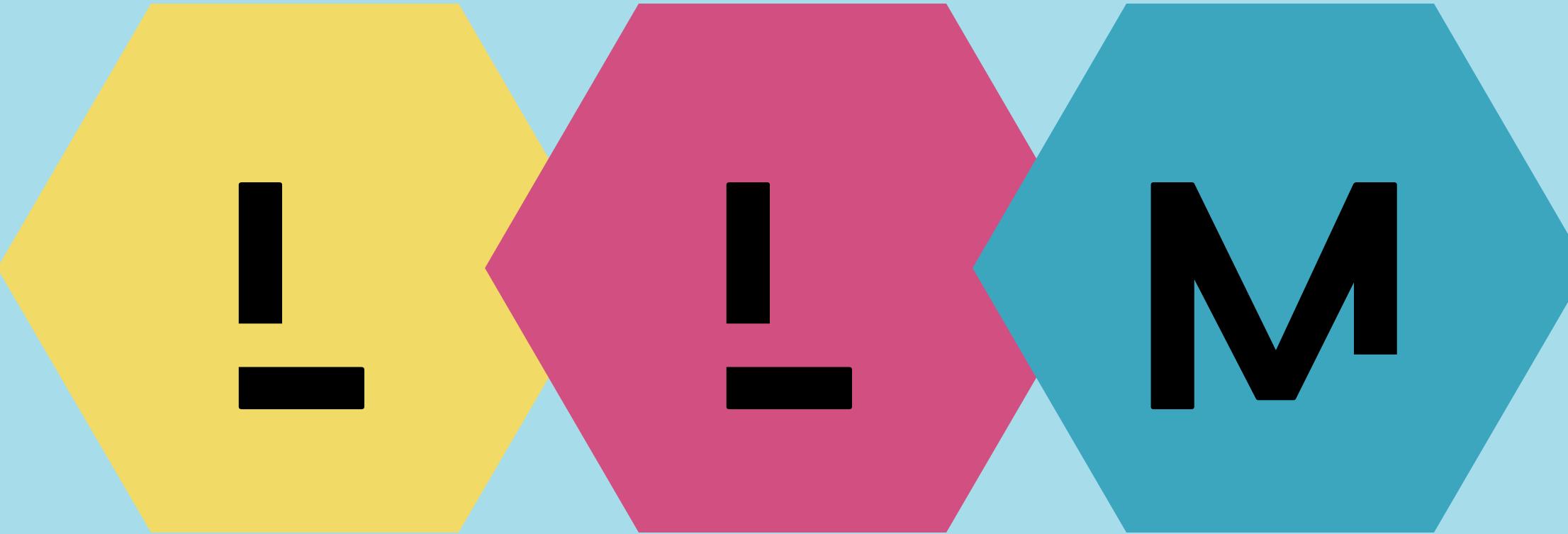


# Introduzione agli



A cura di Mosca Lucrezia e Scelfo Laura

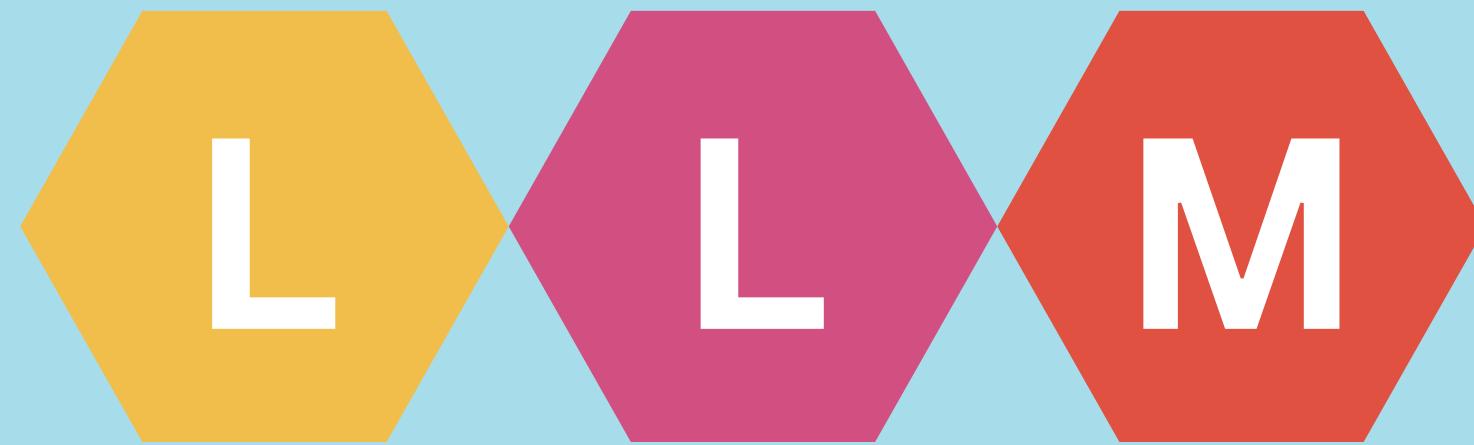
# Cosa sono gli LLM?

LLM sono i modelli di linguaggio di grandi dimensioni:



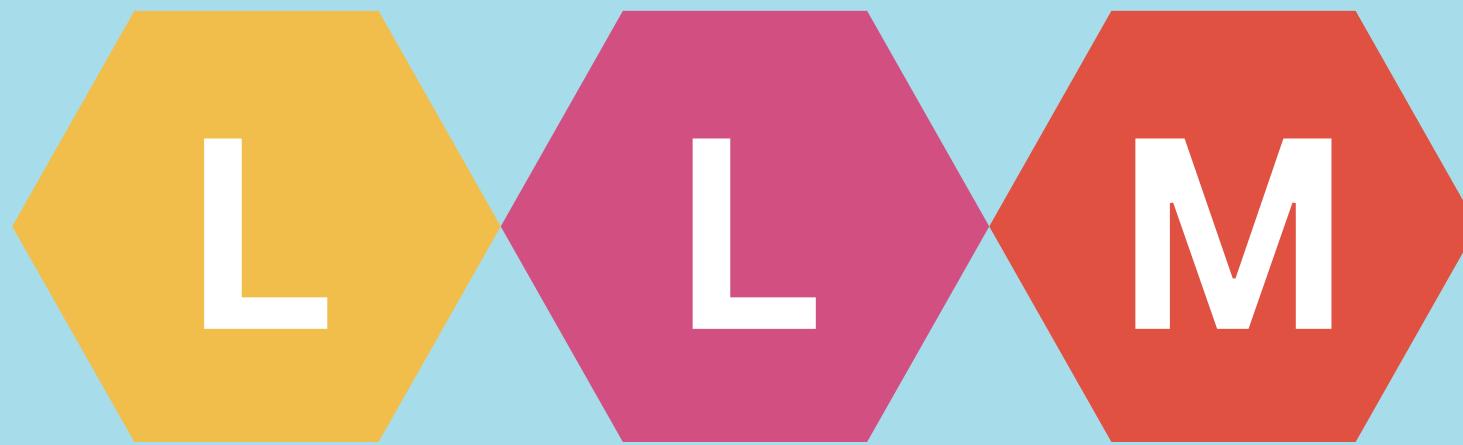
Large  
Language  
Models

Immaginate di avere un super cervello digitale capace di leggere milioni di libri e rispondere alle vostre domande!



I modelli di linguaggio di grandi dimensioni sono proprio questo: modelli che usano l'intelligenza artificiale per comprendere e generare testo in modo sorprendentemente simile a come farebbe una persona.

Immaginate di avere un super cervello digitale capace di leggere milioni di libri e rispondere alle vostre domande!



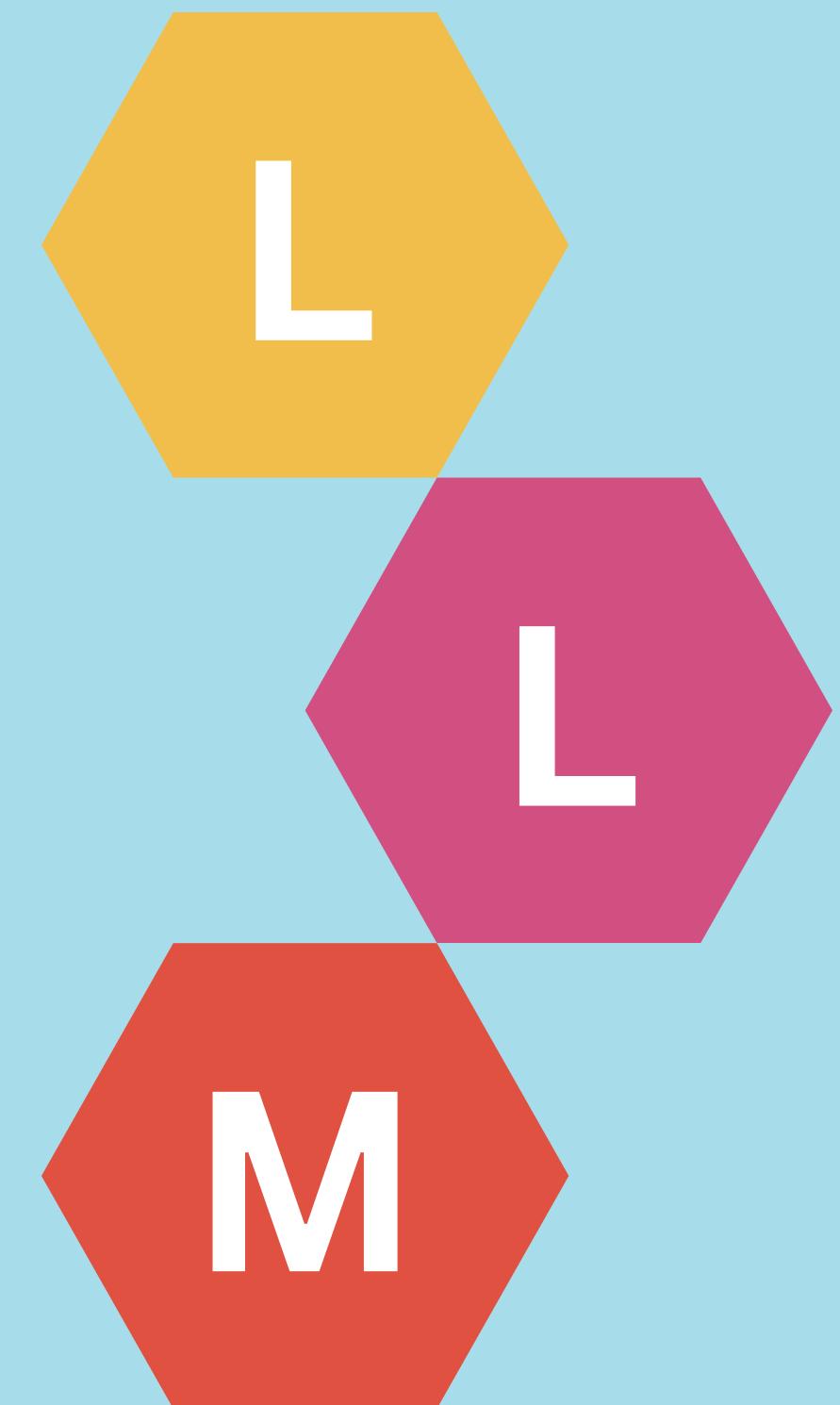
Utilizzano i Transformers, una tipologia di rete neurale, grazie ai quali il modello legge e apprende informazioni sui testi che gli sono stati dati in input per svolgere diversi compiti complessi grazie alla conoscenza acquisita.

# Come Funzionano?

Gli LLM imparano leggendo enormi quantità di testo, trovando schemi e relazioni tra le parole.

Questo li aiuta a:

- **Scrivere testi coerenti:** Creano frasi logiche ben strutturate.
- **Rispondere a domande:** Usano le informazioni acquisite per fornire risposte coerenti su argomenti molto complessi.
- **Tradurre lingue:** Capiscono il significato delle frasi e le trasformano in un'altra lingua mantenendo il senso.



# I Principali Approcci

I modelli possono funzionare in modi diversi a seconda del tipo di compito da svolgere:

- **Decoder only** (es. *GPT*): Generano testo parola dopo parola, come raccontare una storia o rispondere a una domanda.
- **Encoder only** (es. *BERT*): Leggono tutto il contesto in una volta, ideali per capire il significato globale di una frase.
- **Encoder–Decoder** (es. *T5*): Perfetti per traduzioni o riassunti perché trasformano un testo di input in un nuovo testo modificato.

**Quelli che verranno trattati in questo corso sono i Decoder only.**

# Le Principali Architetture – RAG

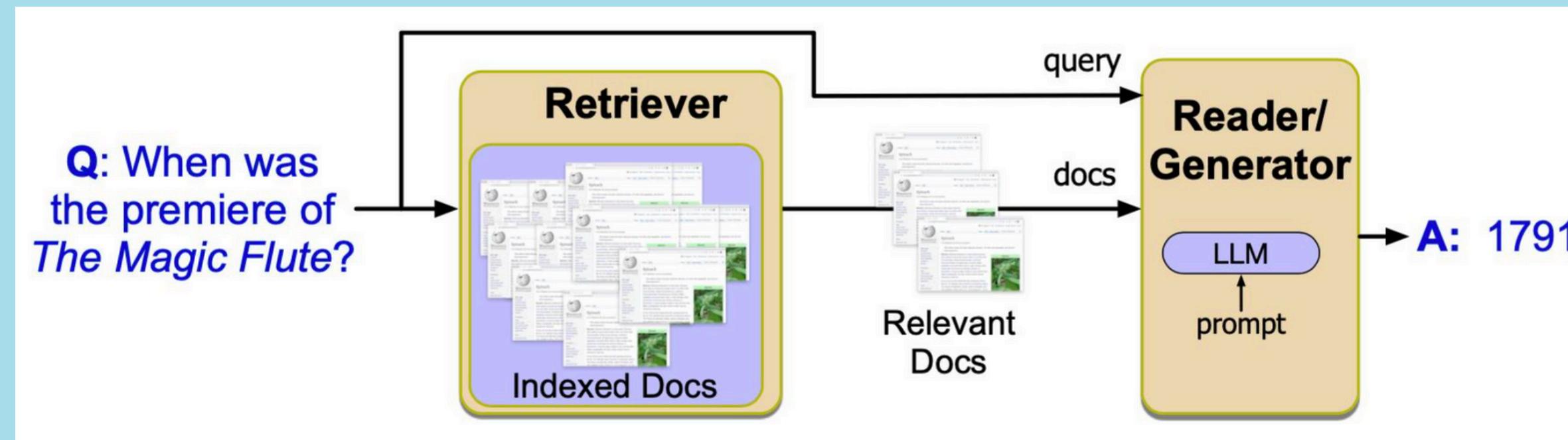
RAG sta per **Retrieval-Augmented Generation** e può essere tradotto come Generazione Aumentata dal Recupero. È un approccio Encoder-Decoder che aiuta i modelli di intelligenza artificiale a rispondere in modo più preciso e aggiornato alle domande.



# Le Principali Architetture – RAG

Questa architettura combina due capacità fondamentali:

- **Recupero delle informazioni (Retrieval)**: il modello cerca informazioni in una grande base di dati o su internet, come farebbe un motore di ricerca.
- **Generazione del testo (Generation)**: usa le informazioni trovate per scrivere una risposta chiara e comprensibile, come se stesse spiegando qualcosa a un amico.



# Esempio Pratico RAG

Immaginiamo di chiedere al modello:

*“Chi ha scoperto la gravità?”*

Un modello tradizionale potrebbe rispondere correttamente perché l'informazione è semplice.



# Esempio Pratico RAG

Ma se dovessimo chiedere:

***“Quali sono le ultime scoperte sull’energia solare?”***

Un modello normale potrebbe dare una risposta datata. Un sistema RAG, al contrario, cercherebbe articoli scientifici recenti e formulerebbe una risposta basata su fonti aggiornate!



# Perché è Utile RAG?

- **Risposte più accurate e aggiornate:** RAG aiuta i modelli a basarsi su informazioni reali, riducendo il rischio di errori.
- **Apprendimento continuo:** Anche se il modello è stato addestrato anni fa, può comunque trovare e usare informazioni nuove.
- **Versatilità:** È utile in tantissimi ambiti, dalla medicina alla storia, fino alla risoluzione di problemi complessi.

# Creare Testo in Modo Intelligente

Quando un modello scrive, sceglie ogni parola in base a una probabilità. Per migliorare la qualità del testo si usano diverse strategie di generazione:

- ***Top-k sampling***: Sceglie tra le parole più probabili, limitando la scelta a un numero fisso di parole con maggiore probabilità. È utile per avere risposte più sicure.
- ***Top-p sampling***: Seleziona dinamicamente le parole più probabili fino a coprire una certa percentuale di probabilità totale, adattandosi meglio al contesto.
- ***Temperature sampling***: Regola il livello di creatività: con una temperatura bassa il modello è più preciso, con una alta diventa più fantasioso e originale.

# Come Vengono Addestrati?

L'addestramento di un LLM è un processo complesso che richiede grandi quantità di dati e una potenza di calcolo elevata. Il modello legge enormi quantità di dati presenti nei dataset (come Wikipedia o articoli online) e cerca di prevedere la parola successiva in una frase.



# Come Vengono Addestrati?

Durante questo processo:

- **Ottimizzazione:** Il modello calcola quanto la sua previsione è distante dalla parola corretta. Aggiorna successivamente i suoi parametri, migliorando progressivamente.
- **Pre-addestramento:** Il modello impara una conoscenza generale del linguaggio leggendo testi di vario tipo, senza essere specializzato su un compito specifico.
- **Fine-tuning:** Il modello può essere perfezionato su dati specifici per compiti particolari (es. domande e risposte), diventando così più preciso per applicazioni reali.

# Valutare un Modello

Per capire se un modello svolge correttamente il suo compito si usano diverse metriche di valutazione. La più comune è la **perplessità** (PPL) che misura quanto bene il modello prevede il testo.

- **Perplessità bassa:** Il modello assegna alte probabilità alle parole corrette, dimostrando di aver capito il contesto.
- **Perplessità alta:** Il modello fatica a prevedere le parole giuste, segno che la sua comprensione è limitata.



# Valutare un Modello

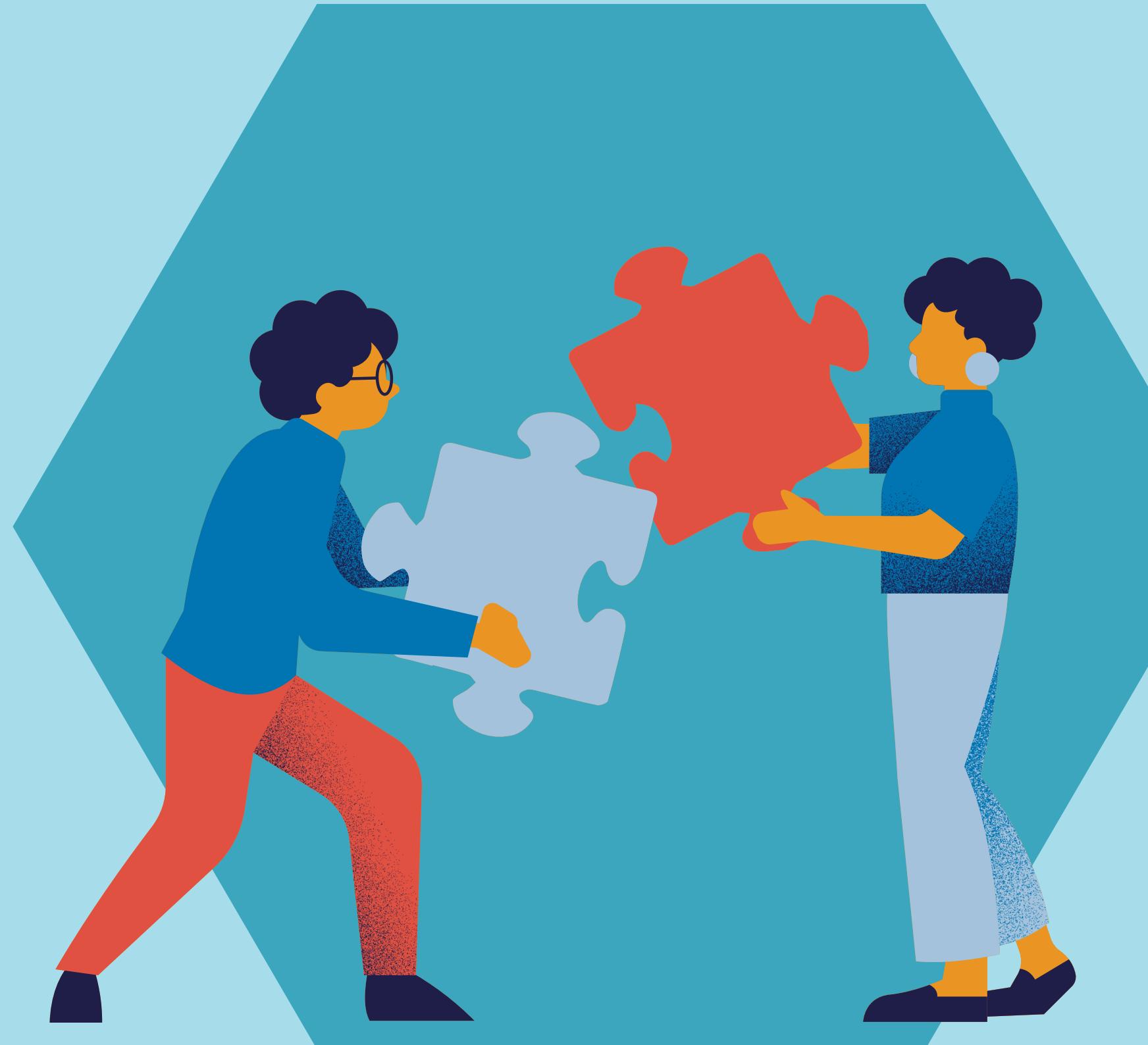
Oltre alla perplessità, ci sono altre metriche importanti:

- **Accuratezza:** Quante volte il modello dà la risposta giusta rispetto a quelle attese.
- **BLEU/ROUGE:** Misurano la somiglianza tra il testo generato e un testo di riferimento (utili per traduzioni o riassunti).
- **Efficienza computazionale:** Quanto tempo ed energia servono per addestrare ed eseguire il modello, fattore importante per la sostenibilità.
- **Equità e bias:** Controllare che il modello non riproduca stereotipi o discriminazioni è fondamentale per un utilizzo etico.



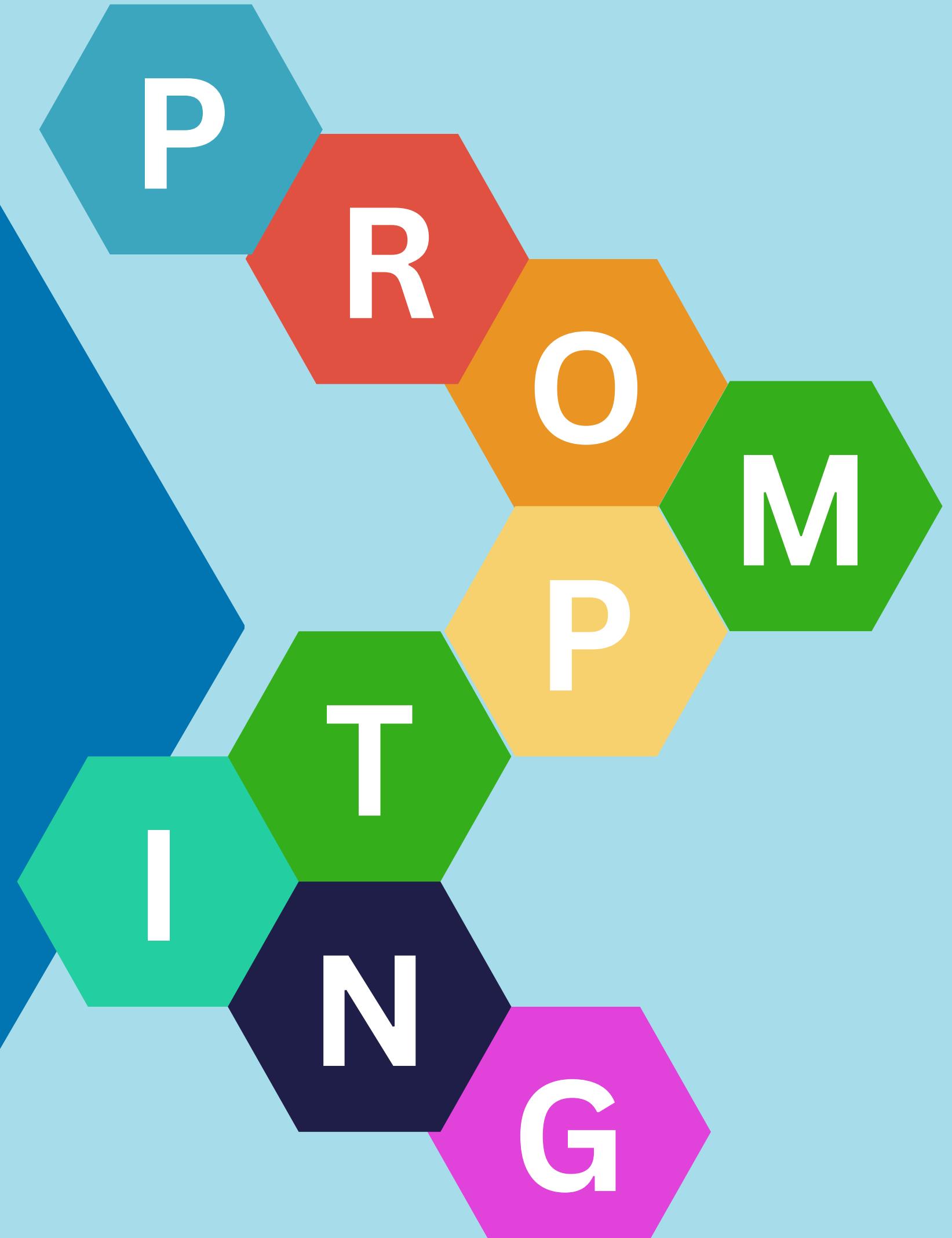
# Prompting come input per il modello

Un LLM (come ChatGPT) è addestrato su enormi quantità di testo. Per ottenere risposte utili, bisogna dare al modello un prompt chiaro. Più il prompt è preciso, più l'output sarà corretto e pertinente.



# Cos'è il Prompting ?

Il prompting è un metodo per comunicare con l'intelligenza artificiale. Vengono fornite delle istruzioni testuali (chiamate prompt) a un modello per ottenere risposte utili. Il modello legge il testo inserito e, passo dopo passo, genera una risposta basata su quello che ha imparato.



# Prompt Engineering e Template

Il Prompt Engineering è l'arte di scrivere prompt chiari, efficaci e precisi per ottenere il miglior risultato possibile. Spesso si usano template (schemi predefiniti) per semplificare il lavoro che aiutano a strutturare le domande in modo semplice.



# Prompt Engineering e Template - Esempio

---

- **Template:** "Spiega il concetto di [argomento] in modo semplice."
- **Input:** "cellula."
- **Prompt finale:** "Spiega il concetto di cellula in modo semplice."
- **Risposta:** "La cellula è l'unità fondamentale di tutti gli esseri viventi..."



# Few-Shot e Zero-Shot Prompting

Ci sono due modi per aiutare il modello a capire meglio cosa vogliamo:

- **Few-shot prompting:** si forniscono al modello alcuni esempi per aiutarlo a capire meglio il compito.
- **Zero-shot prompting:** il modello deve lavorare solo con l'istruzione data, senza esempi.



# Few-Shot e Zero-Shot Prompting – Esempio

Few-shot:

*"Traduci in inglese: 'Ciao' → 'Hello'  
'Buongiorno' → ?"*

Zero-shot:

*"Traduci in inglese la parola 'Buongiorno'."*



# Prompting come input per il modello

Gli LLM e il concetto di prompting possono essere concepiti come due parti che lavorano insieme:

- **Prompting** → Il modo in cui diamo istruzioni al modello.
- **LLM** → Il cervello che legge il prompt e genera la risposta.



# Prompting come input per il modello - Esempio

*Prompt:*

*“Spiega la fotosintesi in modo semplice.”*

*Risposta (LLM):*

*“La fotosintesi è il processo con cui le piante usano la luce solare per trasformare acqua e anidride carbonica in cibo.”*



# Allineamento dei Modelli

I modelli di linguaggio a volte danno risposte imprecise o non seguono bene le istruzioni. Per migliorare questo aspetto si usano due tecniche:

1. ***Instruction Tuning***: il modello viene addestrato su tante istruzioni con risposte corrette.
2. ***Preference Alignment***: si usa un secondo modello per valutare se le risposte sono allineate alle preferenze umane.



# Ottimizzazione Automatica dei Prompt

Esistono algoritmi che migliorano automaticamente i prompt.  
Funzionano nella medesima maniera:

- Partono da un prompt iniziale.
- Valutano l'efficacia del prompt.
- Generano varianti per vedere se migliorano le risposte.
- Usano l'intelligenza artificiale per suggerire come perfezionare il prompt.



# RLHF: Modelli più vicini alle esigenze umane

- *RLHF (Reinforcement Learning from Human Feedback)*: il modello impara dalle valutazioni umane delle sue risposte.

Entrambi i metodi rendono il modello più utile e sicuro.

