Institute of Electronics
National Yang Ming Chiao Tung University
Hsinchu, Taiwan

AI Training Course Series

Introduction to Large Language Model

Lecture 10

Student: Zhi-Kai Xu, Ting-Yu Chang
Advisor: Juinn-Dar Huang, Ph.D.

August 15, 2024

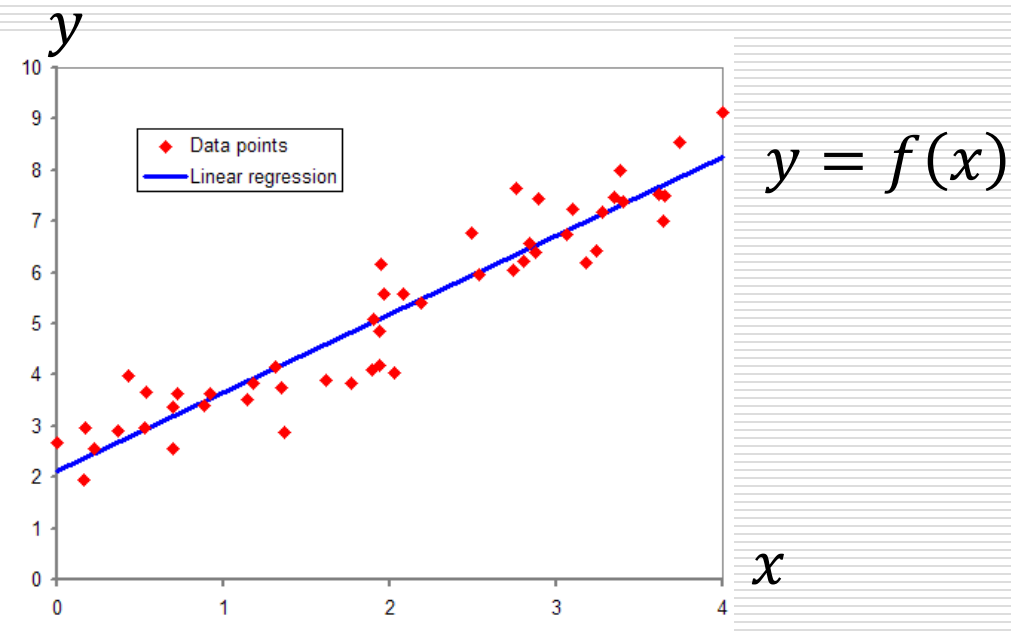
Outline

- Language Model
- GPT Series
- Corpora
- Positional Encoding and RoPE
- LLaMA Series
- Huggingface Environment
- Exercise
- Reference

Language Model

What We Learned Before (1/2)

- What is Machine Learning?
  - Let machine find out a function from data
- Given an input, a function maps it to an output.



$$y = f(x)$$

- Training is to find out such a function.

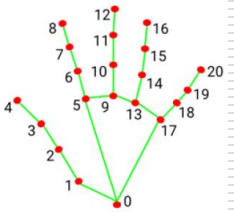
What We Learned Before (2/2)

- A model is a function.

$x \longrightarrow f \longrightarrow y = f(x)$

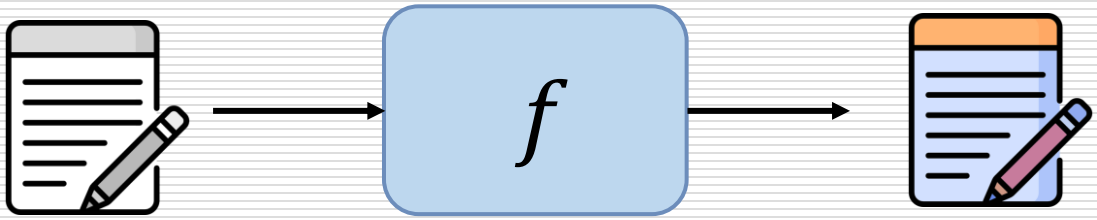
$x =$  $y = f($  $) =$ dog

$x =$  $y = f($  $) =$ “Hi …”

$x =$  $y = f($  $) =$ 

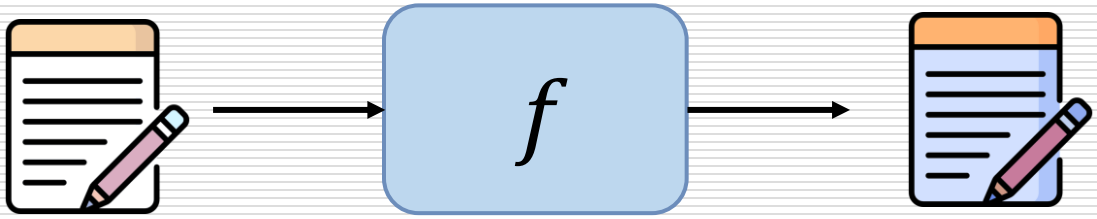
What is a Language Model

- A model designed to understand, generate, and manipulate human language.



x = Hello.	y = Hi, how are you?
x = What is quantum physics?	y = Quantum physics is…
x = Write a story.	y = Once upon a time…

How it does?



- By 文字接龍

f (Hello.) = Hi,

f (Hello. Hi,) = how

f (Hello. Hi, how) = are

f (Hello. Hi, how are) = you

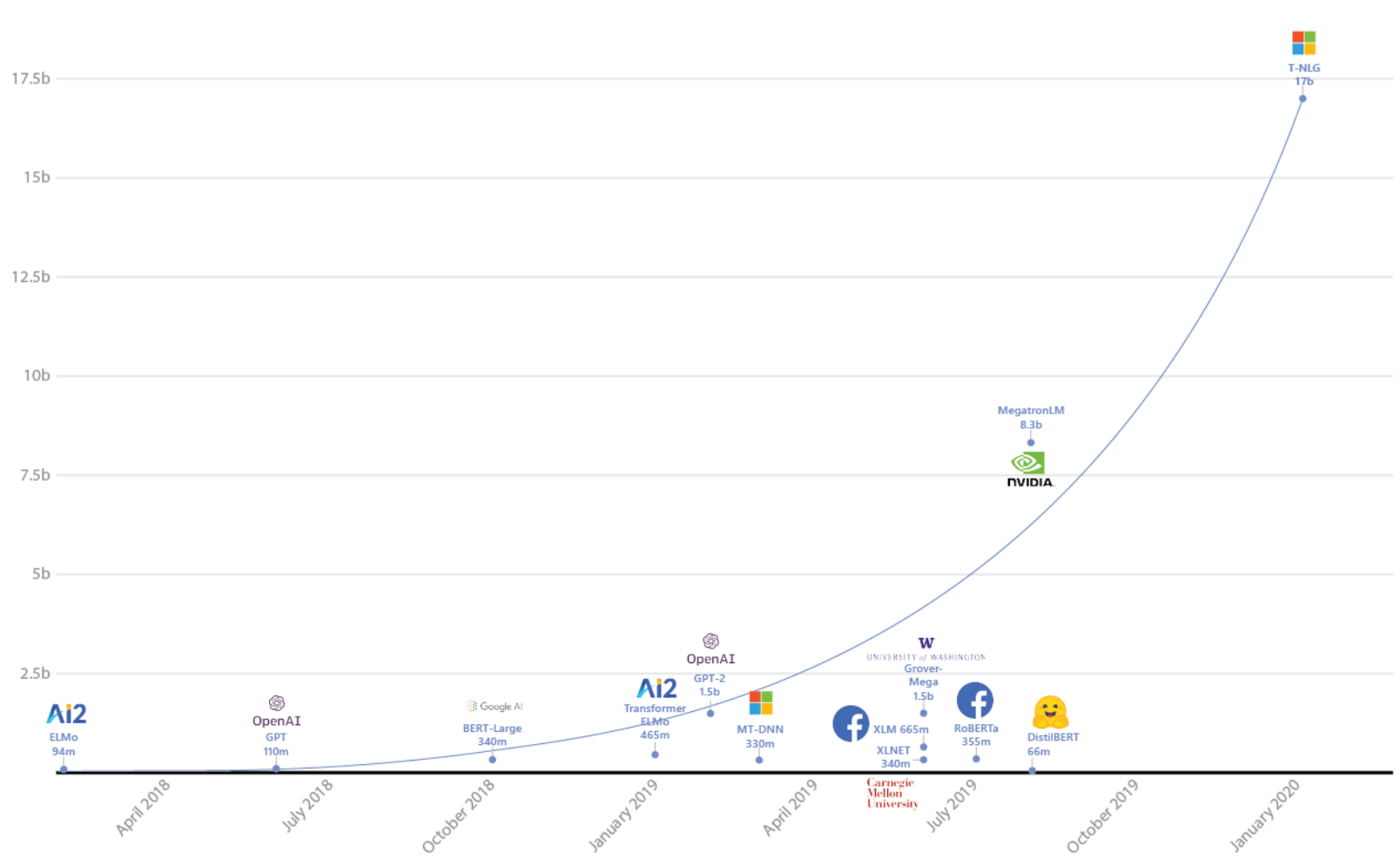
f (Hello. Hi, how are you) = ?

f (Hello. Hi, how are you?) = EOS

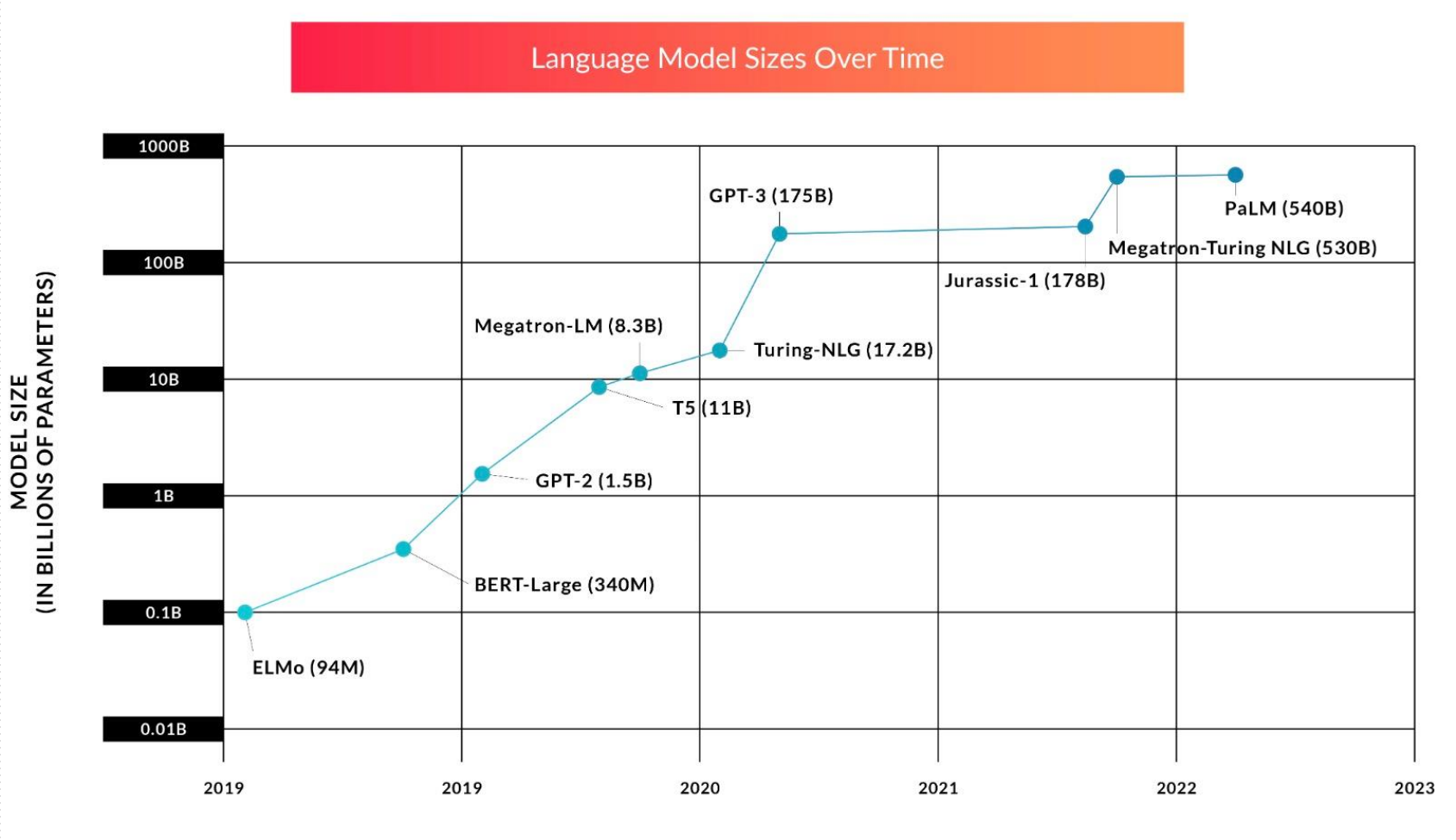
Can be formulated as the product of conditional probabilities

$$p(x) = \prod_{i=1}^{n} p(s_n | s_1, \ldots, s_{n-1})$$

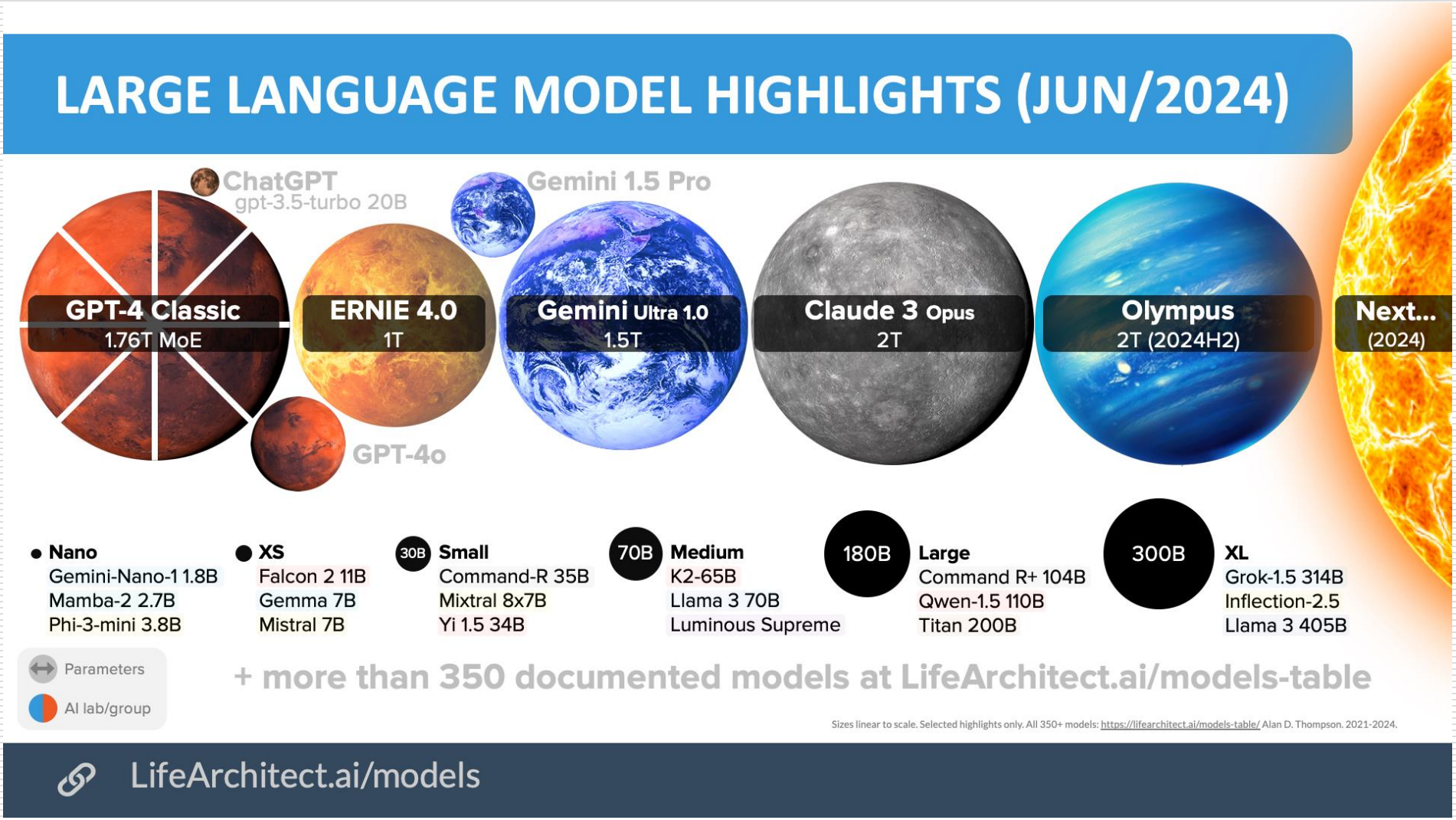
Why “Large”? (Jan 2020)



Larger (Till 2023)



Even Larger (Till Jun 2024)



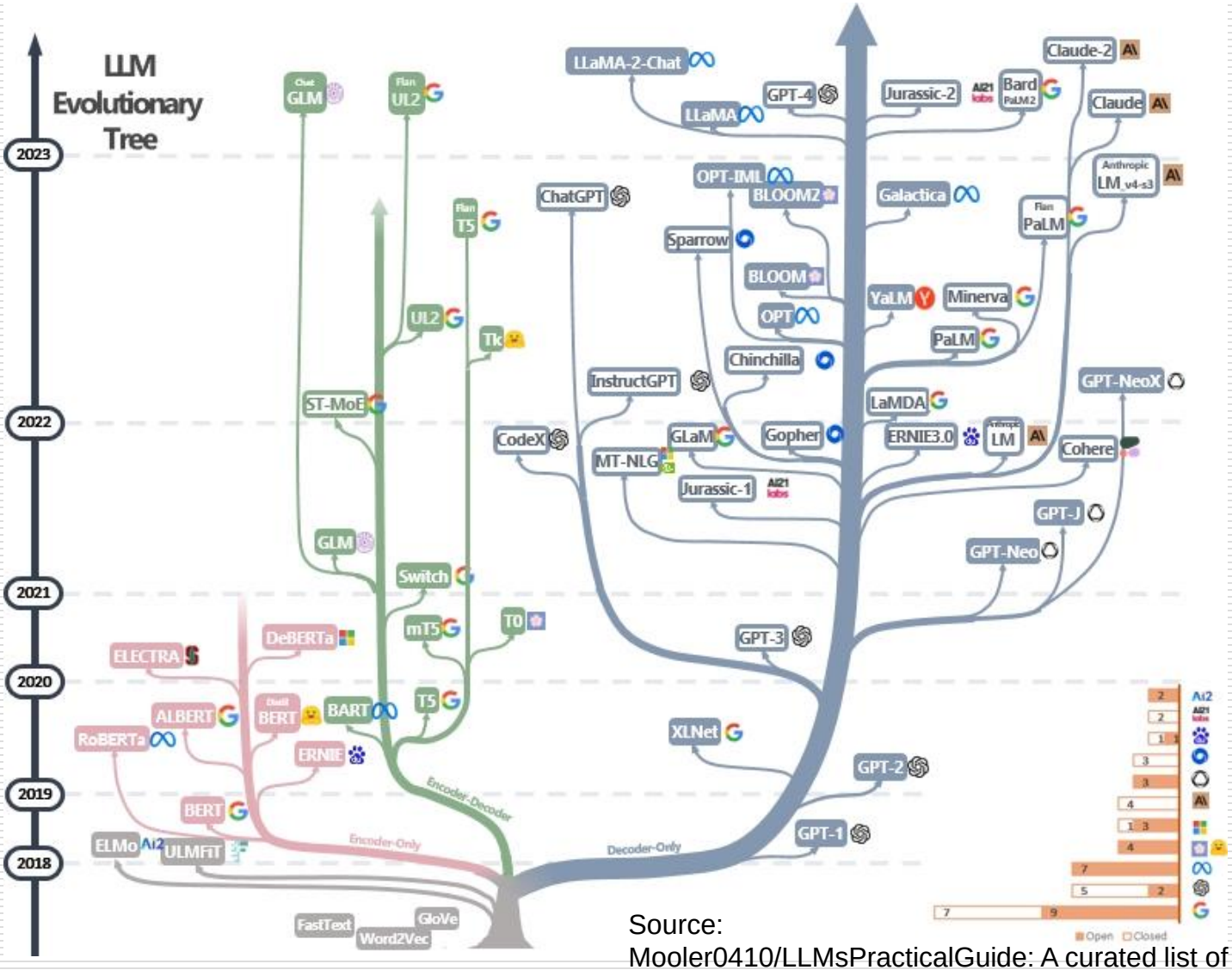
Large Language Model (LLM)

- LLM is a very novel concept under intensive research.
- No formal definition yet
- GPT-3, developed by OpenAI in 2020, is often regarded as the first LLM, at model size of 175B.
- Compared to earlier language models, they exhibit many emergent abilities previous not thought possible

Classification of Language Model (LM)

- Causal Language Model (CLM)
  - Autoregressive (AR)
  - Used by GPT
- Masked Language Model (MLM)
  - Autoencoder (AE)
  - Used by BERT

LLM Roadmap



Source:
Mooler0410/LLMsPracticalGuide: A curated list of practical guide resources of LLMs (LLMs Tree, Examples, Papers) (github.com)

GPT Series

OpenAI

- Founded by Sam Altmen et al. (Elon Musk, Ilya Sutskever, Peter Thiel) in 2015 as a non-profit organization
- Dedicated to friendly AI, open patent, open research, and sharing results with the public
- Major works including DALL-E image generation models and GPT language models
- Transition to limited for-profit in 2019 and invested by Microsoft, not so “open” now… ☹

GPT Series

Source: GPT Models: How Will They Be Used for AI Chatbots in 2024? – Dataquest

GPT–1

Introduction

- **Improving Language Understanding through Generative Pre-Training**
- Alec Radford et al.
- Self-published on OpenAI website in 2018
- Pretrained weight is open for public use

Insights

- Introducing the pretrain-finetune paradigm and showing its potential in NLP application
- Popularizing the adoption of transformer structure and generalizing its application from machine translation to other NLP tasks.
- The most influential model to leverage the decoder-only architecture in language modeling.

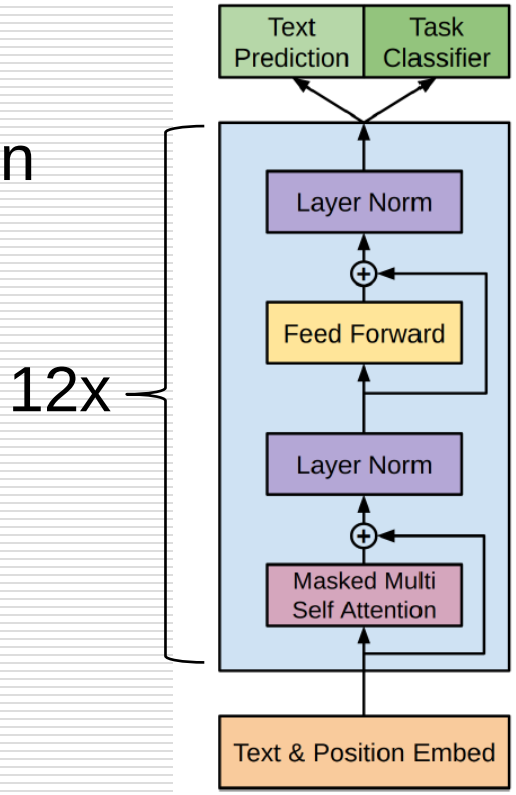
In the paper,

In our experiments, we use a multi-layer *Transformer decoder* [34] for the language model, which is a variant of the transformer [62]. This model applies a multi-headed self-attention operation over the input context tokens followed by position-wise feedforward layers to produce an output distribution

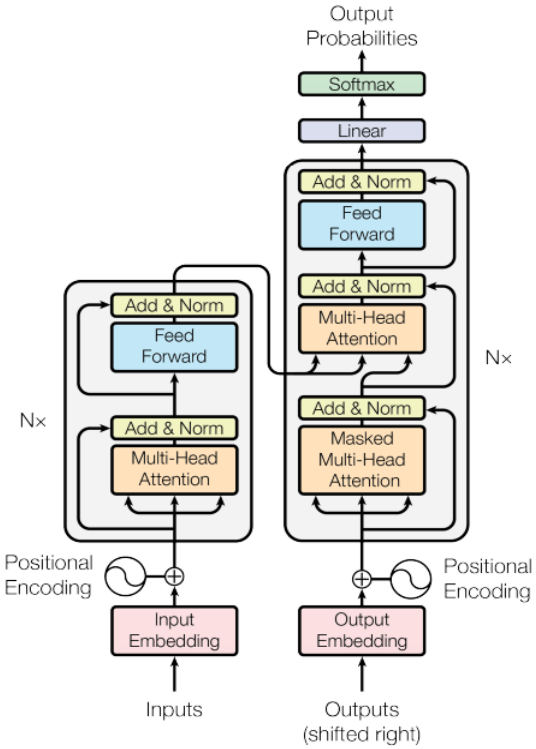
"Attention Is All You Need", Vaswani et al., 2018

Model Structure

- A decoder-only architecture
- 12 layers of Transformer decoder
- # of param = 117M
- Autoregressive
  - Masked Self-Attention

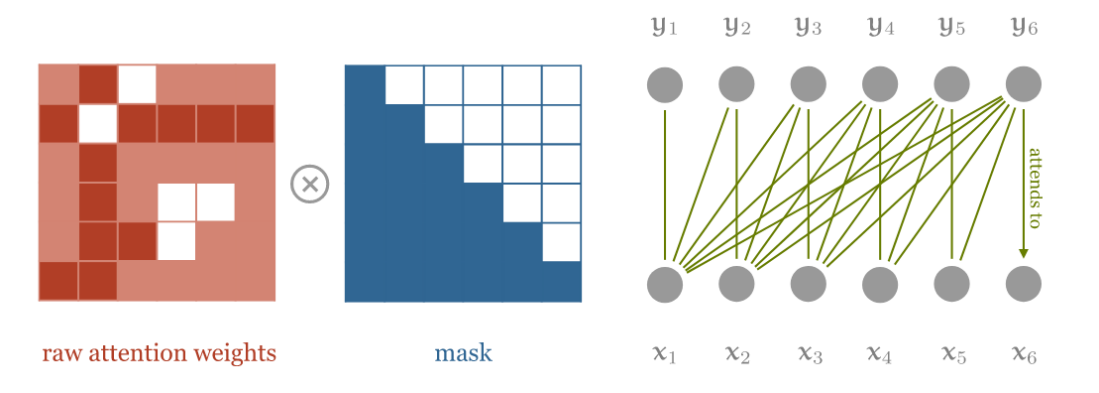


Compared with the original Transformer:



What is Different from a Full Transformer?

- The decoder-only architecture looks almost identical to an encoder, except its attention is causally masked.
- Behaviors
  - of the original encoder: summarization stage
  - of the original decoder: generation stage
- The both are autoregressive.

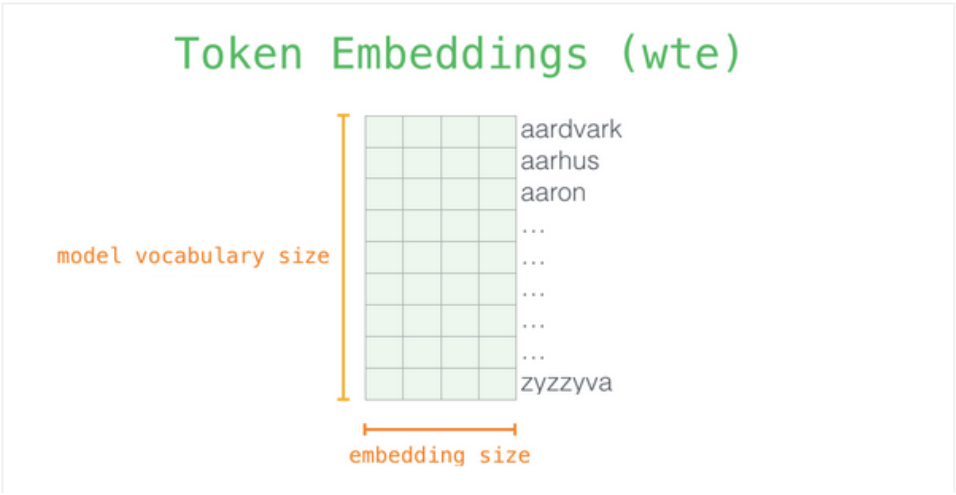


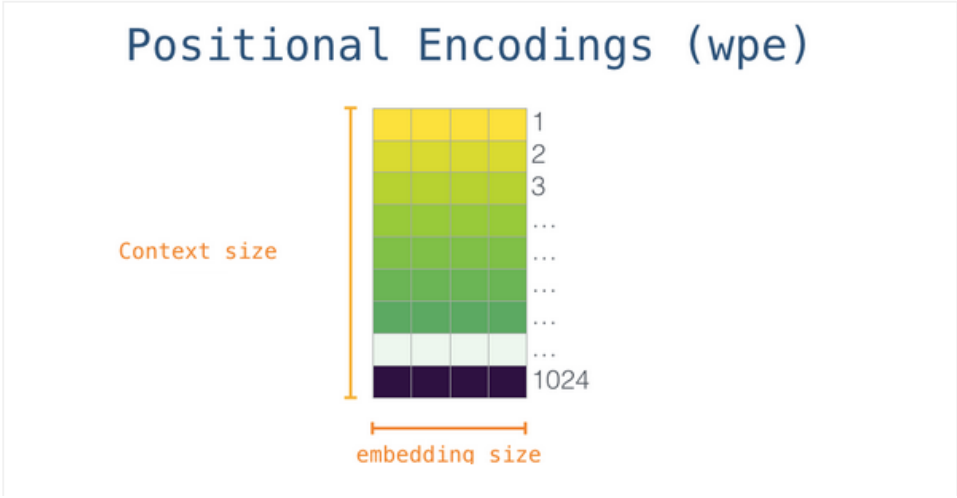
Model Configuration

- Model Configuration
  - Transformer layer = 12
  - Learnable **absolute positional embedding**
  - Hidden dimension = 768
  - Head count = 12
  - FFN dimension = 3072
  - Context length = 512
  - BPE vocabulary: 478 characters + 40,000 merges
  - Model size = 117M

Before Entering Transformer Blocks

- **Token embedding**: Map input tokens to embedding vectors (The vector length is the hidden dim.)
- **Positional embedding**: Add positional information in
- Token and positional embedding table

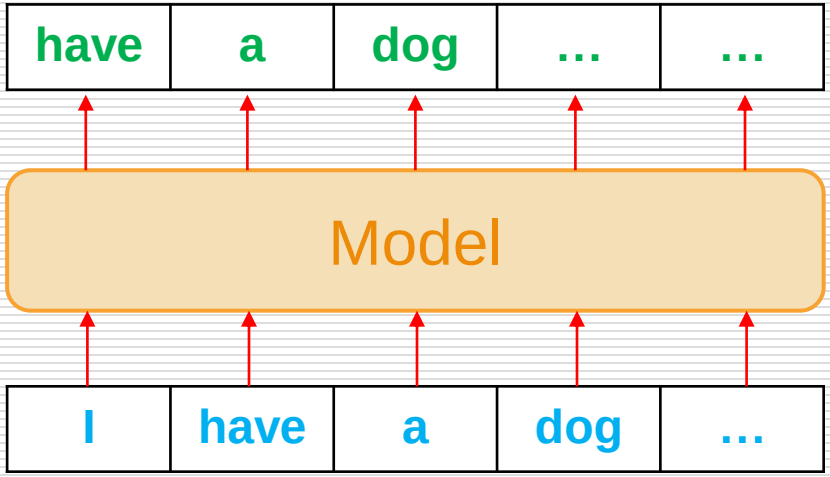




Training – Unsupervised Pretraining

- Goal: Learn how to speak (學會基本文字接龍)
- Unsupervised Pretraining
  - Given a series of tokens
  - The model will maximize the probability of the next token, conditioned on the input of all previous tokens

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$

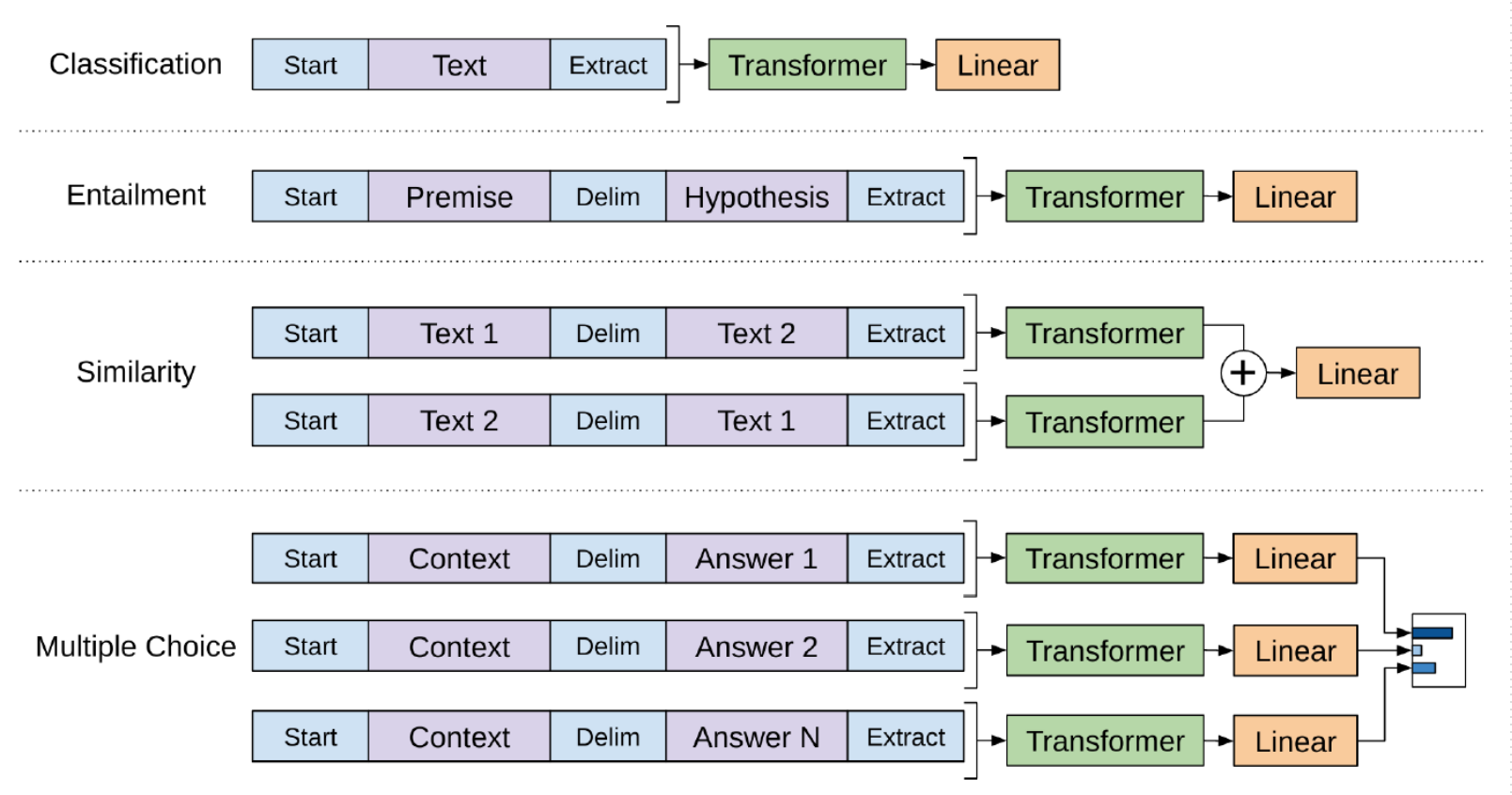


Training – Supervised Finetuning (1/2)

- Goal: Learn to be good in specific tasks (分門學習)
- Supervised Finetuning
  - A linear+softmax is appended after the last transformer block.
  - A task-specific dataset can be used to finetune the model to achieve better performance using supervised training.

Training – Supervised Finetuning (2/2)

- The downstream tasks



Dataset

- Pretrain Corpus
  - BookCorpus, 1.18GB
- Evaluation
  - GLUE (except WNLI)
  - RACE
  - Additional: SNLI, SciTail, Story Cloze

Training Setup

- Training setup
  - Loss function: CrossEntropy
  - Optimizer: Adam
  - Scheduler: Linear warmup + cosine annealing
  - Epoch = 100
  - Batch size = 64
- The setup details
  - LR = 2.5e-4
  - Warmup steps = 2000
  - Dropout = 0.1
- No mention of hardware setup and training time

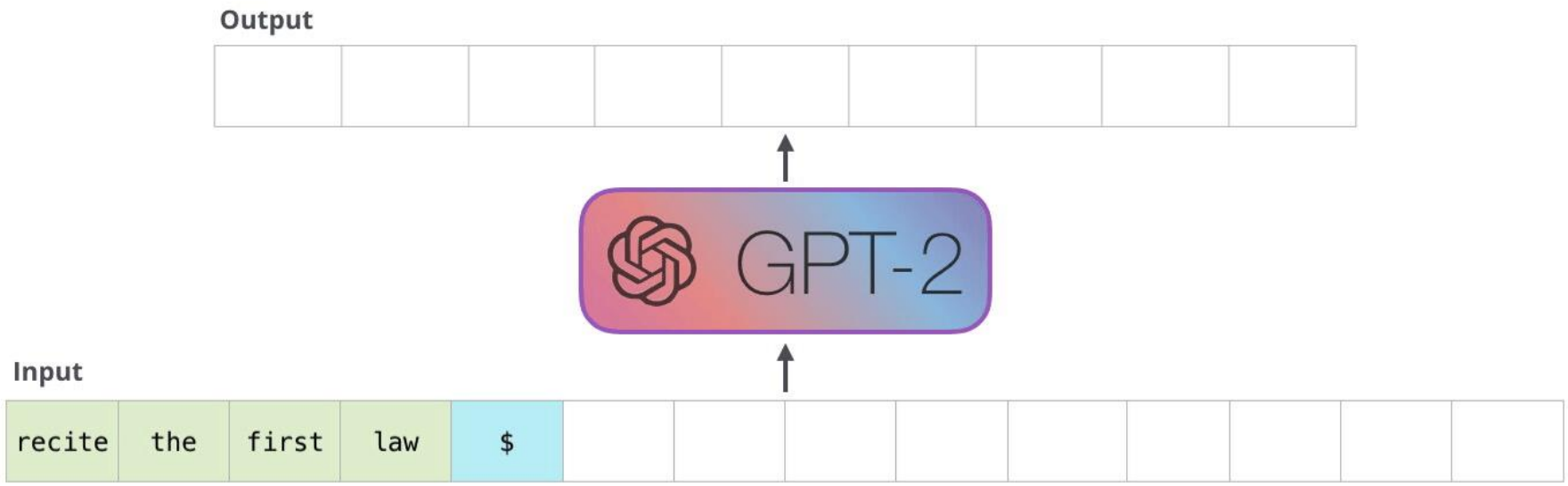
Experiment

- State of the art in 9 out of the 12 tasks studied

Table 2: Experimental results on natural language inference tasks, comparing our model with current state-of-the-art methods. 5x indicates an ensemble of 5 models. All datasets use accuracy as the evaluation metric.

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

Inference Visualization



GPT–2

Introduction

- **Language Models are Unsupervised Multitask Learners**
- Alec Radford et al.
- Self-published on OpenAI website in 2019
- Pretrained weight is open for public use

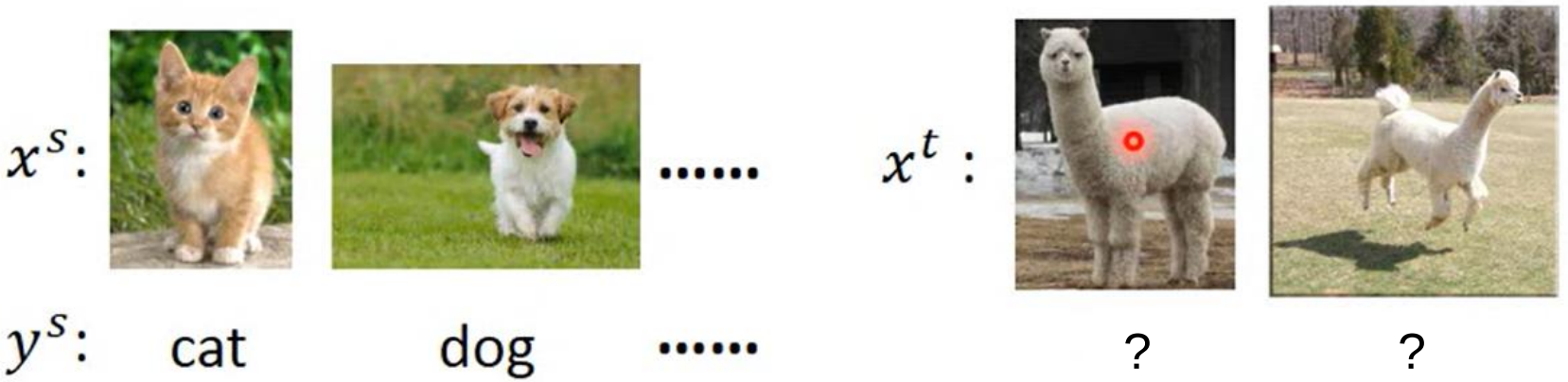
Insights

- LMs can have a strong capability of generation. Finetuning an LM to specific tasks is not needed.
- How?
  - Larger model size
  - Larger dataset
- The model begins to show SOTA performance in multiple tasks in a zero-shot setting.

Zero-Shot Learning (1/3)

- Source data: (x^s, y^s) → Training data
- Target data: (x^t) → Testing data

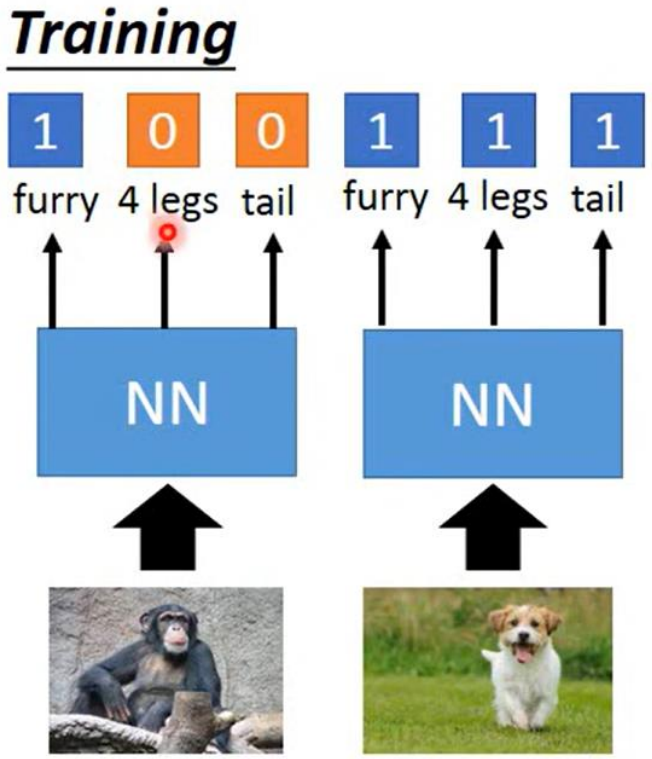
Different tasks



(Source: Hung-Yi Lee)

Zero-Shot Learning (2/3)

- Represent each class by its features
- Learn features, not learn classes



Database

attributes

class

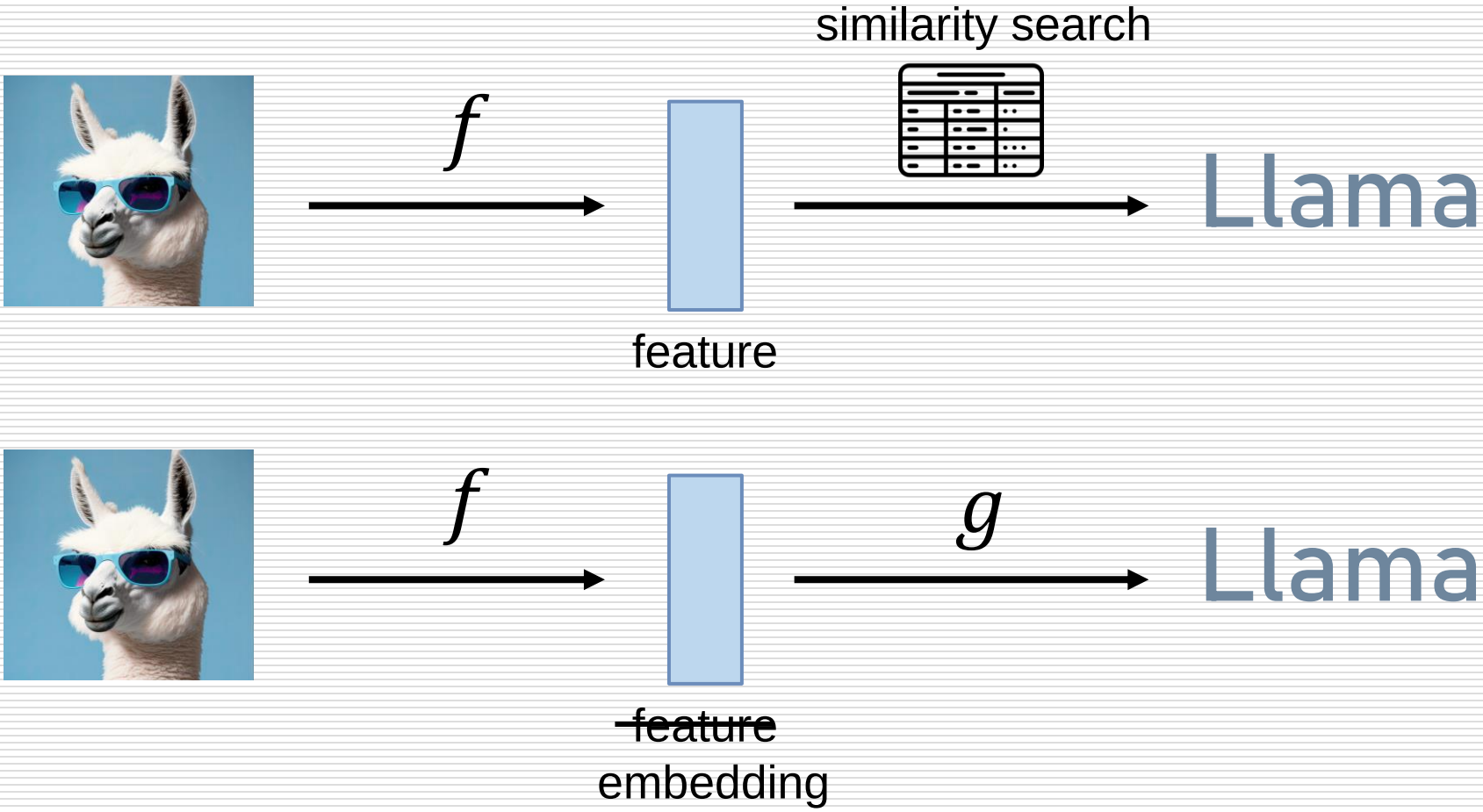
	furry	4 legs	tail	...
Dog	O	O	O	
Fish	X	X	O	
Chimp	O	X	X	
...				

sufficient attributes for one to one mapping

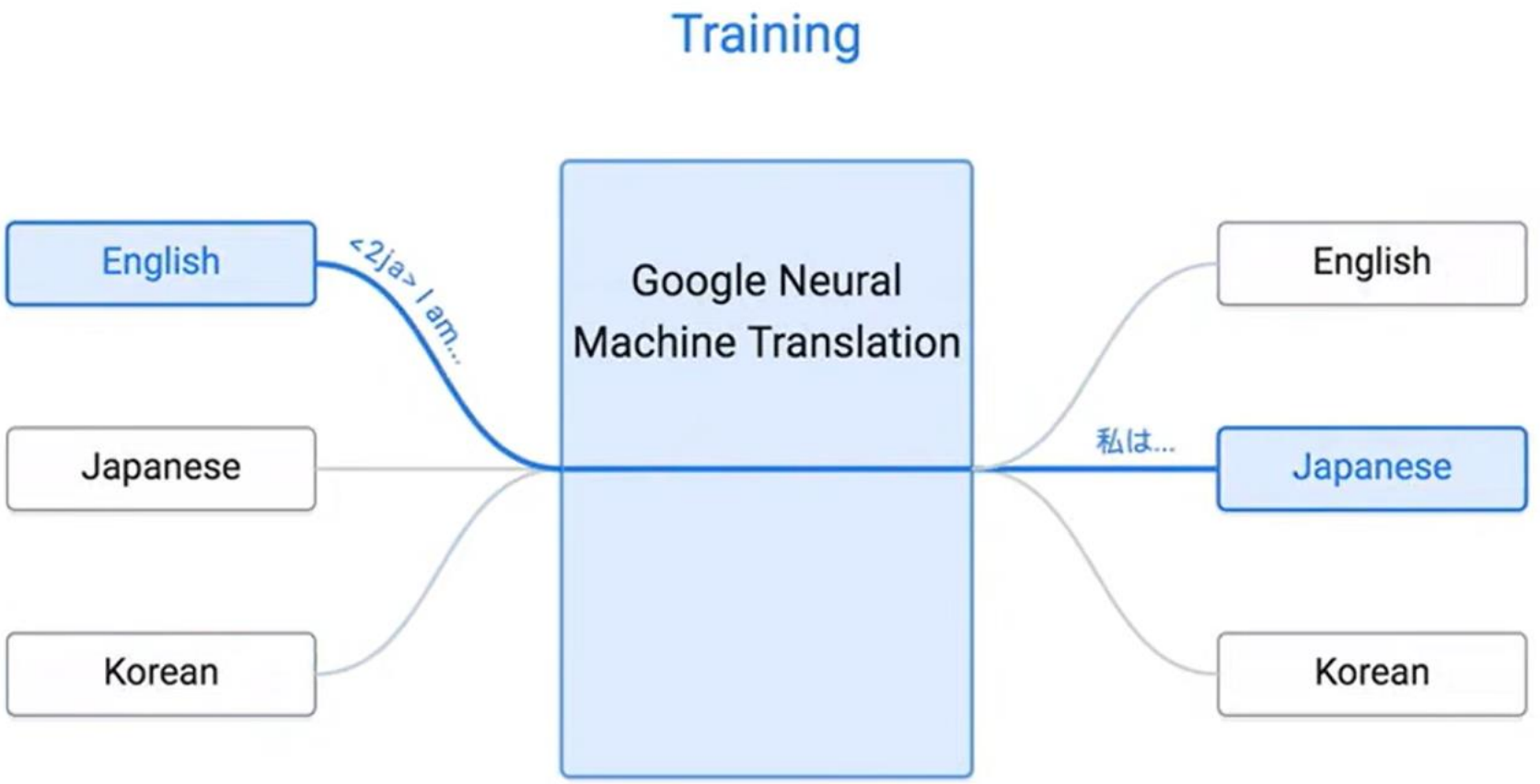
(Source: Hung-Yi Lee)

Zero-Shot Learning (3/3)

- If features are too complex? ➔ embedding space



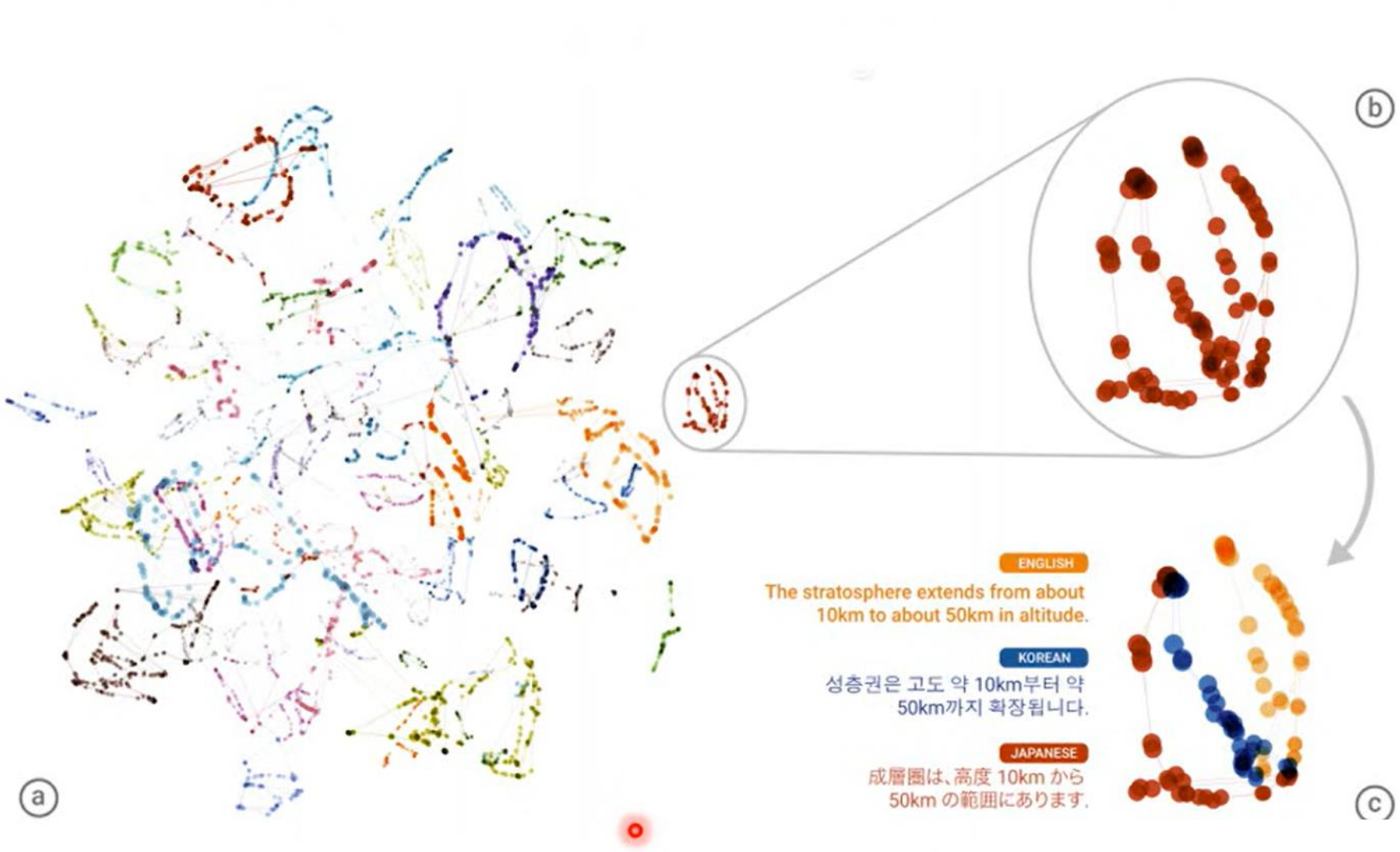
Example on NLP (1/2)



Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation, arXiv preprint 2016

(Source: Hung-Yi Lee)

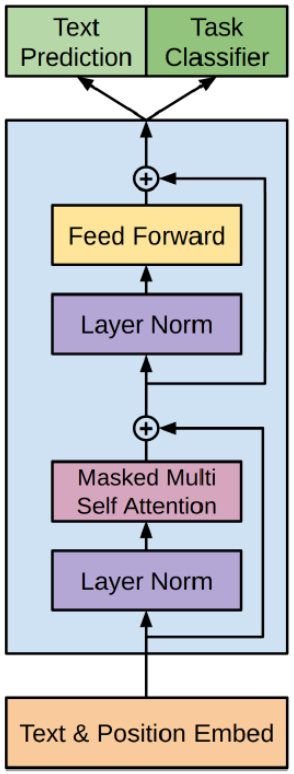
Example on NLP (2/2)



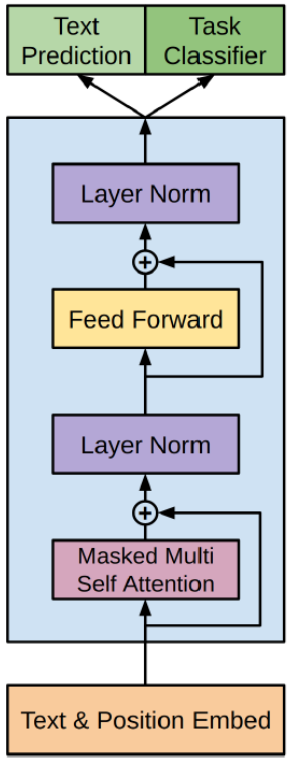
Model Structure – LayerNorm (1/2)

- LayerNorm
  - Position of LayerNorm is moved before attention and FFN

GPT-2



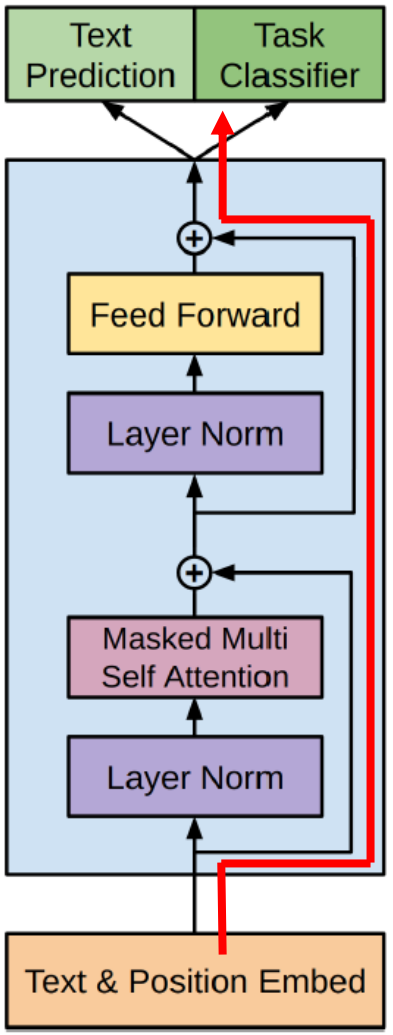
GPT-1



  - Additional LayerNorm after the last Transformer layer

Model Structure – LayerNorm (2/2)

- Explanation
  1. Clean and direct residual path for back propagation allows a deeper model to converge
  2. Input into attention and FFN are now always normalized, which improves stability of the model



Model Configuration

- Model Structure
  - Learnable absolute positional embedding
  - Transformer layer = 48
  - Hidden dimension = 1600
  - Head count = 25
  - FFN dimension = 6400
  - Context length = 1024
  - BPE vocabulary: 257 characters + 50,000 merges
  - Model size = 1.5B (GPT-1 model size = 117M)

Data

- Pretrain Corpus
  - WebText
    - over 8 million documents
    - a total of 40 GB of text
- Evaluation
  - LAMBADA, CBT, WikiText2, PTB, enwik8, text8, WikiText103, 1BW
  - Winograd Schema, CoQA
  - CNN and Daily Mail
  - WMT-14 English ⇔ French

Training Setup

- Training setup
  - Loss function: CrossEntropy
  - Optimizer: Adam
  - Scheduler: Linear warmup + cosine annealing
  - Epoch = 100
  - Batch size = 512
- Setup details
  - LR = (Unknown)
  - Warmup steps = (Unknown)
  - Dropout = (Unknown)
  - No mention of hardware setup and training time

Training

- Trained and benchmarked LMs with four sizes

	Parameters	Layers	d_{model}
(size of GPT-1)	117M	12	768
(size of BERT-Large)	345M	24	1024
	762M	36	1280
GPT-2	1542M	48	1600

- Learning rate was manually tuned for the best perplexity.
- All models still underfit Web-Text.

Experiment

- Achieved SOTA results on 7 out of 8 tested language modeling datasets in a zero-shot setting.

Language Models are Unsupervised Multitask Learners

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

Discussion

- The zero-shot performance of GPT-2 shows its potential, but it is not clear where the ceiling of its performance is with finetuning.

- While GPT-2 shows good zero-shot performance on reading comprehension tasks, the performance on other tasks like summarization is still **basic**.
- There are many practical tasks where GPT-2's performance is still **no better than random**.

GPT–3

Introduction

- **Language Models are Few-Shot Learners**
- Tom Brown et al.
- Published on NIPS 2020
- Pretrained weight is “**NOT**” open to the public

Insights

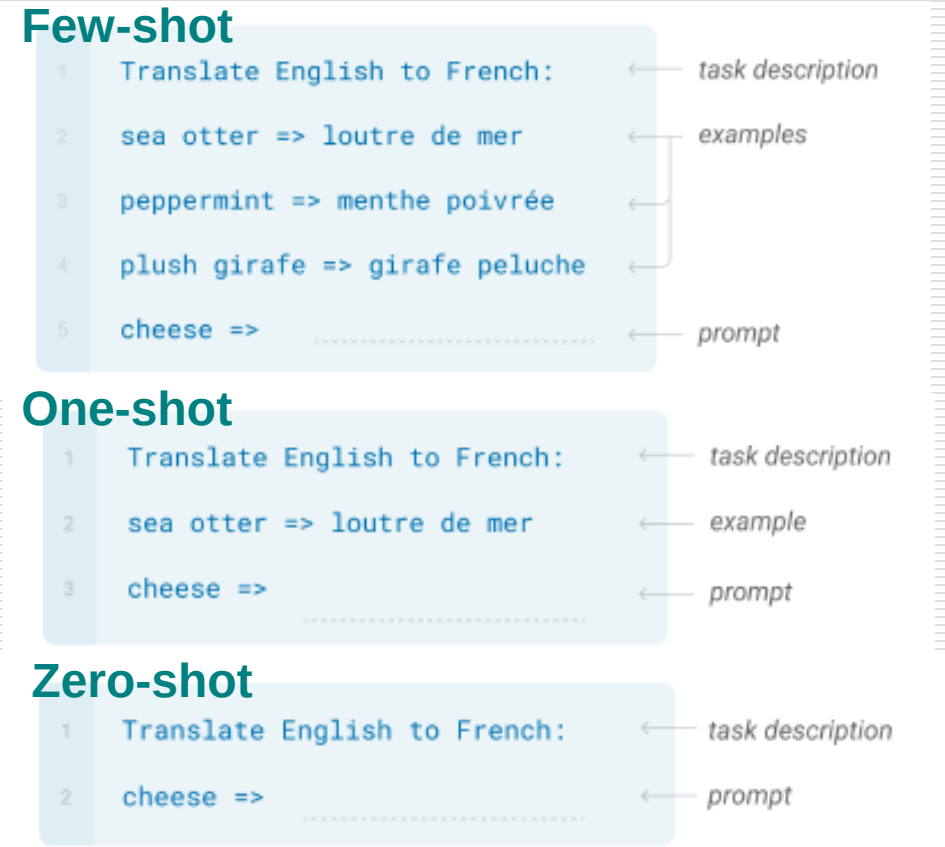
- Scaling up language models greatly improves task-agnostic and few-shot performance.
- In some tasks, it’s even competitiveness with prior SOTA finetuning approaches.

Few/one/zero-Shot

- The definition here is how many examples are provided at inference time.
  - Few-shot: a few example
  - One-shot: one example
  - Zero-shot: no example

- No finetuning for all tasks.

- In-context learning

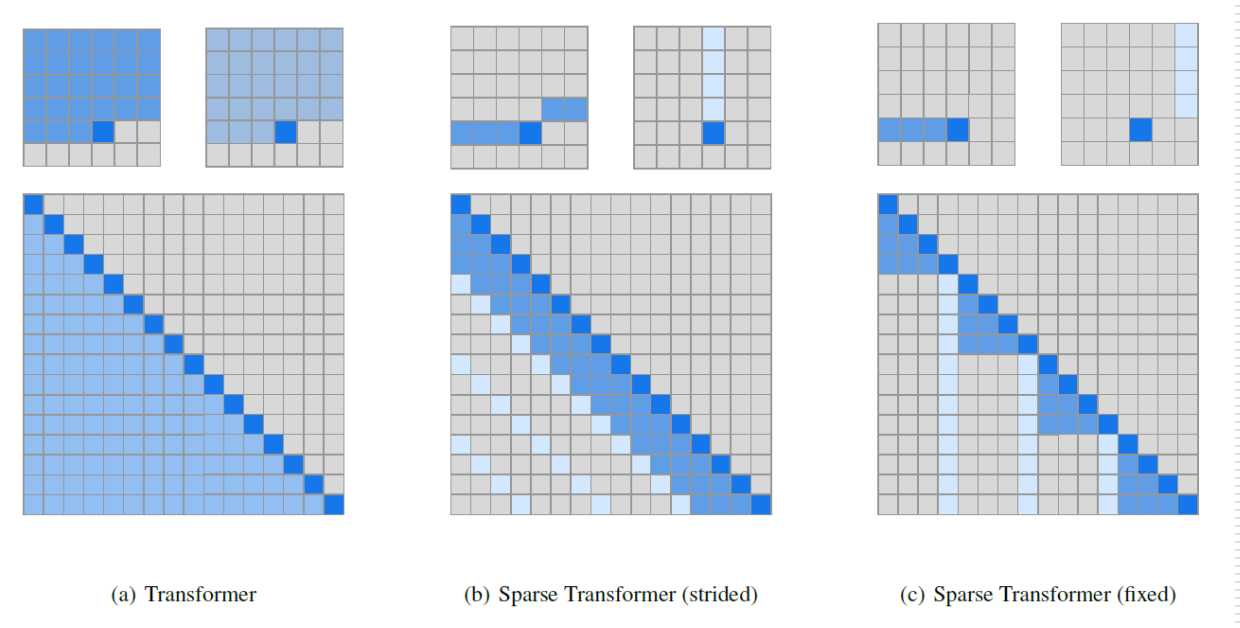


Model Structure

- The same model and architecture as GPT-2
- Use either of dense or locally banded sparse attention, which is similar to the Sparse Transformer
  - To reduce computation complexity
  - The exact interleaving ratio is unknown

Sparse Transformer

- Two forms of sparse attention patterns are proposed, “strided” and “fixed”
- Sparse Transformer is a autoregressive model, thus its masked attention only attends to previous tokens.
- For text input, It is found that “fixed” pattern works better.



(a) Transformer (b) Sparse Transformer (strided) (c) Sparse Transformer (fixed)

(Child et al).

Model Configuration

- Model Structure
  - Learnable absolute positional embedding
  - Transformer layer = 96
  - Hidden dimension = 12288
  - Head count = 96
  - FFN dimension = 49152
  - Context length = 2048
  - BPE vocabulary: (Unknown)
  - Model size = 175B

Training Data

- Pretrain Corpus – 570GB (45T before filtering)
  - Common Crawl, WebText2, English Wiki
  - Book1, Book2 (Unknown)
  - Training data is weighted mixed

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

Evaluation Dataset (1/2)

- Evaluation
  - PTB, LAMBADA, HellaSwag, StoryCloze
  - Natural Question, WebQuestions, TriviaQA
  - WMT En ⇔ Fr, En ⇔ De, En ⇔ Ro
  - Winograd, Winogrande
  - PhysicalQA, ARC, OpenBookQA
  - CoQA, DROP, QuAC, SQuAD 2.0, RACE
  - SuperGLUE

Evaluation Dataset (2/2)

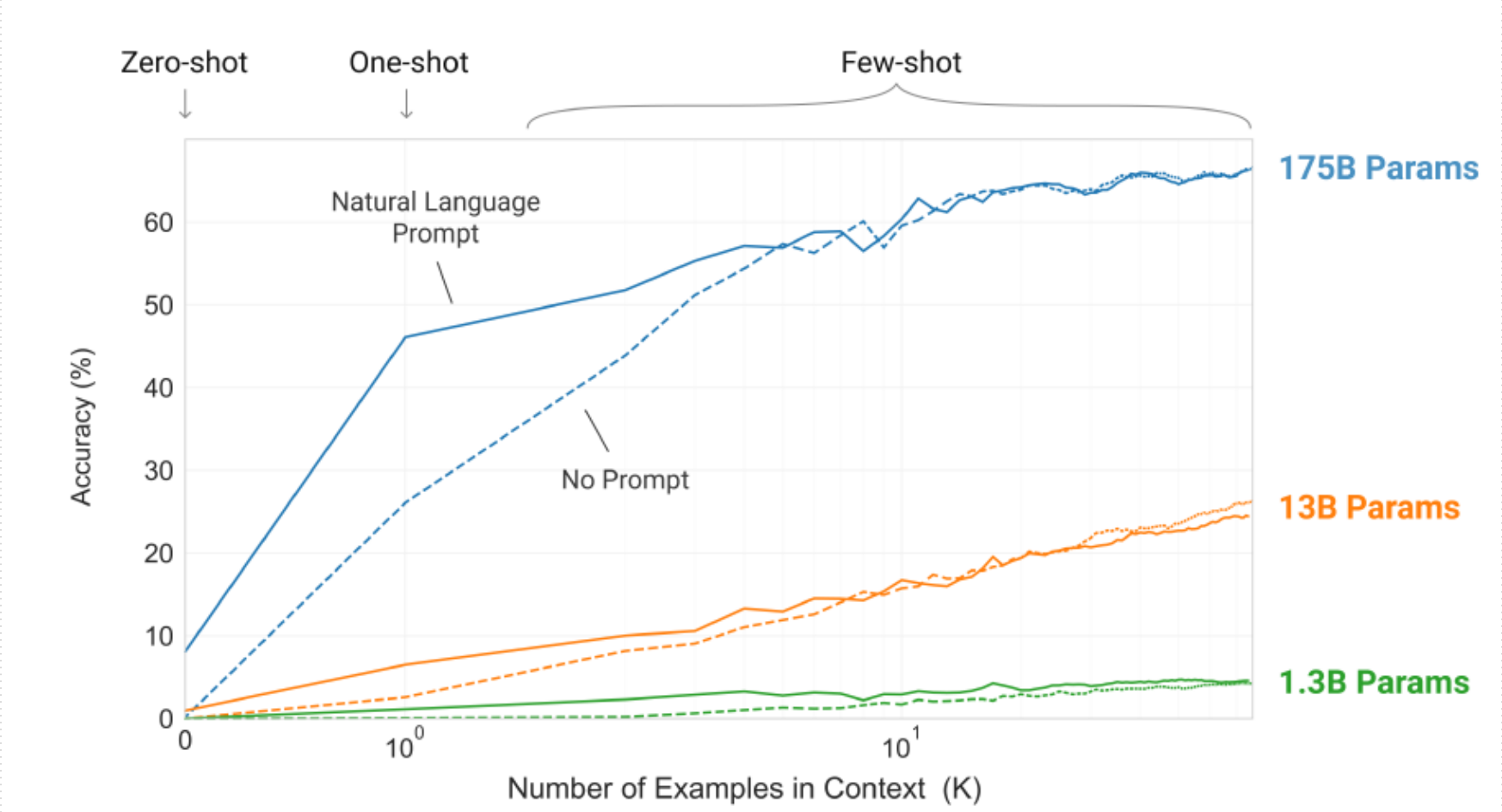
- Evaluation
  - Arithmetic
  - Word de-scrambling and manipulation
  - SAT analogies
  - Novel word learning
  - English grammar correction

Training Setup

- Training Setup
  - Loss function: CrossEntropy
  - Optimizer: Adam
  - Scheduler: Linear warmup + cosine annealing
  - Epoch = (300B tokens)
  - Batch size = 3.2M * (Linearly increase from 32K)
- Setup Details
  - LR = 6e-5
  - Warmup steps = (375M tokens)
  - Dropout = 0.1
  - No mention of hardware setup and training time

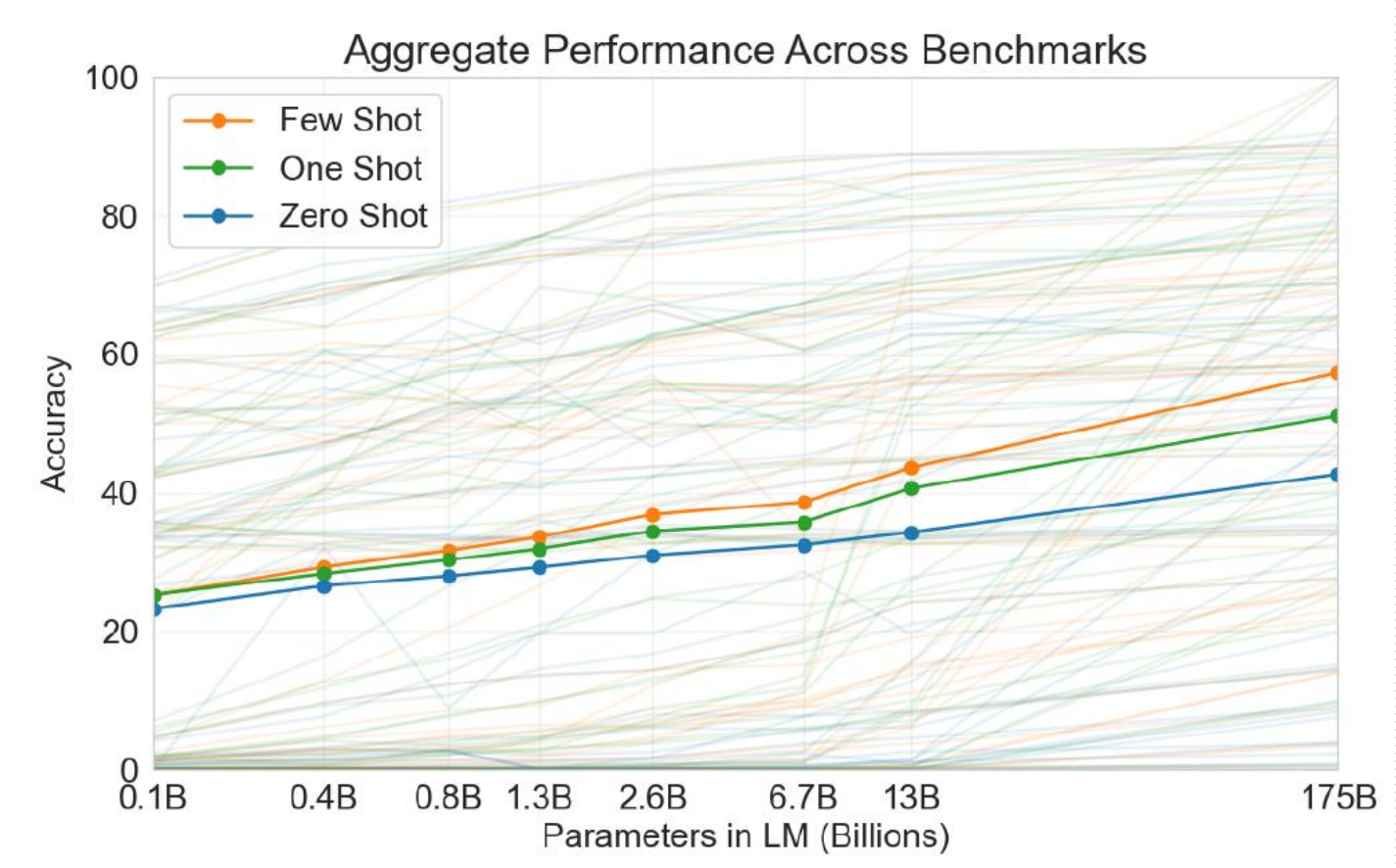
Experiment (1/4)

- Larger models make increasingly efficient use of in-context information.



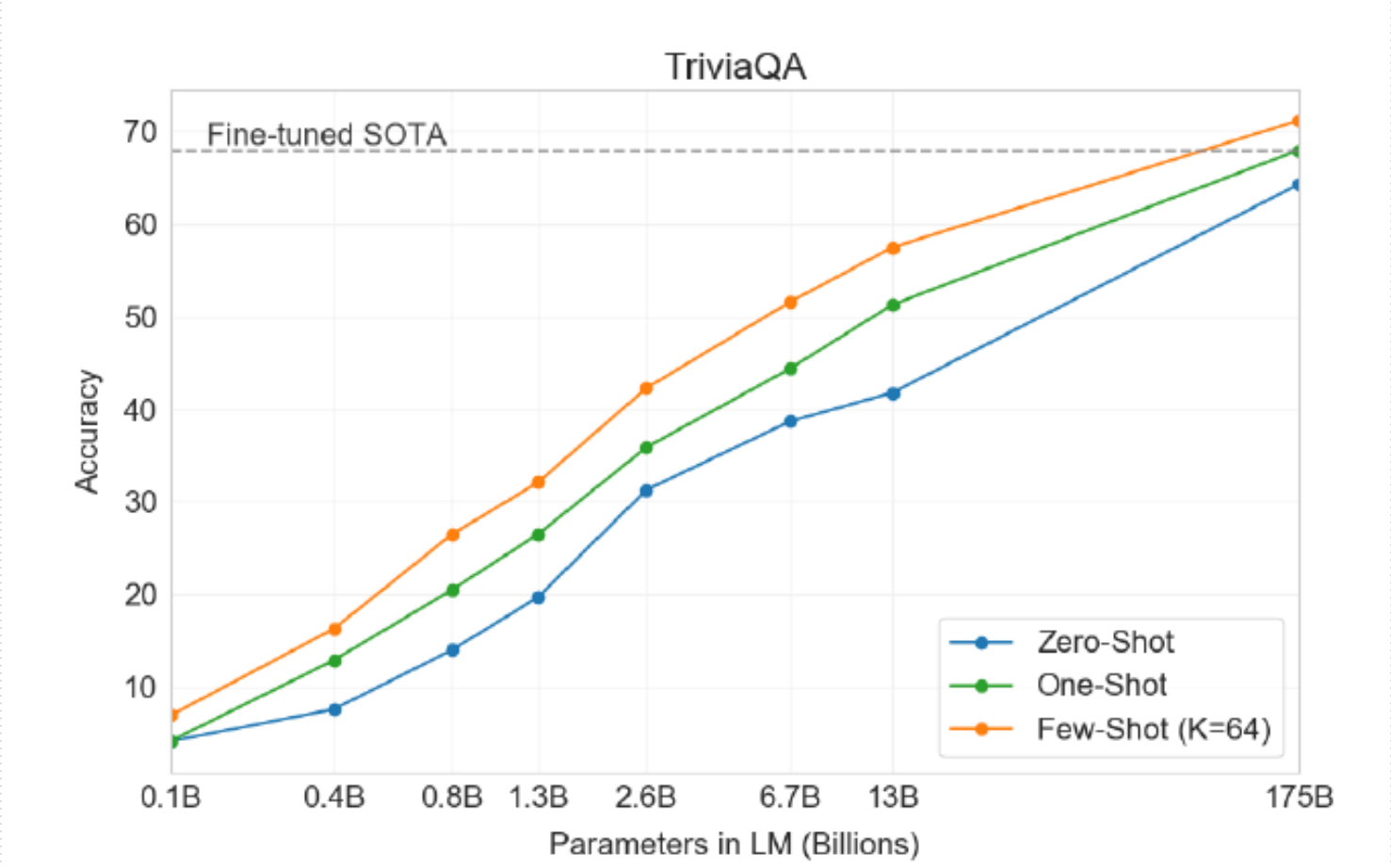
Experiment (2/4)

- With the increasing of few-shot performance increases more rapidly, showing that larger models are more proficient at in-context learning



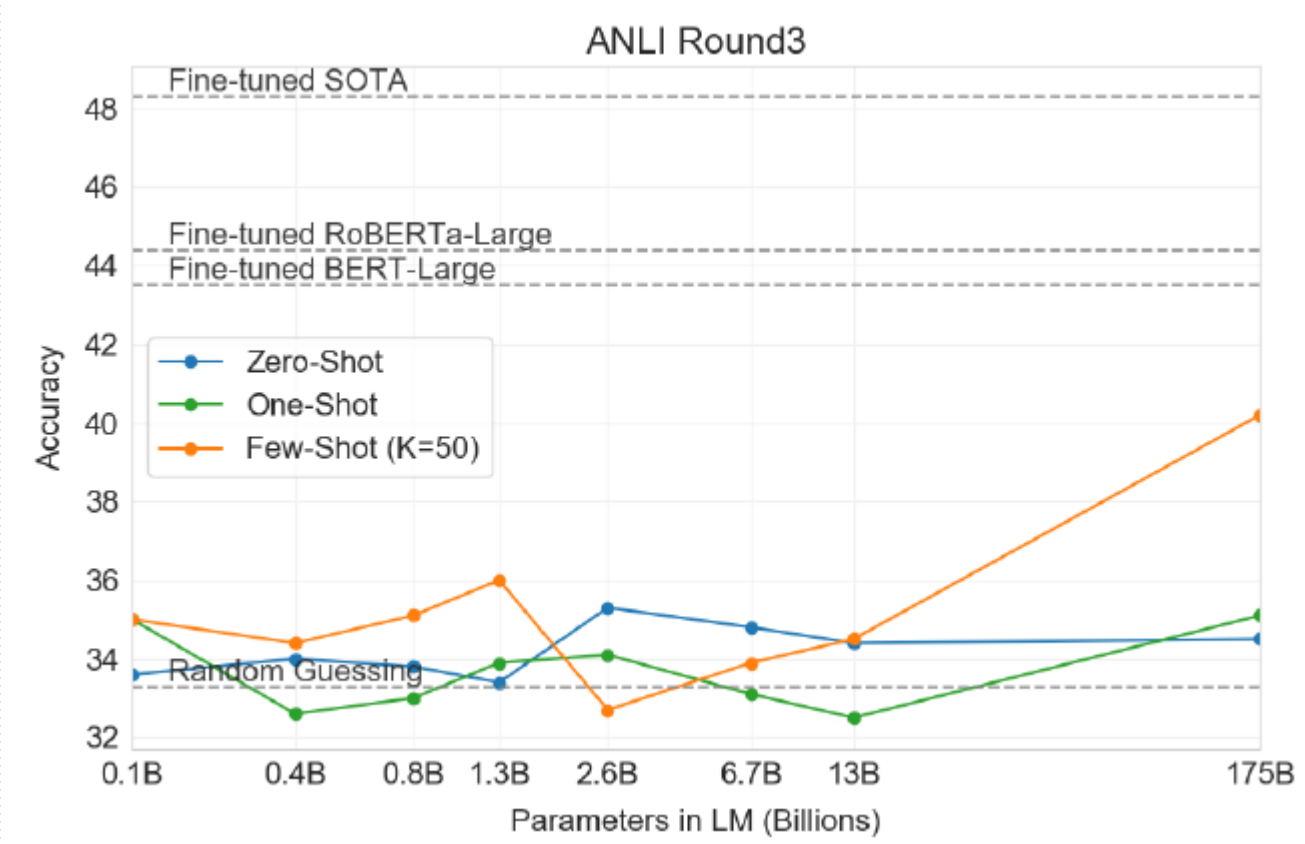
Experiment (3/4)

- In TriviaQA, GPT-3's few-shot performance exceeds the SOTA fine-tuned model, RAG.



Experiment (4/4)

- While in some cases GPT-3 is nearly matching the performance of SOTA finetuned model,
- that doesn’t mean it’s excellent in all tasks.



Anecdote

- The Famous “Mistake”
  - “…To reduce such contamination, we searched for and attempted to remove any overlaps with the development and test sets of all benchmarks studied in this paper. Unfortunately, a bug in the filtering caused us to ignore some overlaps, and due to the cost of training it was not feasible to retrain the model. In Section 4 we characterize the impact of the remaining overlaps, and in future work we will more aggressively remove data contamination.” – Section 2.2

GPT–3.5

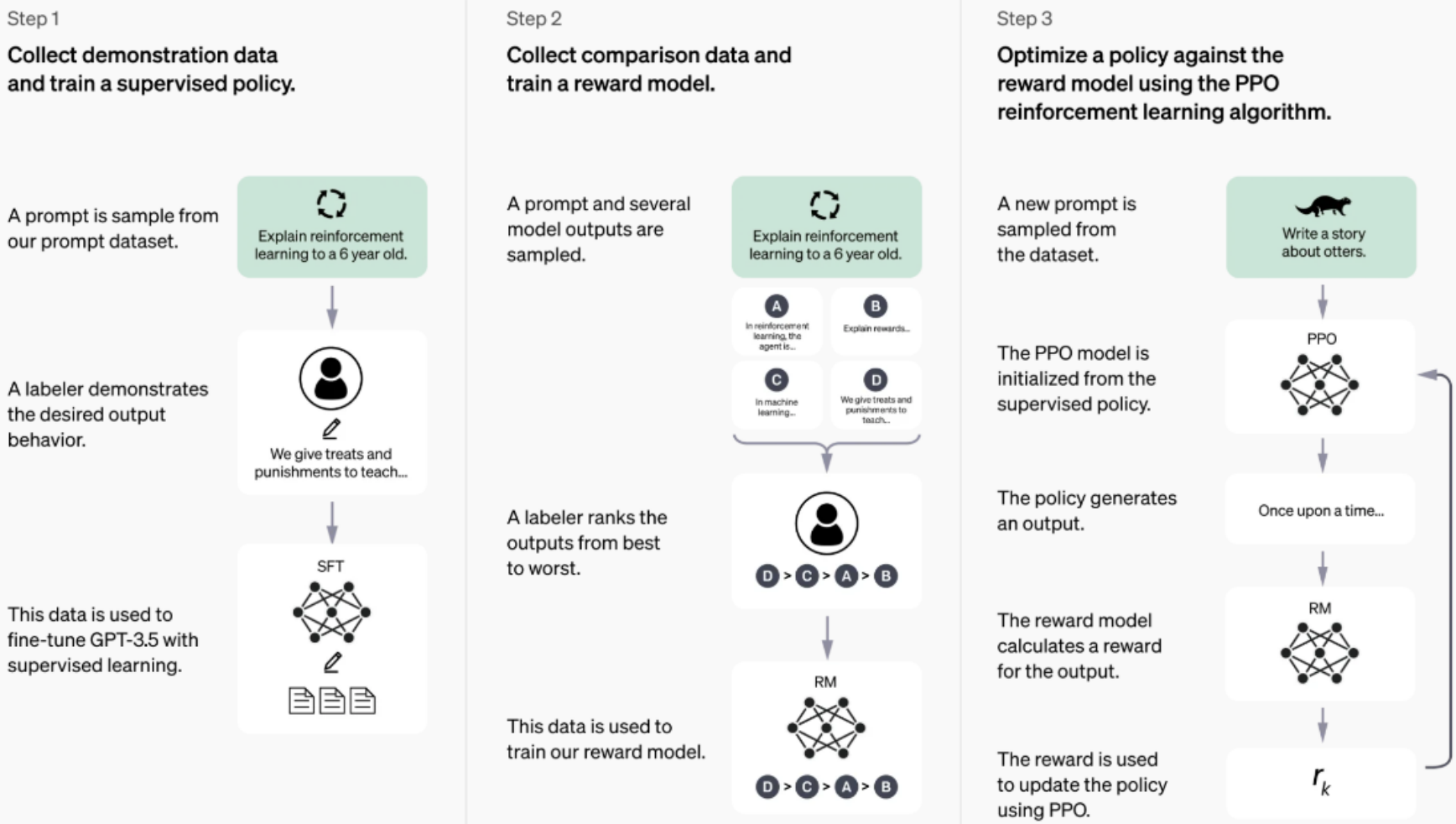
ChatGPT

Introduction

- GPT-3.5 is a sub class of GPT-3 models.

- ChatGPT released in 2022 is a sibling model to InstructGPT.
- InstructGPT is fine-tuned from GPT-3.

ChatGPT – Method



Source: https://openai.com/index/chatgpt/

InstructGPT – Method

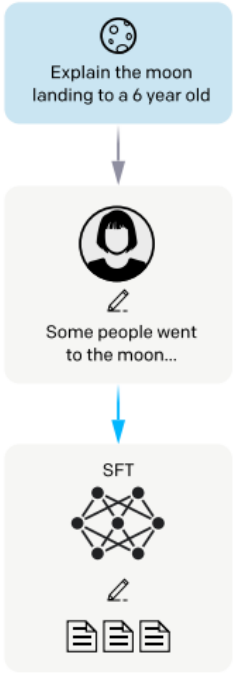
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



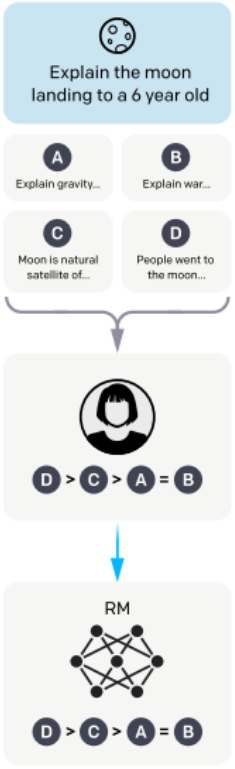
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

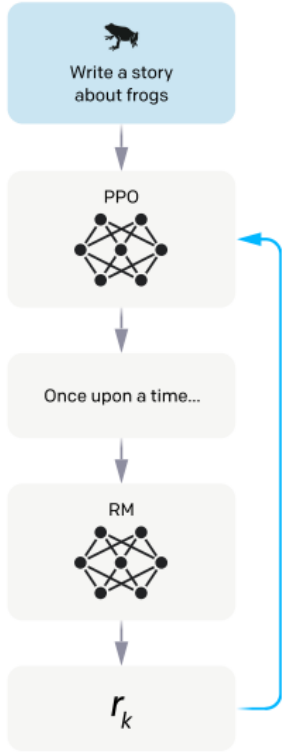
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

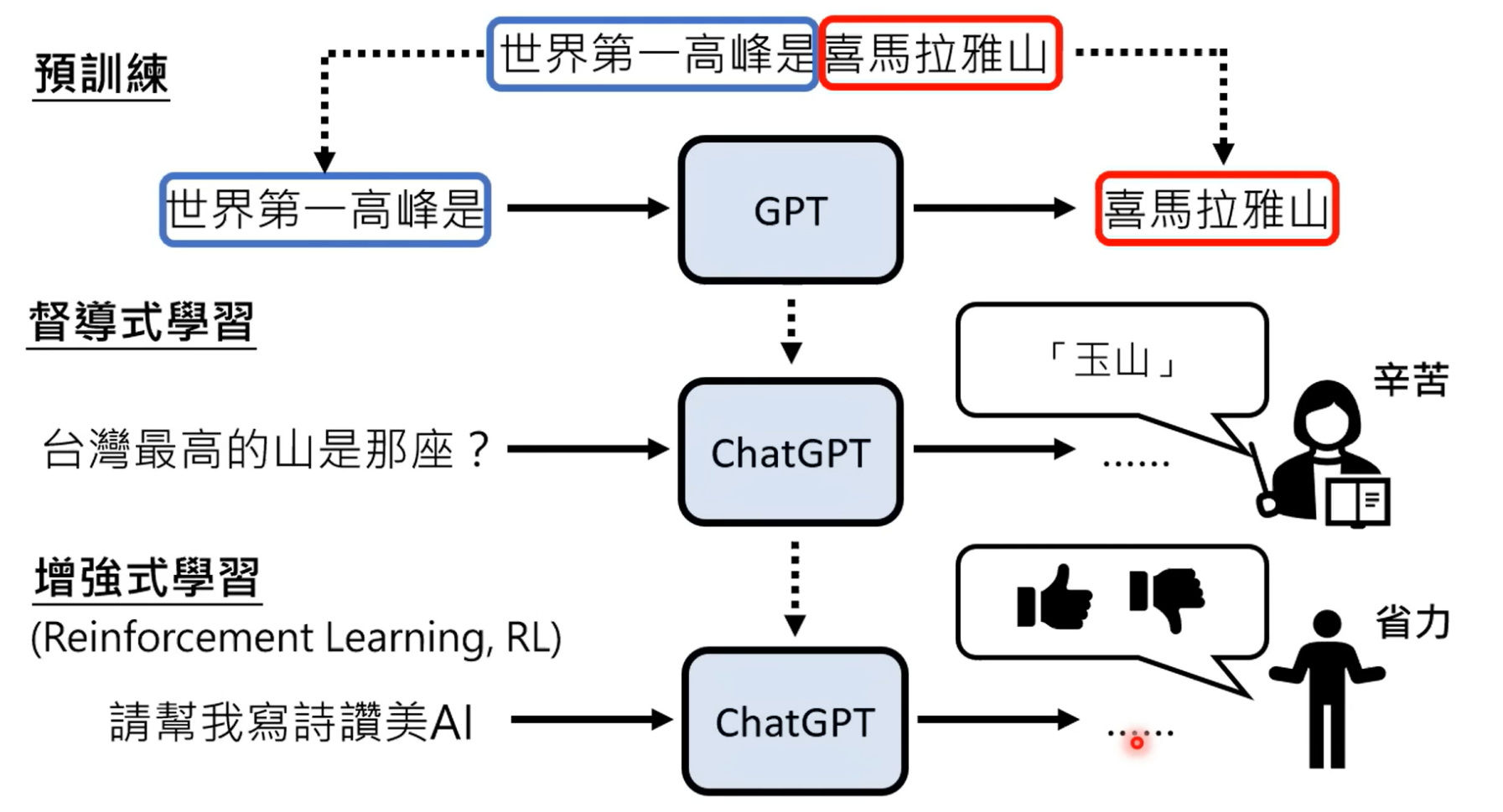
The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Source: https://arxiv.org/pdf/2203.02155

How can ChatGPT chat?

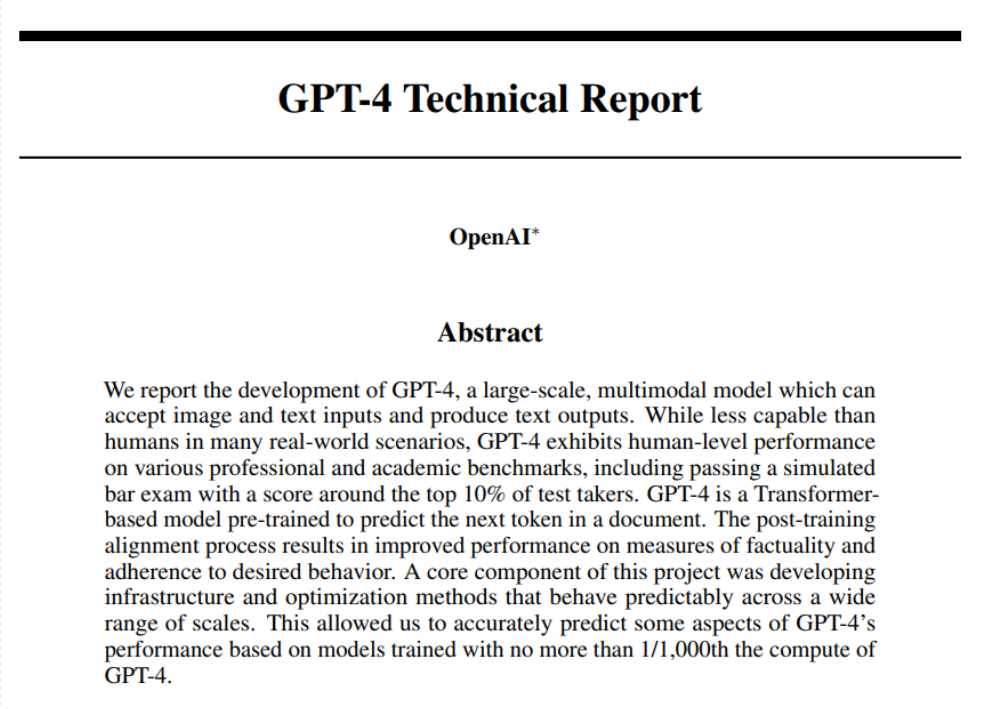


(Source: Hung-Yi Lee)

GPT–4

GPT-4

- **GPT-4 Technical Report**
- Author: OpenAI, 100 pages
  - 23 for paper; 77 for appendix
  - 14 for technical (no details, actually), 3 for author list, 6 for citations
- Published on arXiv in Mar 2024



GPT-4 Technical Report

OpenAI*

Abstract

We report the development of GPT-4, a large-scale, multimodal model which can accept image and text inputs and produce text outputs. While less capable than humans in many real-world scenarios, GPT-4 exhibits human-level performance on various professional and academic benchmarks, including passing a simulated bar exam with a score around the top 10% of test takers. GPT-4 is a Transformer-based model pre-trained to predict the next token in a document. The post-training alignment process results in improved performance on measures of factuality and adherence to desired behavior. A core component of this project was developing infrastructure and optimization methods that behave predictably across a wide range of scales. This allowed us to accurately predict some aspects of GPT-4's performance based on models trained with no more than 1/1,000th the compute of GPT-4.

Comparison among GPT 1~3

	GPT-1	GPT-2	GPT-3
Layer	12	48	96
Dim	768	1600	12288
Head	12	25	96
Dim / Head	64	64	128
FFN (4x)	3072	6400	49152
Length	512	1024	2048
Vocab	40478	50257	(N/A)
Param	**117M**	**1.54B**	**176B**

12.8x

116.7x

Corpora

Common Corpora

- BookCorpus
- English Wikipedia
- Common Crawl
- WebText
- The Pile

BookCorpus

- *Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books*
- Yukun Zhu et al., 2015
- 11,038 unpublished books of 16 genres
- Almost 1B words

English Wikipedia

- 4.4M English webpages from Wikipedia
- 3B words
- Needs to be updated continually

Common Crawl

- Online archive of website crawls
- Organization founded in 2011, supported by Amazon Web Services
- More than 1T words
- Requires extensive text cleaning before use

WebText

- Created for GPT-2, unreleased
- Alec Radford et al., 2018
- 45M outbound links from Reddit posts with > 3 karma, collected up to Dec 2017
- 8M documents after screening
- 6B words

WebText2

- Created for GPT-3, unreleased
- Tom Brown et al., 2020
- Collection extended to Oct 2018
- 20M documents
- 16B words

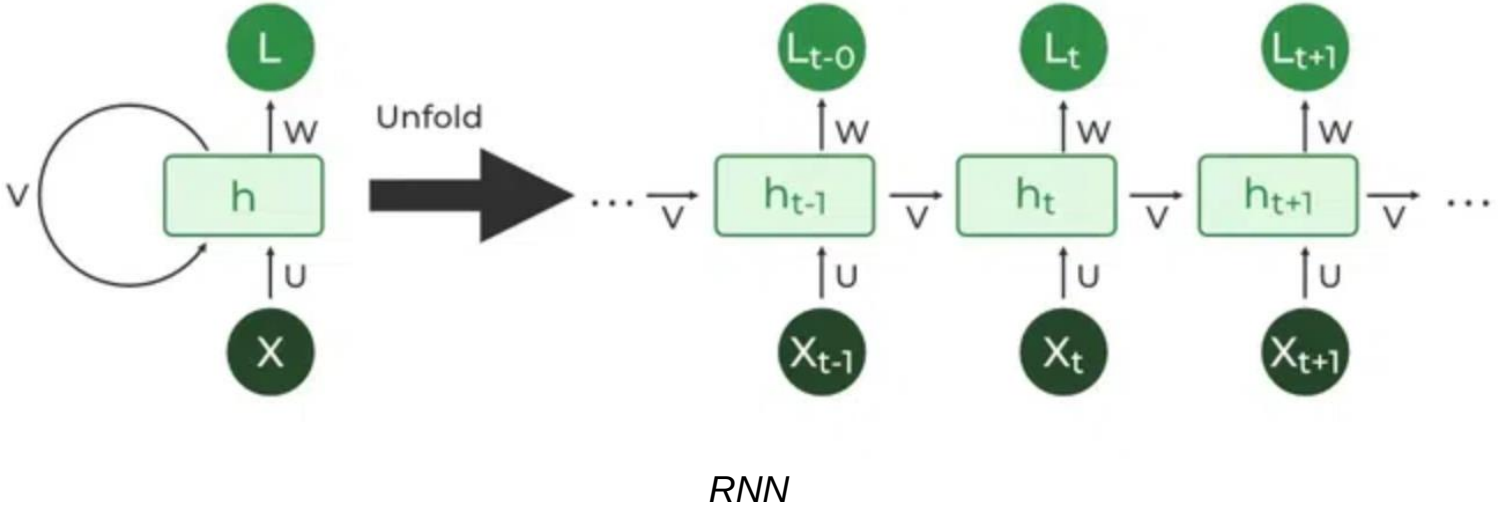
The Pile

- *The Pile: An 800GB Dataset of Diverse Text for Language Modeling*
- Compiled and published by Gao et al. in 2020
- A large and open corpus from many domains, with 22 sub-datasets
  - Academic: ArXiv, FreeLaw, PubMed…
  - Internet: Wikipedia, StackExchange…
  - Prose: Project Gutenberg, BookCorpus2…
  - Dialogue: Subtitles, Hacker News, IRC…
  - Miscellaneous: DM Mathematics, Github…
- Very helpful for academic research

Positional Encoding and RoPE

Why is Position Encoding Needed? (1/2)

- Unlike RNNs that process sequences one step at a time, transformers analyze the entire sequence simultaneously.
  - speed and efficiency but cost a lot



RNN

Why is Position Encoding Needed? (1/2)

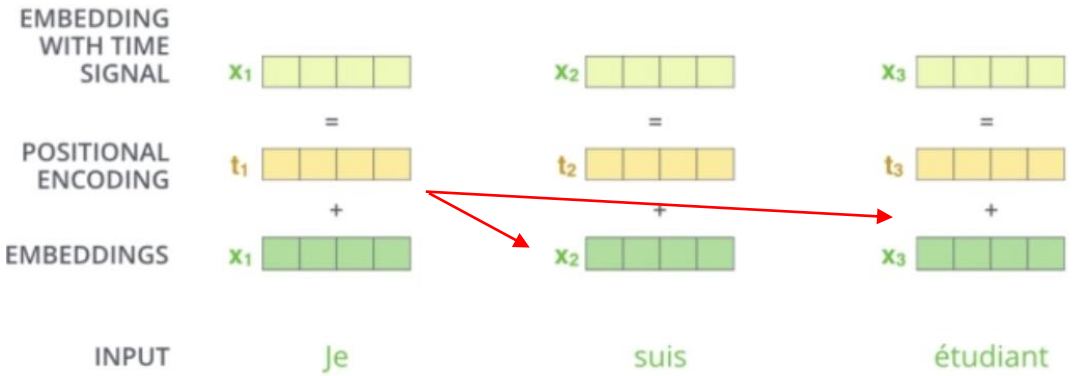
- This parallelism offers significant advantages in terms of speed and efficiency.

- Self-attention is no positional relationship for each word. If the order of each word is disrupted, the attention value obtained remains unchanged.

- To give the model some positional information, one solution is to add to each word information about its position in the sentence.

Implementation (1/3)

- Positional vectors use *sine* and *cosine* functions of different frequencies to encode the absolute position of a token within a sentence.
  - pos : position of the token in the sequence
  - i : dimension



$$\mathrm{PE}(pos, 2i) = sin\left(\frac{pos}{10000^{2i/d_{model}}}\right),$$

$$\mathrm{PE}(pos, 2i+1) = cos\left(\frac{pos}{10000^{2i/d_{model}}}\right).$$

Implementation (2/3)

- The equations in the image make use of the trigonometric identities:

$$\sin(\alpha+\beta) = \sin\alpha\cos\beta + \cos\alpha\sin\beta$$
$$\cos(\alpha+\beta) = \sin\alpha\cos\beta + \cos\alpha\sin\beta$$

- Can be expressed for the d_{model} dimension as follows:

$$\begin{cases} PE(pos+k,2i) = PE(pos,2i) \times PE(k,2i+1) + PE(pos,2i+1) \times PE(k,2i) \\ PE(pos+k,2i+1) = PE(pos,2i+1) \times PE(k,2i+1) - PE(pos,2i) \times PE(k,2i) \end{cases}$$

Implementation (3/3)

- Position Information
  - position (pos+k) can be expressed as a linear combination of the encodings at positions pos and k across dimensions $2i$ and $2i$+1

- Linear Combinations
  - capture the relative position information between tokens in a sequence

Rotary Position Embedding (RoPE)

- RoPE originates from RoFormer
  - RoFormer: Enhanced Transformer with Rotary Position Embedding

- Clearly express the relative position relationships between tokens.

- Offers extensibility (without needing to define a max sequence length beforehand) and better generalization.

General Form (1/2)

- In order to generalize results in 2D to any $x_i \in R^d$ (d is even),
- divide the d-dimension space into $d/2$ sub-spaces and combine them in the merit of the linearity of the inner product.

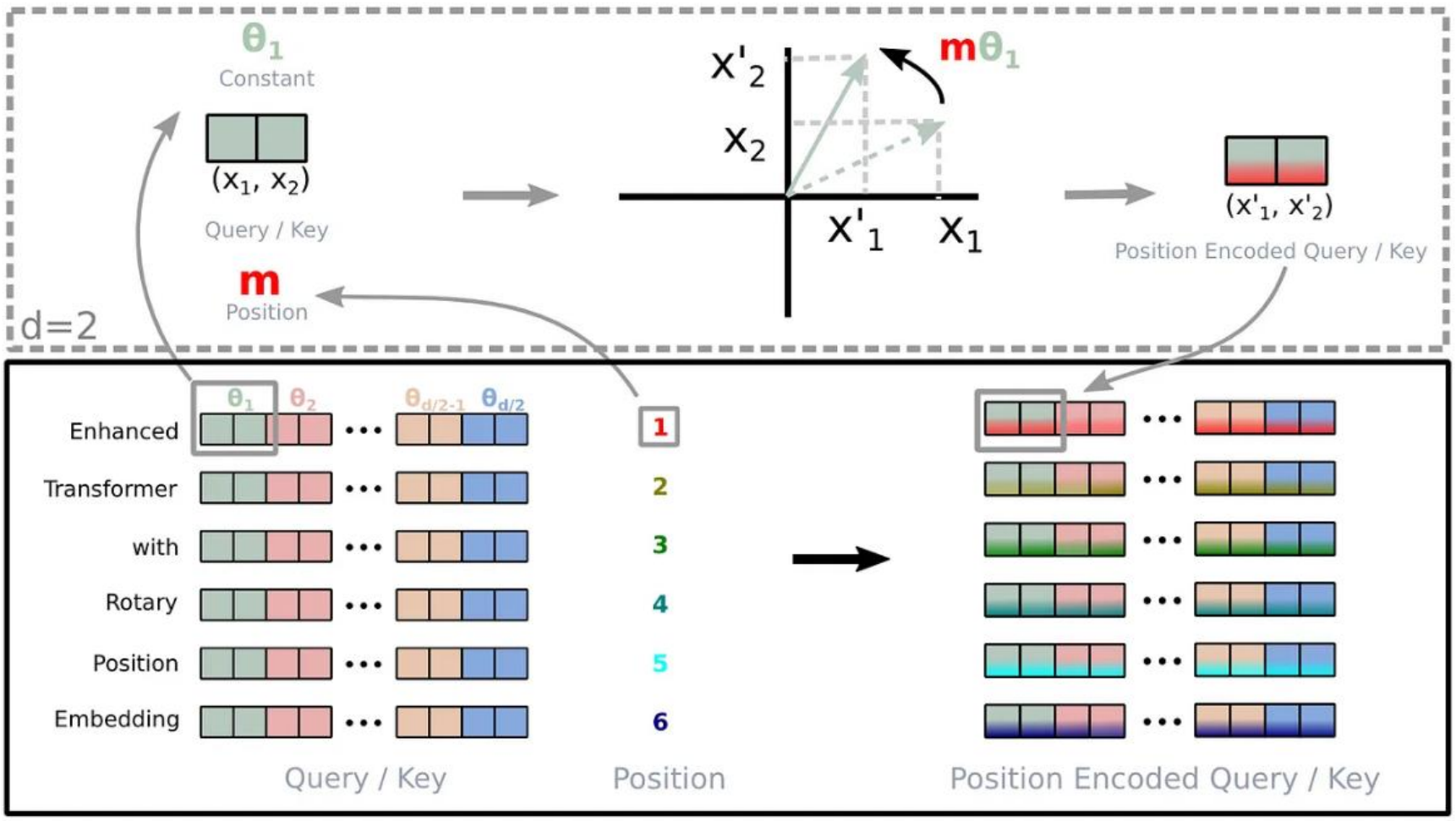
$$f_{\{q,k\}}(\boldsymbol{x}_m, m) = \boldsymbol{R}^d_{\Theta,m}\boldsymbol{W}_{\{q,k\}}\boldsymbol{x}_m$$

where

$$\boldsymbol{R}^d_{\Theta,m} = \begin{pmatrix} \cos m\theta_1 & -\sin m\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ \sin m\theta_1 & \cos m\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos m\theta_2 & -\sin m\theta_2 & \cdots & 0 & 0 \\ 0 & 0 & \sin m\theta_2 & \cos m\theta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos m\theta_{d/2} & -\sin m\theta_{d/2} \\ 0 & 0 & 0 & 0 & \cdots & \sin m\theta_{d/2} & \cos m\theta_{d/2} \end{pmatrix}$$

General Form (2/2)

- Implementation of Rotary Position Embedding (RoPE)



Computational Efficient Realization

- Due to the sparsity of matrix, we can use dot product to replace matrix multiplication.

- More computational efficient realization as below.

$$\boldsymbol{R}^{d}_{\Theta,m}\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{d-1} \\ x_d \end{pmatrix} \otimes \begin{pmatrix} \cos m\theta_1 \\ \cos m\theta_1 \\ \cos m\theta_2 \\ \cos m\theta_2 \\ \vdots \\ \cos m\theta_{d/2} \\ \cos m\theta_{d/2} \end{pmatrix} + \begin{pmatrix} -x_2 \\ x_1 \\ -x_4 \\ x_3 \\ \vdots \\ -x_d \\ x_{d-1} \end{pmatrix} \otimes \begin{pmatrix} \sin m\theta_1 \\ \sin m\theta_1 \\ \sin m\theta_2 \\ \sin m\theta_2 \\ \vdots \\ \sin m\theta_{d/2} \\ \sin m\theta_{d/2} \end{pmatrix}$$

Limitation of RoPE

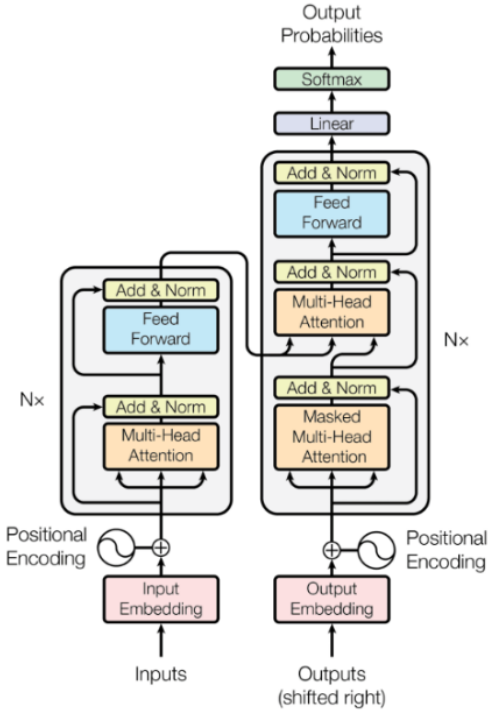
- Lacks of thorough explanations on why RoPE converges faster than baseline models.

- It performs better on long texts compared to peer models, we have not yet provided a satisfactory explanation.

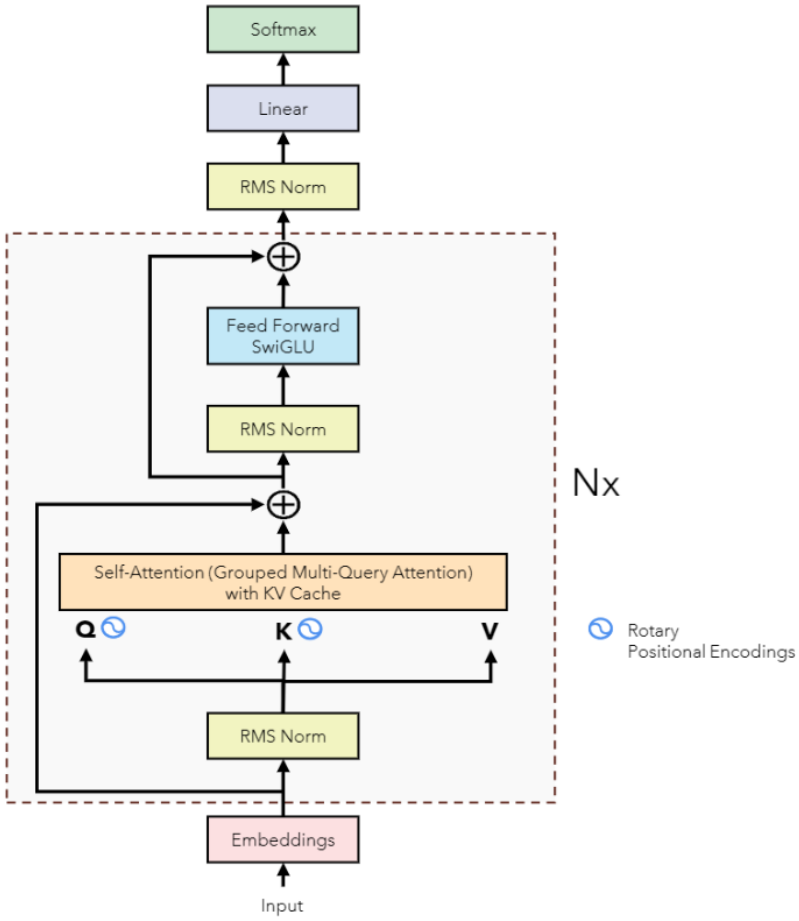
LLaMA Series

Transformer vs LLaMA (1/2)

Transformer vs LLaMA



Transformer
("Attention is all you need")



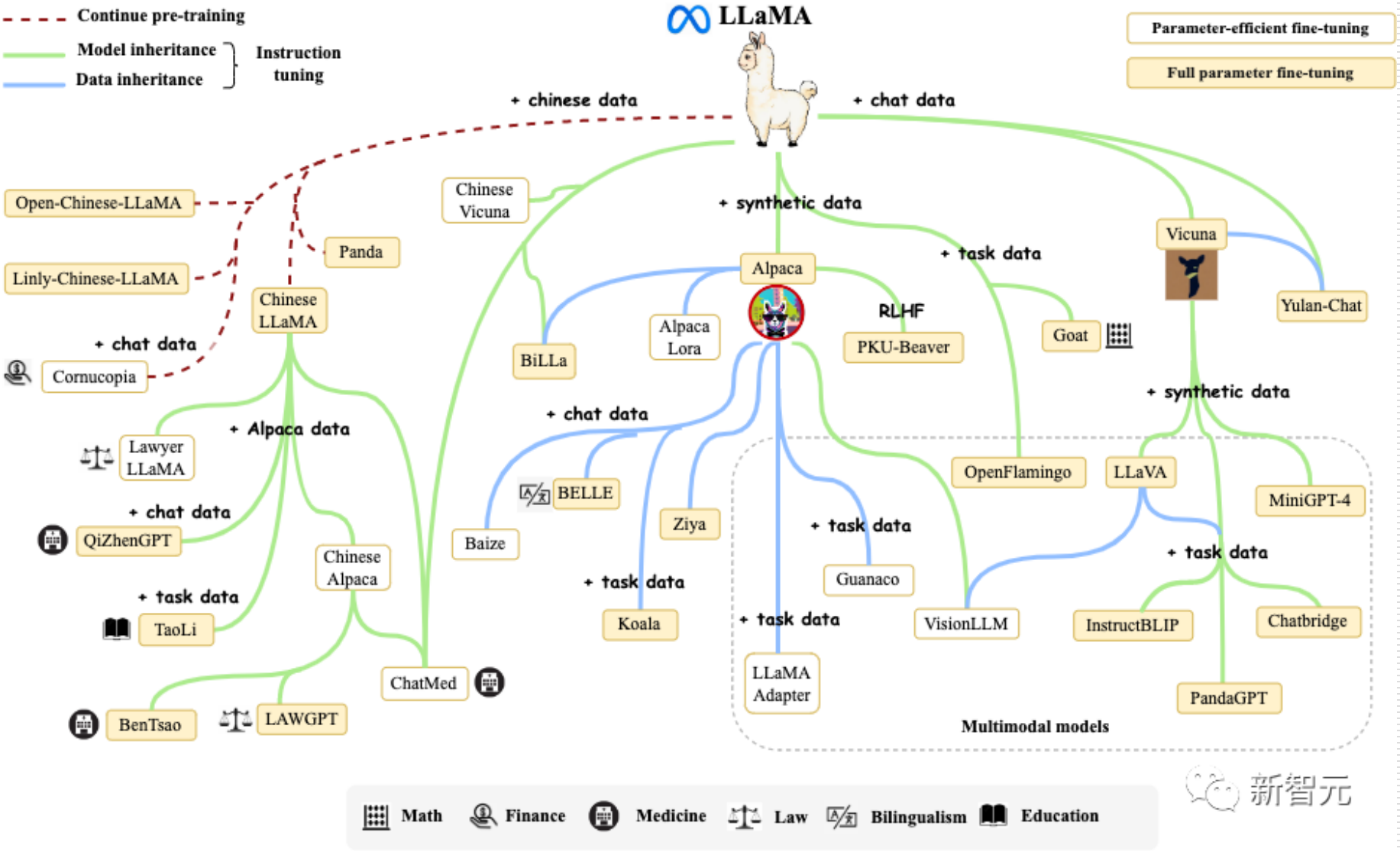
LLaMA

Transformer vs LLaMA (2/2)

- Like the GPT series, the LLaMA model is also a *Decoder-only* architecture.

- Modifications
  - Pre-normalization : Applies RMSNorm to the input of each transformer layer
  - Activation : Uses SwiGLU instead of ReLU/GeLU
  - Positional Encoding: Incorporates RoPE in each layer instead of positional embeddings

LLaMA Family (1/3)



LLaMA Family (2/3)

- An open and efficient large-scale basic language model released by Meta AI
  - with four versions: 7B, 13B, 33B, and 65B

- Use public dataset for training
  - Wikipedia、GitHub、ArXiv、English CommonCrawl、C4
  - total 1.4T tokens

LLaMA Family (3/3)

LLaMA Series

JEFF YANG

LLaMA 1 — February 2023

Release Model
Basic: 7B, 13B, 33B, and 65B parameters

Context Length 2K

Pretraining Token
1-1.4 Trillion tokens

Architecture
- SwiGLU replace ReLU
- Pre-normalization
- Rotary Embeddings

Tokenizer
BPE model based on sentencepiece
Vocabulary Size 32K

Commercially Available No

LLaMA 2 — July 2023

Release Model
Basic: 7B, 13B, 34B, and 70B parameters
Chat: Optimized for dialogue use cases

Knowledge Cutoff
September 2022, some tuning data up to July 2023.

Context Length 4K

Pretraining Token
2 Trillion tokens

Architecture
- Inherit from v1
- Grouped Query Attention (only 34B, 70B Model)
- Ghost Attention

Tokenizer
BPE model based on sentencepiece
Vocabulary Size 32K

Post-training
SFT, Rejection Sampling, PPO and RLHF

Commercially Available YES

LLaMA 3 — April 2024

Release Model
Basic: 8B and 70B parameters
Instruct: Additional instruction-tuning, improved the performance on reasoning and coding tasks

Knowledge Cutoff
8B: March, 2023
70B: December, 2023

Context Length 8K

Pretraining Token
>15 Trillion tokens

Architecture
- Inherit from v2
- Fully Adopts GQA

Tokenizer
Tiktoken: A fast BPE tokeniser
Vocabulary Size 128K

Post-training
SFT, Rejection Sampling, PPO and DPO

Commercially Available YES

LLaMA

LLaMA

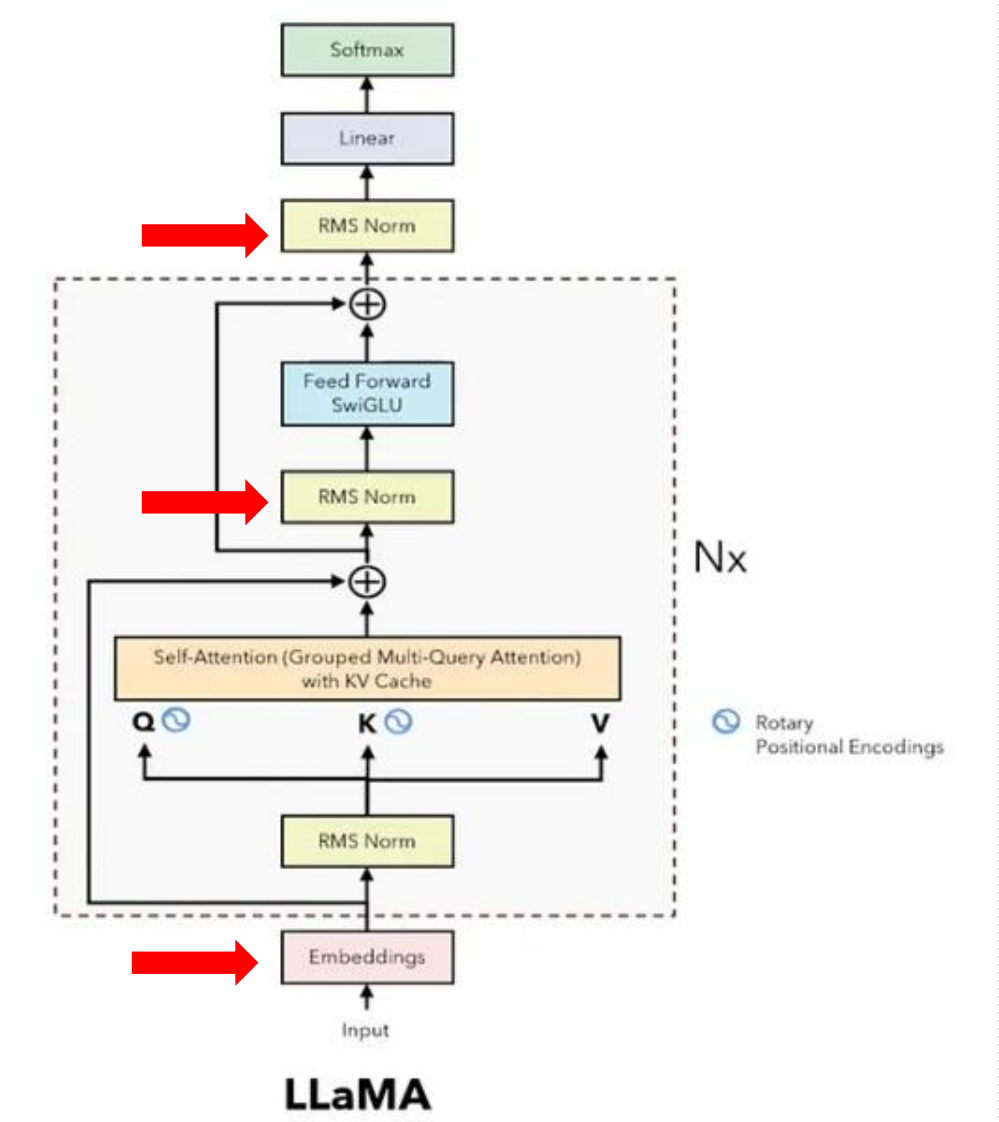
- LLaMA series pioneered by Meta AI (open-source)
  - open and efficient foundation language models (February 2023)
  - mark the official beginning of the *open-source LLMs* era
  - LLaMA-13B outperforms GPT-3 (175B) in most benchmarks
  - LLaMA-65B is quite competitive with the best models Chinchilla70B and PaLM540B.

Models (LLaMA)

params	dimension	n heads	n layers	learning rate	batch size	n tokens
6.7B	4096	32	32	$3.0e^{-4}$	4M	1.0T
13.0B	5120	40	40	$3.0e^{-4}$	4M	1.0T
32.5B	6656	52	60	$1.5e^{-4}$	4M	1.4T
65.2B	8192	64	80	$1.5e^{-4}$	4M	1.4T

Table 2: **Model sizes, architectures, and optimization hyper-parameters.**

Why RMSNorm?



Why RMSNorm?

- Requires less computation compared to Layer Normalization.

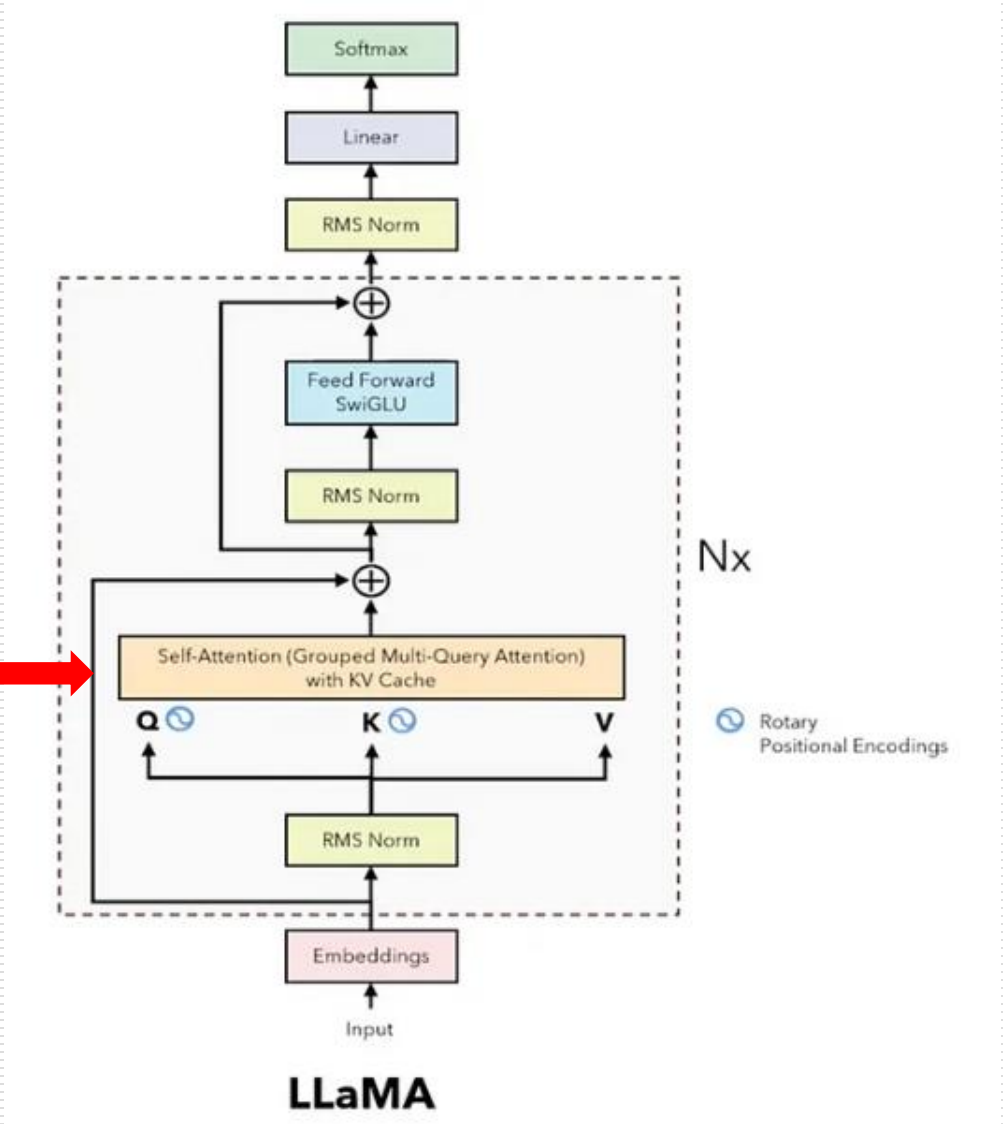
- Works well in practice.

$$\bar{a}_i = \frac{a_i}{\text{RMS}(\mathbf{a})} g_i, \quad \text{where } \text{RMS}(\mathbf{a}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} a_i^2}.$$

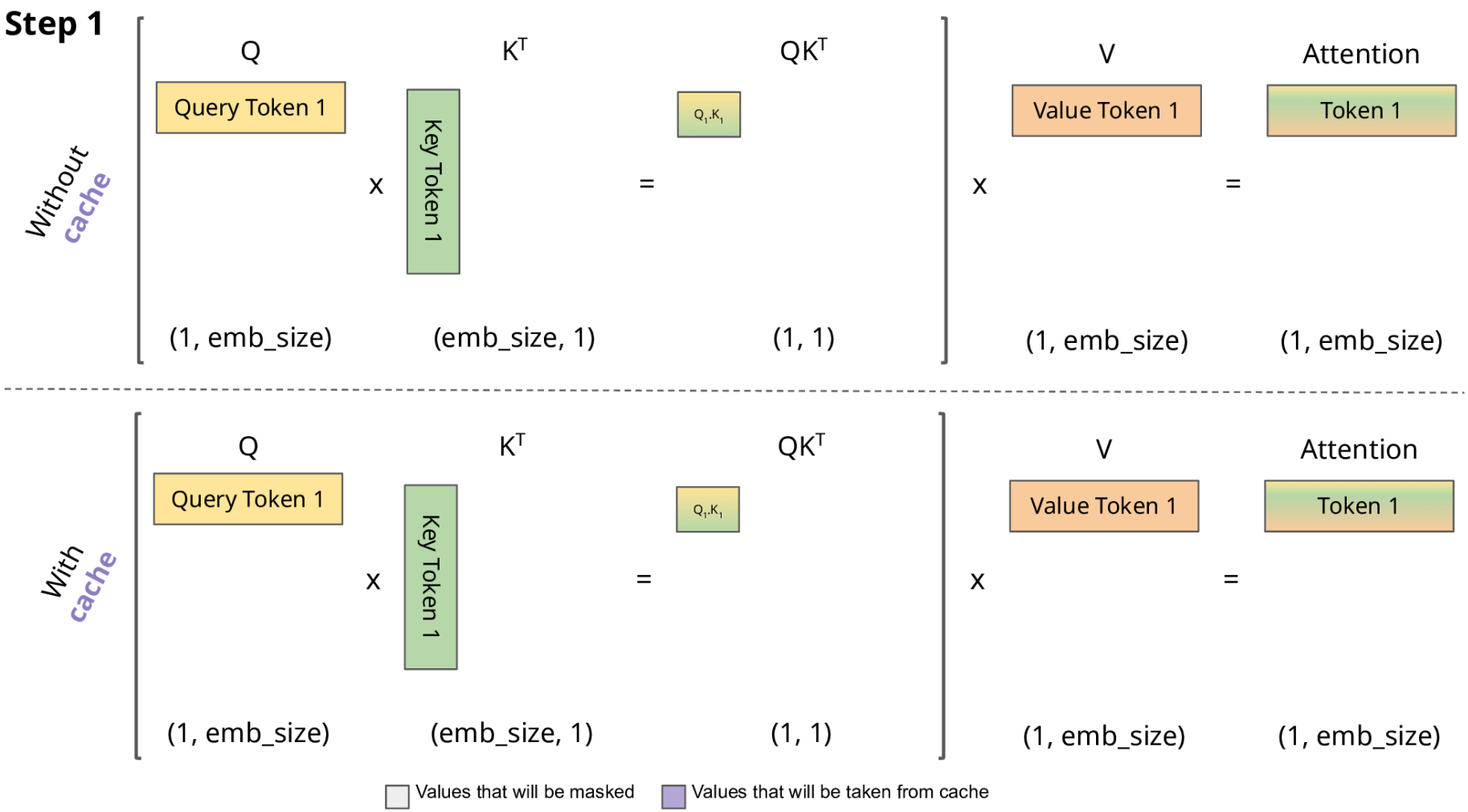
Next Token Prediction

- At every step of the inference, we are only interested in the last token output by the model.
- We already have the previous ones.
- KV cache is there a way to make the model do less computation on the token it has already seen.

Self-Attention



Transformers KV Caching

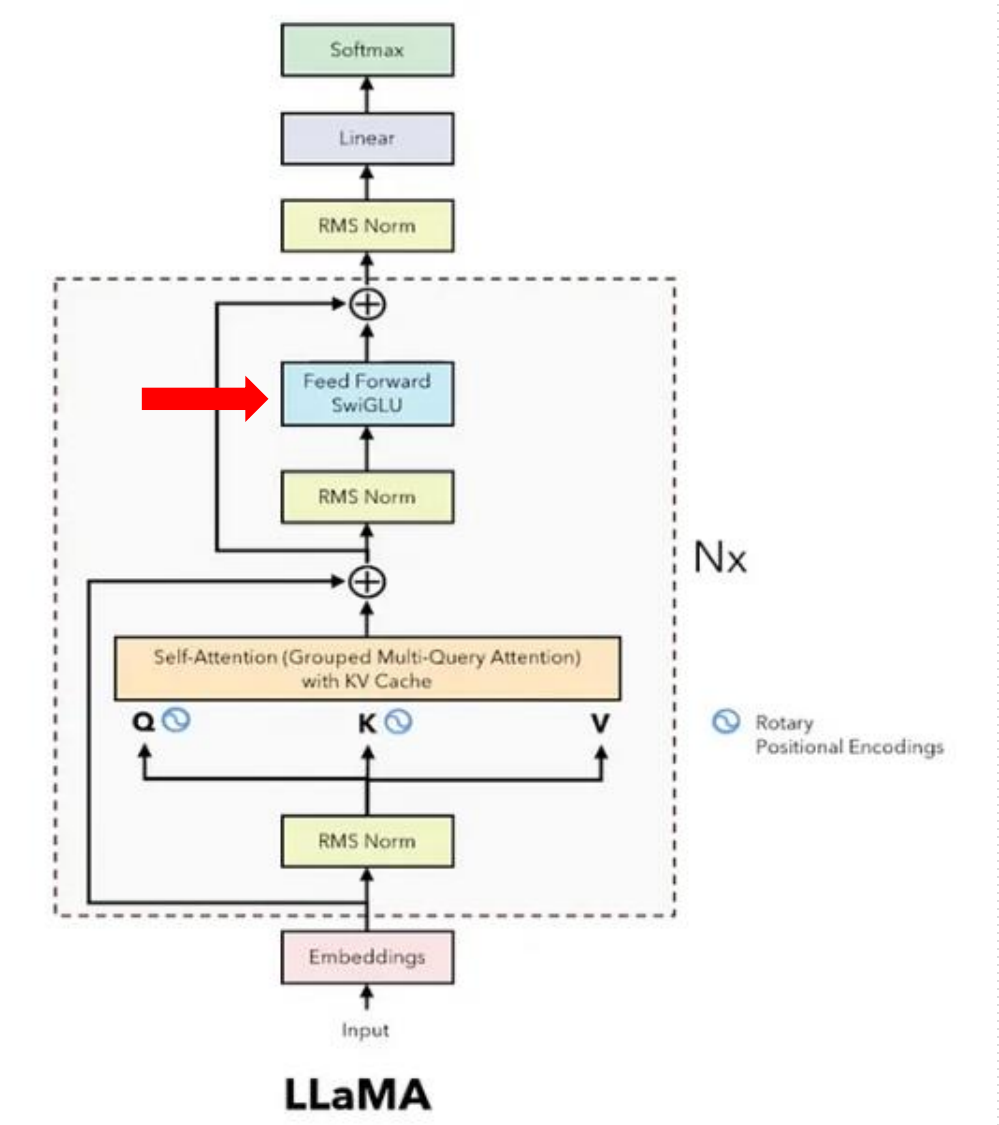


Self-Attention with KV Cache

- The information needed for next token prediction with self-attention is the Key/Value values of previous tokens.

- Instead of recalculating every time there is a new token, we can use the cache mechanism to greatly improve the computing speed.

Why SwiGLU?

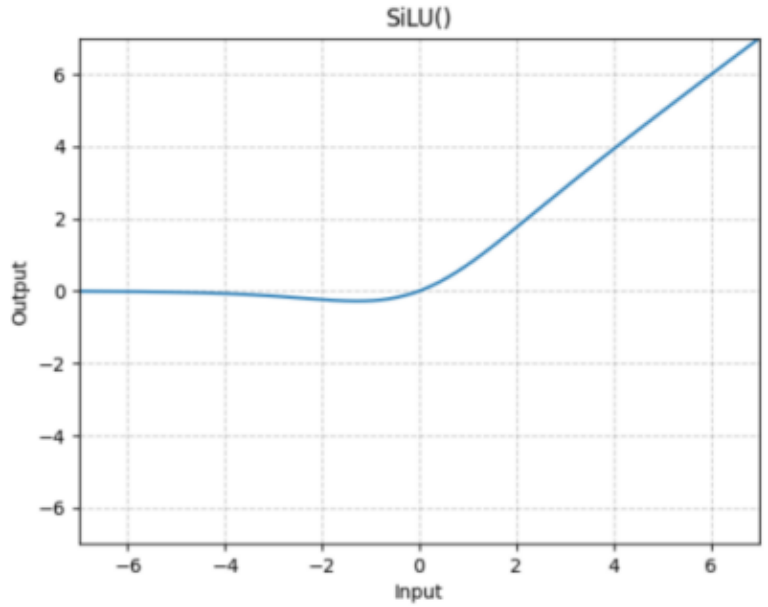


SwiGLU Activation Function

$$FFN_{SwiGLU}(x, W, V, W_2) = (Swish_1(xW) \cdot xV)\, W_2$$

We use the swish function with $\beta = 1$. In this case it's called the **Sigmoid Linear Unit** (**SiLU**) function.

$$\text{swish}(x) = x\,\text{sigmoid}(\beta x) = \frac{x}{1 + e^{-\beta x}}$$



How well does SwiGLU perform?

Table 1: Heldout-set log-perplexity for Transformer models on the segment-filling task from [Raffel et al., 2019]. All models are matched for parameters and computation.

Training Steps	65,536	524,288
FFN_{ReLU} (*baseline*)	1.997 (0.005)	1.677
FFN_{GELU}	1.983 (0.005)	1.679
FFN_{Swish}	1.994 (0.003)	1.683
FFN_{GLU}	1.982 (0.006)	1.663
$FFN_{Bilinear}$	1.960 (0.005)	1.648
FFN_{GEGLU}	**1.942** (0.004)	**1.633**
FFN_{SwiGLU}	**1.944** (0.010)	**1.636**
FFN_{ReGLU}	1.953 (0.003)	1.645

Table 2: GLUE Language-Understanding Benchmark [Wang et al., 2018] (dev).

	Score Average	CoLA MCC	SST-2 Acc	MRPC F1	MRPC Acc	STSB PCC	STSB SCC	QQP F1	QQP Acc	MNLIm Acc	MNLImm Acc	QNLI Acc	RTE Acc
FFN_{ReLU}	83.80	51.32	94.04	**93.08**	**90.20**	89.64	89.42	89.01	91.75	85.83	86.42	92.81	80.14
FFN_{GELU}	83.86	53.48	94.04	92.81	**90.20**	89.69	89.49	88.63	91.62	85.89	86.13	92.39	80.51
FFN_{Swish}	83.60	49.79	93.69	92.31	89.46	89.20	88.98	88.84	91.67	85.22	85.02	92.33	81.23
FFN_{GLU}	84.20	49.16	94.27	92.39	89.46	89.46	89.35	88.79	91.62	86.36	86.18	92.92	**84.12**
FFN_{GEGLU}	84.12	53.65	93.92	92.68	89.71	90.26	90.13	89.11	91.85	86.15	86.17	92.81	79.42
$FFN_{Bilinear}$	83.79	51.02	**94.38**	92.28	89.46	90.06	89.84	88.95	91.69	**86.90**	**87.08**	92.92	81.95
FFN_{SwiGLU}	84.36	51.59	93.92	92.23	88.97	**90.32**	**90.13**	**89.14**	**91.87**	86.45	86.47	**92.93**	83.39
FFN_{ReGLU}	**84.67**	**56.16**	**94.38**	92.06	89.22	89.97	89.85	88.86	91.72	86.20	86.40	92.68	81.59
[Raffel et al., 2019]	83.28	53.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28
ibid. stddev.	0.235	1.111	0.569	0.729	1.019	0.374	0.418	0.108	0.070	0.291	0.231	0.361	1.393

Common Sense Reasoning

- LLaMA-13B outperforms GPT-3 on most benchmarks despite being 10x smaller.

		BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA
GPT-3	175B	60.5	81.0	-	78.9	70.2	68.8	51.4	57.6
Gopher	280B	79.3	81.8	50.6	79.2	70.1	-	-	-
Chinchilla	70B	83.7	81.8	51.3	80.8	74.9	-	-	-
PaLM	62B	84.8	80.5	-	79.7	77.0	75.2	52.5	50.4
PaLM-cont	62B	83.9	81.4	-	80.6	77.0	-	-	-
PaLM	540B	**88.0**	82.3	-	83.4	**81.1**	76.6	53.0	53.4
LLaMA	7B	76.5	79.8	48.9	76.1	70.1	72.8	47.6	57.2
	13B	78.1	80.1	50.4	79.2	73.0	74.8	52.7	56.4
	33B	83.1	82.3	50.4	82.8	76.0	**80.0**	**57.8**	58.6
	65B	85.3	**82.8**	**52.3**	**84.2**	77.0	78.9	56.0	**60.2**

Table 3: **Zero-shot performance on Common Sense Reasoning tasks.**

LLaMA2

Models (LLaMA2)

- LLaMA 2 released in July 2023.

- 34B and 70B use Grouped-Query Attention (GQA) for improved inference scalability.

	Training Data	Params	Context Length	GQA	Tokens	LR
LLAMA 1	*See Touvron et al. (2023)*	7B	2k	✗	1.0T	3.0×10^{-4}
		13B	2k	✗	1.0T	3.0×10^{-4}
		33B	2k	✗	1.4T	1.5×10^{-4}
		65B	2k	✗	1.4T	1.5×10^{-4}
LLAMA 2	*A new mix of publicly available online data*	7B	4k	✗	2.0T	3.0×10^{-4}
		13B	4k	✗	2.0T	3.0×10^{-4}
		34B	4k	✓	2.0T	1.5×10^{-4}
		70B	4k	✓	2.0T	1.5×10^{-4}

Table 1: LLAMA 2 family of models. Token counts refer to pretraining data only. All models are trained with a global batch-size of 4M tokens. Bigger models — 34B and 70B — use Grouped-Query Attention (GQA) for improved inference scalability.

Grouped Query Attention

- A common alternative is to use Multi-Query Attention (MQA) or Grouped Query Attention (GQA) to replace Multi-Head Attention (MHA).

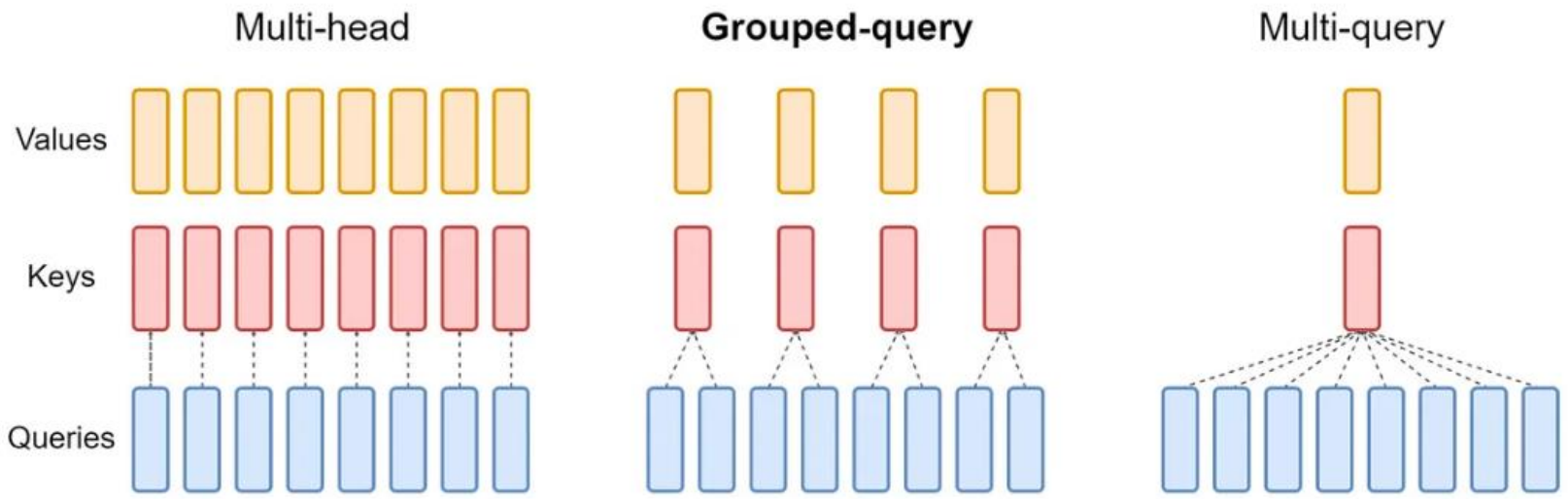


Figure 2: Overview of grouped-query method. Multi-head attention has H query, key, and value heads. Multi-query attention shares single key and value heads across all query heads. Grouped-query attention instead shares single key and value heads for each *group* of query heads, interpolating between multi-head and multi-query attention.

GQA Performance

- GQA is not as extreme as MQA, which only retains a single Key/Value. Instead, it groups Key/Value and shares Query head.

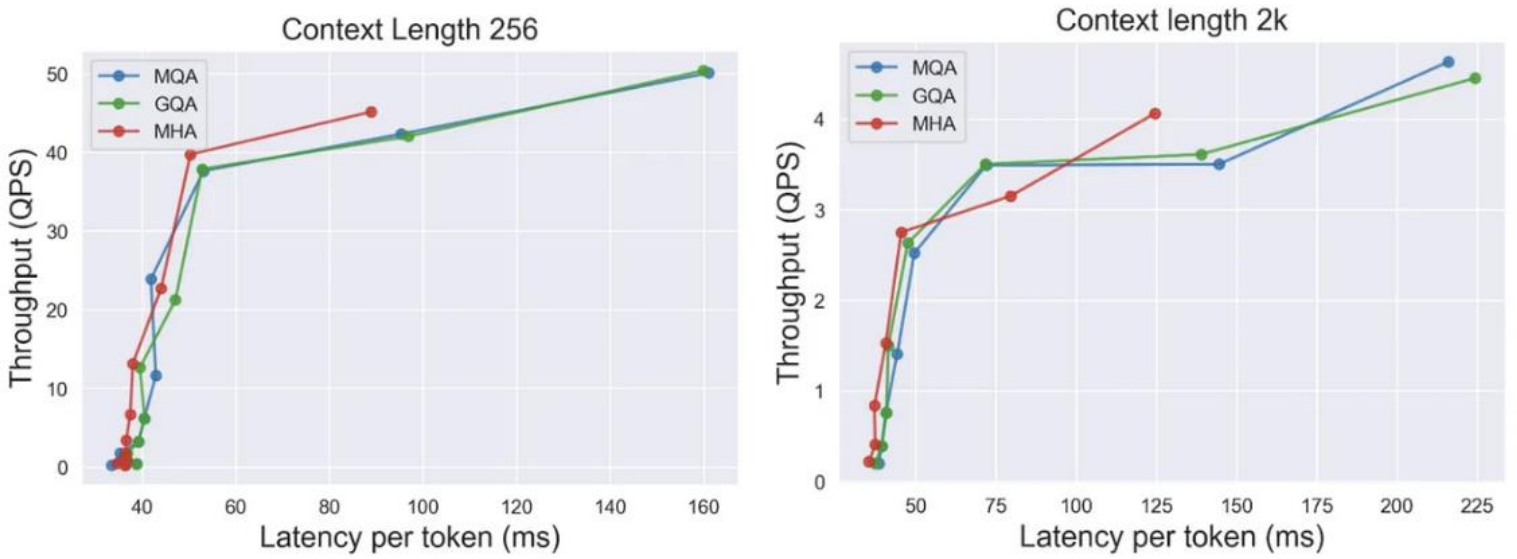


Figure 24: Multi-query variants enable higher throughput with larger batch sizes, and show similar latency on smaller batches. Output length is fixed at 128 tokens. The first data point corresponds to batch size 1, and then we double it until the model runs out of memory. The MHA variant triggers an out-of-memory error at a batch size of 1024 for a context of 256 tokens and at a batch size of 128 for 2k context, whereas MQA and GQA have successful runs in those settings.

Overall Performance

Model	Size	Code	Commonsense Reasoning	World Knowledge	Reading Comprehension	Math	MMLU	BBH	AGI Eval
MPT	7B	20.5	57.4	41.0	57.5	4.9	26.8	31.0	23.5
	30B	28.9	64.9	50.0	64.7	9.1	46.9	38.0	33.8
Falcon	7B	5.6	56.1	42.8	36.0	4.6	26.2	28.0	21.2
	40B	15.2	69.2	56.7	65.7	12.6	55.4	37.1	37.0
LLAMA 1	7B	14.1	60.8	46.2	58.5	6.95	35.1	30.3	23.9
	13B	18.9	66.1	52.6	62.3	10.9	46.9	37.0	33.9
	33B	26.0	70.0	58.4	67.6	21.4	57.8	39.8	41.7
	65B	30.7	70.7	60.5	68.6	30.8	63.4	43.5	47.6
LLAMA 2	7B	16.8	63.9	48.9	61.3	14.6	45.3	32.6	29.3
	13B	24.5	66.9	55.4	65.8	28.7	54.8	39.4	39.1
	34B	27.8	69.9	58.7	68.0	24.2	62.6	44.1	43.4
	70B	**37.5**	**71.9**	**63.6**	**69.4**	**35.2**	**68.9**	**51.2**	**54.2**

Table 3: Overall performance on grouped academic benchmarks compared to open-source base models.

Comparison Table

FEATURE	LLAMA	LLAMA 2
Number of parameters	65B	70B, 13B, 7B
Training data	1.56T tokens	2.2T tokens
Context length	2048 tokens	4096 tokens
Attention mechanism	Transformer	Grouped-query attention
Fine-tuned models	No	Yes (Llama 2-Chat)
Performance	Good	Better than Llama on most benchmarks
Computational requirements	High	Very high (70B model)
Availability	Open source	Open source
Reinforcement learning from human feedback	No	Yes
Number of languages supported	20 languages	20 languages
Suitable for	General-purpose tasks, such as answering questions, generating text, and translating languages	Best for more demanding tasks, such as reasoning, coding, and proficiency tests

LLaMA3

LLaMA3

- LLaMA 3 released in April 2024.
- Improvement mainly comes from scaling law
  - enhance the model through more and higher-quality data

	Meta Llama 3 8B	Gemma 7B - It Measured	Mistral 7B Instruct Measured
MMLU 5-shot	**68.4**	53.3	58.4
GPQA 0-shot	**34.2**	21.4	26.3
HumanEval 0-shot	**62.2**	30.5	36.6
GSM-8K 8-shot, CoT	**79.6**	30.6	39.9
MATH 4-shot, CoT	**30.0**	12.2	11.0

	Meta Llama 3 70B	Gemini Pro 1.5 Published	Claude 3 Sonnet Published
MMLU 5-shot	**82.0**	81.9	79.0
GPQA 0-shot	39.5	**41.5** CoT	38.5 CoT
HumanEval 0-shot	**81.7**	71.9	73.0
GSM-8K 8-shot, CoT	**93.0**	91.7 11-shot	92.3 0-shot
MATH 4-shot, CoT	50.4	**58.5** Minerva prompt	40.5

Comparison Table

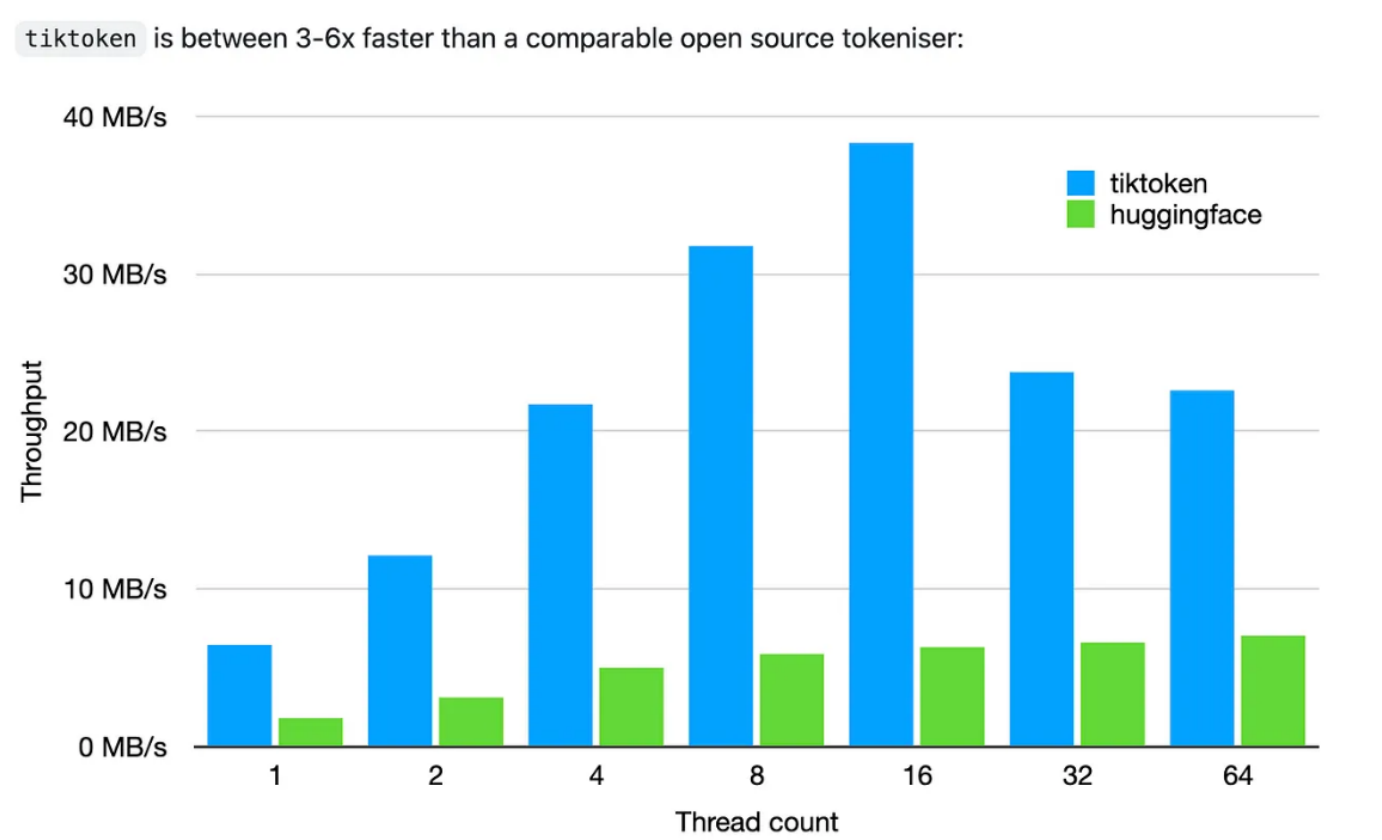
FEATURE	Llama 3	Llama 2
Number of Parameters	8B, 70B	70B, 13B, 7B
Training Data	15T tokens	2.2T tokens
Context Length	8192 tokens	4096 tokens
Attention Mechanism	Grouped-query attention	Grouped-query attention
Fine-Tuned Models	Yes	Yes
Performance	Better than Llama 2 on all benchmarks	Better than Llama 1 on most benchmarks
Computational Requirements	Very high (70B model)	Very high (70B model)
Availability	Open source	Open source
Reinforcement learning from human feedback	Yes	Yes
Number of languages supported	30 languages	20 languages
Suitable for	Best for more demanding tasks, such as reasoning, coding, and proficiency tests	Good for more demanding tasks, such as reasoning, coding, and proficiency tests

Models (LLaMA3)

- Llama 3 uses a tokenizer with a vocabulary of 128K tokens that encodes language much more efficiently.
- Adopted grouped query attention (GQA) across both the 8B and 70B sizes.
- Trained on sequences of 8192 tokens, using a mask to ensure self-attention does not cross document boundaries.

Tokenizer

- Titoken is the tokenizer used by llama3.
  - sentencepiece was used before
  - vocabulary size : 32K -> 128K



LLaMA Configuration

Attribute	Llama-2-7b	Llama-2-70b	Llama-3-8b	Llama-3-70b
hidden_size	4096	8192	4096	8192
intermediate_size	11008	28672	14336	28672
num_attention_heads	32	64	32	64
num_key_value_heads	32	8	8	8
num_hidden_layers	32	80	32	80
max_position_embeddings	4096	4096	8192	8192
hidden_act	silu	silu	silu	silu
vocab_size	32000	32000	128256	128256

Huggingface Environment

What is Huggingface ?

- Open source repository of models, datasets, and tools.
  - transformers library designed NLP applications

- Thousands of language models freely available to use.
  - LangChain : open-source framework for connecting LLMs

Hugging Face

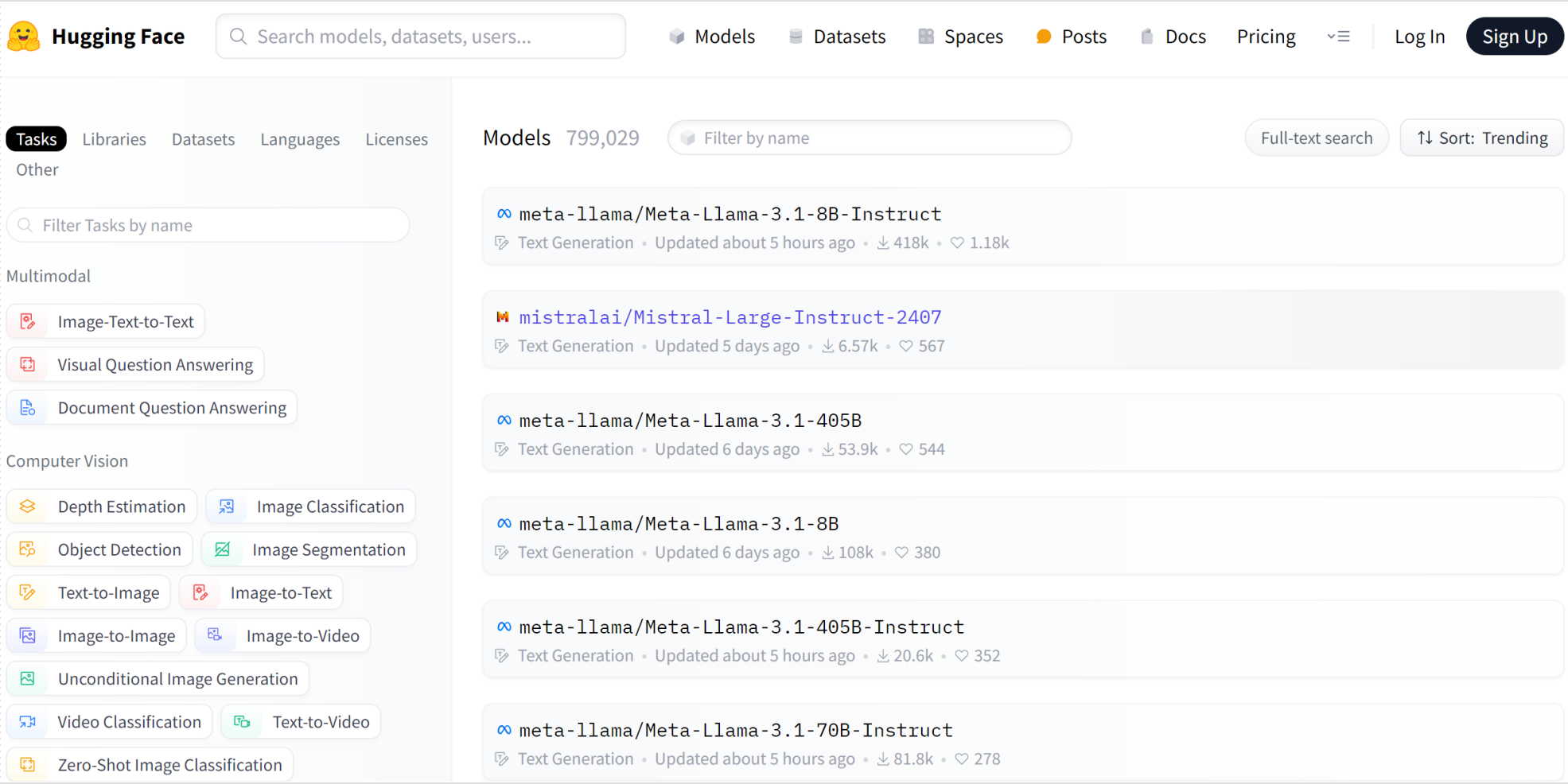
https://huggingface.co/

Huggingface

- An open-source platform of Python packages
- Specializes in NLP research
- Hosting a hub for model weights
- Many educational blogs and tutorial
  - Main Page: https://huggingface.co/
  - NLP Tutorial: https://huggingface.co/learn/nlp-course/chapter1/1

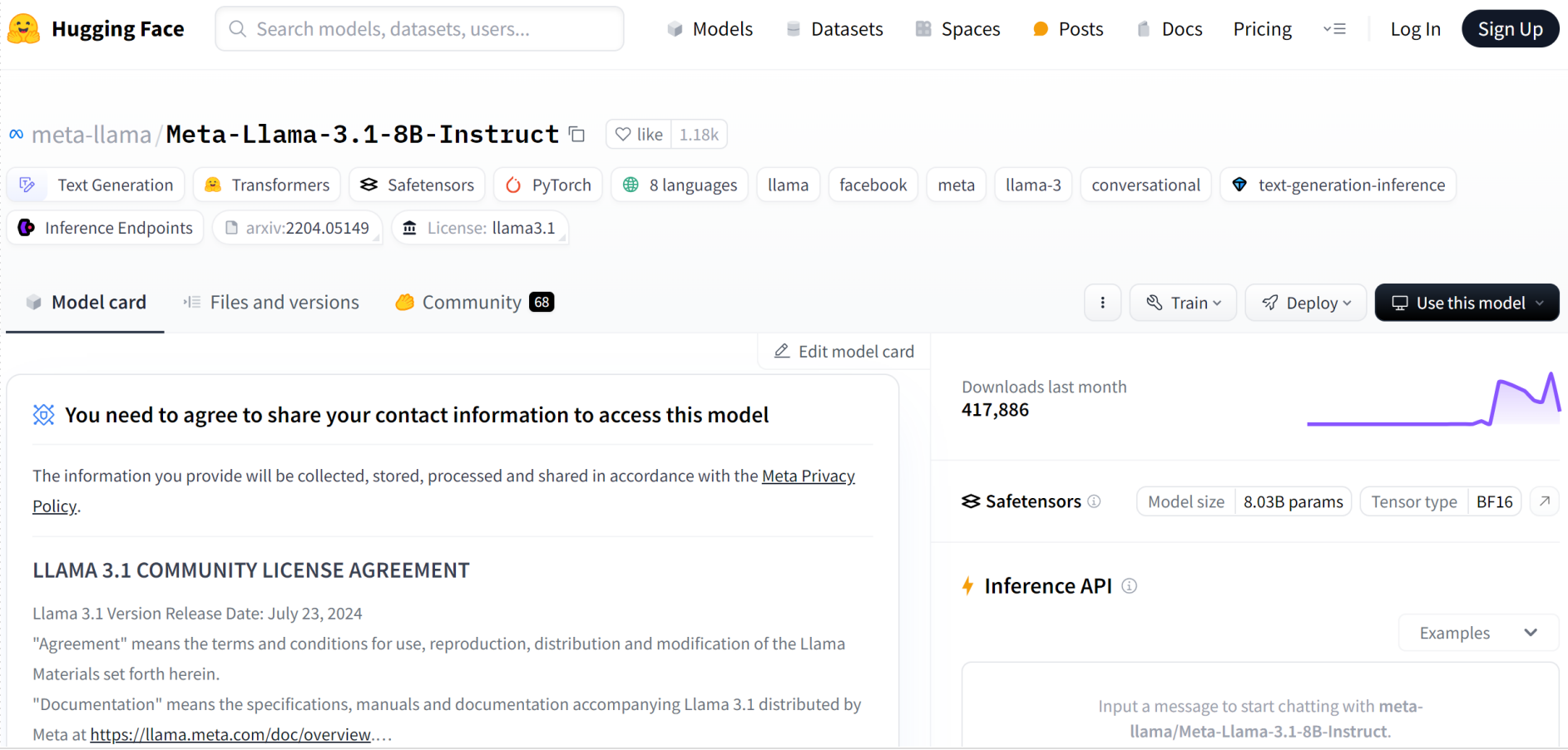
Models

- Open-source pre-trained in here



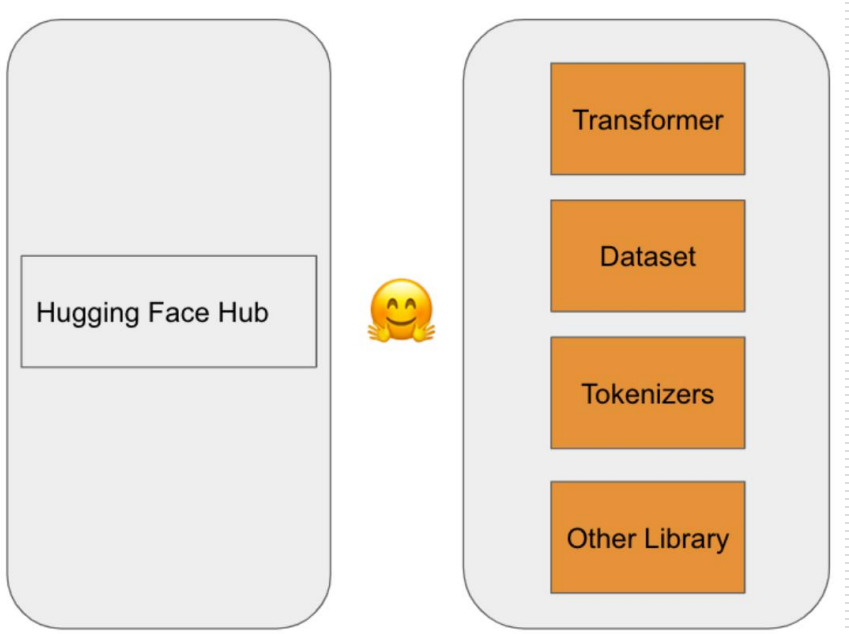
Model Details

- There is also a Hosted inference API on the right panel, where you can chat with this model.



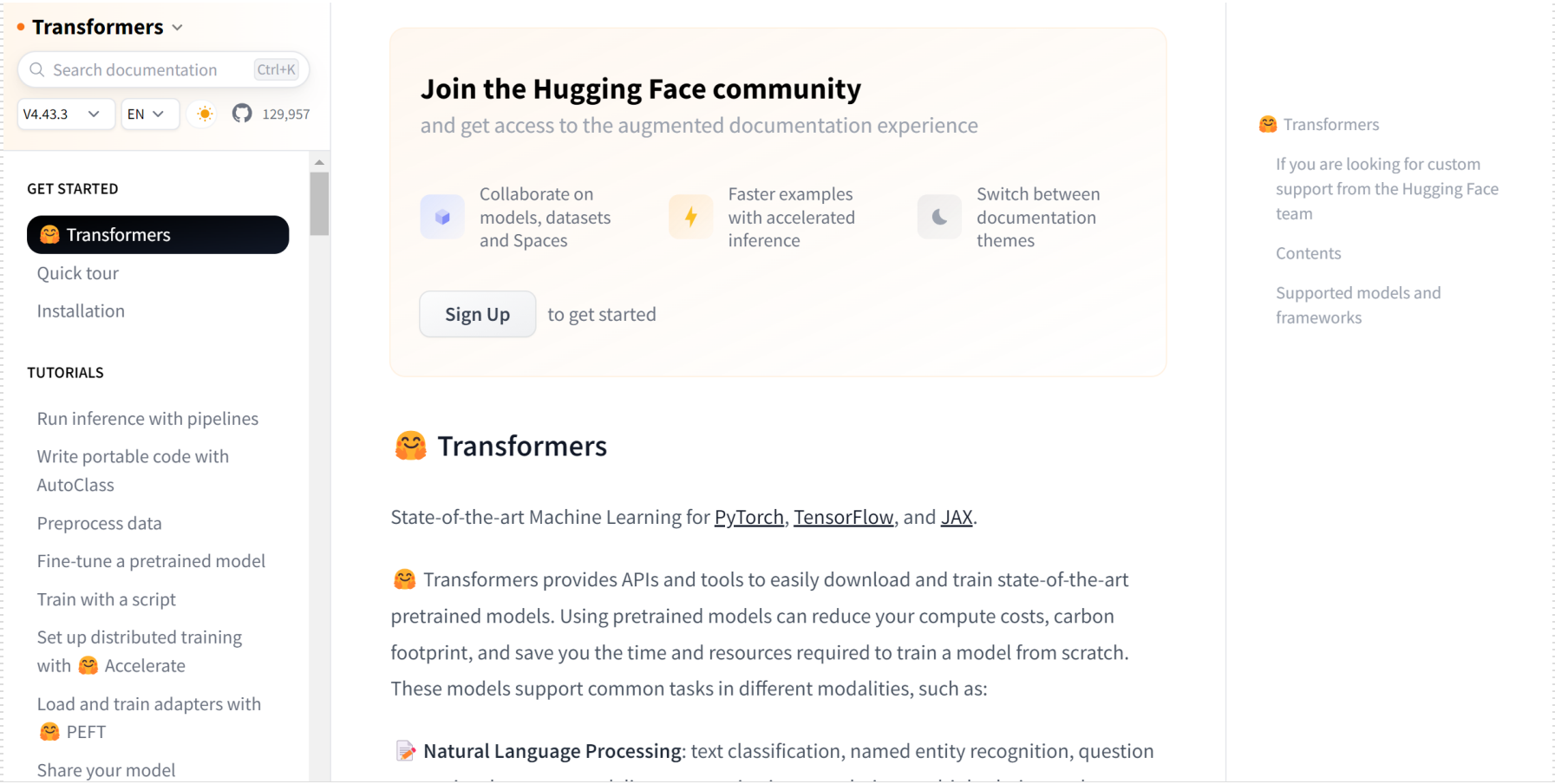
Huggingface Library

- Transformer
  - Many transformer architecture, based on both Pytorch and Tensorflow
- Tokenizers
  - Tokenizing method and setting
- Dataset
  - Corpus repository
- Accelerate
  - Distributed training



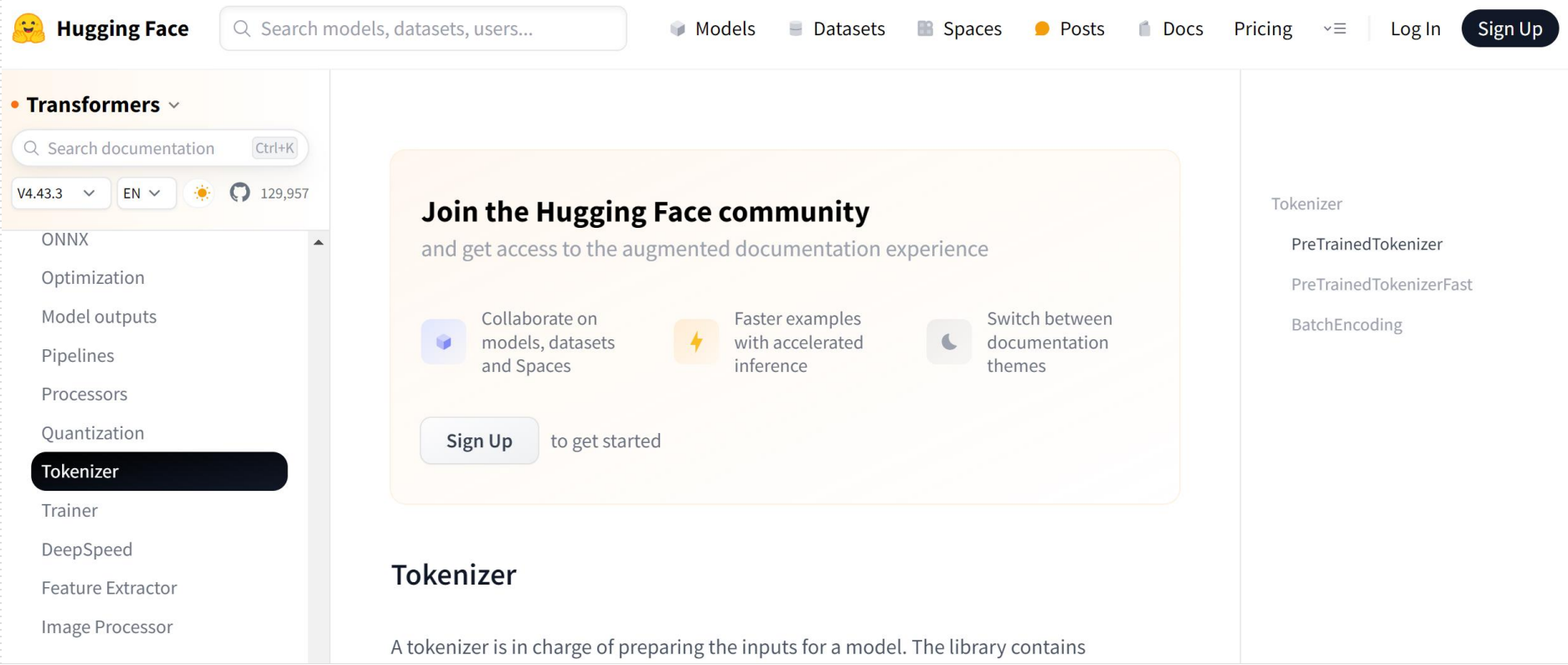
Transformers

- Providing an open-source to deploy various pre-trained language models.



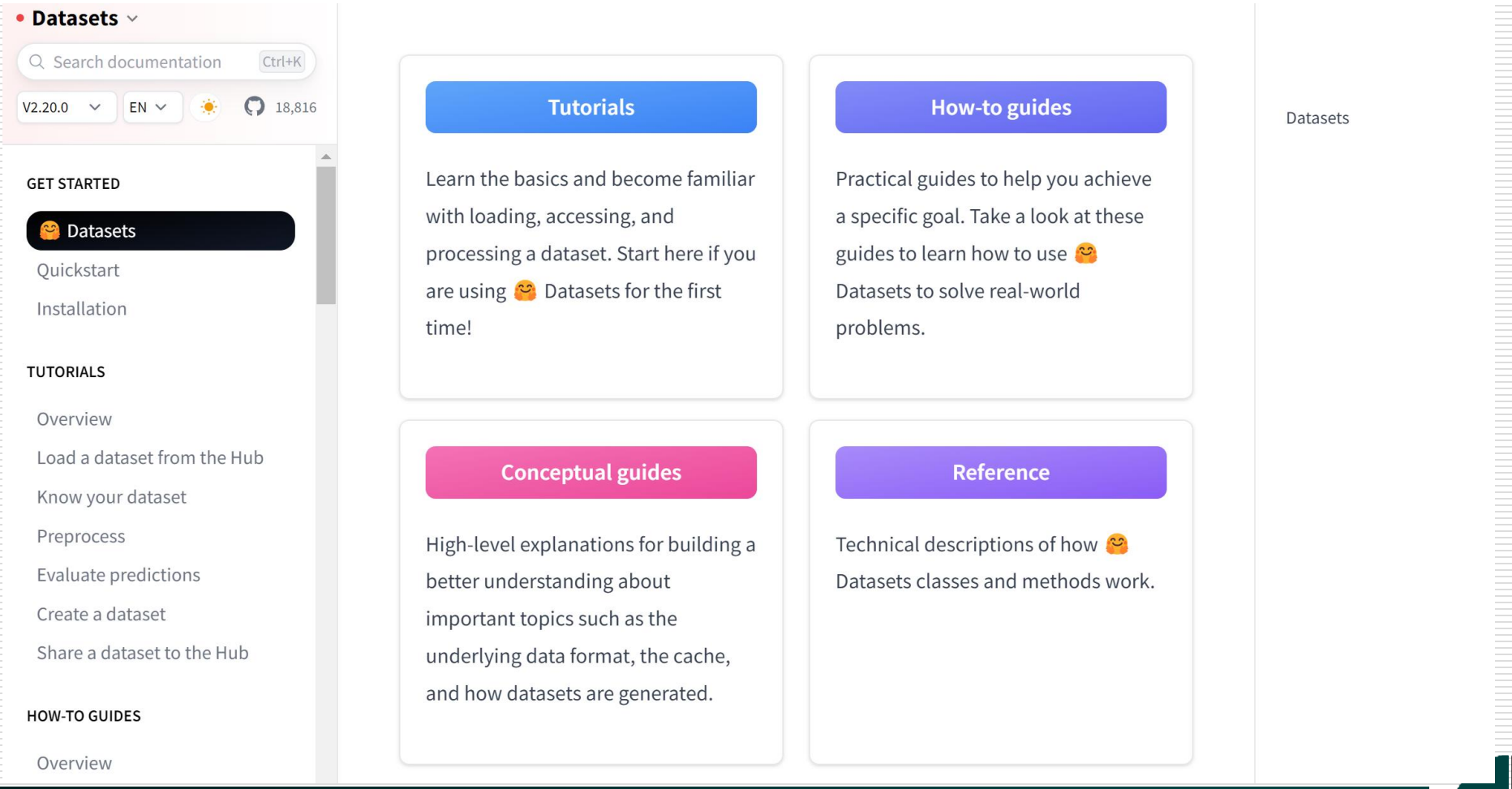
Tokenizers

- Break out every word in a sentence.
- Provide pre-processing and post-processing functions.



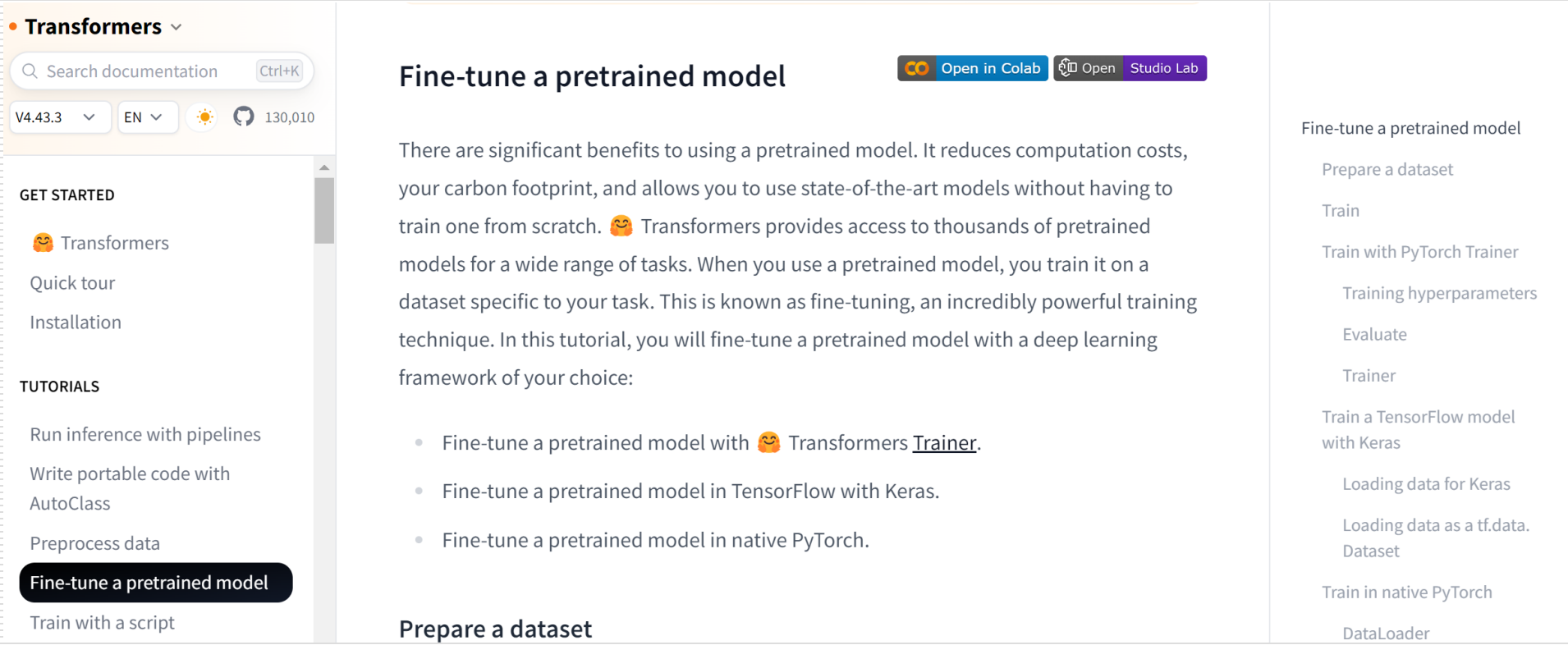
Datasets

- Provides an API to obtain data sets from Hub, and can also be used interactively with Pandas.



Fine-tune Model

- You can learn how to use Huggingface Transformer fine-tune model from here
  - https://huggingface.co/docs/transformers/training



Exercise

Exercise

- 1 (50%) : Please explain the training methods in ChatGPT.
- 2 (50%): Please explain why do we mostly use fine-tuning instead of training from scratch on language models? And explain what is instruction fine-tuning?
- Please submit your Report as STuser_xx_hw10.pdf file.

Reference

Reference (1/4)

- Generative AI
  - 【生成式AI導論 2024】第1講：生成式AI是什麼？ (youtube.com)
- GPT
  - GPT-1: Improving Language Understanding by Generative Pre-Training
  - GPT-2: Language Models are Unsupervised Multitask Learners
  - GPT-3: Language Models are Few-Shot Learners
- 李宏毅教授的GPT介紹
  - ChatGPT (可能)是怎麼煉成的 - GPT 社會化的過程 – YouTube
  - 【生成式AI】ChatGPT 原理剖析 (1/3) — 對 ChatGPT 的常見誤解
  - 【生成式AI】ChatGPT 原理剖析 (2/3) — 預訓練 (Pre-train)
  - 【生成式AI】ChatGPT 原理剖析 (3/3) — ChatGPT 所帶來的研究問題
  - [DLHLP 2020] 來自獵人暗黑大陸的模型 GPT-3 (youtube.com)

Reference (2/4)

- Positional Encoding
  - https://medium.com/@weidagang/demystifying-transformers-positional-encoding-955dd018c76c
  - https://medium.com/image-processing-with-python/positional-encoding-in-the-transformer-model-e8e9979df57f
  - https://alu2019.home.blog/2021/02/06/nlp-2-attention-transformer-model/

- RoPE
  - https://arxiv.org/abs/2104.09864
  - https://hackmd.io/@_E6GATP9Sz2h9WVqOqMDbg/SJgIu7xRp?utm_source=preview-mode&utm_medium=rec

Reference (3/4)

- LLaMA
  - https://github.com/hkproj/pytorch-llama-notes/blob/main/README.md
  - https://arxiv.org/abs/2002.05202
  - https://arxiv.org/pdf/1706.03762
- LLaMA2
  - https://arxiv.org/abs/2307.09288
  - https://arxiv.org/abs/2307.09288
  - https://www.apps4rent.com/blog/llama-vs-llama-2/
- LLaMA3
  - https://zhuanlan.zhihu.com/p/37189203
  - https://www.apps4rent.com/blog/llama-3-vs-llama-2/

Reference (4/4)

- Huggingface
  - https://ithelp.ithome.com.tw/articles/10291757
  - https://huggingface.co/

Thank you