12. Large Language Models

Un modello linguistico è una distribuzione di probabilità definita su sequenze di token. Può essere utilizzato per generare frasi che, intuitivamente, desideriamo siano sintatticamente e semanticamente corrette. Per catturare il contesto delle parole, i modelli linguistici di grandi dimensioni (LLM, Large Language Models) condizionano la probabilità che un token compaia sulla base dei n token che lo precedono, seguendo un approccio simile a quello adottato dai modelli linguistici neurali e dai modelli N-gramma.

Le reti neurali ricorrenti (RNN, Recurrent Neural Networks) permettono che la distribuzione condizionata di un token dipenda dall'intero contesto precedente, ma risultano difficili da addestrare. I Transformer, invece, continuano a modellare il contesto fino ai precedenti n termini, ma sono più facili da addestrare grazie al parallelismo computazionale offerto dalle GPU. Un valore sufficientemente grande di n (finestra di contesto) può essere utilizzato per addestrare molte applicazioni pratiche, come ad esempio nel caso di GPT-3 dove $n = 2048$.

Transformer

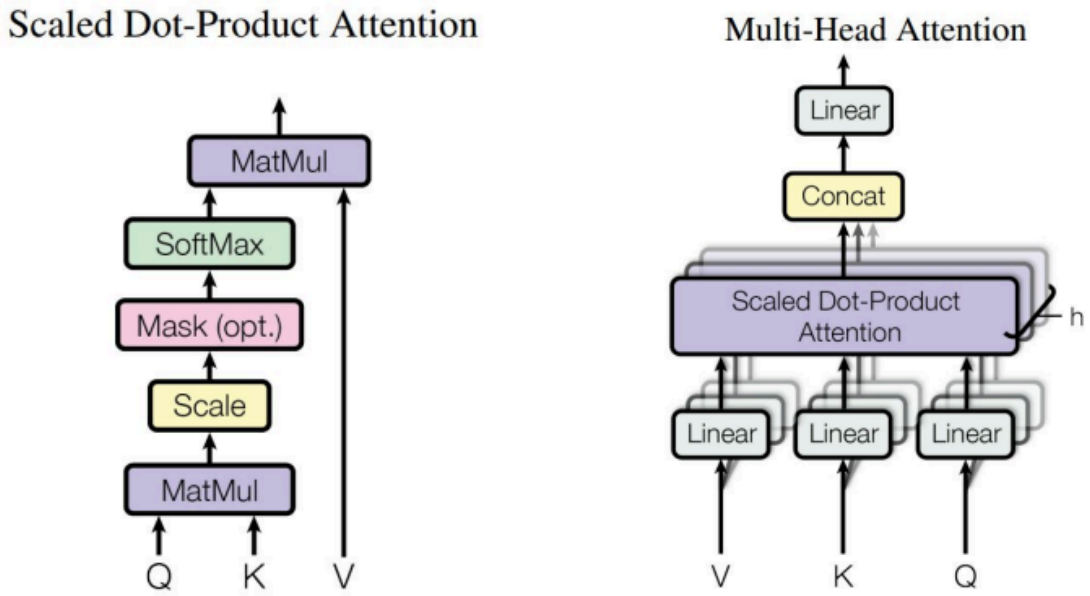
Il Transformer è il primo modello di trasduzione di sequenze basato interamente sull'attenzione, sostituendo gli strati ricorrenti che erano comunemente utilizzati nelle architetture encoder-decoder con meccanismi di self-attention multi-testa. L'attenzione è definita come un meccanismo in cui a ogni token nella finestra di contesto viene assegnato un peso a ogni istanza di esecuzione. Questo consente di catturare meglio la rilevanza di token distanti all'interno di una frase, evitando di fare affidamento solo su token vicini, come invece accade nelle RNN.

Multi-headed attention

I Transformer utilizzano questo tipo più avanzato di attenzione, che impiega matrici di Query, Key e Value invece di vettori. La matrice di output viene calcolata con la seguente formula:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

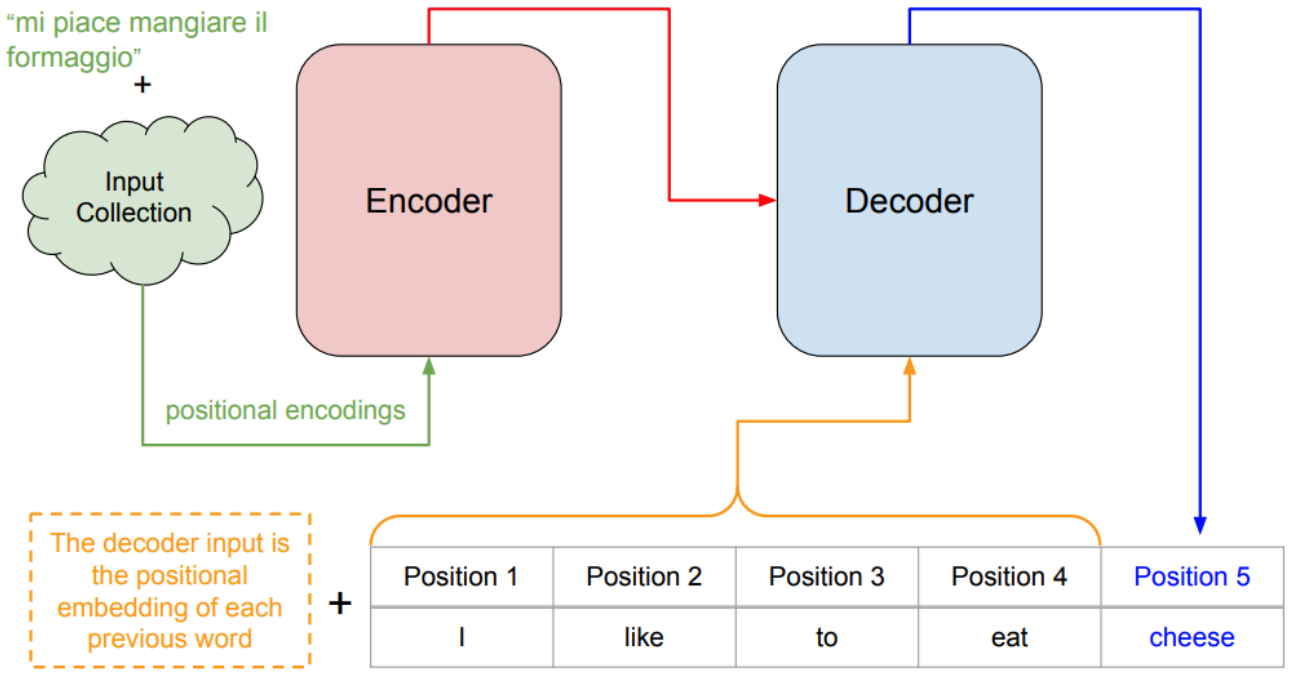
Nel caso in cui d_k (la dimensione del vettore Key) sia grande, la funzione softmax può produrre gradienti molto piccoli. Questo effetto indesiderato viene contrastato dividendo il prodotto scalare per $\sqrt{d_k}$.



Un transformer in azione

Il compito che si desidera svolgere è la traduzione di una frase dall'italiano all'inglese, dove si fornisce in input una frase in italiano ("mi piace mangiare il formaggio") e il Transformer ne produce la traduzione in inglese. Il blocco encoder utilizza l'attenzione multi-testa per trasformare la sequenza di token in vettori che verranno poi utilizzati dal decoder per generare i token successivi della frase tradotta.

Il decoder prende come input le uscite precedenti (in questo esempio manca soltanto l'ultima parola) e i vettori generati dall'encoder, eseguendo diversi passaggi di attenzione multi-testa per restituire la parola successiva più probabile. Ogni parola in output può essere vista come una combinazione lineare di tutte le parole presenti nella frase, quindi il decoder seleziona la parola successiva più probabile — in questo caso "cheese" — attraverso molteplici moltiplicazioni di matrici.



LLM per il recupero dell'informazione (IR)

I LLM stanno rivoluzionando il recupero dell'informazione grazie alla loro capacità di comprendere il linguaggio naturale in modo profondo. Secondo Zhu et al., possono essere distinti cinque ruoli principali nei quali i LLM contribuiscono a migliorare le prestazioni dei sistemi IR. Questi ruoli sono rappresentati come componenti interconnessi attorno al LLM

Query Rewriting

Il modulo di riscrittura della query ha il compito di riformulare automaticamente le richieste dell'utente, migliorandone la chiarezza, la specificità o l'allineamento con il linguaggio del corpus informativo. Questo processo è particolarmente utile quando l'input iniziale è vago o ambiguo. Il LLM viene utilizzato per generare versioni alternative della query che risultino più efficaci nel processo di ricerca.

Retriever

Il retriever ha la funzione di recuperare rapidamente un insieme iniziale di documenti potenzialmente rilevanti, solitamente da un indice invertito o da una base di conoscenza di grandi dimensioni. I LLM possono essere impiegati per potenziare il retriever semantico, producendo embedding (rappresentazioni vettoriali) della query e dei documenti e calcolando la similarità tra questi vettori, anziché basarsi solo su corrispondenze lessicali.

Reader

Il reader prende in input i documenti recuperati e analizza i contenuti in profondità per estrarre risposte puntuali o riassumere le informazioni rilevanti. Questo modulo sfrutta la comprensione contestuale dei LLM per identificare esattamente dove e come viene trattato l'argomento di interesse all'interno dei testi. Il reader è particolarmente efficace in compiti di question answering.

Reranker

Il reranker si occupa di riorganizzare l'elenco dei documenti recuperati in base alla loro rilevanza rispetto alla query, affinando l'output del retriever. I LLM vengono usati per valutare la pertinenza semantica e contestuale di ciascun documento rispetto alla richiesta iniziale. Questo processo migliora significativamente la precisione dei risultati finali presentati all'utente.

Search Agent

L'agente di ricerca è un componente che funge da coordinatore del processo complessivo. Utilizza le capacità dei LLM per orchestrare le interazioni tra i moduli precedenti. Può decidere, ad esempio, quando è necessario riformulare una query, quali risultati passare al

reader o se interrogare nuovamente il sistema con un prompt modificato. In alcuni casi, l'agente può anche condurre ricerche iterative per convergere su una risposta ottimale.

LLM per il riassunto di testi

Il riassunto di testi è un problema classico del Natural Language Processing (NLP), in cui, a partire da uno o più documenti, si cerca di produrre una versione condensata delle informazioni contenute nell'input. I LLM sono stati recentemente utilizzati in questo campo grazie alla loro capacità di comprendere in profondità i testi e di generare nuovi contenuti.

Il riassunto può essere suddiviso in due categorie principali: il **riassunto astrattivo**, in cui vengono create nuove frasi che non erano presenti nei documenti originali, e il **riassunto estrattivo**, che seleziona e utilizza frasi direttamente dal testo di partenza.

Stato dell'arte: i LLM sono davvero efficaci?

In generale, come dimostrato da Pu et al., i riassunti generati dai LLM (principalmente GPT-3.5 e GPT-4) superano costantemente sia quelli prodotti da modelli specializzati sia quelli scritti da esseri umani, in compiti come il riassunto di singole notizie, di più fonti giornalistiche e di dialoghi. Tuttavia, secondo Shen et al., questi stessi modelli non sono ancora in grado di **valutare** i riassunti in modo comparabile agli esseri umani, in particolare nei casi di riassunto estrattivo, dove la precisione semantica e la fedeltà al testo originale sono più critiche.

Esempi utili per il nostro contesto

Liu et al. hanno sviluppato un sistema per generare articoli di Wikipedia a partire da più documenti sorgente, utilizzando un metodo combinato tra riassunto estrattivo e astrattivo, implementato con diversi modelli basati su Transformer. Uno dei problemi principali nel riassumere contenuti per domini diversi (come cardiologia rispetto a dermatologia) è la necessità di addestrare modelli differenti per ogni contesto. A questo proposito, Hu et al. propongono **LoRA (Low-Rank Adaptation)**, una tecnica che consente di derivare modelli adattivi a partire da un modello comune, modificando solo i pesi di adattamento. Questo approccio permette di mantenere o migliorare le prestazioni, riducendo al contempo i costi di addestramento.