



LARGE LANGUAGE MODELS AND HOW TO INSTRUCTION TUNE THEM (IN A SUSTAINABLE WAY)

+ .
o

DANILO CROCE AND CLAUDIO D. HROMEI
UNIVERSITY OF ROMA, «TOR VERGATA»

CLIC-IT 2023 – TUTORIAL
VENICE, DEC 2, 2023

Tutorial Material

https://github.com/crux82/CLiC-it_2023_tutorial

You can access to the material for the tutorial:

- These slides
- Code



Tutorial Overview

- **Tutorial Objective**
- **Historical Overview:**
 - Tracing a (small) part of the journey from Transformer Architecture to the concept of Instruction Tuning.
- **Delving into the ExtremITA experience:**
 - Integrating LLMs with Instruction Tuning for Multi-task Learning.
- **Hands-on coding session**

Before starting

Remember:

- There are no *stupid questions*- only *silly answers!* 😊
- The real goof is the one who's teaching (that's me/us!) 🎓
- Your questions make this journey insightful and fun for everyone!

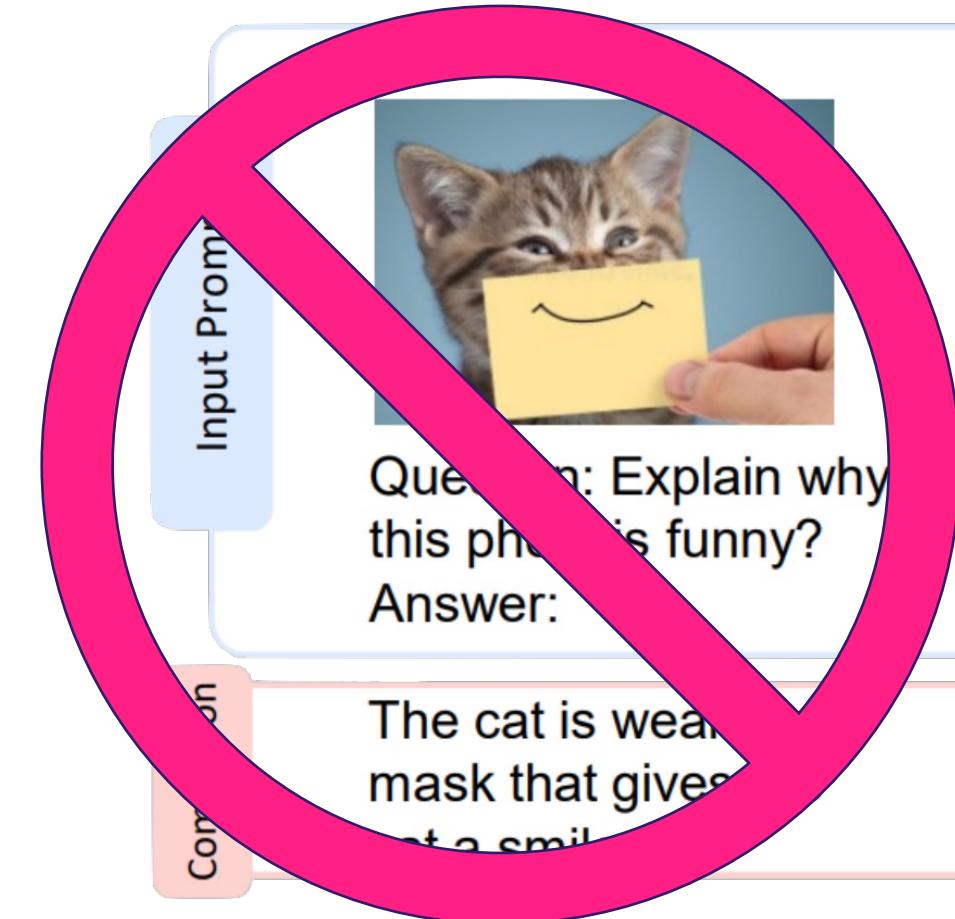
"In the World of Learning, Every Question is a Step Forward"

• So, let's make this interactive:

- 🙋‍♀️🙋‍♂️ Don't hesitate to ask anything that comes to mind.
- 🔎 Explore, inquire, and challenge ideas- it's all part of the learning process!
- 💡 Every question you ask is a chance for us all to learn something new.

Disclaimer(s)

No multi-modality with LLMS, even thought is such an interesting problem..



Anyway if you are interested in «Multimodal LLMs»

from <https://huyenchip.com/2023/10/10/multimodal.html>

[Microsoft COCO Captions: Data Collection and Evaluation Server](#) (Apr 2015)

[VQA: Visual Question Answering](#) (May 2015)

[VideoBERT: A Joint Model for Video and Language Representation Learning](#) (Google, Apr 3, 2019)

[LXMERT: Learning Cross-Modality Encoder Representations from Transformers](#) (UNC Chapel Hill, Aug 20, 2019)

[\[CLIP\] Learning Transferable Visual Models From Natural Language Supervision](#) (OpenAI, 2021)

[Unifying Vision-and-Language Tasks via Text Generation](#) (UNC Chapel Hill, May 2021)

[BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation](#) (Salesforce, Jan 28, 2022)

[Flamingo: a Visual Language Model for Few-Shot Learning](#) (DeepMind, April 29, 2022)

[GIT: A Generative Image-to-text Transformer for Vision and Language](#) (Microsoft, May 2, 2022)

[Multilnstruct: Improving Multi-Modal Zero-Shot Learning via Instruction Tuning](#) (Xu et al., Dec 2022)

[BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models](#) (Salesforce, Jan 30, 2023)

[Cross-Modal Fine-Tuning: Align then Refine](#) (Shen et al., Feb 11, 2023)

Anyway if you are interested in «Multimodal LLMs»

from <https://huyenchip.com/2023/10/10/multimodal.html>

[KOSMOS-1: Language Is Not All You Need: Aligning Perception with Language Models](#) (Microsoft, Feb 27, 2023)

[PaLM-E: An Embodied Multimodal Language Model](#) (Google, Mar 10, 2023)

[LLaMA-Adapter: Efficient Fine-tuning of Language Models with Zero-init Attention](#) (Zhang et al., Mar 28, 2023)

[mPLUG-Owl: Modularization Empowers Large Language Models with Multimodality](#) (Ye et al., Apr 2, 2023)

[LLaMA-Adapter V2: Parameter-Efficient Visual Instruction Model](#) (Gao et al., Apr 28, 2023)

[LLAVA: Visual Instruction Tuning](#) (Liu et al., Apr 28, 2023)

[X-LLM: Bootstrapping Advanced Large Language Models by Treating Multi-Modalities as Foreign Languages](#) (Chen et al., May 7, 2023)

[InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning](#) (Salesforce, May 11, 2023)

[Towards Expert-Level Medical Question Answering with Large Language Models](#) (Singhal et al., May 16, 2023)

[Cheap and Quick: Efficient Vision-Language Instruction Tuning for Large Language Models](#) (Luo et al., May 24, 2023)

[Shikra: Unleashing Multimodal LLM's Referential Dialogue Magic](#) (SenseTime, Jun 3, 2023)

[Macaw-LLM: Multi-Modal Language Modeling with Image, Audio, Video, and Text Integration](#) (Tencent, Jun 15, 2023)

Let's start from the beginning



EVALITA

- EVALITA is a **periodic evaluation campaign** of Natural Language Processing (NLP) and speech tools for the **Italian language**,
 - an **initiative of the Associazione Italiana di Linguistica Computazionale (AILC)**
- **Shared benchmark for different tracks:**
 - Affect Detection
 - Authorship Analysis
 - Computational Ethics
 - Linguistic Analysis (syntactic/semantic parsing, POS-tagging)
 - Information Extraction (named-entity recognition)
 - Discourse Coherence
 - ...



<https://www.evalita.it>

EVALITA 2023

13 tasks (22 subtasks)

- **Affect**

- [EMit](#) – Categorical Emotion Detection in Italian Social Media
- [EmotivITA](#) – Dimensional and Multi-dimensional emotion analysis

- **Authorship Analysis**

- [PoliticIT](#) – Political Ideology Detection in Italian Texts
- [GeoLingIt](#) – Geolocation of Linguistic Variation in Italy
- [LangLearn](#) – Language Learning Development



<https://www.evalita.it>

EVALITA 2023

13 tasks (22 subtasks)

- **Computational Ethics**

- [HaSpeeDe 3](#) – Political and Religious Hate Speech Detection
- [HODI](#) – Homotransphobia Detection in Italian
- [MULTI-Fake-DetectiVE](#) – MULTImodal Fake News Detection and VErification
- [ACTI](#) – Automatic Conspiracy Theory Identification

- **New Challenges in Long-standing Tasks**

- [NERMuD](#) -Named-Entities Recognition on Multi-Domain Documents
- [CLinkaRT](#) – Linking a Lab Result to its Test Event in the Clinical Domain
- [WiC-ITA](#) – Word-in-Context task for Italian
- [DisCoTEX](#) – Assessing DIScourse COherence in Italian TEXts



<https://www.evalita.it>

A little «technological» context

- What was the hype in AI/NLP in February 2023?
(still nowadays)

After one year with, we **ask everything to ChatGPT**

- Even not really smart questions like

How to make pasta

and we expect clever/useful/kind answers

You
How to make pasta

ChatGPT
Making pasta is a relatively simple process, and you can enjoy a delicious homemade meal with just a few ingredients. Here's a basic recipe for making traditional Italian pasta from scratch, along with instructions for cooking it:

Ingredients:

- 2 cups all-purpose flour
- 3 large eggs
- A pinch of salt

Instructions:

1. Make the Dough:
 - a. On a clean surface or in a large mixing bowl, mound the flour and create a well in the center.
 - b. Crack the eggs into the well and add a pinch of salt.
 - c. Using a fork, gradually incorporate the flour into the eggs until a dough starts to form.
 - d. Once the dough comes together, knead it for about 8-10 minutes until it becomes smooth and elastic. If the dough is too sticky, you can add a bit more flour.
 - e. Wrap the dough in plastic wrap and let it rest for at least 30 minutes at room temperature. This

A bit of context (2)

Clearly it could be interesting to use it and participate **win** a task

Let us consider the GeoLingIT task:
“Given the text of a **tweet** exhibiting non-standard Italian language, **predict its region of provenance** [...] and **its location** in terms of longitude and latitude coordinates.”

 You
Scrivi la regione di appartenenza di chi ha scritto questo testo, seguito dalla latitudine, seguita dalla longitudine.

"Si amico mio. Te accolli. Altro che sette. Saranno minimo na ventina a esse boni."

 ChatGPT
Il testo fornito sembra essere scritto in dialetto italiano, con alcune espressioni tipiche di regioni meridionali, come ad esempio il termine "na" anziché "una" e l'uso della parola "boni" che potrebbe essere una variante dialettale per indicare persone.

Senza ulteriori informazioni sul contesto o sulla provenienza specifica del testo, è difficile identificare esattamente la regione di provenienza. Tuttavia, basandosi sulle espressioni dialettali, potrebbe essersi parlato in una regioni del Sud Italia, come la Campania, la Calabria o la Sicilia.



Ricorda che le espressioni dialettali possono variare notevolmente anche all'interno di una stessa regione. La latitudine e la longitudine specifiche non possono essere determinate solo dal testo fornito.

Leveraging LLMs for Specific Tasks

Can Large Language Models be tailored to address precise language processing challenges?

- E.g., NER, POS tagging, Hate Speech Detection, Discourse Coherence...

Our Expectations:

- **Natural Language Interaction:** Use the model using conversational language.
- **Prompt Engineering:** Utilize sophisticated prompt-design techniques for enhanced model performance.
- **Coherent Responses:** Obtain answers that align with our specified tasks.
- **High-Performance Outcomes:** Achieve *good* results in task execution.

The «crazy» idea

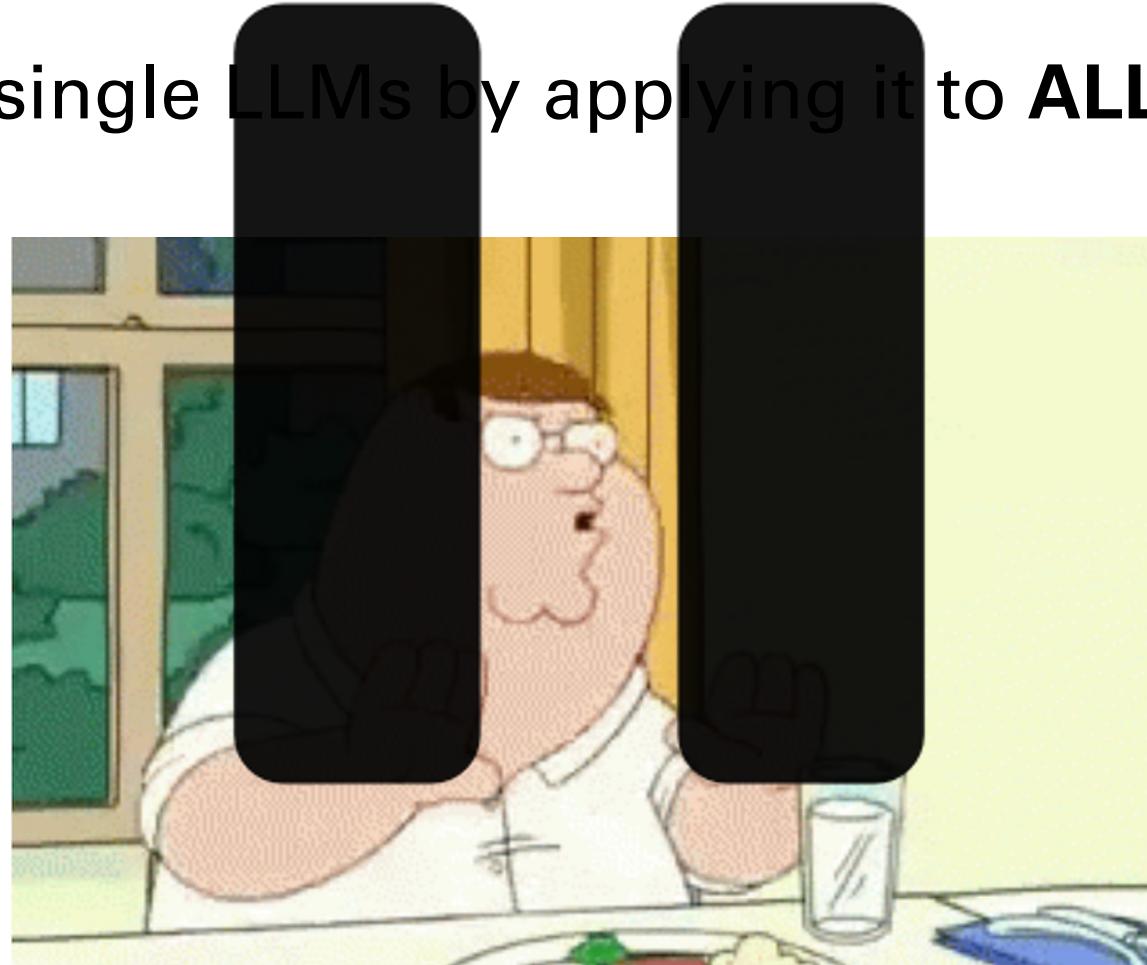
Fine-tuning a single LLMs by applying it to **ALL EVALITA tasks**



How did we get here???

The «crazy» idea

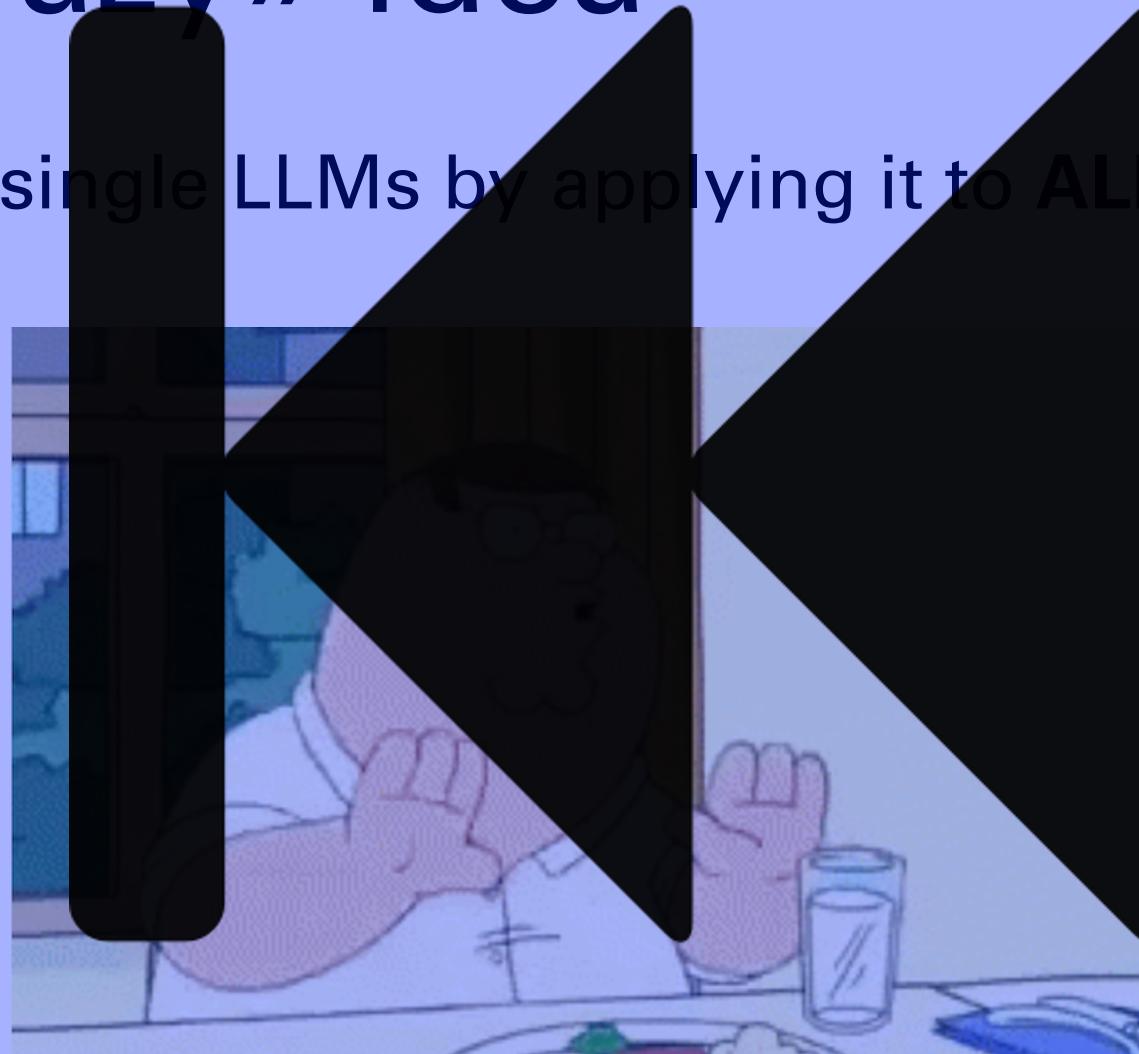
Fine-tuning a single LLMs by applying it to **ALL EVALITA tasks**



How did we get here???

The «crazy» idea

Fine-tuning a single LLMs by applying it to ALL EVALITA tasks



How did we get here???

Going back in time to 2017: the Transformer

(Vaswani et al. 2017)

A Transformer: a neural architecture designed for **sequence-to-sequence tasks**.

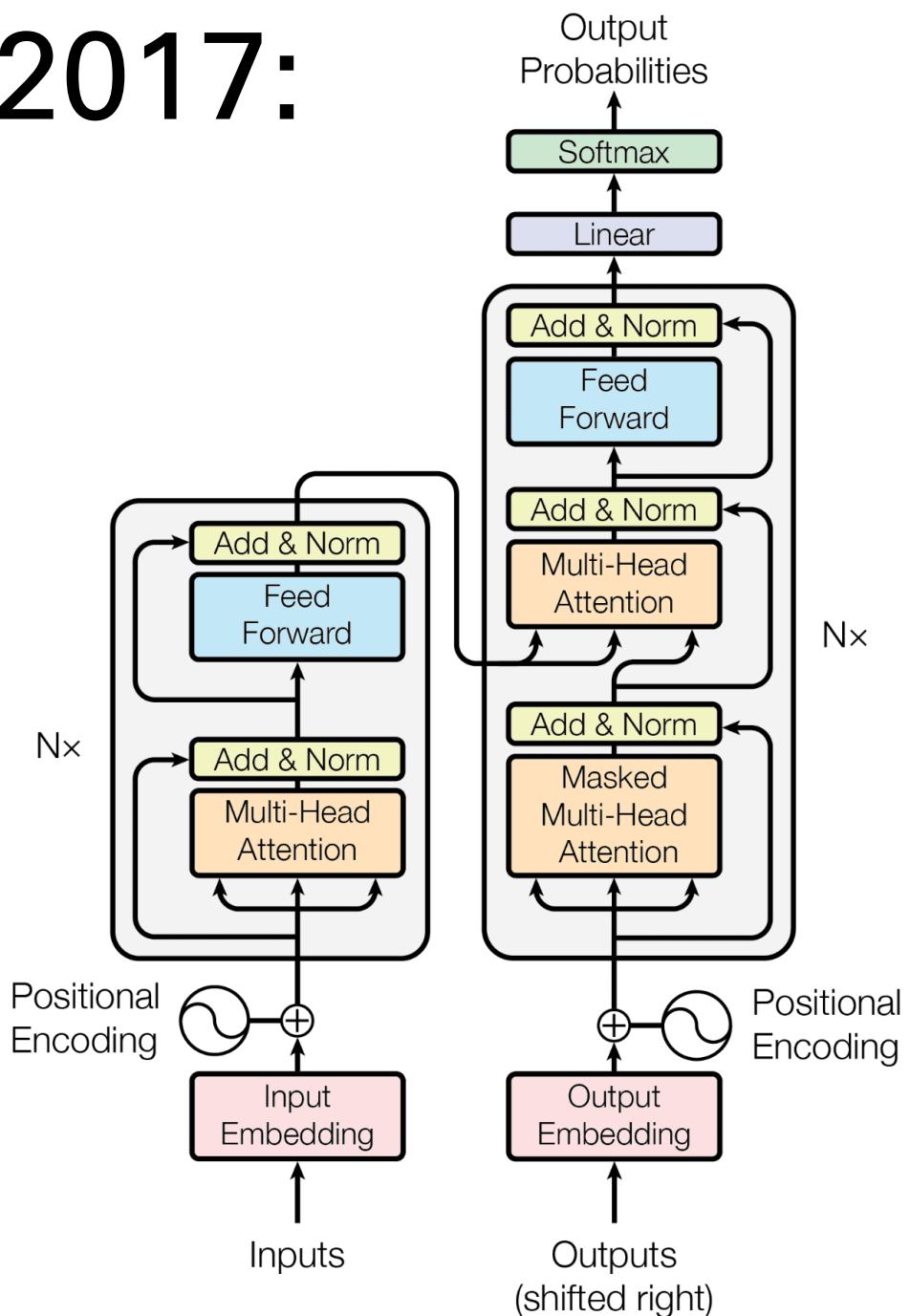
- It takes a sequence of symbols as input and produces a sequence of symbols as output.

Before Transformers:

- Until 2017, these tasks were implemented using **Recurrent Neural Networks (RNNs)**.
 - with limitations in handling long sequences.

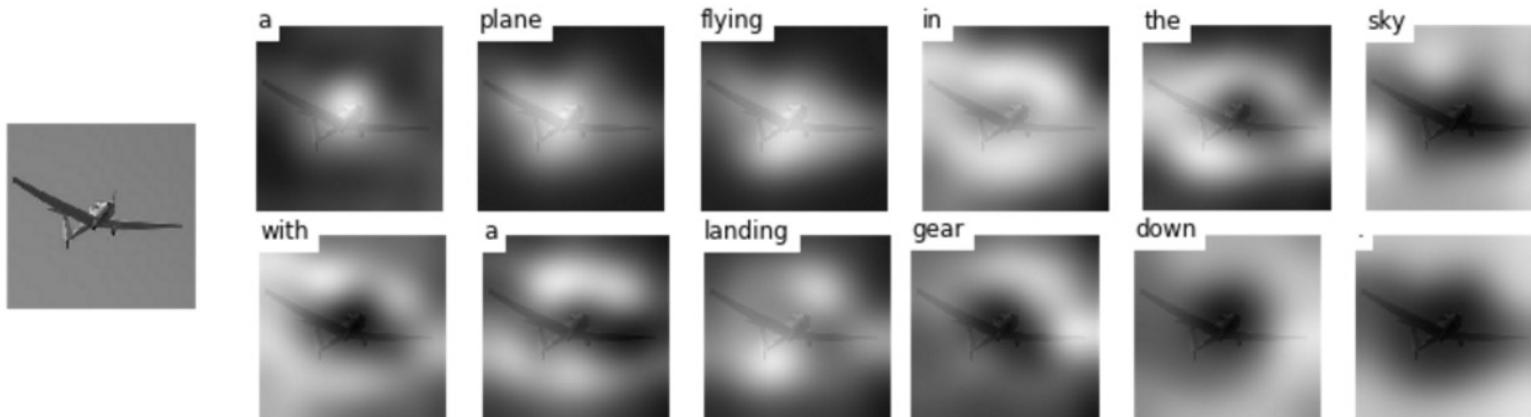
Emergence of Attention:

- Heavily used since 2015, allowed models to **focus on specific sections of a sequence** for better inference.



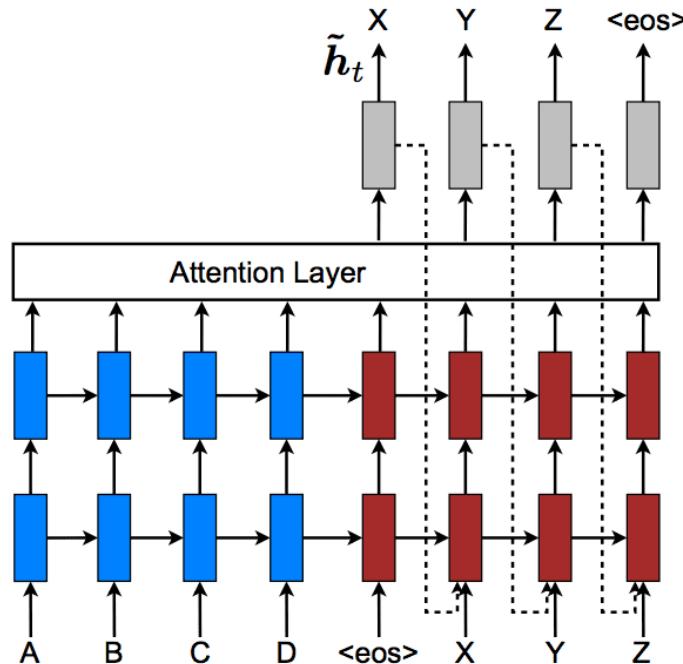
The Importance of Attention in Neural Learning

- **Revolution in Computer Vision**
 - It significantly improved **object detection** and **recognition in computer vision**.
 - It enables models to focus on relevant parts of an image, improving accuracy and efficiency.
 - **An interesting Survey:** <https://github.com/MenghaoGuo/Awesome-Vision-Attentions>
- **Breakthrough in tasks such as Image Captioning:**
 - Attention helps in **identifying key components** within images **to generate accurate and contextually relevant descriptions**.
 - **Seminal work:** (Xu et al, 2015) <https://arxiv.org/abs/1502.03044>

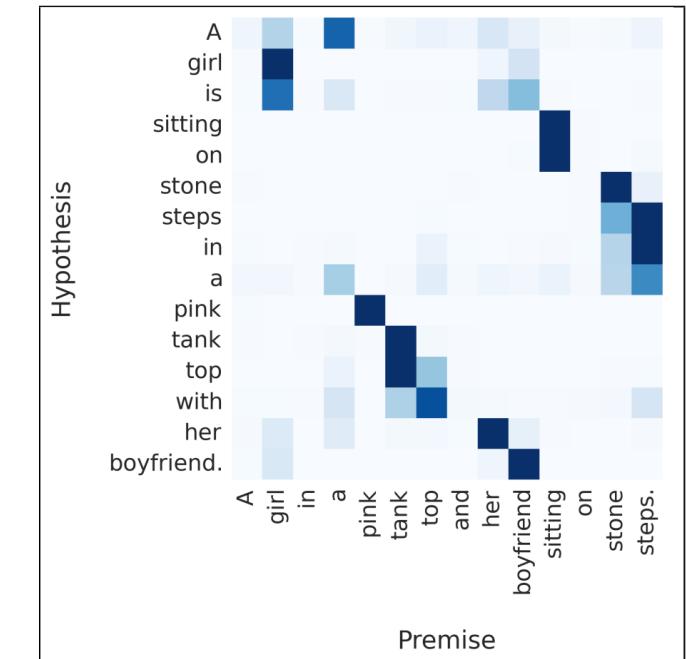


Enhancement in Recurrent Neural Networks

- For RNNs, **attention mechanisms** were used to address the **challenge of handling long sequences**.
- It allows RNNs to **focus on important parts** of the input sequence
 - improving performance in tasks like language translation and speech recognition.



(Luong et al, 2015)
Effective Approaches to Attention-based Neural Machine Translation
<https://arxiv.org/abs/1508.04025>

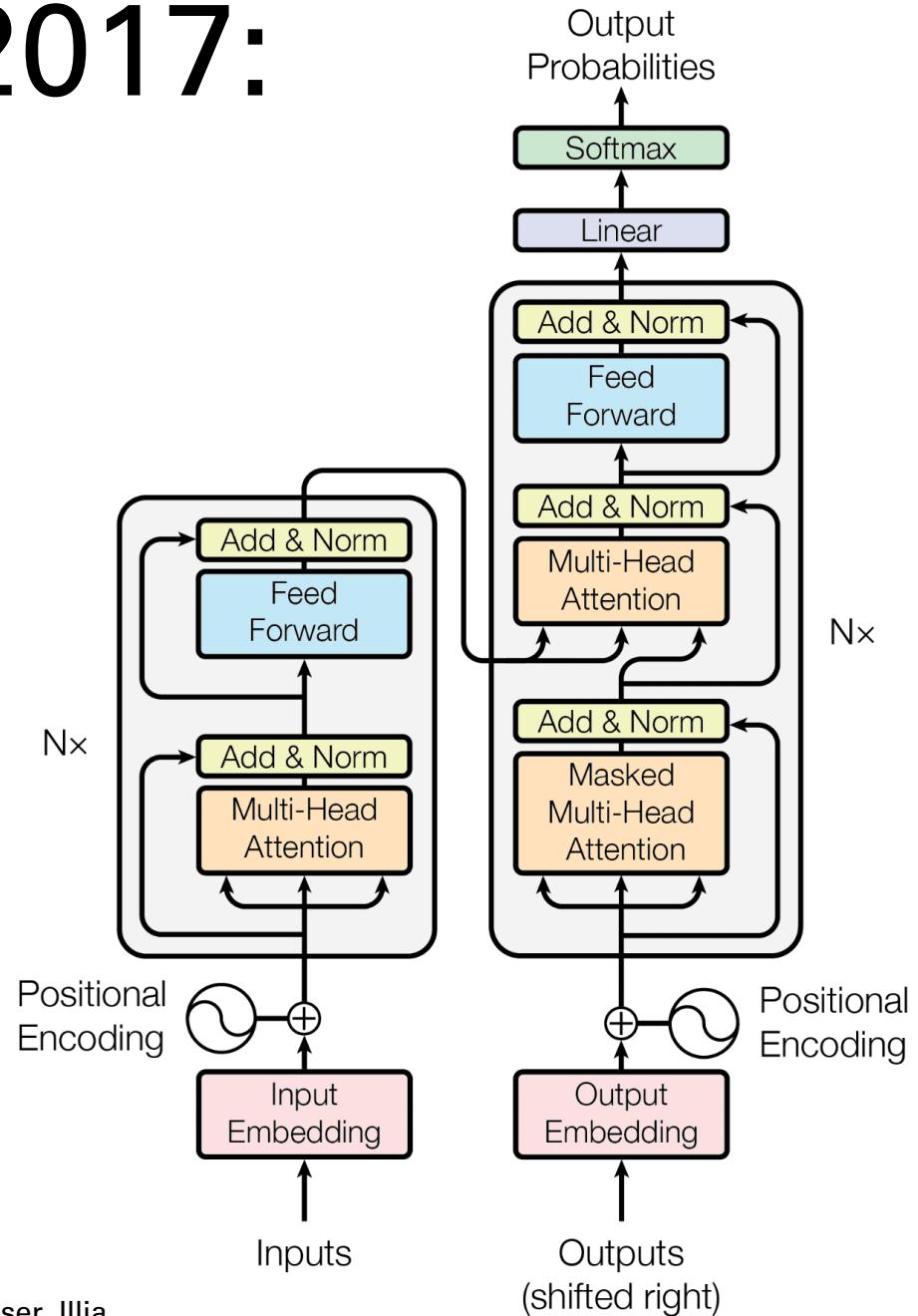


(Rocktäschel et, al., 2015)
Reasoning about Entailment with Neural Attention, 2015.
<https://arxiv.org/abs/1508.04025>

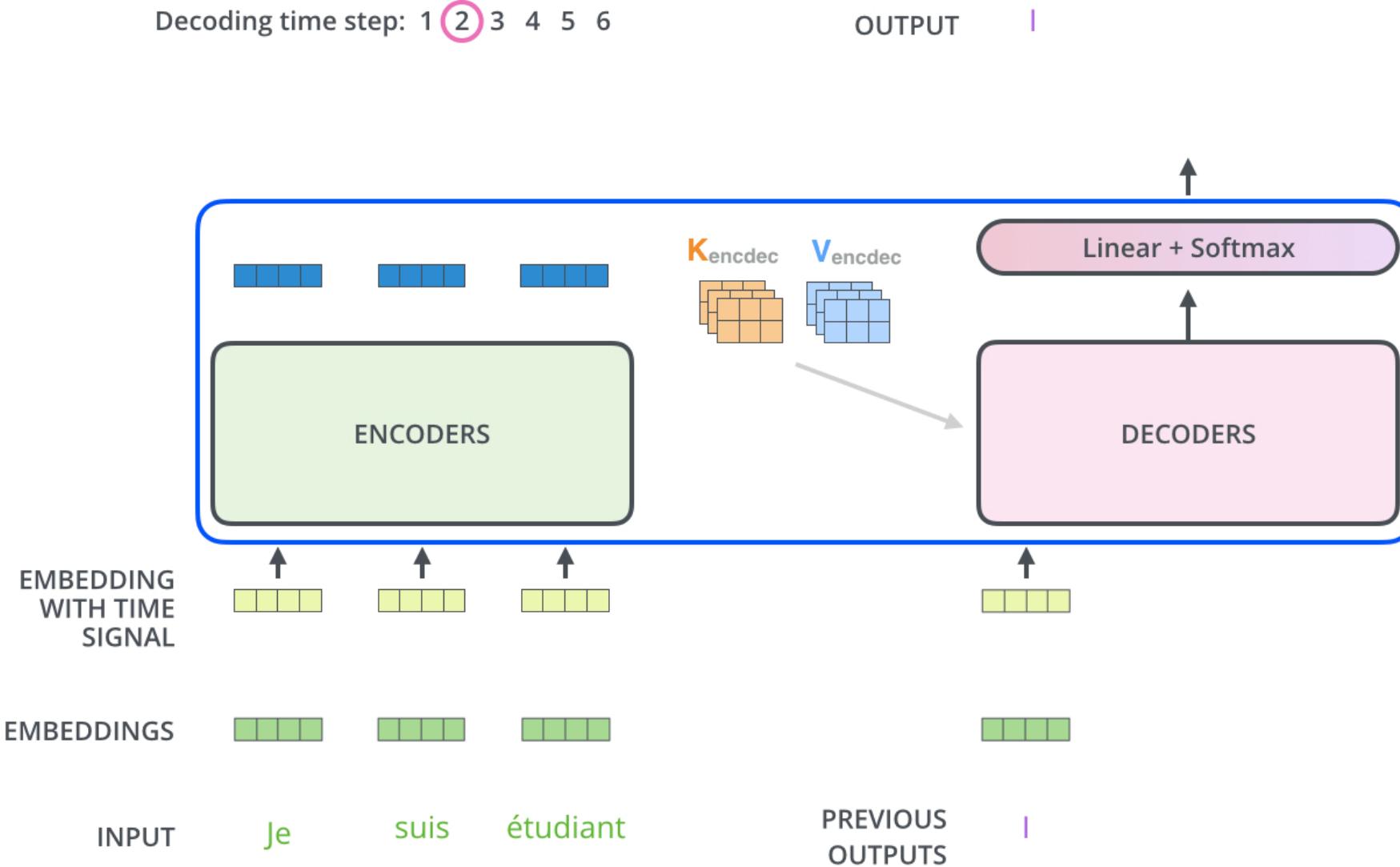
Going back in time to 2017: the Transformer

(Vaswani et al. 2017)

- **Attention in Transformers:**
 - In 2017, the **attention mechanism became an integral part** of this architecture.
 - a **significant evolution in seq2seq modeling**
- **Main advantages:**
 - Better with long range dependencies
 - Parallel processing (more scalable than RNNs)
 - State-of-the-art performances
- **Originally meant for Automatic machine translation:**
 - E.g., French to English

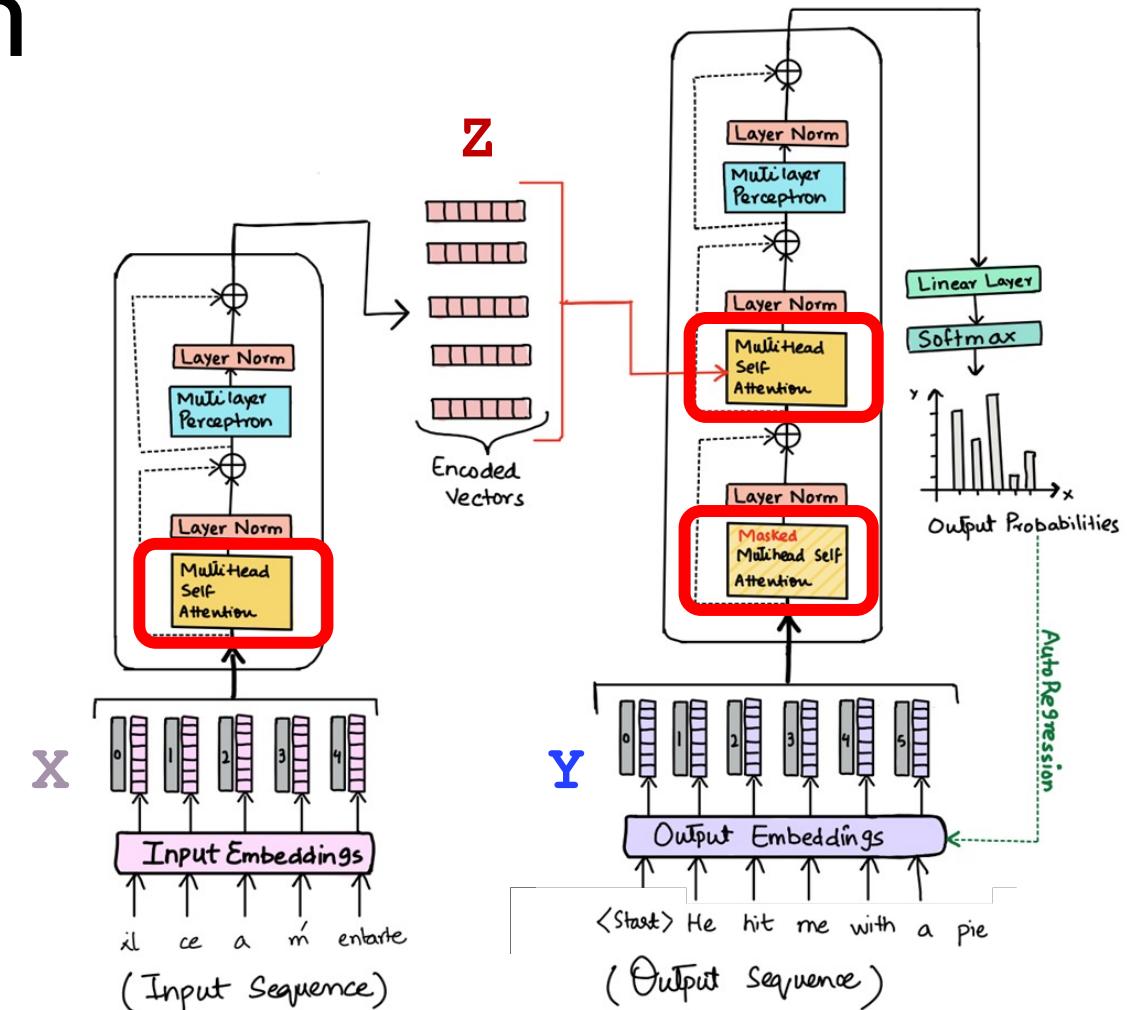


Seq2Seq: A transformer in action



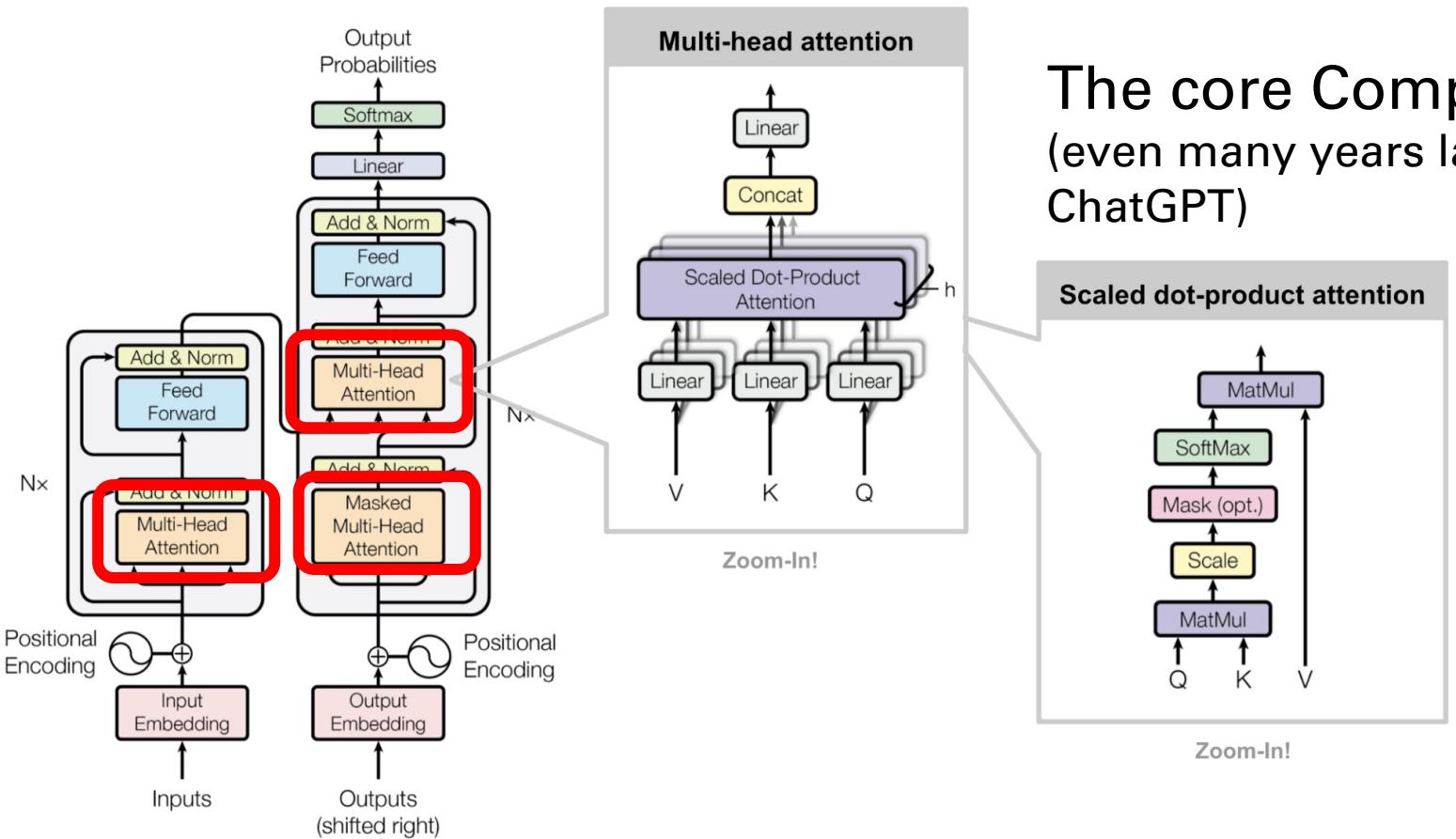
Encoding/Decoding Architecture with Attention Mechanism

- **Two components**
 - **Encoder:** Maps input sequence $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ to continuous representations $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_n)$.
 - **Decoder:** Decoder uses \mathbf{z} to generate output sequence $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_m)$
- Encoder/Decoder process input vectors through **self-attention layer** and feed-forward network.
 - It enables to selectively **concentrate on pertinent parts of the input**
 - It improves **context awareness**
 - It allows to **consider positions** in the input that also depends on the output

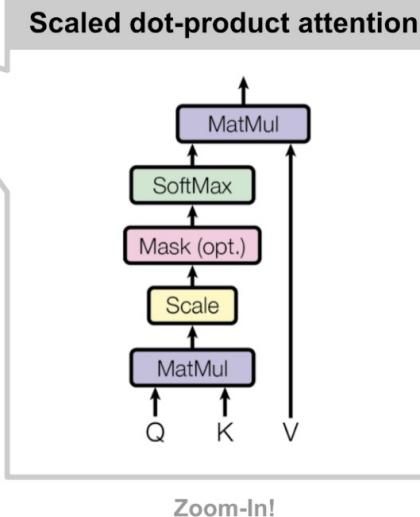


How does Self-attention work?

It is not magic, it is not a human brain, it is **just matrix multiplication**



The core Component
(even many years later in
ChatGPT)



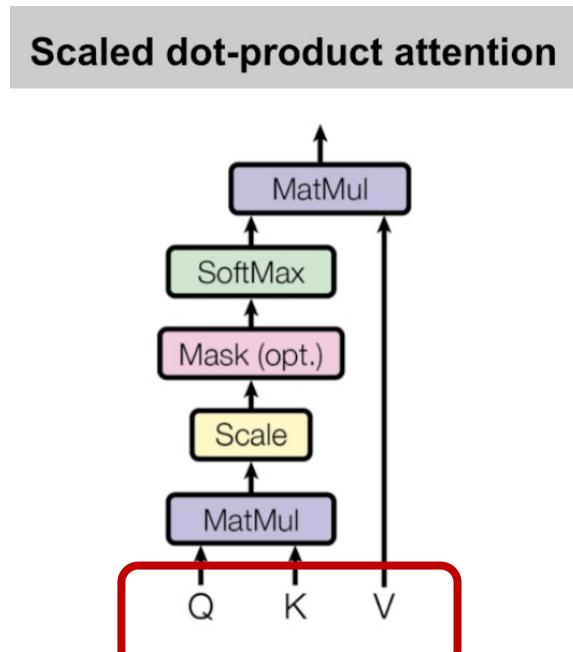
It's «simpler» than it looks
even though I'll be simplifying quite a few things.



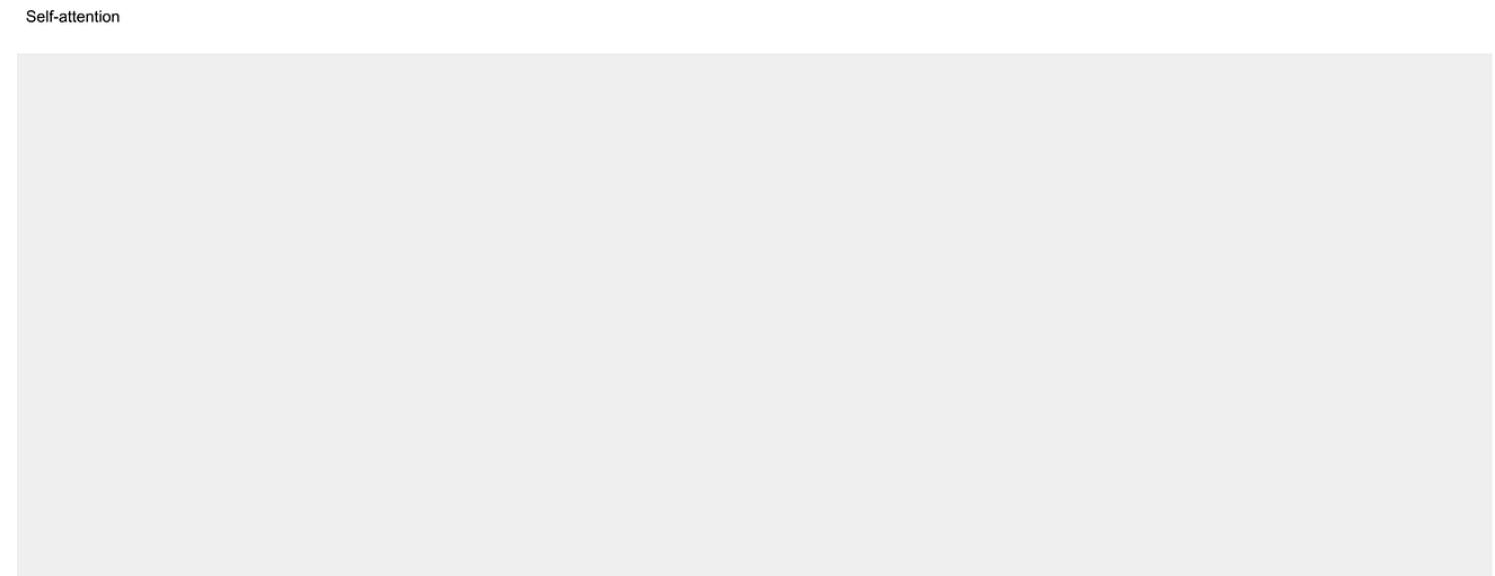
«Attention in action»

Prepare inputs

Each word is associated to embeddings



Positional Encoding: Part of these vector encodes the tokens' position



input #1
I
1 0 1 0

input #2
like
0 2 0 2

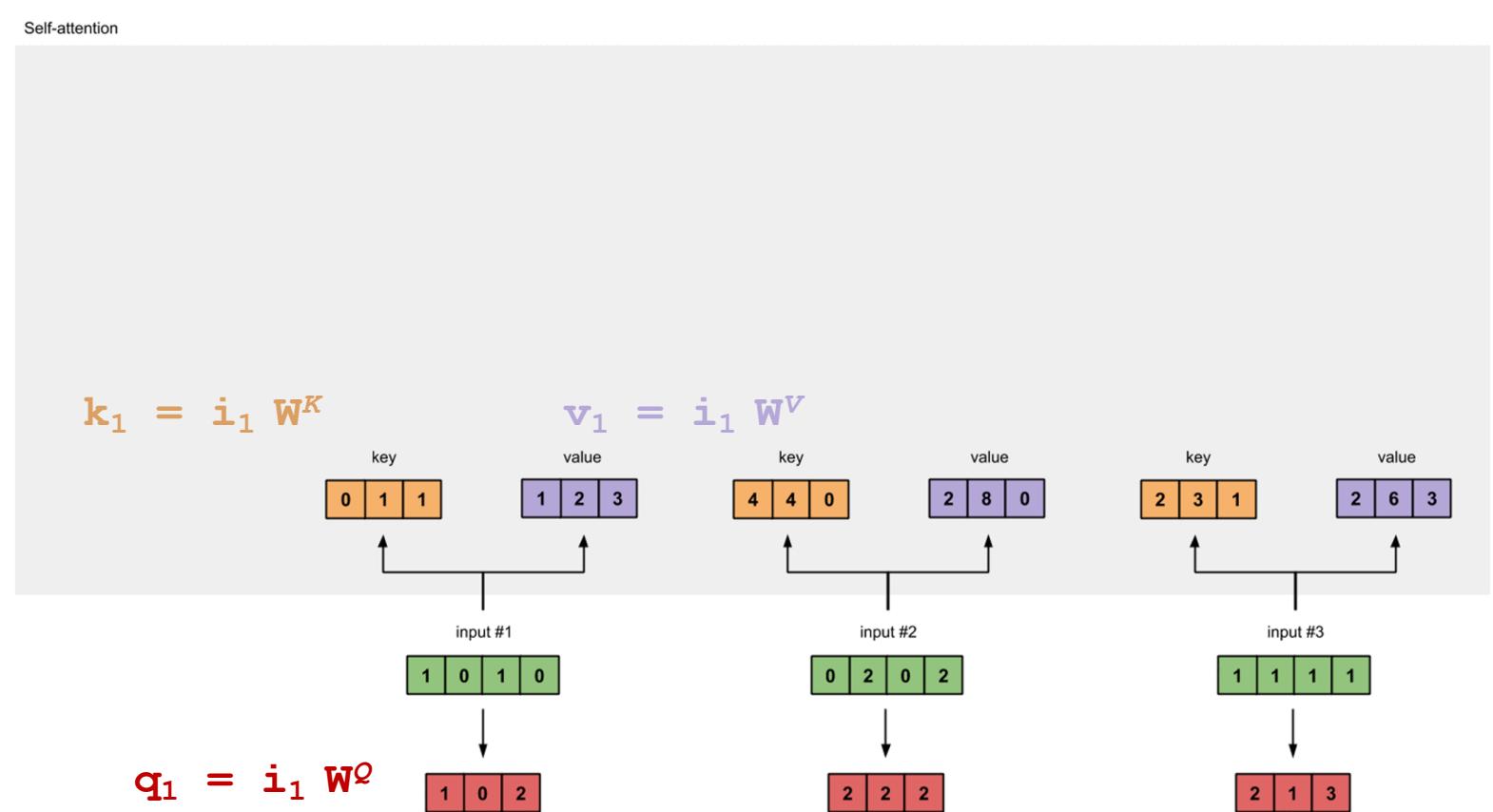
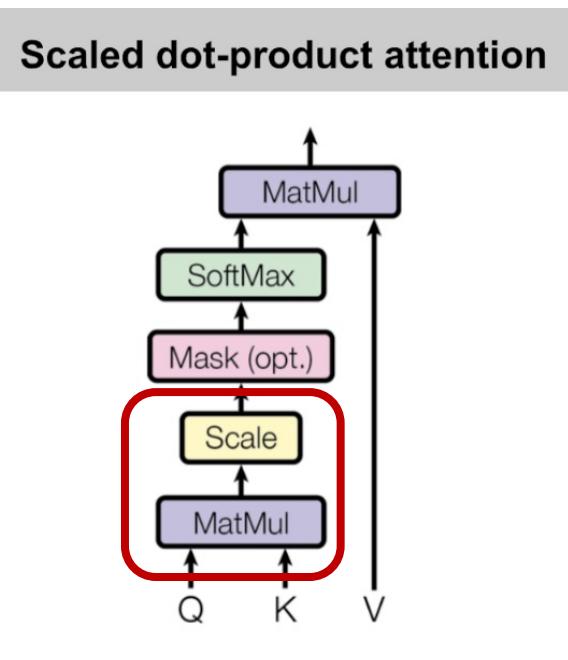
input #3
pizza
1 1 1 1

«Attention in action»

Compute Query, Key, and Value Vectors

For each word vector, calculate the Query, Key, and Value vectors by multiplying with respective weight matrices W^Q , W^K , W^V .

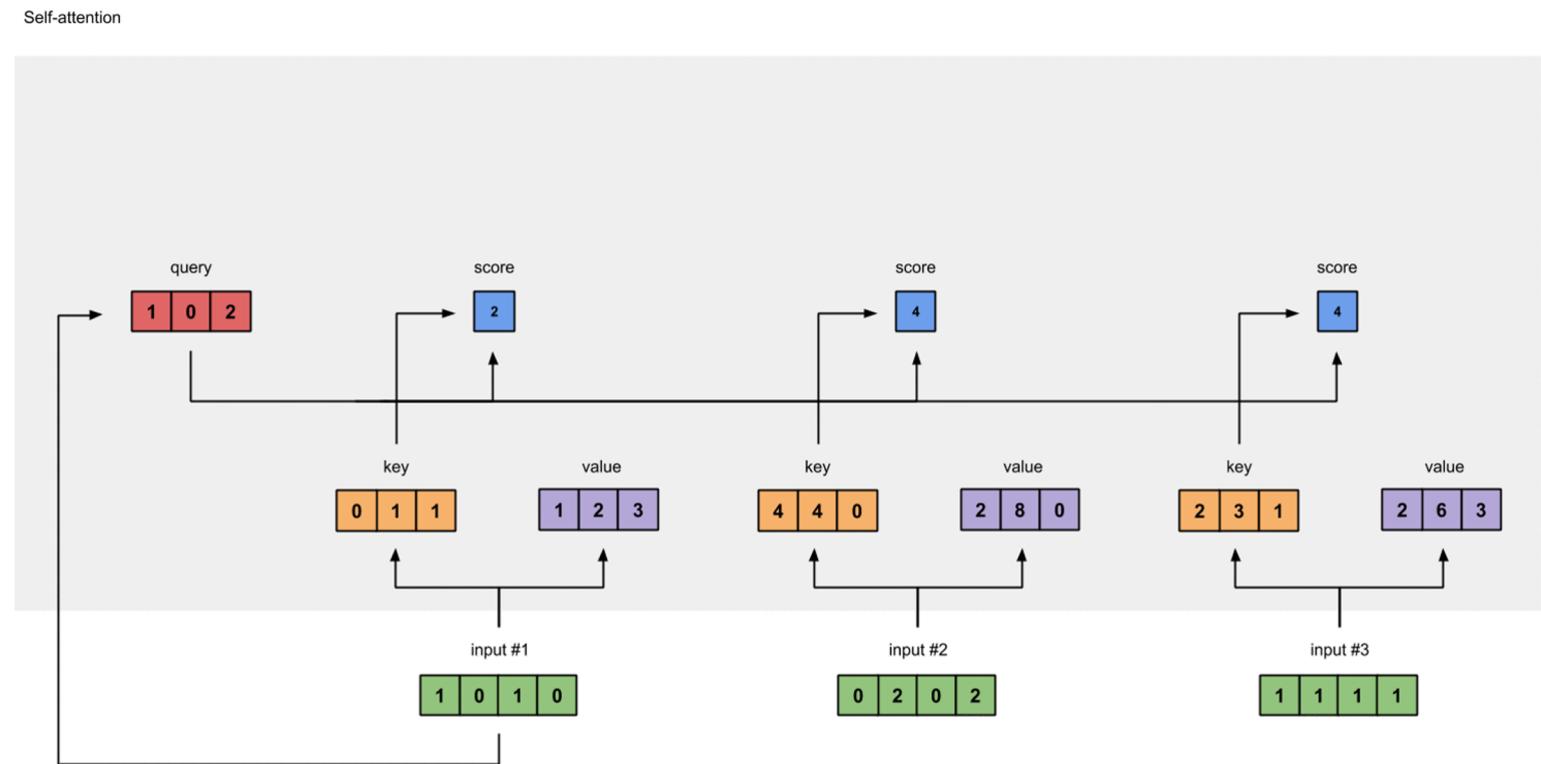
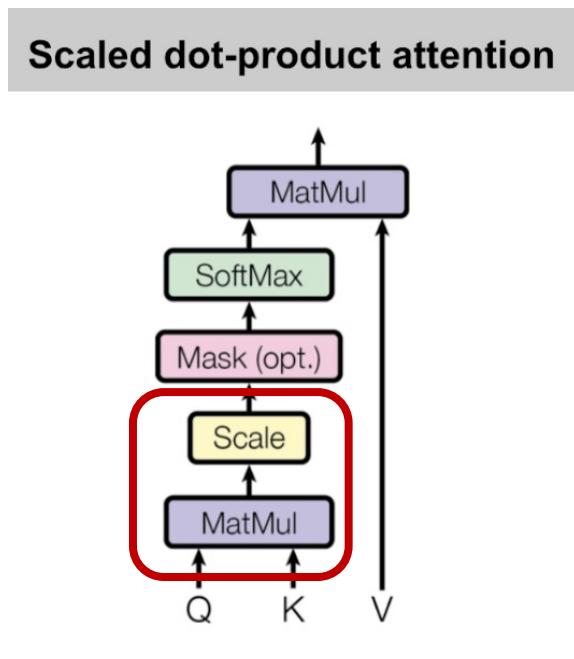
- $\text{key}_i = \text{input}_i W^K$
- $\text{value}_i = \text{input}_i W^V$
- $\text{query}_i = \text{input}_i W^Q$



«Attention in action»

Calculate the Attention scores for **input₁**

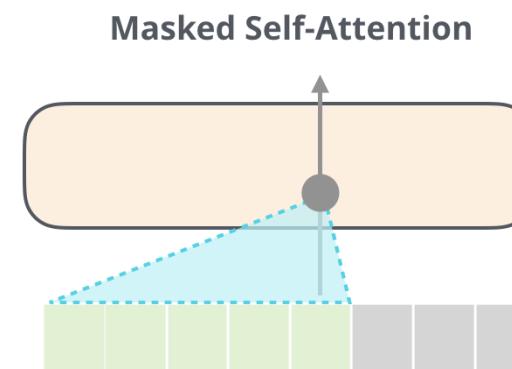
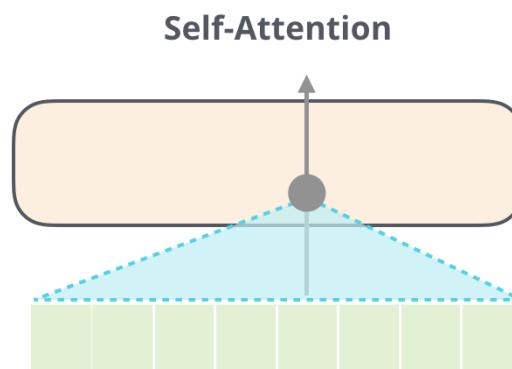
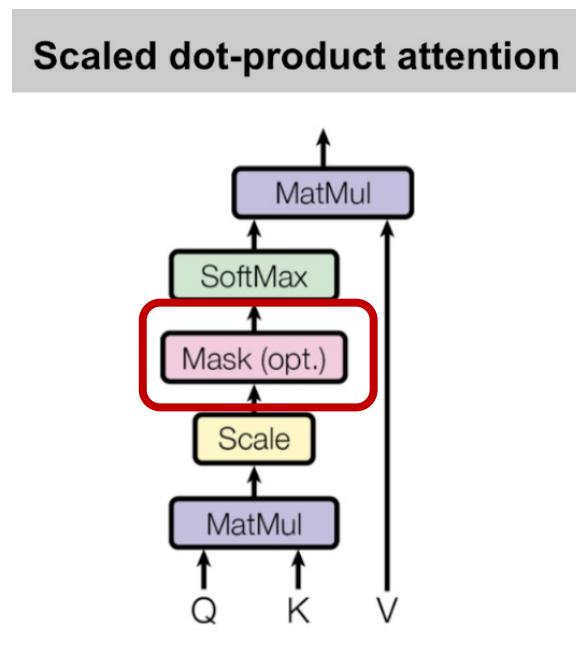
Attention scores are computed to «weight » the contribution of ALL words in the input sequence when representing **input₁**.



«Attention in action»

Role of Masked Attention

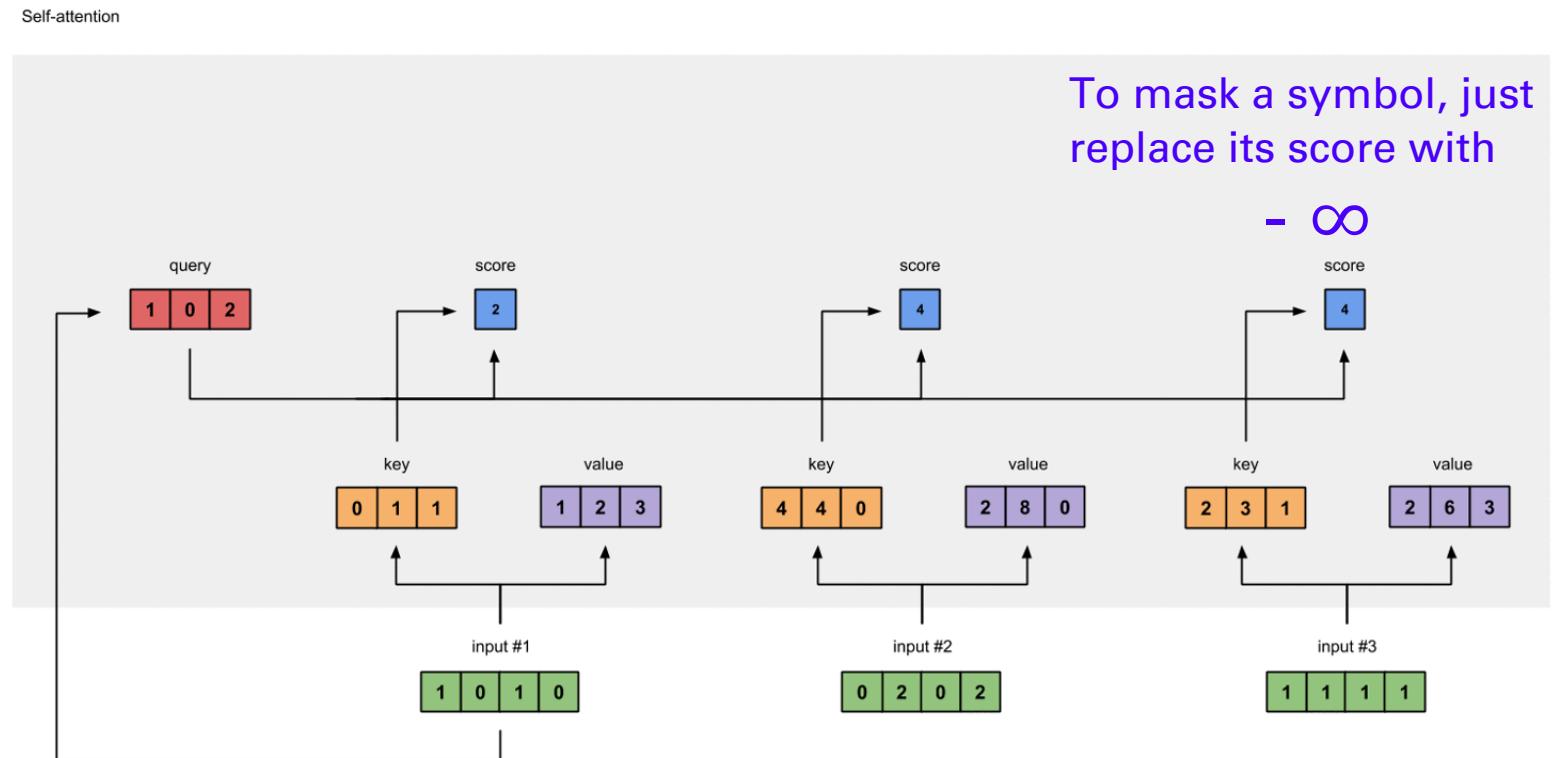
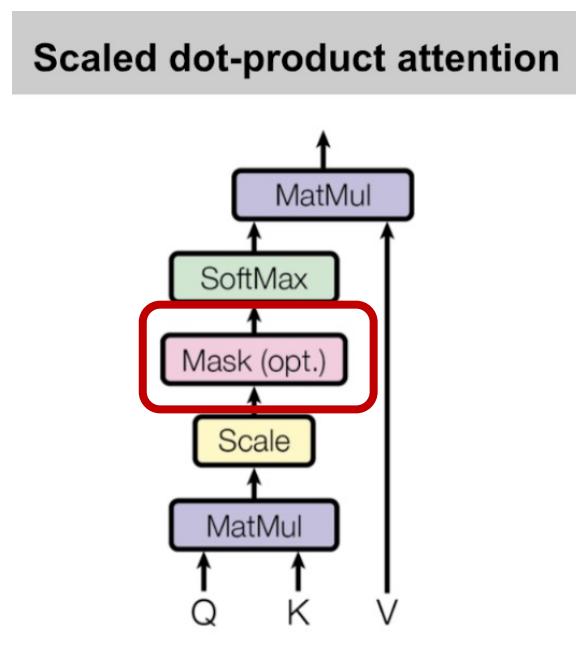
Use masked attention to handle sequences of different lengths



«Attention in action»

Role of Masked Attention (2)

Use masked attention to handle sequences of different lengths

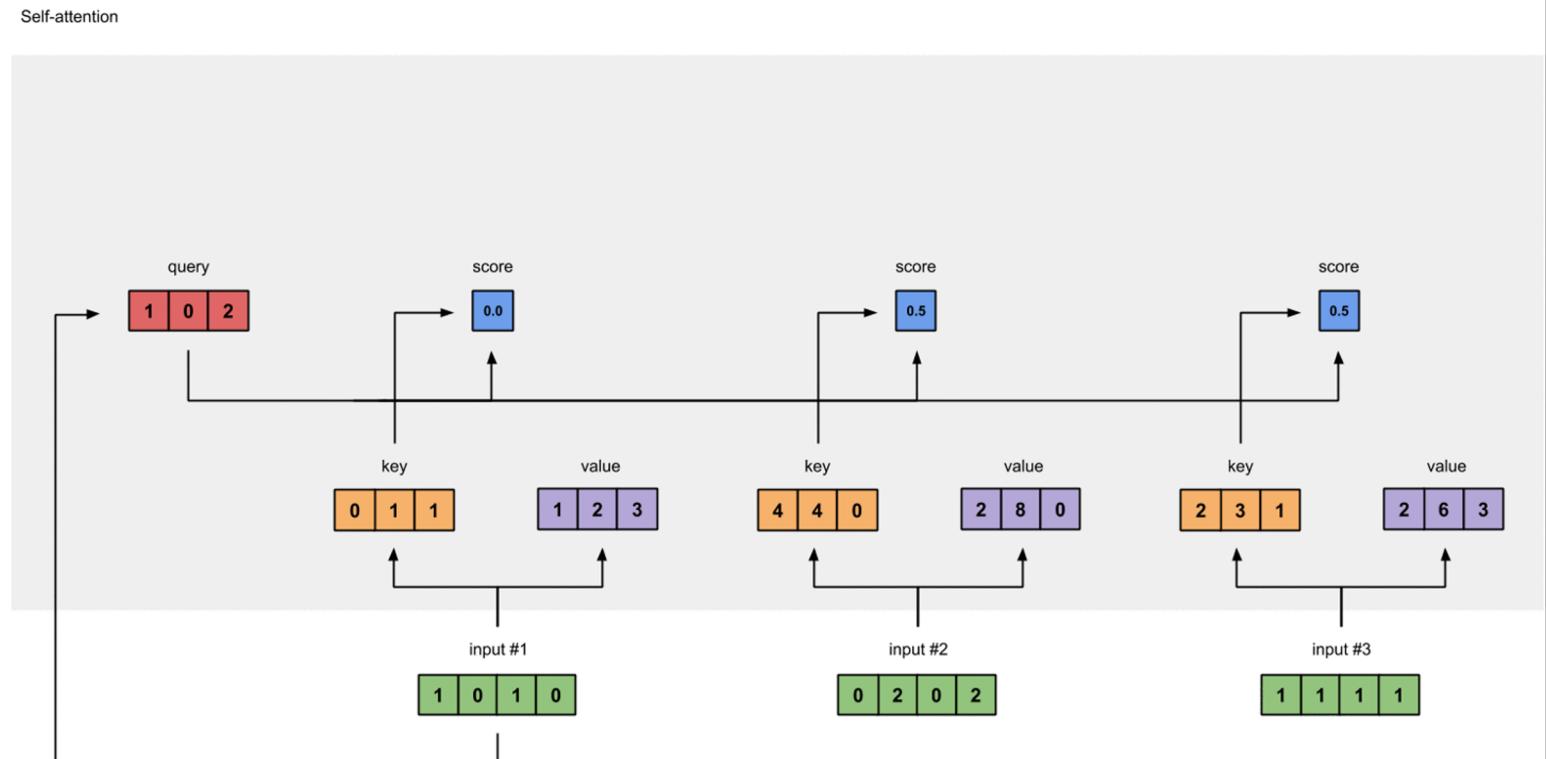
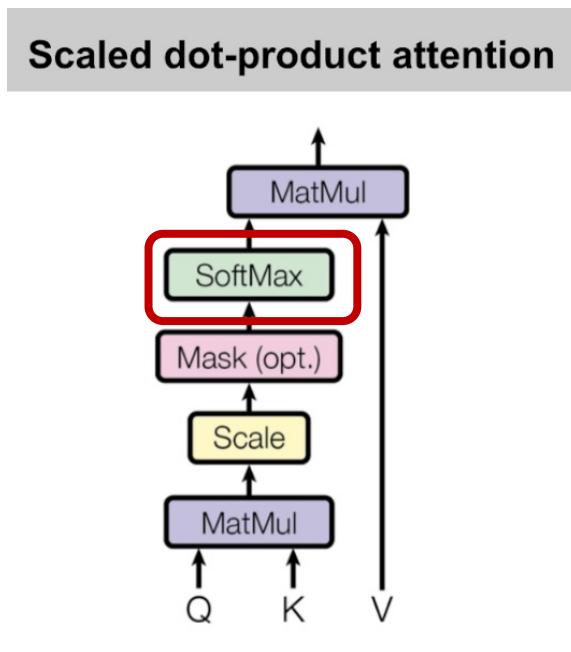


«Attention in action»

Calculate softmax

Softmax «simply» maps **attention scores** to a «probability» in $[0, 1]$

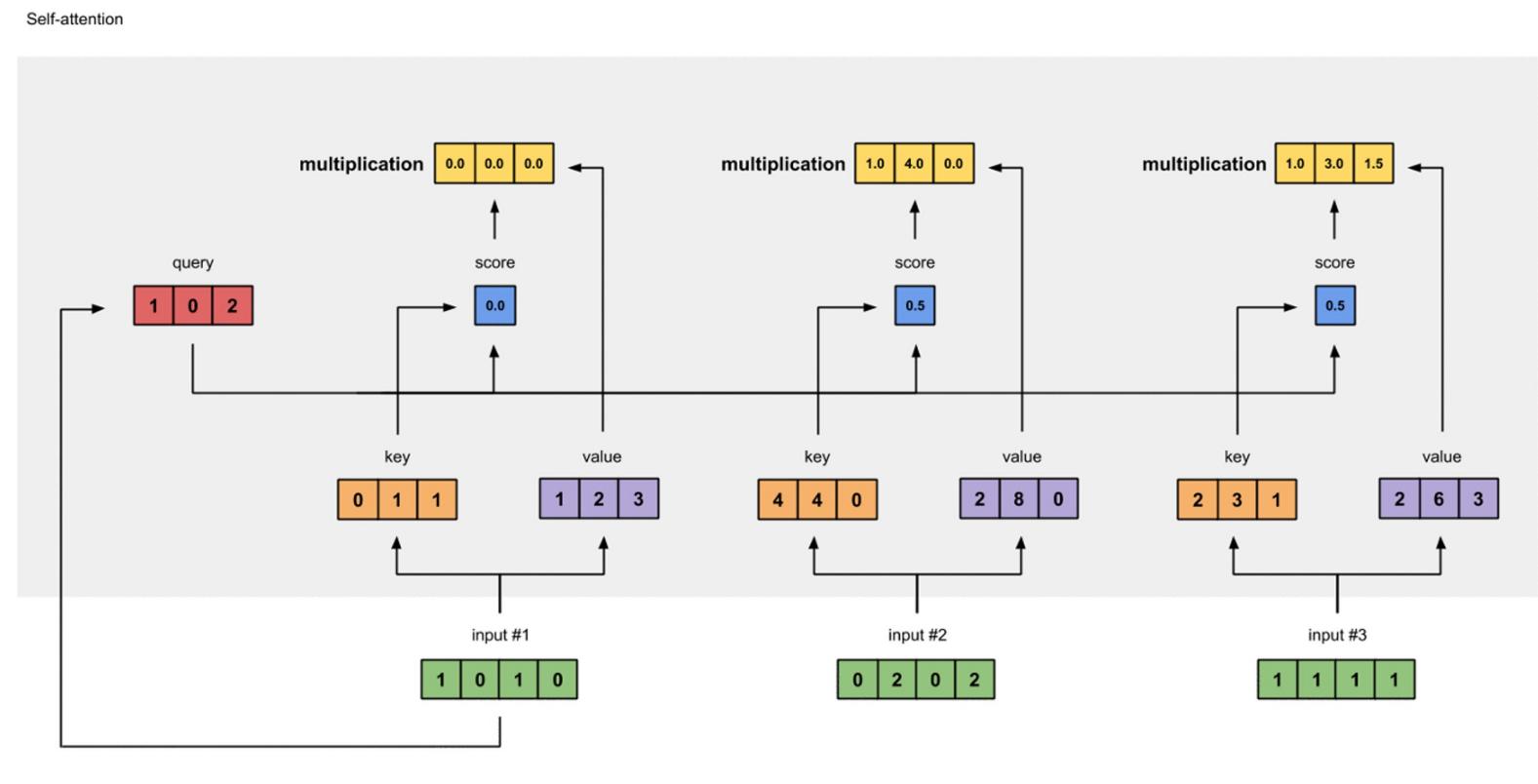
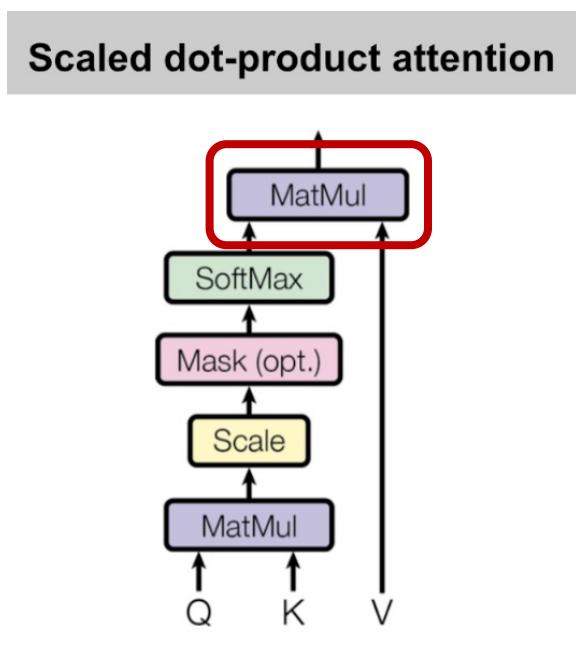
- The masked elements (i.e., with $-\infty$ gets a score near to 0)



«Attention in action»

Multiply scores with values

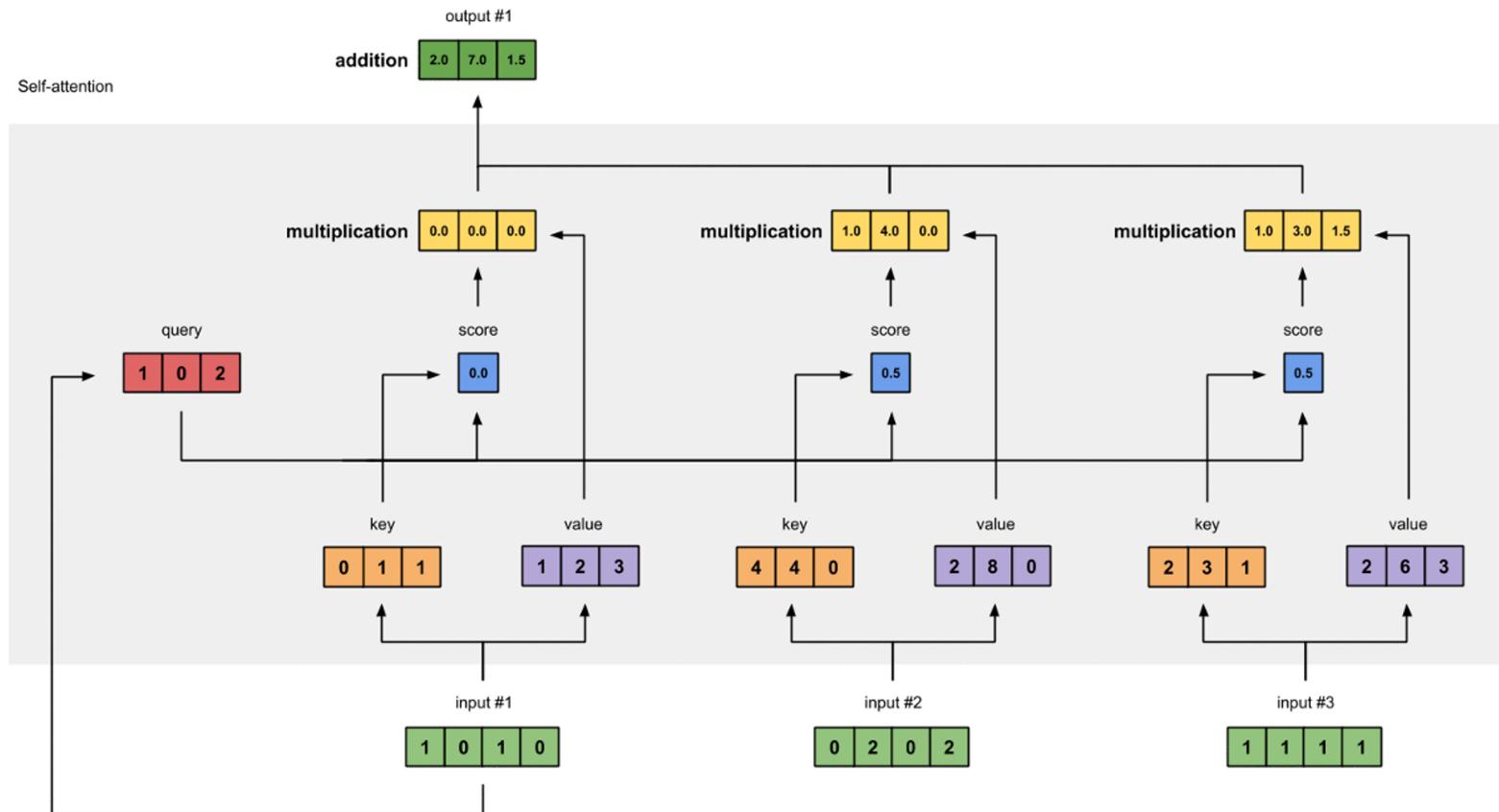
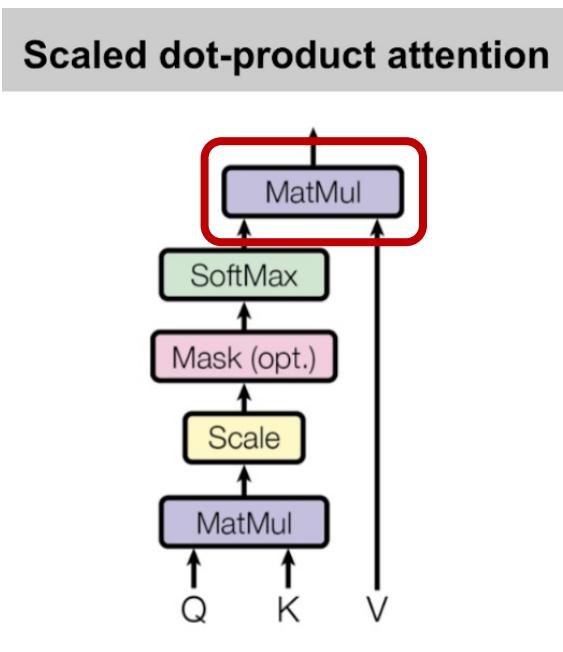
Each **input_n** (through each **value_n**) is weighted based of its importance in representing **input₁**



«Attention in action»

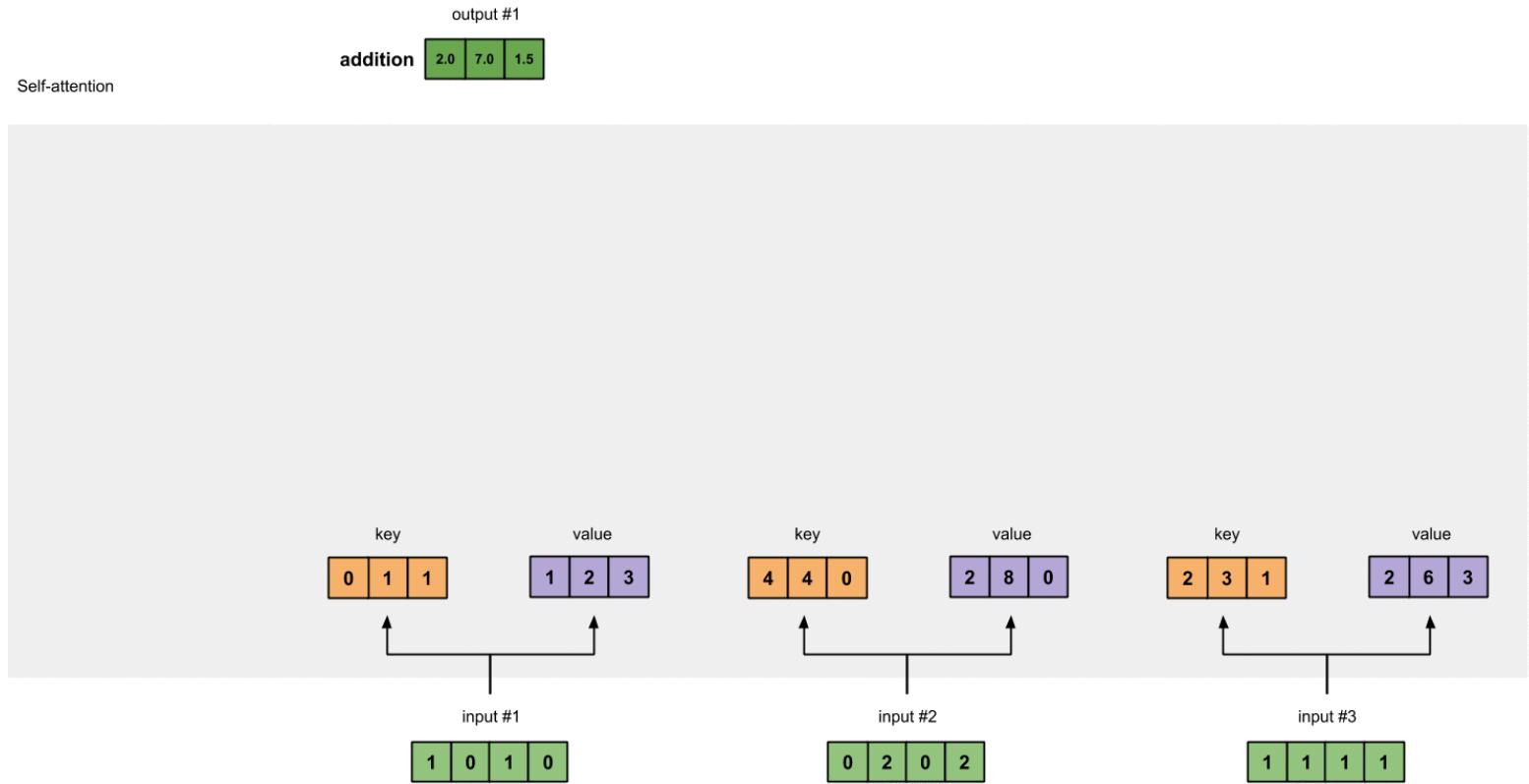
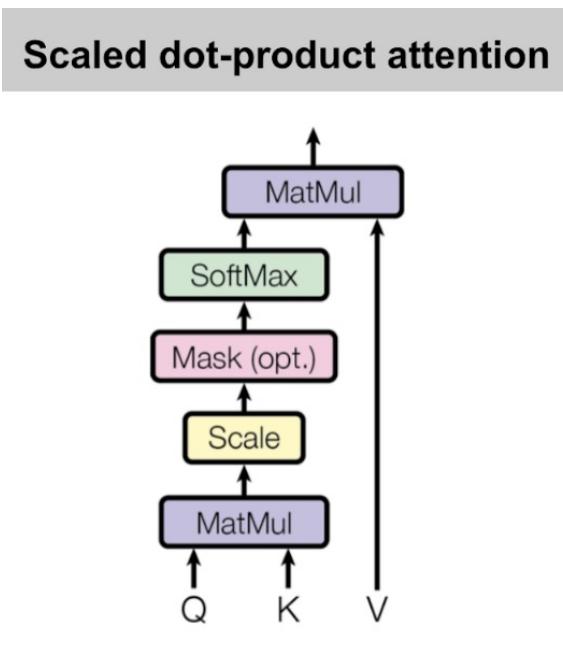
Complete the linear combination

Sum weighted values to get **output₁** that is the linear combination of all input elements (represented as **values**) weighted through the **attention scores**



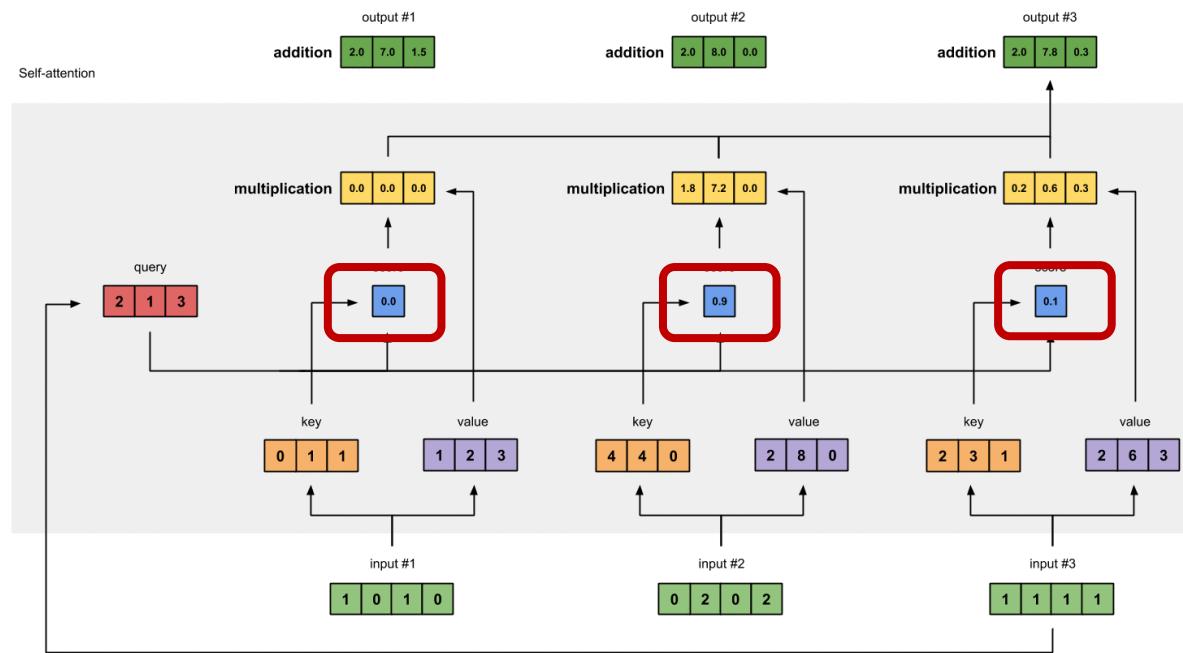
«Attention in action»

Repeat for **input₂** and **input₃**

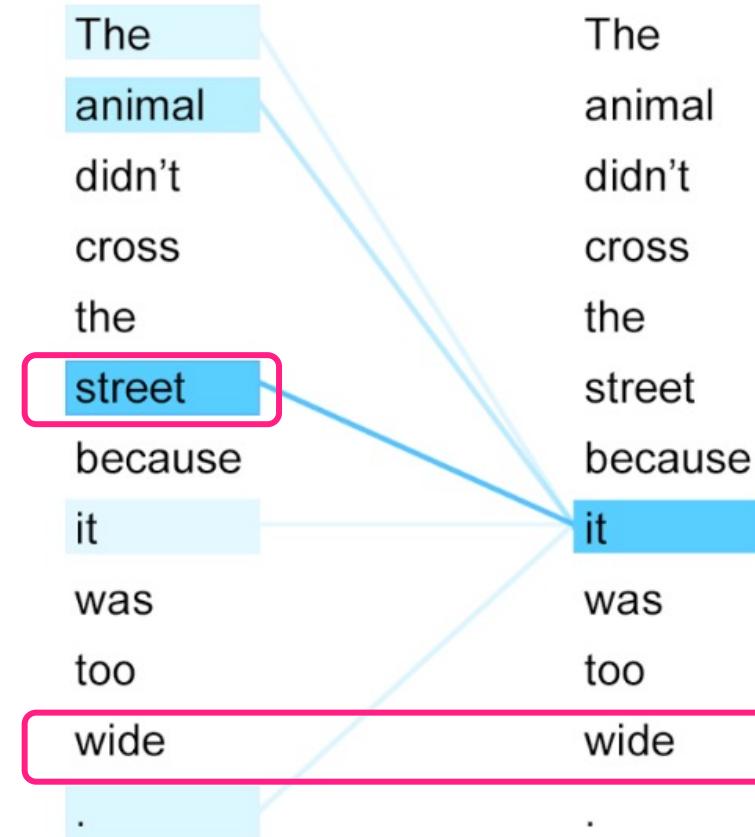
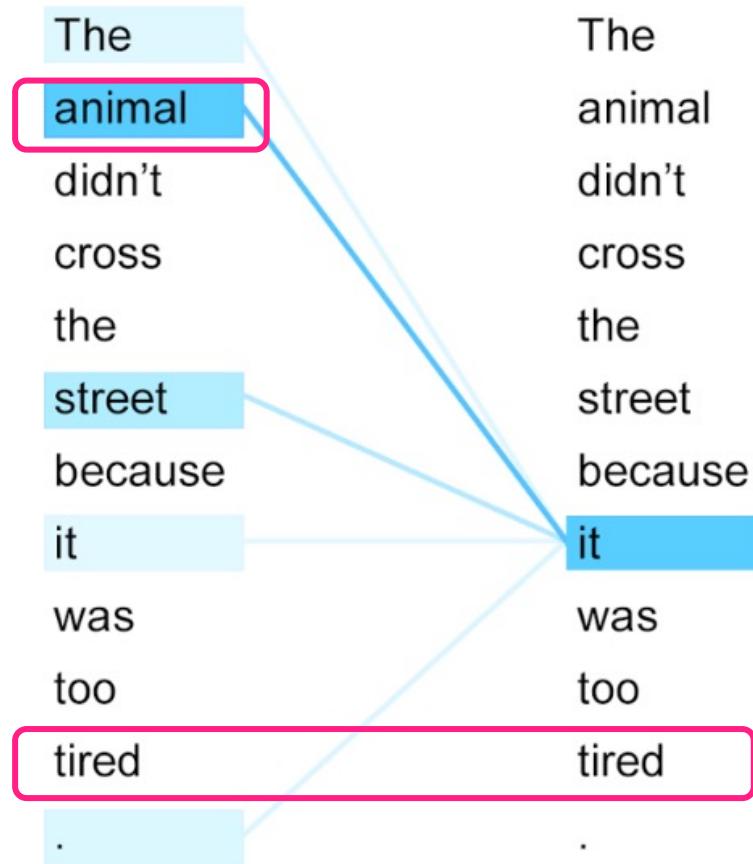


So... what is self-attention?

- It is not just a number, but a «probability distribution» for each symbol in input
 - And it allows weighting how **all words** are combined to generate the (hidden) representation of each word

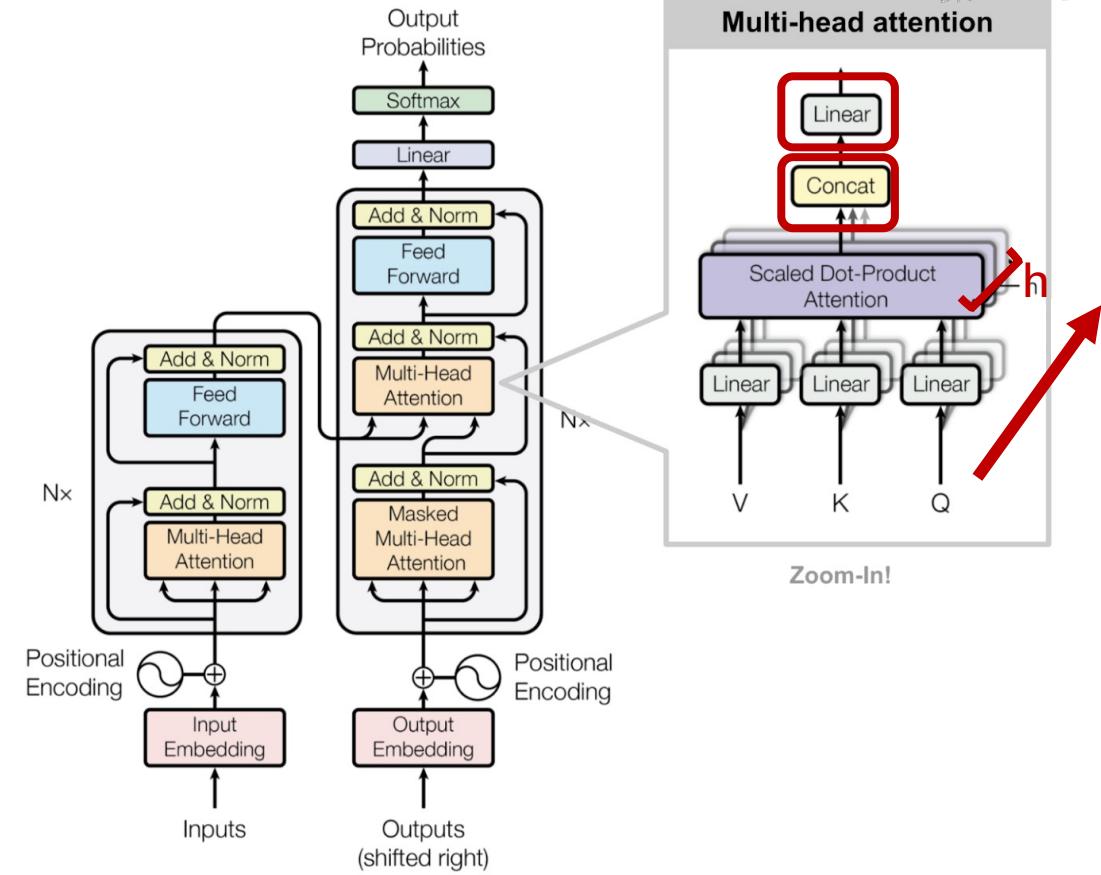


Self-Attention

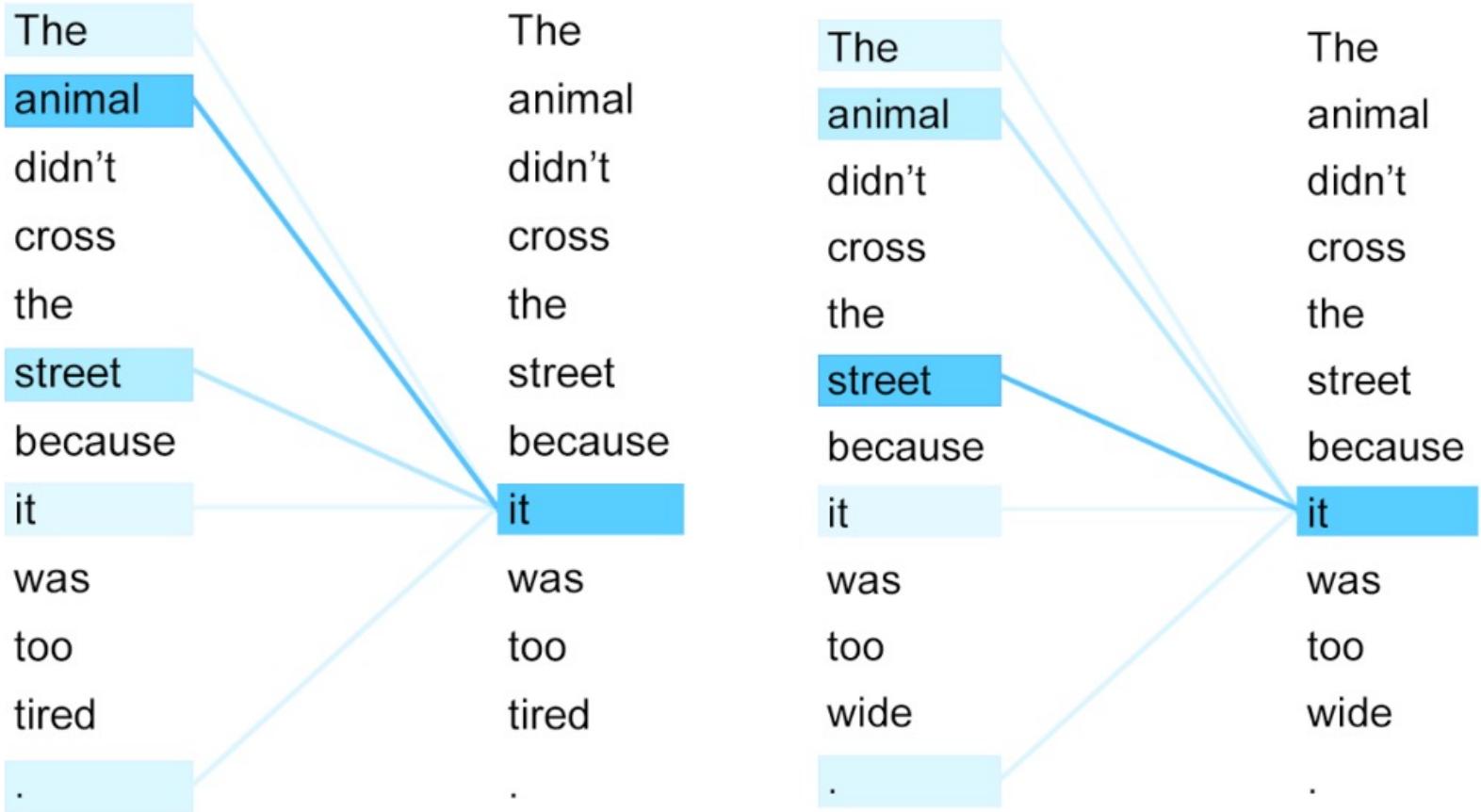


The Multi-Headed «Beast»

- Humans can attend to many things simultaneously.
- Can we extend attention to achieve the same?
- **Idea:** apply **Redundancy**, i.e., Scaled Dot-Product Attention **multiple times**
 - For each input, just generate h output
 - using h different different $(\mathbf{w}^Q, \mathbf{w}^K, \mathbf{w}^V)$
 - Concatenate the h output vectors of each input
 - Use a linear layer to “restore” the initial dimensionality
 - **But combining all multiple evidences**



From «simple» attention...



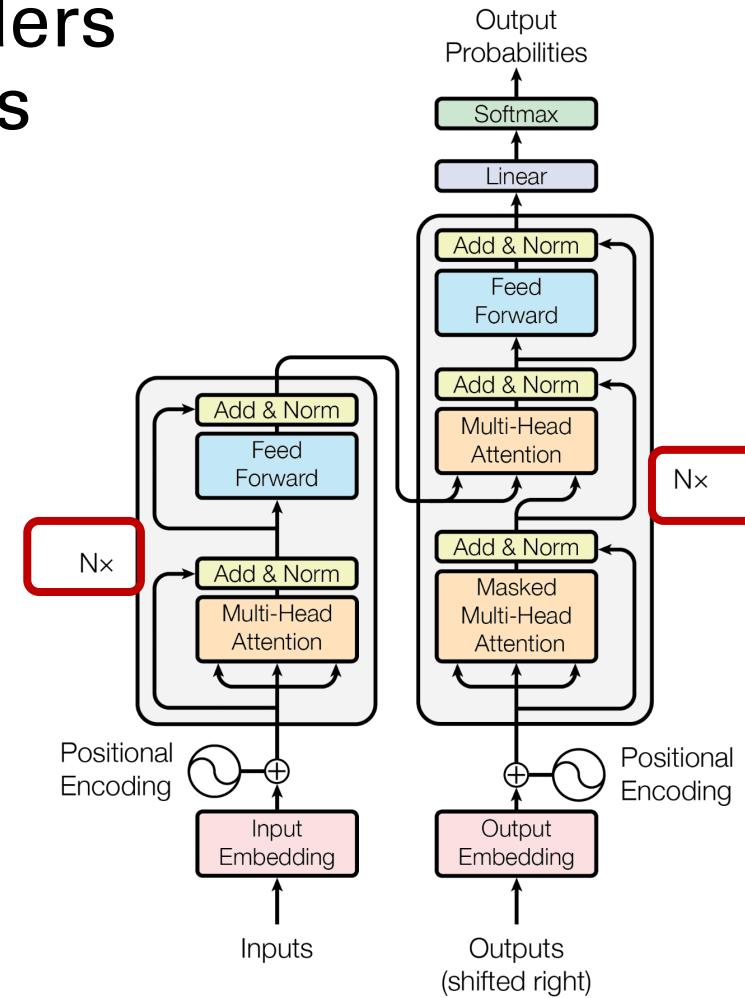
... to Multi-head Attention



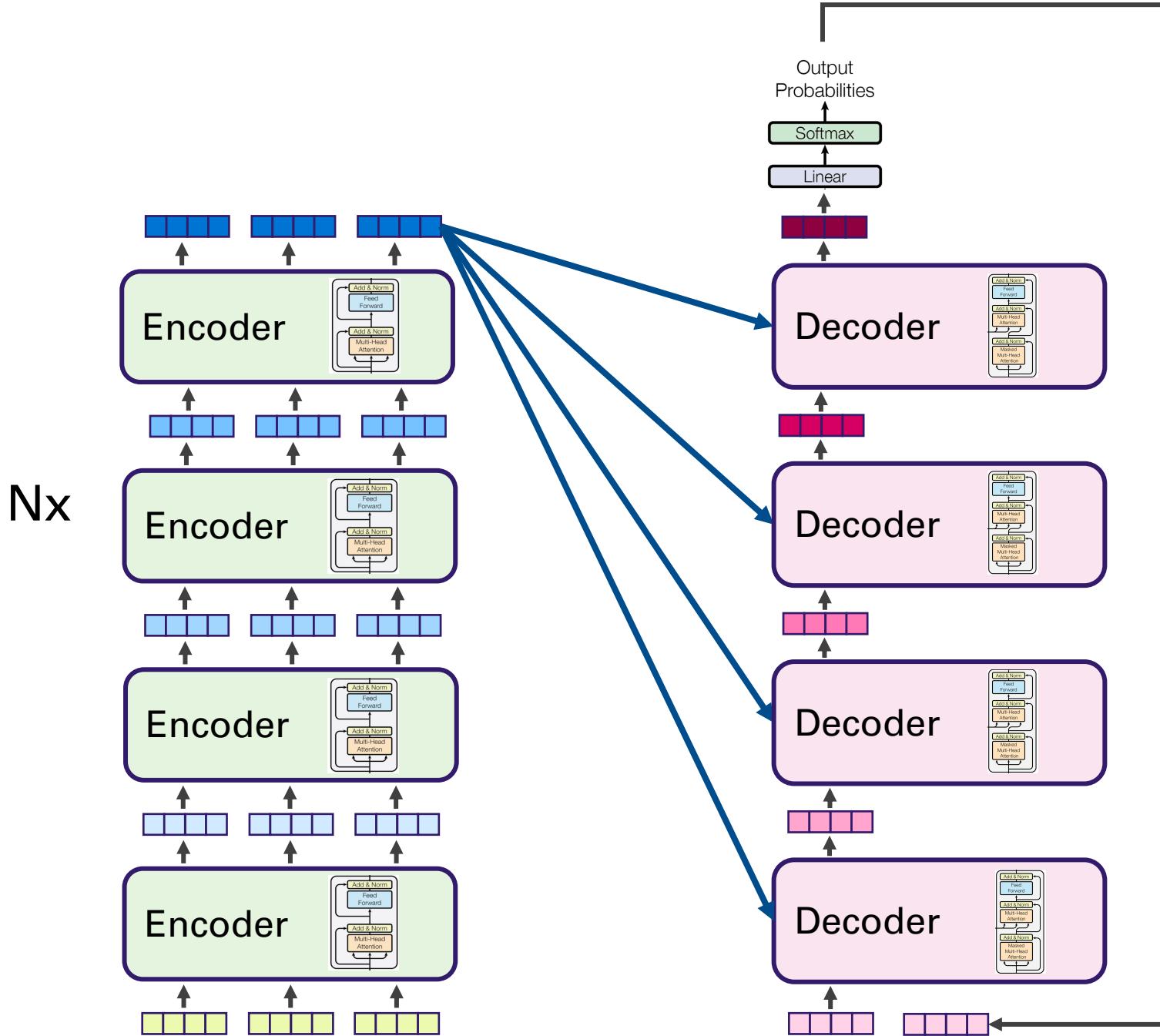
The figure displays a neural network architecture for processing text. The input layer consists of word embeddings for the sentence "The Law will never be perfect," followed by punctuation and padding tokens. The output layer consists of word embeddings for "The Law will never be perfect," "but its application should be just," "this is what we are missing," and "in my opinion." The hidden layer between them contains two neurons. The vector from the input embedding of "missing" points towards the bottom right, indicating it is associated with the second neuron in the hidden layer. Other words like "application" and "opinion" have vectors pointing upwards, indicating they are associated with the first neuron.

Where is the «Deep Learning»?

Encoders and decoders
are repeated N times

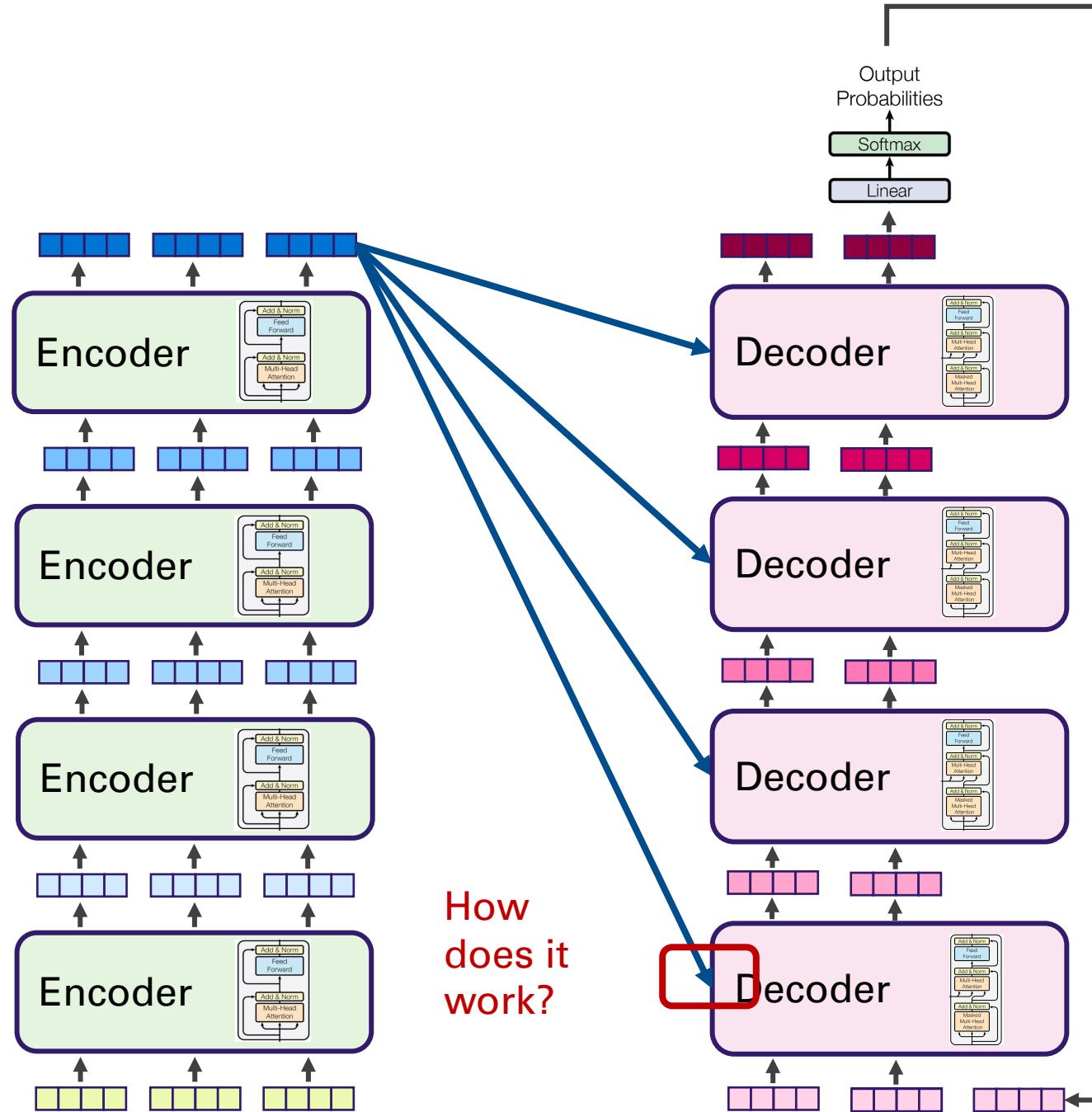


Again: the Transformer in Action



Generated hidden representations for each symbol initially rely on the first token, called `<start>`. These representations are influenced by all hidden representations from the encoder.

Again: the Transformer in Action



After the generation begins...

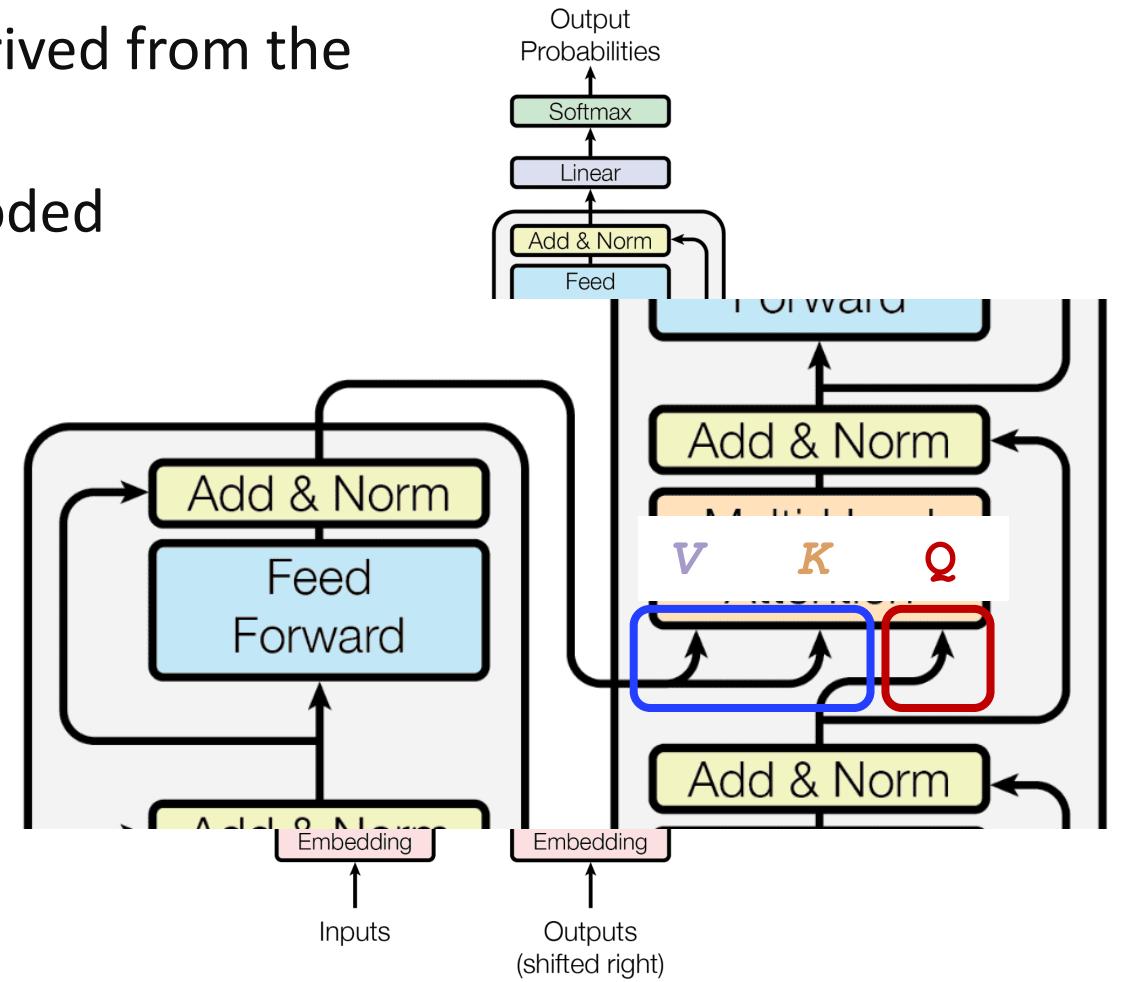
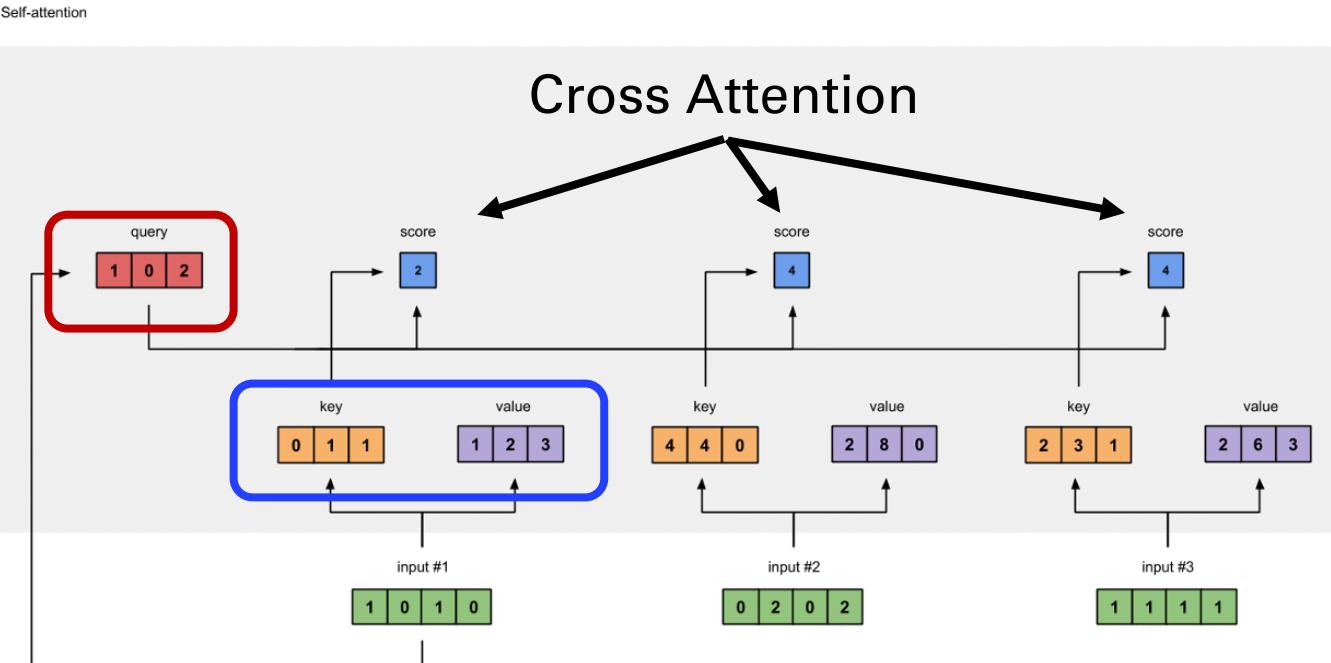
the decoder's hidden representations simultaneously depend on all input tokens attended to in the encoder...

but on the decoder's own previously generated hidden representations up to that point.



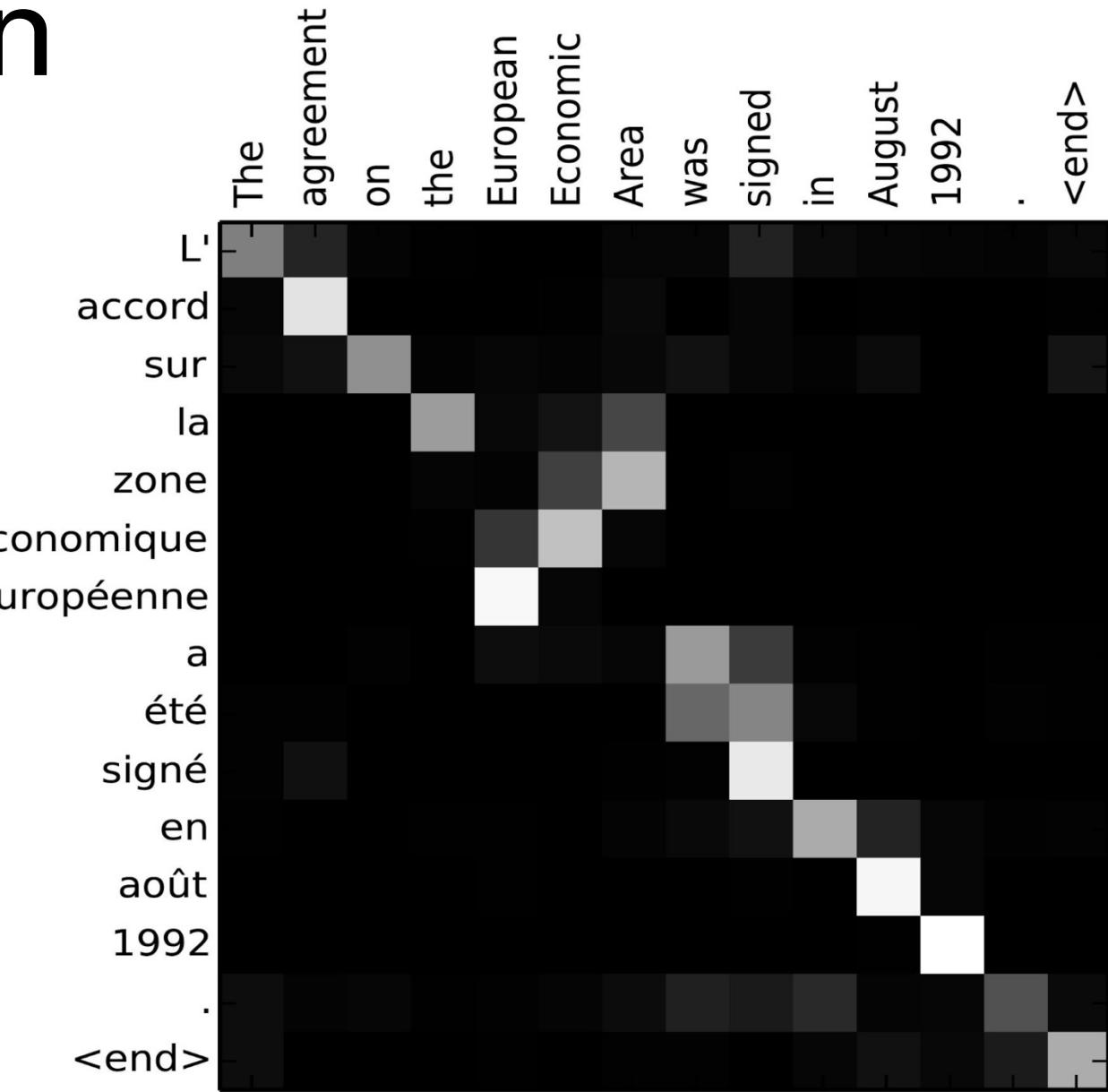
How to combine Encoders and Decoders?

- In the decoder, **key** and **value vector** are derived from the input.
- The **query**, in contrast, depends on the decoded sequence.



Cross-Attention

- Attention scores between input and output words
- White equals higher score
- The diagonal is highly correlated
- The scores reveal the grammatical difference for adjectives for the two languages (*zone économique européenne*)



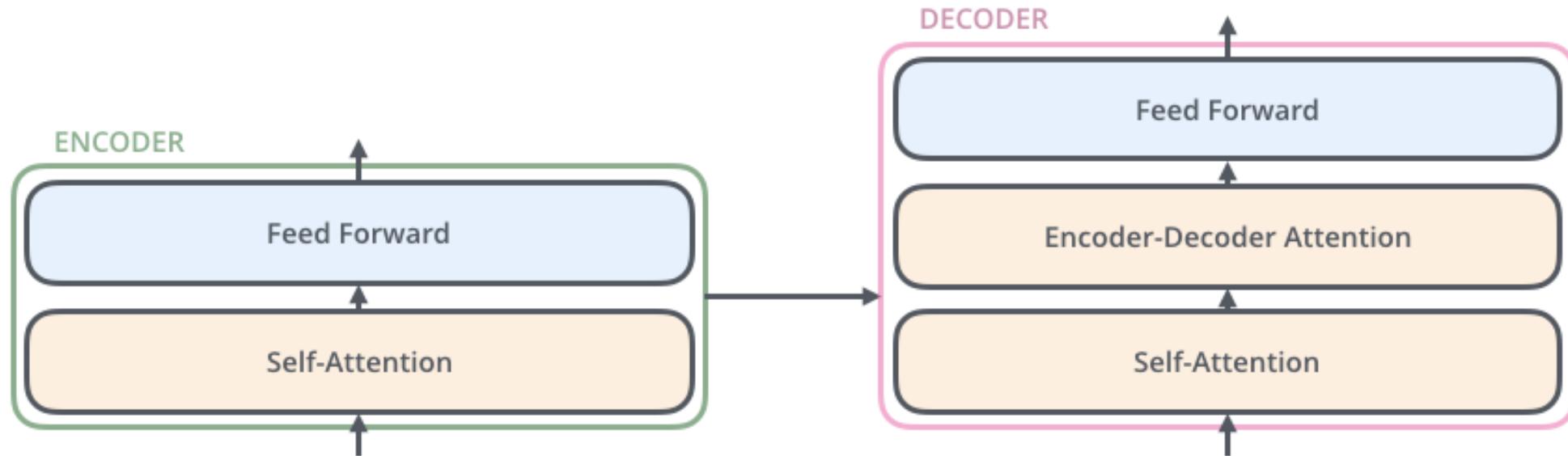
Advantages of Attention

- **Targeted Focus in Decoding:**
 - The decoder, with attention, can **strategically concentrate on relevant segments** of the source text
 - leading to more coherent and **contextually accurate** translations.
- **Addressing Vanishing Gradient Problem:** The mechanism offers a solution to the vanishing gradients issue
 - creating shortcuts between distant states in the sequence, facilitating smoother gradient flow during backpropagation.
- **Enhancing Model Interpretability:** we gain insights into what the model focuses on at each step

The Transformer was only the beginning

A transformer is made of two components

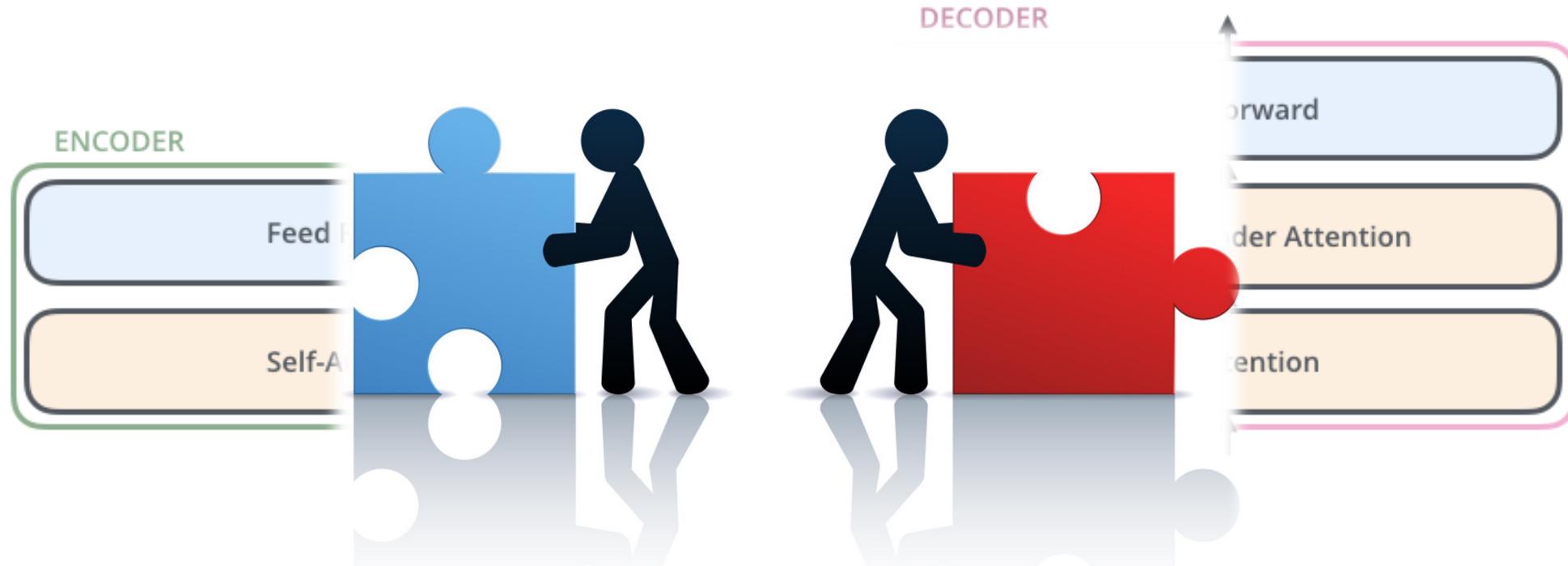
- Encoder
- Decoder



The Transformer was only the beginning

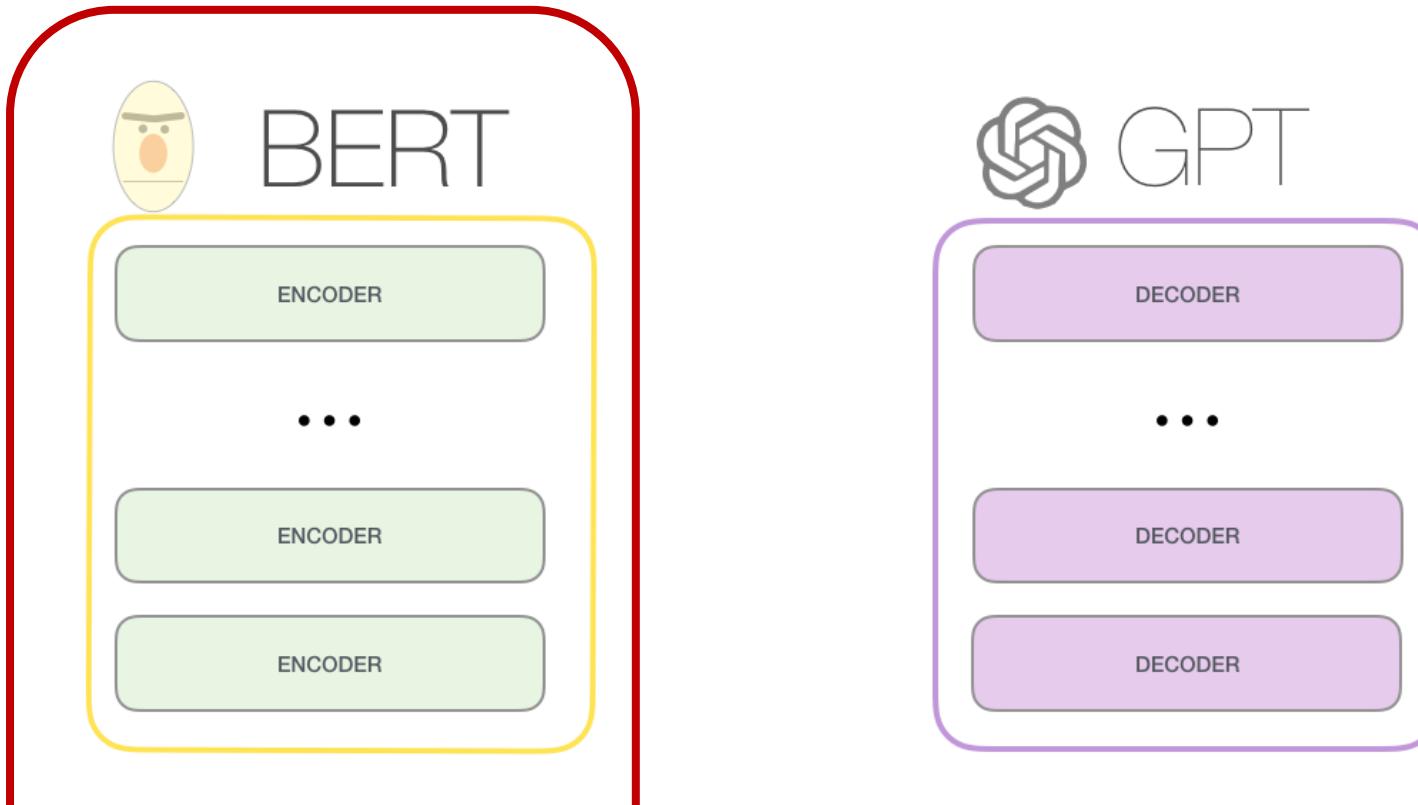
A transformer is made of two components

- Encoder
- Decoder



The transformer was only the beginning

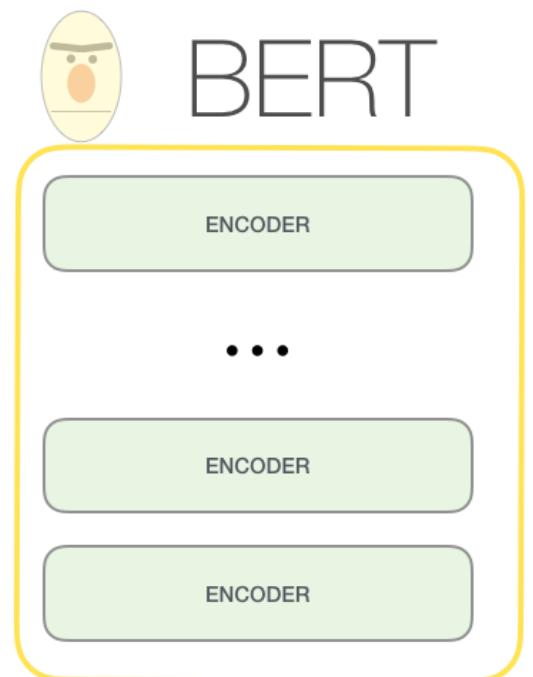
- This separation led to two «classes» of methods
 - **«Encoder-only» models:** the most famous one is BERT
 - **«Decoder–only» models:** the most famous one is GPT



BERT (Devlin et al, 2018)

Bidirectional Encoder Representations from Transformers

- Only the encoder is used
- Designed to generate **contextual meaningful representation** of input words
 - Representations are **context sensitive**, thanks to self-attention
 - Understand the context of a word in a sentence from **both left and right sides** (bidirectionally).
- Representations are embeddings
 - not suitable for text generation
 - ... but for many other tasks



Images from <https://jalammar.github.io/illustrated-bert/>

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.

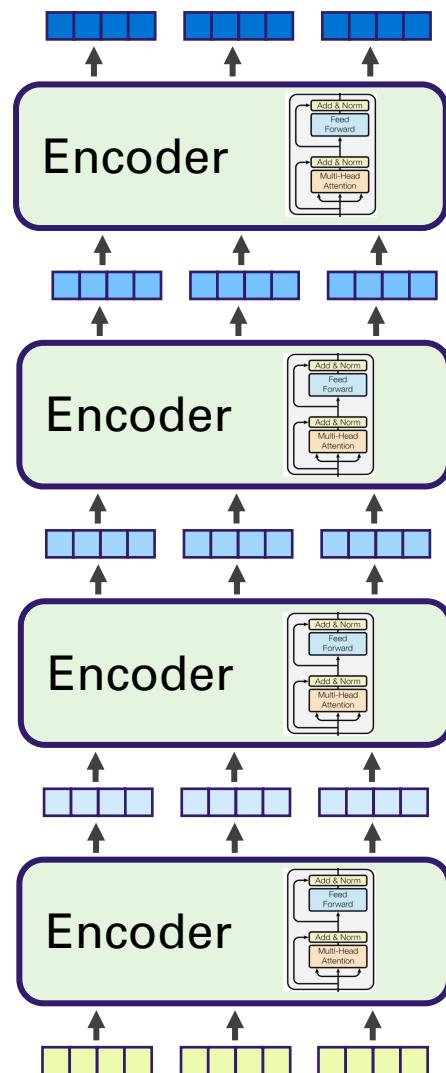
BERT (Devlin et al, 2018)

🤔 Why should it work?

- It is just a piece of the Transformer architecture seen a few slides ago.

💡 The GREAT IDEA: Pre-Training the encoder

- Pre-trained on a large corpus of text and then fine-tuned for specific tasks like question answering, sentiment analysis, etc.



Images from <https://jalammar.github.io/illustrated-bert/>

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.

No pre-training no party! The Revolution of Pre-Training in NLP

- **Simple idea:** train a (possibly large) model on a different task and re-use it on your task
 - circumventing the need for training from scratch
 - facilitating “quicker”, more effective deployment of the model
- **Precedent in Computer Vision:**
 - This strategy mirrors developments in computer vision
 - Architectures pre-trained on classification tasks using datasets like ImageNet
 - When applied on related task, these “starting point” achieve very good results
- **Addressing Overfitting in Large Models:**
 - With **increasing model sizes** and parameter counts, the **risk of overfitting grows**
 - Pre-training on vast datasets mitigates this by providing a broad learning base.

Towards Foundation Models

- **Emergence of Foundation Models in NLP:**
 - Large-scale models trained on linguistic tasks, forming a versatile base that can be fine-tuned for various specific applications.
- **Everybody worked on customizing Foundation Models:**
 - Leverage the extensive knowledge encapsulated in Foundation Models by fine-tuning them for particular NLP tasks.
- If you are interested in foundation models
 - [Zhou et al, 2023] A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT
 - <https://arxiv.org/abs/2302.09419>

Pretraining BERT

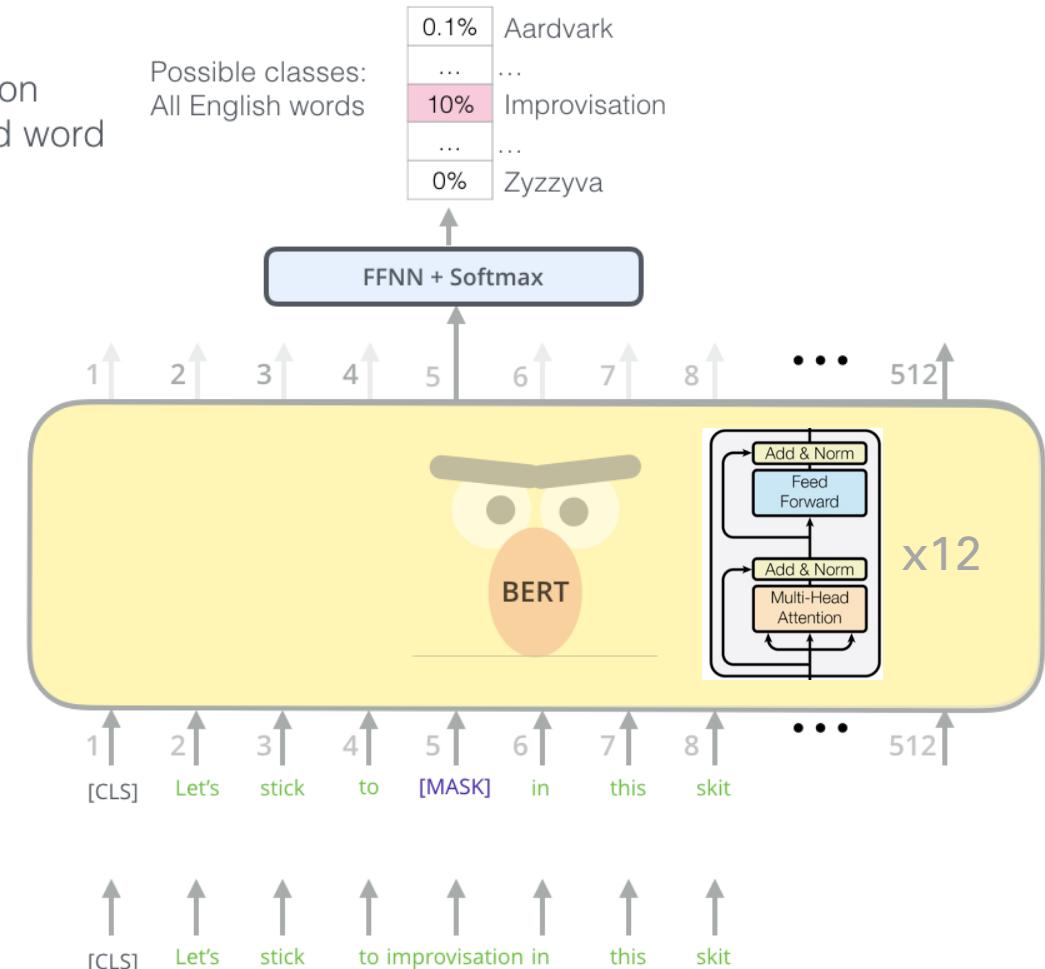
- BERT takes a sequence of tokens as input
 - Utilizes **self-attention across layers** to generate **context-aware** representations of each token in the sequence.
 - In each layer, $h=12$ $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V$ matrices
- **Pre-training tasks:**
 - **Masked-language modeling**



Randomly mask
15% of tokens

Input

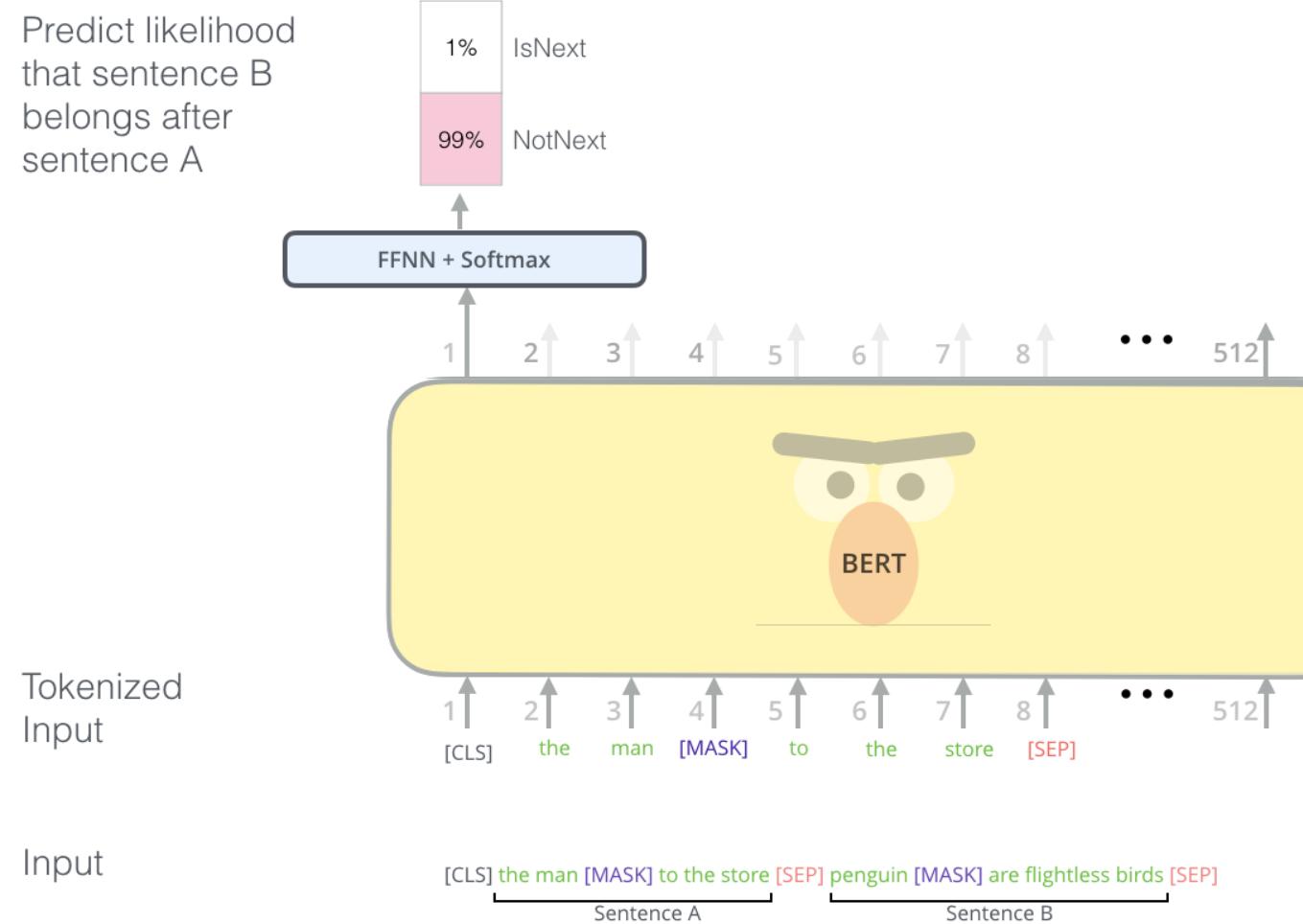
Use the output of the masked word's position to predict the masked word



Pretraining BERT (2)

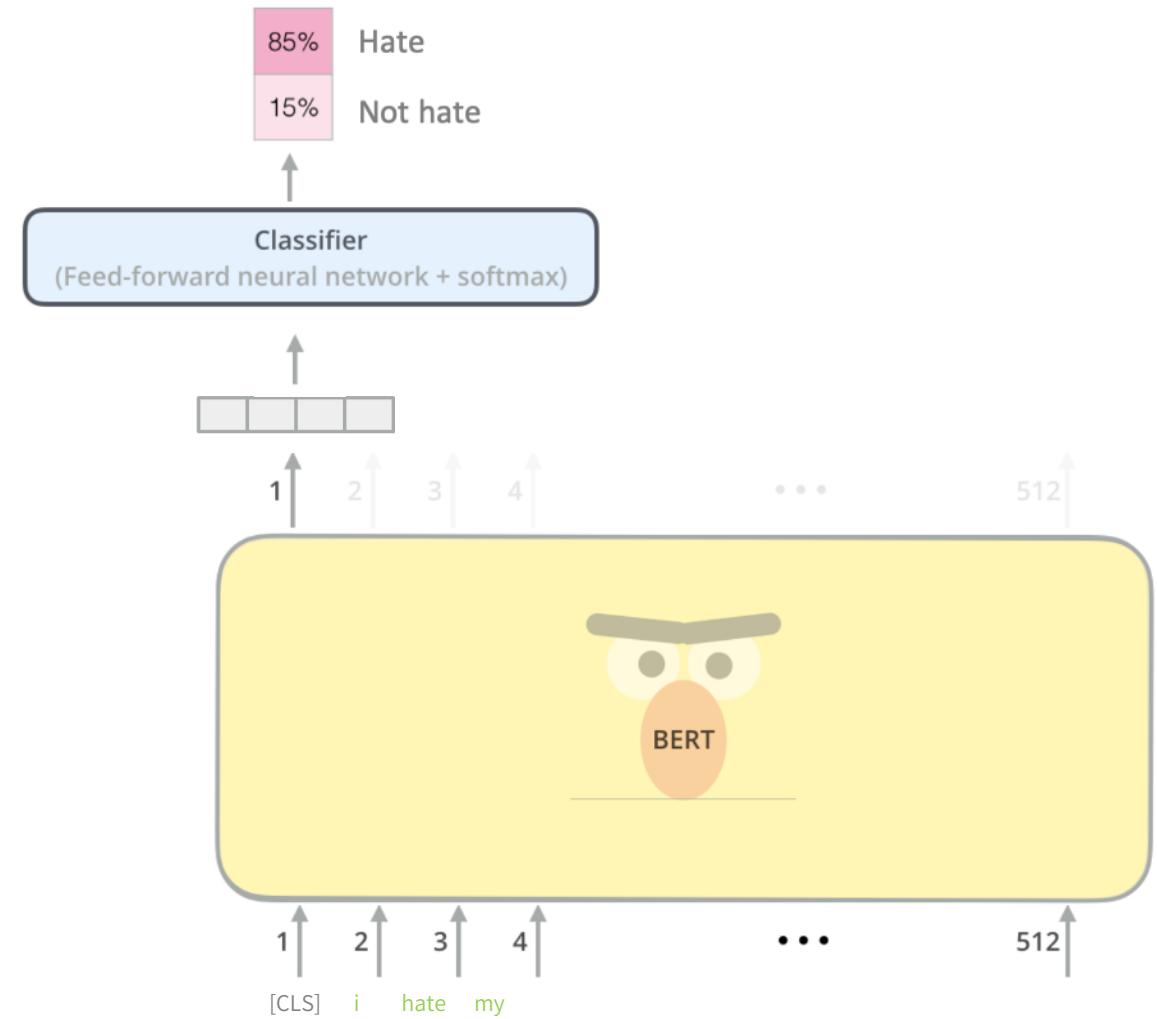
- BERT takes a sequence of tokens as input
 - Utilizes **self-attention across layers** to generate **context-aware** representations of each token in the sequence.
 - In each layer, $h=12$ $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V$ matrices
- **Pre-training tasks:**
 - Masked-language modeling
 - **Next sentence prediction**

Pretrained using the Toronto BookCorpus (800M words) and English Wikipedia (2,500M words)



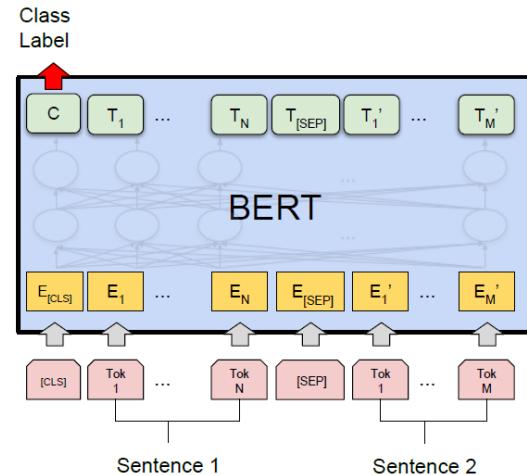
BERT and fine-tuning

- Once pretrained, we can **apply it to new sentences**
- BERT will **produce encoded representations** for each input symbol
- And it can be used in **different classification tasks**, just adding a new (linear) classifier...
- ... through fine-tuning of the entire architecture
- not trivial to forget what learned during the pre-training

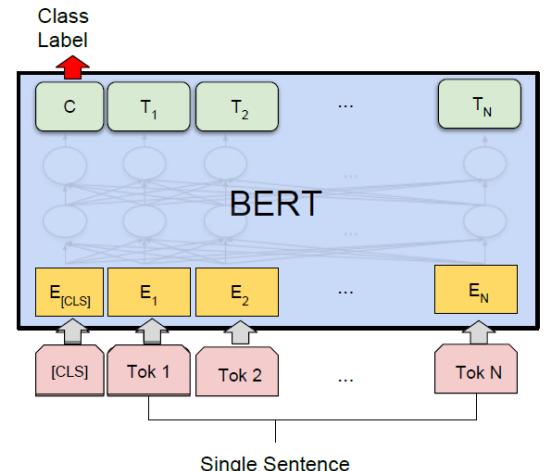


BERT in action

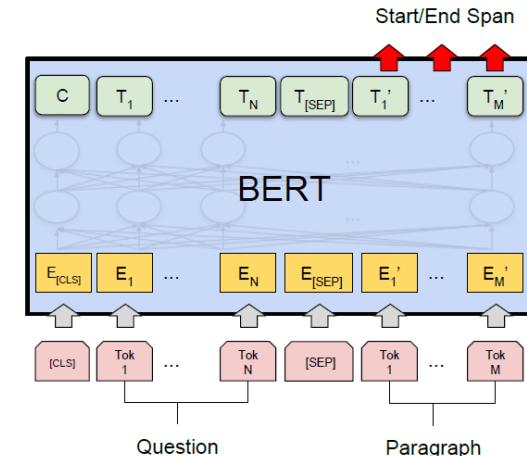
- The final layer outputs hidden representations, which can be utilized with a simple linear classifier
- to address a broad spectrum of NLP tasks efficiently.



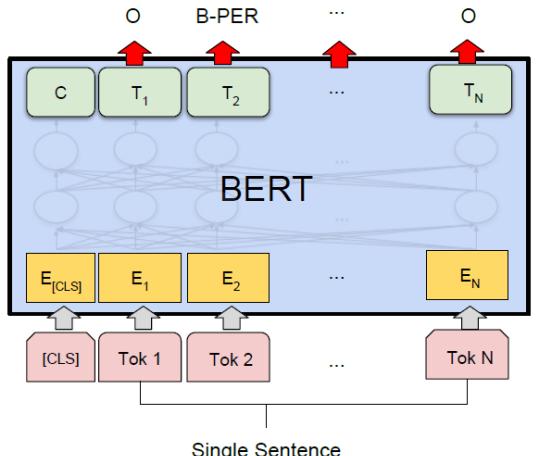
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



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



(c) Question Answering Tasks:
SQuAD v1.1



(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER



HUGGING FACE

BERT and «sons»

- A wide array of architectures, modeled and pre-trained, draw inspiration from BERT, featuring subtle but impactful variations.
- **Selected Notable Variants:**
 - **RoBERTa**: Streamlines BERT's pre-training approach and leverages an expanded dataset for training.
 - **ELECTRA**: Introduces a unique twist by replacing tokens with similar ones, then focusing on identifying these alterations.
 - **ALBERT**: Implements layer-wise parameter sharing to enhance efficiency.
 - **XLM**: Extends RoBERTa's capabilities across multiple languages.
- **Italian Contributions:**
 - Italy's own spin-offs include Alberto, UMBERTO, and Gilberto, each adding unique flavors to the BERT family.

If you are interested...

<https://github.com/crux82/AILC-lectures2021-lab>

The screenshot shows a GitHub repository page for 'AILC-lectures2021-lab'. The repository is public and has 1 branch and 0 tags. It was created by 'crux82' on Jun 18, 2021, with 6 commits. The repository contains files: AILC_Lectures_2021_Training_BER..., LICENSE, README.md, and using-bert.jpg. The README.md file is open, displaying text about training a sentence classifier using BERT-based models in Pytorch and its compatibility with HuggingFace.

AILC-lectures2021-lab Public

main 1 branch 0 tags

Go to file Add file Code

crux82	Creata con Colaboratory	2a73f5f on Jun 18, 2021	6 commits
📄	AILC_Lectures_2021_Training_BER...	Creata con Colaboratory	2 years ago
📄	LICENSE	Initial commit	2 years ago
📄	README.md	Update README.md	2 years ago
📄	using-bert.jpg	Add files via upload	2 years ago

README.md

Lectures on Computational Linguistics 2021

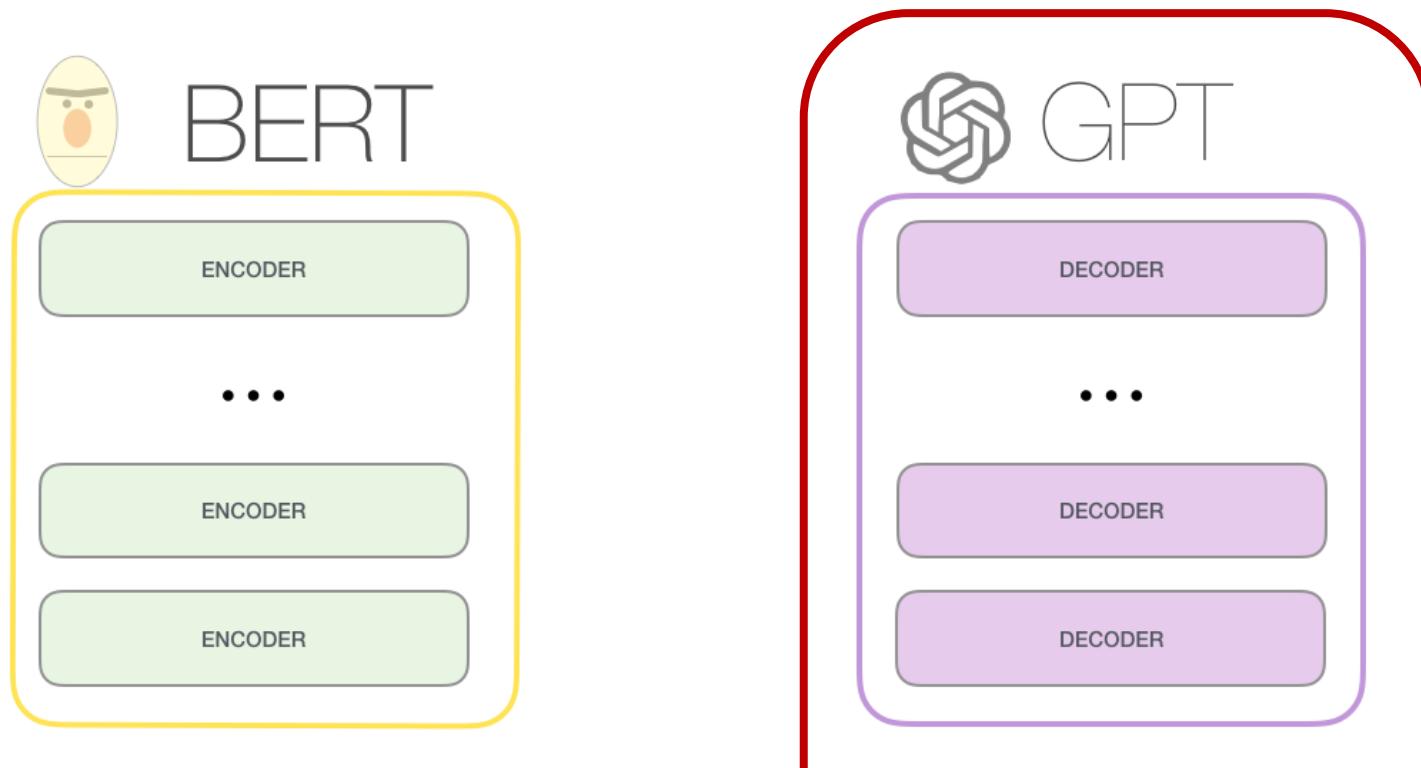
Tranining a sentence classifier using BERT-based models (in Pytorch and compatible with HuggingFace)

Author: Danilo Croce

This is a Pytorch (+ Huggingface transformers) implementation of a "simple" text classifier defined using BERT-

The transformer was only the beginning (2)

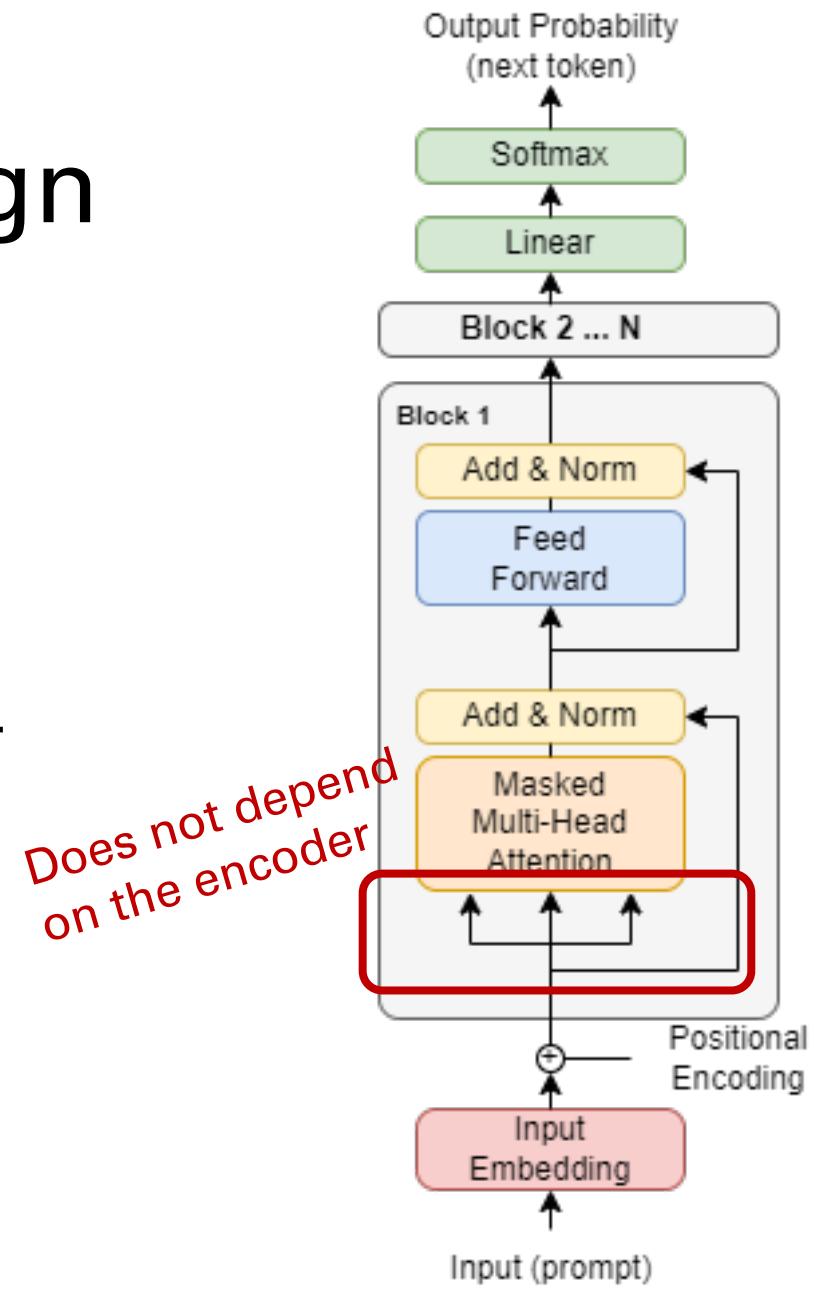
- This separation led to two «classes» of methods
 - **«Encoder-only»:** the most famous one is BERT
 - **«Decoder–only»:** the most famous one is GPT



The GPT Architecture and Its Decoder-Only Design

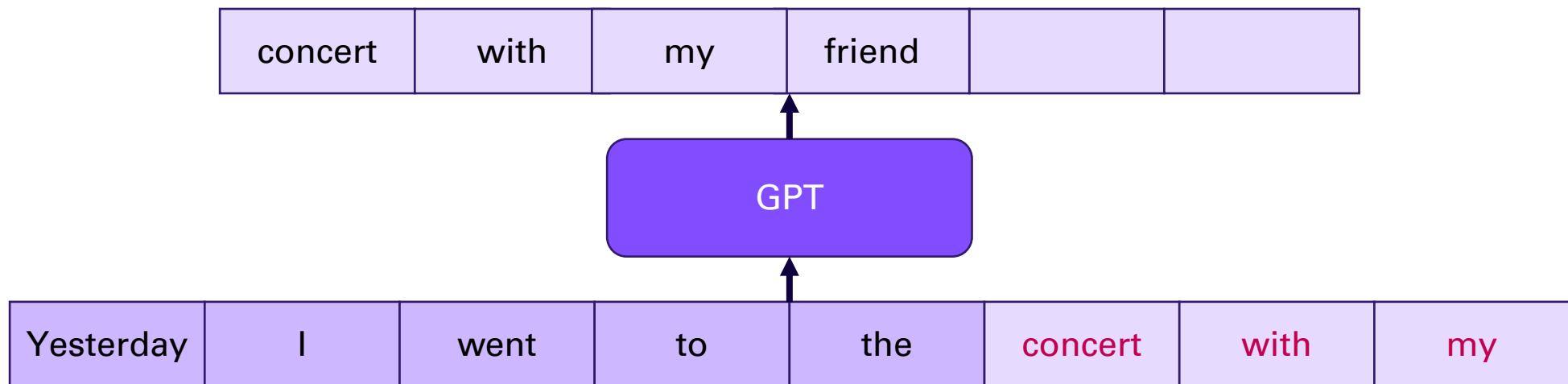
(Radford et al., 2018)

- **Decoder-Focused Architecture:**
 - GPT (Generative Pre-trained Transformer) is built on a decoder-only framework, exclusively using the decoder part of the original Transformer model.
- **Purpose of Decoder-Only Approach:**
 - to **generate meaningful text**, focusing on producing **coherent and contextually relevant** output sequences.



The task: Next Token Prediction

GPT is trained to **predict the next token in a sequence**, learning to generate text based on the preceding context.



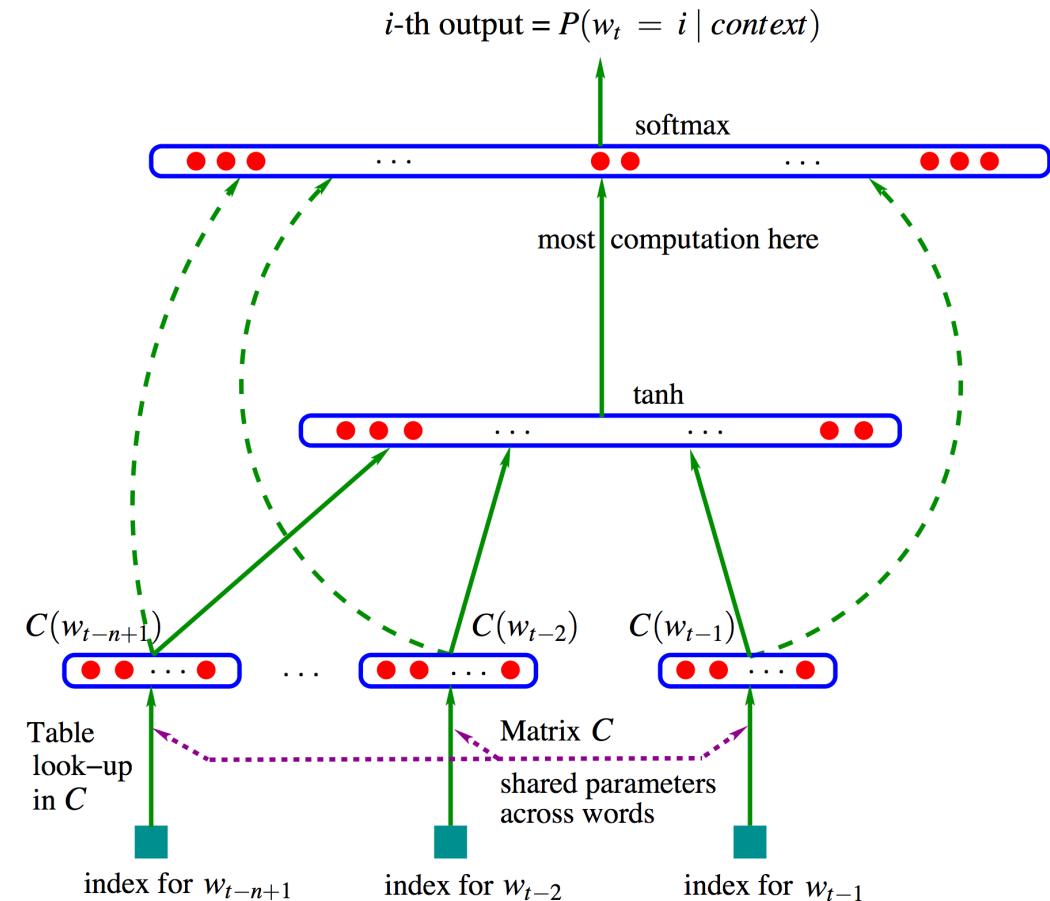
Language modeling is not a new task

- The objective is to learn a mapping

$$f(w_t, \dots, w_{t-n+1}) = P(w_t | w_1, \dots, w_{t-1})$$

- General Idea:**

- Associate with each word in the vocabulary an embedding
- Express the joint probability function of word sequences in terms of the feature vectors of these words in the sequence, and
- Learn simultaneously both notions:**
 - the word feature vectors as a matrix of lexical feature vectors and
 - the parameters that corresponds to the NN that estimate the probability function of the language model.



(Bengio et al, 2003): Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. [A neural probabilistic language model](#). J. Mach. Learn. Res. 3 (March 2003), 1137-1155.

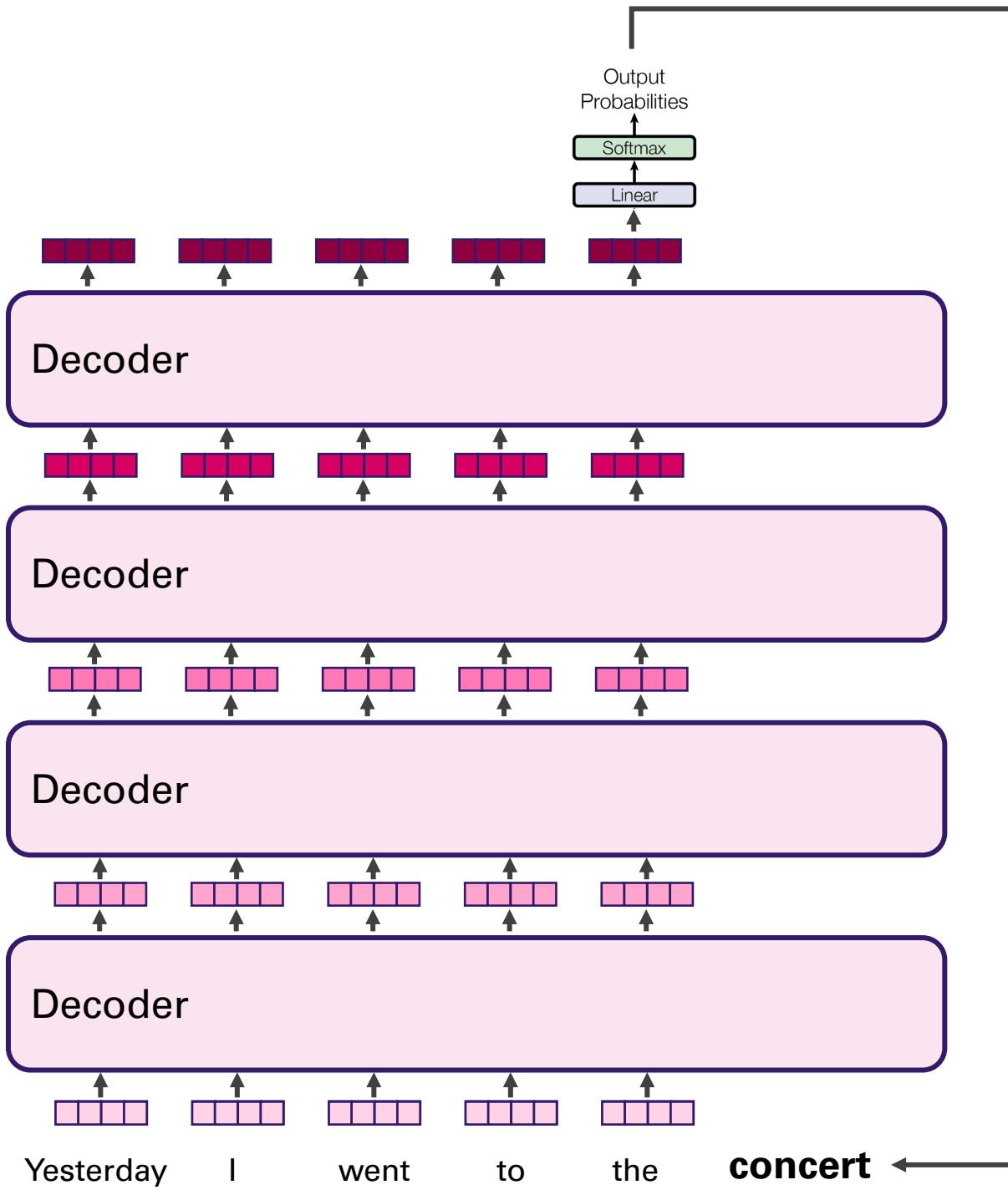
Believe it or not

In the beginning, GPT was not a stochastic parrot, but a complex sentence completion tool based on the multiplication of large matrices.



Generated using GPT4 (it seems it improved)

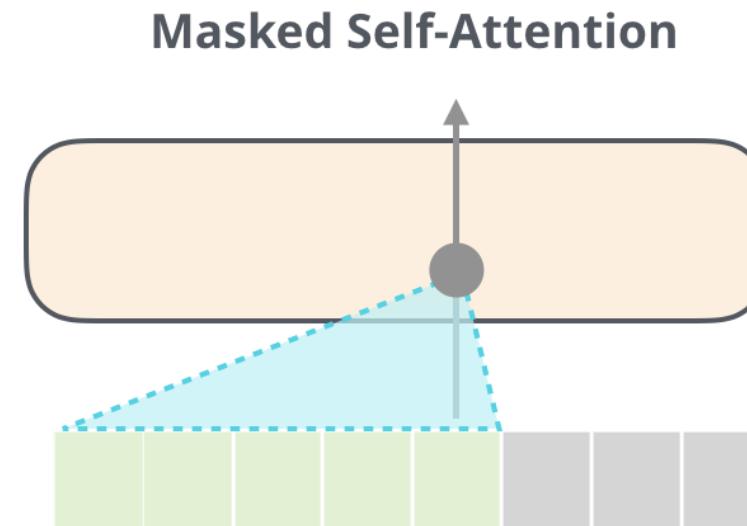
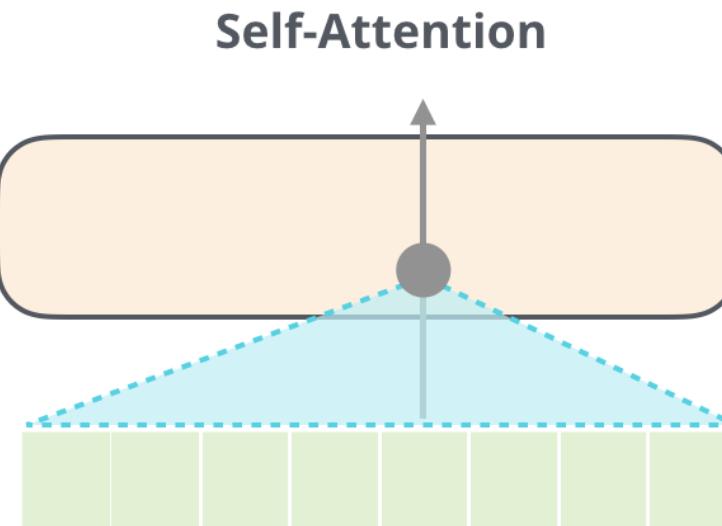
the «Pure» Decoder in Action



- It works similarly as in the Transformer
 - But query, value and key only depends on the input sequence
- Auto-regressive
 - Masked attention is crucial

Role of Masked Attention

Utilizes masked attention to prevent the model from accessing future tokens, ensuring that each prediction is based solely on past and current information.



What about results?

- At first, in classification tasks BERT seemed more promising, but in few years things changed

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
BERT _{BASE}	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT _{LARGE}	86.7/85.9	72.1	92.7	94.9	60.5	86.5	89.3	70.1	82.1

Table 1: GLUE Test results, scored by the evaluation server (<https://gluebenchmark.com/leaderboard>). The number below each task denotes the number of training examples. The “Average” column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.⁸ BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

Challenges in Fine-Tuning and Decoding in NLP Models

- **Complexity of Fine-Tuning:**
 - The intricacies of fine-tuning were not delved into deeply due to its complexity and the technical expertise required.
- **Resource Intensive Process:**
 - Fine-tuning is costly, necessitating multiple GPUs
 - Need of Frameworks for Scalable and Distributed Calculus
 - Extensive computational time, sometimes spanning 'years' of processing.

Challenges in Fine-Tuning and Decoding in NLP Models

- **Gradient Descent Management:**

- Careful oversight is needed to manage gradient descent, considering the vast amount of data and billions of adjustments involved.

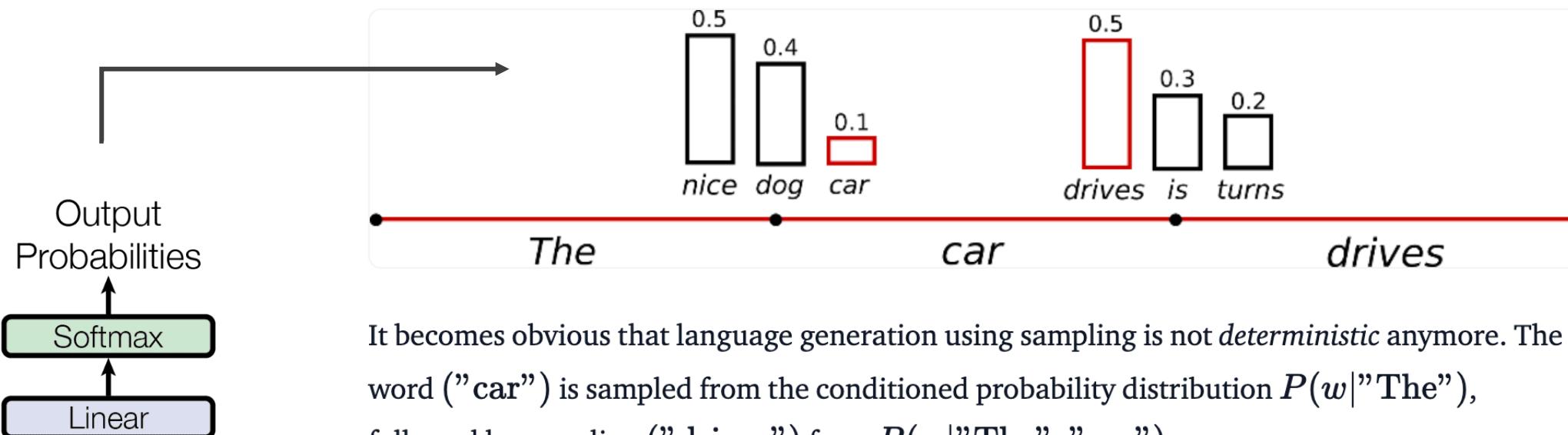
- **Decoding Process Simplified:**

- The decoding process has been simplified for this discussion.
- In reality, various techniques exploit the probability distribution over output symbols, e.g., such as
 - beam search
 - probabilistic methods (*any parrot was involved*)
- I strongly suggest to read:
 - <https://huggingface.co/blog/how-to-generate>

... why stochastic ...

- (also because) the most used decoding technique is based on **sampling**
 - In its most basic form, sampling means randomly picking the next word w_t according to its conditional probability distribution:

$$w_t \sim P(w|w_{1:t-1})$$



It becomes obvious that language generation using sampling is not *deterministic* anymore. The word ("car") is sampled from the conditioned probability distribution $P(w | \text{"The"})$, followed by sampling ("drives") from $P(w | \text{"The"}, \text{"car"})$.

But does GPT 'only' know how to predict the next word in a sentence?

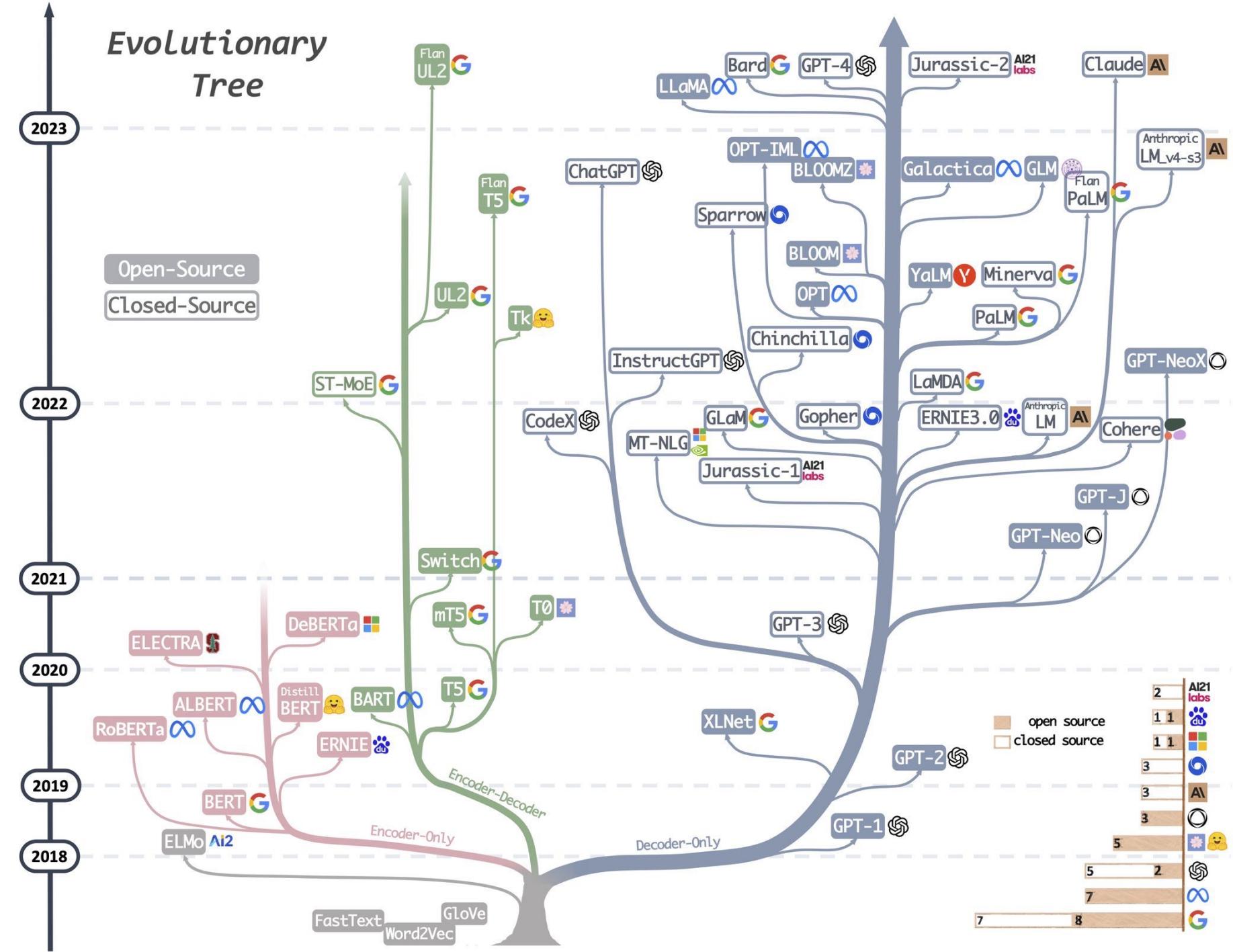
- If we are smart enough, we can use the generation capability of GPT to solve a task, but...
 - We can ask GPT to do something, e.g. write an article:

Title: United Methodists Agree to Historic Split

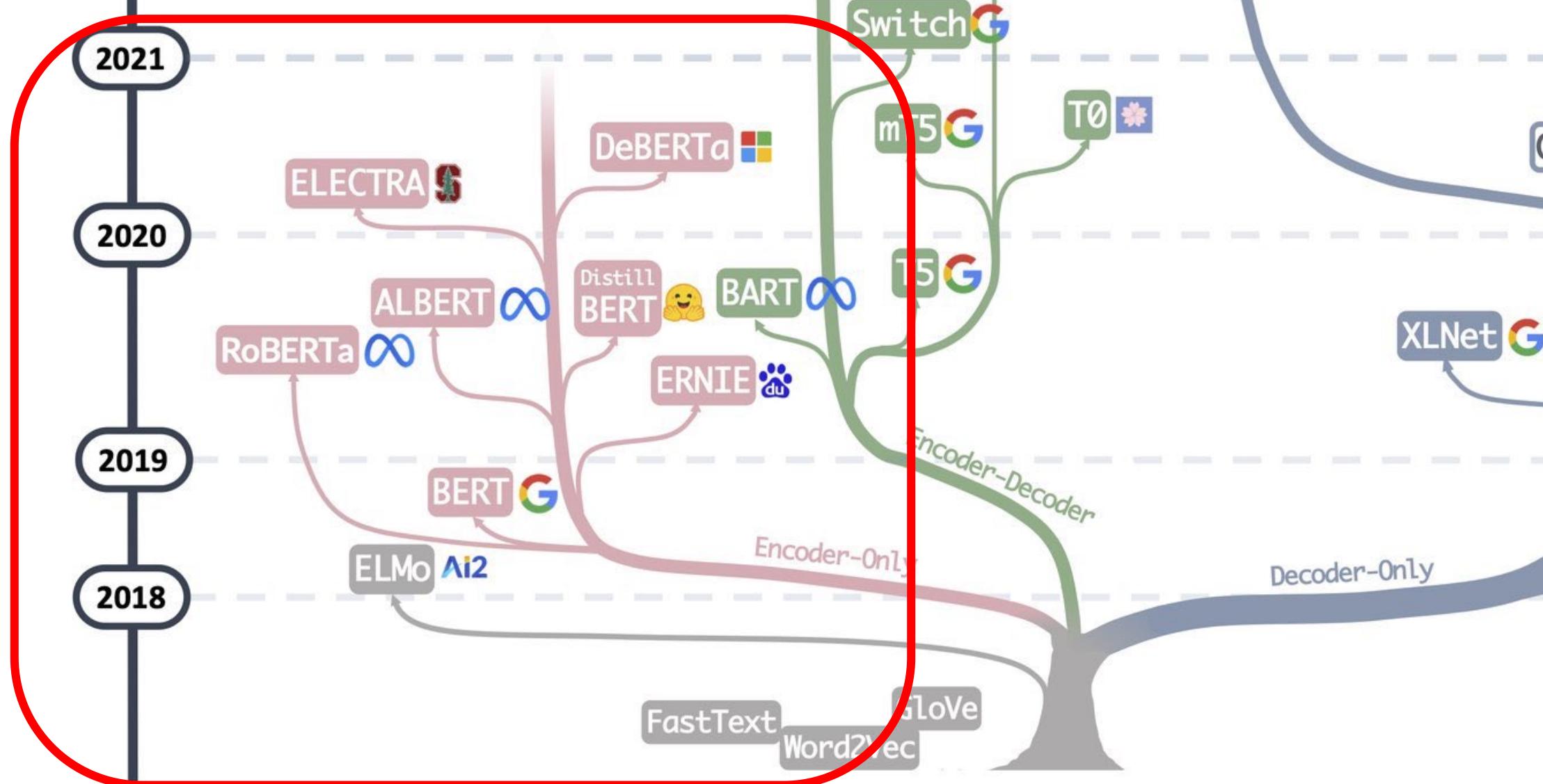
Subtitle: Those who oppose gay marriage will form their own denomination

Article: After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination. The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.," but that it has been shrinking in recent decades. The new split will be the second in the church's history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed

The rest is a family tree

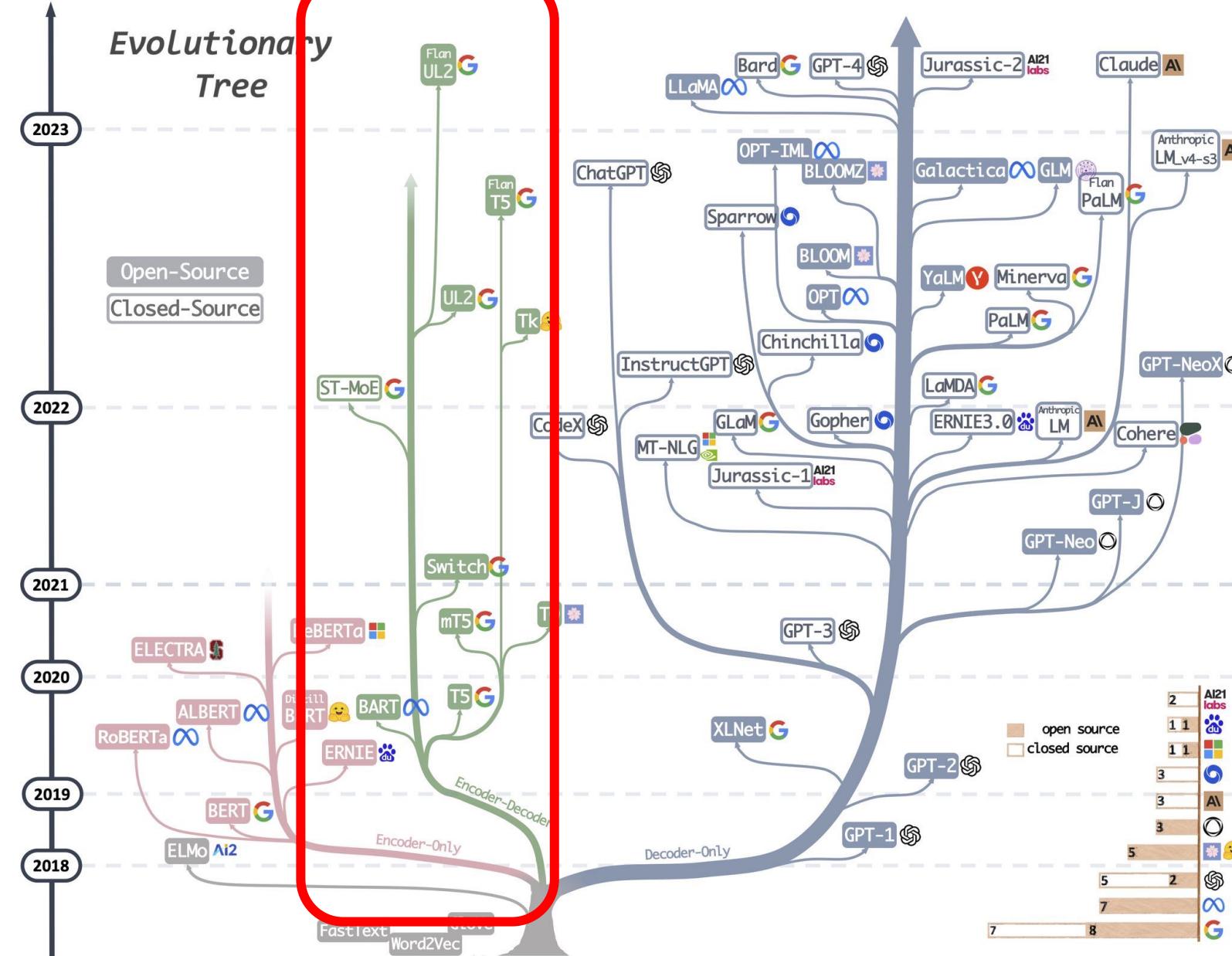


The «Encoder Only» family



Encoder-based architectures experienced **rapid initial growth** and enormous success until 2021, **after which interest shifted**.

The «Encoder/Decoder» family



Encoder-Decoder based architectures experienced a **more limited success but largely used**, especially tasks requiring generation

A digression

Introduction to Google's T5 Model

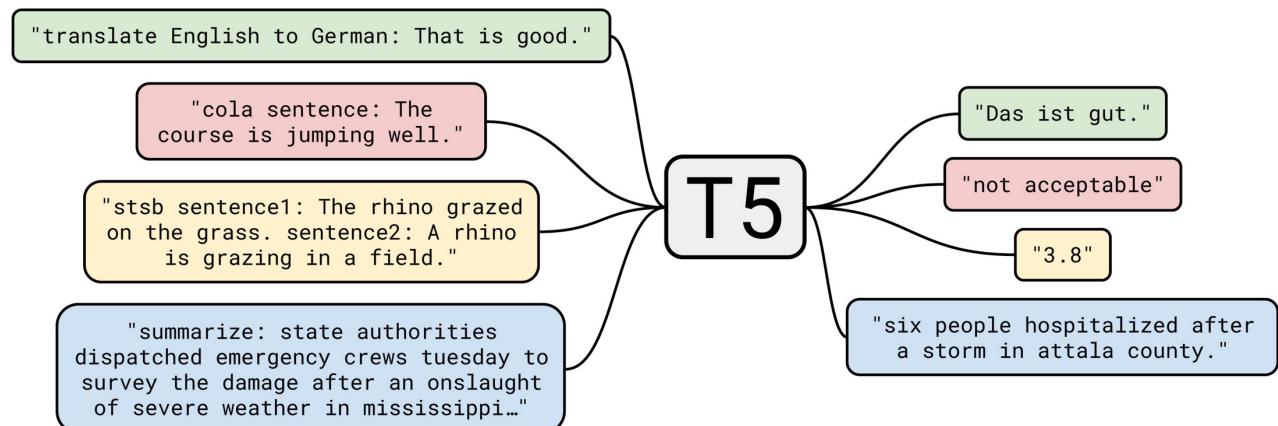
(Raffel et al, 2019)

- **Encoder-Decoder Structure:**

- T5, or Text-to-Text Transfer Transformer, is based on an encoder-decoder architecture.
- Trained on large-scale datasets like C4 (Colossal Clean Crawled Corpus) containing a diverse range of internet text.

- **Versatile Pre-Training:**

- Notably **pre-trained on multiple tasks**
- Tasks involve reconstructing corrupted versions of input strings, showcasing its adaptability.



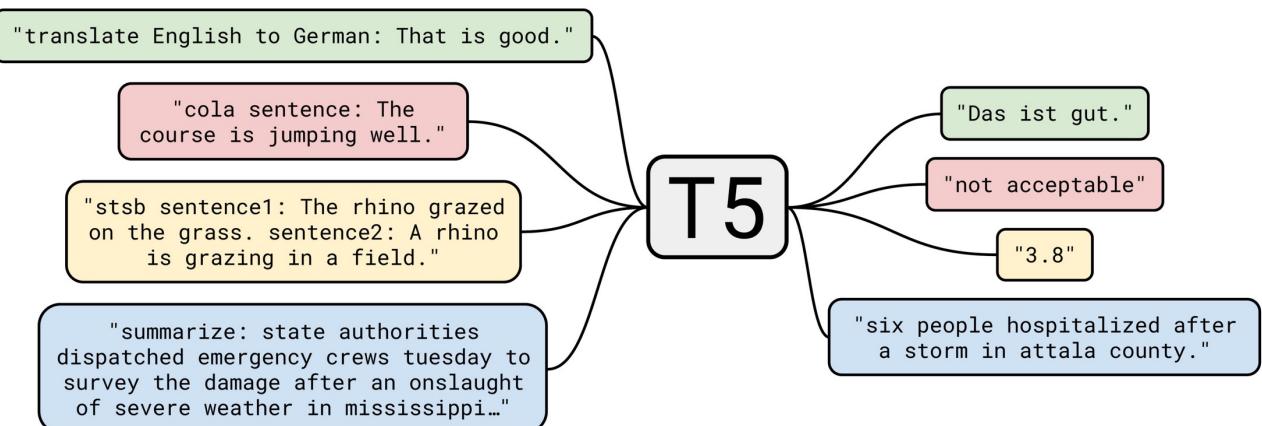
T5's Flexibility and Impact

- **Diverse Task Handling:**

- Classification (inputting a string and task name to output the class),
- Machine translation,
- Summarization
- Regression tasks (outputting a numerical value)

- **Fine-Tuning and Performance:**

- Served as a precursor to subsequent Google models, including FLAN-T5



Towards Extreme Multi-task Learning

T5's approach to **framing all tasks** as seq2seq problems has been instrumental in the initial “crazy” idea

- Demonstrating a groundbreaking method in **handling diverse NLP challenges**.
- (Aribandi et al, 2021) <https://openreview.net/pdf?id=Vzh1BFUCiIX>

Published as a conference paper at ICLR 2022

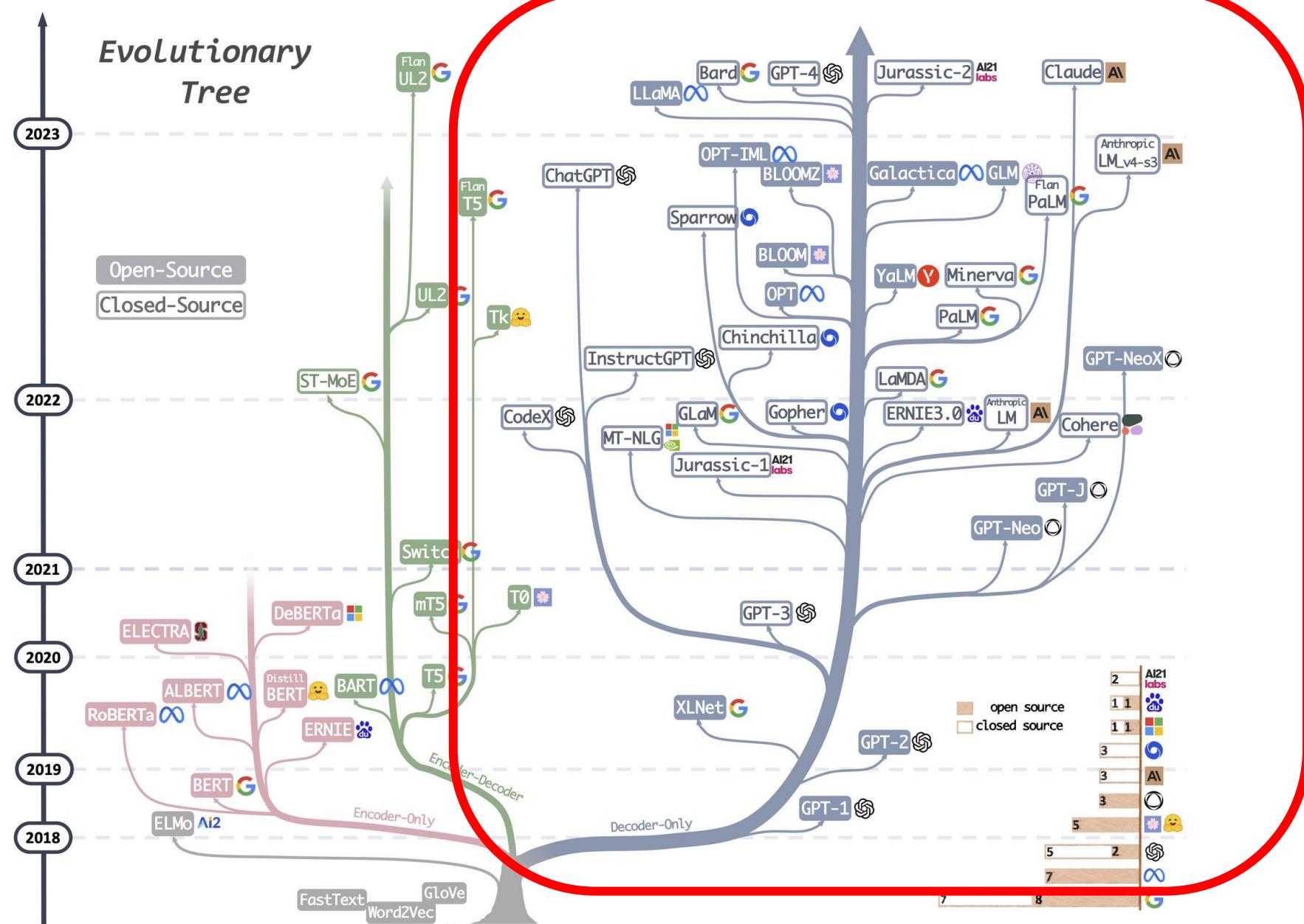
EXT5: TOWARDS EXTREME MULTI-TASK SCALING FOR TRANSFER LEARNING

Vamsi Aribandi^{*†}, Yi Tay[†], Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng,
Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni,
Jai Gupta, Kai Hui, Sebastian Ruder^{*}, Donald Metzler
Google Research, [‡]DeepMind
`{aribandi, yitay}@google.com`

ABSTRACT

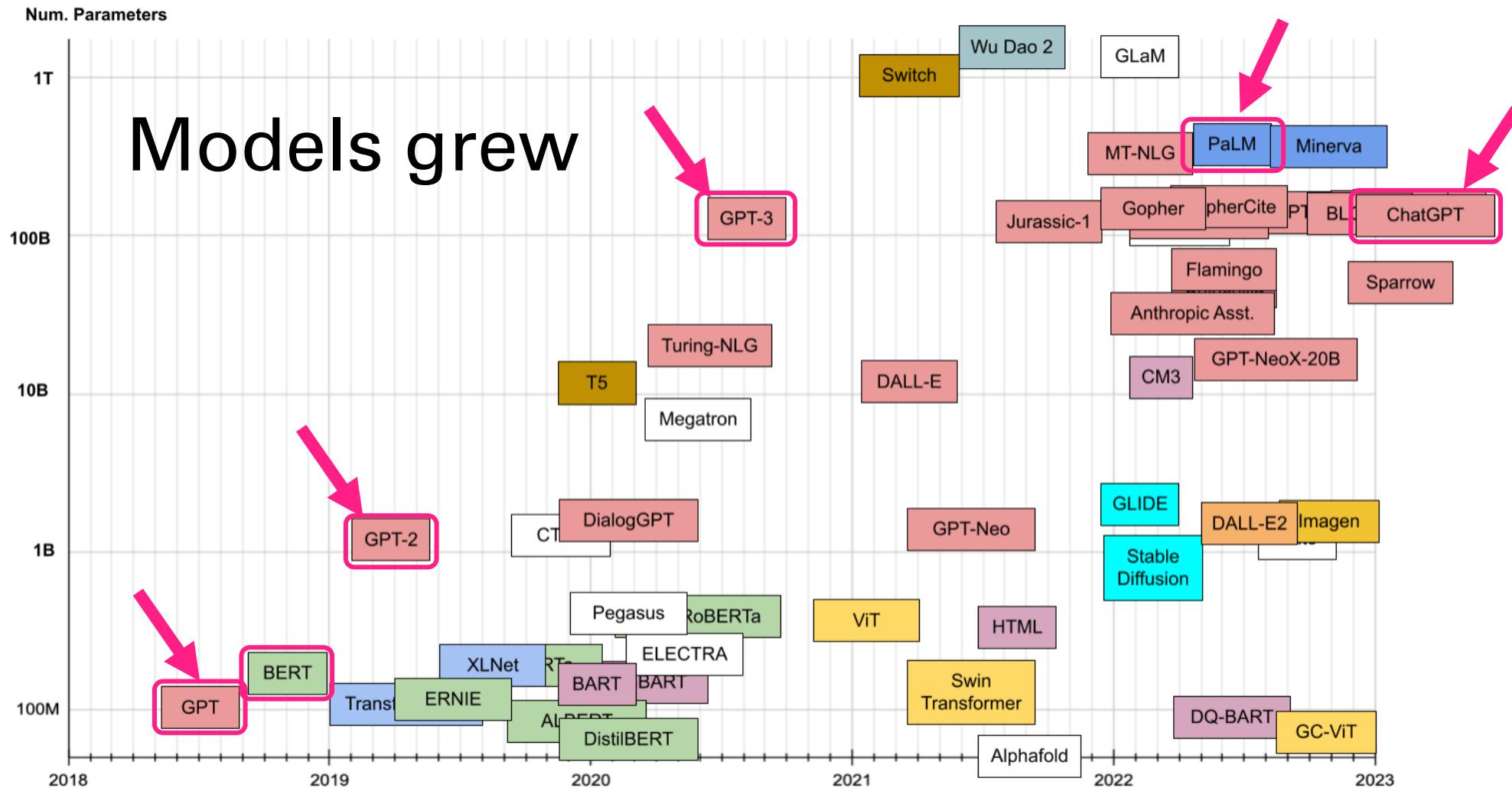
Despite the recent success of multi-task learning and transfer learning for natural language processing (NLP), few works have systematically studied the effect of scaling up the number of tasks during pre-training. Towards this goal, this paper introduces EXMIX (Extreme Mixture): a massive collection of 107 supervised NLP tasks across diverse domains and task-families. Using EXMIX, we study the effect of multi-task pre-training at the largest scale to date, and analyze co-training transfer amongst common families of tasks. Through this analysis, we

The rest is a family tree



Initially, decoder-based architectures did not attract 'much interest' (BERT had better performance), but with the increase in model size, things have radically changed.

But what happened in 2021 to cause this change of interest?



But it is not only a matter of size

«Ecce GPT-3»

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*

Jared Kaplan[†] Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry

Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan

Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter

Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray

Benjamin Chess Jack Clark Christopher Berner

Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI

Abstract

Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, humans can generally perform a new language task from only a few examples or from simple instructions – something which current NLP systems still largely

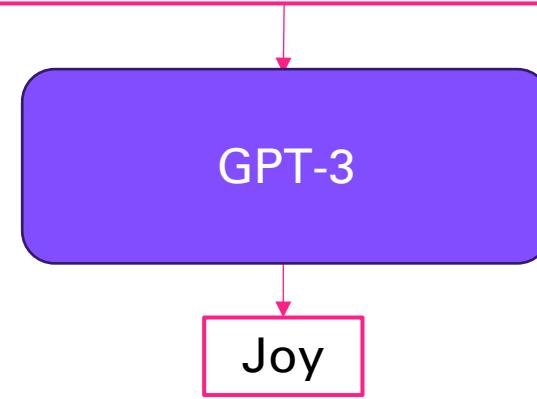
Some Data

	Bert-base	GPT-1	GPT-2	GPT-3
Parameters	110 Million	117 Million	1.5 Billion	175 Billion
Layers	12	12	48	96
Context Token Size	512	512	1024	2048
Hidden Layer	768	768	1600	12288
Dataset	BookCorpus + Wikipedia	BooksCorpus	WebText	The Pile
Number of Tokens	~3.3 billion	~1 billion	~8 billion	Hundreds of billions
Memory Size	-	~40 GB (uncompressed)	~40 GB (compressed)	~570 GB (compressed)
Batch Size	256	64	512	3.2M

The «powers» of GPT3

- **Pre-Training and Fine-Tuning Paradigm**
 - Traditional NLP models show **gains by pre-training** on large text corpora and then fine-tuning on specific tasks
 - **but require extensive task-specific datasets.**
- **GPT-3's «Breakthrough» in Few-Shot Learning**
 - GPT-3, with 175 billion parameters, demonstrates substantial **improvement in task-agnostic,**
 - **few-shot performance,**
 - **rivaling traditional fine-tuning methods.**

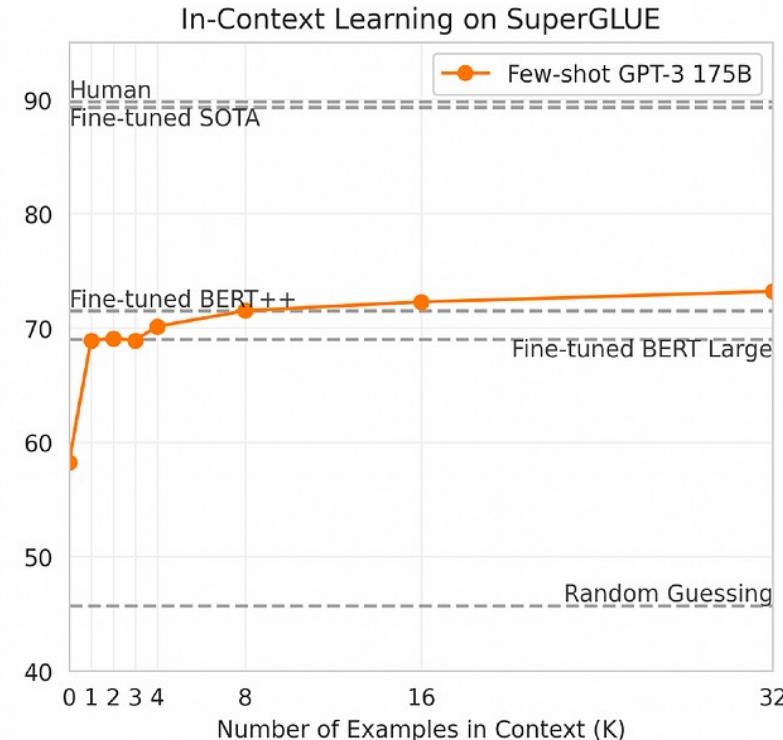
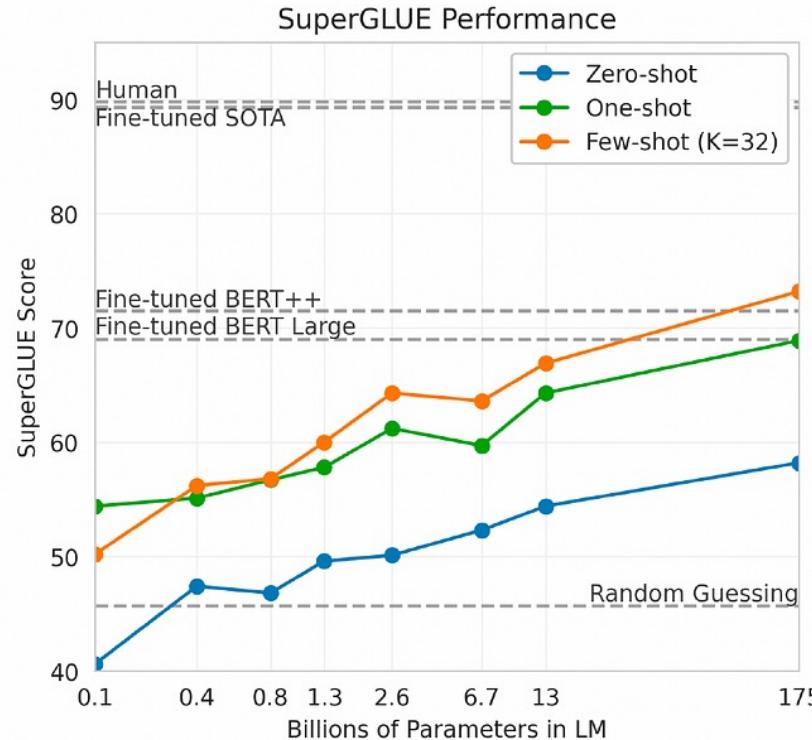
This sentence "*Such a wonderfull day*" evokes 'joy'.
This sentence "*Unfortunately I lost*" evokes 'sadness'.
This sentence "*I can't wait to see you*" evokes ...



The «powers» of GPT3

Diverse Task Performance Without Fine-Tuning

- Exhibits strong performance across various NLP tasks through text interactions alone, including translation, question-answering, and reasoning





But these models exhibits other capabilities...

v:22006.07682v2 [cs.CL] 26 Oct 2022

Published in Transactions on Machine Learning Research (08/2022)

Emergent Abilities of Large Language Models

Jason Wei¹

Yi Tay¹

Rishi Bommasani²

Colin Raffel³

Barret Zoph¹

Sebastian Borgeaud⁴

Dani Yogatama⁴

Maarten Bosma¹

Denny Zhou¹

Donald Metzler¹

Ed H. Chi¹

Tatsunori Hashimoto²

Oriol Vinyals⁴

Percy Liang²

Jeff Dean¹

William Fedus¹

¹ Google Research ² Stanford University ³ UNC Chapel Hill ⁴ DeepMind

Reviewed on OpenReview: <https://openreview.net/forum?id=yzkSU5zdwD>

Abstract

Scaling up language models has been shown to predictably improve performance and sample efficiency on a wide range of downstream tasks. This paper instead discusses an unpredictable phenomenon that we refer to as *emergent abilities* of large language models. We consider an ability to be emergent if it is not present in smaller models but is present in larger models. Thus, emergent abilities cannot be predicted simply by extrapolating the performance of

jasonwei@google.com
yitay@google.com
nlprishi@stanford.edu
craffel@gmail.com
barretzoph@google.com
sborgeaud@deepmind.com
dyogatama@deepmind.com
bosma@google.com
dennyyzhou@google.com
metzler@google.com
edchi@google.com
tashim@stanford.edu
vinyals@deepmind.com
pliang@stanford.edu
jeff@google.com
liamfedus@google.com



... many capabilities

Table 1: List of emergent abilities of large language models and the scale (both training FLOPs and number of model parameters) at which the abilities emerge.

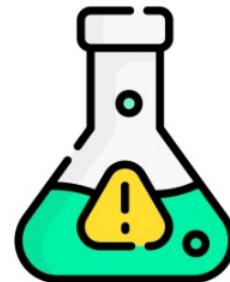
	Emergent scale		Model	Reference
	Train. FLOPs	Params.		
<u>Few-shot prompting abilities</u>				
• Addition/subtraction (3 digit)	2.3E+22	13B	GPT-3	Brown et al. (2020)
• Addition/subtraction (4-5 digit)	3.1E+23	175B		
• MMLU Benchmark (57 topic avg.)	3.1E+23	175B	GPT-3	Hendrycks et al. (2021a)
• Toxicity classification (CivilComments)	1.3E+22	7.1B	Gopher	Rae et al. (2021)
• Truthfulness (Truthful QA)	5.0E+23	280B		
• MMLU Benchmark (26 topics)	5.0E+23	280B		
• Grounded conceptual mappings	3.1E+23	175B	GPT-3	Patel & Pavlick (2022)
• MMLU Benchmark (30 topics)	5.0E+23	70B	Chinchilla	Hoffmann et al. (2022)
• Word in Context (WiC) benchmark	2.5E+24	540B	PaLM	Chowdhery et al. (2022)
• Many BIG-Bench tasks (see Appendix E)	Many	Many	Many	BIG-Bench (2022)
<u>Augmented prompting abilities</u>				
• Instruction following (finetuning)	1.3E+23	68B	FLAN	Wei et al. (2022a)
• Scratchpad: 8-digit addition (finetuning)	8.9E+19	40M	LaMDA	Nye et al. (2021)
• Using open-book knowledge for fact checking	1.3E+22	7.1B	Gopher	Rae et al. (2021)
• Chain-of-thought: Math word problems	1.3E+23	68B	LaMDA	Wei et al. (2022b)
• Chain-of-thought: StrategyQA	2.9E+23	62B	PaLM	Chowdhery et al. (2022)
• Differentiable search index	3.3E+22	11B	T5	Tay et al. (2022b)
• Self-consistency decoding	1.3E+23	68B	LaMDA	Wang et al. (2022b)
• Leveraging explanations in prompting	5.0E+23	280B	Gopher	Lampinen et al. (2022)
• Least-to-most prompting	3.1E+23	175B	GPT-3	Zhou et al. (2022)
• Zero-shot chain-of-thought reasoning	3.1E+23	175B	GPT-3	Kojima et al. (2022)
• Calibration via P(True)	2.6E+23	52B	Anthropic	Kadavath et al. (2022)
• Multilingual chain-of-thought reasoning	2.9E+23	62B	PaLM	Shi et al. (2022)
• Ask me anything prompting	1.4E+22	6B	EleutherAI	Arora et al. (2022)

With great power comes great responsibility



Limitations and Societal Impacts

- Acknowledges **challenges in certain tasks and potential methodological issues**, while highlighting the model's ability to produce human-like text, **raising important societal considerations**.



Toxicity

Harmful or discriminatory language or content



Hallucination

Factually incorrect content



Legal Aspects

Data Protection, Intellectual Property, and the EU AI Act

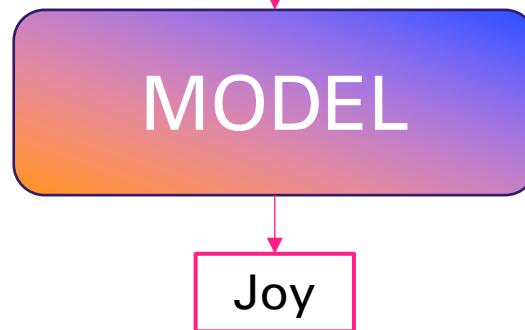
Material for another tutorial



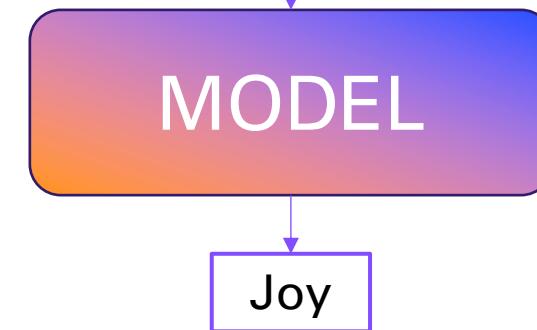
We like it, but ...

We just want to move from answers based on completion to **execute instructions**...

This sentence "*Such a wonderfull day*" evokes 'joy'.
This sentence "*Unfortunately I lost*" evokes 'sadness'.
This sentence "*I can't wait to see you*" evokes ...



Given this sentence, please tell me what emotion it evokes between 'joy', 'sadness', ... : "*I can't wait to see you*"

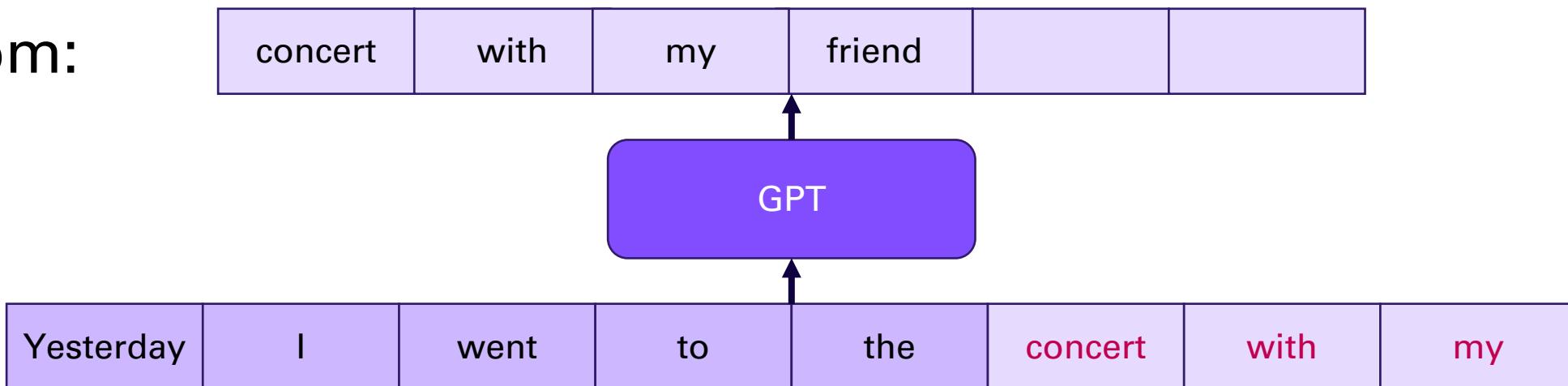


From GPT to Instruct-GPT

Evolving to Instruct-GPT:

- need for a **model that could understand and execute human-like instructions**
- similar to how humans follow commands.

From:

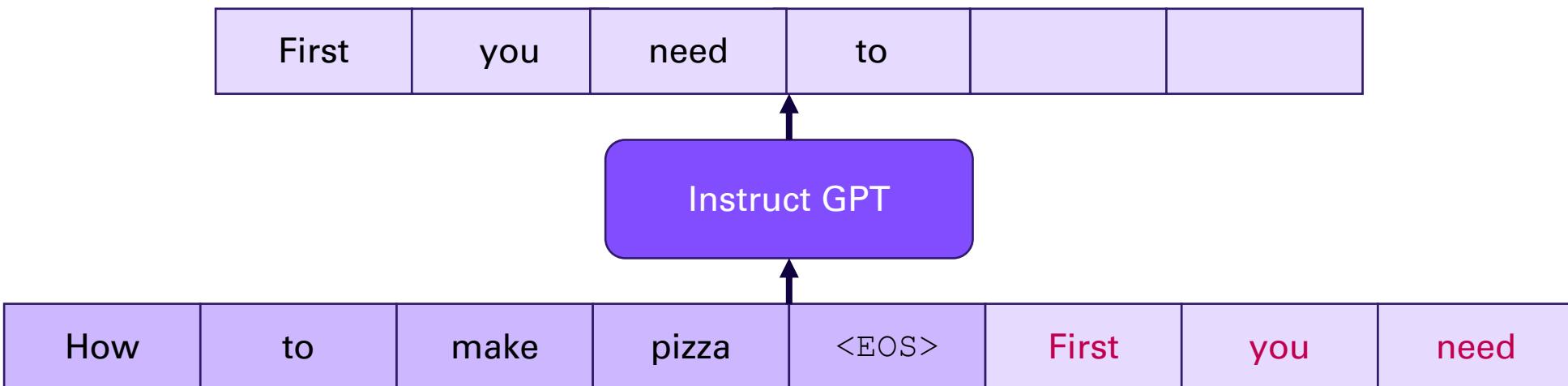


From GPT to Instruct-GPT

Evolving to Instruct-GPT:

- need for a **model that could understand and execute human-like instructions**
- similar to how humans follow commands.

To:



Instruct-GPT in 3 steps

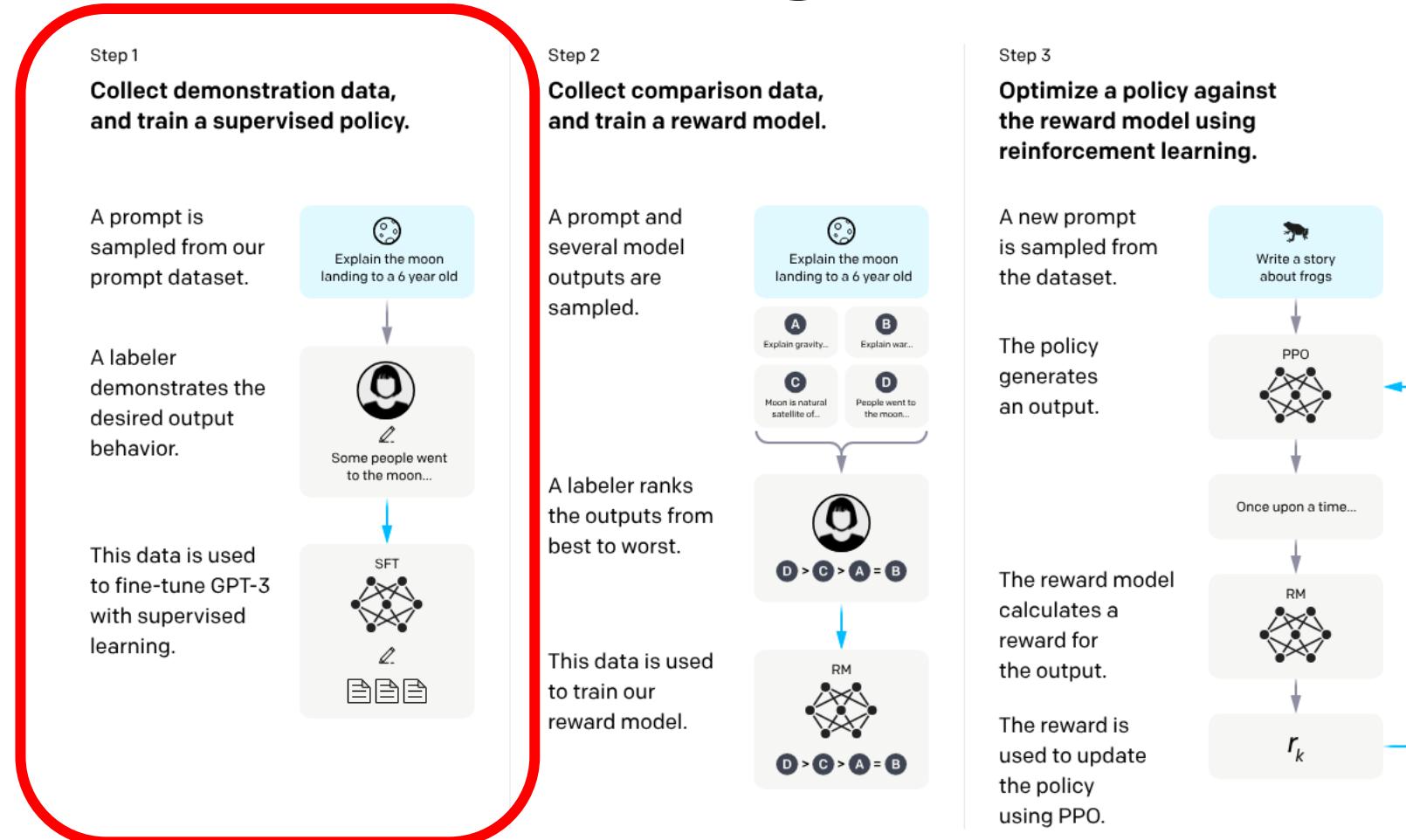
Step 1: Supervised Fine-Tuning

Methodology: Training on a dataset of labeled examples

- each prompt paired with an ideal response

Objective: Teach the model correct responses to various prompts

- Imitating human-like behavior from examples



From: <https://openai.com/research/instruction-following>

Instruct-GPT in 3 steps

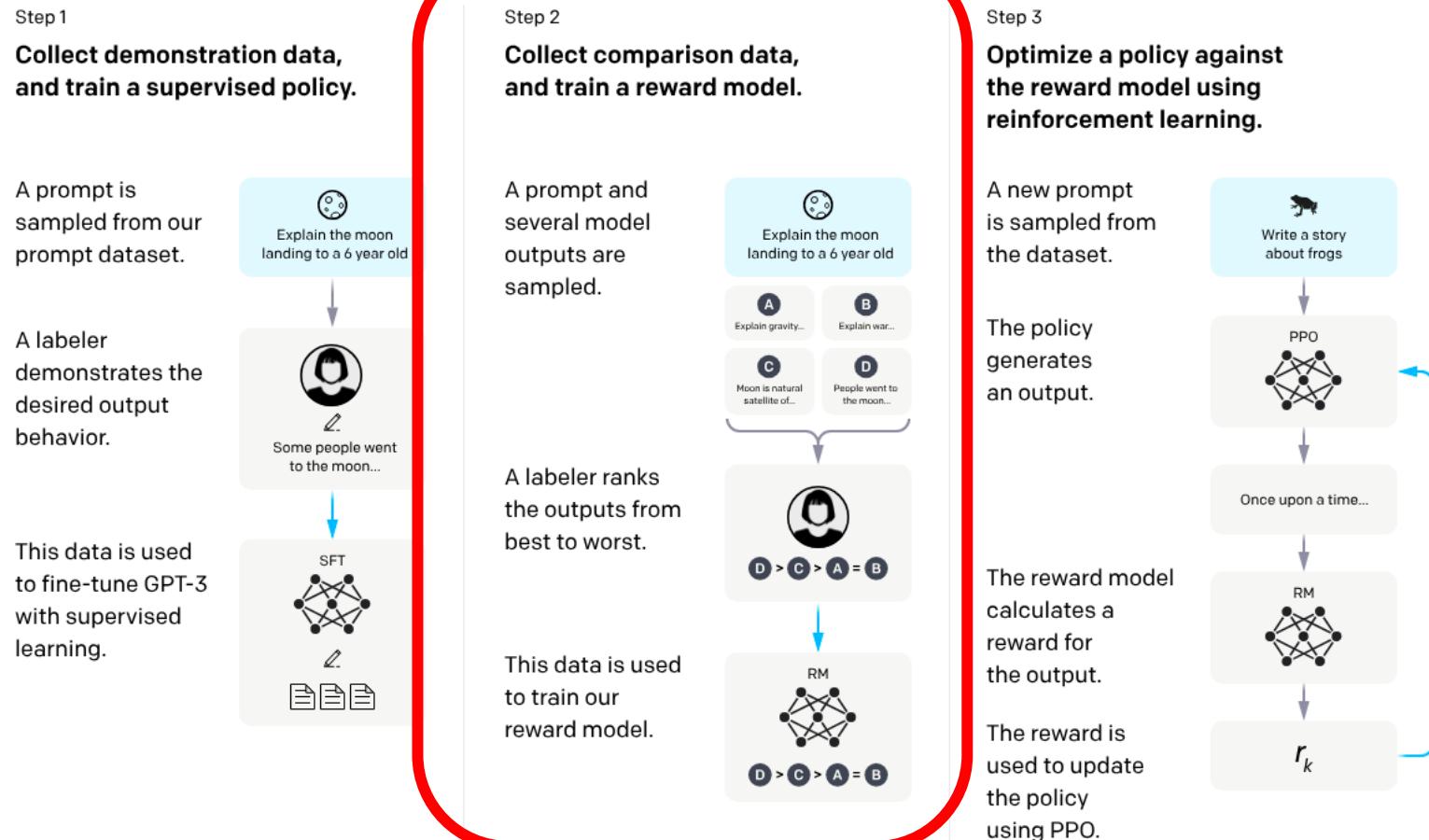
Step 2: Reward Model (RM) Training

Methodology: Develop a model that assigns rewards to responses based on human preferences.

- Re-rank the responses

Objective: Prepare the model to understand and evaluate the quality of its responses

- beyond just accuracy.



Instruct-GPT in 3 steps

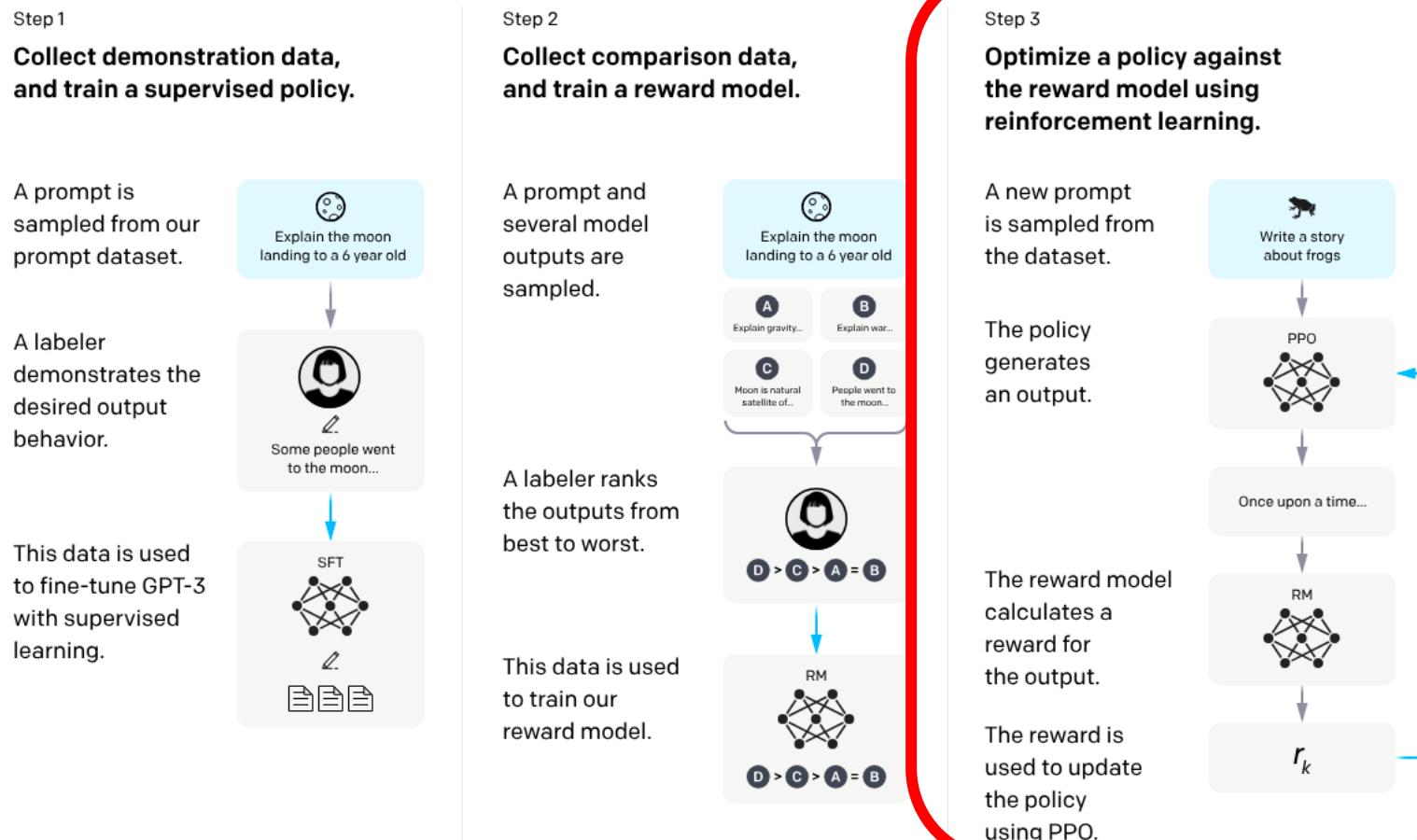
Step 3: Reinforcement Learning via PPO

Methodology: Uses Reinforcement Learning to fine-tune responses

- the model is rewarded for high-quality outputs.

Objective: Enhance the model's ability to generate relevant, useful responses in varied and complex scenarios.

- **optimizing responses for quality and contextual appropriateness**
- not just replicating correct answers.



From: <https://openai.com/research/instruction-following>

Evolution from Transformer architecture in OpenAI

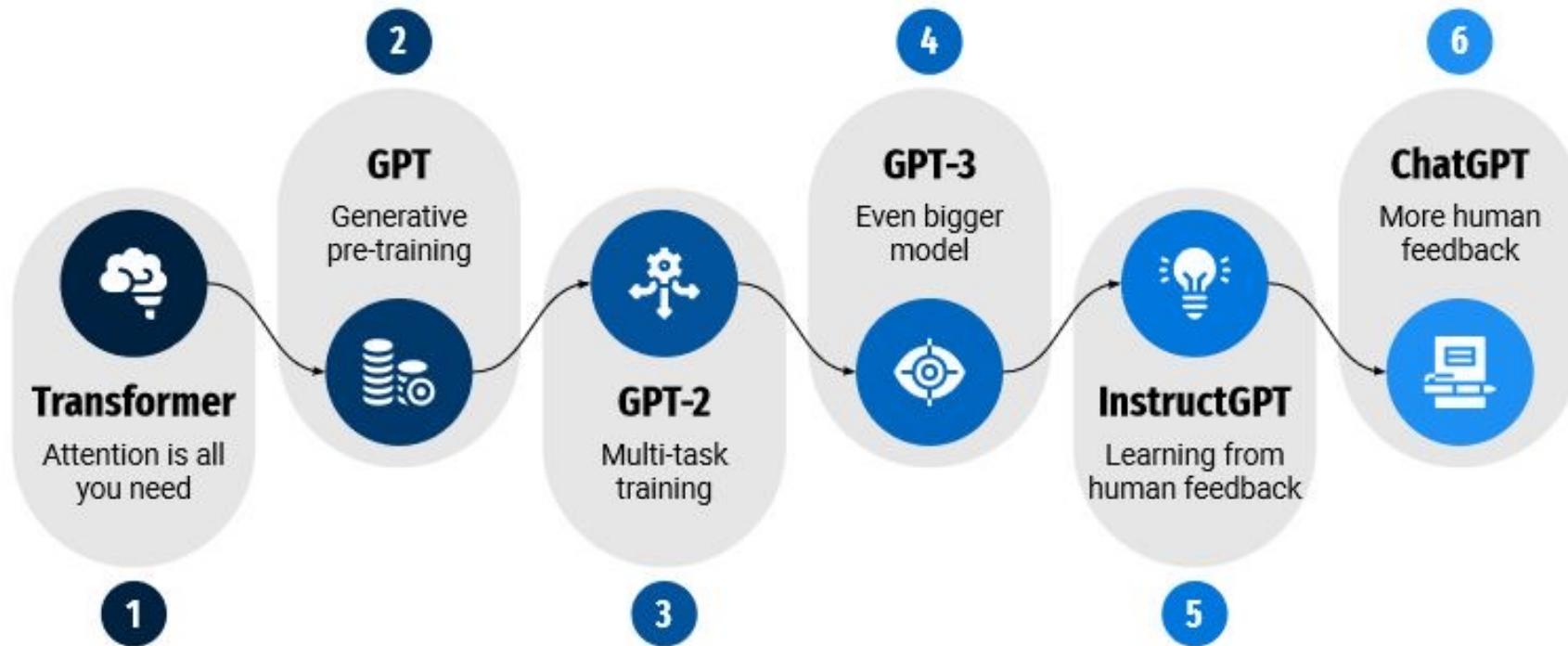


Image from: <https://pwr.edu.pl/en/university/news/chatgpt-has-sparked-a-revolution-how-will-it-change-our-lives-10621.html>

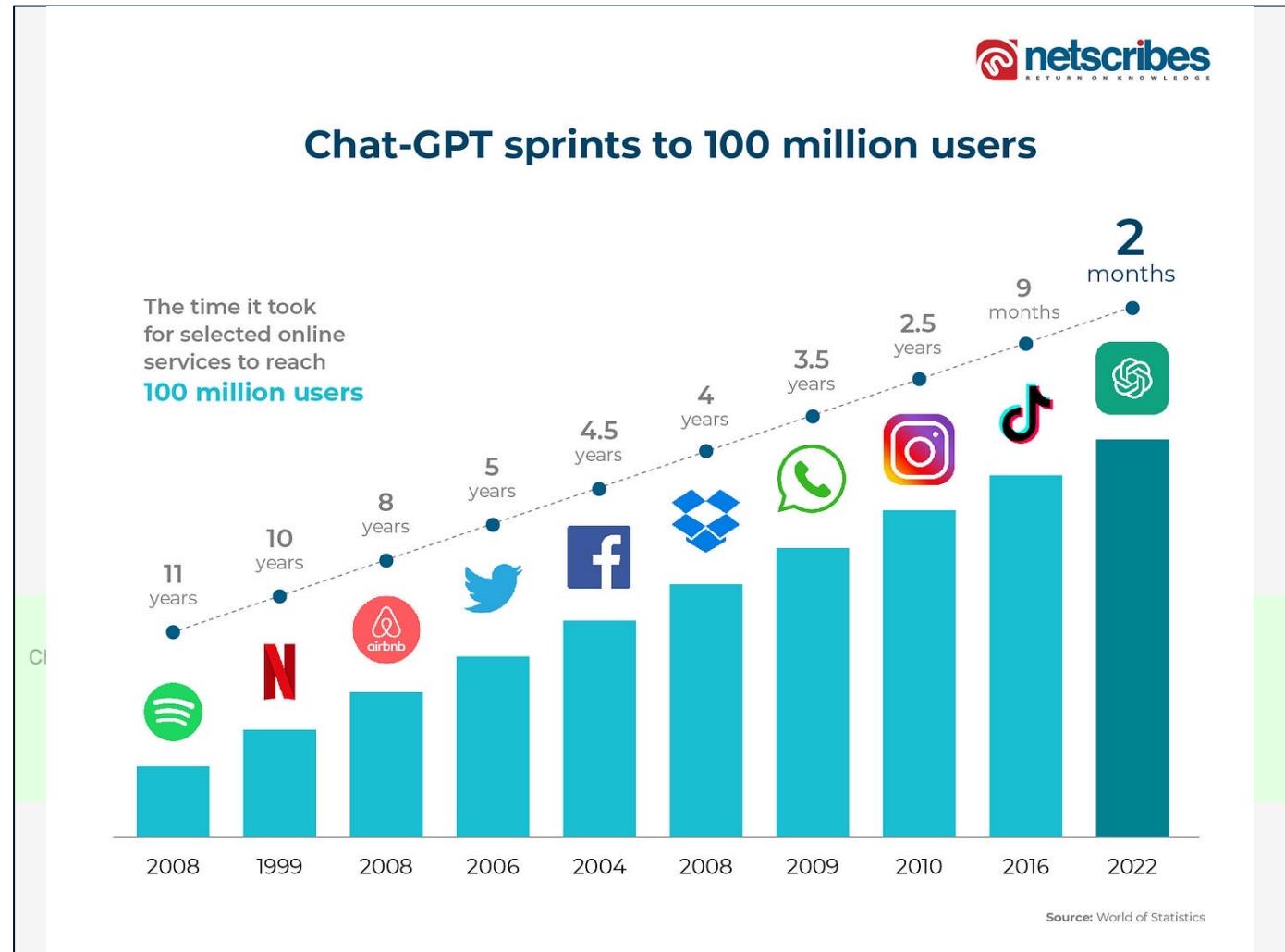
«Ecce ChatGPT»

Nov 28th, 2023

The prior approaches culminated in OpenAI releasing ChatGPT a 'revolutionary' system capable of:

- Having a human-like dialogue in natural language
- Handle multiple languages
- Answering follow up questions
- Admitting its mistakes
- Challenging incorrect premises
- Rejecting inappropriate requests
- ~~Making a good coffee~~
- ...

But at what cost?

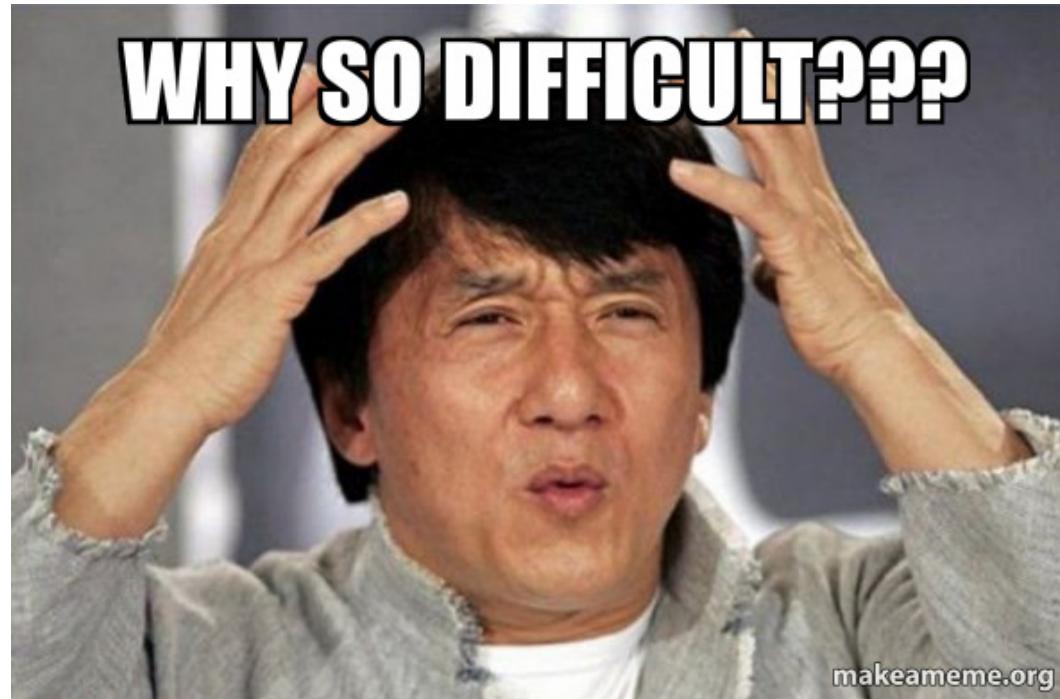


Resources need for ChatGPT

<https://lifearchitect.ai/chatgpt/>

- **Training hardware:** Access to [a supercomputer with ~10,000 GPUs and ~285,000 CPU cores.](#)
- **Staffing:** For training, you'll need access to the smartest PhD-level data scientists in the world. OpenAI paid their Chief Scientist Ilya Sutskever [\\$1.9 million dollars per year](#) (USD) in 2016. Perhaps budget >\$200 million for staffing the first year.
- **Time (data collection):** It took a solid 12-18 months to agree on, collect, clean, and prepare data.
- **Time (training ~300 years):** Expect a model to take 9-12 months of training, and that's if everything goes perfectly. You may need to run it several times, and you may need to train several models in parallel. Things do go wrong, and they can completely mess up the results (see the [GPT-3 paper](#), [China's GLM-130B](#) and [Meta AI's OPT-175B logbook](#)).
- **Inference:** Fairly beefy computers, plus devops staffing resources, but this is the least of your worries. Good luck!

Is this a groundbreaking revolution or
merely a stochastic parrot?



Material for
another tutorial



2022: An odyssey in Prompt-Engineering and In-Context Learning

- **Prompt-engineering**, especially **in-context learning**, has become a key research area in 2022.
 - Involves demonstrating or explaining tasks to LLMs through tailored descriptions or examples within prompts (few-shot)
- **Why Prompt-Engineering?**
 - Aims to **leverage the rich, emergent information captured by LLMs** to perform tasks.
- A necessity **driven by resource constraints?**
 - The luxury of fine-tuning **100B models** remains **inaccessible for many**
 - **Prompt-engineering offers a more resource-efficient and accessible way** to harness the power of LLMs without extensive computational resources.

... but in the end ...



- **Transformer-based models are powerful Tools**
 - Suitable for Multi-task learning
- **State-of-the-Art Performance**
 - Thanks to **extensive pre-training**
- **Potentially applicable to EVALITA**

The «crazy» idea

Fine-tuning a single LLMs by applying it to **ALL EVALITA tasks**



How did we get here?

The «crazy» idea

Fine-tuning a single LLMs by applying it to **ALL EVALITA tasks**

- **Modeling Tasks as Prompts:**
 - The core idea is to frame each task as a «linguistic» prompt
 - Enabling a unified approach to diverse NLP challenges
- **Leveraging EVALITA Training Data:**
 - By aggregating training examples from EVALITA
 - Fine-tuning a decoder-based architecture like GPT
- **Power of Scale:**
 - If the model is sufficiently large, it should possess the capability to solve all tasks.



Why «crazy»?

- **Limited Access to Large Language Models (LLMs):**
 - We **didn't have an LLM ready**, nor the capacity or resources equivalent to 300 years to develop one.
 - While open-source versions like BLOOM existed, the quality of models with less than 170 billion parameters wasn't optimal.
- **How to handle instructions:**
 - The zero/few-shot capability were not appealing
 - We **lacked a method or resources to "convert" an LLM to execute instructions.**
 - Data from the Instruct GPT experience was not available

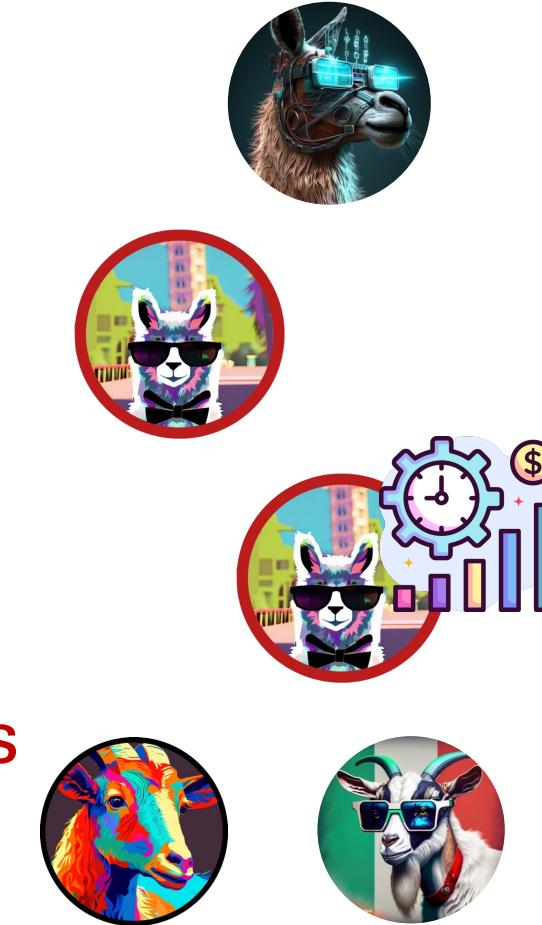


Why «crazy»? (2)

- **Computational Constraints:**
 - Even with data available for the initial step of Instruct GPT, **our computational power was limited** (only 1 GPU T4 with 16GB of RAM), insufficient for fine-tuning the model.
- **Language-Specific Model Limitations:**
 - We did not have a model or instruction examples specifically tailored for Italian, presenting a significant language barrier.

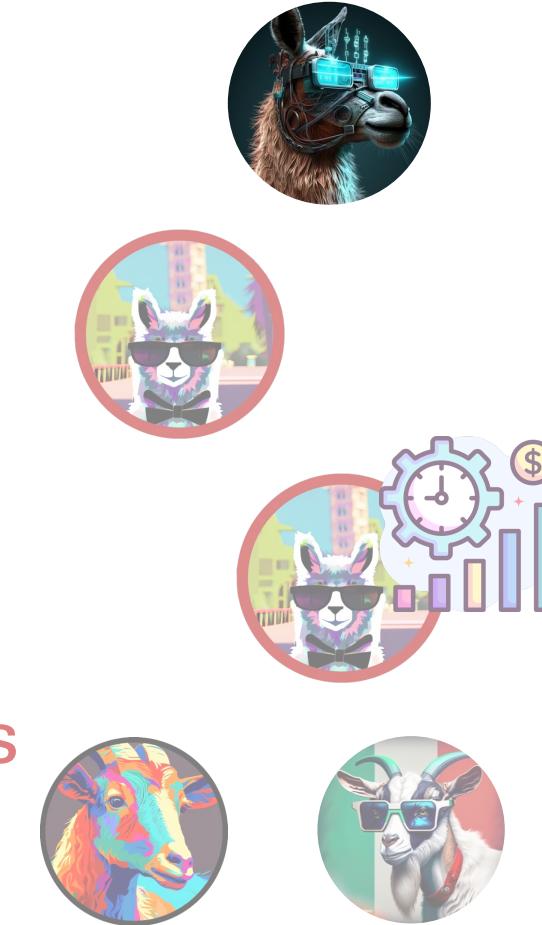
But then many camelids appeared after February 17

- Limited Access to LLMs
 - FEB 24: LLaMA
- How to handle instructions
 - MAR 13: Alpaca
- Computational Constraints
 - MAR 23: Alpaca-LoRA
- Language-Specific Model Limitations
 - APR 02: Cabrita/Camoscio



But then many camelids appeared after February 17

- Limited Access to LLMs
 - FEB 24: LLaMA
- How to handle instructions
 - MAR 13: Alpaca
- Computational Constraints
 - MAR 23: Alpaca-LoRA
- Language-Specific Model Limitations
 - APR 02: Cabrita/Camoscio



LLAmA: Large Language Model Meta AI (Touvron et al., 2023)

- **LLaMA**: A collection of foundation **language models** with parameters ranging from 7B, 13B to 65B.
- **Training Data**: Models **trained on trillions of tokens** exclusively using publicly available datasets.
- **Performance**: **LLaMA-13B outperforms GPT-3 (175B)** in most benchmarks, while LLaMA-65B competes with Chinchilla-70B and PaLM-540B.
- **Accessibility**: **All models leaked released to the research community** for wide accessibility...

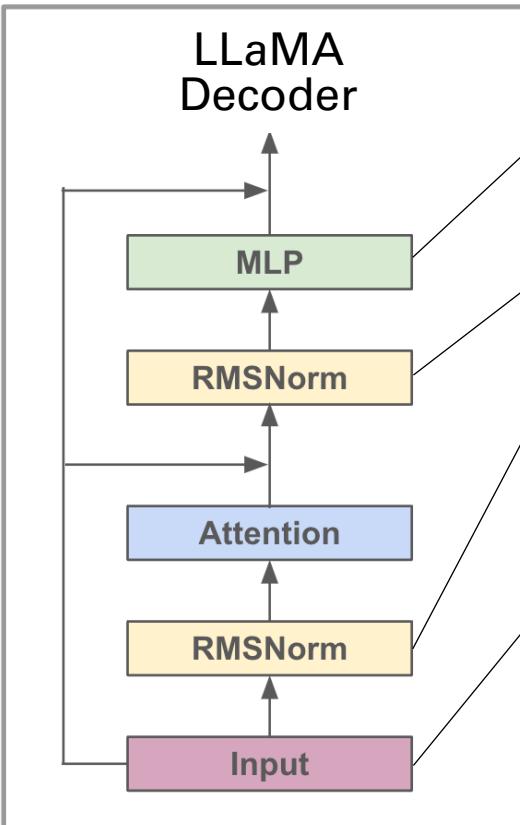
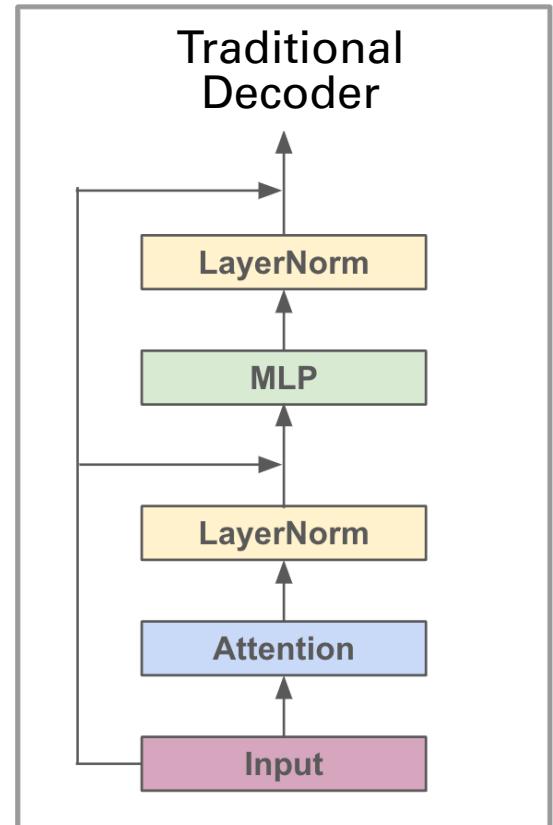
• <https://www.theverge.com/2023/3/8/23629362/meta-ai-language-model-llama-leak-online-misuse>



Making LMs publicly available: LLaMA

(Touvron et al., 2023)

Large Language Model Meta AI is a series of Foundational models: 7, 13, 32 and 65 billion parameters

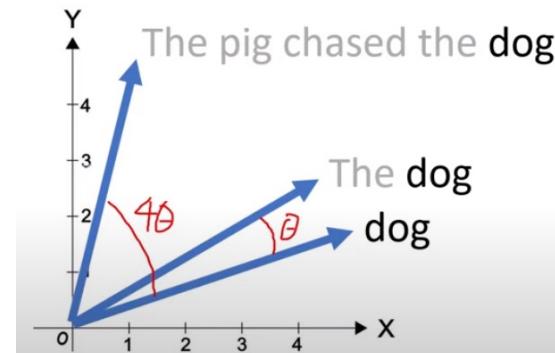


$$\text{SwiGLU}(x) = \text{Swish}(xW) \cdot xV$$

$$\text{Swish}(x) = x \cdot \text{Sigmoid}(\beta x)$$

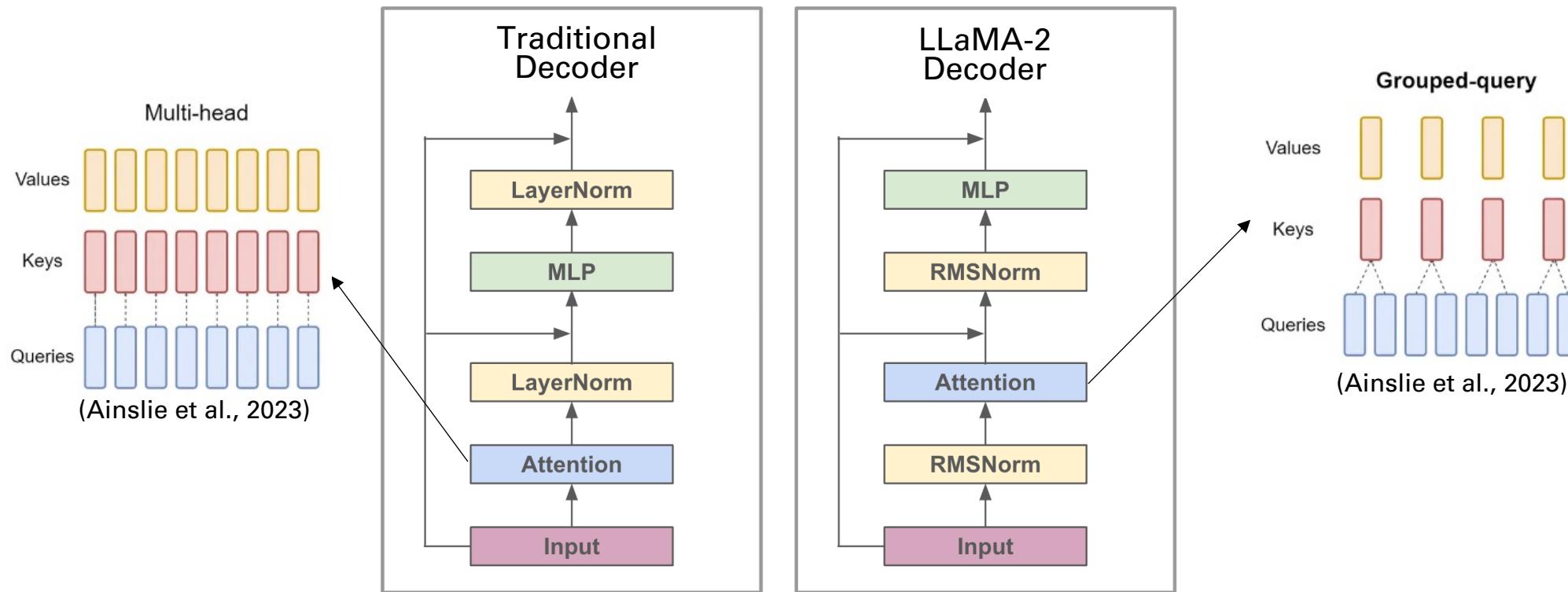
$$\bar{a}_i = \frac{a_i}{\text{RMS}}, \text{ where } \text{RMS} = \sqrt{\frac{1}{n} \sum_{i=1}^n a_i^2}$$

Rotary Positional Embeddings (RoPE) instead of Absolute/Relative



Making LMs publicly available: LLaMA-2

A big difference lies in the Attention block

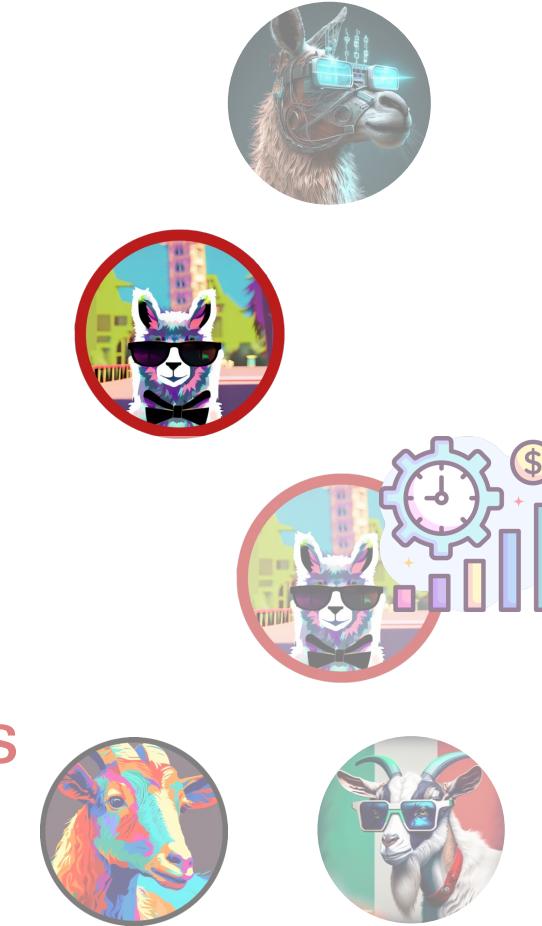


LLaMA: some cons

- **Bias and Toxicity:** Like other LLMs, LLaMA models encode biases and potential toxicity, necessitating further research in responsible AI.
- **Data Intensity:** Even smaller models require extensive data (up to 1T tokens), posing challenges in terms of data collection and management.
- **Model Scaling:** Determining optimal model size for specific tasks remains a challenge, balancing between compute resources and desired performance levels.

But then many camelids appeared after February 17

- Limited Access to LLMs
 - FEB 24: LLaMA
- How to handle instructions
 - MAR 13: Alpaca
- Computational Constraints
 - MAR 23: Alpaca-LoRA
- Language-Specific Model Limitations
 - APR 02: Cabrita/Camoscio



Instruction-tuning: Alpaca

(Taori et al. 2023)

- **LLaMA Limitations:** Originally a language model, not designed for instruction-following tasks.
- **Challenges in LLMs:** Issues with misinformation, stereotypes, and toxic language.
- **Alpaca:** Fine-tuned from Meta's LLaMA 7B, specifically for instruction-following.
 - Let's say, only the step 1 from Instruct GPT is used
- **Training:** Utilizes 52K demonstrations in the style of self-instruct from text-davinci-003 (a variant of GPT3).
- **Efficiency:** Smaller, more accessible, and easier to reproduce compared to models like GPT-3.5.



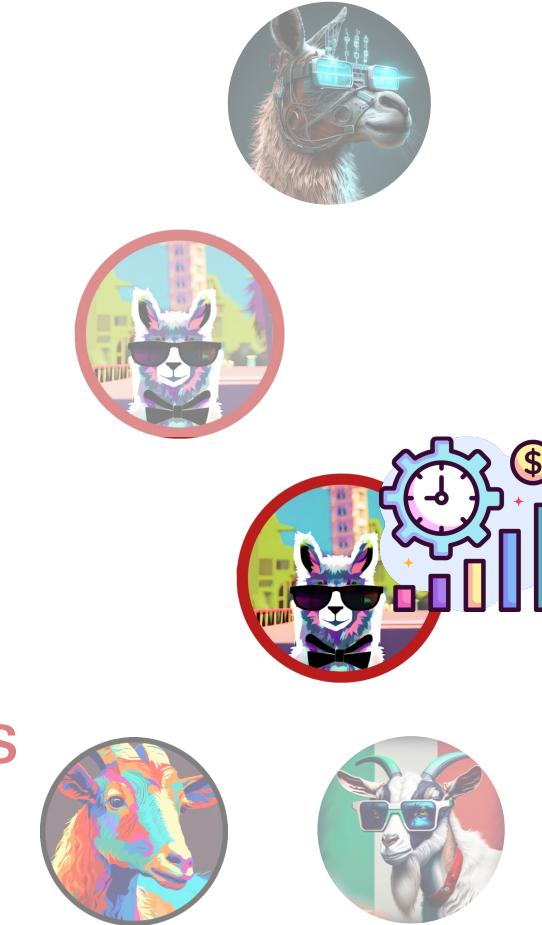
Instruction-tuning: Alpaca

(Taori et al. 2023)



But then many camelids appeared after February 17

- Limited Access to LLMs
 - FEB 24: LLaMA
- How to handle instructions
 - MAR 13: Alpaca
- Computational Constraints
 - MAR 23: Alpaca-LoRA
- Language-Specific Model Limitations
 - APR 02: Cabrita/Camoscio



Challenge: 16GB GPU resources

ChatGPT's resources:
10.000 GPUs



1xTesla T4



Low-Rank Adaptation (LoRA)

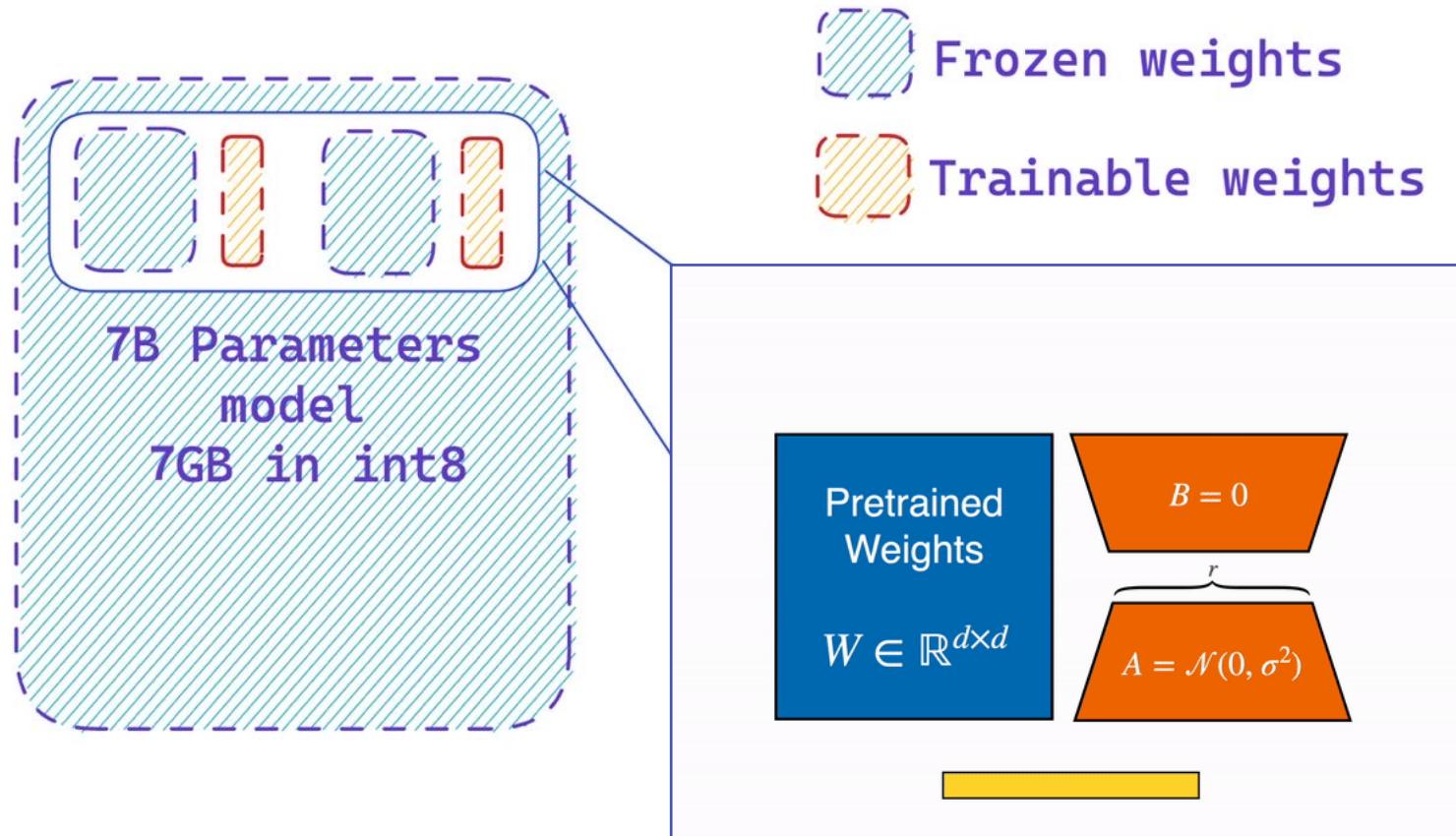
(Hu et al., 2021)

Low Rank Adaptation (LoRA: Hu et al., 2021): create the parallel (fine-tunable) adapters as smaller matrices:

- add the adapters to the base model while keeping the base model frozen

LoRA is **NOT** learning any parameter, but the **changes in the parameters!**

The diagram illustrates the decomposition of a total weight matrix $W_0 + \Delta W$ into two components: W_0 and ΔW . In the 'Traditional FT' section, the total weight is shown as a single block with a red border. In the 'LoRA FT' section, the total weight is shown as a sum of two blocks: W_0 (with a blue border) and ΔW (with a red border). The ΔW block is further divided into two sub-blocks: BA (with a yellow border) and U (with a purple border).



Alpaca LoRA

Within just a few days following the release of Alpaca's training material, LoRA was utilized to fine-tune LLaMA into Alpaca efficiently, using only a «small» GPU:

- <https://github.com/tloen/alpaca-lora>

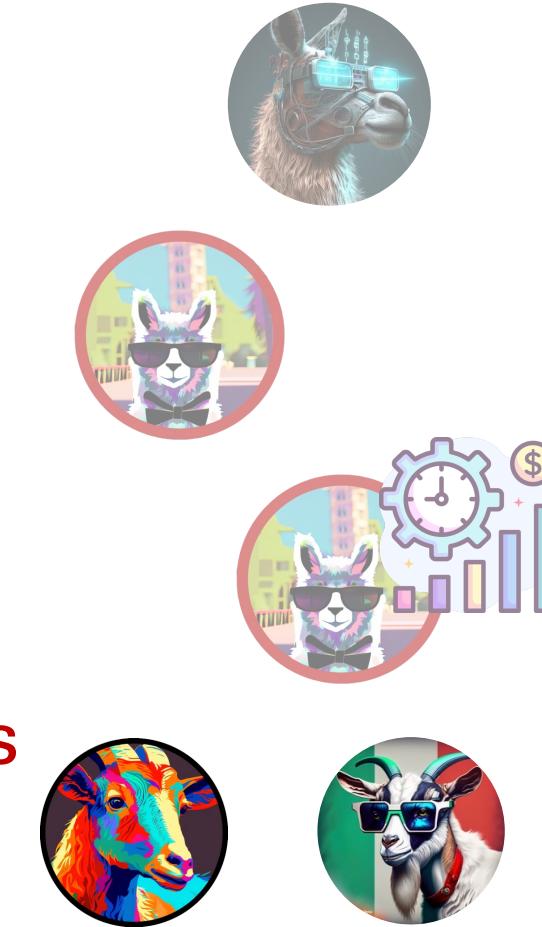
The screenshot shows the GitHub repository page for 'alpaca-lora'. At the top, there is a navigation bar with a profile icon, the repository name 'alpaca-lora' (marked as 'Public'), and buttons for 'Watch' (155), 'Fork' (2.1k), and 'Star' (17.5k). Below the navigation bar, there are buttons for switching between branches ('main' is selected), 'Go to file', 'Add file', and 'Code'. On the right, there is an 'About' section with the text 'Instruct-tune LLaMA on consumer hardware'. The main content area displays two recent commits:

- juletx Add machine-translated Alpaca dataset in 6 languages an... (on Apr 18, 152 interactions)
- .github/workflows Fix linters (#185) (8 months ago)
- templates Templated prompter (#184) (8 months ago)

On the far right, there is a sidebar with links to 'Readme', 'Apache-2.0 license', 'Activity', and '17.5k stars'.

But then many camelids appeared after February 17

- Limited Access to LLMs
 - FEB 24: LLaMA
- How to handle instructions
 - MAR 13: Alpaca
- Computational Constraints
 - MAR 23: Alpaca-LoRA
- Language-Specific Model Limitations
 - APR 02: Cabrita/Camoscio



Multilingual Adaptation of Alpaca: The Birth of Language-Specific Models

Shortly after Alpaca's dataset release, several groups opted to use automatic translation methods (even using ChatGPT) to translate the Alpaca dataset.

- **Objective:** Enable the training of language-specific versions of Alpaca, adapting LLaMa to each new language.

 <https://huggingface.co/22h/cabrita-lora-v0-1>
 <https://huggingface.co/qychen/luotuo-lora-7b-0.1>
 <https://huggingface.co/ziqingyang/chinese-alpaca-lora-7b>
 <https://huggingface.co/kunishou/Japanese-Alapaca-LoRA-7b-v0>
 <https://huggingface.co/bofenghuang/vigogne-lora-7b>
 <https://huggingface.co/Thaweewat/thai-buffala-lora-7b-v0-1>
 https://huggingface.co/thisserand/alpaca_lora_german
 <https://huggingface.co/mmosiolek/polpaca-lora-7b>

 <https://huggingface.co/chrisociepa/alpaca-lora-7b-pl>
 <https://huggingface.co/teelinsan/camoscio-7b-llama>
 https://huggingface.co/IlyaGusov/llama_7b_ru_turbo_2
 [alpaca_lora](#)
 <https://huggingface.co/robinhad/ualpaca-7b-llama>
 <https://huggingface.co/mchl-labs/stambecco-7b-plus>
 <https://huggingface.co/plncmm/guanaco-lora-7b>
   <https://huggingface.co/HiTZ/alpaca-lora-7b-en-pt-es-ca-eu-gl-at>

Among the camelids: Camoscio



Camoscio (Santilli and Rodolà, 2023): A LLaMA-based model capable of executing commands in Italian.

- Fine-tuned using the data from Alpaca translated using ChatGPT

It seemed a good starting point for our «crazy» idea so...

... so, we started thinking



ExtremITA at EVALITA

Evaluation of NLP and Speech Tools for Italian

Multi-Task Sustainable Scaling to Large Language Models at its Extreme

+
•
o

C.D. Hromei, D. Croce, V. Basile, R. Basili



Associazione Italiana di
Linguistica Computazionale



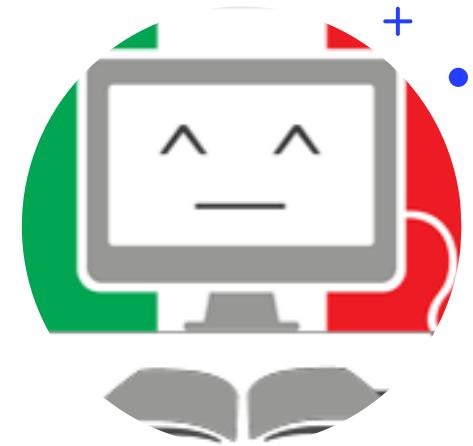
Università di Roma
Tor Vergata



UNIVERSITÀ
DI TORINO

Motivation

- Recent development of LLaMA (Touvron et al., 2023) **foundational models has opened new ways of exploiting natural language** for task specific training through the use of prompting
- **Multitask joint training for a single monolithic architecture** is appealing when coupled with instructions
- **Tasks are modelled as linguistic problems** (see ChatGPT): from traditional classification to solve a task to prompting and natural language inference



Our Method

22

Istruzione: Quali emozioni sono espresse in questo testo? Puoi scegliere una o più emozioni tra 'Rabbia', 'Anticipazione', 'Disgusto', 'Paura', 'Gioia', 'Amore', 'Tristezza', 'Sorpresa', 'Fiducia', o 'Neutro'.
Input: "Che bella giornata"

Istruzione: Scrivi le menzioni di entità nel testo, indicandone il tipo: [PER] (persona), [LOC] (luogo), [ORG] (organizzazione).
Input: "La Banca d'Italia"

...

Istruzione: Quanto è coerente questa frase, su una scala da 0 a 5?
Input: "Che bella giornata"

LLM-based Decoder



Gioia

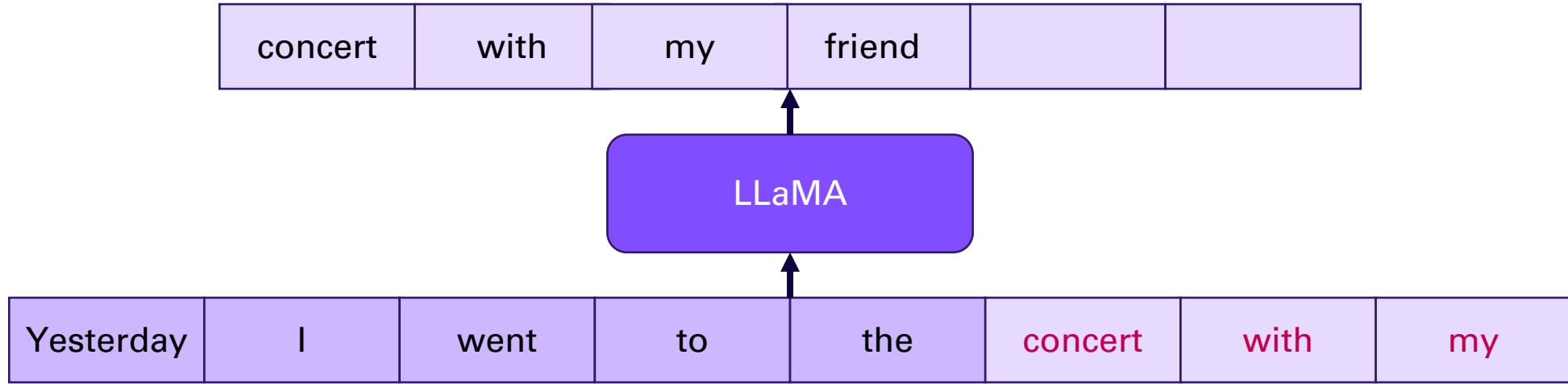
+

[ORG] Banca d'Italia

...

4.0

LLaMA (Touvron et. al 2023): Autoregressive Decoder-only

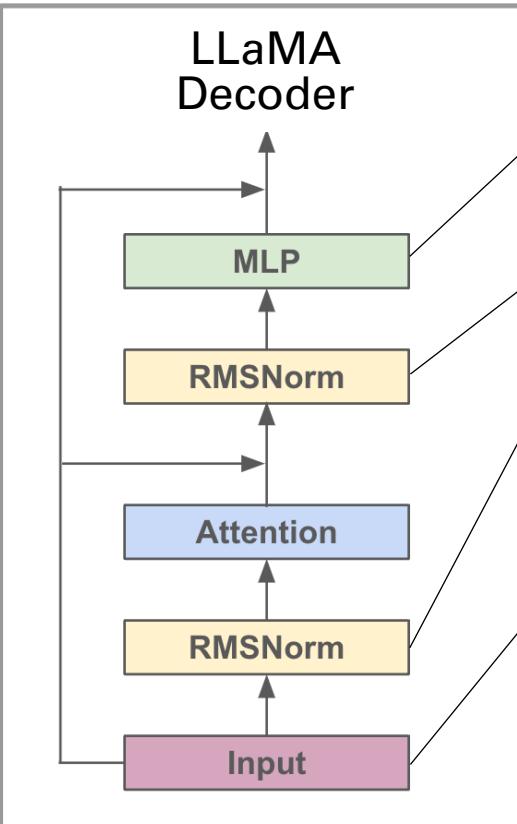
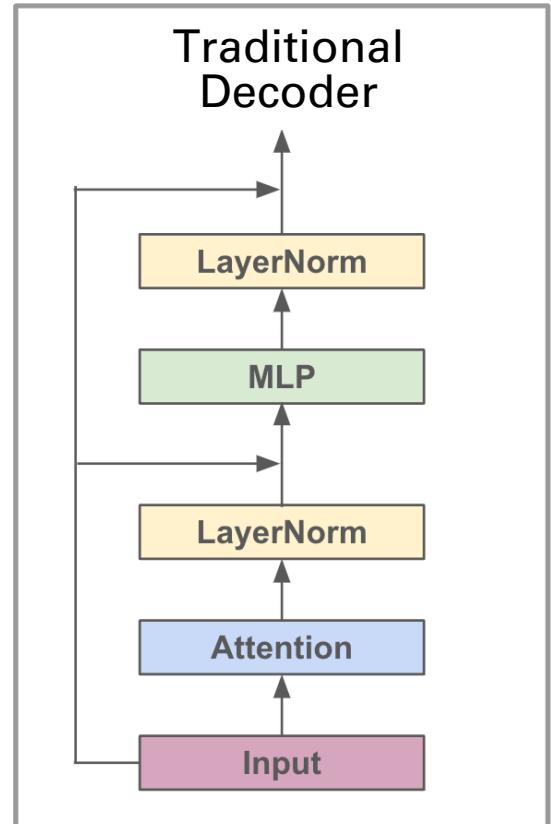




Making LMs publicly available: LLaMA

(Touvron et al., 2023)

Large Language Model Meta AI is a series of Foundational models: 7, 13, 32 and 65 billion parameters

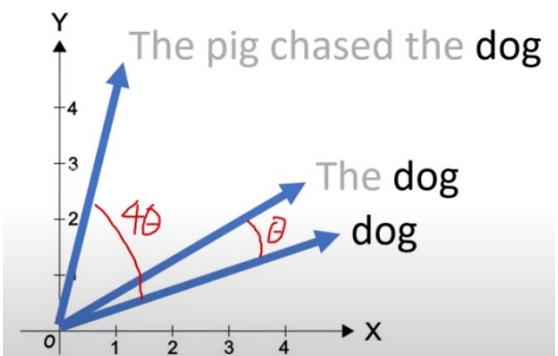


$$\text{SwiGLU}(x) = \text{Swish}(xW) \cdot xV$$

$$\text{Swish}(x) = x \cdot \text{Sigmoid}(\beta x)$$

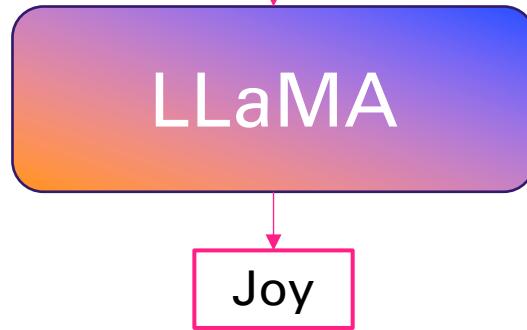
$$\bar{a}_i = \frac{a_i}{\text{RMS}}, \text{ where } \text{RMS} = \sqrt{\frac{1}{n} \sum_{i=1}^n a_i^2}$$

Rotary Positional Embeddings (RoPE) instead of Absolute/Relative

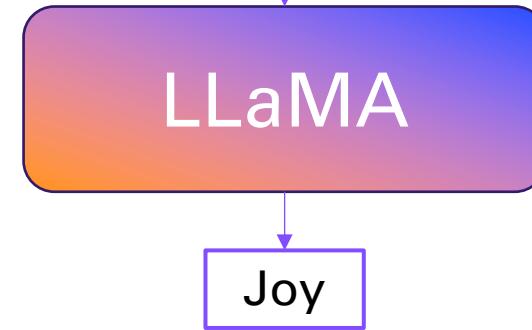


LLaMA: Few-shot learning vs Instruction-tuning

This sentence *"Such a wonderfull day"* evokes 'joy'.
This sentence *"Unfortunately I lost"* evokes 'sadness'.
This sentence *"I can't wait to see you"* evokes ...

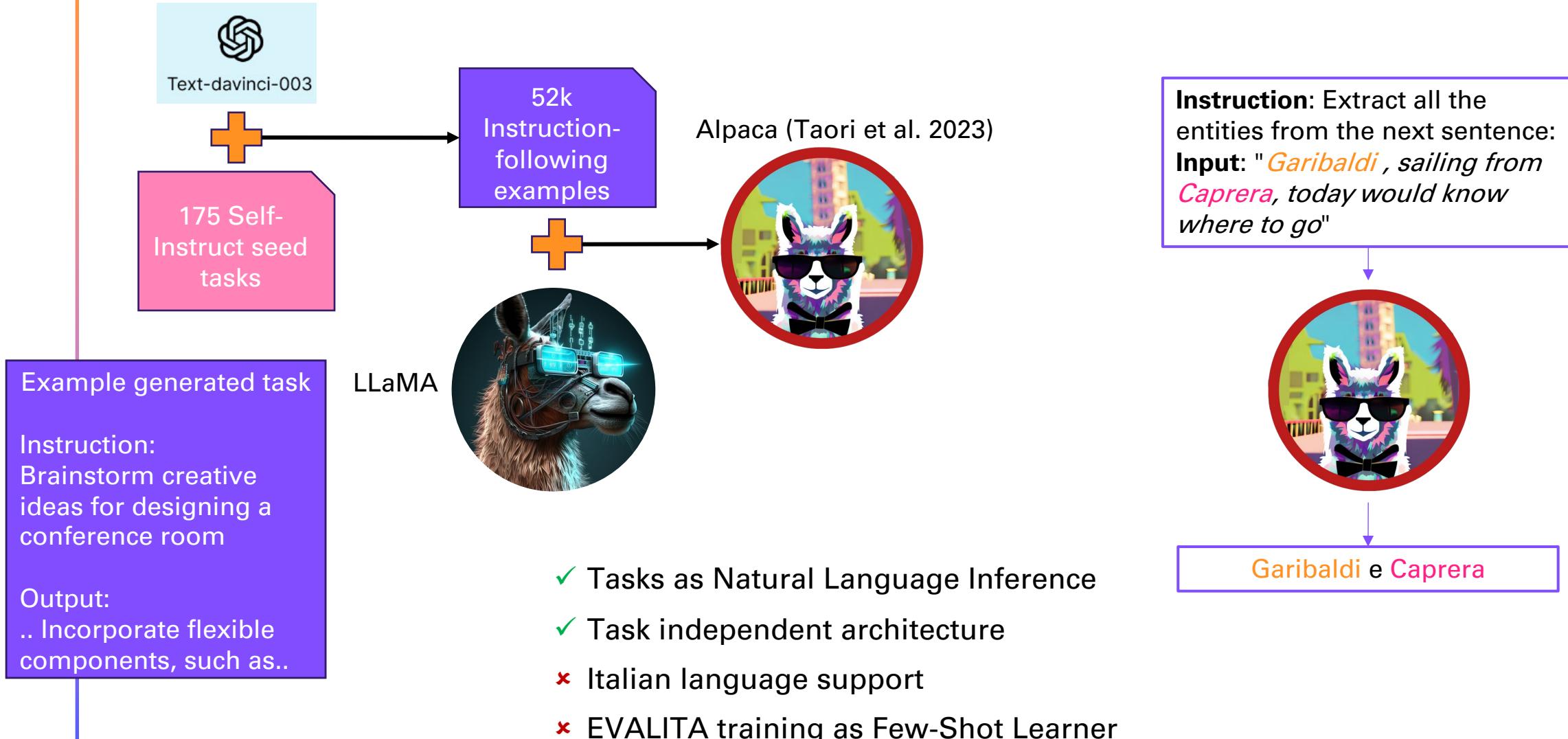


Given this sentence, please tell me what emotion it evokes between 'joy', 'sadness', ... : *"I can't wait to see you"*

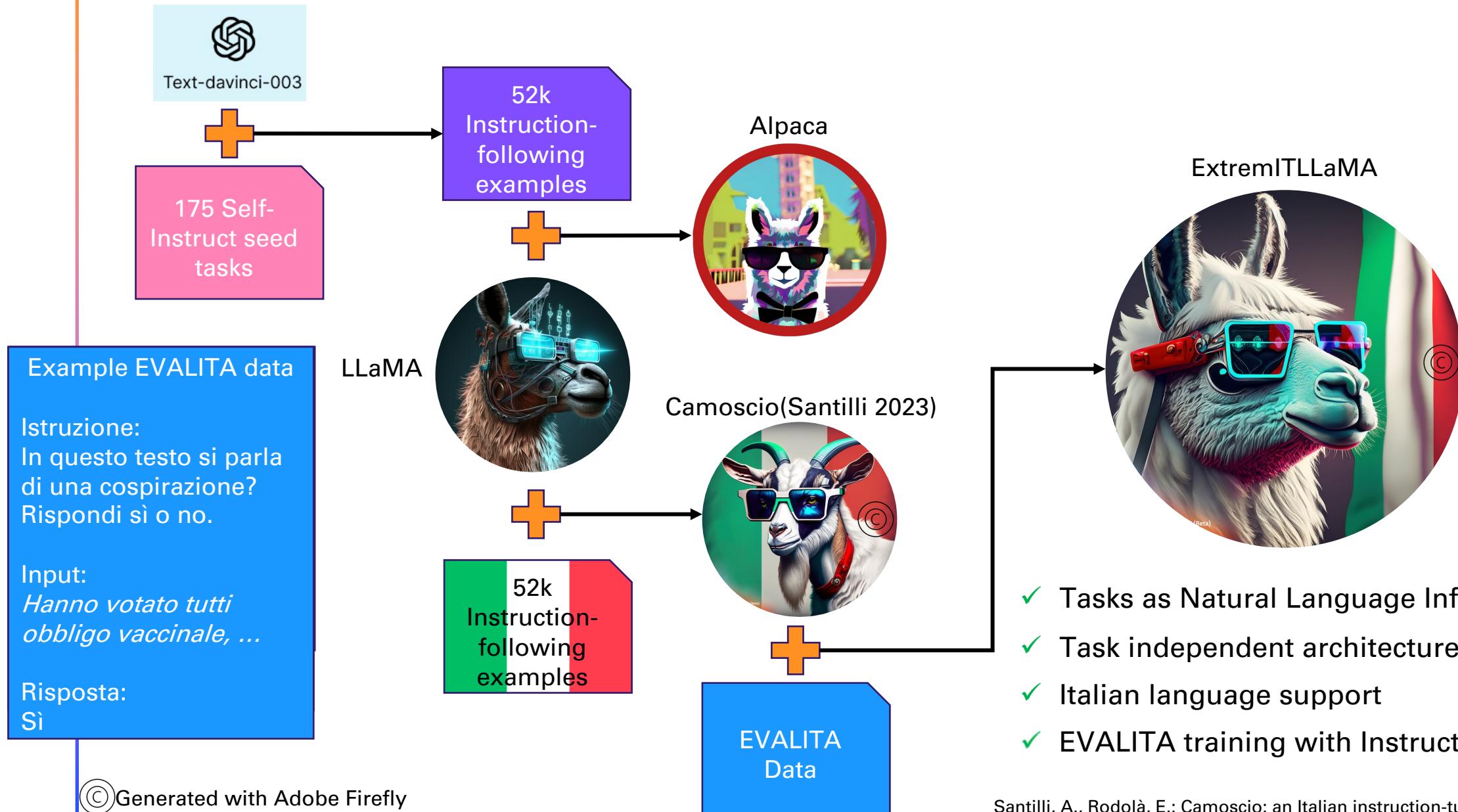


Objective: fine-tune a multitask architecture (with limited resources) to follow instructions

Large Language Models: the Camelids tree



Large Language Models: the Camelids tree



Task Prompts

Task name	Natural language instruction
CLinkaRT	Trova i risultati dei test e delle misurazioni nel testo. Per ogni risultato, scrivi '[BREL]', seguito dal risultato seguito da '[SEP]', seguito dal test, seguito da '[EREL]'. Se non trovi nessun risultato, scrivi '[NOREL]'.

Task Prompts

Task name	Natural language instruction
CLinkaRT	Trova i risultati dei test e delle misurazioni nel testo. Per ogni risultato, scrivi '[BREL]', seguito dal risultato seguito da '[SEP]', seguito dal test, seguito da '[EREL]'. Se non trovi nessun risultato, scrivi '[NOREL]'.
EMit A	Quali emozioni sono espresse in questo testo? Puoi scegliere una o più emozioni tra 'rabbia', 'anticipazione', 'disgusto', 'paura', 'gioia', 'amore', 'tristezza', 'sorpresa', 'fiducia', o 'neutro'.
EMit B	Di cosa parla il testo, tra 'direzione', 'argomento', 'entrambi', 'non specificato'
EmotivITA	Scrivi quanta valenza è espressa in questo testo su una scala da 1 a 5, seguito da quanto stimolo è espresso in questo testo su una scala da 1 a 5, seguito da quanto controllo è espresso in questo testo su una scala da 1 a 5.
PoliticIT	Scrivi se l'autore del testo è 'uomo' o 'donna', seguito dalla sua appartenenza politica tra 'destra', 'sinistra', 'centrodestra', 'centrosinistra'.
GeoLingIT	Scrivi la regione di appartenenza di chi ha scritto questo testo, seguito dalla latitudine, seguita dalla longitudine.
LangLearn	Questi due testi separati da [SEP] sono presentati nell'ordine in cui sono stati scritti? Rispondi sì o no.
HaSpeeDe 3	In questo testo si esprime odio? Rispondi sì o no.
HODI A	In questo testo si esprime odio omotransfobico? Rispondi sì o no.
HODI B	Con quali parole l'autore del testo precedente esprime odio omotransfobico? Separa le sequenze di parole con [gap].
Multifake-Detective	L'evento riportato nel testo è 'certamente vero', 'probabilmente vero', 'probabilmente falso', o 'certamente falso'?
ACTI A	In questo testo si parla di una cospirazione? Rispondi sì o no.
ACTI B	Di quale teoria cospirazionista parla questo testo, tra 'Covid', 'Qanon', 'Terrapiattista', 'Russia'?
NERMuD	Scrivi le menzioni di entità nel testo, indicandone il tipo: [PER] (persona), [LOC] (luogo), [ORG] (organizzazione).
WiC-ITA	La parola compresa tra [TGTS] e [TGTE] ha lo stesso significato in entrambe le frasi? Rispondi sì o no.
DisCoTEX 1	Le due frasi precedenti, separate da '[SEP]', sono coerenti tra loro? Rispondi sì o no.
DisCoTEX 2	Quanto è coerente questa frase, su una scala da 0 a 5?

Acceptable Answers

Task name	Output Templates
EMit A	{"Rabbia", "Anticipazione", "Disgusto", "Paura", "Gioia", "Amore", "Tristezza", "Sorpresa", "Fiducia"}+ v "Neutrale"
EMit B	{"Direzione", "Argomento", "Entrambi", "Non specificato"}
EmotivITA	"Valenza: {0-5} Stimolo: {0-5} Controllo: {0-5}"
PoliticIT	"Gender: {"Uomo", "Donna"} PIB: {"Sinistra", "Destra"} PIM: {"Sinistra", "Destra", "Centro Sinistra", "Centro Destra"}"
GeoLingIT	"Regione: {Abruzzo, .., Veneto} Latitudine: {} Longitudine: {}"
LangLearn	{"Sì", "No"}
HaSpeeDe 3	{"Sì", "No"}
HODI A	{"Sì", "No"}
HODI B	<Homotransphobia_mention>
Multifake-Detective	{"Certamente Falso", "Probabilmente Falso", "Probabilmente Vero", "Certamente Vero"}
ACTI A	{"Sì", "No"}
ACTI B	{"Terrapiattista", "Covid", "Qanon", "Russia"}
NERMuD	[<entity_type>] <text_span_that_evokes_entity>
CLinkaRT	"[BREL] <Rml_entity_mention> [SEP] <Event_entity_mention> [EREL]"
WiC-ITA	{"Sì", "No"}
DisCoTEX 1	{"Sì", "No"}
DisCoTEX 2	{0-5}

Prompting in CLinkaRT



- From a Clinical Information Extraction to a sequence to sequence generation
- Heuristics applied to reconstruct back the original form of the desired output

Istruzione: Trova i risultati dei test e delle misurazioni nel testo. Per ogni risultato, scrivi '[BREL]', seguito dal risultato seguito da '[SEP]', seguito dal test, seguito da '[EREL]'. Se non trovi nessun risultato, scrivi '[NOREL]'.

Input: [...] *Gli ESAMI di laboratorio sono normali, se si fa eccezione per una lieve anemia, così come normali sono lo STUDIO radiologico del piccolo e del grosso intestino.* [...]



[BREL] *normali* [SEP] *Gli ESAMI di laboratorio* [EREL]
[BREL] *normali* [SEP] *lo STUDIO radiologico del piccolo e del grosso intestino* [EREL]

Sustainable training with Low-Rank Adaptation (Hu et al., 2021)

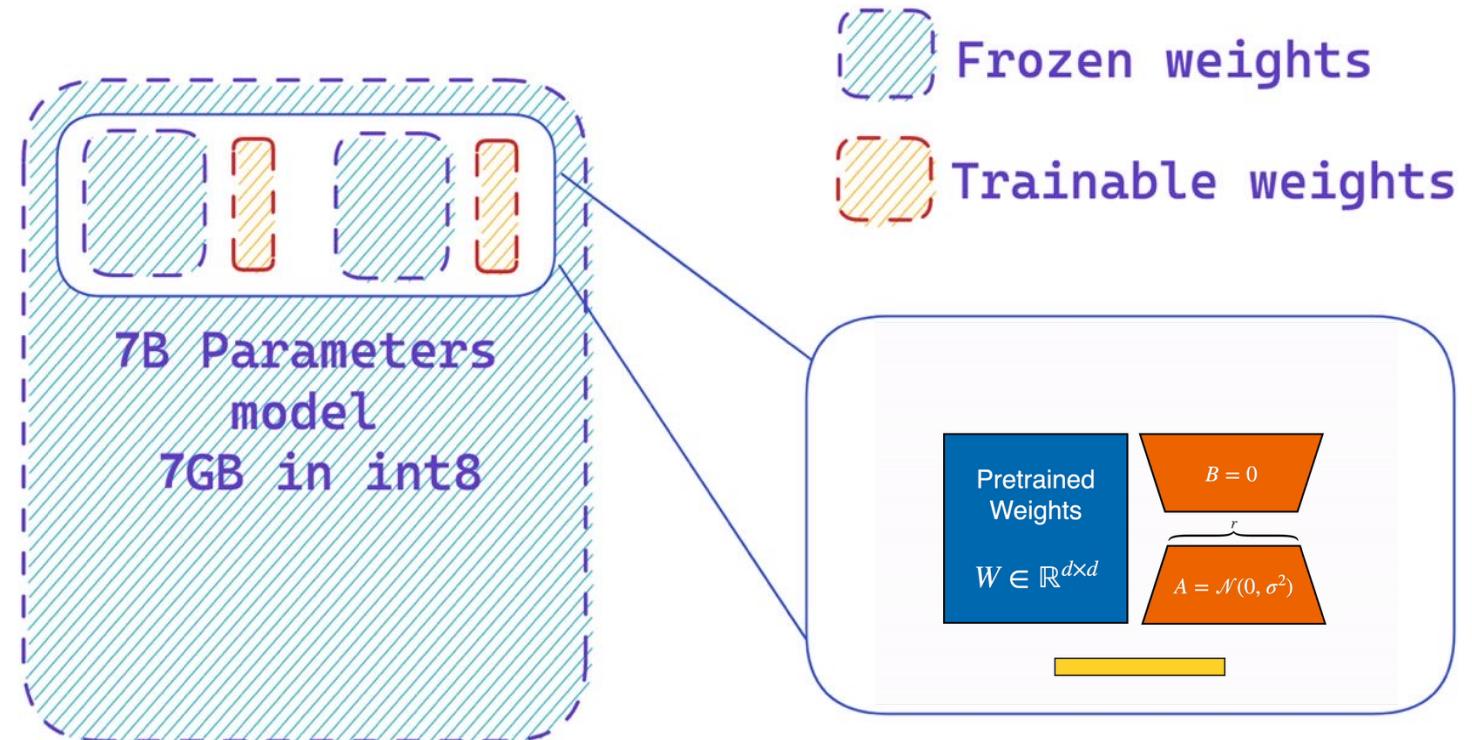
Low Rank Adaptation (LoRA: Hu et al., 2021): create the parallel (fine-tunable) adapters as smaller matrices

- add the adapters to the base model while keeping the base model frozen

We can further **scale down the memory** required by using **int8 approximation**, instead of float32

As a result, **we successfully fine-tuned a 7B parameters model with a T4 GPU** (16GB memory).

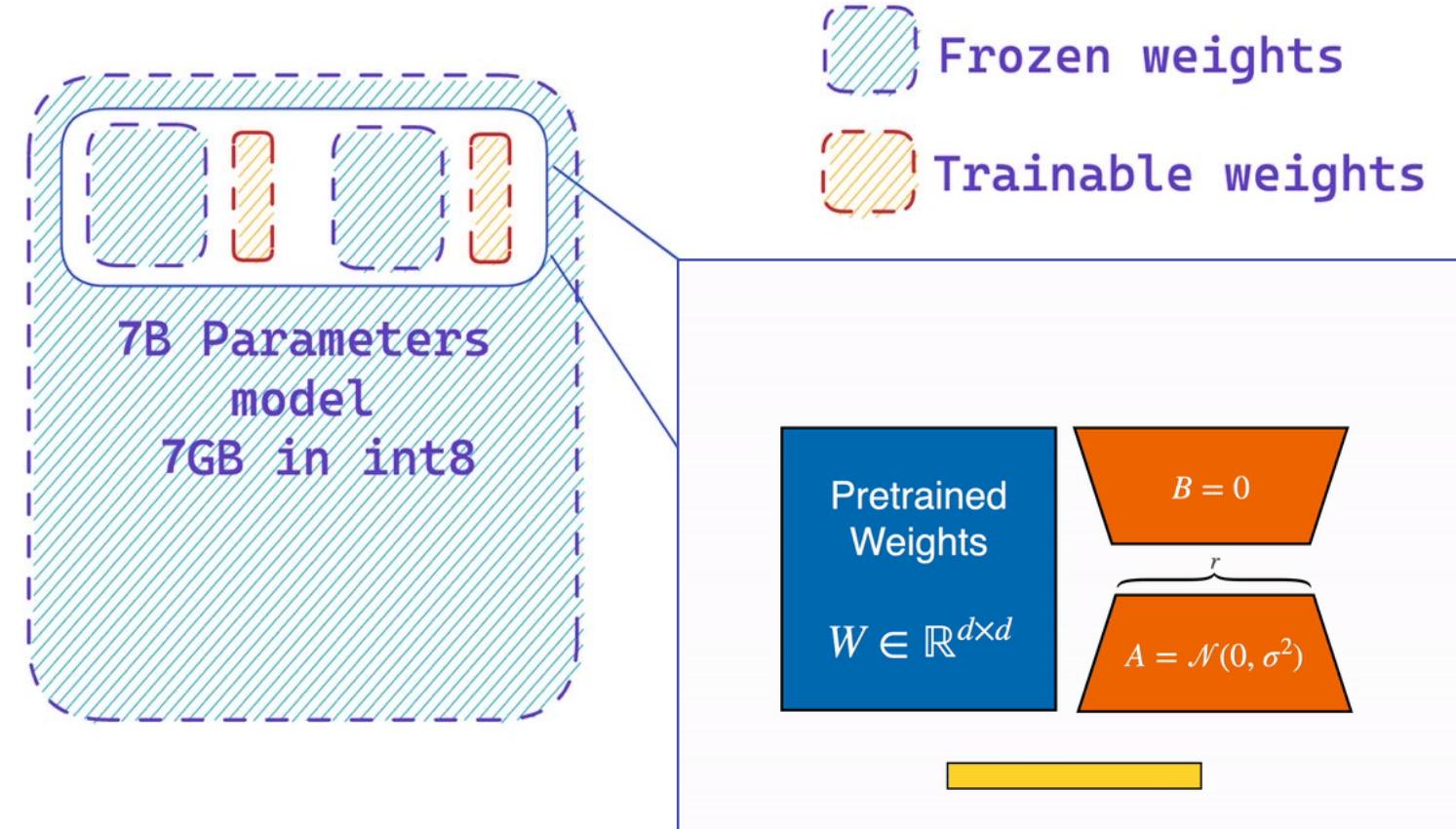
Training took 144 hours for 2 epochs on a dataset composed of 134,018 examples.



Sustainable training with Low-Rank Adaptation (Hu et al., 2021)

LoRA is **NOT** learning any parameter, but the **changes in the parameters!**

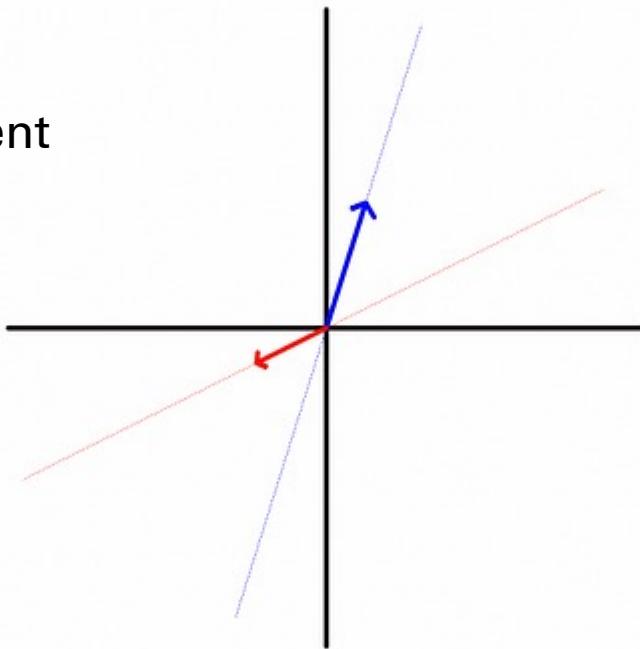
$$\underbrace{W_0 + \Delta W}_{\text{Traditional FT}} = \underbrace{W_0 + BA}_{\text{LoRA FT}}$$



Independence and matrix rank

2 linear
independent
vectors

1	3
-2	-1



r=1

A

A

r=2

B

B

Change Matrix

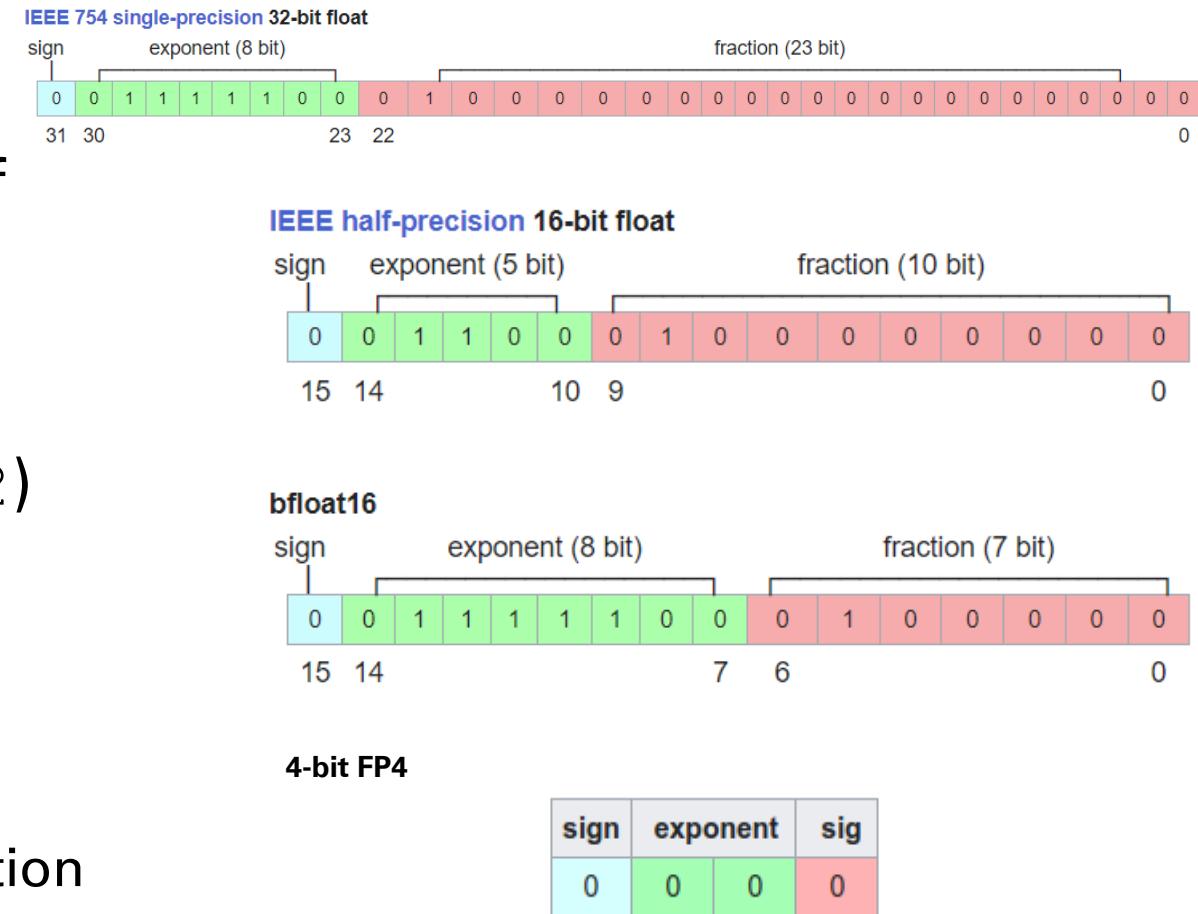
Change Matrix

$$A \times B = \begin{matrix} 2 & -3 & -6 & 5 & 7 \\ 6 & -9 & -18 & 15 & 21 \\ 8 & -12 & -24 & 20 & 28 \\ 14 & -21 & -42 & 35 & 49 \\ 18 & -27 & -54 & 45 & 63 \end{matrix}$$

=

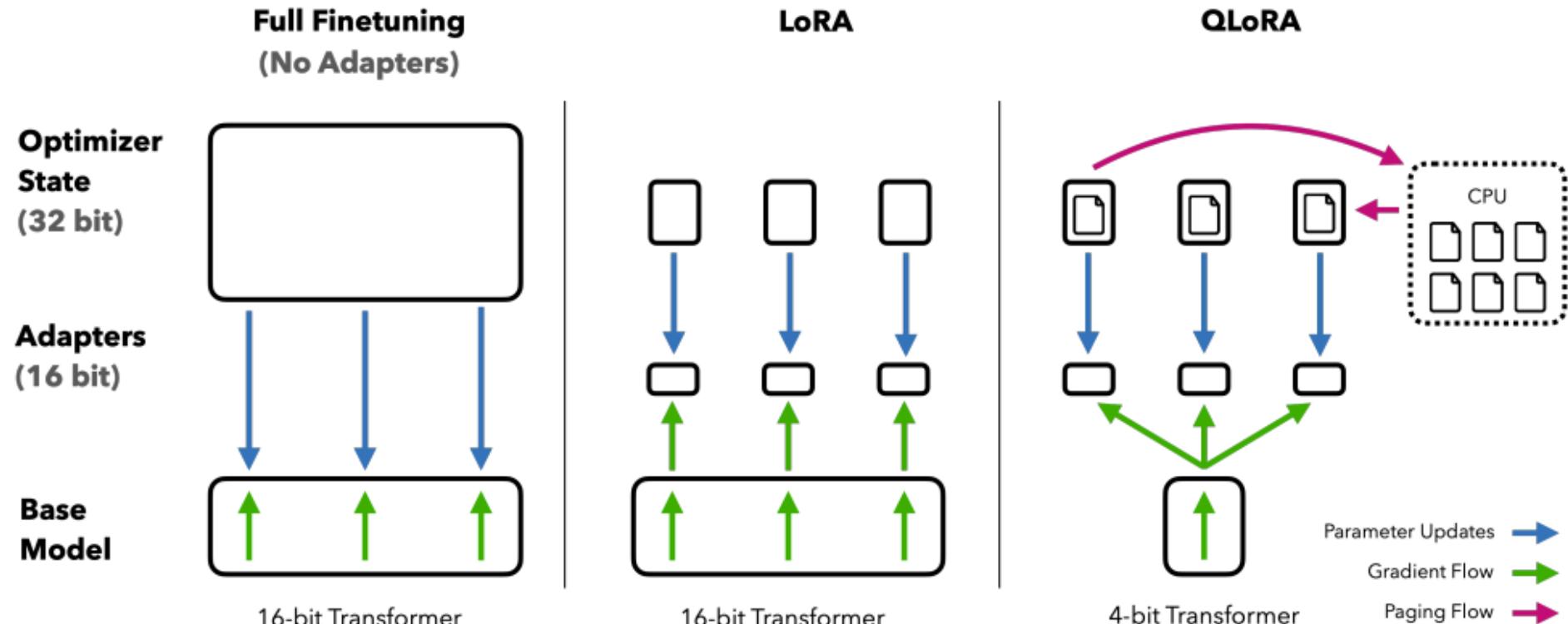
Model Quantization

- Technique to reduce the computational and memory costs of running inference
- Represents the weights and activations with low-precision data types: 32-bit floating point (`float32`)
→ 8-bit integer (`int8`)
- Advantages:
 - Requires less memory storage
 - Consumes less energy (in theory)
 - Faster operations like matrix multiplication



QLoRA: Quantized LLMs (Dettmers et al., 2023)

- 4-bit quantization to compress a pretrained language model
- Double quantization of constants (save additional 0.37 bits)
- LoRA fine-tuning
- Paged optimizers for memory spikes (drop on CPUs)



Results

Task name	Eval Metric	ExtremITLLaMA Score	ExtremITLLaMA Rank	Best Competitor Score	Best Competitor Rank
EMit	F1	0.6028 0.6459	1 1	0.4994 0.6184	3 3
EmotivITA	Pears Val	0.8110		0.8110	
	Pears Aro	0.6330	1	0.6520	2
	Pears Dom	0.6300		0.6540	
PoliticIT	F1	0.7719	3	0.8241	1
GeoLingIT	F1	0.3818	11	0.6630	1
	Avg Km	145.15	9	97.74	1
LangLearn	F1	0.5500	8	0.7500	1
	F1	0.6100	8	0.9300	1
HaSpeeDe 3	F1 – text/context/xPolitic	0.9034	3	0.9128	1
	F1 – xRel	0.6525	1	0.6461	2
HODI	F1	0.7942	5	0.8108	1
	F1	0.7228	1	0.7051	2
Multifake-Detective	F1	0.5070	2	0.5120	1
	F1	0.4640	1	0.4600	2
ACTI	F1	0.8565	2	0.8571	1
	F1	0.8556	5	0.9123	1
NERMuD	F1	0.8900	1	na	na
CLinkaRT	F1	0.5916	2	0.6299	1
WiC-ITA	F1 it-it	0.5100	10	0.7300	1
	F1 it-en	0.5400	8	0.7400	1
DisCoTEX	Acc	0.8150	1	0.7200	2
	HM*	0.6500	1	0.6300	2

Results

Task name	Eval Metric	ExtremITLLaMA Score	ExtremITLLaMA Rank	Best Competitor Score	Best Competitor Rank
EMit	F1	0.6028 0.6459	1 1	0.4994 0.6184	3 3
EmotivITA	Pears Val	0.8110		0.8110	
	Pears Aro	0.6330	1	0.6520	2
	Pears Dom	0.6300		0.6540	
PoliticIT	F1	0.7719	3	0.8241	1
GeoLingIT	F1	0.3818	11	0.6630	1
	Avg Km	145.15	9	97.74	1
LangLearn	F1	0.5500	8	0.7500	1
	F1	0.6100	8	0.9300	1
HaSpeeDe 3	F1 – text/context/xPolitic	0.9034	3	0.9128	1
	F1 – xRel	0.6525	1	0.6461	2
HODI	F1	0.7942	5	0.8108	1
	F1	0.7228	1	0.7051	2
Multifake-Detective	F1	0.5070	2	0.5120	1
	F1	0.4640	1	0.4600	2
ACTI	F1	0.8565	2	0.8571	1
	F1	0.8556	5	0.9123	1
NERMuD	F1	0.8900	1	na	na
CLinkaRT	F1	0.5916	2	0.6299	1
WiC-ITA	F1 it-it	0.5100	10	0.7300	1
	F1 it-en	0.5400	8	0.7400	1
DisCoTEX	Acc	0.8150	1	0.7200	2
	HM*	0.6500	1	0.6300	2

Results

Task name	Eval Metric	ExtremITLLaMA Score	ExtremITLLaMA Rank	Best Competitor Score	Best Competitor Rank
EMit	F1	0.6028 0.6459	1 1	0.4994 0.6184	3 3
EmotivITA	Pears Val	0.8110	1	0.8110	2
	Pears Aro	0.6330		0.6520	
	Pears Dom	0.6300		0.6540	
PoliticIT	F1	0.7719	3	0.8241	1
GeoLingIT	F1	0.3818	11	0.6630	1
	Avg Km	145.15	9	97.74	1
LangLearn	F1	0.5500	8	0.7500	1
	F1	0.6100	8	0.9300	1
HaSpeeDe 3	F1 – text/context/xPolitic	0.9034	3	0.9128	1
	F1 – xRel	0.6525	1	0.6461	2
HODI	F1	0.7942	5	0.8108	1
	F1	0.7228	1	0.7051	2
Multifake-Detective	F1	0.5070	2	0.5120	1
	F1	0.4640	1	0.4600	2
ACTI	F1	0.8565	2	0.8571	1
	F1	0.8556	5	0.9123	1
NERMuD	F1	0.8900	1	na	na
CLinkaRT	F1	0.5916	2	0.6299	1
WiC-ITA	F1 it-it	0.5100	10	0.7300	1
	F1 it-en	0.5400	8	0.7400	1
DisCoTEX	Acc	0.8150	1	0.7200	2
	HM*	0.6500	1	0.6300	2

Prompting in NERMuD

- From a token classification problem to a sequence to sequence generation
- Heuristics applied to reconstruct back the original form of the desired output

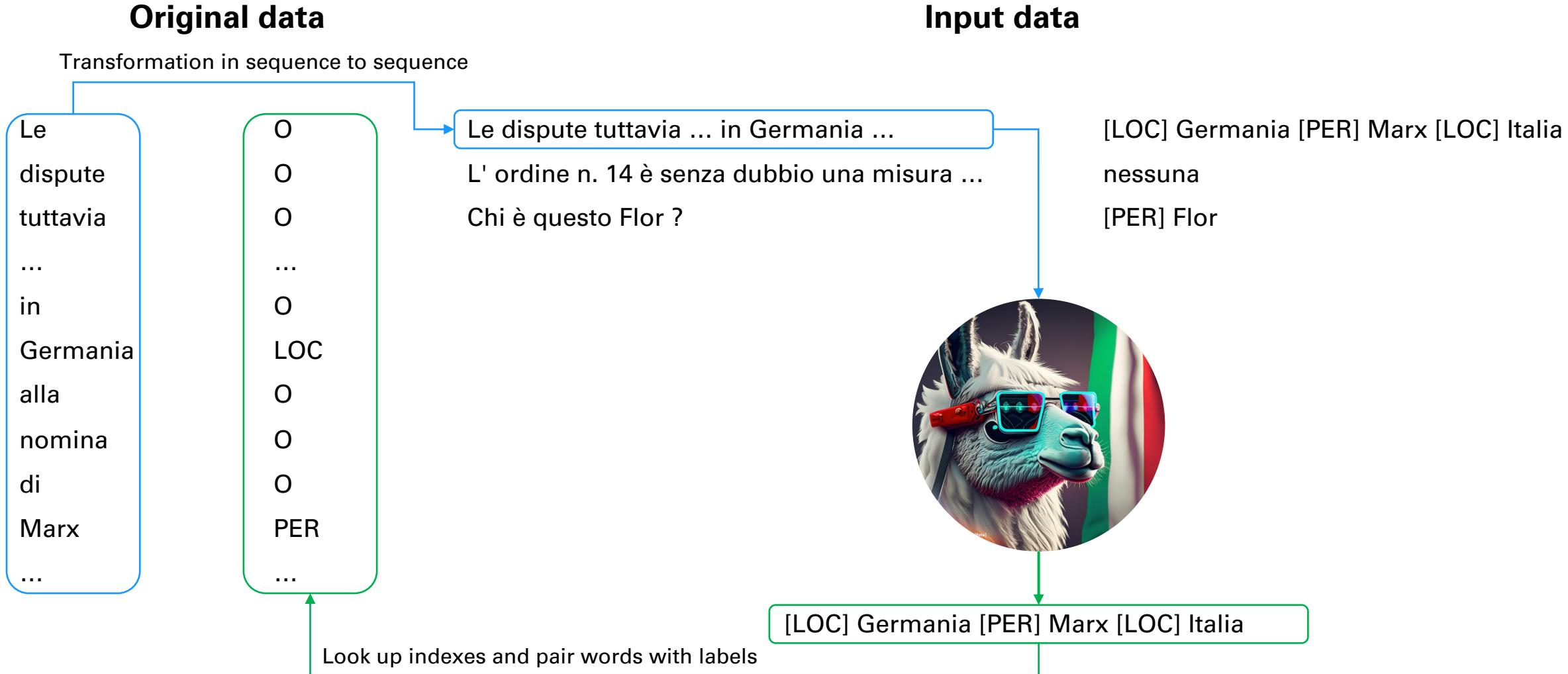
Istruzione: Scrivi le menzioni di entità nel testo, indicandone il tipo: [PER] (persona), [LOC] (luogo), [ORG] (organizzazione).

Input: *Le dispute tuttavia fra cattolici non cessarono [...] gli argomenti che furono opposti in Germania alla nomina di Marx e [...] collaborazione coi socialisti vennero fatte in Italia.*



[LOC] Germania [PER] Marx
[LOC] Italia

Construction of NERMuD data



NERMuD: error analysis

Text	Gold Standard	ExtremITLLaMA
Informa il Consiglio che Nenni gli ha chiesto se il governo porrà ostacoli all' ingresso in Italia di dieci membri dell' Esecutivo internazionale dei partigiani della pace .	[ORG] Consiglio [PER] Nenni [LOC] Italia [ORG] Esecutivo internazionale dei partigiani della pace	[ORG] Consiglio [PER] Nenni [LOC] Italia ✓
Occorre ricordare che l' America vincerà , passerà molto tempo ma vincerà e questa adesione nostra salva il futuro dell' Italia .	[ORG] America [ORG] Italia	[LOC] America [LOC] Italia ✓
Intervento al Senato della Repubblica	[ORG] Senato [ORG] Repubblica	[ORG] Senato della Repubblica ✗
Bisognava capire una buona volta che con un Governo come il nostro , con partiti nemici come abbiamo noi , a dir « tutto o nulla » ci restava e ci resterà sempre la seconda parte .	nessuna	[ORG] Governo ✗
Netflix rimuove un documentario sull' AIDS a seguito di alcune proteste	nessuna	[ORG] Netflix ✓
Russia	[ORG] Russia	[LOC] Russia ?



DisCoTEX: prompts and examples

Task	Instruction
DisCoTEX 1	Le due frasi precedenti, separate da '[SEP]', sono coerenti tra loro? Rispondi sì o no.
DisCoTEX 2	Quanto è coerente questa frase, su una scala da 0 a 5?

Istruzione: Le due frasi precedenti, separate da '[SEP]', sono coerenti tra loro? Rispondi sì o no.

Input: "Ah... domani piove..." Mi asseendarono come si fa coi pazzi. In effetti io mi sentivo completamente folle. Non ero molto convinta. [SEP] Perché il movimento trasforma, impregna, informa di sé.

Istruzione: Quanto è coerente questa frase, su una scala da 0 a 5?.

Input: Lo fa allo stadio. Se lo fa alle poste è una cosa un po' strana. Ma non è l'esercizio il problema, è il contesto. Nel contesto giusto, tutti insieme per 30 secondi.



No

3.0



DiscoTEX: Error analysis

First Sentence	Second Sentence	Gold Standard	ExtremITLLaMA
Quando la paura non supera l'incontro? Quando ciò non lo consideriamo normale. Ma esiste la normalità?	E, quindi, ritornando un po' indietro a quando rispondere a quei bimbi è complicato, e deve essere per forza una bugia.	Sì	No
Per l'orografia del luogo il lato secondario ha un piano in più. Questo lato è in laterizio con tre fornici. L'ingresso è incorniciato da lesene e posto tra due aperture.	L'aspetto è molto monumentale, ha ordine regolare di finestre architravate con timpano curvilineo, e finto bugnato agli angoli.	No	Sì

LangLearn: prompts and examples

Task name	Natural language instruction
LangLearn	Questi due testi separati da [SEP] sono presentati nell'ordine in cui sono stati scritti? Rispondi sì o no.

Istruzione: Questi due testi separati da [SEP] sono presentati nell'ordine in cui sono stati scritti? Rispondi sì o no.
Input: *Io e Luca siamo molto amici, ci conosciamo dall' asilo, siamo molto legati infatti certa gente ci invidia per la nostra grande amicizia. [SEP] Sono pronto a calciare la palla mi tremano le mani. Sono eccitato. Calcio, sento. Uno strappo alla gamba. Cado a Terra. Fuori la palla.*



Sì

LangLearn: error analysis

First Sentence	Second Sentence	Gold Standard	ExtremITLLaMA
I gruppi sono formati da amici, stretti tra loro da un legame di amicizia pura. Purtroppo questo legame anche quando è ben " saldo", anche per causa di un piccolo scherzo ..., il gruppo anche per il comportamento non adeguato di una persona potrebbe sciogliersi.	Sin dall'inizio l'uomo è nomade, si sposta in cerca di animali -per poi ricavarne carne, latte e pellicce- e in cerca di frutti e radici. Con il tempo il territorio sfruttato al massimo, si impoverì. ... Le tribù così diventarono sedentarie; cominciarono ad esplorare i territori dove vi erano insediati.	Sì	No
Il mio fratellino ha gli occhi azzurri come il cielo, dei capelli gialli come il sole, e la pelle bianca come il ghiaccio.	Un giorno due bambini si avventurarono in un bosco. Camminando i bambini si accorsero che si stava facendo buglio , ... Entrarono e videro otto bare, scapparono al piano di sopra e dal buglio spuntarono un esercito di zombi non morti.	No	Sì



MultiFake-Detective: prompts and error analysis

Task name	Natural language instruction
MultiFake-Detective	L'evento riportato nel testo è 'certamente vero', 'probabilmente vero', 'probabilmente falso', o 'certamente falso'?

Task name	Output Templates
MultiFake-Detective	{"Certamente Falso", "Probabilmente Falso", "Probabilmente Vero", "Certamente Vero"}

Input text	Gold Standard	ExtremITLLaMA
E niente gli sputa inchiostro dell' AnZa non riescono a dare una notizia senza cadere nella subdola propaganda. Il contestato è Letta colpevole di favorire l'invio di armi all'Ucraina e loro pubblicano una foto contro Putin. Buffoni!!! #buonafestadellaliberazione https://t.co/wh9fY0UeVN	Probabilmente vero	Probabilmente Falso ?

Conclusions



9/22
(41%)



14/22
(64%)

- ✓ One architecture for all tasks
- ✓ Task-independent architecture
- ✓ Sequence to sequence
- ✓ Excellent performance
- ✓ Straightforward NL prompts
- ✓ Task dependent prompts
- ✓ Code on GitHub and models on HuggingFace

- ✗ Data hungry
- ✗ GPUs and time hungry
- ✗ Non satisfactory performance on all tasks:
 - ✗ GeoLingIT
 - ✗ LangLearn
 - ✗ WiC-ITA

Now: Let's fine-tune an LLM

The objective is to take some data, transform it into a sequence to sequence format and train the model.

The tutorial is split into 4 steps, reflecting the aforementioned process:

- Step 1 - Encoding the data
- Step 2 - Training the LLaMA model
- Step 3 - Inference: generating answers
- Step 4 - Decoding the data

LET'S CODE!

https://github.com/crux82/CLiC-it_2023_tutorial



Generated using GPT4



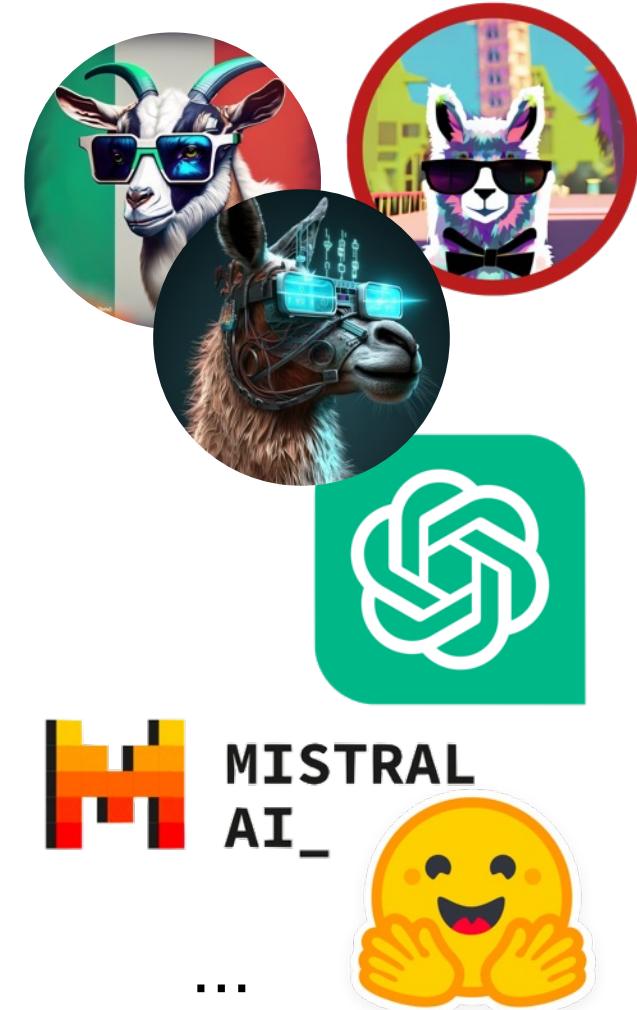
The wave of other «camelids»

Instruction-Tuned Models:

- [Vicuna](#) (03/2023)
- [ColossalChat](#) (03/2023)
- [Koala](#) (04/2023)
- [Guanaco](#) (05/2023)
- [Orca](#) (06/2023)
- [LLaMA2Chat](#) (07/2023)
- [OpenChat](#) (07/2023)
- [Code-LLaMA](#) (08/2023)
- [Orca2](#) (11/2023)

Large Language Models:

- [Colossal-LLaMA-2](#) (09/2023)
- [Falcon](#) (05/2023)
- [LLaMA2](#) (07/2023)
- [Mistral](#) (09/2023)
- [Zephyr](#) (10/2023)
- [CausalLM](#) (10/2023)



A complete bibliography on LLMs

<https://github.com/RUCAIBox/LLMSurvey>

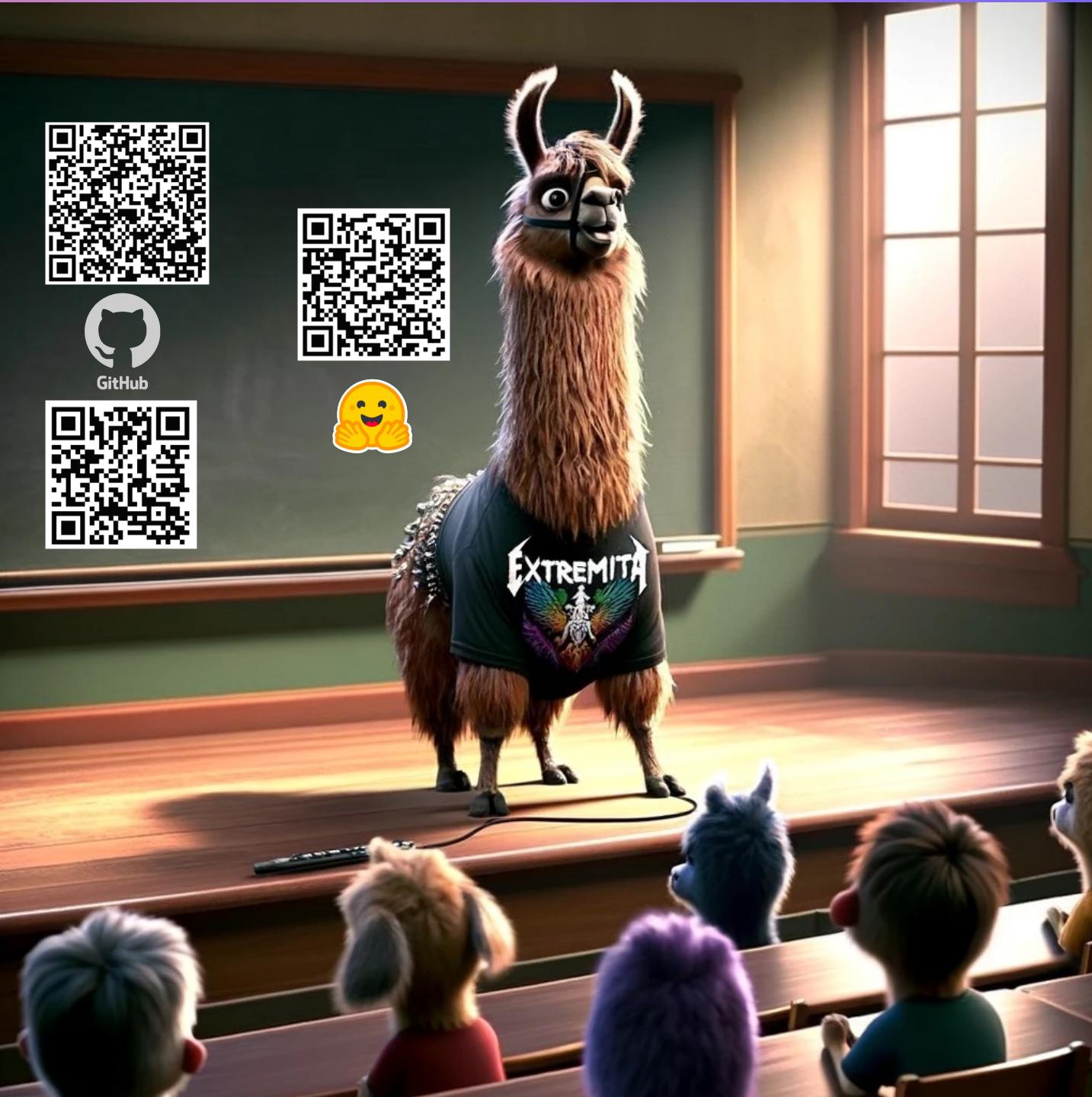
<https://arxiv.org/pdf/2303.18223.pdf>

1

A Survey of Large Language Models

Wayne Xin Zhao, Kun Zhou*, Junyi Li*, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie and Ji-Rong Wen

Abstract—Ever since the Turing Test was proposed in the 1950s, humans have explored the mastering of language intelligence by machine. Language is essentially a complex, intricate system of human expressions governed by grammatical rules. It poses a significant challenge to develop capable artificial intelligence (AI) algorithms for comprehending and grasping a language. As a major approach, *language modeling* has been widely studied for language understanding and generation in the past two decades, evolving from statistical language models to neural language models. Recently, pre-trained language models (PLMs) have been proposed by pre-training Transformer models over large-scale corpora, showing strong capabilities in solving various natural language processing (NLP) tasks. Since the researchers have found that model scaling can lead to an improved model capacity, they further investigate the scaling effect by increasing the parameter scale to an even larger size. Interestingly, when the parameter scale exceeds a certain level, these enlarged language models not only achieve a significant performance improvement, but also exhibit some special abilities (e.g., in-context learning) that are not present in small-scale language models (e.g., BERT). To discriminate the language models in different parameter scales, the research community has coined the term *large language models (LLM)* for the PLMs of significant size (e.g., containing tens or hundreds of billions of parameters). Recently, the research on LLMs has been largely advanced by both academia and industry, and a remarkable progress is the launch of ChatGPT (a powerful AI chatbot developed based on LLMs), which has attracted widespread attention from society. The technical evolution of LLMs has been making an important impact on the entire AI community, which would revolutionize the way how we develop and use AI algorithms. Considering this rapid technical progress, in this survey, we review the recent advances of LLMs by introducing the background, key findings, and mainstream techniques. In particular,



THANK YOU
FOR THE
ATTENTION

CONTACTS

croce@info.uniroma2.it

hromei@ing.uniroma2.it

<https://github.com/crux82/ExtremITA>

https://github.com/crux82/CLiC-it_2023_tutorial