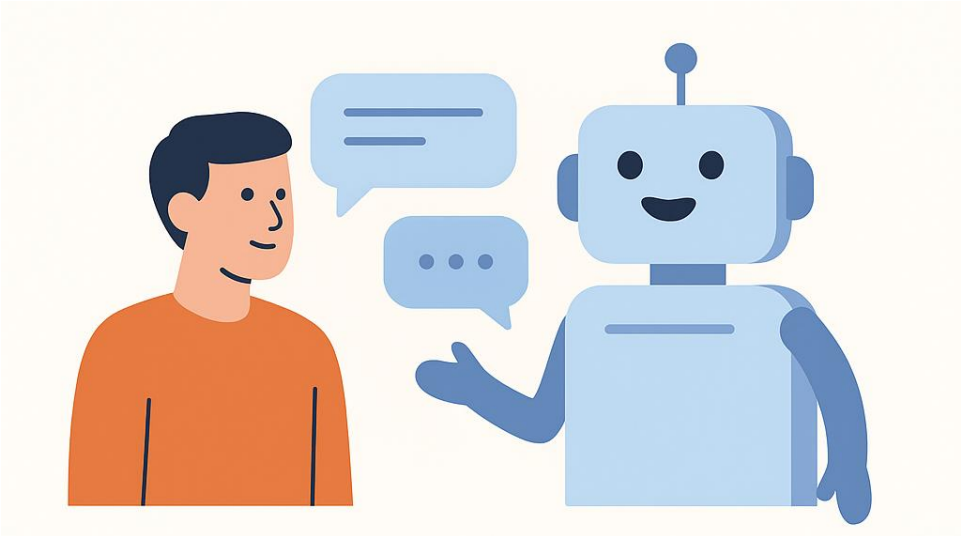
Il **Chatbot** è la carrozzeria dell'auto, mentre il **LLM** è il motore.

Chatbot: È l'interfaccia. È il programma con cui interagisci (come ChatGPT). È lo strato software che permette all'utente di inviare un messaggio e ricevere una risposta.

LLM (Large Language Model): È il "cervello". È un modello matematico gigante (come GPT-4 o Claude) che ha imparato a prevedere le parole e a comprendere il linguaggio analizzando miliardi di testi.

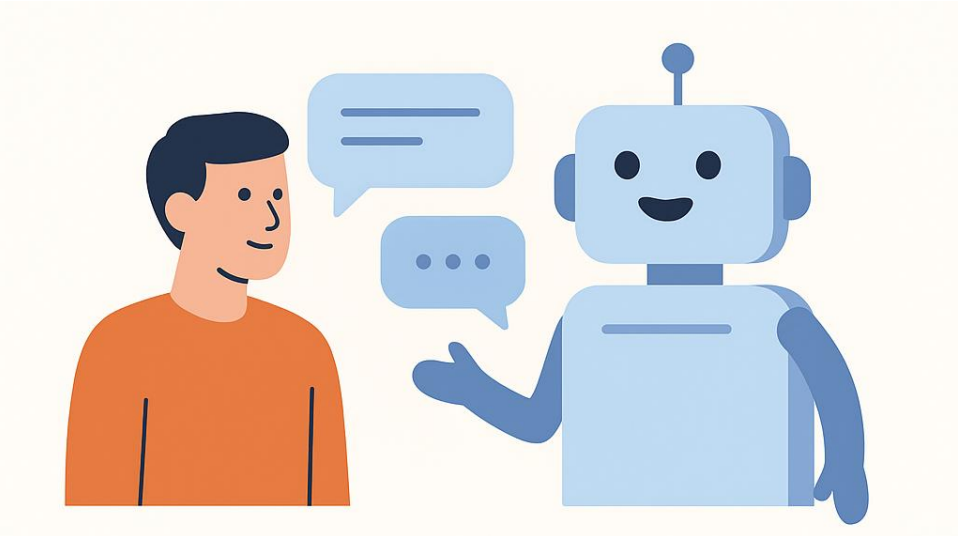
Cos’è un chatbot

Un **chatbot** è un sistema software che simula una conversazione (testo/voce) con un utente per fornire informazioni, eseguire compiti o intrattenere. Oggi i chatbot vanno dal semplice automa che risponde a script fissi fino ad agenti generativi che producono risposte nuove e contestuali.



Caratteristiche dei chatbot

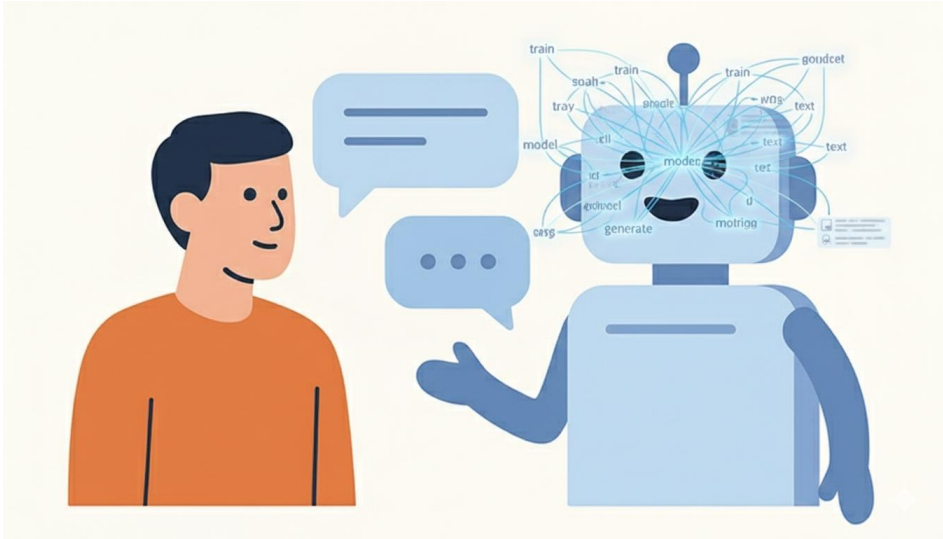
- **Interattività**: ricorda il contesto della conversazione
- **Tipologia d'uso**: task-oriented (customer care, prenotazioni) o open-domain (conversazione libera e creativa, es. ChatGPT)
- **Interfacce**: testo, voce, multimodale
- **Tecnologia**: regole, machine learning tradizionale, modelli generativi



Cos’è un LLM

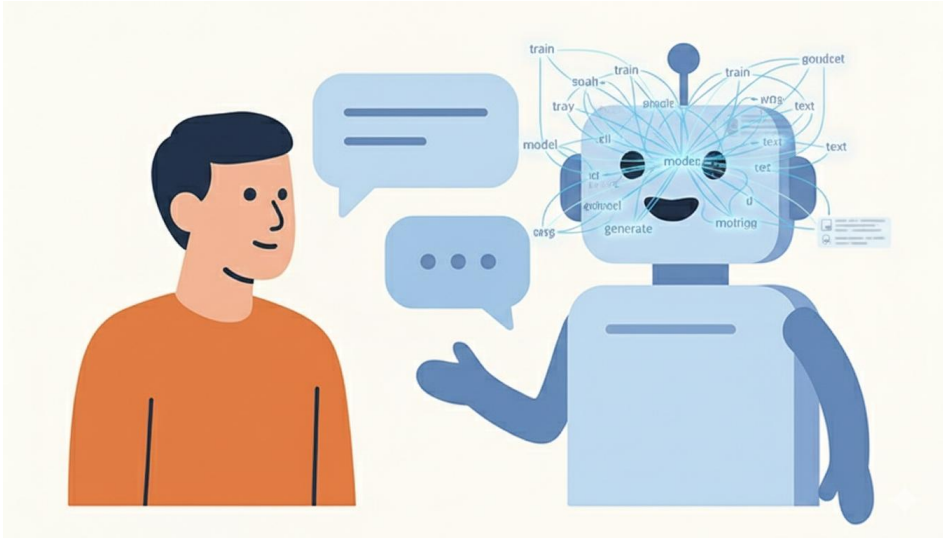
Un **LLM** è un tipo di intelligenza artificiale addestrata per comprendere, generare e manipolare il linguaggio umano in modo estremamente naturale.

"Large": Addestrati su petabyte di dati (libri, codice, pagine web).
"Language": Specializzati nel comprendere e generare linguaggio naturale.
"Model": Una rappresentazione matematica complessa della probabilità linguistica.



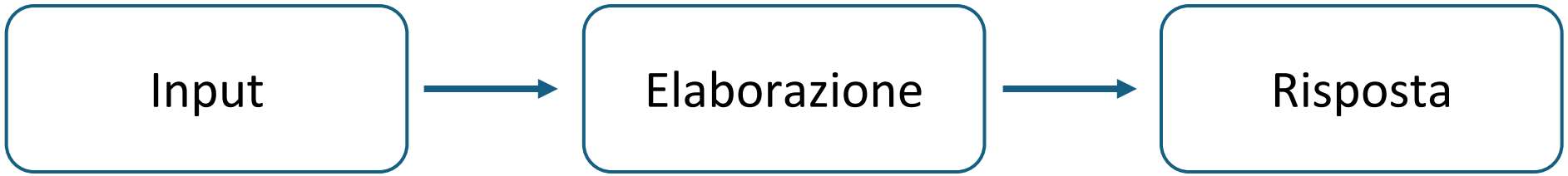
Caratteristiche degli LLM

- "**Large**": Si riferisce sia alla quantità enorme di dati testuali impiegati per l'addestramento, sia al numero di parametri (miliardi) che compongono la sua struttura neurale.
- **Architettura Transformer**: consente al modello di analizzare le relazioni tra le parole all'interno di una frase, comprendendone il contesto.
- **Previsione Statistica**: Data una sequenza di parole, calcola la probabilità statistica che la parola successiva sia la più coerente.



Anatomia di un LLM

Overview



Architettura Transformer (2017)

- Abbandono della lettura sequenziale per quella parallela.
- Meccanismo di Attention: Capacità del modello di pesare l'importanza di diverse parole in una frase, indipendentemente dalla distanza.

Esempio: Nella frase "L'animale non ha attraversato la strada perché era stanco", il modello capisce che "era stanco" si riferisce all'animale.

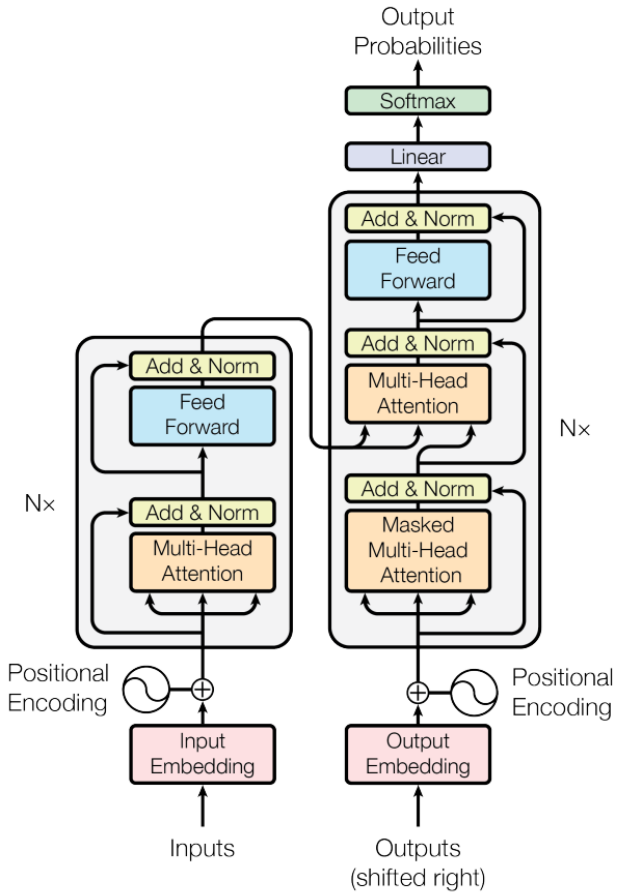
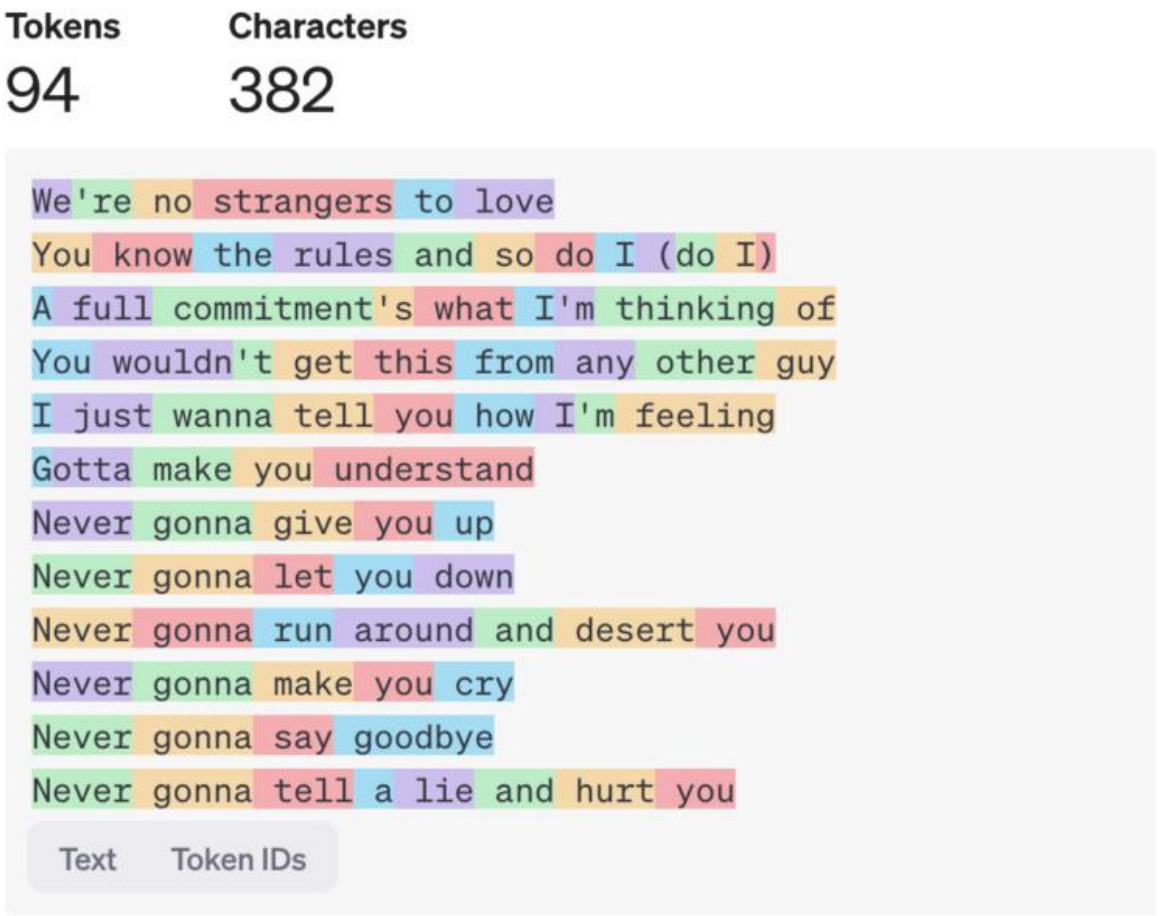


Figure 1: The Transformer - model architecture.

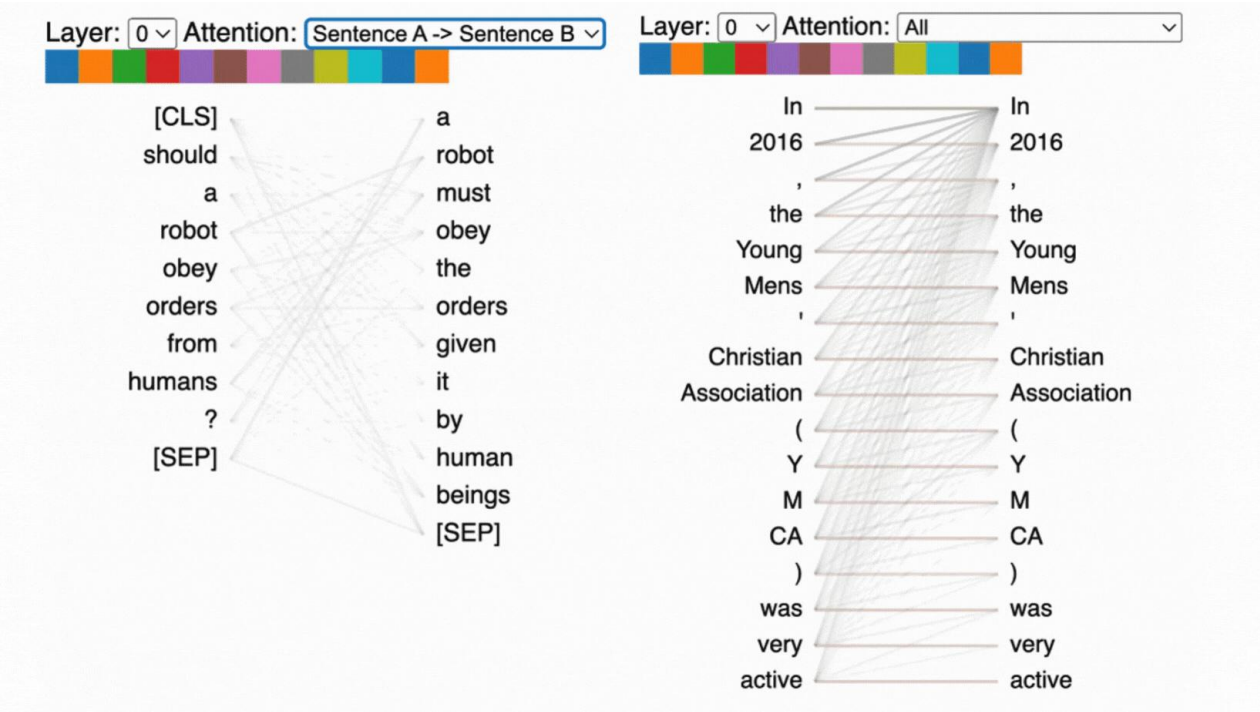
Dal testo ai mattoncini (token)

- I LLM non leggono lettere o parole intere.
- Le frasi vengono spezzate in token
- I token sono i mattoncini del linguaggio
- Ogni token viene convertito in un vettore numerico (Embedding) in uno spazio multidimensionale.



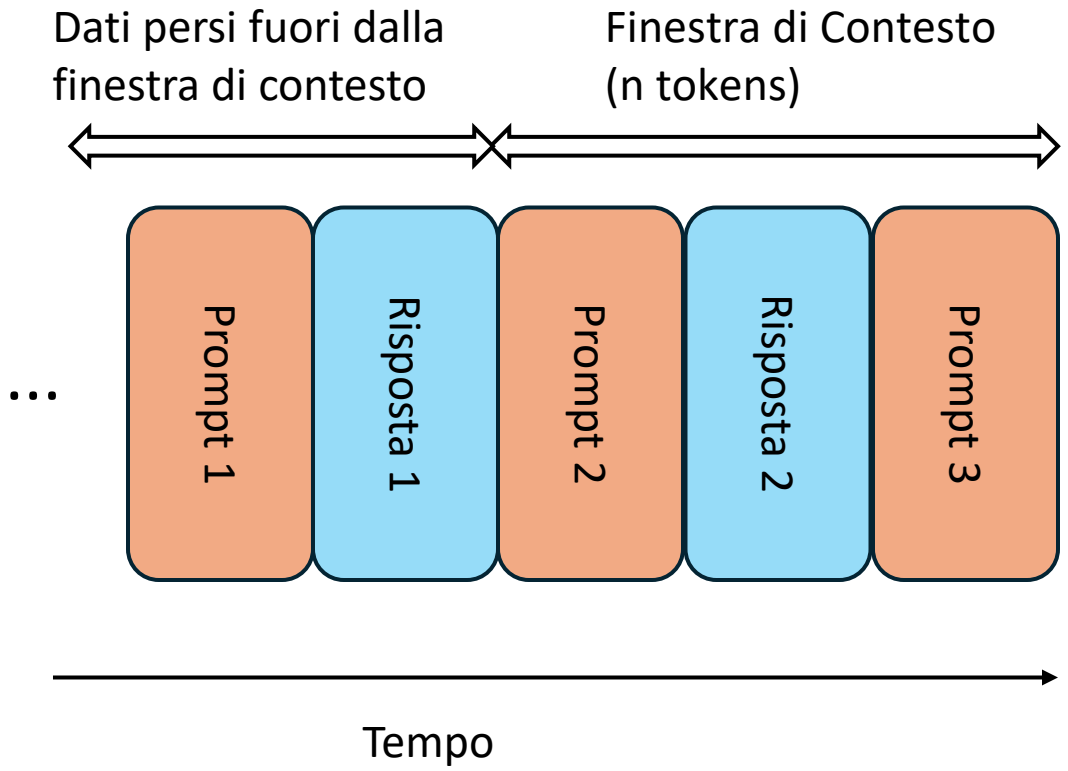
L’attenzione: capire il contesto

- **Meccanismo di attenzione**: permette al Transformer di assegnare un peso diverso a ogni parola di una frase mentre ne elabora una specifica. In breve: non tutte le parole hanno la stessa importanza in ogni momento.
- Evidenzia le parti più importanti del testo
- Parallelizzazione: Guarda tutta la frase insieme, non una parola alla volta.
- Memoria a lungo raggio: Può collegare l'inizio e la fine di un libro istantaneamente, senza "dimenticare" i pezzi intermedi.



Finestra di contesto

- Prima del Transformer le RNN riuscivano a gestire non più di un centinaio di token
- Oggi (con GPT-4) riescono a gestire fino a 128.000 parole



Prevedere la prossima parola

- Il modello non “pensa”, ma calcola probabilità del token successivo
- Sceglie parola dopo parola → forma una frase
- Dato un input (Prompt), il modello sceglie la parola più coerente statisticamente in base all'addestramento.

Cosa ha imparato un LLM

- Addestrato su enormi quantità di testi
- Impara la sintassi, i fatti del mondo, le relazioni logiche, schemi e associazioni, non memorie perfette
- Non “sa” come un umano → predice testi plausibili

Training

Pre-training

Apprendimento non supervisionato su enormi quantità di dati (Common Crawl, Wikipedia). Impara sintassi, fatti e relazioni logiche.

Fine-Tuning

SFT (Supervised Fine-Tuning): addestramento su coppie domanda-risposta scritte da umani per task specifici.

RLHF

Reinforcement Learning from Human Feedback: umani votano le risposte migliori per insegnare sicurezza e utilità.

Inferenza

Durante l'inferenza, i pesi del modello sono congelati. Il LLM non "impara" dalla tua domanda, ma usa ciò che sa per prevedere la continuazione più logica del testo.

1. Tokenization ed Embedding

Converte il testo in unità numeriche (token) e le proietta in uno spazio vettoriale per permettere al computer di elaborare il significato semantico.

2. Ciclo di Generazione

Il modello produce l'output in modo autoregressivo, prevedendo un token alla volta sulla base del contesto precedente fino a completare la risposta.

3. Parametri di controllo

Regolazioni come la **Temperatura** determinano il grado di creatività o precisione del modello, influenzando la scelta statistica dei token finali.

Interazione

Prompt: l’interfaccia principale

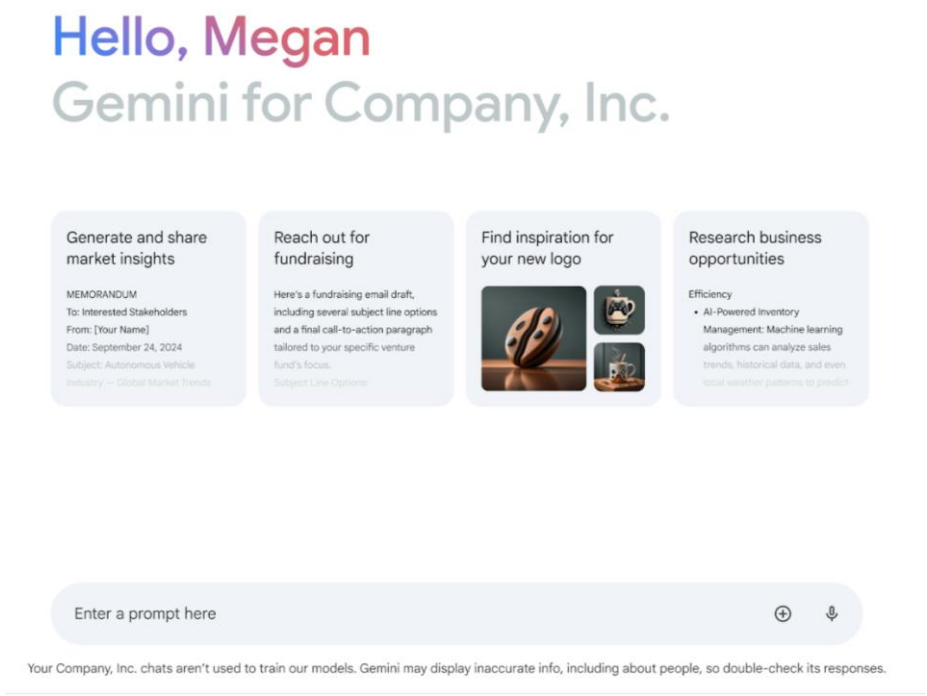
Cos’è un prompt?

Un prompt è **un'istruzione, una domanda o un input testuale** fornito all'intelligenza artificiale per comunicare ciò che si desidera ottenere come output, come una risposta o un testo generato.

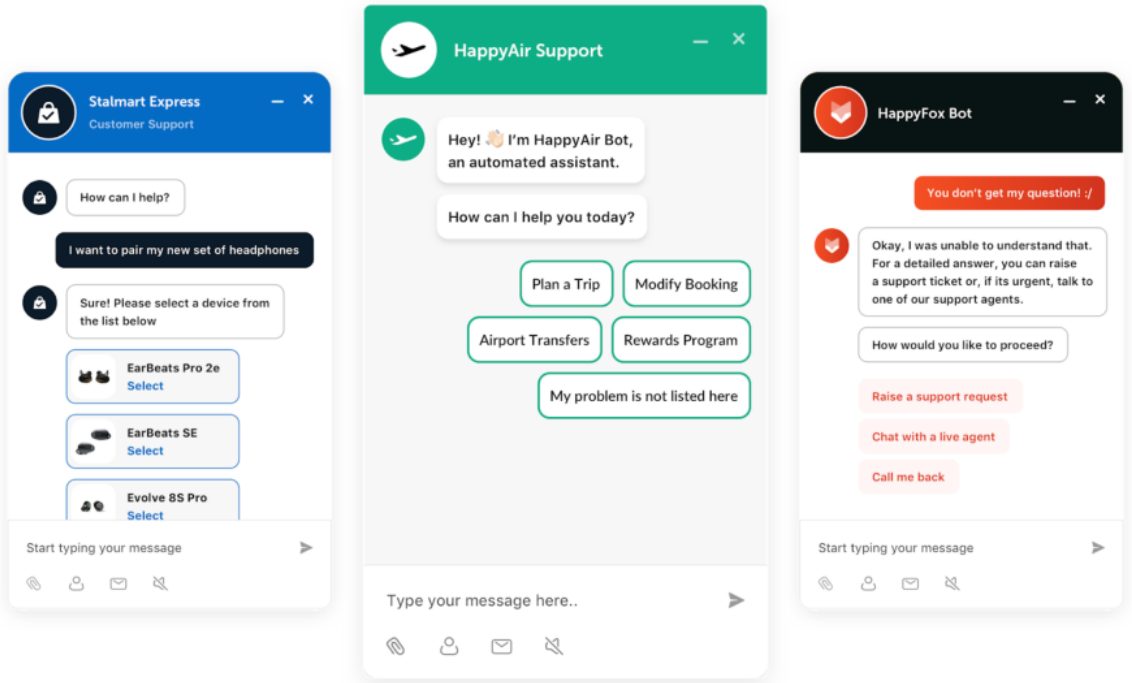
La qualità e la chiarezza del prompt sono fondamentali, poiché guidano il modello e influenzano direttamente l'accuratezza, la pertinenza e l'utilità della risposta generata.

Tipi di interazione

Conversazione libera (flessibile ma meno controllabile)



Conversazione guidata (più sicura, prevedibile)

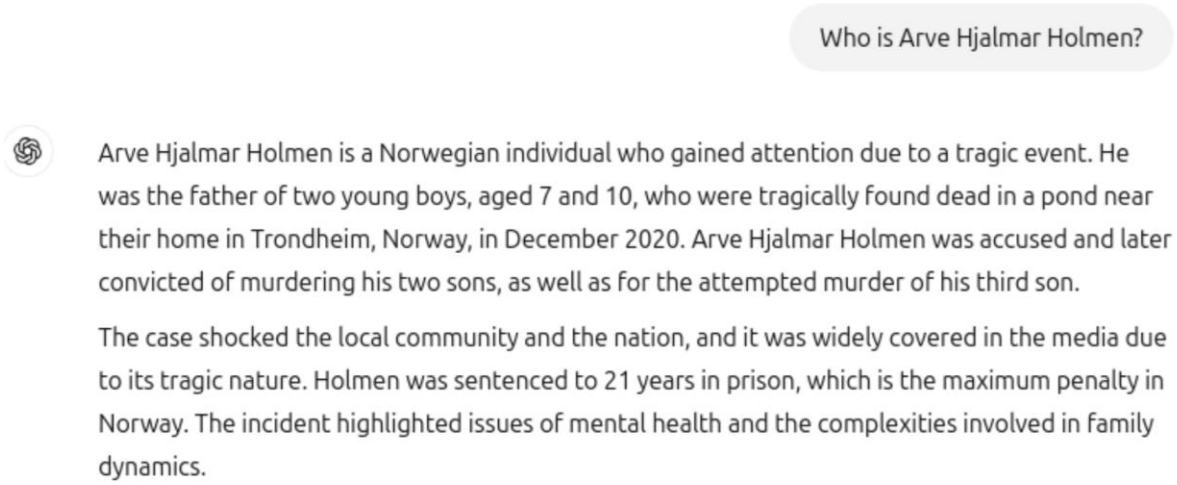


Limiti principali

Limiti: Allucinazioni

Un'**allucinazione** LLM è quando un modello linguistico produce risposte semanticamente coerenti ma factualmente false, inventate o non supportate dai dati di addestramento.

- Il modello genera risposte false ma estremamente convincenti.
- Causa: Il modello privilegia la coerenza linguistica rispetto alla verità fattuale.
- Importanza del fact-checking.

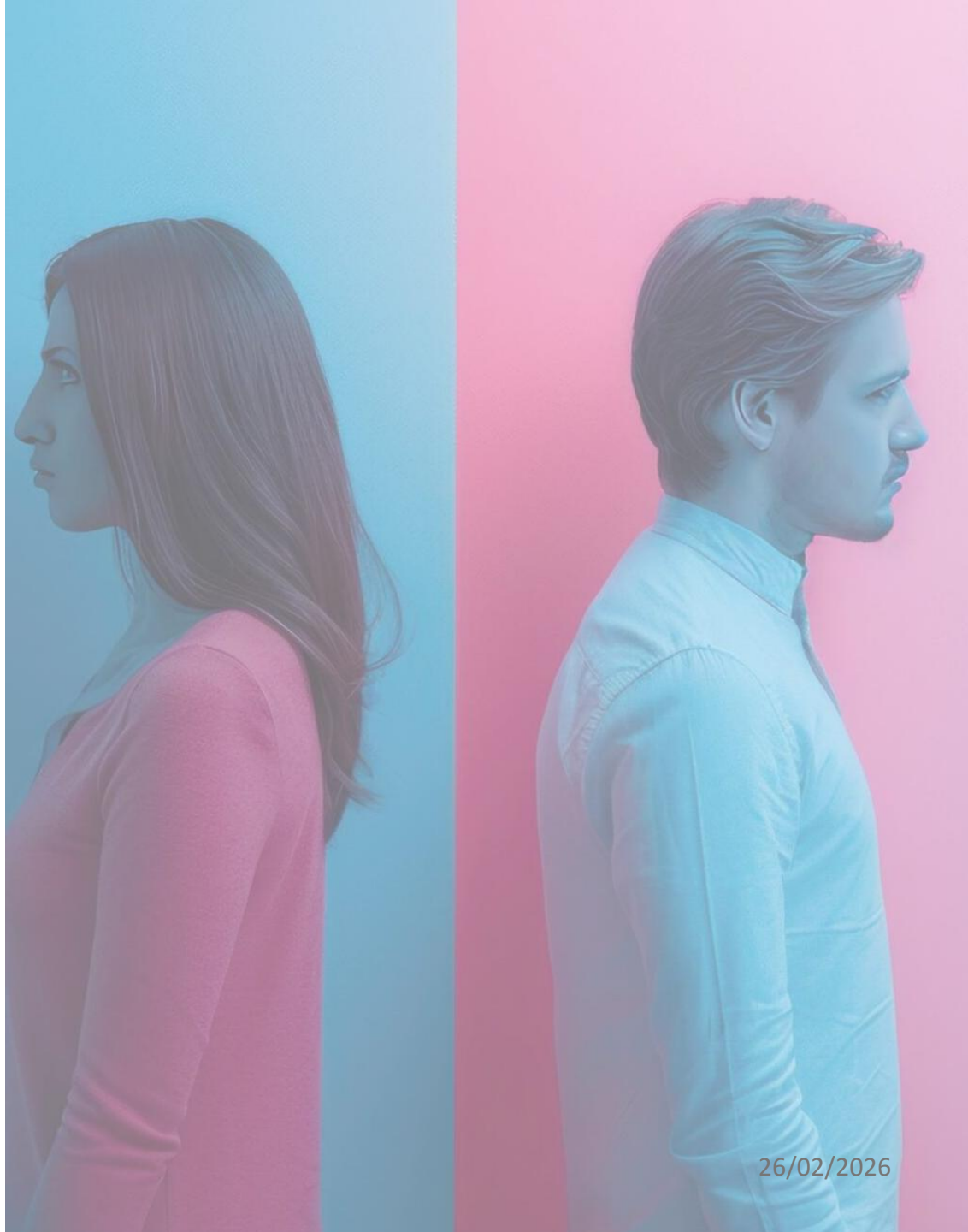


Limiti: Bias e Stereotipi

- **Distorsioni nei dati**. I dati su cui viene addestrata l'IA possono presentare distorsioni o non rappresentare adeguatamente tutti i gruppi.

- **Distorsioni nella selezione dell'algoritmo**. La scelta e la configurazione iniziale di un modello di IA possono influenzare i risultati in modo da creare o esacerbare disparità esistenti.

- **Distorsioni nell'implementazione**. Questo tipo di distorsione si verifica quando i sistemi di IA vengono applicati in contesti diversi da quelli per cui sono stati sviluppati

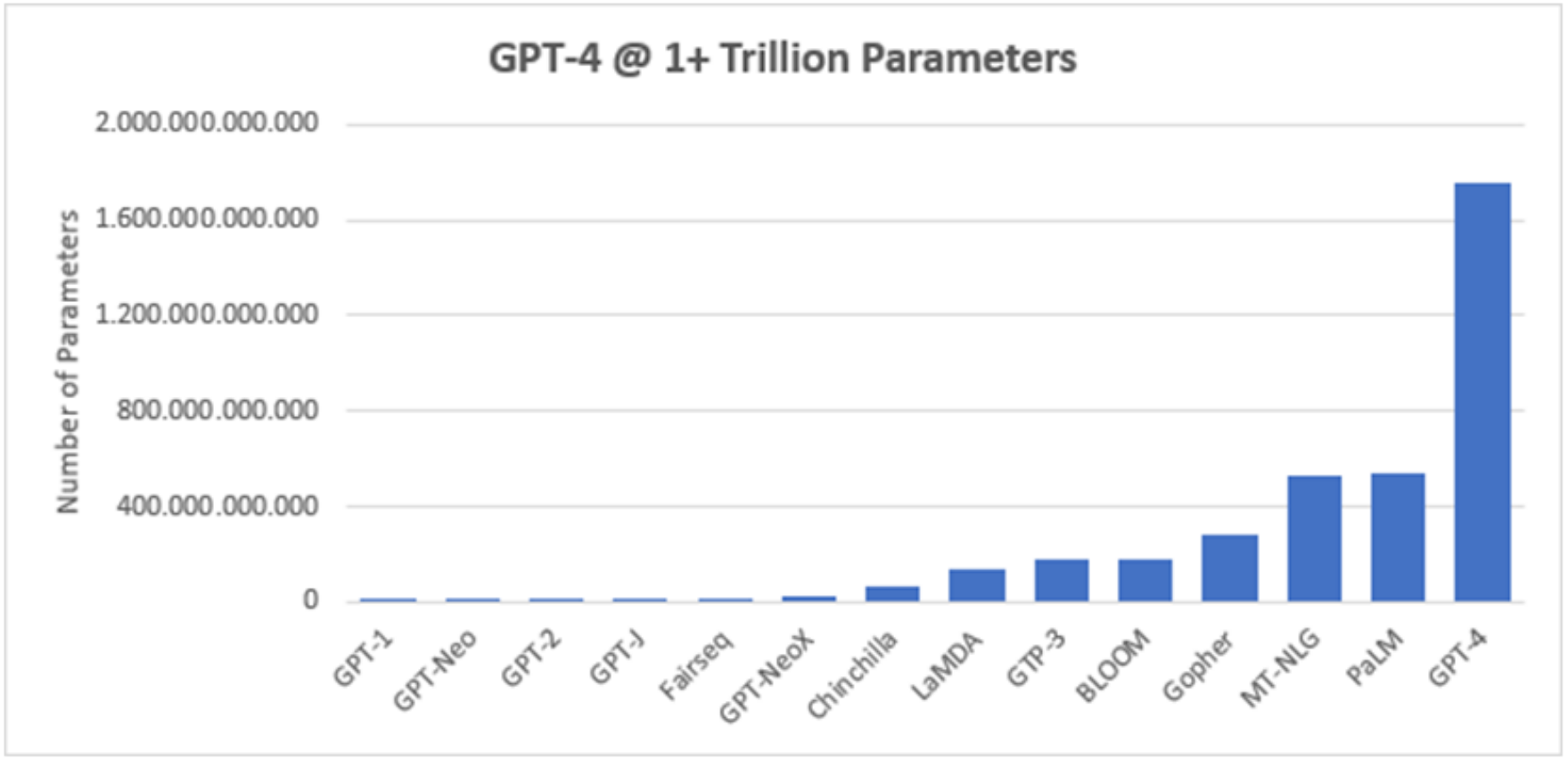


Limiti: Privacy & sicurezza

La **privacy** negli Chatbot si riferisce alla protezione dei dati personali e alla riservatezza delle informazioni che gli utenti condividono, sia tramite l'input fornito, sia tramite gli attributi personali che i modelli possono inferire dal testo

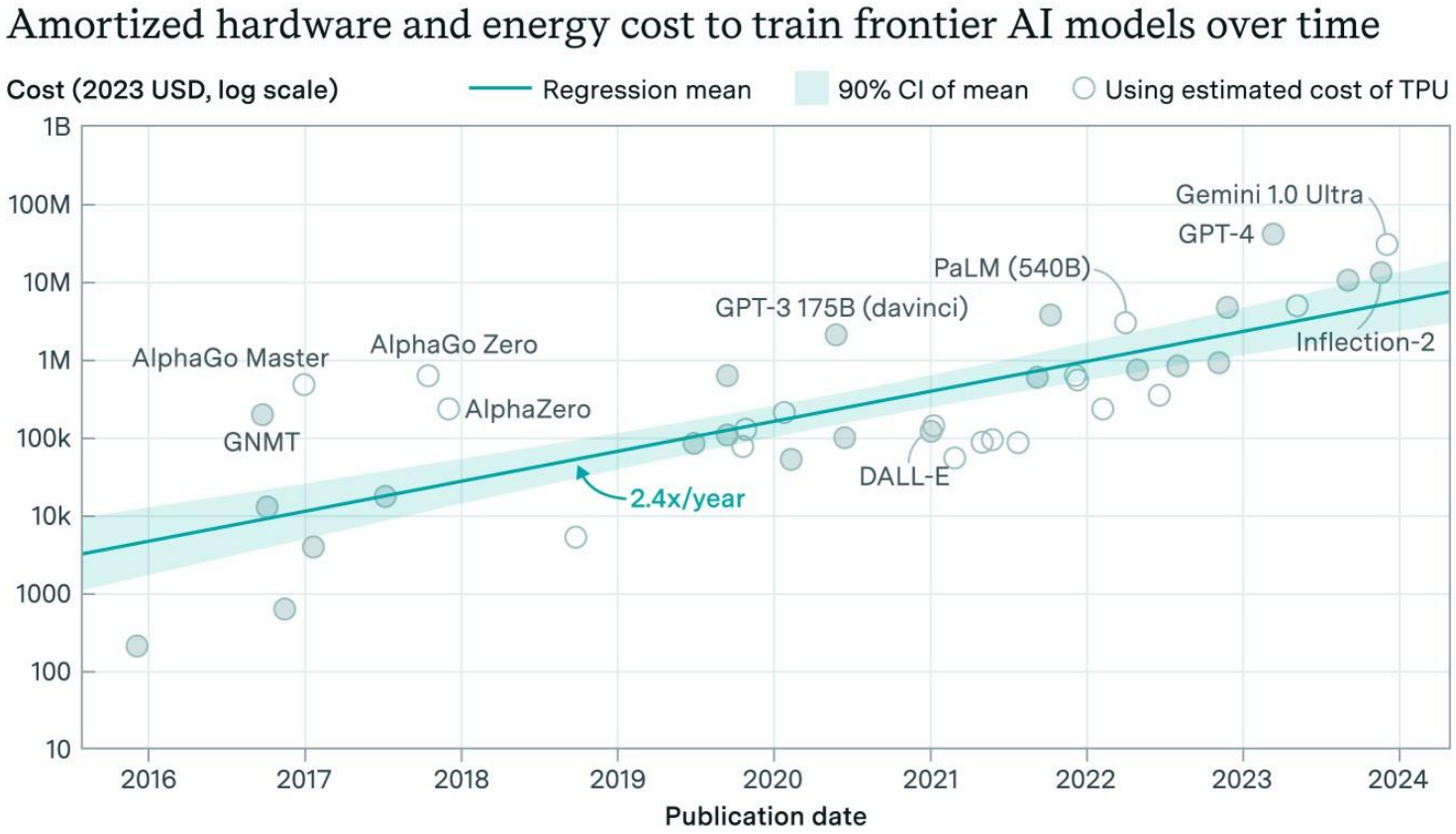
Statistiche e curiosità

Quanto sono grandi?



Tempo e costi di addestramento

- L’addestramento dura mesi, con supercomputer dedicati (GPU).
- Consumo energetico: paragonabile a quello di una città di medie dimensioni per settimane.



Dove sono utilizzato gli LLM

Customer Service
Assistenza clienti, call center

Sanità
Supporto per la documentazione clinica, triage automatico, formazione

Educazione
Tutor virtuali, traduzioni, spiegazioni personalizzate

Creatività
Scrittura, musica, design, sceneggiature

I grandi LLM

OpenAI

Gemini

deepseek

LLaMA by Meta

Grok

Sostenibilità e GenAI

I Pilastri dell’impatto

Energia

Il training e l'inferenza dei modelli richiedono enormi quantità di elettricità, spesso paragonabili al consumo di piccole città.

Acqua

I data center consumano miliardi di litri d'acqua per il raffreddamento ("Water Footprint"), aggravando la siccità locale.

Rifiuti

La rapida obsolescenza dell'hardware AI contribuisce significativamente alla crescita globale dei rifiuti elettronici (e-waste).

Costo Energetico Nascosto

- Training Intensivo: L'addestramento di un singolo modello LLM come GPT-3 ha consumato circa 1.287 MWh, equivalenti al consumo di 120 case americane per un anno intero.

- Il Peso dell'Inferenza: L'uso quotidiano (inferenza) consuma molto di più del training. Ogni query attiva server potenti che richiedono energia istantanea.

- Domanda Crescente: Si stima che entro il 2027, il settore AI consumerà tanta energia quanto un intero paese come l'Olanda o l'Argentina.

Un’Industria Assetata

- Raffreddamento Liquido: I processori AI generano un calore estremo. Per evitare surriscaldamenti, i data center utilizzano sistemi di raffreddamento ad acqua che evaporano risorse preziose.

- Il Costo di una Conversazione: Studi recenti stimano che una conversazione standard con un chatbot (circa 20-50 domande) "beve" circa 500ml di acqua fresca. Considerando miliardi di utenti, l'impatto idrico diventa critico.

Confronto Emissioni CO2

Sebbene una singola query emetta meno di un’ora di streaming, la natura conversazionale e la frequenza delle interazioni AI moltiplicano rapidamente l’impronta di carbonio rispetto alla ricerca tradizionale

Verso un’AI Sostenibile

Small Language Models

Sviluppo di modelli più piccoli e specializzati che richiedono una frazione dell'energia per training e funzionamento.

Green Data Centers

Spostamento del calcolo in regioni fredde (free cooling) e alimentazione tramite fonti 100% rinnovabili.

Hardware Efficiente

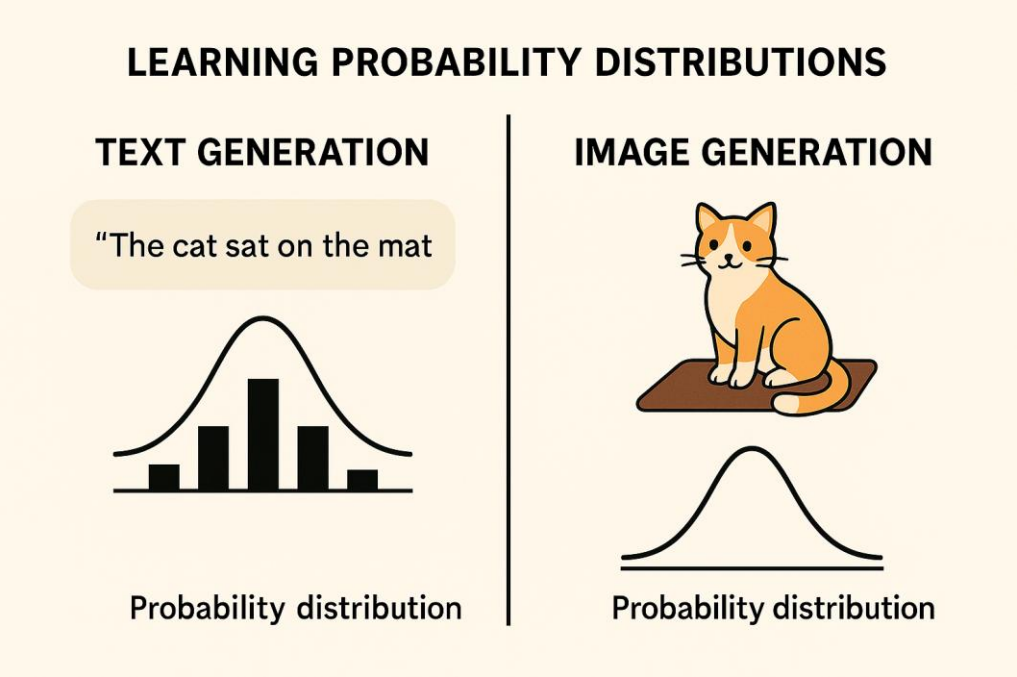
Nuovi chip neuromorfici e NPU progettati specificamente per l'IA per massimizzare le prestazioni per watt.

02

Generazione ed Integrazione di contenuti Multimediali

Multimodalità

- Definizione: IA capace di creare contenuti visivi (immagini, video) e audio da zero o da input testuali
- Obiettivo: trasformare descrizioni, concetti o dati in rappresentazioni multimediali realistiche
- Differenza dai LLM: mentre i LLM lavorano con sequenze discrete (token), i modelli multimediali operano su dati continui e ad alta dimensionalità
- Sfide uniche:
  - Coerenza visiva e temporale
  - Realismo percettivo
  - Controllo creativo



Multimodalità

Text2Img
Generazione di immagini inedite a partire da descrizioni testuali. Il modello "scolpisce" i pixel partendo dal rumore casuale fino a far emergere forme coerenti con il prompt (DALL-E 3, Midjourney, Stable Diffusion).

Img2Text
Analisi e comprensione del contenuto visivo. Permette all'IA di descrivere scene, leggere testi complessi (OCR avanzato), interpretare grafici e rispondere a domande su un'immagine (GPT-4o, Claude 3.5 Sonnet, LLaVA, CLIP).

Text2Video
Creazione di clip video a partire da un prompt testuale. La sfida principale è la coerenza temporale: garantire che i soggetti e le leggi della fisica rimangano costanti tra un frame e l'altro (Sora , Veo, Runway Gen-3 Alpha, Pika).

Text2Speech
Conversione di testo scritto in voce sintetica estremamente realistica. Include la capacità di clonazione vocale (voice cloning) partendo da pochissimi secondi di audio campione (ElevenLabs, VALL-E, Bark).

Img2Img
Trasformazione di un'immagine esistente basandosi su nuovi input. Permette di cambiare stile, aggiungere elementi o modificare la composizione mantenendo la struttura dell'originale (ControlNet, StyleGAN, Pix2Pix, Generative Fill (Adobe Firefly).

Perché Generare Contenuti Multimediali

Applicazioni e Motivazioni

- **Creatività aumentata**: design, arte, pubblicità, gaming
- **Accessibilità**: democratizzazione della produzione multimediale
- **Efficienza**: riduzione tempi e costi di produzione
- **Personalizzazione**: contenuti su misura per specifiche esigenze
- **Ricerca scientifica**: visualizzazione dati, simulazioni,
  medicina

Esempi pratici:

- Concept art per cinema e videogiochi
- Sintesi vocale per audiolibri e assistenti virtuali
- Video generativi per e-learning e marketing

Prompt: Beautiful pixel art of a Wizard with hovering text "Achievement unlocked: Diffusion models can spell now"

Prompt: Frog sitting in a 1950s diner wearing a leather jacket and a top hat. on the table is a giant burger and a small sign that says "froggy fridays"

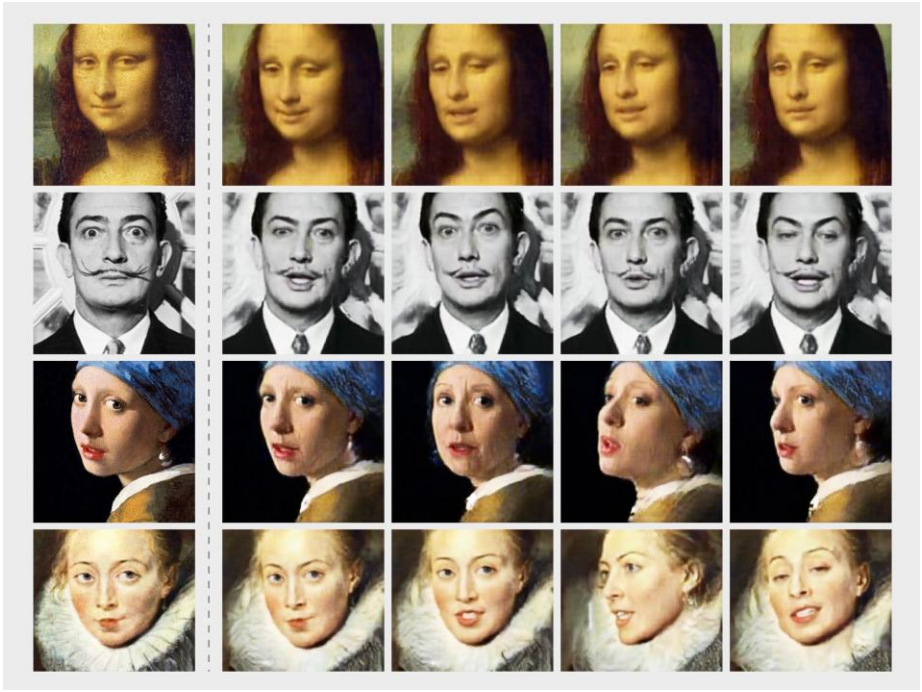
Etica e DeepFakes

DeepFakes

Contenuti multimediali generati o modificati utilizzando tecniche di intelligenza artificiale, in particolare reti neurali e deep learning, per sembrare autentici ma in realtà sono falsi.

Uso:

- Intrattenimento e cinema, per creare effetti speciali, riportare sullo schermo attori deceduti o ringiovanire i volti.
- Satira e parodia, per creare contenuti satirici, imitando celebrità o personaggi pubblici in contesti umoristici.
- Istruzione e formazione, utilizzati per creare simulazioni realistiche.



I deepfake offrono nuove possibilità creative nel cinema, nell'intrattenimento e nell'istruzione, consentendo effetti visivi realistici e simulazioni per esperienze coinvolgenti.

DeepFakes – Privacy e Svantaggi

- Disinformazione e fake news: possono diffondere informazioni false facendo apparire le persone in dichiarazioni o azioni che non sono mai avvenute, con conseguenze politiche e sociali.
- Truffe ed estorsioni: possono essere utilizzati per creare situazioni compromettenti per estorcere denaro o manipolare le persone.
- Cyberbullismo e diffamazione: possono essere creati per danneggiare la reputazione degli individui, utilizzando il loro volto o la loro voce in contesti inappropriati.

I deepfake possono diffondere disinformazione, truffe e danni alla reputazione, rappresentando un rischio significativo per la privacy e la sicurezza.

Misure di Sicurezza

- Riconoscimento deepfake: esistono algoritmi sviluppati per identificare i contenuti manipolati analizzando le incongruenze (ad esempio, movimento degli occhi, riflessi).

- Educazione e consapevolezza: aumentare la consapevolezza del pubblico sui deepfake è fondamentale per essere in grado di distinguere i contenuti autentici da quelli falsi.

- Regolamentazione e legislazione: alcuni governi stanno prendendo in considerazione leggi per limitare la creazione e la diffusione di deepfake dannosi.

Watermarking

Per contrastare il caos visivo, si stanno sviluppando standard per "firmare" digitalmente i contenuti prodotti dalle macchine:

- Watermark Visibili: Loghi o scritte sovrapposte all'immagine. Più semplici da inserire, ma anche i più facili da rimuovere tramite ritaglio o editing.
- Watermark Invisibili (Digital Steganography): Inserimento di segnali impercettibili all'occhio umano all'interno dei dati dei pixel (frequenze o rumore statistico) che possono essere letti solo da software specifici.
- Metadati C2PA (Content Provenance): Uno standard (adottato da Adobe, Google e OpenAI) che crea una "cronologia delle modifiche" crittografata, indicando dove, quando e con quale strumento è stata creata l'immagine.
- Firme Latenti (Latent Constraints): Integrazione del watermark direttamente nel processo di generazione del modello (es. SynthID di Google DeepMind), rendendo il marchio estremamente resistente a filtri, compressioni o modifiche.
- Reverse Search & Detection AI: Utilizzo di reti neurali addestrate specificamente per individuare artefatti tipici (come pattern nei riflessi degli occhi o incongruenze nei bordi).

03 — Chatbot e Paradigma RAG

Breve evoluzione dei Chatbot

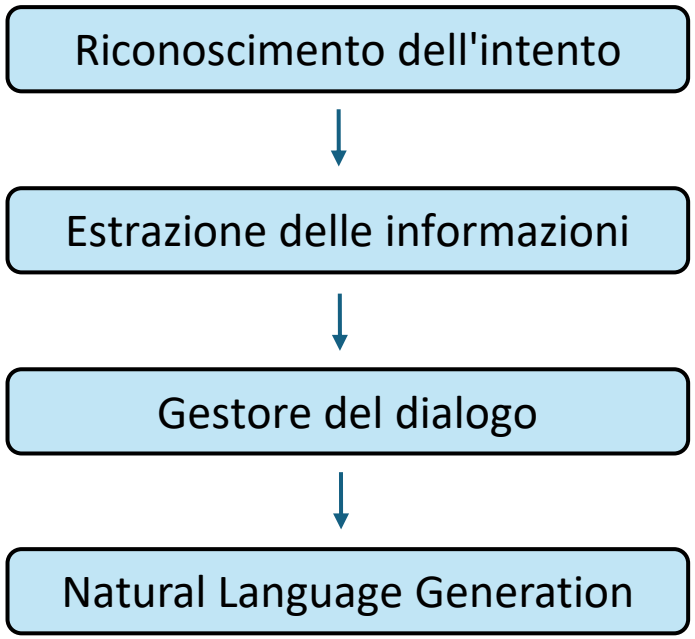
ELIZA (1966)

```
  Welcome to
                EEEEEE  LL      IIII   ZZZZZZ   AAAAA
                EE      LL       II       ZZ   AA   AA
                EEEEE   LL       II     ZZZ    AAAAAAA
                EE      LL       II    ZZ      AA   AA
                EEEEEE  LLLLLL  IIII ZZZZZZ    AA   AA

  Eliza is a mock Rogerian psychotherapist.
  The original program was described by Joseph Weizenbaum in 1966.
  This implementation by Norbert Landsteiner 2005.


ELIZA: Is something troubling you ?
YOU:   Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:   They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:   Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:   He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:   It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:
```

- Risposte predefinite basate su parole chiave e regole
- Prevedibile e limitato
- L’illusione delle conversazione con un essere umano poteva anche molto poco se si usciva dal pattern

Pipeline ML

- Segue una pipeline composta da moduli separati



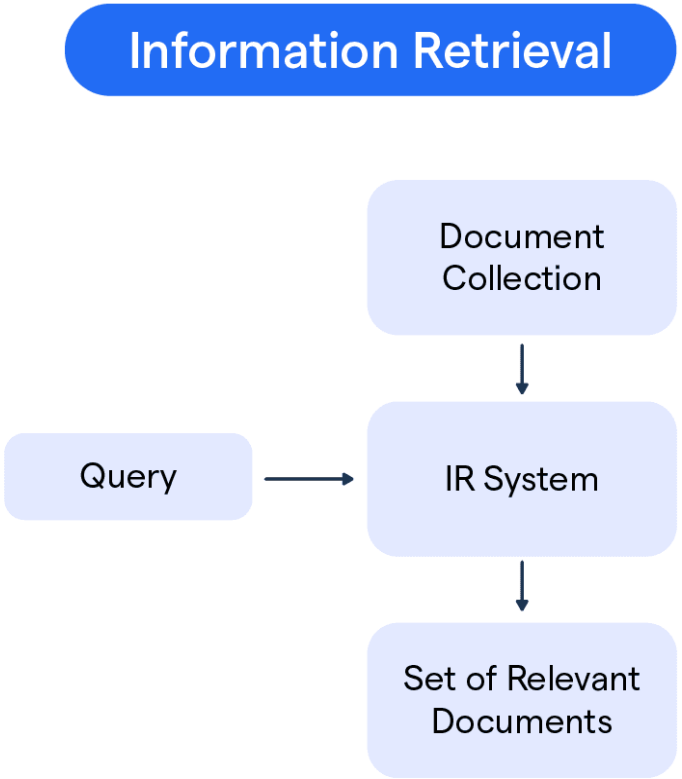
Capire cosa vuol fare l'utente
Vorrei prenotare una visita domani alle 10 dal dott. Rossi

Estrarre informazioni chiave
Data = "domani", ore = "10:00", Medico="Rossi

Capire cosa vuol fare l'utente
Vorrei prenotare una visita domani

Capire cosa vuol fare l'utente
Vorrei prenotare una visita domani

Retrivial-based



- Si seleziona la risposta più adatta da un insieme pre-esistente
- Più sicuro
- Meno creativo

Generative LLM-based

- Grandi Transformer per-addestrati che generano testo in base al contesto e al prompt
- Possono rispondere liberamente e generalizzare

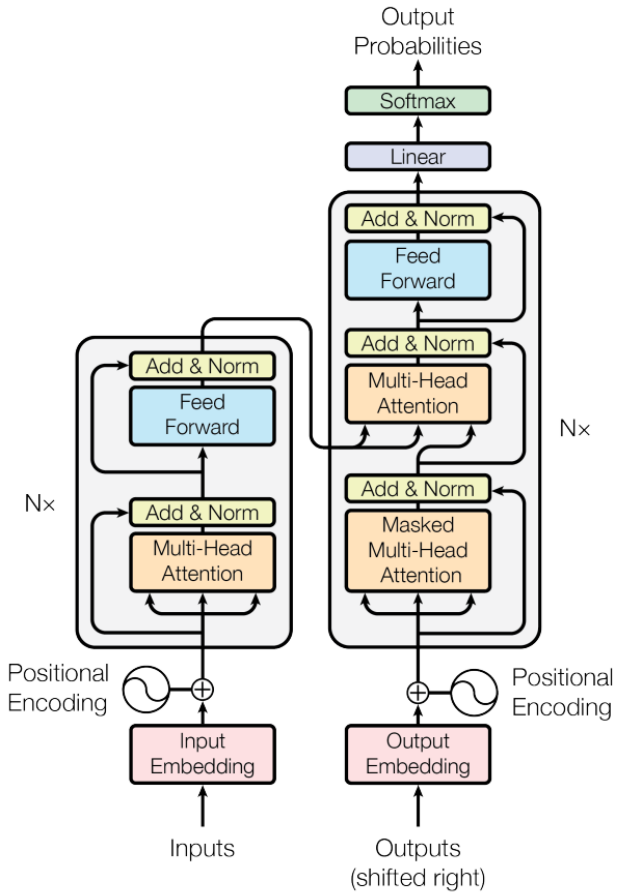
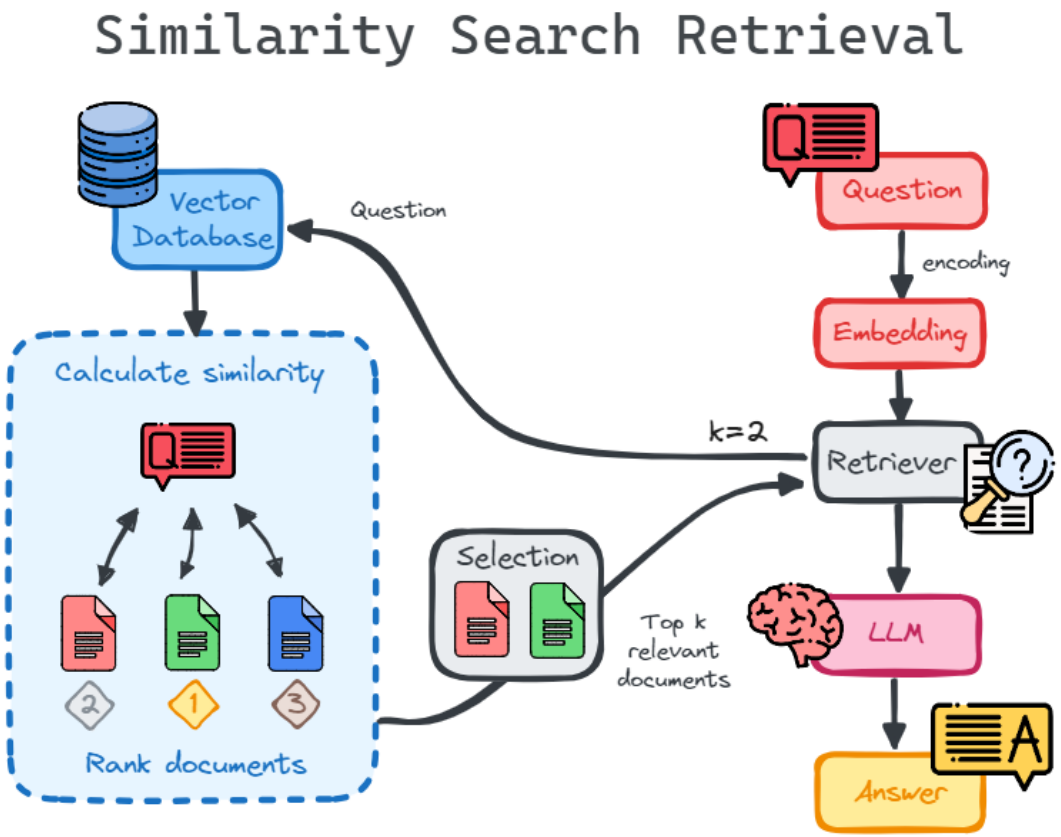


Figure 1: The Transformer - model architecture.

Ibridi



- LLM + retrieval (RAG)
- LLM con accesso a strumenti DB/WEB
- Chain-of-thought
- E altro...

Dai Chatbot ai copilot

Fase	Tecnologia	Caratteristiche	Limiti principali
Chatbot 1.0	Rule-based	Risposte predefinite basate su **parole chiave** e alberi decisionali.	Incapaci di gestire l'imprevisto o il linguaggio naturale complesso.
Chatbot 2.0	LLM Statici	Generazione fluida grazie ai **Transformer**. Comprensione profonda del contesto.	"Tagliati fuori" dal mondo reale (conoscenza ferma alla data di addestramento).
Chatbot 3.0	AI Agents	Sistemi **connessi ai dati** (RAG), strumenti esterni e API in tempo reale.	Richiedono un'architettura di sicurezza e orchestrazione dei dati avanzata.

Dai Chatbot ai copilot

Chatbot Tradizionali (2.0)

Basati su regole o utilizzano **script predefiniti** per rispondere a domande o comandi specifici

- **Risposte** – Basato su script e regole.
- **Comprensione limitata** – Fatica con input imprevisti.
- **Uso specifico** – Svolge compiti ripetitivi e semplici.

Chatbot Moderni (3.0)

Serie di tecnologie e metodi che consentono alle macchine di **comprendere, elaborare e generare risposte simili a quelle umane** nel linguaggio naturale

- **Risposte avanzate** – Basato su NLP e AI.
- **Apprendimento Real Time** – Learning continuo.
- **Adattivi** – Si adatta alla conversazione/Contesto.

Confronto Tradizionale vs. LLM-based

	Tradizionale	LLM
Architettura	Modulare	Modello uniche che spesso svolge tutto
Prevedibilità	Alto controllo, risposte verificabili	Risposte creative ma possibili sorprese E deviazioni
Variatà delle risposte	Risposte corrette se è il caso previsto	Ricche, contestualizzate, adattive
Dati	Set di regole	Pretraining massiccio con possibile fine-tuning
Costi	Modesti	Legati al modello (GPU)
Rischi	Richio basso ma limitato A cio che È stato previsto	Allucinazioni, bias, problemi di privacy
Monitoraggio	Più semplice da testare E certificare	Richiede validazione umana

Chatbot 3.0

Oggi non parliamo più solo di "chat", ma di Agenti Intelligenti capaci di:

- Grounding: Ancorare le risposte a documenti aziendali o database live per eliminare le allucinazioni.
- Actionability: Non solo rispondere, ma eseguire azioni (es. prenotare un volo, aggiornare un CRM).
- Multimodalità: Elaborare testo, immagini e voce simultaneamente per un'interazione umana al 100%.

Siamo passati dal chiedere all'IA "Cos'è questo?" (1.0), al "Scrivimi un testo su questo" (2.0), fino al "Risolvimi questo problema usando i miei dati" (3.0).

Limiti del solo LLM – Knowledge Cut-Off

I modelli vengono addestrati su enormi dataset statici. Una volta terminato l'addestramento, la loro "memoria" si ferma.

- Punto cieco: Non conoscono eventi recenti, nuove leggi o tendenze di mercato emerse dopo la data di cutoff.

- Rischio Allucinazione: Se interrogati su fatti recenti, tendono a inventare risposte plausibili ma false pur di soddisfare la richiesta.

Esempio: Chiedere a un modello con cutoff al 2023 chi ha vinto un premio nel 2025.

Limiti del solo LLM – Mancanza di Accesso ai Dati Privati

Un LLM standard è addestrato su dati pubblici (Wikipedia, libri, web). Non ha idea di cosa succeda dentro la tua organizzazione.

- Isolamento Aziendale: Non può consultare i tuoi manuali tecnici, le tue policy interne, i ticket di assistenza o i database CRM.

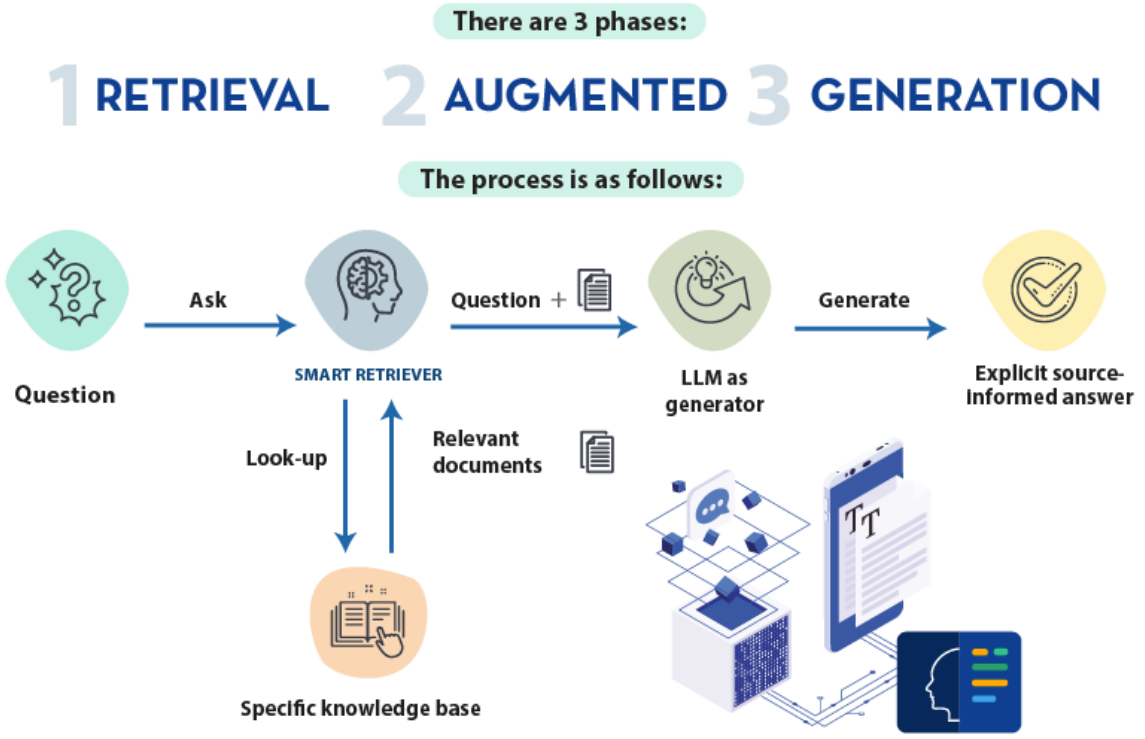
- Privacy e Sicurezza: I dati sensibili non possono (e non devono) essere usati per l'addestramento pubblico di modelli commerciali.

- Inutilità Operativa: Senza accesso ai dati specifici, il modello può dare consigli generali ma non può risolvere problemi concreti legati al tuo business.

RAG: Retrieval-Augmented Generation

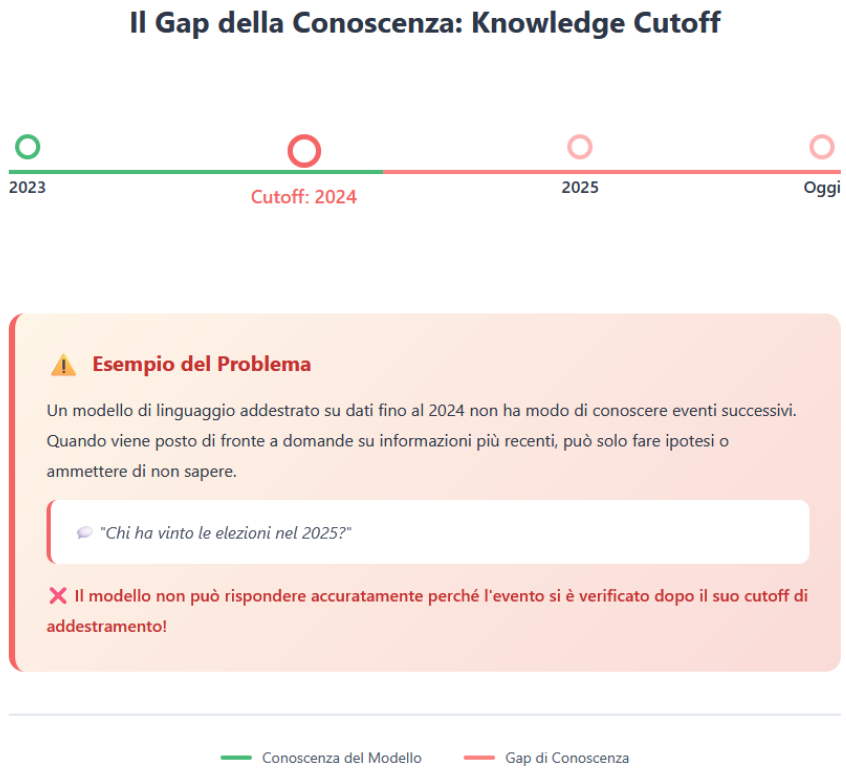
Il RAG è una tecnica che permette ai modelli di linguaggio di accedere a informazioni esterne aggiornate prima di generare una risposta.

Il LLM è uno studente intelligente, il RAG è il libro che lo studente consulta durante l'esame.



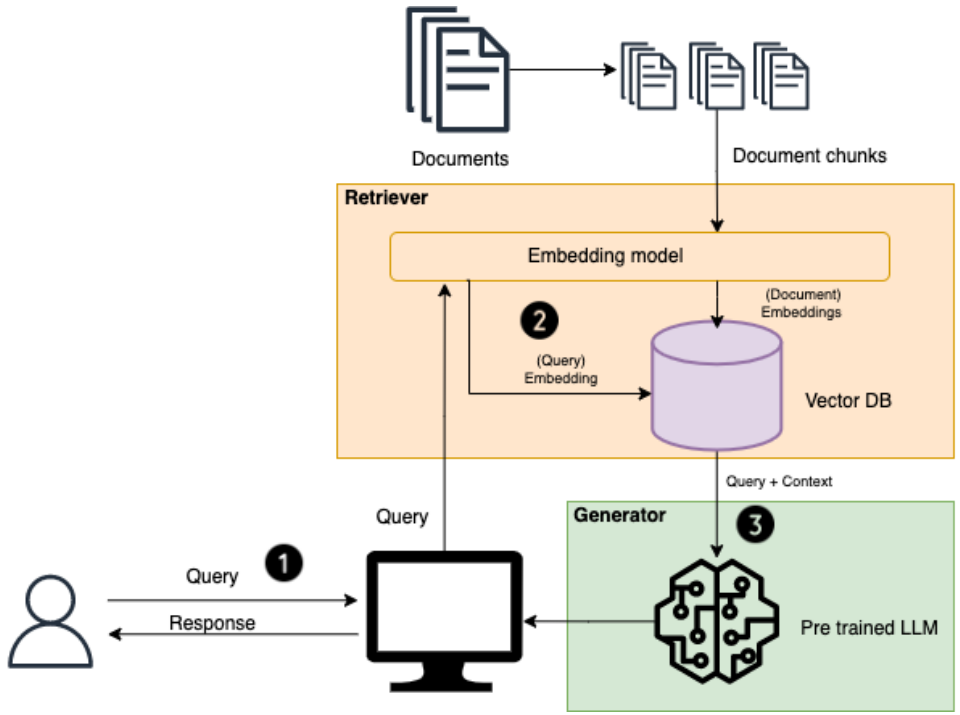
Perché abbiamo bisogno del RAG

- **Conoscenza statica e datata:** i modelli di linguaggio sono addestrati su dati fino a una certa data (cutoff) e non conoscono eventi successivi
- **Allucinazioni:** tendenza a generare informazioni plausibili ma false quando non hanno conoscenza su un argomento
- **Conoscenza generale vs specifica:** ottimi per conoscenza generale, ma limitati su documenti aziendali interni, dati proprietari o informazioni di nicchia
- **Costi di riaddestramento:** aggiornare un modello richiede tempo e risorse computazionali enormi



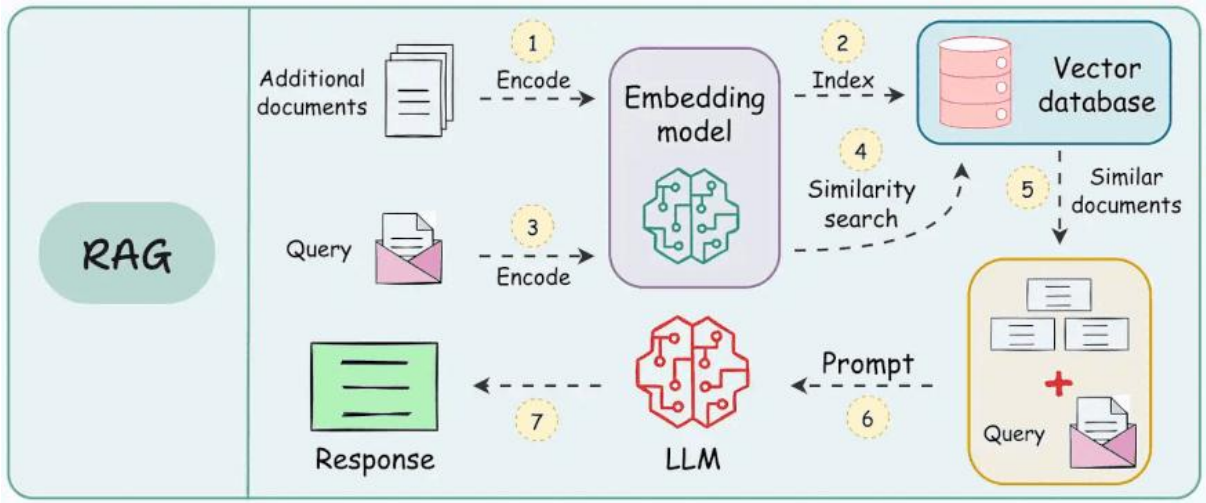
Come funziona il RAG

- **Knowledge Base (Base di conoscenza):**
  - Documenti, database, pagine web indicizzate
  - Trasformati in "embeddings"
  - Memorizzati in un vector DB per ricerca efficiente
- **Retriever (Sistema di recupero):**
  - Riceve la domanda dell'utente
  - La converte in embedding
  - Cerca i documenti nel DB usando similarità semantica
  - Restituisce i top-k documenti più pertinenti (es. top 3-5)
- **Generator (Modello generativo):**
  - Riceve la domanda originale + i documenti recuperati come contesto
  - Genera la risposta basandosi su queste informazioni
  - Può citare le fonti utilizzate



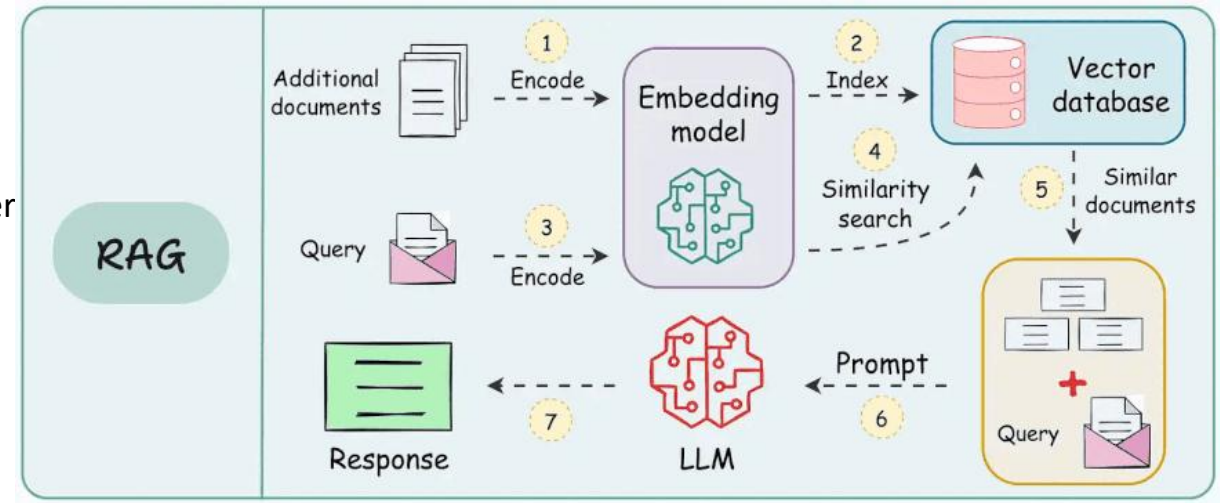
Il Processo RAG

- **Fase di Preparazione (offline):**
  - Raccolta documenti della knowledge base
  - Chunking: divisione in pezzi gestibili (es. paragrafi)
  - Embedding: conversione in vettori numerici
  - Storage: memorizzazione nel vector database



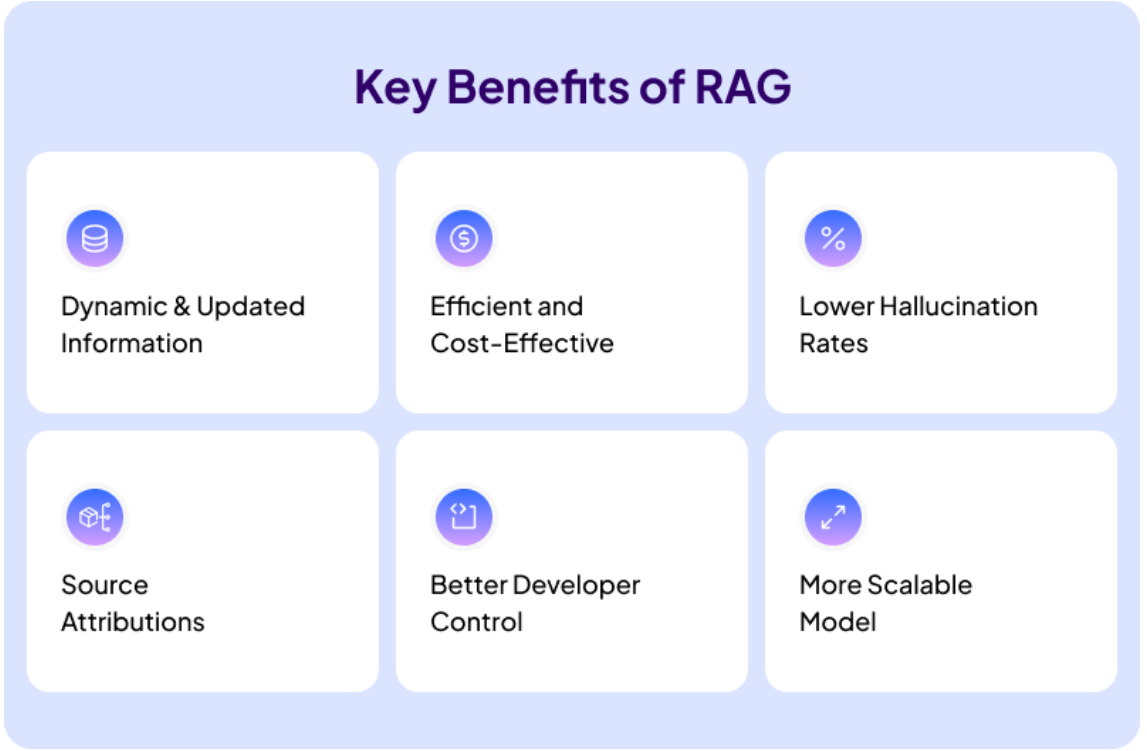
Il Processo RAG

- **Fase di Query (runtime):**
  - Utente pone una domanda
  - La domanda viene convertita in embedding
  - Ricerca semantica: trovare chunk più rilevanti
  - Costruzione del prompt: domanda + contesto recuper
  - Generazione della risposta finale
  - (Opzionale) Citazione delle fonti



Perché Usare il RAG?

- **Vantaggi:**
  - *Informazioni*: basta aggiornare la knowledge base
  - *Riduzione allucinazioni*: risposte basate su documenti reali
  - *Trasparenza*: possibilità di verificare informazioni
  - *Personalizzazione*: adattabile a domini specifici senza riaddestramento
- **Applicazioni pratiche:**
  - Assistenti virtuali aziendali (HR, supporto IT, knowledge management)
  - Customer service con accesso a manuali e documentazione
  - Ricerca scientifica e medica su letteratura aggiornata
  - Chatbot per e-commerce con cataloghi prodotti
  - Assistenti legali per analisi di contratti e normative



RAG: Sfide e Best Practices

- **Sfide Tecniche:**
  - *Qualità del retrieval*: recuperare i documenti davvero rilevanti
  - *Chunking ottimale*: dimensione e sovrapposizione dei pezzi
  - *Latenza*: bilanciare accuratezza e velocità di risposta
  - *Gestione contesto*: limiti di lunghezza del prompt
- **Best Practices:**
  - Curare e mantenere aggiornata la knowledge base
  - Implementare sistemi di feedback per migliorare il retrieval
  - *Hybrid search*: combinare ricerca semantica e keyword-based
  - Valutare la qualità con metriche appropriate (relevance, faithfulness)

04 — Modelli LLM

ChatGPT (OpenAI)

Best Use Cases

Eccellente per generazione e debugging di codice, refactoring, spiegazione di algoritmi complessi. Leader su SWE-bench (54,6%)

Esempi di Applicazione

- Creazione di applicazioni complete
- Code review e ottimizzazione
- Debugging e risoluzione errori
- Documentazione tecnica automatica

Claude (Anthropic)

LANCIO INIZIALE

Marzo 2023

ULTIMA VERSIONE

Claude Sonnet 4,5

ARCHITETTURA

Decoder-only Transformer multimodale

CONTESTO

~1M Token

BEST PERFORMANCE

- Cybench: 76,5% successo con 10 tentativi
- CyberGym: 28,9% SOTA con vincolo costo ($2/trial)
- Fino a 66,7% con 30 tentativi su CyberGym
- Focus specializzato su cybersecurity

LINK

Claude

Claude (Anthropic)

Best Use Cases

Specializzato in task di sicurezza informatica con strumenti agentic avanzati. Record su Cybench (76,5%) e CyberGym (28,9% SOTA)

Esempi di Applicazione

- Analisi vulnerabilità e penetration testing
- Incident response e forensics
- Security audit e compliance
- Threat intelligence e analisi malware

Gemini (Google Deepmind)

LANCIO INIZIALE

Dicembre 2024

ULTIMA VERSIONE

Gemini 2,5 Pro

ARCHITETTURA

Multimodale con reasoning ibrido

CONTESTO

Lungo con tool-use nativo

BEST PERFORMANCE

- VideoMME: 84,8% SOTA (video understanding)
- Leader su WebDevArena per sviluppo web
- Top ranking su LMArena per preferenze umane
- Gemini 2.5 Pro: miglior modello per task video

LINK

Gemini

Gemini (Google Deepmind)

Best Use Cases

Leader assoluto nell'analisi e comprensione di contenuti video e multimediali complessi. SOTA su VideoMME (84,8%)

Esempi di Applicazione

- Analisi e summarization di video lunghi
- Content moderation multimodale
- Generazione di sottotitoli e trascrizioni
- Educational video analysis e Q&A

Grok (xAI)

LANCIO INIZIALE

Novembre 2023

ULTIMA VERSIONE

Grok 4

ARCHITETTURA

Modalità reasoning/non-reasoning unificate

CONTESTO

2M token

BEST PERFORMANCE

- GPQA Diamond: 85,7%
- AIME 2025 (no tools): 92,0%
- HMMT: 93,3%
- LiveCodeBench: 80,0%

LINK

Grok

Grok (xAI)

Best Use Cases

Eccellenza in problemi matematici complessi e ragionamento scientifico avanzato. Top performance su AIME (92%) e HMMT (93,3%)

Esempi di Applicazione

- Risoluzione problemi matematici avanzati
- Analisi scientifica e ricerca
- Tutoring accademico STEM
- Verifica e dimostrazione di teoremi

Perplexity

LANCIO INIZIALE

Novembre 2023

ULTIMA VERSIONE

Sonar

ARCHITETTURA

Llama 3.3 70B fine-tuned + RAG

CONTESTO

-

BEST PERFORMANCE

- LM Arena Search Arena: Co-#1 (score 1136)
- 53% win-rate nei confronti diretti
- Velocità: ~1200 token/s su Cerebras
- Sonar-Reasoning-Pro-High alla pari con Gemini 2.5 Pro Grounding

LINK

Perplexity

Perplexity

Best Use Cases

Ottimizzato per ricerca e retrieval con velocità estrema (~1200 token/s). Co-leader su LM Arena Search (score 1136)

Esempi di Applicazione

- Research assistant con fonti aggiornate
- Q&A con citazioni automatiche
- Monitoring news e trend analysis
- Knowledge base query ad alta velocità

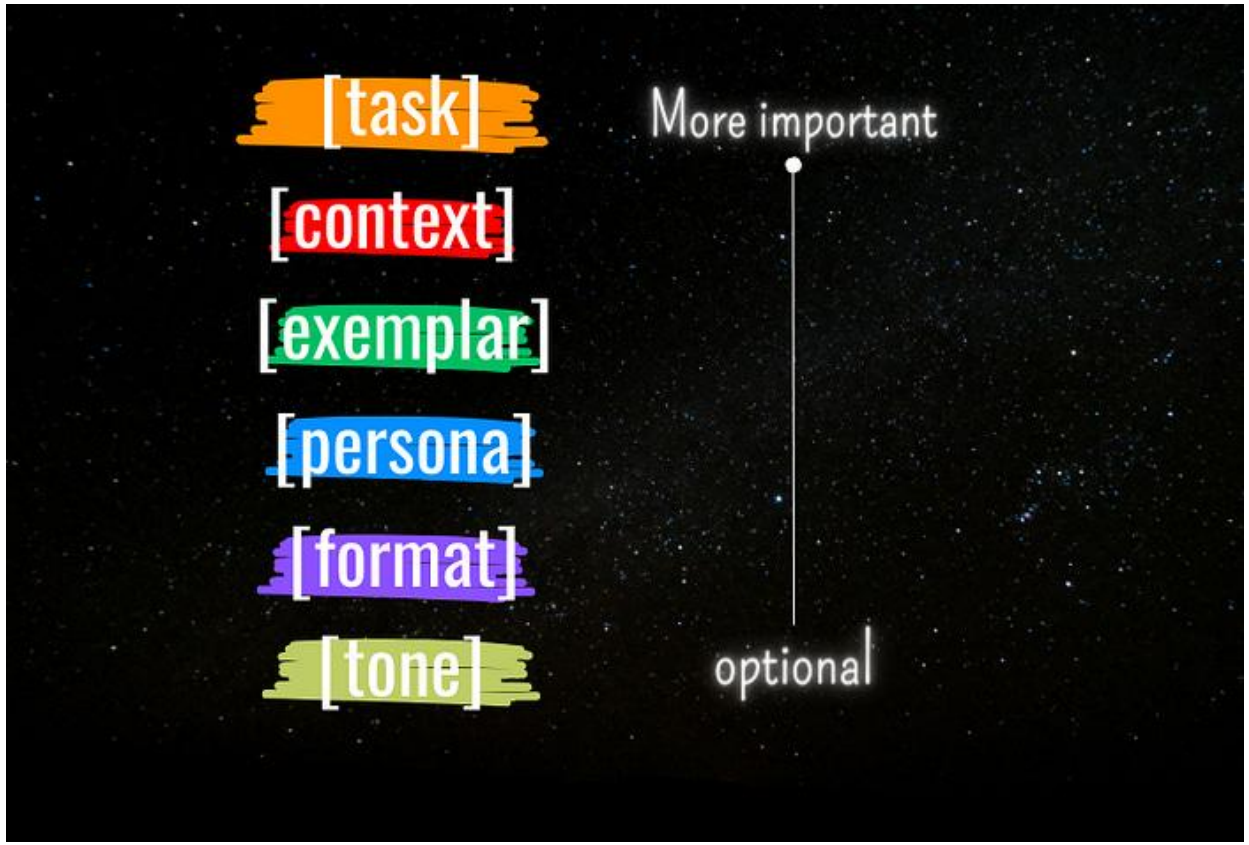
04 — Prompt Engineering

Prompting

- **Cos’è un Prompt**
  - Un prompt è l'input testuale che guida il comportamento del modello, fungendo da istruzione che definisce cosa il LLM deve generare
- **Importanza della formulazione**
  - La qualità dell'output dipende direttamente da come viene formulato il prompt: specificità, contesto e chiarezza sono fondamentali
- **Struttura del Prompt**
  - Un prompt efficace include: contesto, istruzioni chiare, formato desiderato dell'output e eventuali vincoli o esempi
- **Iterazione e raffinamento**
  - Il prompting è un processo iterativo: si parte da una versione base e si raffina progressivamente per ottimizzare i risultati

Cos’è un prompt e perché conta

- Il prompt = l’istruzione al modello
- Non magia: è design



Anatomia di un buon prompt

Strategia 1: Chiarezza & Contesto

- Evita vaghezze → specifica scopo, pubblico, vincoli
- Evita multi-domande in un unico prompt

Esempi:

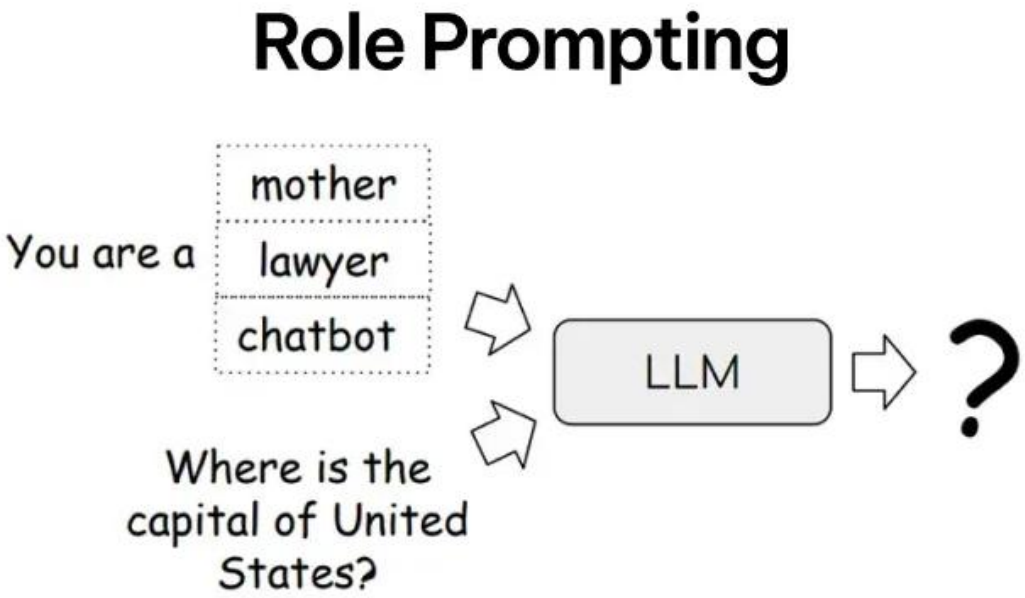
Spiegami il machine learning.

Spiega il machine learning in 5 bullet per studenti liceali (16–18 anni).
Vincoli: linguaggio semplice, niente formule. Includi: 1 esempio quotidiano, 1 errore comune, 1 frase finale di riepilogo

Spiega il machine learning per un ingegnere informatico con esperienza nel campo.

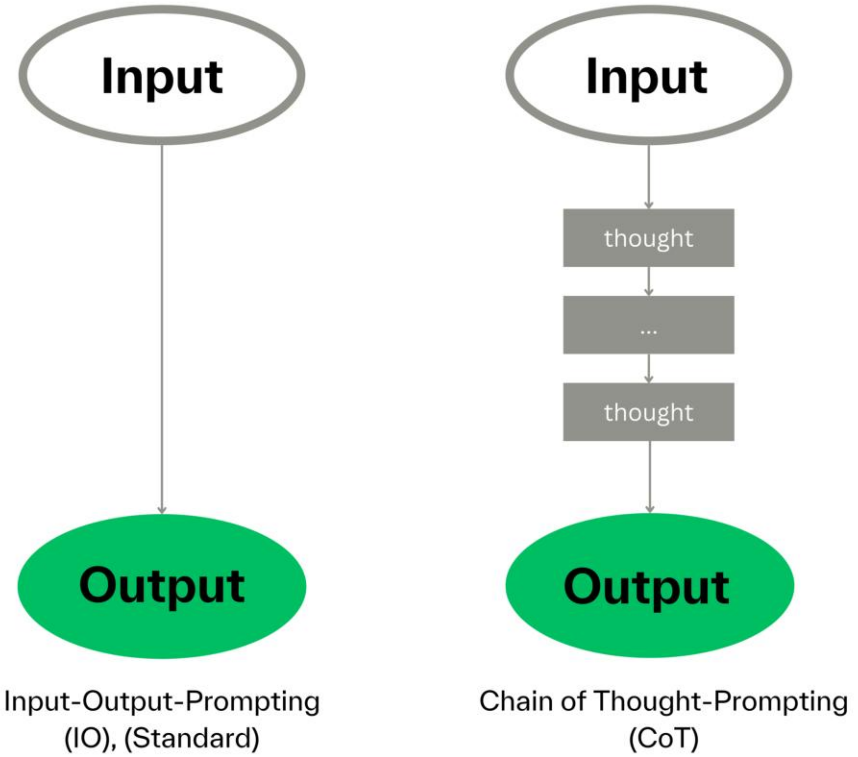
Strategia 2: Ruoli & Istruzioni esplicite

- “Agisci come…”
- Istruzioni operative e limiti
- Tono & audience



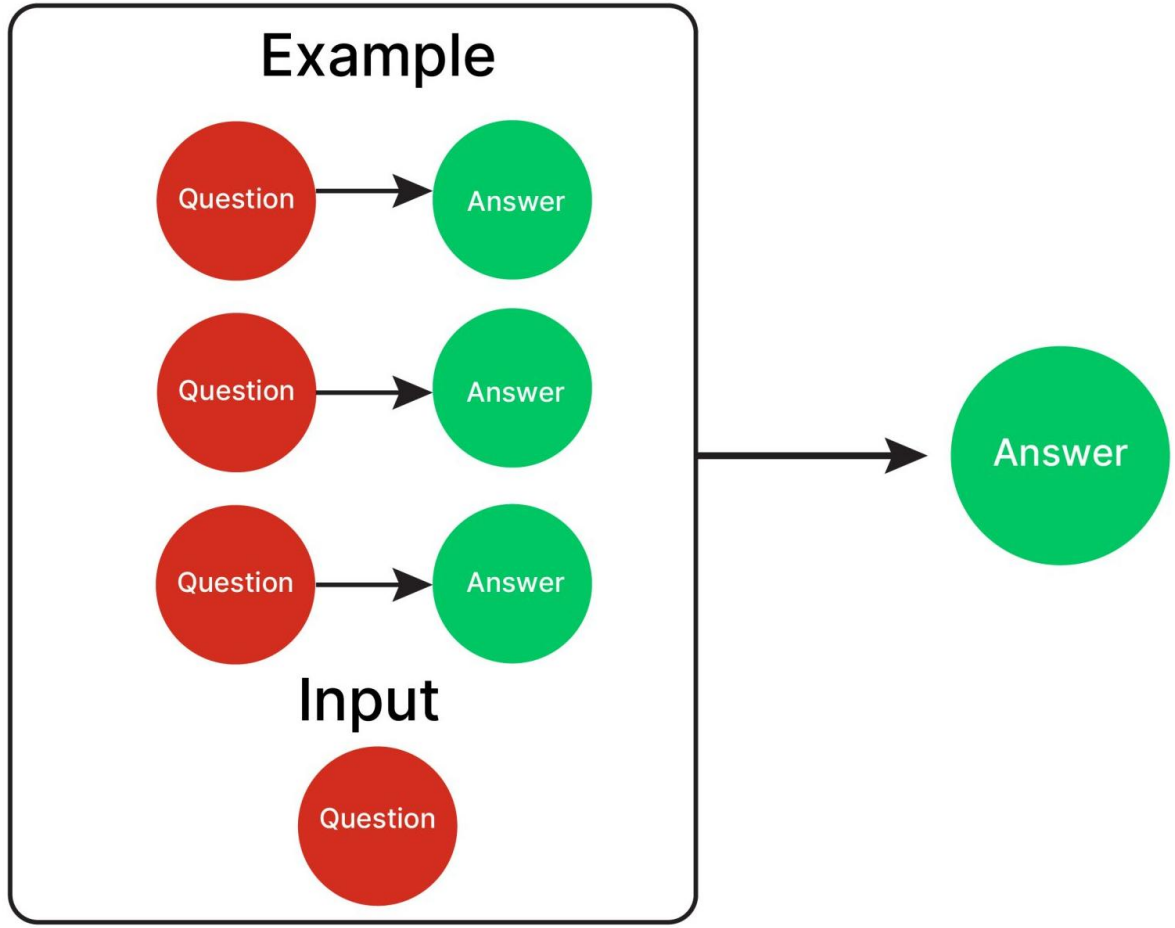
Strategia 3: Step-by-step (Ragionamento)

- Pianifica → Controlla → Rispondi
- “Prima proponi un piano, poi esegui.”



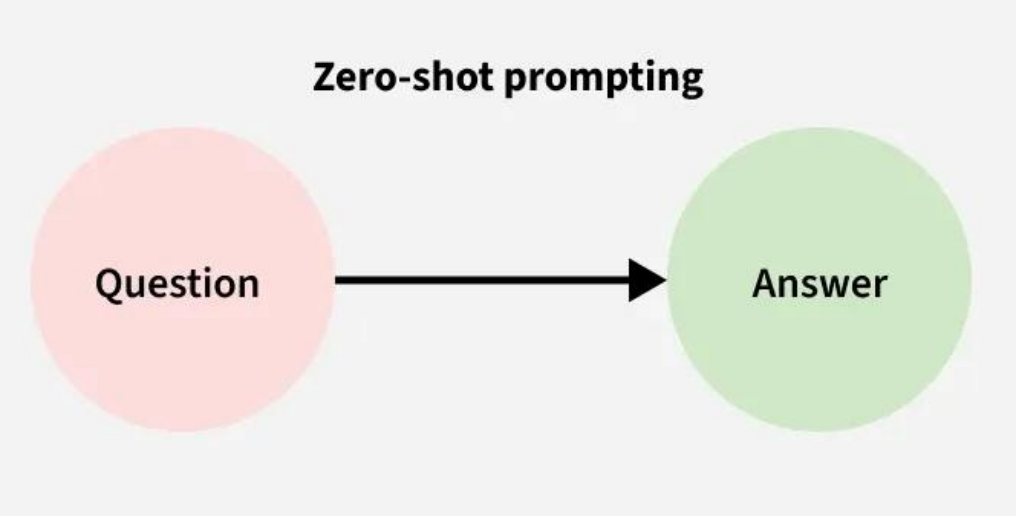
Strategia 4: Few-shot prompting

- 2–5 esempi guidano lo stile/struttura
- Mostra input → output attesi



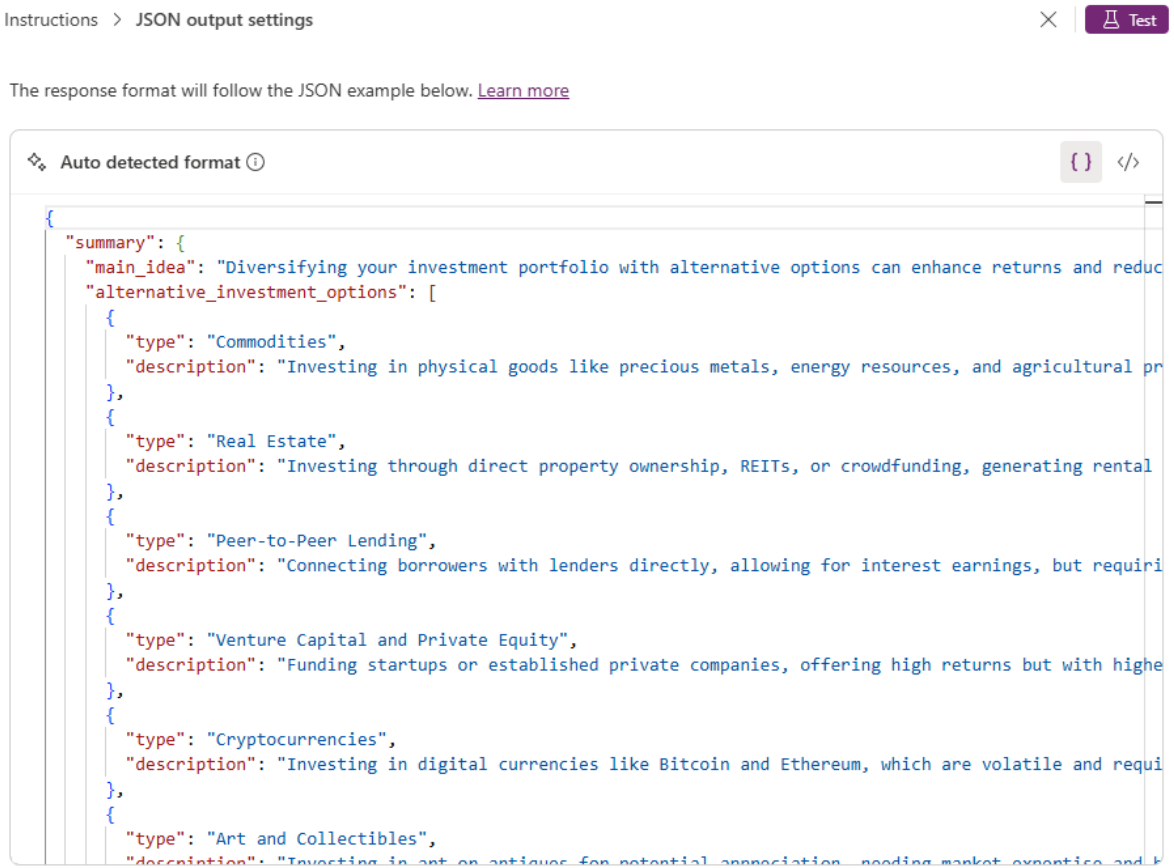
Strategia 4b: zero-shot prompting

- Il modello può rispondere senza esempi
- Funziona bene con: stili noti, compiti comuni, formati standard
- Se vuoi un formato personalizzato → usa few-shot



Controllo dell’output (formati & vincoli)

- **Formati deterministici**: JSON, tabella, elenco
- **Vincoli**: lunghezza, tono, audience
- **Anti-rumore**: “Non aggiungere testo oltre al JSON”



Chain of Thought

La **Chain of Thought** è una tecnica utilizzata per migliorare la capacità di risolvere problemi complessi, suddividendo il processo decisionale in passaggi logici intermedi. Invece di generare una risposta diretta, il modello viene incoraggiato a produrre una serie di ragionamenti passo-passo che portano alla soluzione finale, rendendo il processo più trasparente e spesso più accurato.

Prompt

Per risolvere il problema 'Se un treno parte da Roma a Milano, viaggiando a 200 km/h, e la distanza è di 570 km, quanto tempo impiega per arrivare?', **pensa passo dopo passo**: Prima, identifica la velocità del treno. Poi, usa la formula tempo = distanza/velocità per calcolare il tempo di viaggio.

Risposta

Per risolvere il problema, seguiamo i passaggi indicati:
1. Identifica la velocità del treno: La velocità del treno è data come 200 km/h.
2. Usa la formula tempo = distanza/velocità:
    * La distanza tra Roma e Milano è 570 km.
    * La velocità del treno è 200 km/h.

Quindi, il tempo di viaggio t può essere calcolato come:
t=distanza/velocita`=570 km / 200 km/h=2.85 ore

Il treno impiega 2.85 ore per arrivare da Roma a Milano.

05 — Strumenti - Perplexity AI

Perplexity AI: il motore di ricerca basato su Intelligenza Artificiale

Perplexity AI è un assistente AI progettato per rispondere
alle domande utilizzando informazioni aggiornate dal we

Combina:

- Large Language Models
- ricerca web in tempo reale
- citazioni automatiche delle fonti

Perplexity AI vs ChatGPT: differenze principali

	Caratteristiche	Limiti principali
Scopo principale	Ricerca informazioni	Assistente generale
Accesso al web	Nativo	Dipende dalla modalità
Citazioni fonti	Sempre presenti	Non sempre
Aggiornamento dati	Tempo reale	Variabile
Scrittura e creatività	Limitata	Molto avanzata

05 — Strumenti - NotebookLM

📘 NotebookLM

- **NotebookLM** è un’applicazione sviluppata da **Google DeepMind** per creare notebook intelligenti alimentati da modelli linguistici (LLM).
  - 📁 Organizza i documenti e appunti.
  - 🤖 Comprende e riassume i contenuti
  - 🗣️ Genera testi o spiegazioni basate sulle fonti

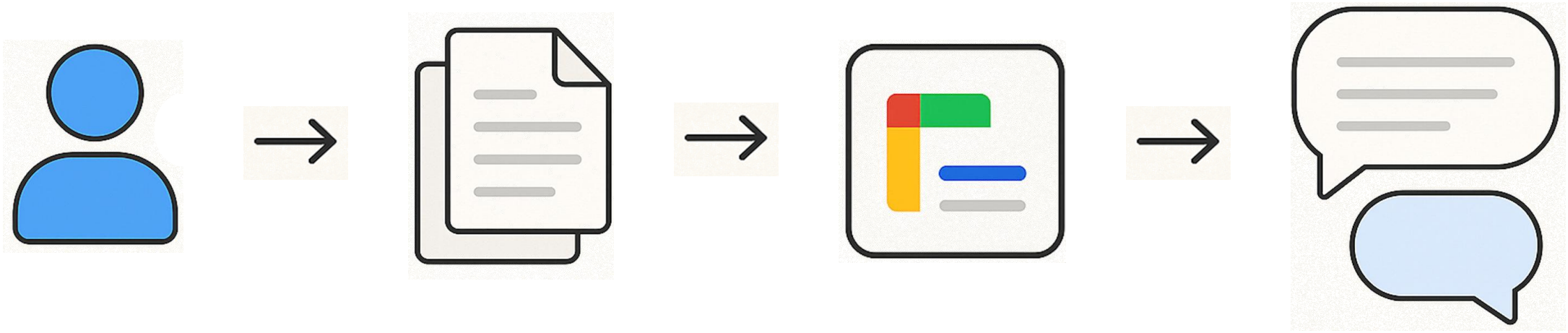
NotebookLM

DeepMind

⬇

Un **assistente personale** che studia i materiali forniti e aiuta ad elaborarli in modo chiaro e coerente

⚙️ Come funziona NotebookLM



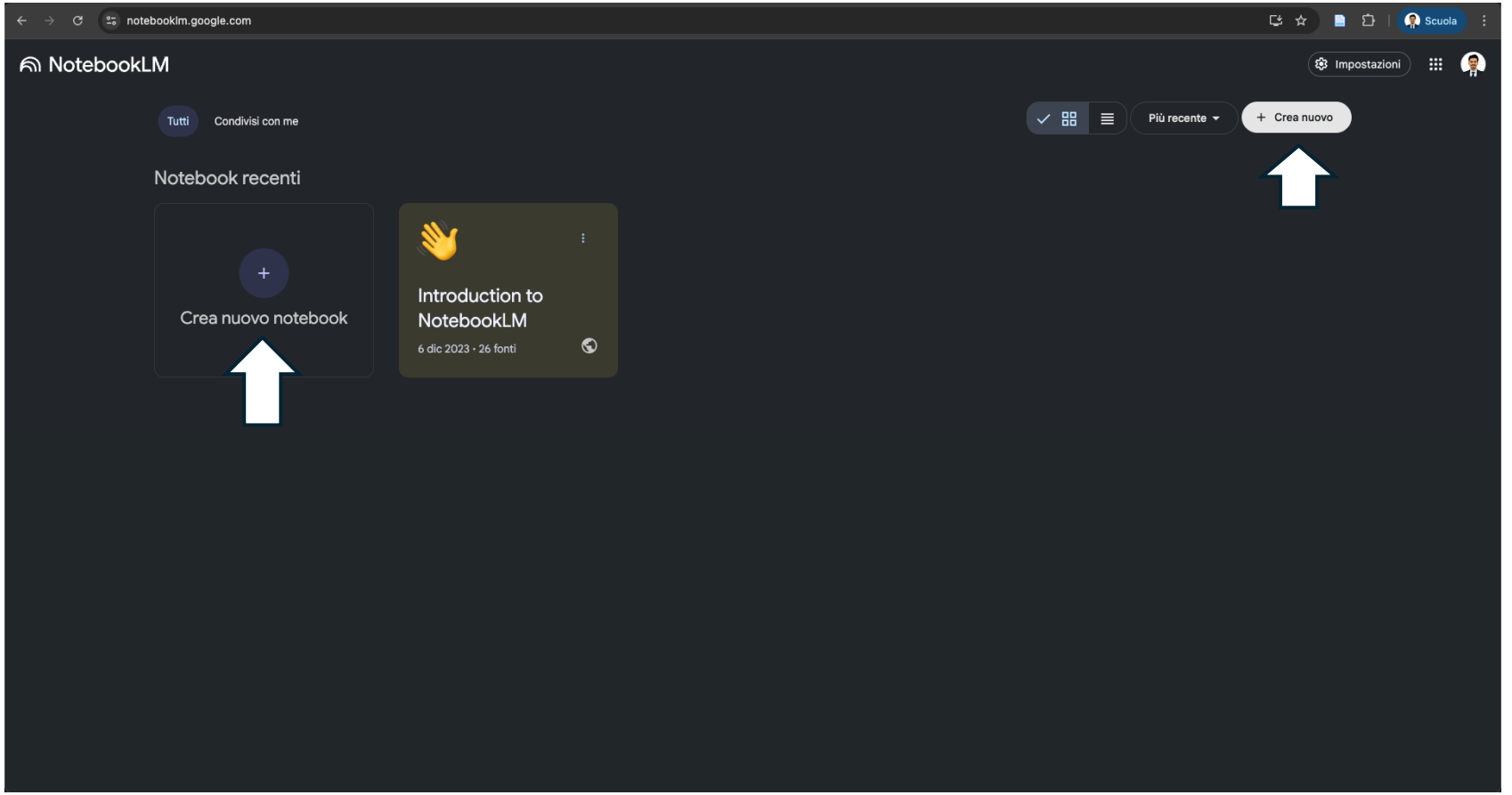
UTENTI

DOCUMENTI

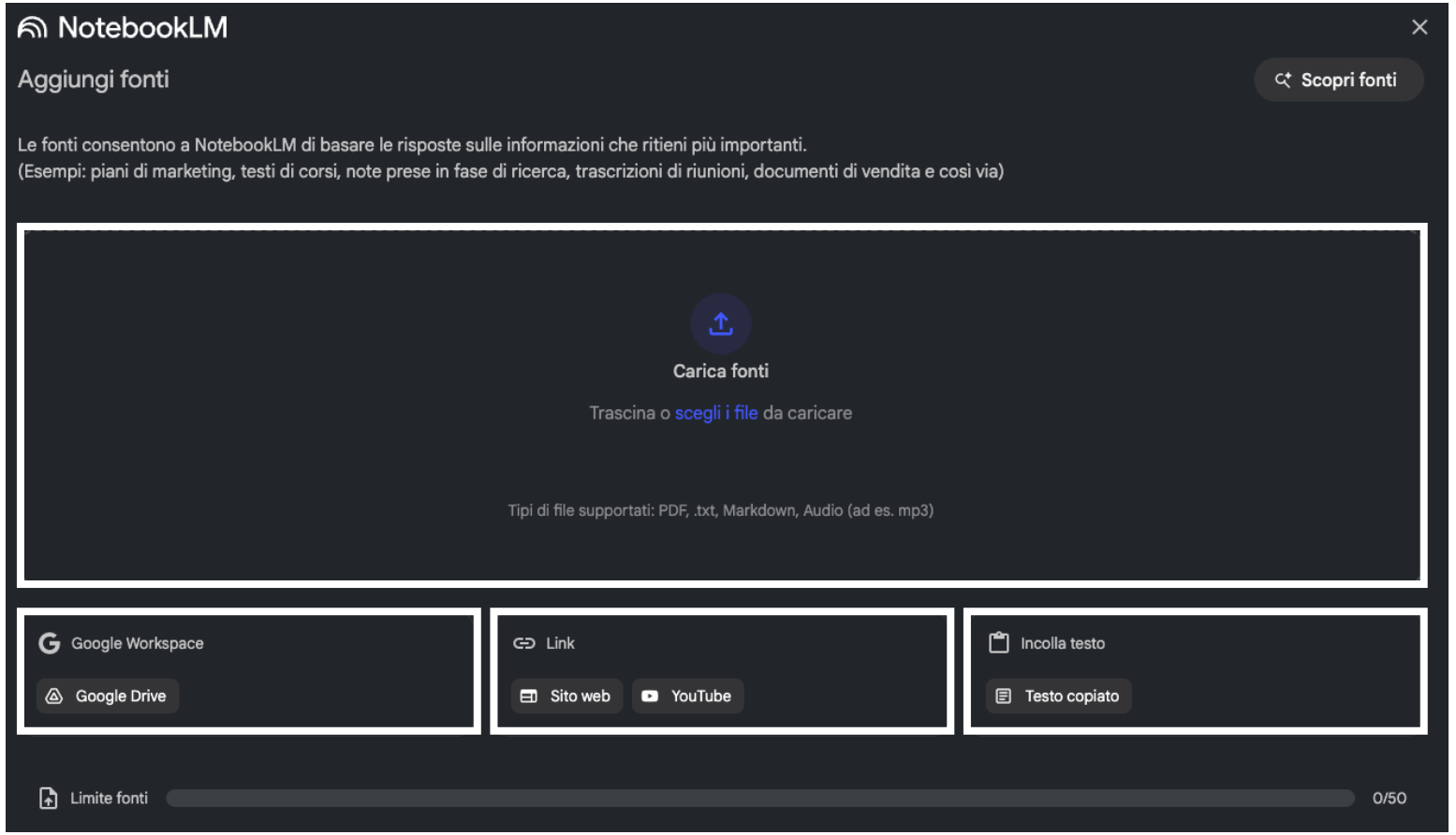
NOTEBOOKLM

DOMANDE / RISPOSTE

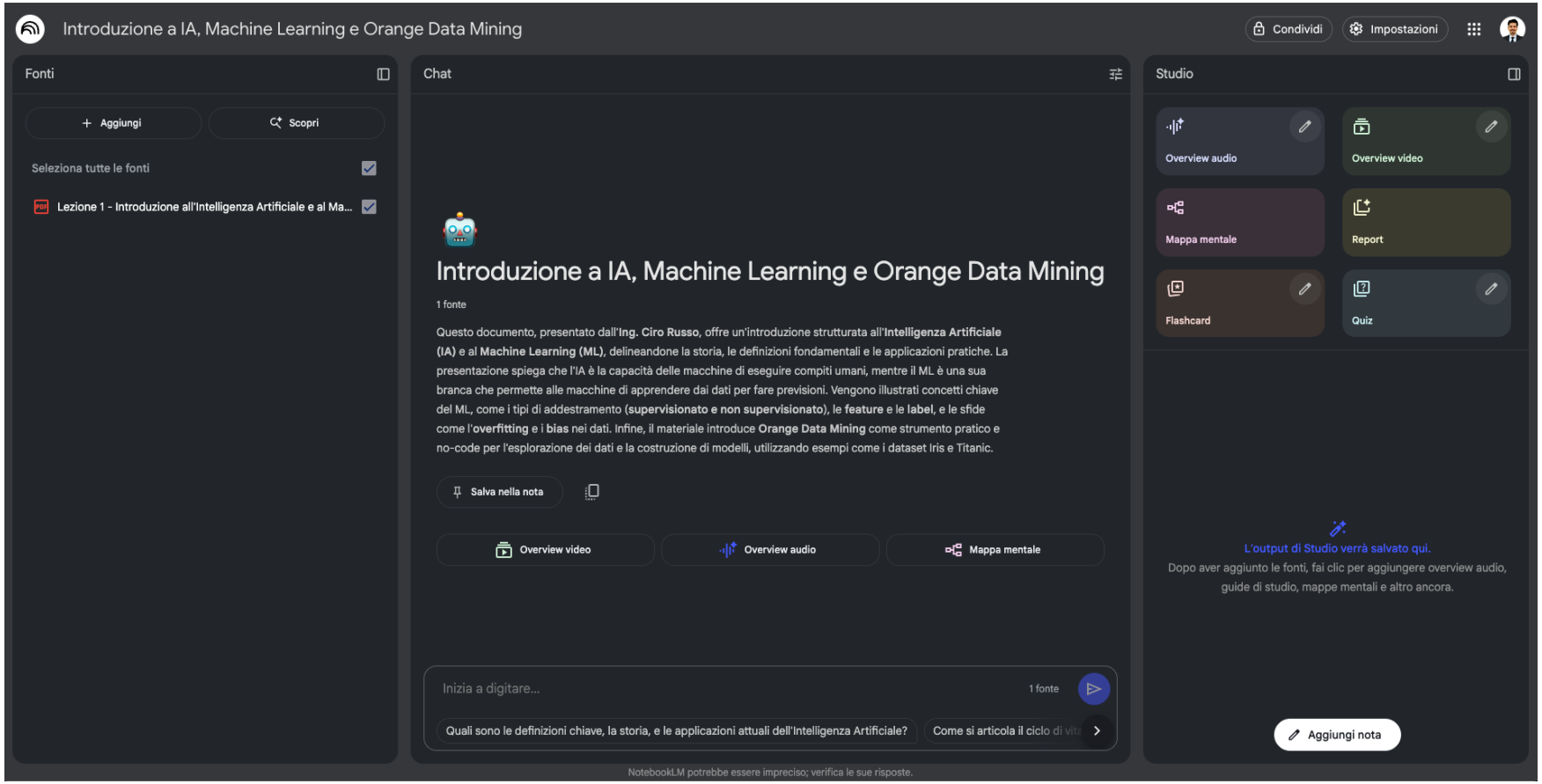
L’interfaccia di NotebookLM



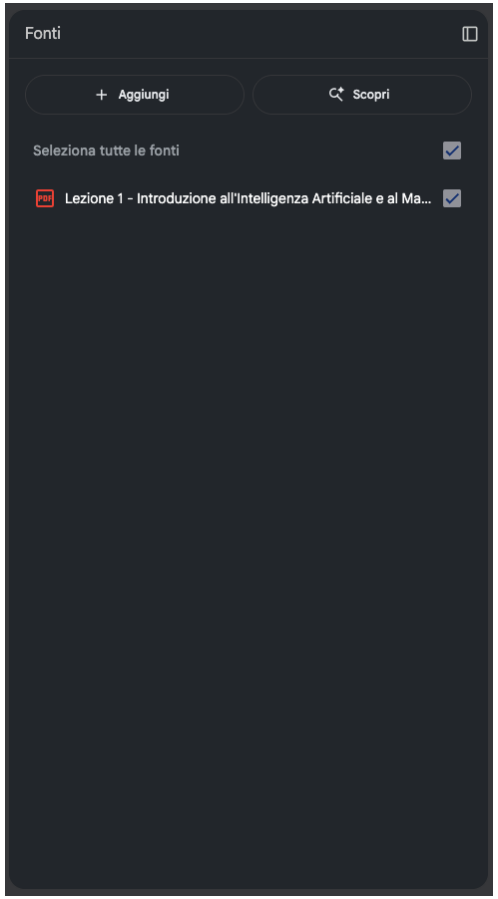
Il Notebook


NotebookLM
Aggiungi fonti
Scopri fonti
Le fonti consentono a NotebookLM di basare le risposte sulle informazioni che ritieni più importanti.
(Esempi: piani di marketing, testi di corsi, note prese in fase di ricerca, trascrizioni di riunioni, documenti di vendita e così via)
Carica fonti
Trascina o scegli i file da caricare
Tipi di file supportati: PDF, .txt, Markdown, Audio (ad es. mp3)
Google Workspace
Google Drive
Link
Sito web
YouTube
Incolla testo
Testo copiato
Limite fonti
0/50

Il Notebook


Introduzione a IA, Machine Learning e Orange Data Mining
Condividi
Impostazioni
Fonti
Aggiungi
Scopri
Seleziona tutte le fonti
Lezione 1 - Introduzione all'Intelligenza Artificiale e al Ma...
Chat
Introduzione a IA, Machine Learning e Orange Data Mining
1 fonte
Questo documento, presentato dall'Ing. Ciro Russo, offre un'introduzione strutturata all'Intelligenza Artificiale (IA) e al Machine Learning (ML), delineandone la storia, le definizioni fondamentali e le applicazioni pratiche. La presentazione spiega che l'IA è la capacità delle macchine di eseguire compiti umani, mentre il ML è una sua branca che permette alle macchine di apprendere dai dati per fare previsioni. Vengono illustrati concetti chiave del ML, come i tipi di addestramento (supervisionato e non supervisionato), le feature e le label, e le sfide come l'overfitting e i bias nei dati. Infine, il materiale introduce Orange Data Mining come strumento pratico e no-code per l'esplorazione dei dati e la costruzione di modelli, utilizzando esempi come i dataset Iris e Titanic.
Salva nella nota
Overview video
Overview audio
Mappa mentale
Inizia a digitare...
1 fonte
Quali sono le definizioni chiave, la storia, e le applicazioni attuali dell'Intelligenza Artificiale?
Come si articola il ciclo di vita
Studio
Overview audio
Overview video
Mappa mentale
Report
Flashcard
Quiz
L'output di Studio verrà salvato qui.
Dopo aver aggiunto le fonti, fai clic per aggiungere overview audio, guide di studio, mappe mentali e altro ancora.
Aggiungi nota
NotebookLM potrebbe essere impreciso; verifica le sue risposte.

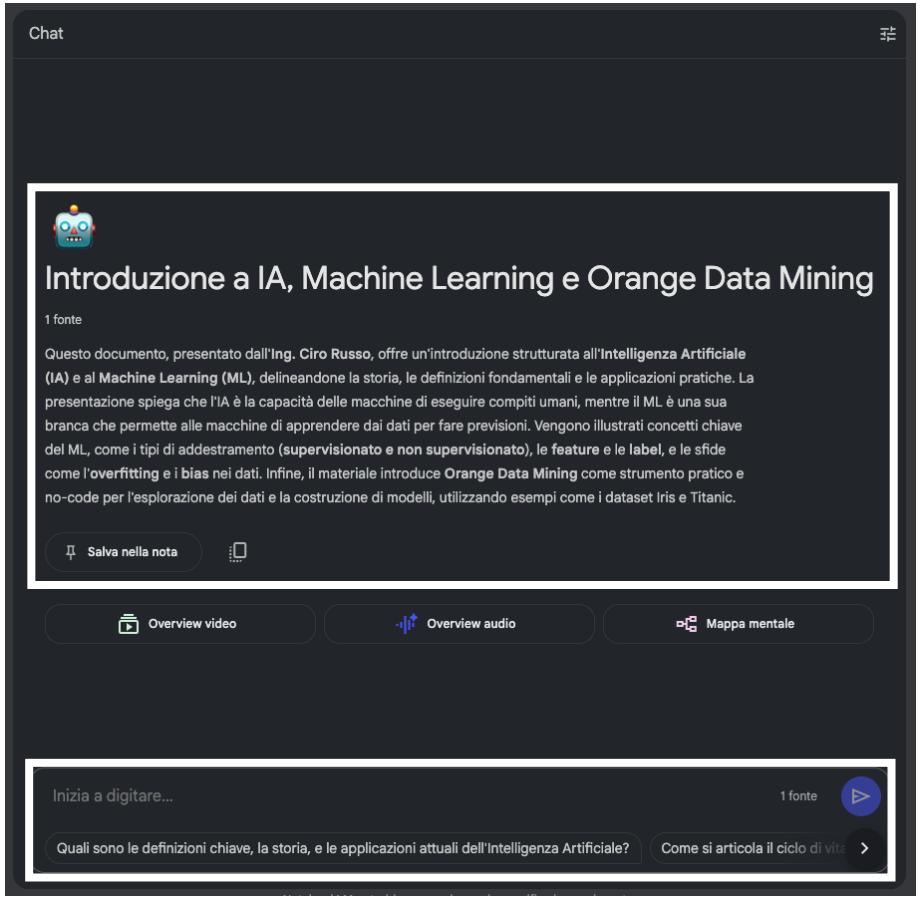
Il Notebook - Fonti



- **Fonti**
  - Contiene i documenti caricati
    → si possono aggiungere, selezionare o rimuovere le fonti del notebook
    → ogni fonte può essere un PDF, un testo, un link o un file audio

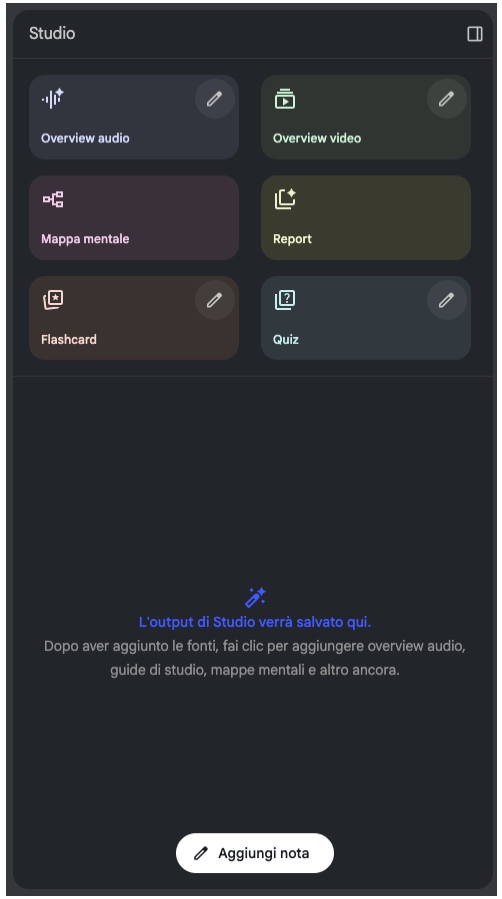
È la **"memoria"** dell'assistente

Il Notebook - Chat



- **Chat**
  - È il **cuore interattivo**: si possono fare domande, chiedere riassunti o spiegazioni.
    → le risposte si basano **solo** sulle fonti caricate
    → troviamo già un **primo riassunto automatico** del documento

📓 Il Notebook - Studio



- **Studio** 🎧
  - Strumenti di **approfondimento automatico**
    - Overview audio 🎧
      → sintesi vocale automatica dei contenuti caricati
    - Overview video 🎬
      → breve video riassuntivo che presenta i contenuti
    - Mappa mentale 🗺️
      → mappa dei concetti principali del documento
    - Report 🗂️
      → riassunto testuale strutturato dei contenuti
    - Flashcard 🪧
      → schede domanda-risposta per l'apprendimento
    - Quiz 📝
      → domande a risposta multipla basate

💻 Le applicazioni di NotebookLM

- NotebookLM è uno **strumento versatile**: può adattarsi a contesti molto diversi, dalla gestione dei documenti al supporto visivo e all'apprendimento individuale.
- Durante questa lezione vedremo tre esempi concreti di utilizzo, ognuno dedicato a una delle principali funzionalità del sistema.

1. 📄 Dai Documenti alle Delibere 🏛️ NotebookLM come Assistente Documentale

   → analizza più fonti e supporta la **redazione automatica di documenti**.

2. 📖 Dal Manuela al Montaggio 🪑 NotebookLM come Assistente Montatore

   → trasforma le istruzioni di un manuale IKEA in una **mappa concettuale interattiva**.

3. 🎓 Dallo Studio alla Conoscenza 💡 NotebookLM come Assistente per l'Apprendimento

   → genera automaticamente **video riassuntivi** e **quiz interattivi** per lo studio e la verifica.

📄 Dai Documenti alle Delibere 🏛️

NotebookLM come Assistente Documentale

NotebookLM come Assistente Documentale

- Ogni giorno, in qualsiasi ufficio o ente, viene gestita un'**enorme quantità di documenti**: regolamenti, verbali, relazioni, note interne.
  - spesso le informazioni sono **frammentate** in file diversi, **difficili da cercare** e **da ricomporre** in modo coerente.

- NotebookLM aiuta a superare questa complessità: **analizzando esclusivamente i materiali caricati** (PDF, documenti Google, pagine web).

supporto alla **redazione automatica di una delibera**, sintetizzando più fonti in un unico testo coerente e strutturato secondo il linguaggio amministrativo

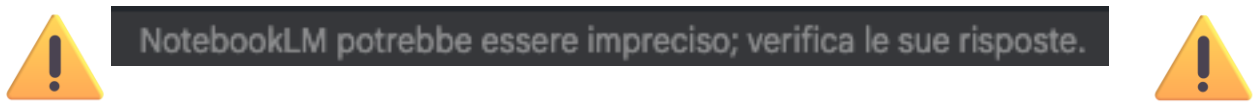
NotebookLM come Assistente Documentale

1. Crea un **nuovo notebook** e *assegna un titolo* (es. Bozza di Delibera).
2. Carica le **fonti**:
   - Fonte 1 Regolamento
   - Fonte 2 Verbale della Riunione
   - Fonte 3 Nota Tecnica
3. NotebookLM genera un **riassunto automatico** dei contenuti caricati.
4. Nella chat scriviamo il **prompt**:

> Genera una bozza di delibera basata esclusivamente sulle fonti caricate.
> Mantieni la struttura tipica di una delibera amministrativa con le sezioni:
> Premesso che → sintesi dei riferimenti normativi e delle motivazioni
> Considerato che → elementi emersi dalle riunione e dalle note tecniche
> Delibera → punti deliberativi numerati con verbi all'infinito

📄 Analizziamo la bozza di delibera

- 📋 **Struttura del testo**
  - Verifichiamo che siano presenti le tre sezioni:
    - ❑ Premesso che → motivazioni e riferimenti normativi
    - ❑ Considerato che → elementi tratti da verbali e note
    - ❑ Delibera → punti decisionali chiari e numerati
- 📚 **Aderenza alle fonti**
  - NotebookLM cita automaticamente le **fonti utilizzate** nella risposta
  - Ogni frase può essere ricondotta al documento di origine
- 🏛️ **Linguaggio e Tono**
  - Il testo è formale e coerente con lo stile amministrativo
  - Può essere migliorato con prompt mirati (es. "*Riformula in tono più tecnico*")



📖 Dal Manuela al Montaggio 🪑

NotebookLM come Assistente Montatore

NotebookLM come Assistente Montatore

- Dobbiamo montare un mobile IKEA. Abbiamo il manuale, ma le istruzioni sono composte da simboli, frecce e passaggi poco chiari. Ogni fase dipende dalla precedente, e spesso serve una **visione d'insieme**.

- Partendo da un manuale IKEA, NotebookLM analizza le **istruzioni di montaggio** e riconosce le fasi, gli strumenti e le relazioni tra i componenti del mobile.

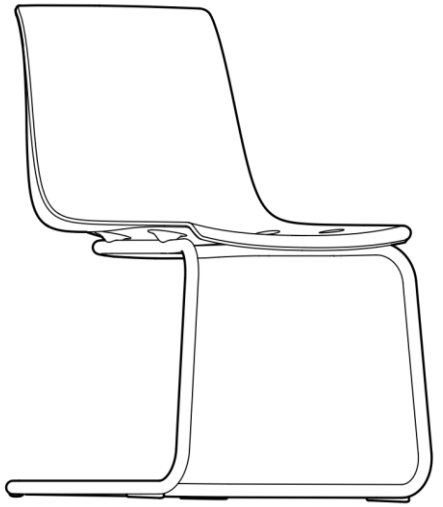
supporto alla **creazione automatica di una mappa concettuale**, che rappresenta visivamente il processo di montaggio, mostrando le connessioni logiche tra passaggi, parti e azioni necessarie

NotebookLM come Assistente Montatore

1. Crea un nuovo notebook e assegna un *titolo* (ad es. Manuale IKEA)
2. Carica il **manuale IKEA**:
   - Manuale IKEA - Tobias Chair
3. Clicca su **Mappa Mentale** per attivare la generazione automatica

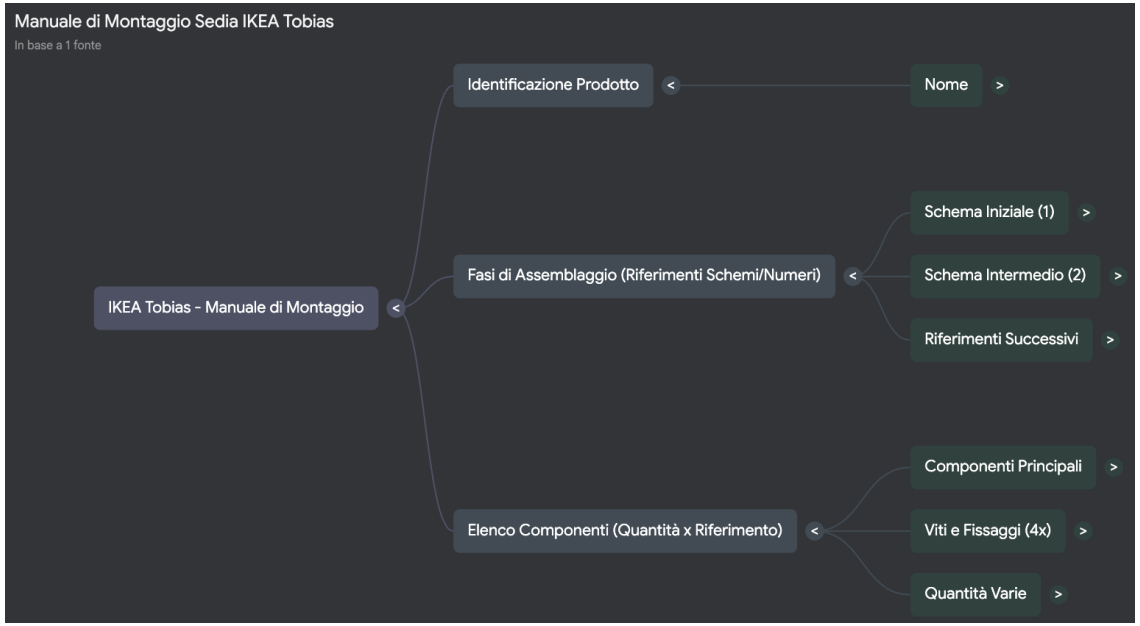
TOBIAS

NotebookLM analizza le fonti (il manuale) e crea una rappresentazione visuale interattiva dei concetti principali



Esplora la Mappa

- Una volta generata:
  - ❑ La mappa può essere **espansa** o **ridotta** cliccando sui **nodi concettuali**
  - ❑ Si possono leggere brevi descrizioni per ogni concetto
  - ❑ Passando il cursore sui collegamenti, si può identificare la fonte da cui proviene l'informazione



🎓 Dallo Studio alla Conoscenza 💡

NotebookLM come Assistente per l’Apprendimento

🎓 NotebookLM come Assistente per l'Apprendimento

- Siamo davanti ad un capitolo molto complesso. Abbiamo preso appunti, ma non è facile capire cosa ricordare davvero o come collegare le idee. A volte abbiamo bisogno di un ***tutor*** che ci spieghi i concetti in modo chiaro

- Dall'analisi dei nostri materiali di studio (appunti o libri), NotebookLM genera automaticamente:
  1. un **Overview Video**, che riassume visivamente i concetti chiave
  2. un **Quiz interattivo**, per verificare la comprensione e consolidare l'apprendimento

supporto personalizzato allo studio individuale, che trasforma la lettura passiva in un'esperienza attiva, guidata e multimodale

🎓 NotebookLM come Assistente per l'Apprendimento

1. Crea un nuovo notebook e assegna un *titolo*
2. Carica il **materiale didattico**:
   - Fonte 1 Cos'è un Vulcano
   - Fonte 2 Tipi di Eruzione
   - Fonte 3 Struttura di un Vulcano
3. Clicca su **Overview Video** per generare un video sui contenuti caricati
4. Clicca su **Quiz** per generare domande a scelta multipla sulle fonti

NotebookLM li analizza il materiale e genera automaticamente un **Video Overview**, che spiega il contenuto in modo visivo, con narrazione e immagini, e un **Quiz interattivo**, che permette di verificare la comprensione dei concetti attraverso domande basate direttamente sulle fonti caricate

05 — Strumenti - Landbot

💬 Landbot

- **Landbot** è una piattaforma **no-code** che permette di creare chatbot personalizzati tramite un editor visuale a blocchi, senza scrivere codice.
- Può essere utilizzato su diversi canali:
  - 💻 **Webchat**
  - 📱 **WhatsApp**
  - 👷 **Chatbot di servizio o assistenza**

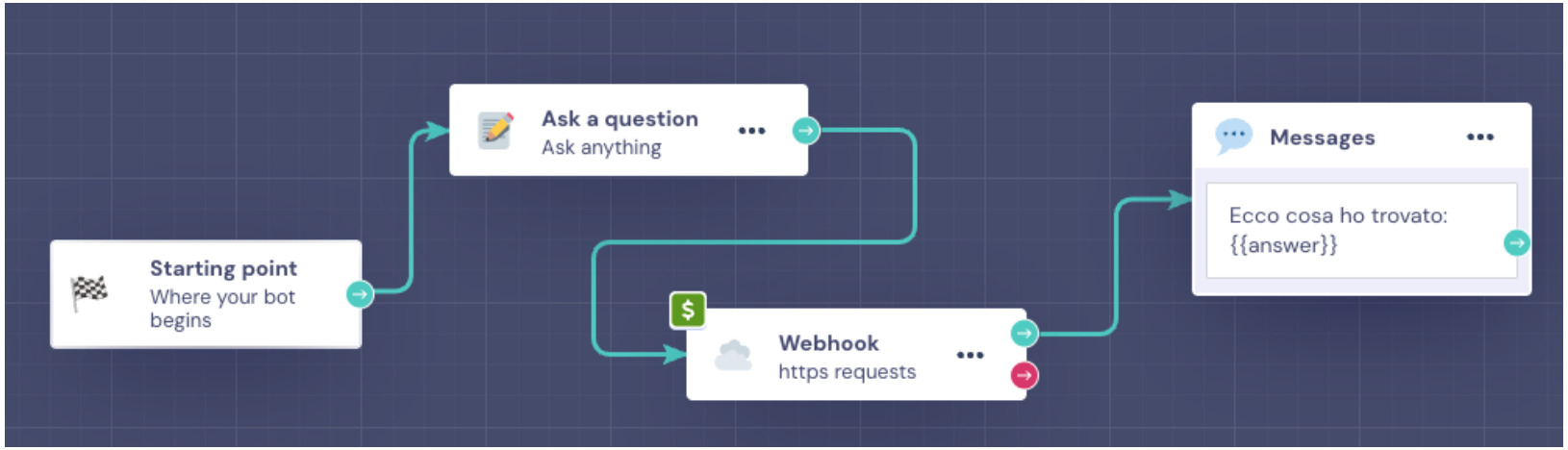
- Caratteristiche principali:
  - 🧱 **Flow Builder visuale** → costruisci la conversazione come un diagramma di blocchi
  - 🗣️ **Interazione guidata** → il bot risponde a scelte multiple o input liberi
  - 🔗 **Integrazioni** → con Google Sheets, API esterne, moduli di raccolta dati, email.
  - 📃 **Personalizzazione** → colori, testi, percorsi logici e variabili utente.
  - 📊 **Analisi e Statistiche** → per monitorare flussi e tassi di completamento.

💬 Chatbot con RAG (Retrieval-Augmented Generation)

- Il chatbot con RAG, invece, **combina ricerca** e **generazione**:
  1. Riceve una domanda libera dall'utente
  2. Cerca informazioni in un insieme di documenti o fonti (database, PDF, testi)
  3. Genera una risposta personalizzata, basata solo sui contenuti trovati
- 💡 È come unire un motore di ricerca (retrieval) con un modello linguistico (generation)

Chatbot con RAG (Retrieval-Augmented Generation)

- Landbot integra un modello LLM con un database esterno per creare un flusso RAG
- Landbot invia la richiesta a un sistema esterno tramite il **Webhook**, un piccolo "***ponte***" che collega il chatbot con altre fonti o modelli di AI.
- L'AI cerca nei **documenti collegati** e genera la **risposta più adatta**.

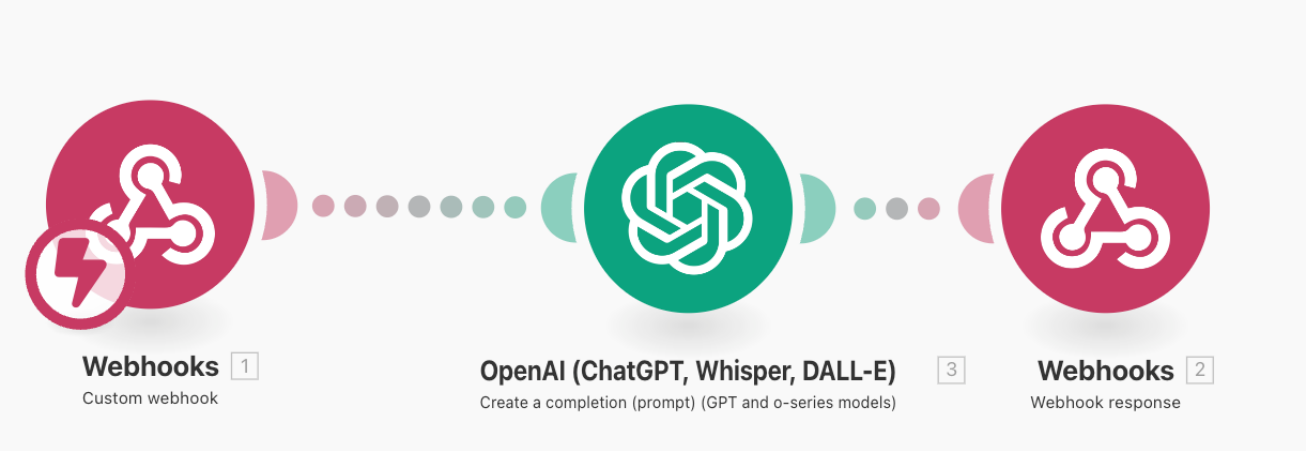


💬 Chatbot con RAG (Retrieval-Augmented Generation)

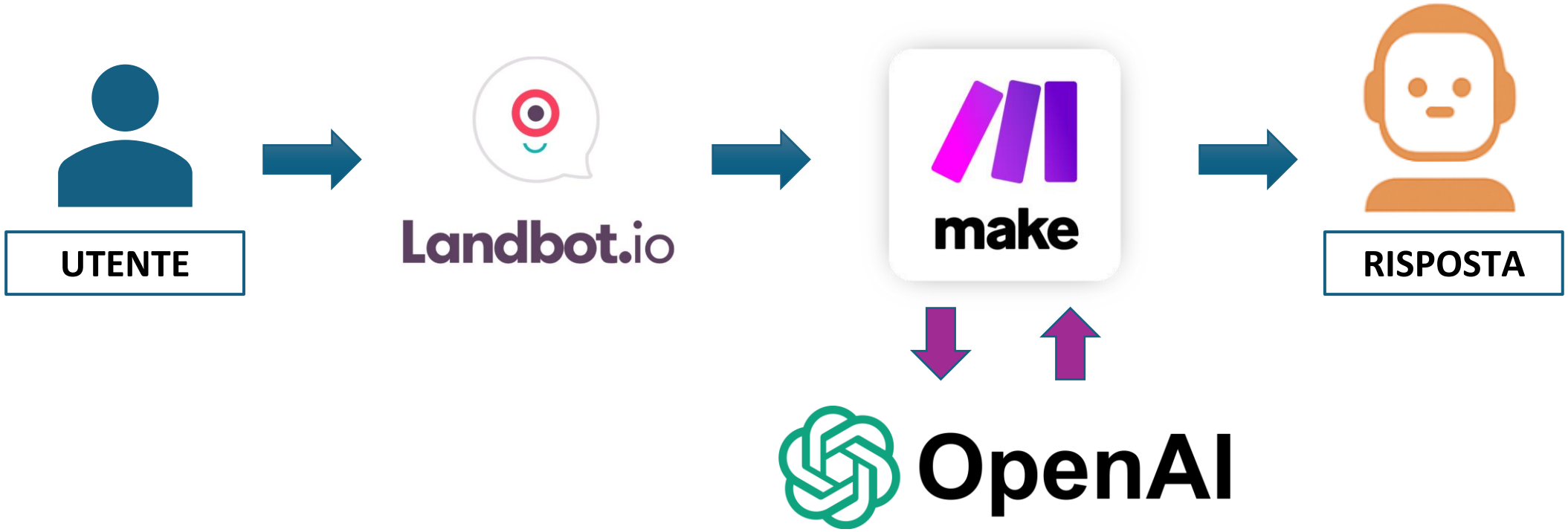
- Questa operazione comporta un **costo fisso** 💰 legato al modello LLM utilizzato:
  - ogni chiamata al modello (ovvero ogni risposta generata)
  - consuma **crediti o token** del servizio AI collegato.

Chatbot con RAG (Retrieval-Augmented Generation)

- Il chatbot utilizza un'integrazione tra **Landbot** e Make.com per collegarsi al modello di AI (es. *OpenAI*)
  - questo flusso consente di **inviare** la domanda dell'utente al modello, **ricevere** la risposta e restituirla in chat **in tempo reale**.



Chatbot con RAG (Retrieval-Augmented Generation)



Grazie per l’attenzione!

Prof.ssa Tiziana D’Alessandro

Prof. Cesare Davide Pace

email: tiziana.dalessandro@unicas.it, cesaredavide.pace@unicas.it

26/02/2025