Anatomia degli LLMs

2. Introduzione ai Transformers

Alessio Miaschi
ItaliaNLP Lab, Istituto di Linguistica Computazionale (CNR-ILC), Pisa
alessio.miaschi@ilc.cnr.it
https://alemiaschi.github.io/
http://www.italianlp.it/alessio-miaschi/

Introduzione ai Transformers

Transformer

- L’architettura ad oggi più utilizzata per lo sviluppo di NLM contestuali è quella del **Transformer**, introdotto per la prima volta qui: Attention is All you Need (Vaswani et al., 2017)

Transformer

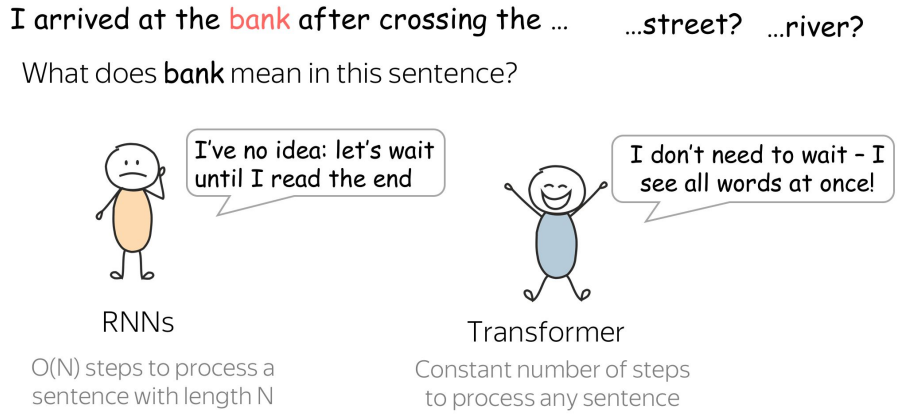
- L’architettura ad oggi più utilizzata per lo sviluppo di NLM contestuali è quella del **Transformer**, introdotto per la prima volta qui: Attention is All you Need (Vaswani et al., 2017)

- Il Transformer è una rete neurale (Encoder-Decoder) che sfrutta uno specifico meccanismo, l’***Attention***, per concentrarsi solo su alcune porzioni fondamentali di una frase e creare rappresentazioni contestuali delle parole

Transformer

- L’architettura ad oggi più utilizzata per lo sviluppo di NLM contestuali è quella del **Transformer**, introdotto per la prima volta qui: [Attention is All you Need (Vaswani et al., 2017)]

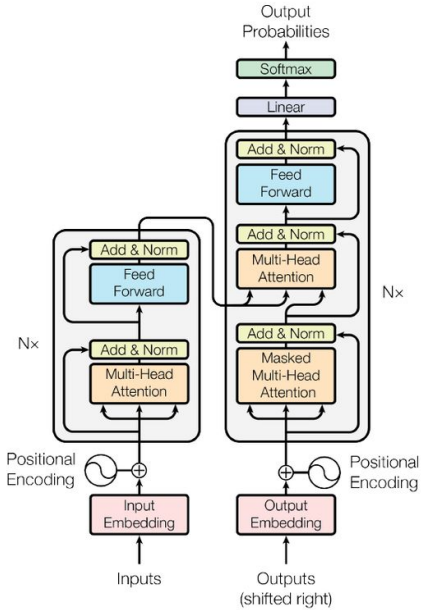
- Il Transformer è una rete neurale (Encoder-Decoder) che sfrutta uno specifico meccanismo, l’***Attention***, per concentrarsi solo su alcune porzioni fondamentali di una frase e creare rappresentazioni contestuali delle parole



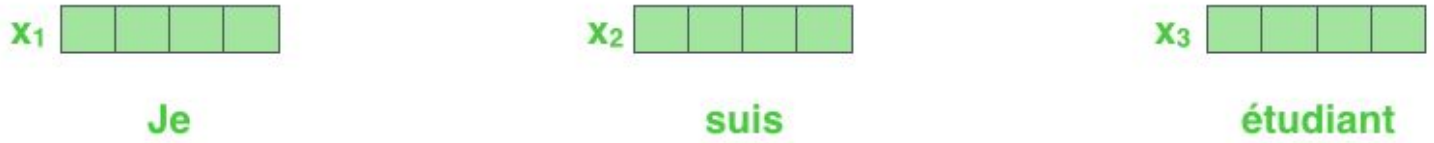
Transformer

- L’architettura ad oggi più utilizzata per lo sviluppo di NLM contestuali è quella del **Transformer**, introdotto per la prima volta qui: [Attention is All you Need (Vaswani et al., 2017)]

- Il Transformer è una rete neurale (Encoder-Decoder) che sfrutta uno specifico meccanismo, l’***Attention***, per concentrarsi solo su alcune porzioni fondamentali di una frase e creare rappresentazioni contestuali delle parole

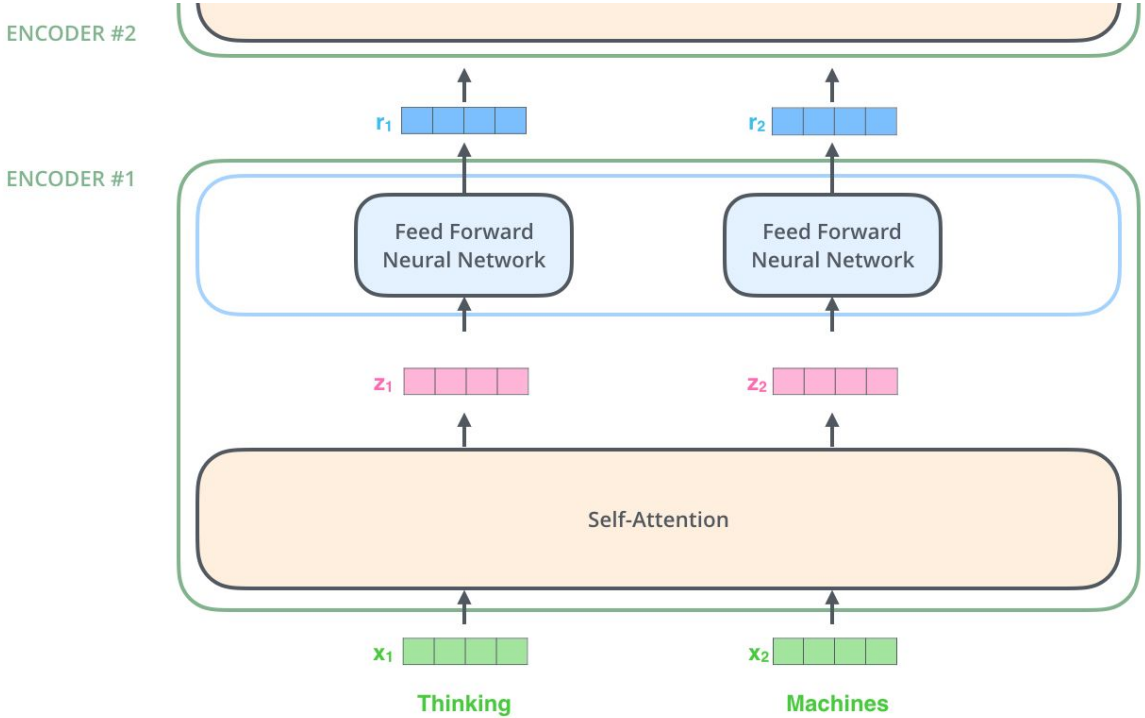


Transformer - Attention



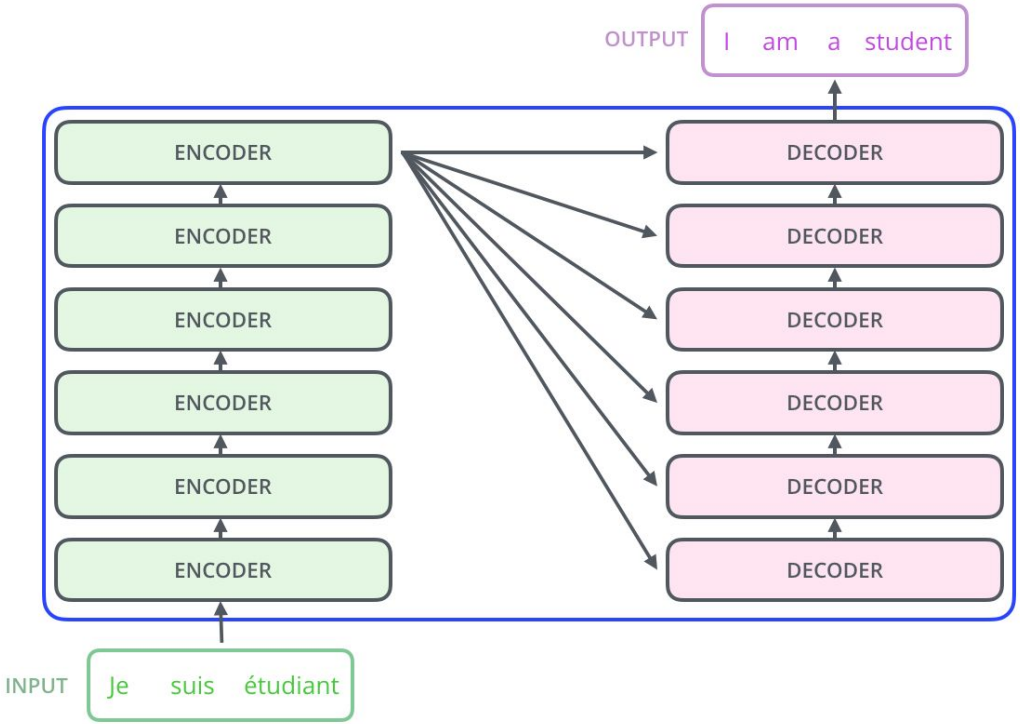
Da: [The Illustrated Transformer (Jay Alammar)]

Transformer - Attention



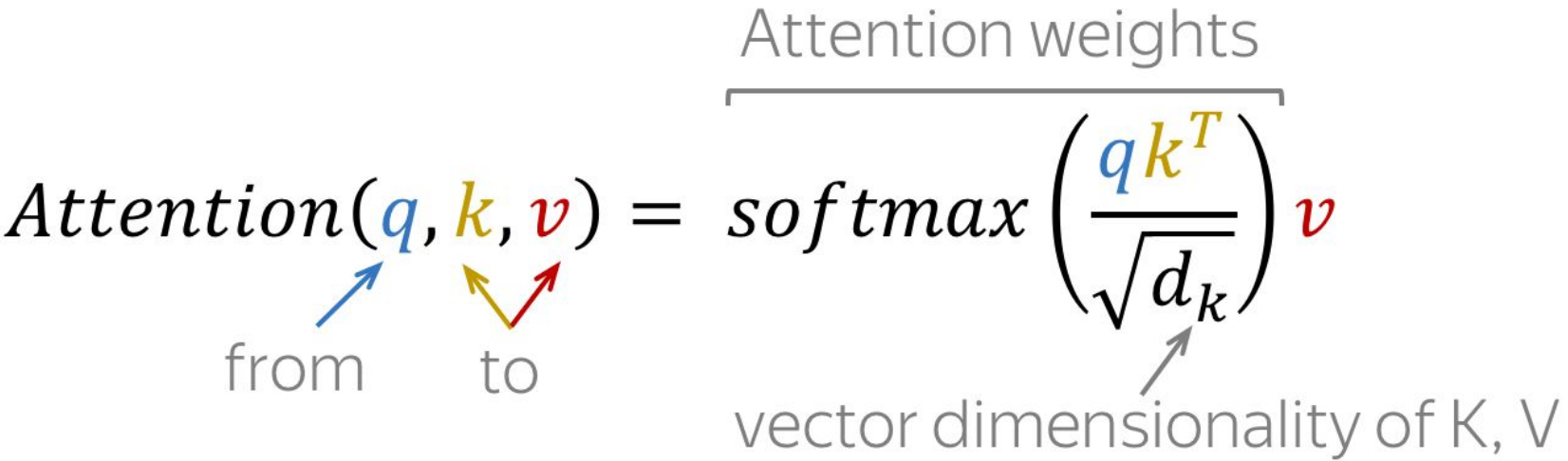
Da: [The Illustrated Transformer (Jay Alammar)]

Transformer - Attention



Da: [The Illustrated Transformer (Jay Alammar)]

Transformer - Attention



Da: Sequence to Sequence (seq2seq) and Attention (NLP Course, Lena Voita)

Transformer - Attention

Each vector receives three representations (“roles”)

W_Q × = Query: vector **from** which the attention is looking

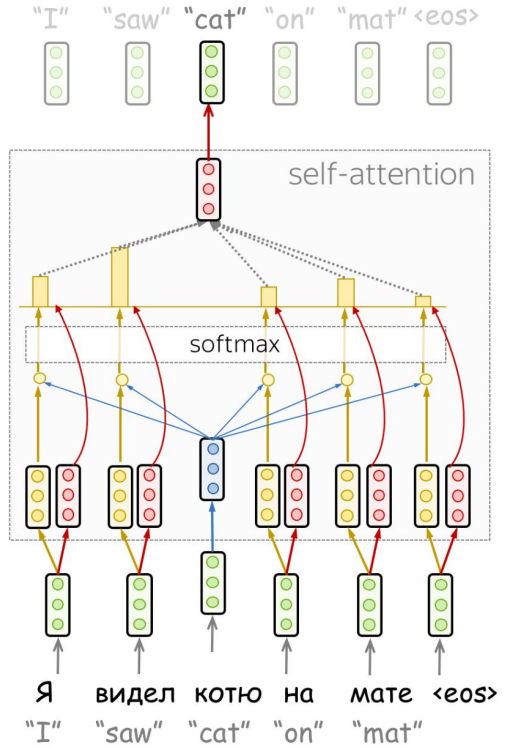
“Hey there, do you have this information?”

W_K × = Key: vector **at** which the query looks to compute weights

“Hi, I have this information – give me a large weight!”

W_V × = Value: their weighted sum is attention output

“Here’s the information I have!”



Da: [Sequence to Sequence (seq2seq) and Attention (NLP Course, Lena Voita)]

Transformer - Attention

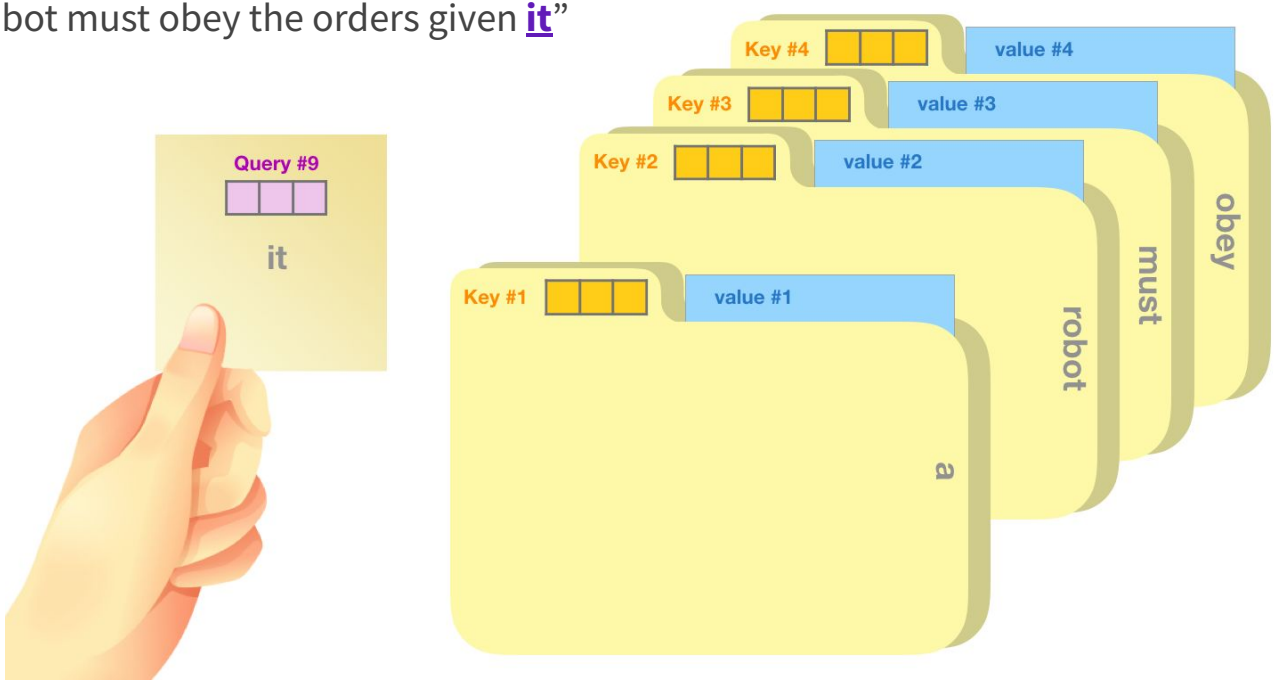
Input: “A robot must obey the orders given it”

Transformer - Attention

Input: “A robot must obey the orders given **it**”

Transformer - Attention

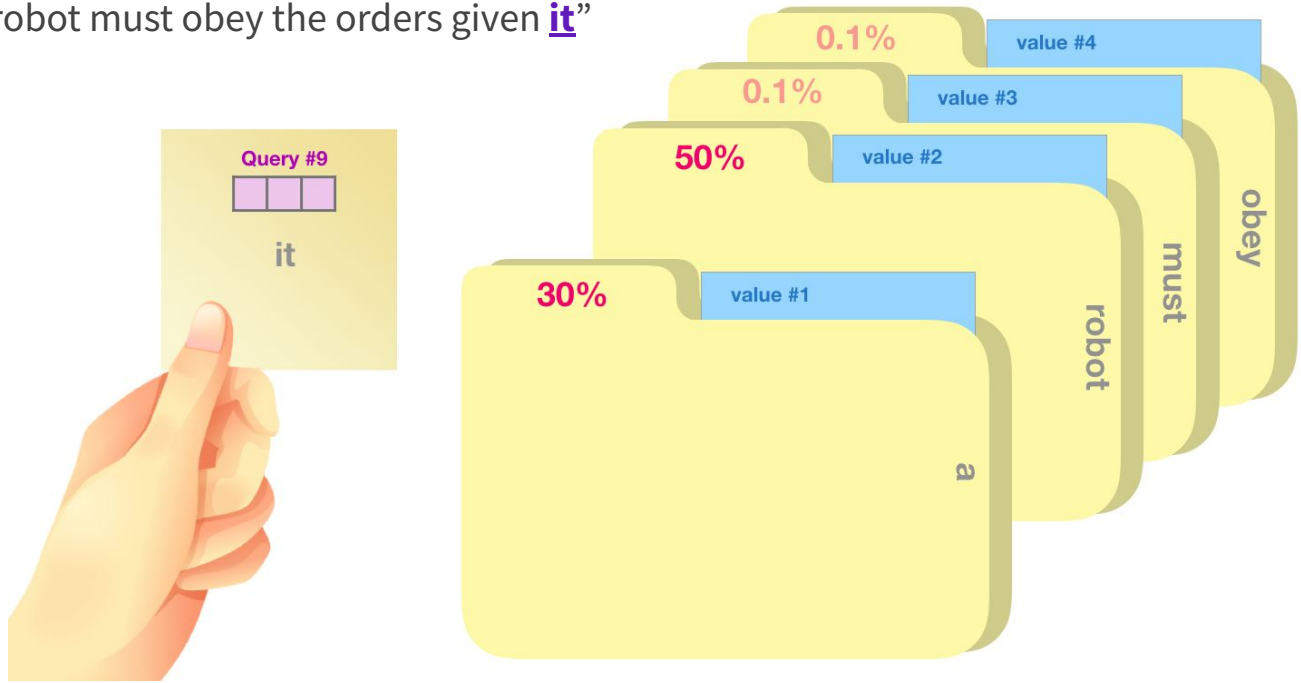
Input: “A robot must obey the orders given **it**”



Da: The Illustrated GPT-2 (Jay Alammar)

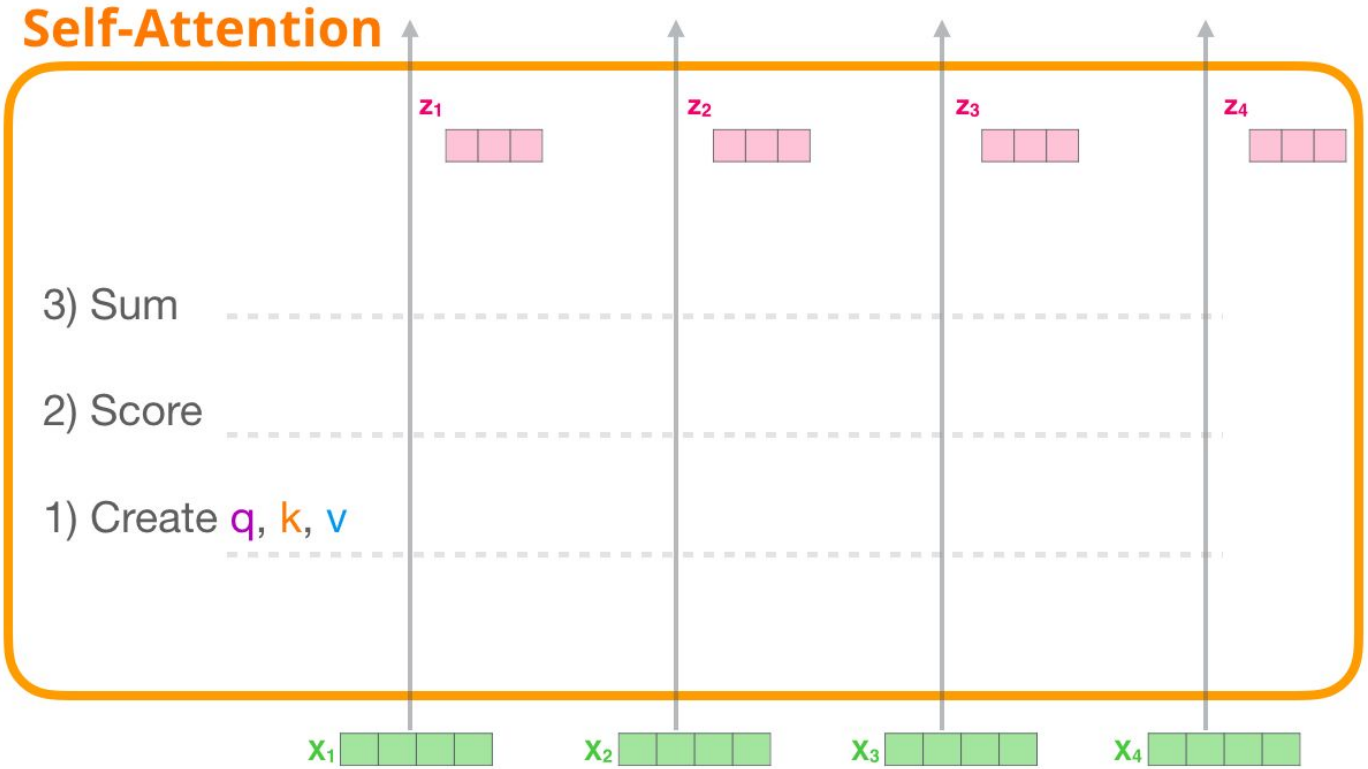
Transformer - Attention

Input: “A robot must obey the orders given **it**”



Da: The Illustrated GPT-2 (Jay Alammar)

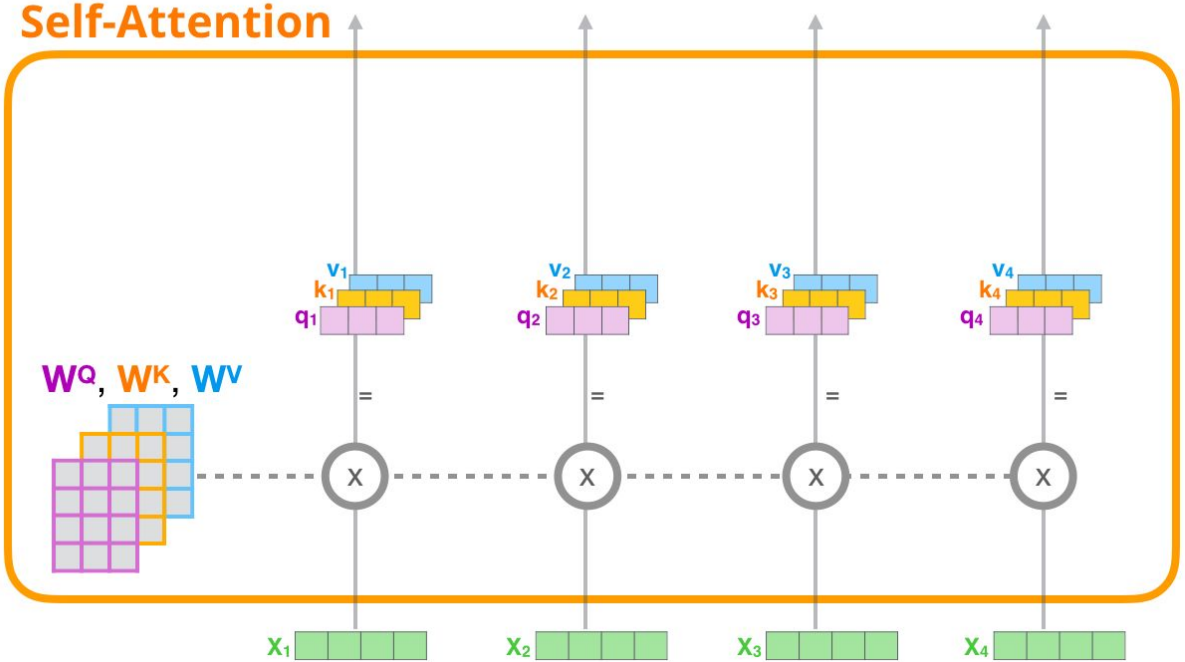
Transformer - Attention



Da: [The Illustrated GPT-2 (Jay Alammar)]

Transformer - Attention

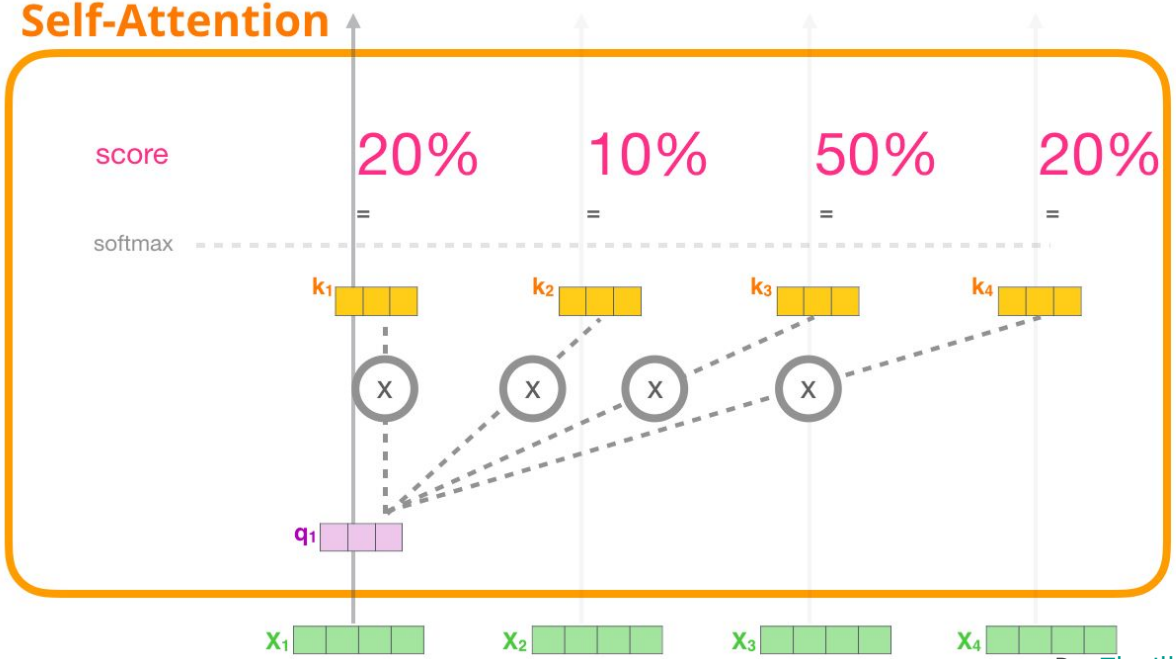
1) For each input token, create a query vector, a key vector, and a value vector by multiplying by weight Matrices W^Q, W^K, W^V



Da: [The Illustrated GPT-2 (Jay Alammar)]

Transformer - Attention

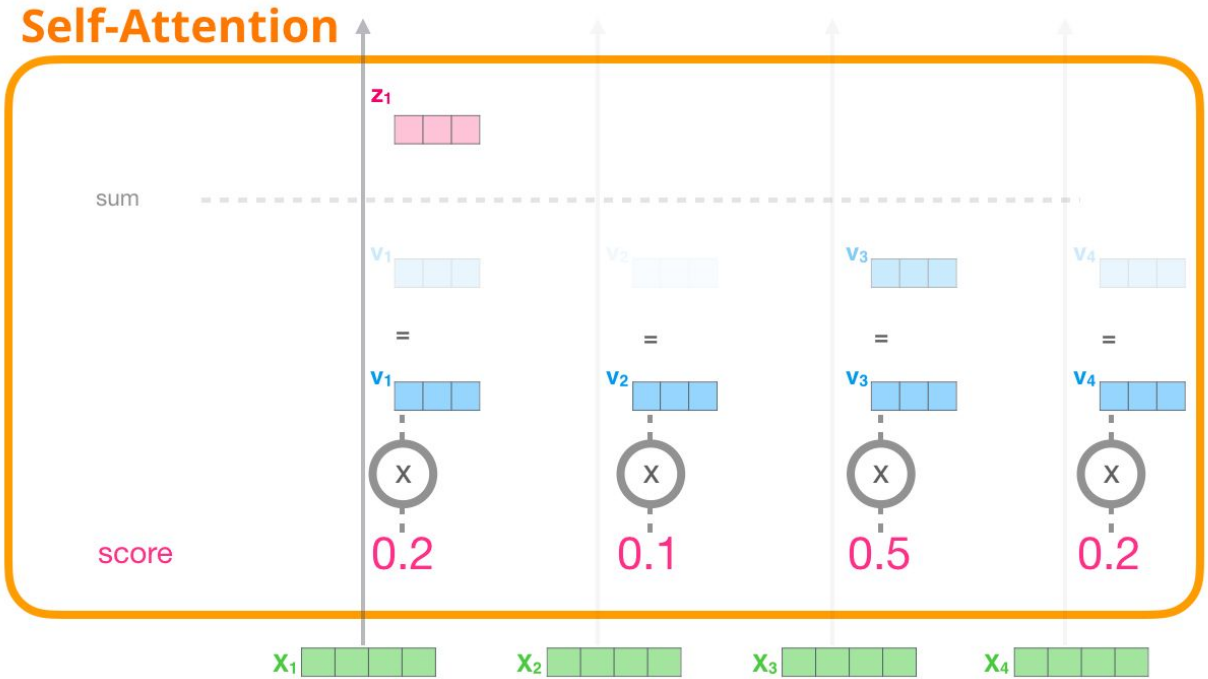
2) Multiply (dot product) the current query vector, by all the key vectors, to get a score of how well they match



Da: [The Illustrated GPT-2 (Jay Alammar)]

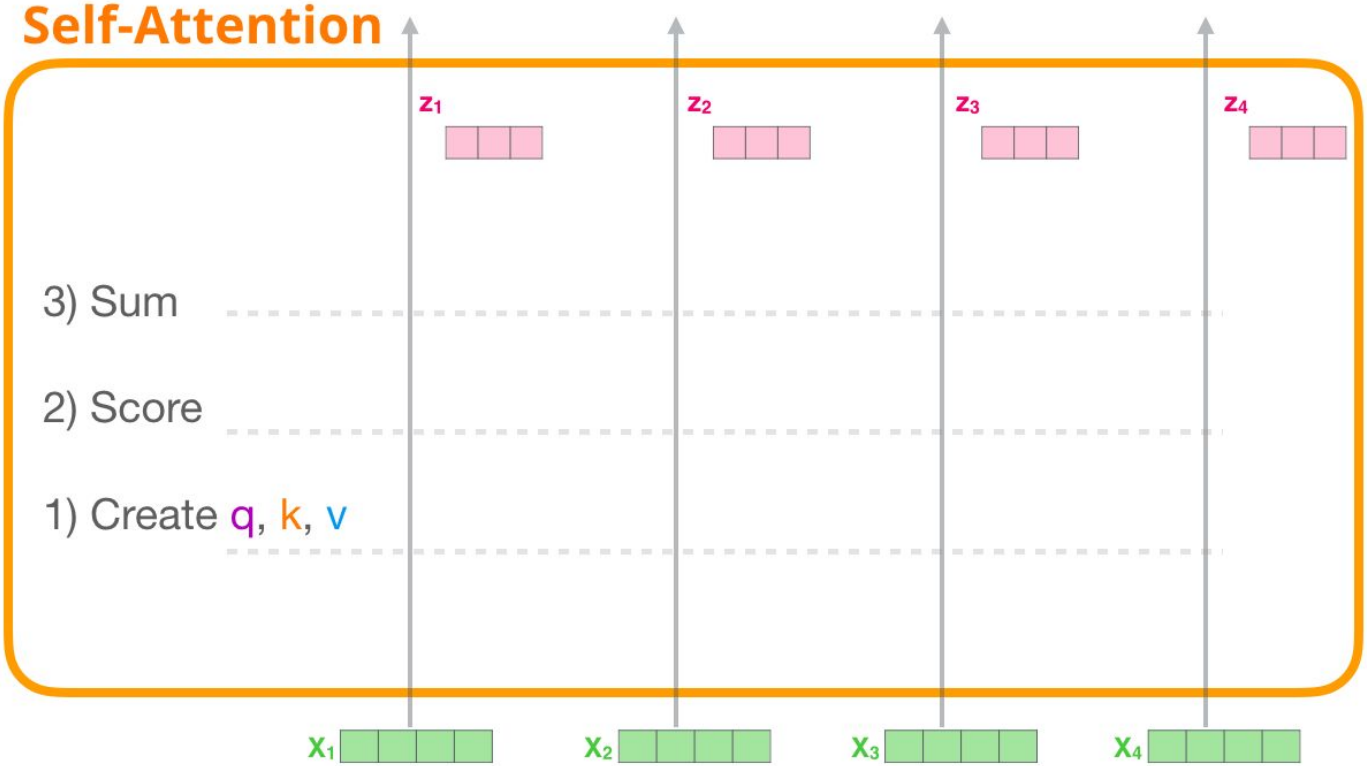
Transformer - Attention

3) Multiply the value vectors by the scores, then sum up



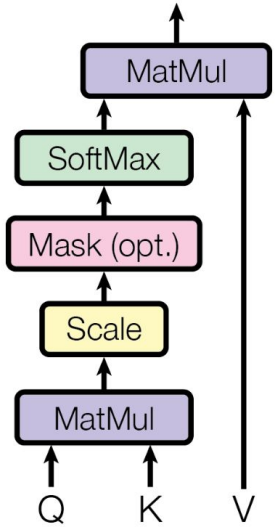
Da: [The Illustrated GPT-2 (Jay Alammar)]

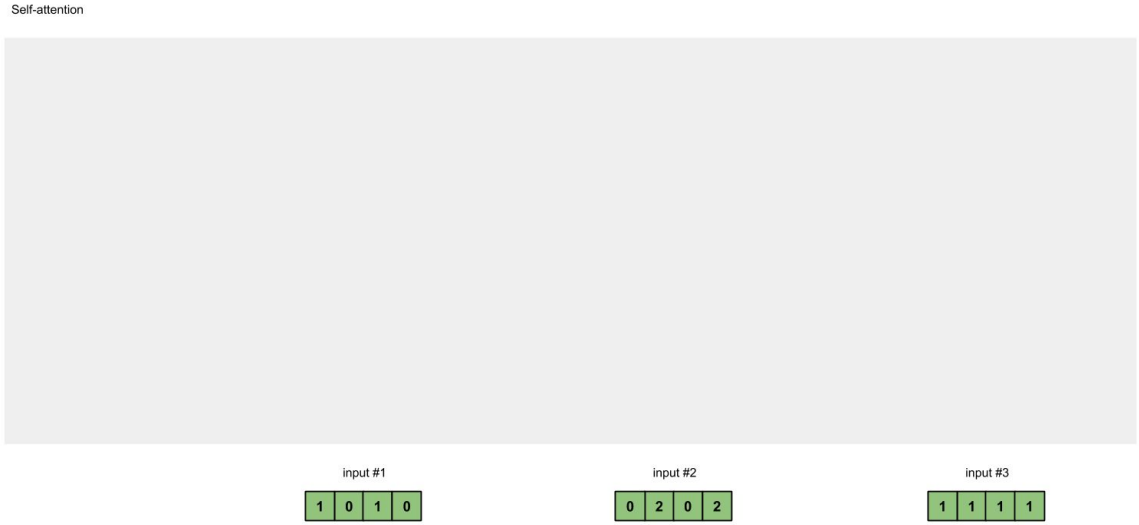
Transformer - Attention



Da: [The Illustrated GPT-2 (Jay Alammar)]

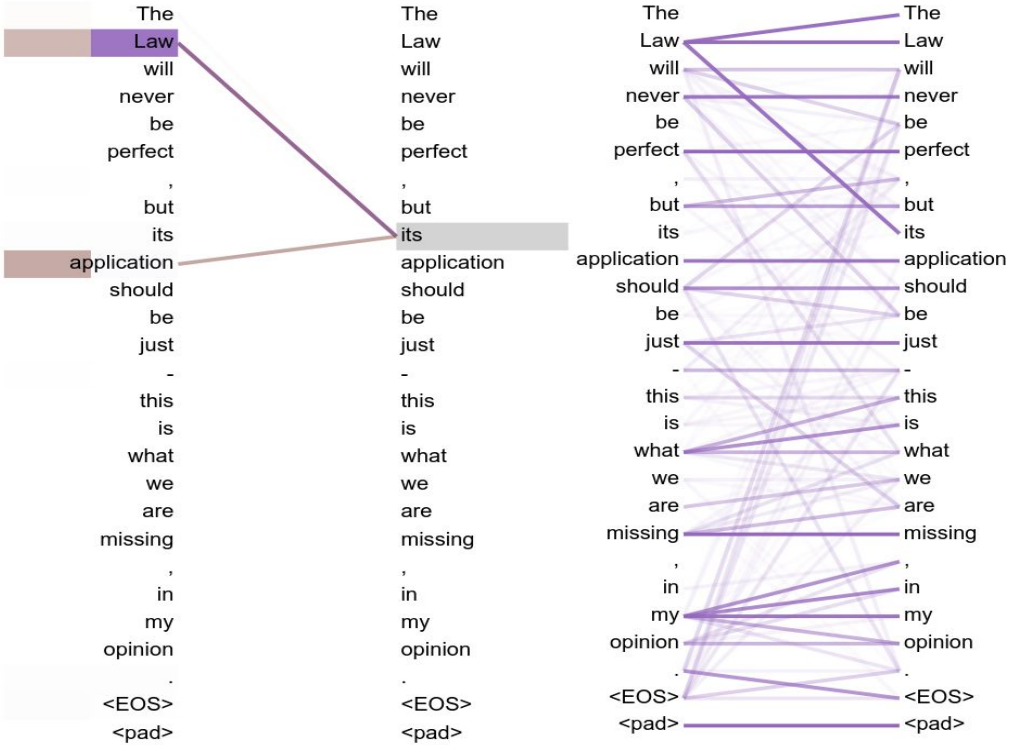
Transformer - Attention





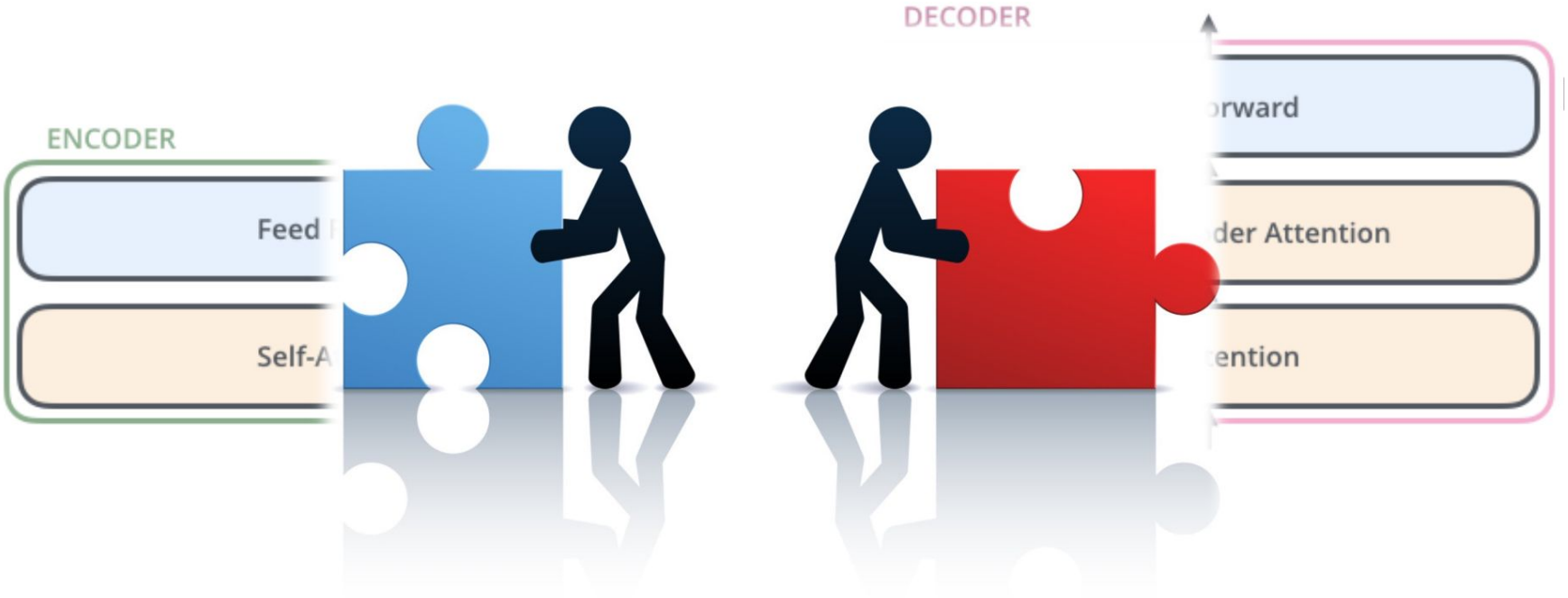
Da: [Illustrated: Self-Attention]

Transformer - Attention



Transformer-based NLMs

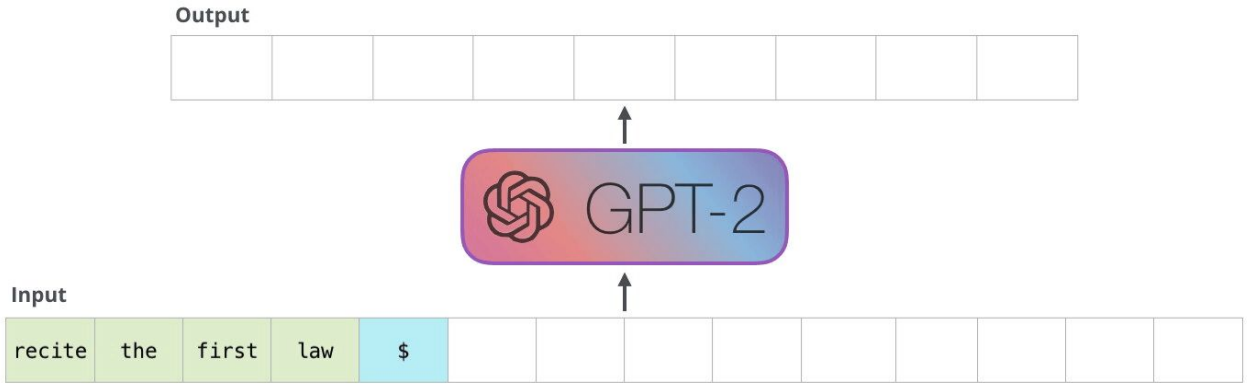
Transformer-based NLMs



Fonte: Large Language Models and How To Instruct Them (CLiC-it 2023 Tutorial)

GPT (Radford et al, 2018), GPT-2 (Radford et al, 2019), etc

- Decoder Transformer model
- Addestrato per approssimare la funzione di **Language Modeling** (**LM**)
- Modello generativo → pensato per generare testo



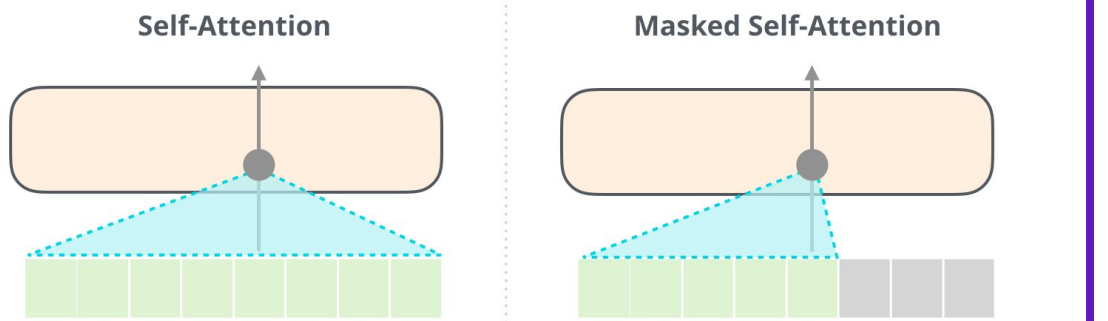
Improving Language Understanding by Generative Pre-Training (Radford et al,, 2018), https://openai.com/research/language-unsupervised

Language Models are Unsupervised Multitask Learners (Radford et al., 2019), https://openai.com/research/better-language-models

GPT (Radford et al, 2018), GPT-2 (Radford et al, 2019), etc

- Decoder Transformer model
- Addestrat
- Modello g

Masked Self-Attention!



Inp

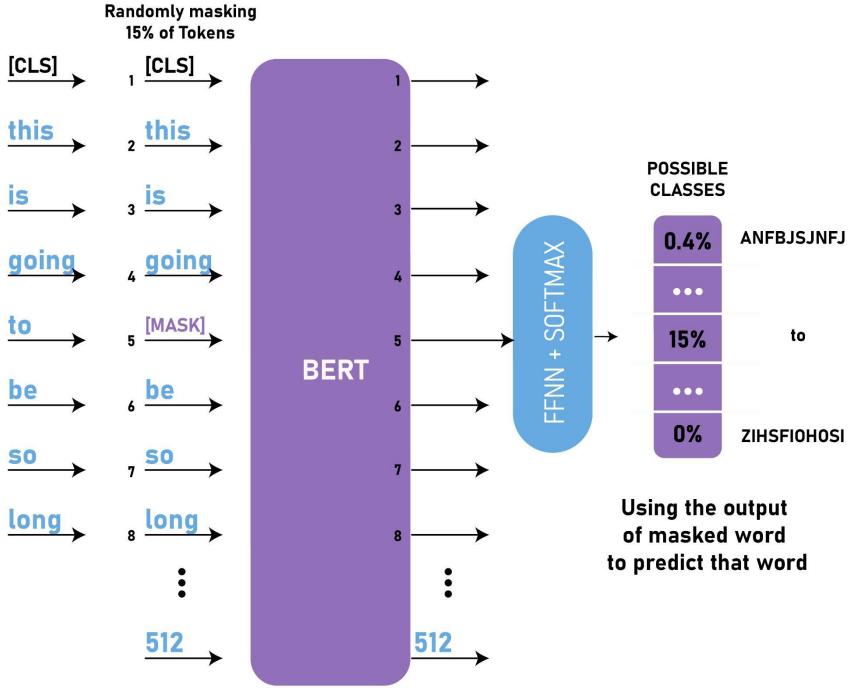
re

Improving Language Understanding by Generative Pre-Training (Radford et al,, 2018), https://openai.com/research/language-unsupervised

Language Models are Unsupervised Multitask Learners (Radford et al., 2019), https://openai.com/research/better-language-models

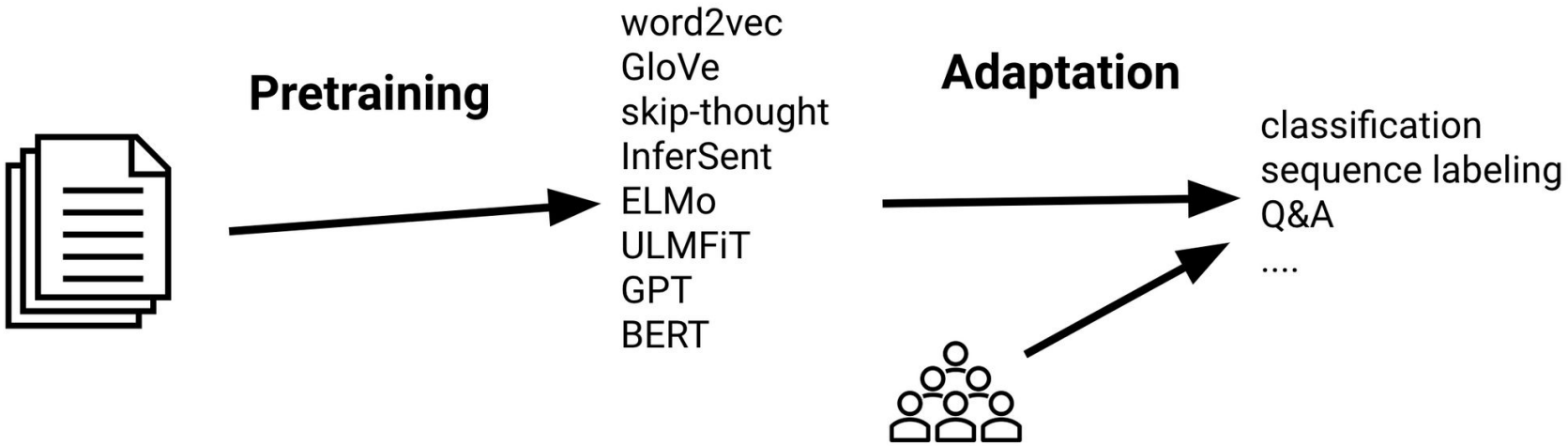
BERT (Devlin et al., 2019) 

- Encoder Transformer model (12/24 layers)
- Addestrato per approssimare la funzione di **Masked Language Modeling** (**MLM**)
- Il modello può poi essere ri-addestrato (fine-tuning) per risolvere svariati task di NLP:
  - Sentiment analysis;
  - Question answering;
  - Textual entailment;
  - etc.



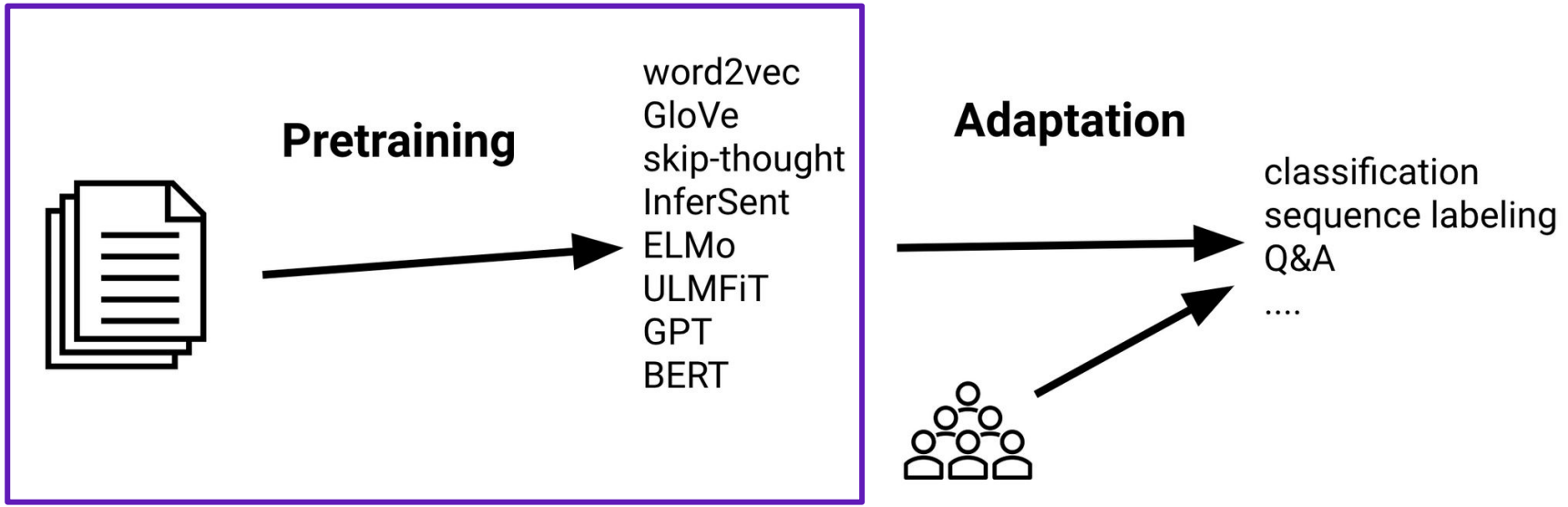
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et al., 2019), https://aclanthology.org/N19-1423.pdf

Transfer Learning



The State of Transfer Learning in NLP: https://ruder.io/state-of-transfer-learning-in-nlp/

Transfer Learning



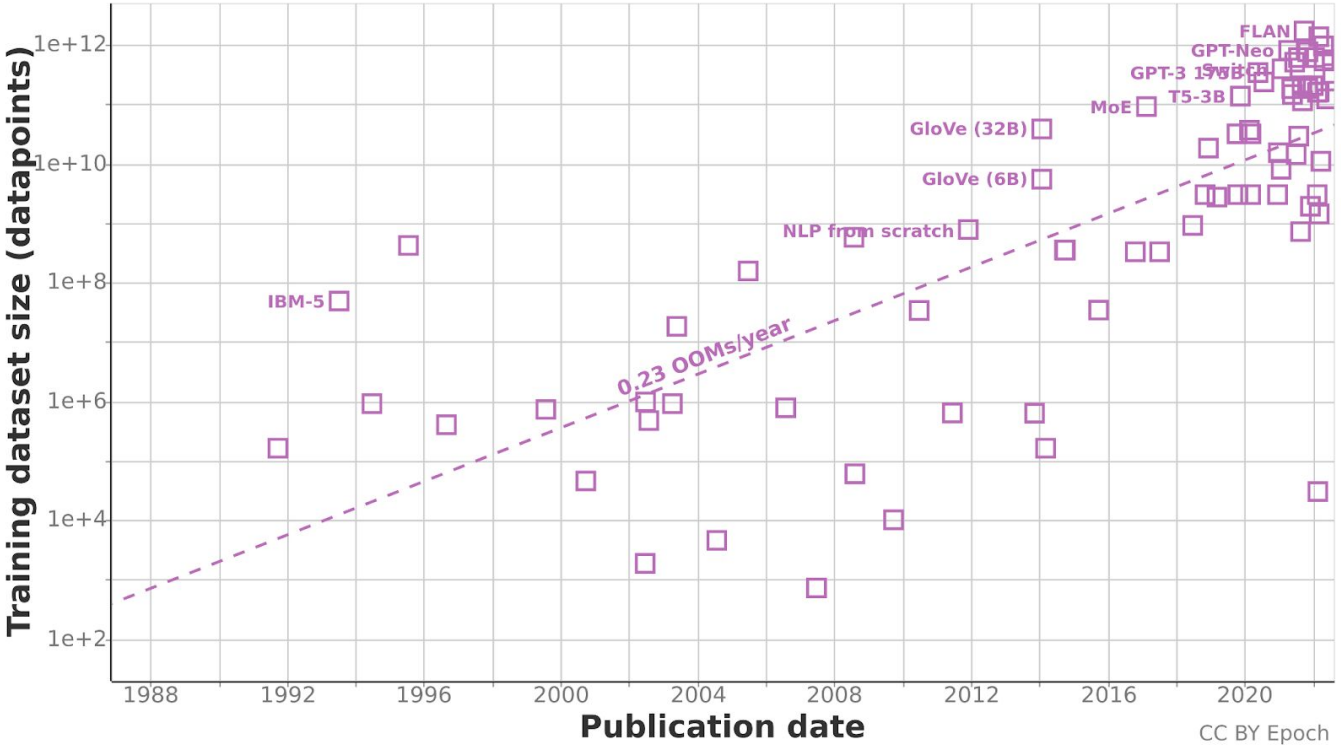
The State of Transfer Learning in NLP: https://ruder.io/state-of-transfer-learning-in-nlp/

Pre-training

- Durante la fase di “*Pre-training*”, il modello viene addestrato in maniera unsupervised (e.g. LM, MLM) su una grande quantità di dati grezzi

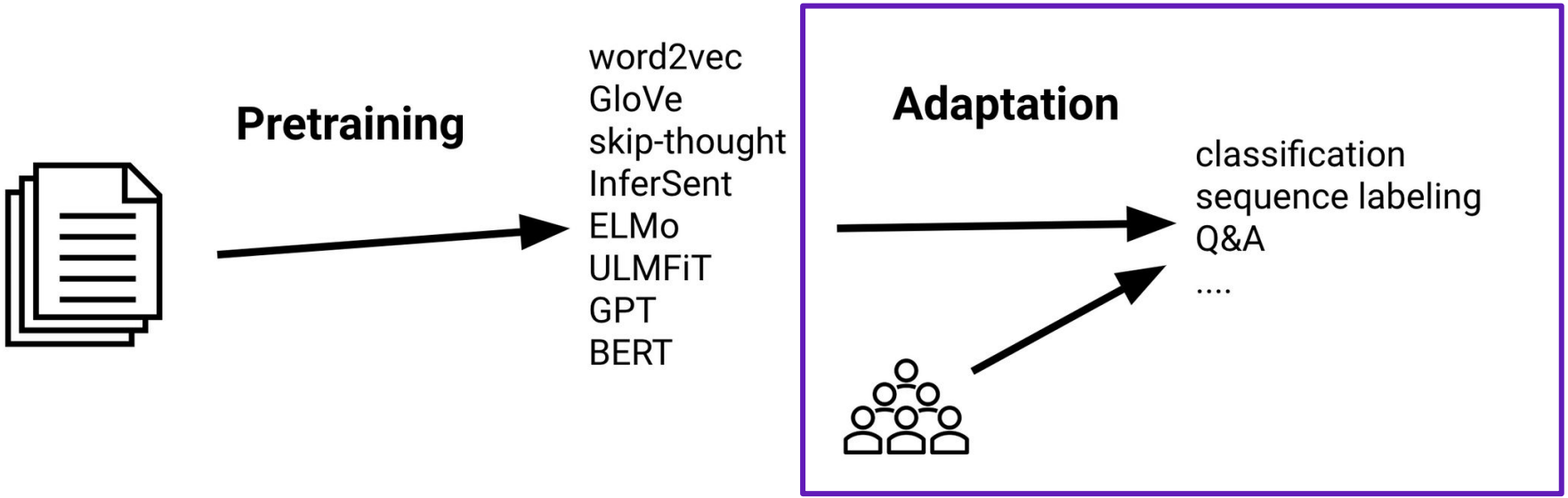
- Alcuni esempi:
  - **Training di BERT**: BookCorpus (800M di parole) e Wikipedia Inglese (2500M di parole)
  - **Training di GPT-3**: CommonCrawl + WebText2 + Books1 + Books2 + Wikipedia (circa 500B di parole)

Pre-training



Fonte: https://www.lesswrong.com/posts/asqDCb9XzXnLjSfgL/trends-in-training-dataset-sizes

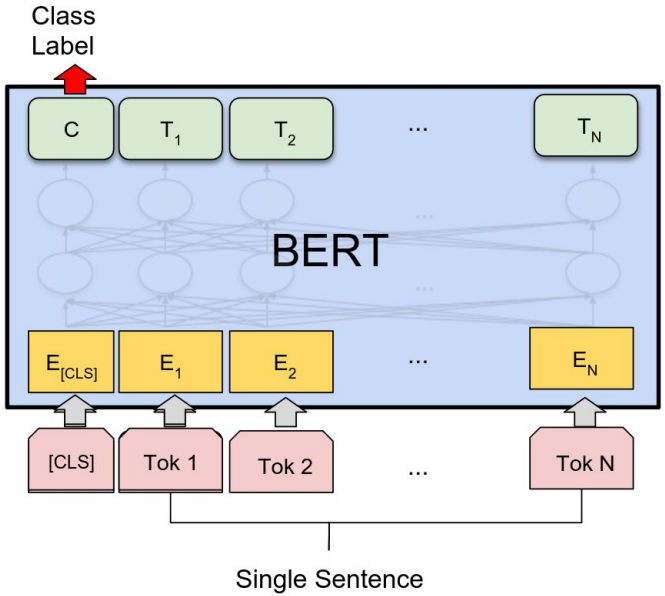
Transfer Learning



The State of Transfer Learning in NLP: https://ruder.io/state-of-transfer-learning-in-nlp/

Fine-tuning (Adaptation)

- Durante la fase di “*Fine-tuning*”, si va a specializzare il modello su un determinato task
  - In altri termini, si prende il modello pre-trainato e si continua l’addestramento su un nuovo dataset e modificando la sua funzione obiettivo (e.g. sentiment analysis, textual entailment, sentence complexity, etc.)



(b) Single Sentence Classification Tasks: SST-2, CoLA

Prompting → Large Language Models (LLMs)

- Negli ultimi anni, lo sviluppo dei NLMs si è spostato verso la creazione di modelli generativi:
  - Scopo principale: considerarsi qualsiasi task (e.g. classificazione, translation, question answering, etc) come task di **generazione**

Prompting → Large Language Models (LLMs)

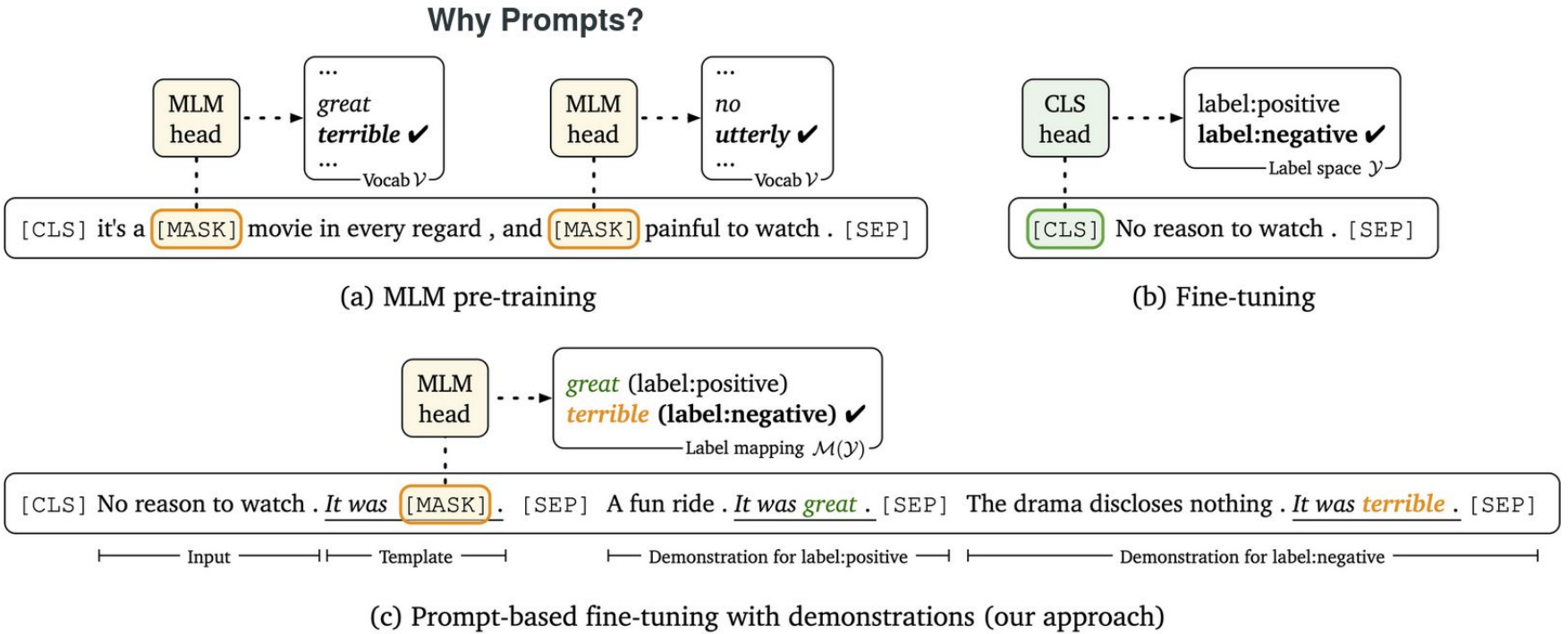
- Negli ultimi anni, lo sviluppo dei NLMs si è spostato verso la creazione di modelli generativi:
  - Scopo principale: considerarsi qualsiasi task (e.g. classificazione, translation, question answering, etc) come task di **generazione**

Prompting

"A prompt is a piece of text inserted in the input examples, so that the original task can be formulated as a (masked) language modeling problem."

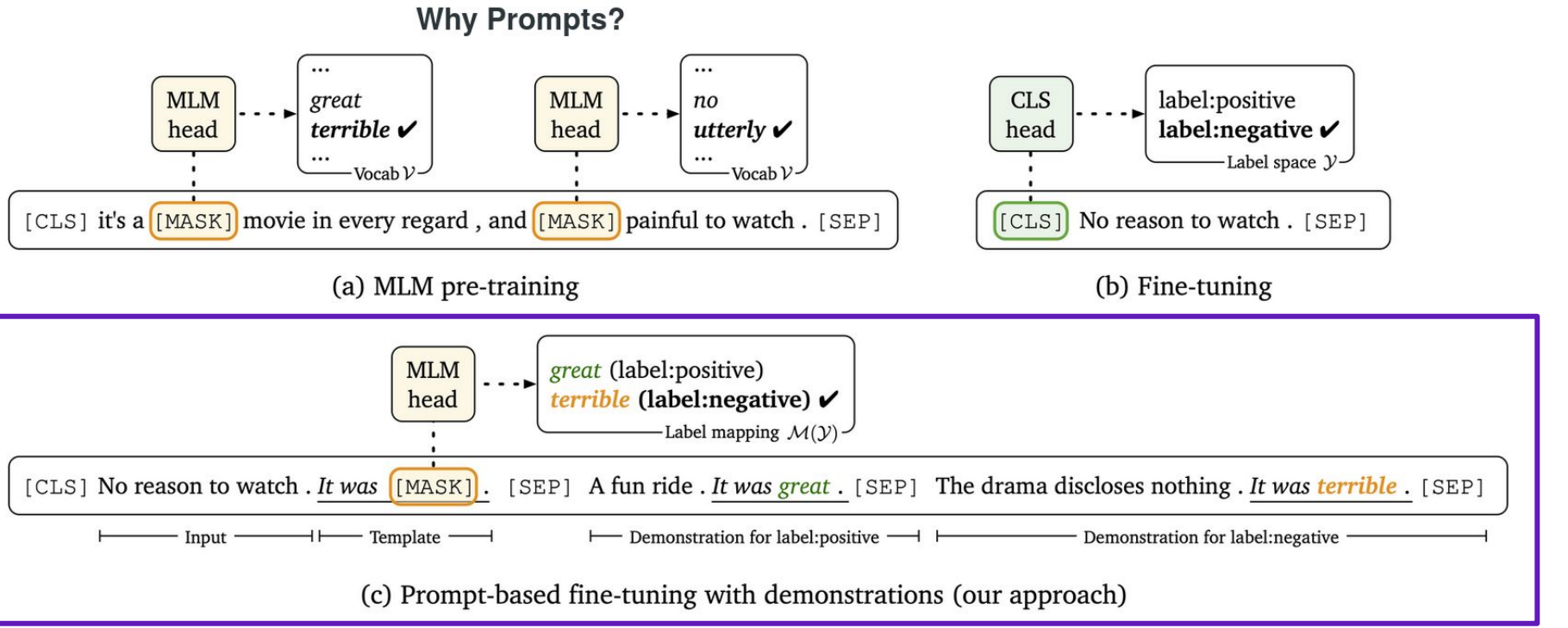
(Prompting: Better Ways of Using Language Models for NLP Tasks, The Gradient)

Prompting → Large Language Models (LLMs)



Fonte: [Prompting: Better Ways of Using Language Models for NLP Tasks, The Gradient]

Prompting → Large Language Models (LLMs)



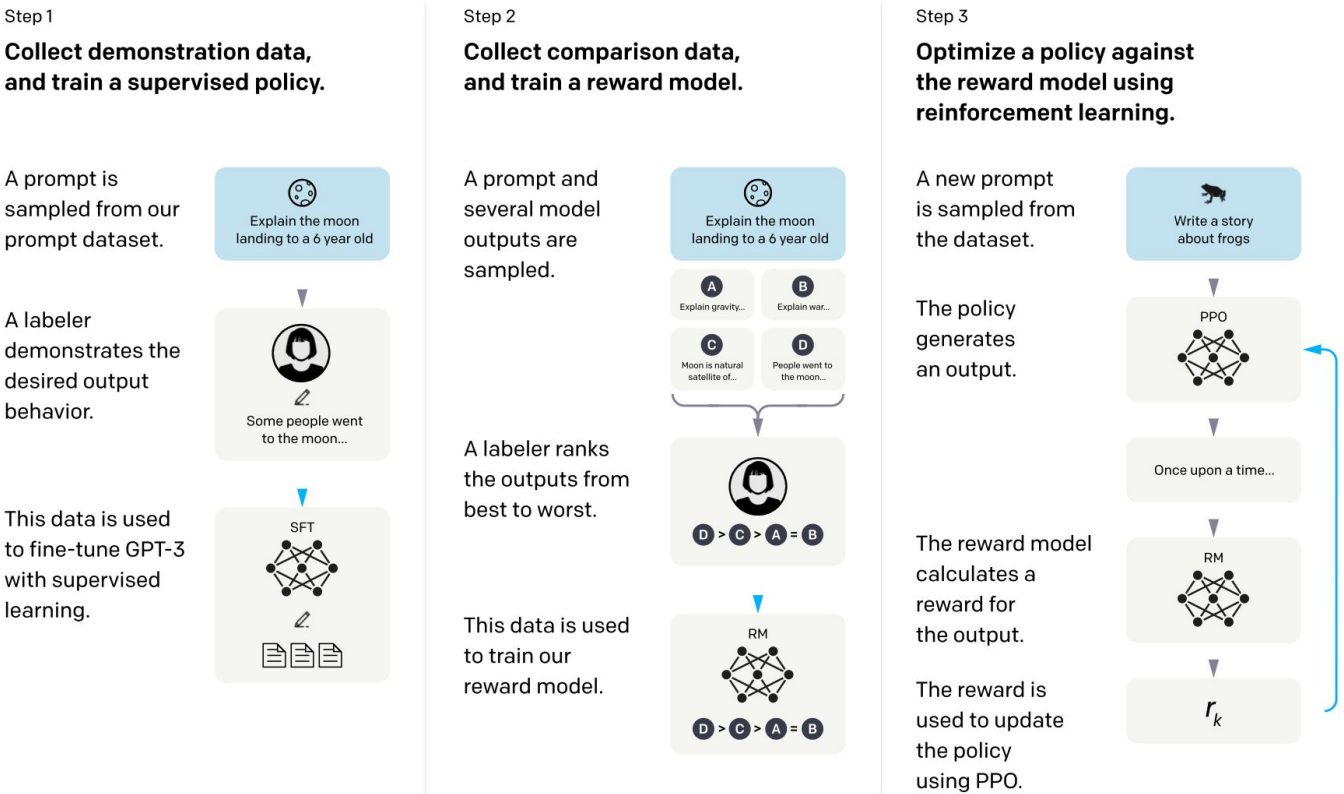
Fonte: [Prompting: Better Ways of Using Language Models for NLP Tasks, The Gradient]

T5 (Raffel et al., 2020)

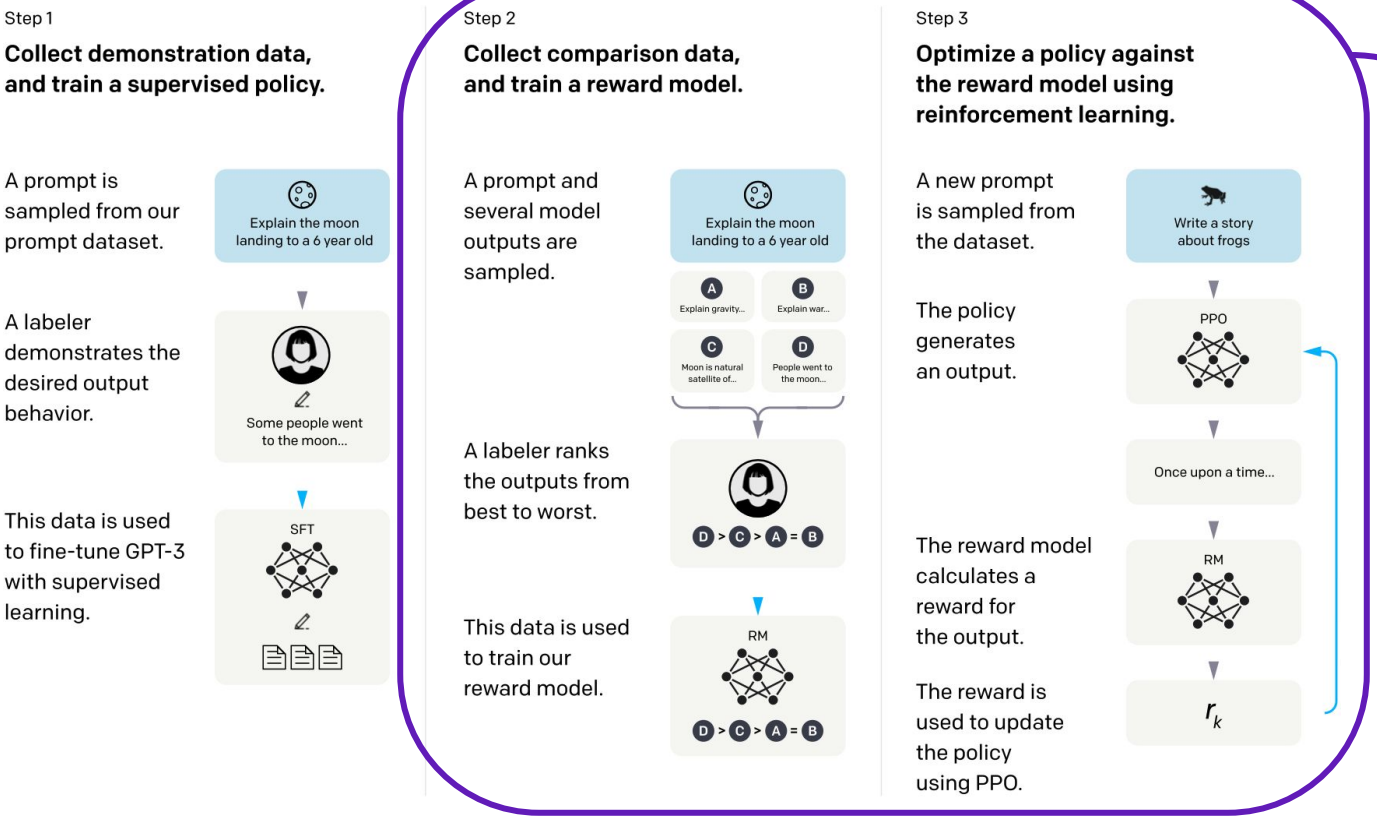
- Encoder-Decoder Tranformer model
- Pensato con l'intenzione di riformulare tutti i task di NLP in un formato di tipo "text-to-text", dove input e output sono quindi sempre stringhe di testo

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (Raffel et al., 2019), https://dl.acm.org/doi/pdf/10.5555/3455716.3455856

Instruction Tuning e RLHF: da GPT-3 a InstructGPT



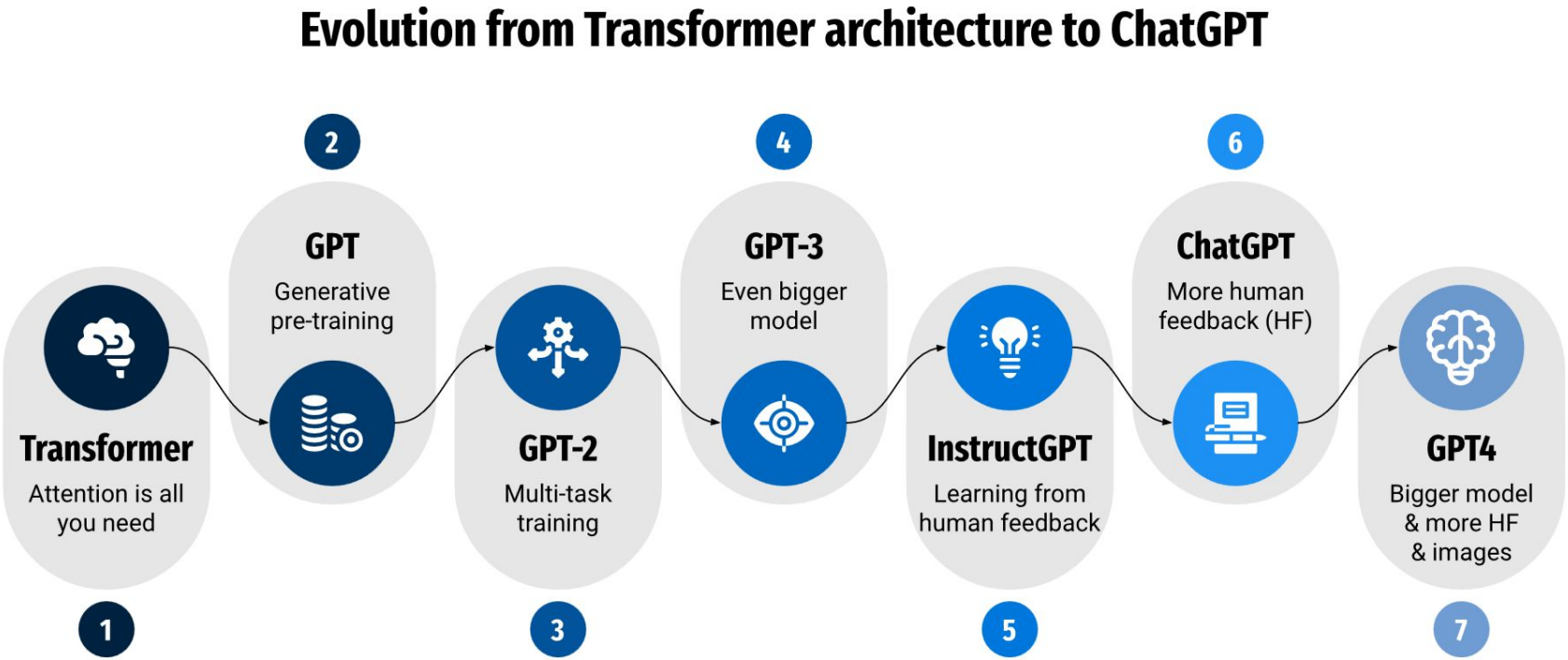
Instruction Tuning e RLHF: da GPT-3 a InstructGPT



Reinforcement Learning from Human Feedback (RLHF)

https://huggingface.co/blog/rlhf

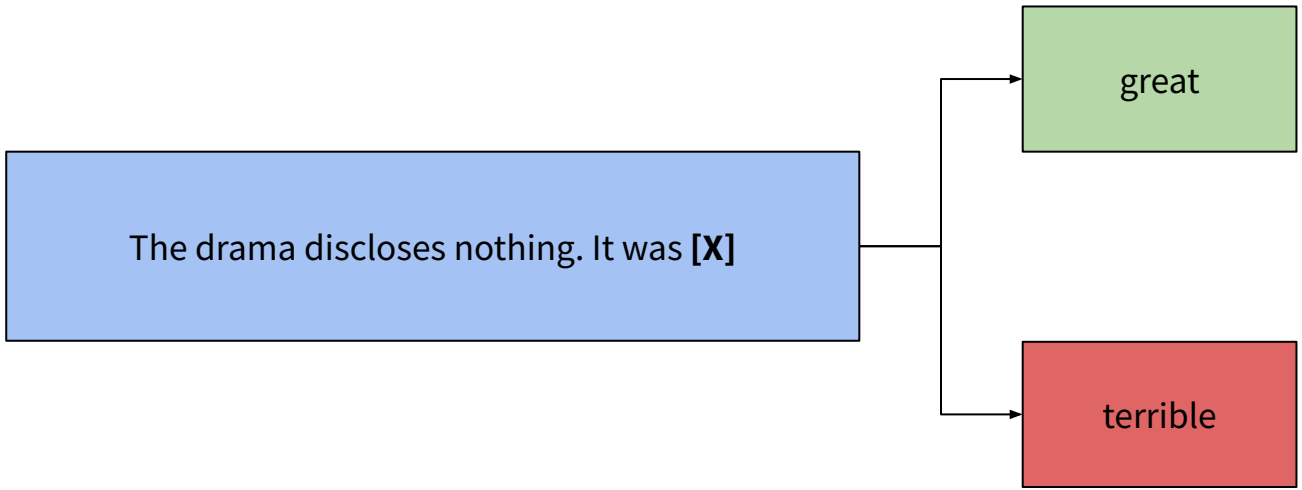
Dal Transformer a GPT4



ChatGPT: Jack of all trades, master of none (Kocoń et al., 2023), https://www.sciencedirect.com/science/article/pii/S156625352300177X

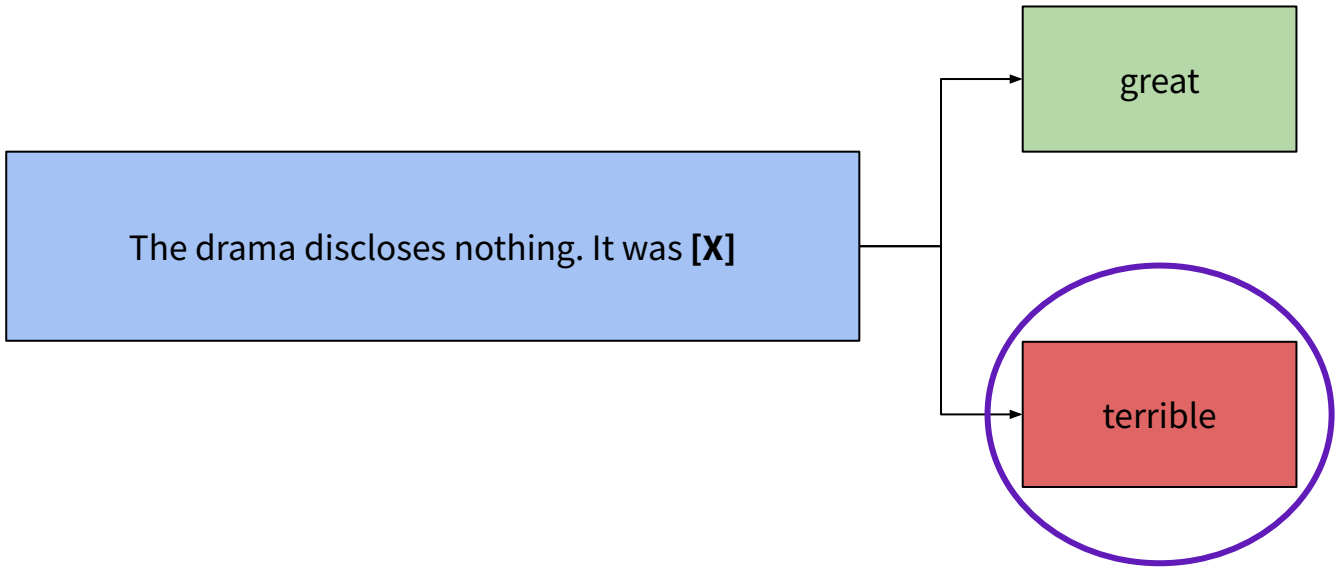
Large Language Models (LLMs)

Zero-Shot Text Classification



Large Language Models (LLMs)

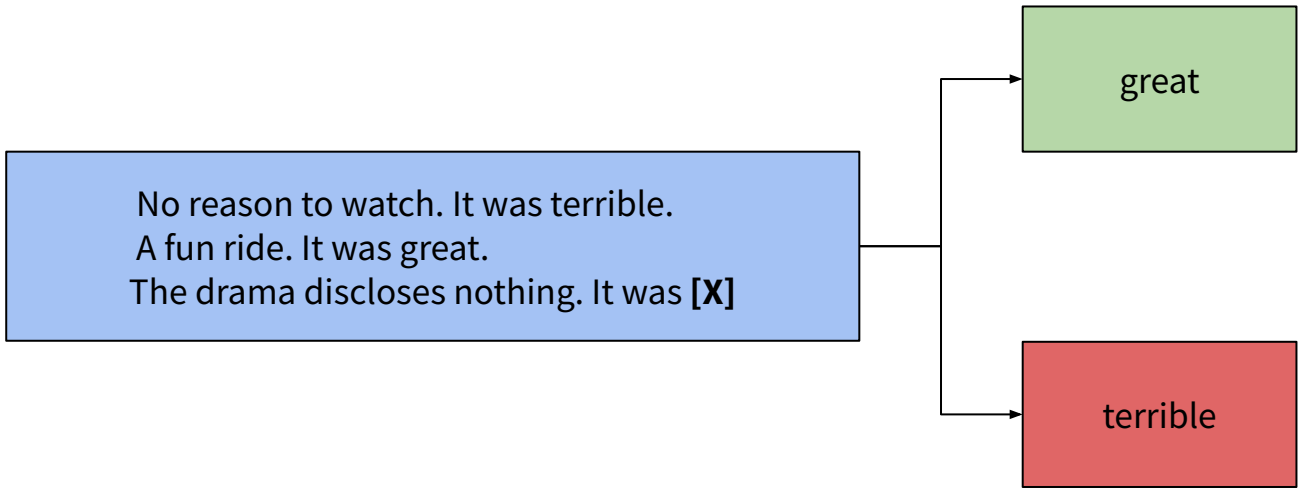
Zero-Shot Text Classification



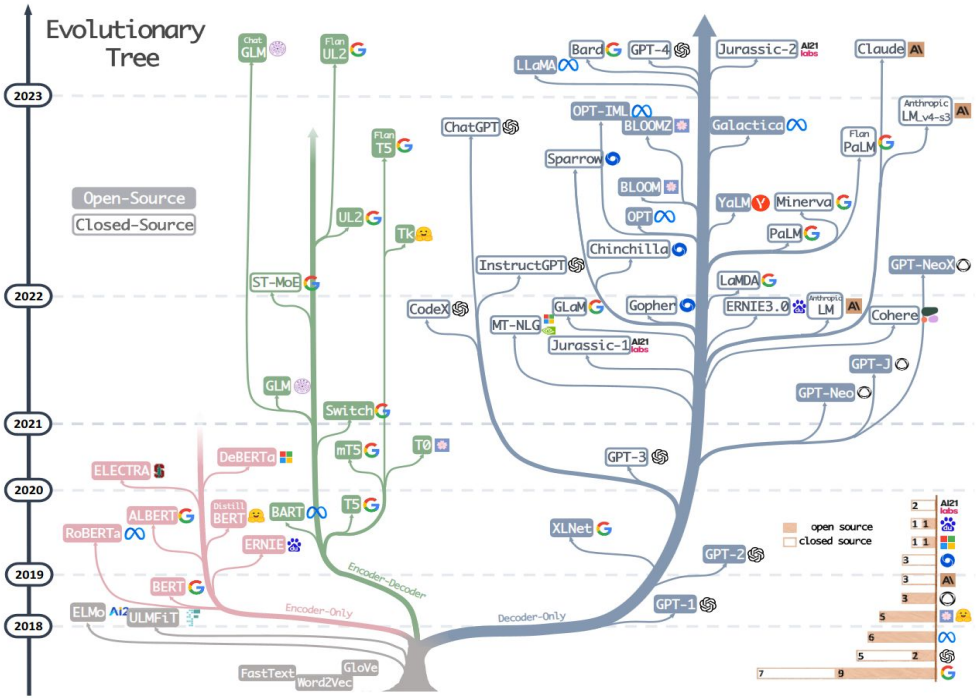
Si confronta la probabilità che il modello generi il token "great" rispetto al token "terrible".

Large Language Models (LLMs)

Few-Shot Text Classification



“Evolutionary Tree” dei recenti NLMs



Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond (Yang et al., 2024), https://dl.acm.org/doi/10.1145/3649506

Un po’ di pratica

https://github.com/michelepapucci/llms-anatomy-course