Understanding the Lifecycle of Large Language Models (part 1)

From Model Definition to Deployment

Nathan Crock
Department of Scientific Computing
Florida State University

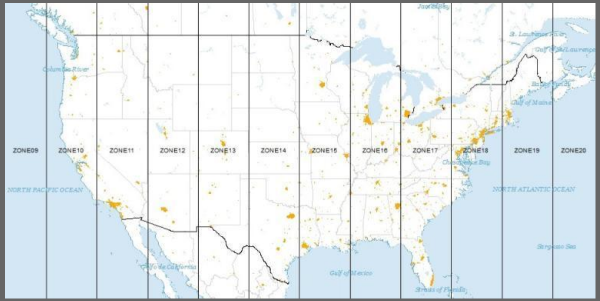
Oct 6th, 2023
FSU, Machine Learning Seminar

Outline

Here is what we will do today…

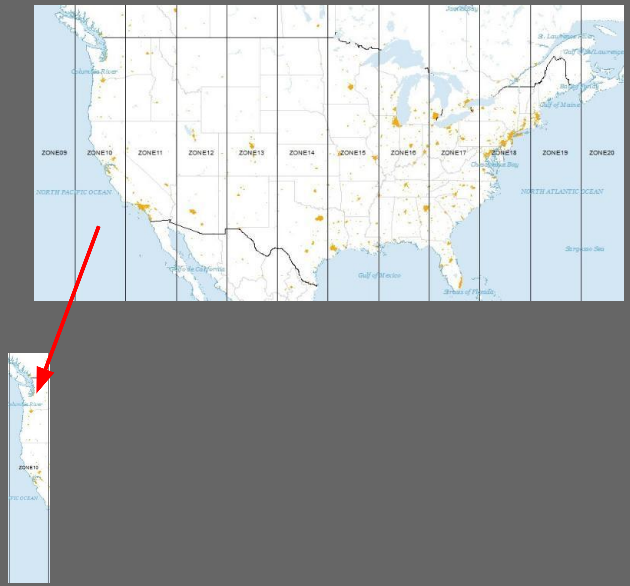
Outline

Here is what we will do today…



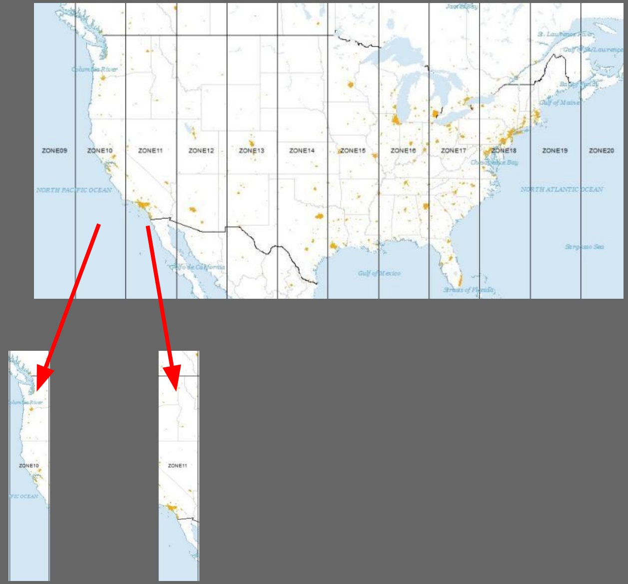
Outline

Here is what we will do today…



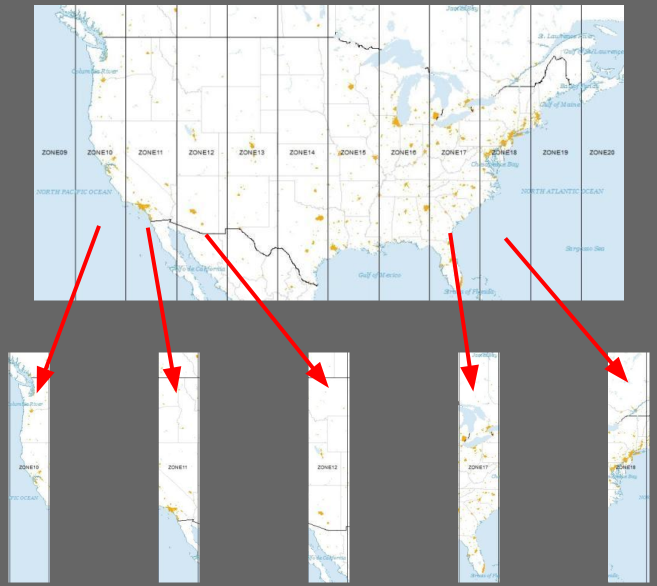
Outline

Here is what we will do today…



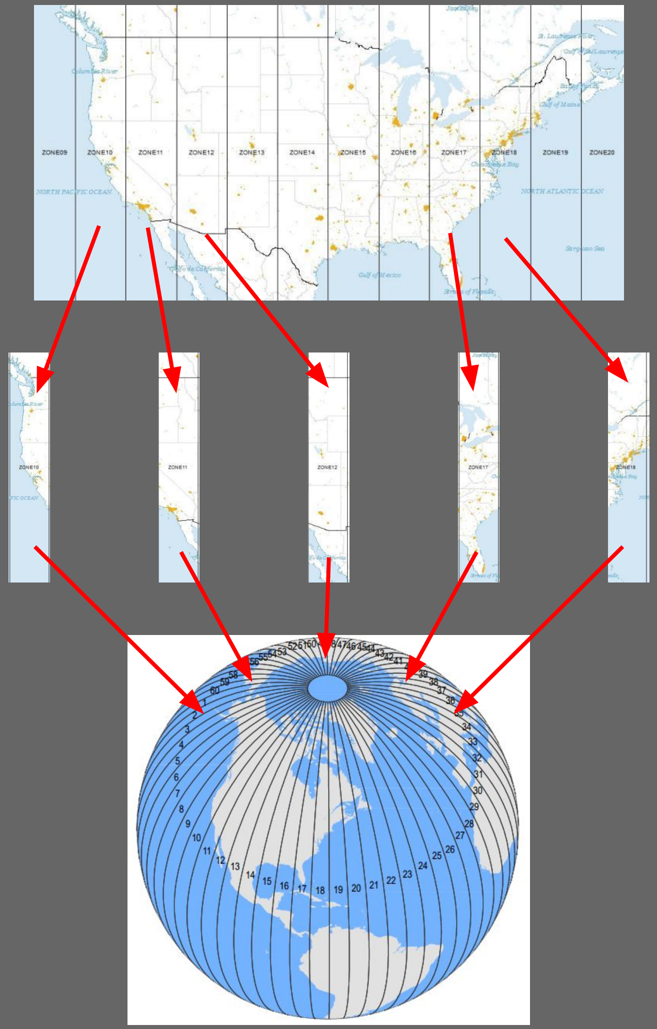
Outline

Here is what we will do today…



Outline

Here is what we will do today…



Outline

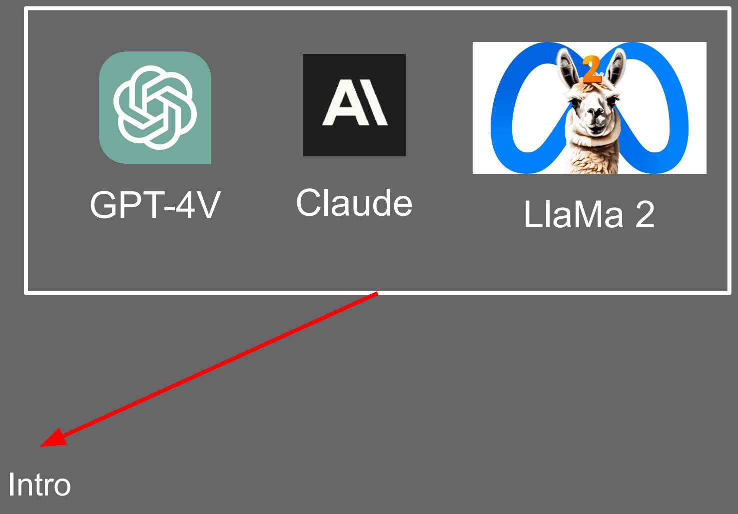
GPT-4V

Claude

LlaMa 2

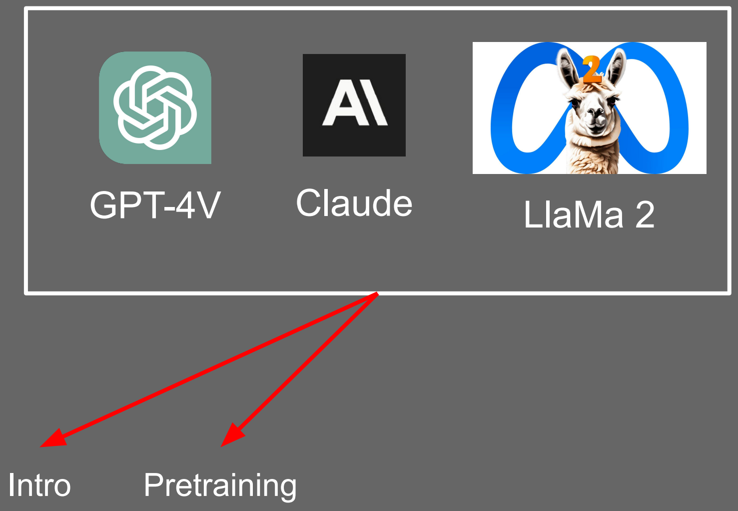
Outline

- Intro to LLMs



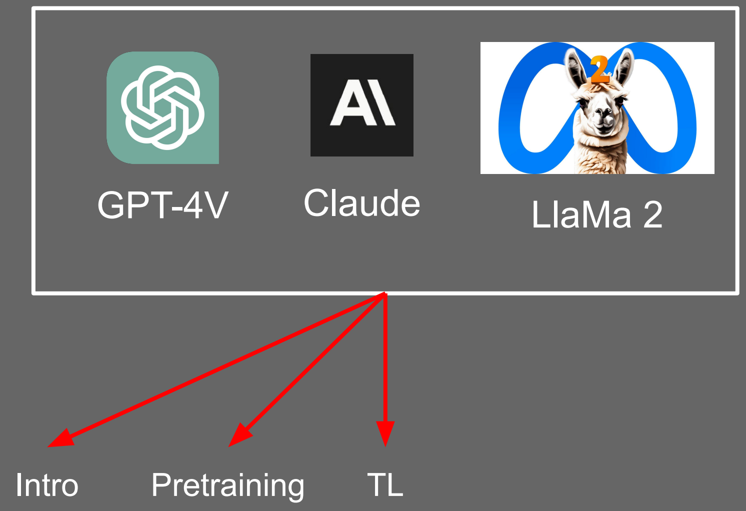
Outline

- Intro to LLMs
- Pretraining



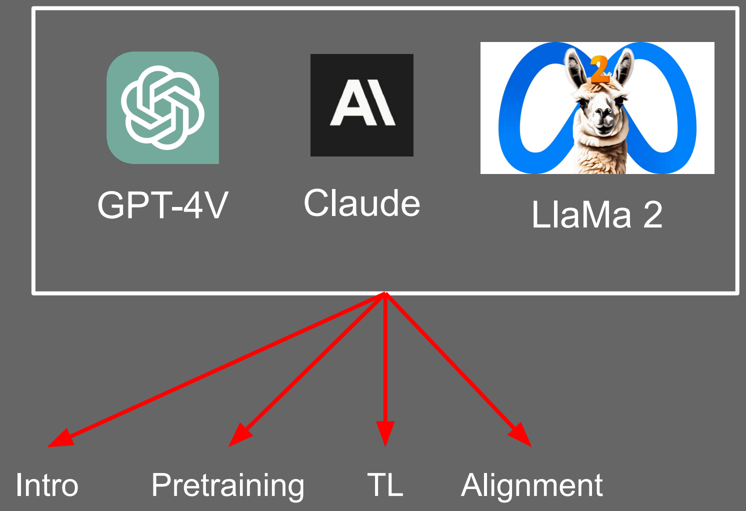
Outline

- Intro to LLMs
- Pretraining
- Transfer Learning (TL)



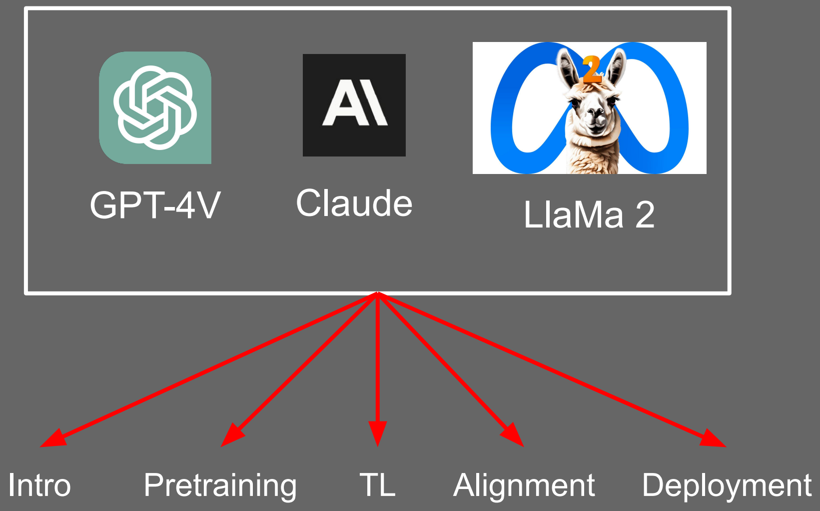
Outline

- Intro to LLMs
- Pretraining
- Transfer Learning (TL)
- Alignment



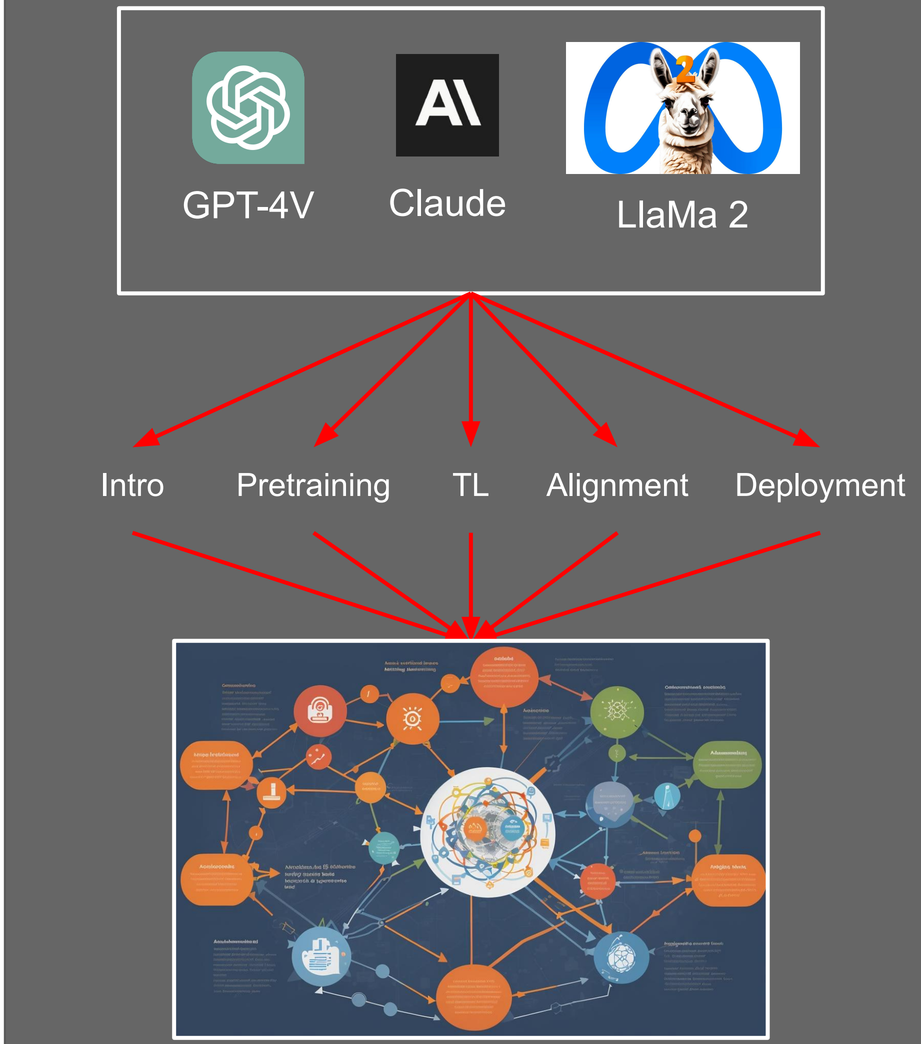
Outline

- Intro to LLMs
- Pretraining
- Transfer Learning (TL)
- Alignment
- Deployment



Outline

- Intro to LLMs
- Pretraining
- Transfer Learning (TL)
- Alignment
- Deployment



Intro to LLMs

Source: leonardo.ai

Prompt: “Title: Introduction to Large Language Models”

Autoregressive Models

"Autoregressive models predict a variable based on its previous values in a sequence."

Brockwell, P. J., & Davis, R. A. (2002)

Autoregressive Models

“Autoregressive models predict a variable based on its previous values in a sequence.”

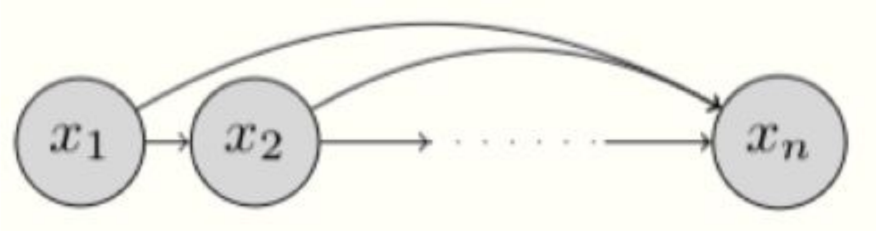
$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i | x_1, x_2, \ldots, x_{i-1}) = \prod_{i=1}^{n} p(x_i | \mathbf{x}_{<i})$$

Brockwell, P. J., & Davis, R. A. (2002)

Autoregressive Models

"Autoregressive models predict a variable based on its previous values in a sequence."

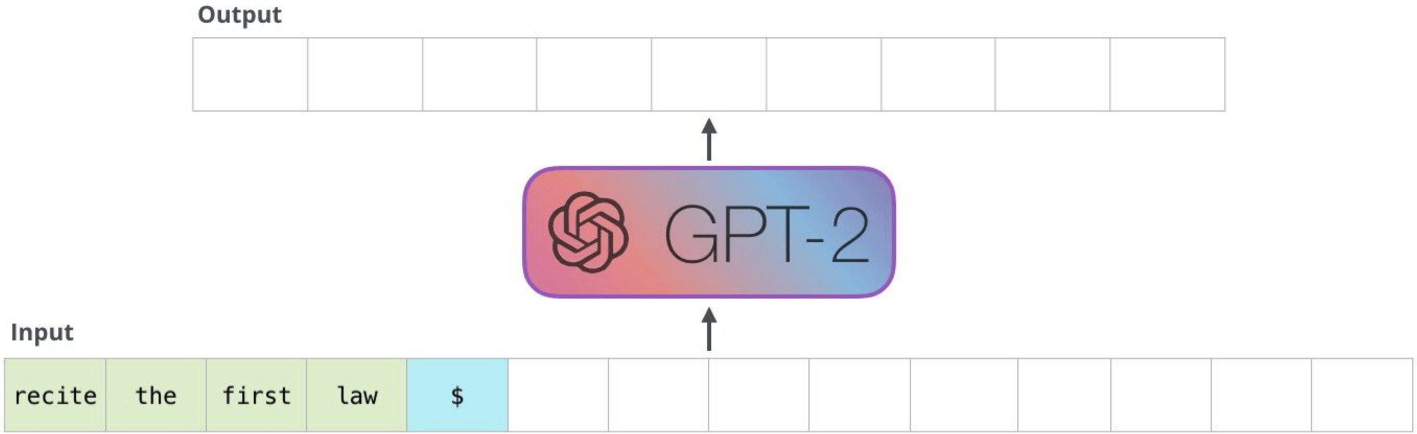
$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i | x_1, x_2, \ldots, x_{i-1}) = \prod_{i=1}^{n} p(x_i | \mathbf{x}_{<i})$$



Brockwell, P. J., & Davis, R. A. (2002)

Autoregressive Models

"Autoregressive models predict a variable based on its previous values in a sequence."



Source: [Step by step into GPT]()

Variations on Transformers

Variations on Transformers

RNNs

$$h_{t+1} = f(x_t, h_t, \theta_t)$$

Variations on Transformers

RNNs

$$h_{t+1} = f(x_t, h_t, \theta_t)$$

$$f(x_0, h_0, \theta_0) \rightarrow h_1$$
$$f(x_1, f(x_0, h_0, \theta_0), \theta_1) \rightarrow h_2$$
$$\vdots$$

Variations on Transformers

RNNs

$$h_{t+1} = f(x_t, h_t, \theta_t)$$

$$f(x_0, h_0, \theta_0) \rightarrow h_1$$

$$f(x_1, f(x_0, h_0, \theta_0), \theta_1) \rightarrow h_2$$

$$\vdots$$

Variations on Transformers

RNNs

$$h_{t+1} = f(x_t, h_t, \theta_t)$$

$$f(x_0, h_0, \theta_0) \rightarrow h_1$$

$$f(x_1, f(x_0, h_0, \theta_0), \theta_1) \rightarrow h_2$$

$$\vdots$$

Linear

$$ax_1 + bx_2 + cx_3 = y$$

Variations on Transformers

RNNs

$$h_{t+1} = f(x_t, h_t, \theta_t)$$

$$f(x_0, h_0, \theta_0) \rightarrow h_1$$
$$f(x_1, f(x_0, h_0, \theta_0), \theta_1) \rightarrow h_2$$
$$\vdots$$

Linear

$$ax_1 + bx_2 + cx_3 = y$$

Recurrent

$$ax_1 \rightarrow h$$
$$bx_2 + h \rightarrow h$$
$$cx_3 + h \rightarrow h = y$$

Variations on Transformers

RNNs

$$h_{t+1} = f(x_t, h_t, \theta_t)$$

$$f(x_0, h_0, \theta_0) \rightarrow h_1$$
$$f(x_1, \boxed{f(x_0, h_0, \theta_0)}, \theta_1) \rightarrow h_2$$
$$\vdots$$

Linear

$$ax_1 + bx_2 + cx_3 = y$$

Recurrent

$$ax_1 \rightarrow h$$
$$bx_2 + h \rightarrow h$$
$$cx_3 + h \rightarrow h = y$$

Parallel

$$[x_1, x_2, x_3] \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = y$$

Variations on Transformers

RNNs

$$h_{t+1} = f(x_t, h_t, \theta_t)$$

$$f(x_0, h_0, \theta_0) \rightarrow h_1$$
$$f(x_1, f(x_0, h_0, \theta_0), \theta_1) \rightarrow h_2$$
$$\vdots$$

Transformers

$$Q = \mathbf{x}W_Q$$
$$K = \mathbf{x}W_K$$
$$V = \mathbf{x}W_V$$

$$\mathbf{y} = \sigma\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Linear

$$ax_1 + bx_2 + cx_3 = y$$

Recurrent

$$ax_1 \rightarrow h$$
$$bx_2 + h \rightarrow h$$
$$cx_3 + h \rightarrow h = y$$

Parallel

$$[x_1, x_2, x_3] \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = y$$

Variations on Transformers

RNNs

$$h_{t+1} = f(x_t, h_t, \theta_t)$$

$$f(x_0, h_0, \theta_0) \rightarrow h_1$$
$$f(x_1, \boxed{f(x_0, h_0, \theta_0)}, \theta_1) \rightarrow h_2$$
$$\vdots$$

Linear

$$ax_1 + bx_2 + cx_3 = y$$

Recurrent

$$ax_1 \rightarrow h$$
$$bx_2 + h \rightarrow h$$
$$cx_3 + h \rightarrow h = y$$

Parallel

$$[x_1, x_2, x_3] \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = y$$

Transformers

$$Q = \mathbf{x}W_Q$$
$$K = \mathbf{x}W_K$$
$$V = \mathbf{x}W_V$$

$$\mathbf{y} = \boxed{\sigma\left(\frac{QK^T}{\sqrt{d_k}}\right)} V$$

Variations on Transformers

RNNs

$$h_{t+1} = f(x_t, h_t, \theta_t)$$

$$f(x_0, h_0, \theta_0) \rightarrow h_1$$
$$f(x_1, \boxed{f(x_0, h_0, \theta_0)}, \theta_1) \rightarrow h_2$$
$$\vdots$$

Transformers

$$Q = \mathbf{x}W_Q$$
$$K = \mathbf{x}W_K$$
$$V = \mathbf{x}W_V$$
$$\mathbf{y} = \boxed{\sigma\left(\frac{QK^T}{\sqrt{d_k}}\right)}V$$

Linear

$$ax_1 + bx_2 + cx_3 = y$$

Recurrent

$$ax_1 \rightarrow h$$
$$bx_2 + h \rightarrow h$$
$$cx_3 + h \rightarrow h = y$$

Parallel

$$[x_1, x_2, x_3] \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = y$$

Linear Transformers

$$Q = \mathbf{x}W_Q$$
$$K = \mathbf{x}W_K$$
$$V = \mathbf{x}W_V$$
$$\mathbf{y} = \frac{\phi(Q)\phi(K)^T V}{\sqrt{d_k}}$$

Variations on Transformers

RNNs

$$h_{t+1} = f(x_t, h_t, \theta_t)$$

$$f(x_0, h_0, \theta_0) \rightarrow h_1$$
$$f(x_1, f(x_0, h_0, \theta_0), \theta_1) \rightarrow h_2$$
$$\vdots$$

Transformers

$$Q = \mathbf{x}W_Q$$
$$K = \mathbf{x}W_K$$
$$V = \mathbf{x}W_V$$

$$\mathbf{y} = \sigma\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Linear

$$ax_1 + bx_2 + cx_3 = y$$

Recurrent

$$ax_1 \rightarrow h$$
$$bx_2 + h \rightarrow h$$
$$cx_3 + h \rightarrow h = y$$

Parallel

$$[x_1, x_2, x_3] \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = y$$

Linear Transformers

$$Q = \mathbf{x}W_Q$$
$$K = \mathbf{x}W_K$$
$$V = \mathbf{x}W_V$$

$$\mathbf{y} = \frac{\phi(Q)\phi(K)^T V}{\sqrt{d_k}}$$

Modern Architectures Implementing AR Models

Architectures	Training Parallelization	Inference Cost	Long-Sequence Memory Complexity	Performance
Transformer	✔	$O(N)$	$O(N^2)$	✔✔
Linear Transformer	✔	$O(1)$	$O(N)$	✘
Recurrent NN	✘	$O(1)$	$O(N)$	✘
RWKV	✘	$O(1)$	$O(N)$	✔
H3/S4	✔	$O(1)$	$O(N \log N)$	✔
Hyena	✔	$O(N)$	$O(N \log N)$	✔
RetNet	✔	$O(1)$	$O(N)$	✔✔

Table 1: Model comparison from various perspectives. RetNet achieves training parallelization, constant inference cost, linear long-sequence memory complexity, and good performance.

Sun, Y. (2023). [Retentive Network]

The Distributional Hypothesis

Harris, Z. (1954). Distributional structure. Word, 10(2-3), 146-162

The Distributional Hypothesis

Definition: Words that occur in the same contexts tend to have similar meanings.

Harris, Z. (1954). Distributional structure. Word, 10(2-3), 146-162

The Distributional Hypothesis

Definition: Words that occur in the same contexts tend to have similar meanings.

Example

What is umbër?

Harris, Z. (1954). Distributional structure. Word, 10(2-3), 146-162

The Distributional Hypothesis

Definition: Words that occur in the same contexts tend to have similar meanings.

Example

What is umbër?

- She follows her umbër for painting with dedication.

Harris, Z. (1954). Distributional structure. Word, 10(2-3), 146-162

The Distributional Hypothesis

Definition: Words that occur in the same contexts tend to have similar meanings.

Example

What is umbër?

- She follows her umbër for painting with dedication.
- His umbër for conservation motivates him to protect wildlife.

Harris, Z. (1954). Distributional structure. Word, 10(2-3), 146-162

The Distributional Hypothesis

Definition: Words that occur in the same contexts tend to have similar meanings.

Example

What is umbër?

- She follows her umbër for painting with dedication.
- His umbër for conservation motivates him to protect wildlife.
- They loved the umbër juice that Uncle Joey brought.

Harris, Z. (1954). Distributional structure. Word, 10(2-3), 146-162

The Distributional Hypothesis

Definition: Words that occur in the same contexts tend to have similar meanings.

Example

What is umbër?

- She follows her umbër for painting with dedication.
- His umbër for conservation motivates him to protect wildlife.
- They loved the umbër juice that Uncle Joey brought.
- Umbër drives you to overcame obstacles, and persevere..

Harris, Z. (1954). Distributional structure. Word, 10(2-3), 146-162

The Distributional Hypothesis

Definition: Words that occur in the same contexts tend to have similar meanings.

Example

What is umbër?

- She follows her umbër for painting with dedication.
- His umbër for conservation motivates him to protect wildlife.
- They loved the umbër juice that Uncle Joey brought.
- Umbër drives you to overcame obstacles, and persevere.
- The courtroom's umbër gripped everyone present.

The context in which a word appears tells a lot about what it means

Harris, Z. (1954). Distributional structure. Word, 10(2-3), 146-162

The Distributional Hypothesis

Definition: Words that occur in the same contexts tend to have similar meanings.

Example

What is umbër?

- She follows her umbër for painting with dedication.
- His umbër for conservation motivates him to protect wildlife.
- They loved the umbër juice that Uncle Joey brought.
- Umbër drives you to overcame obstacles, and persevere.
- The courtroom's umbër gripped everyone present.

The context in which a word appears tells a lot about what it means

Significance: Underpins the effectiveness of token embeddings, and subsequently, transformers.

Harris, Z. (1954). Distributional structure. Word, 10(2-3), 146-162

The Distributional Hypothesis

Definition: Words that occur in the same contexts tend to have similar meanings.

Example

What is umbër?

- She follows her umbër for painting with dedication.
- His umbër for conservation motivates him to protect wildlife.
- They loved the umbër juice that Uncle Joey brought.
- Umbër drives you to overcame obstacles, and persevere.
- The courtroom's umbër gripped everyone present.

"Words that flock together talk together"

The context in which a word appears tells a lot about what it means

Significance: Underpins the effectiveness of token embeddings, and subsequently, transformers.

Harris, Z. (1954). Distributional structure. Word, 10(2-3), 146-162

Pretraining

Source: leonardo.ai

Prompt: “Title: Pretraining Large Language Models on Trillions of Tokens.”

Pretraining LLMs

Icabod likes headless horsemen

Pretraining LLMs

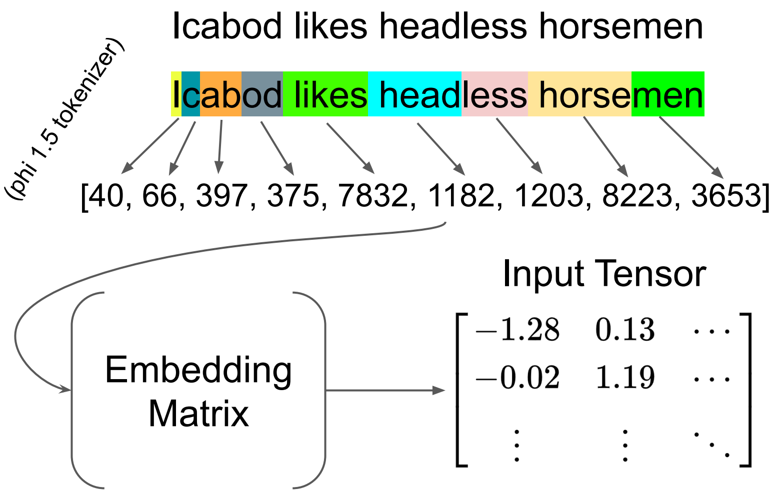
Icabod likes headless horsemen

(phi 1.5 tokenizer)

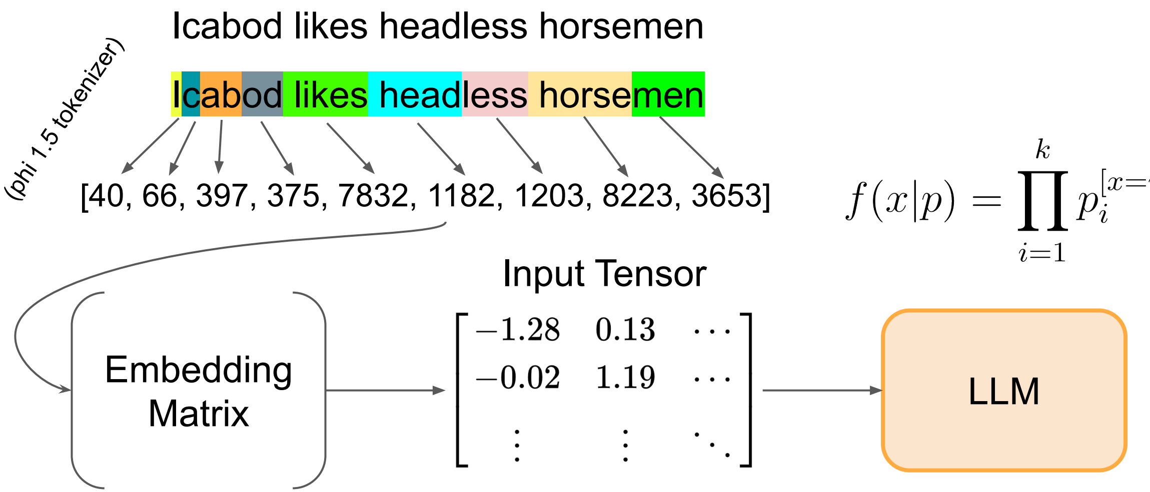
I | c | ab | od | likes | head | less | horse | men

[40, 66, 397, 375, 7832, 1182, 1203, 8223, 3653]

Pretraining LLMs

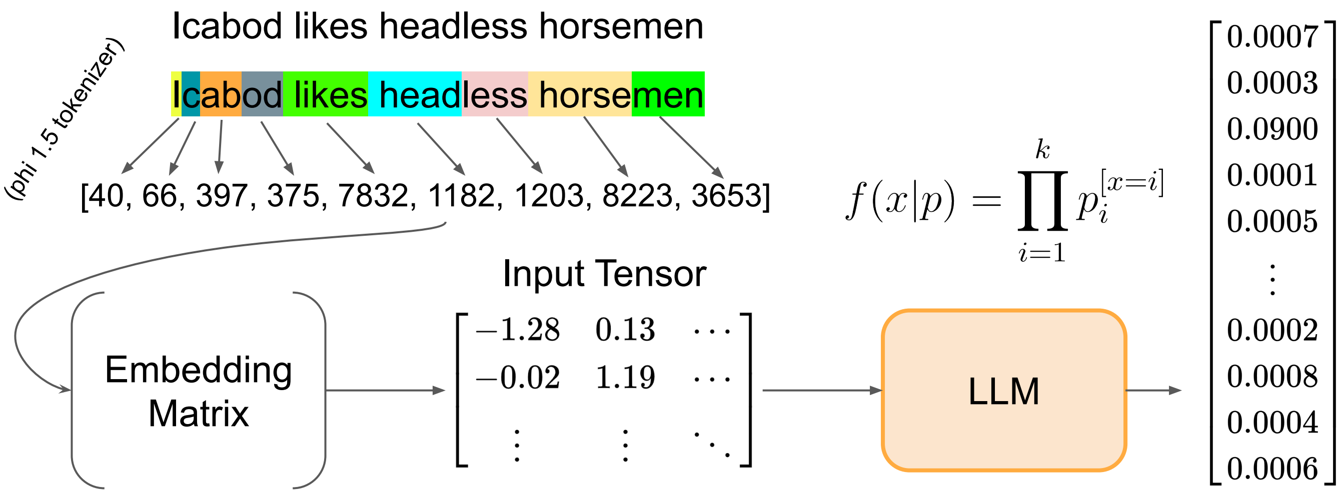


Pretraining LLMs

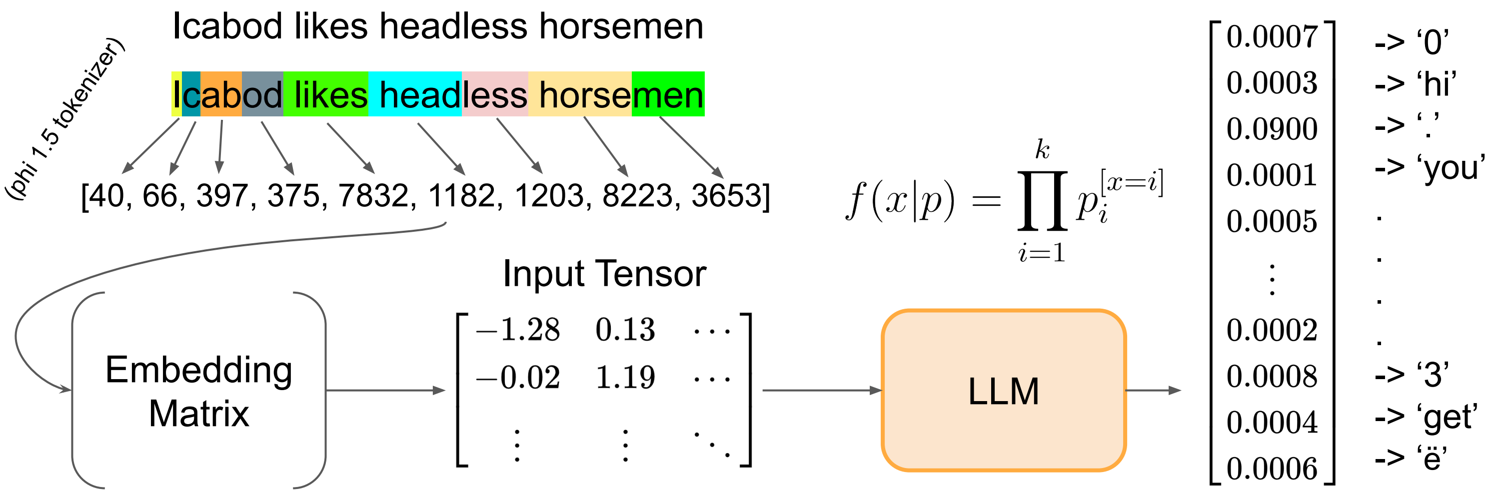


$$f(x|p) = \prod_{i=1}^{k} p_i^{[x=i]}$$

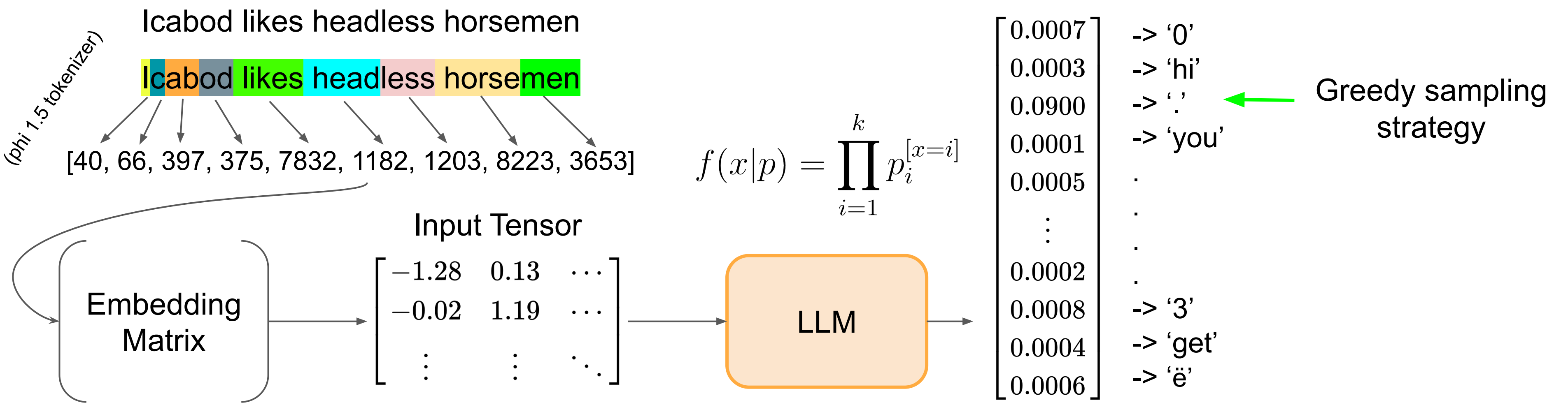
Pretraining LLMs



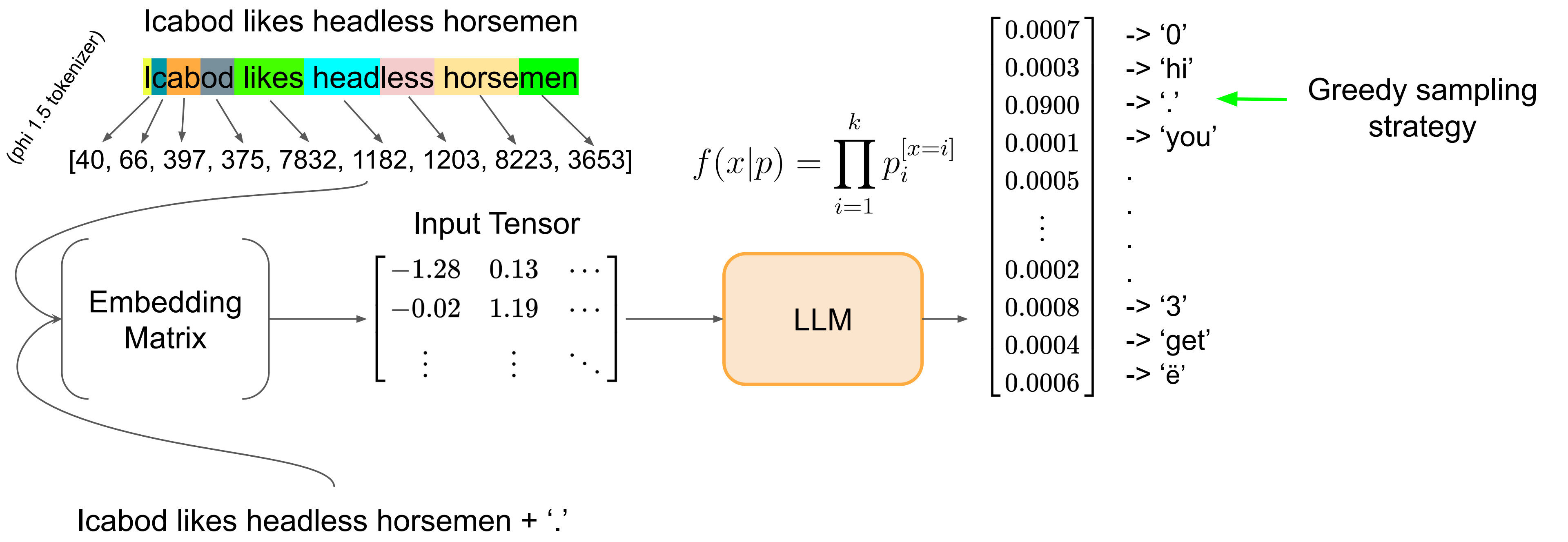
Pretraining LLMs



Pretraining LLMs



Pretraining LLMs



Pretraining LLMs (Datasets)

Corpora	Size	Source	Latest Update Time
BookCorpus [138]	5GB	Books	Dec-2015
Gutenberg [139]	-	Books	Dec-2021
C4 [73]	800GB	CommonCrawl	Apr-2019
CC-Stories-R [140]	31GB	CommonCrawl	Sep-2019
CC-NEWS [27]	78GB	CommonCrawl	Feb-2019
REALNEWs [141]	120GB	CommonCrawl	Apr-2019
OpenWebText [142]	38GB	Reddit links	Mar-2023
Pushift.io [143]	2TB	Reddit links	Mar-2023
Wikipedia [144]	21GB	Wikipedia	Mar-2023
BigQuery [145]	-	Codes	Mar-2023
the Pile [146]	800GB	Other	Dec-2020
ROOTS [147]	1.6TB	Other	Jun-2022

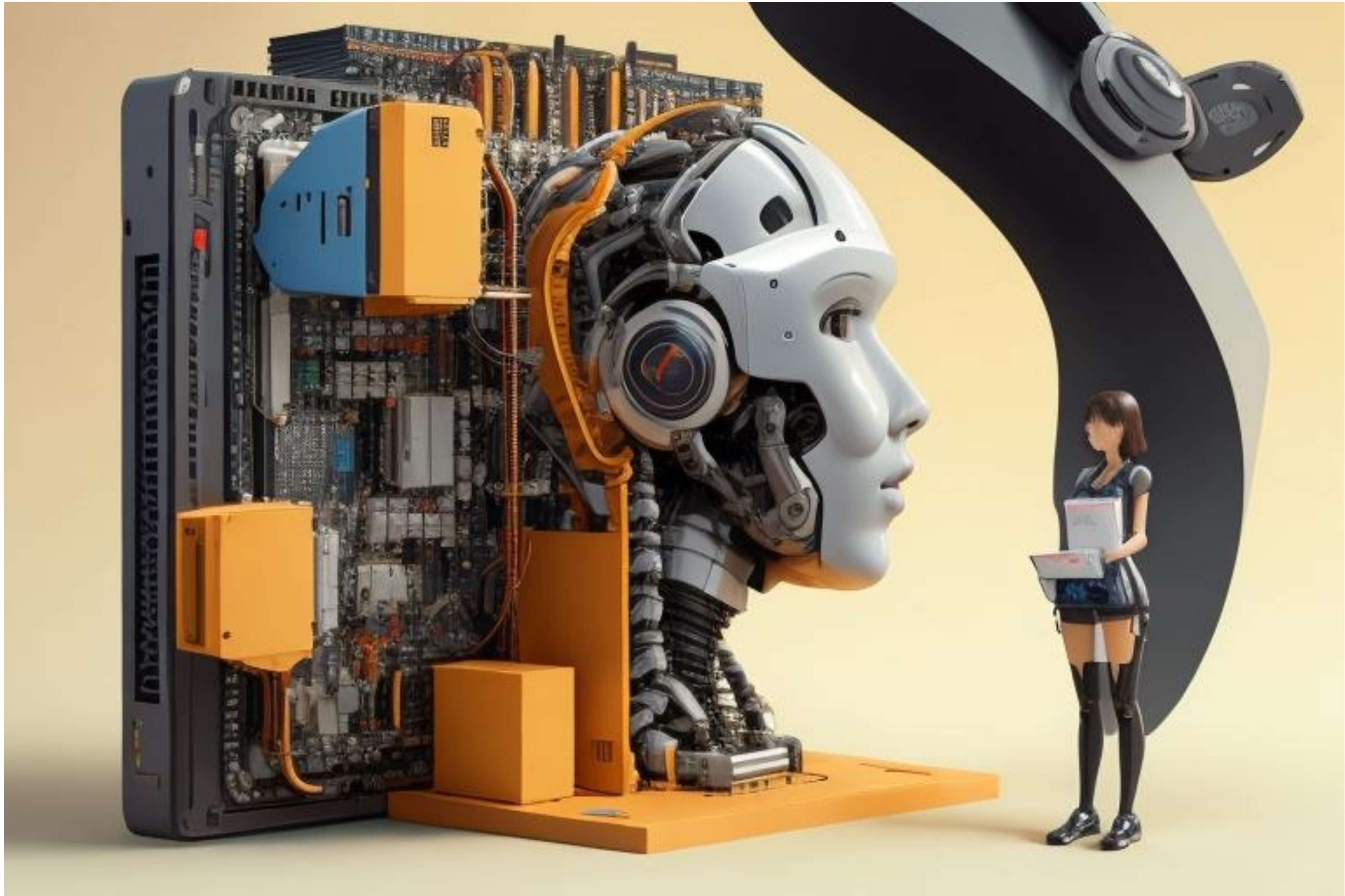
"A Survey of Large Language Models" (Zhao et al., 2023)

Pretraining LLMs (Models)

“A Survey of Large Language Models” (Zhao et al., 2023)

Model	Release Time	Size (B)	Base Model	Adaptation IT	Adaptation RLHF	Pre-train Data Scale	Latest Data Timestamp	Hardware (GPUs / TPUs)	Training Time	Evaluation ICL	Evaluation CoT
T5 [73]	Oct-2019	11	-	-	-	1T tokens	Apr-2019	1024 TPU v3	-	✓	-
mT5 [74]	Oct-2020	13	-	-	-	1T tokens	-	-	-	✓	-
PanGu-α [75]	Apr-2021	13*	-	-	-	1.1TB	-	2048 Ascend 910	-	✓	-
CPM-2 [76]	Jun-2021	198	-	-	-	2.6TB	-	-	-	-	-
T0 [28]	Oct-2021	11	T5	✓	-	-	-	512 TPU v3	27 h	✓	-
CodeGen [77]	Mar-2022	16	-	-	-	577B tokens	-	-	-	✓	-
GPT-NeoX-20B [78]	Apr-2022	20	-	-	-	825GB	-	96 40G A100	-	✓	-
Tk-Instruct [79]	Apr-2022	11	T5	✓	-	-	-	256 TPU v3	4 h	✓	-
UL2 [80]	May-2022	20	-	-	-	1T tokens	Apr-2019	512 TPU v4	-	✓	✓
OPT [81]	May-2022	175	-	-	-	180B tokens	-	992 80G A100	-	✓	-
NLLB [82]	Jul-2022	54.5	-	-	-	-	-	-	-	✓	-
CodeGeeX [83]	Sep-2022	13	-	-	-	850B tokens	-	1536 Ascend 910	60 d	✓	-
GLM [84]	Oct-2022	130	-	-	-	400B tokens	-	768 40G A100	60 d	✓	-
Flan-T5 [64]	Oct-2022	11	T5	✓	-	-	-	-	-	✓	✓
BLOOM [69]	Nov-2022	176	-	-	-	366B tokens	-	384 80G A100	105 d	✓	-
mT0 [85]	Nov-2022	13	mT5	✓	-	-	-	-	-	✓	-
Galactica [35]	Nov-2022	120	-	-	-	106B tokens	-	-	-	✓	✓
BLOOMZ [85]	Nov-2022	176	BLOOM	✓	-	-	-	-	-	✓	-
OPT-IML [86]	Dec-2022	175	OPT	✓	-	-	-	128 40G A100	-	✓	✓
LLaMA [57]	Feb-2023	65	-	-	-	1.4T tokens	-	2048 80G A100	21 d	✓	-
Pythia [87]	Apr-2023	12	-	-	-	300B tokens	-	256 40G A100	-	✓	-
CodeGen2 [88]	May-2023	16	-	-	-	400B tokens	-	-	-	✓	-
StarCoder [89]	May-2023	15.5	-	-	-	1T tokens	-	512 40G A100	-	✓	✓
LLaMA2 [90]	Jul-2023	70	-	✓	✓	2T tokens	-	2000 80G A100	-	✓	-

Transfer Learning



Source: leonardo.ai

Prompt: “Title: Teaching Large Language Models to Follow Instructions.”

Transfer Learning

Transfer Learning

1. **Fine-tuning** (First demonstrated in **2018** ULMFiT)
   After training a model on a large corpus of data, it can be further specialized via fune-tuning

Transfer Learning: Fine-Tuning

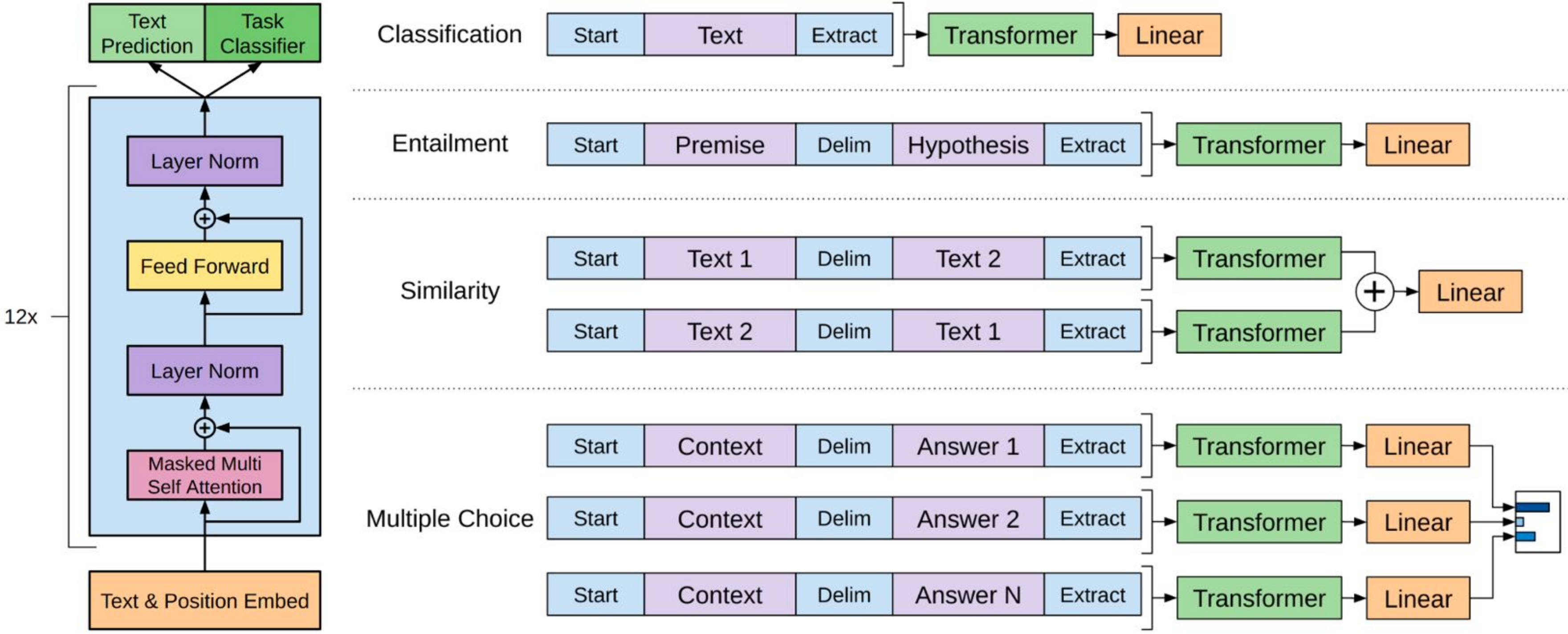


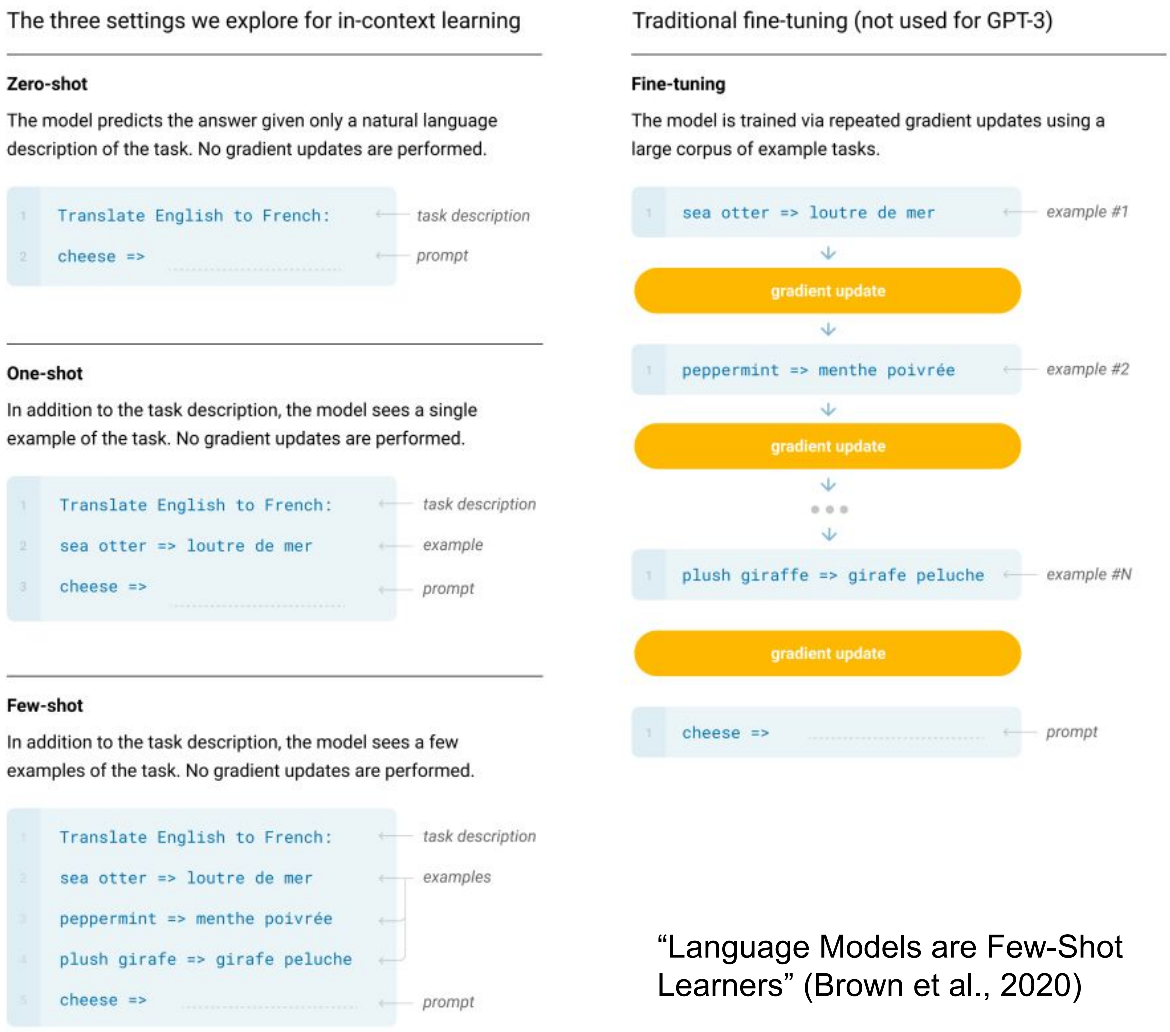
Figure from "Improving Language Understanding by Generative Pre-Training" (Radford et al., 2018)

Transfer Learning

1. **Fine-tuning** (First demonstrated in **2018** ULMFiT)
   After training a model on a large corpus of data, it can be further specialized via fine-tuning

2. **In-context Learning** (First demonstrated in **2020** GPT3)
   If examples of a task are included in the prompt to an LLM it completes completes the task ***better*** on unseen data

Transfer Learning: In-context Learning



"Language Models are Few-Shot Learners" (Brown et al., 2020)

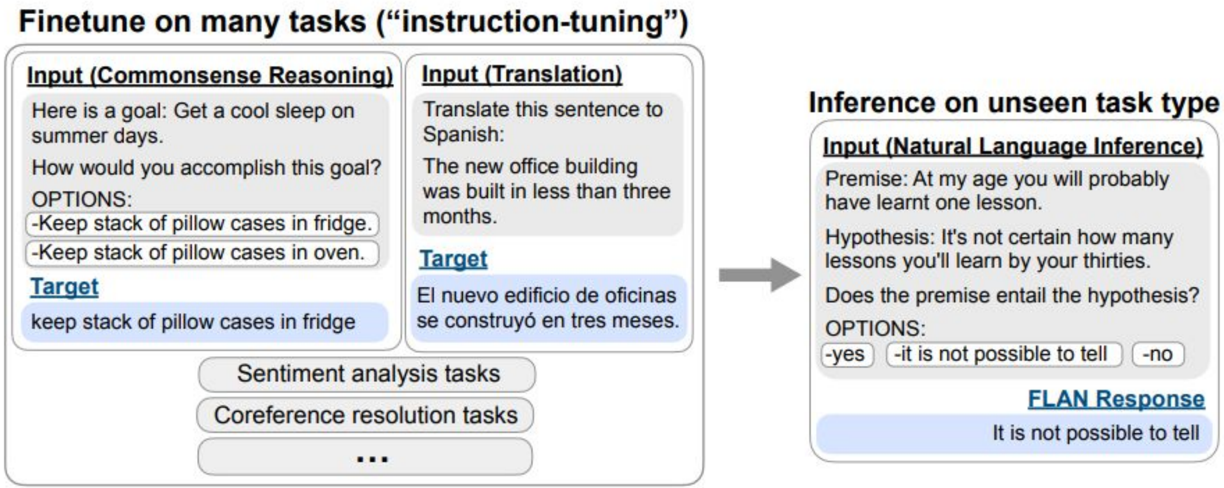
Transfer Learning

1. **Fine-tuning** (First demonstrated in **2018** ULMFiT)
   After training a model on a large corpus of data, it can be further specialized via fune-tuning

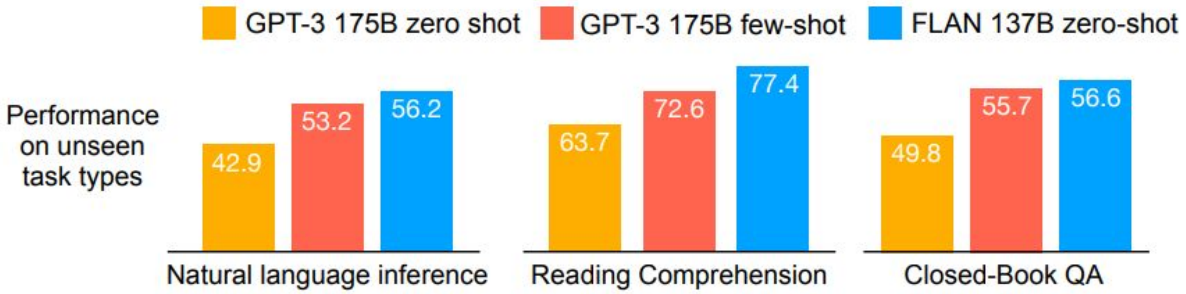
2. **In-context Learning** (First demonstrated in **2020** GPT3)
   If examples of a task are included in the prompt to an LLM it completes completes the task ***better*** on unseen data

3. **Instruction Tuning** (First demonstrated in **2022** FLAN)
   Fine-tuning the model on instructions (similar to the in-context examples) enables to model to generalize to numerous tasks in a zero-shot manner

Transfer Learning: Instruction Tuning



Finetuned Language Models are Zero-Shot Learners (Wei et al., 2022)



Transfer Learning: Evolution

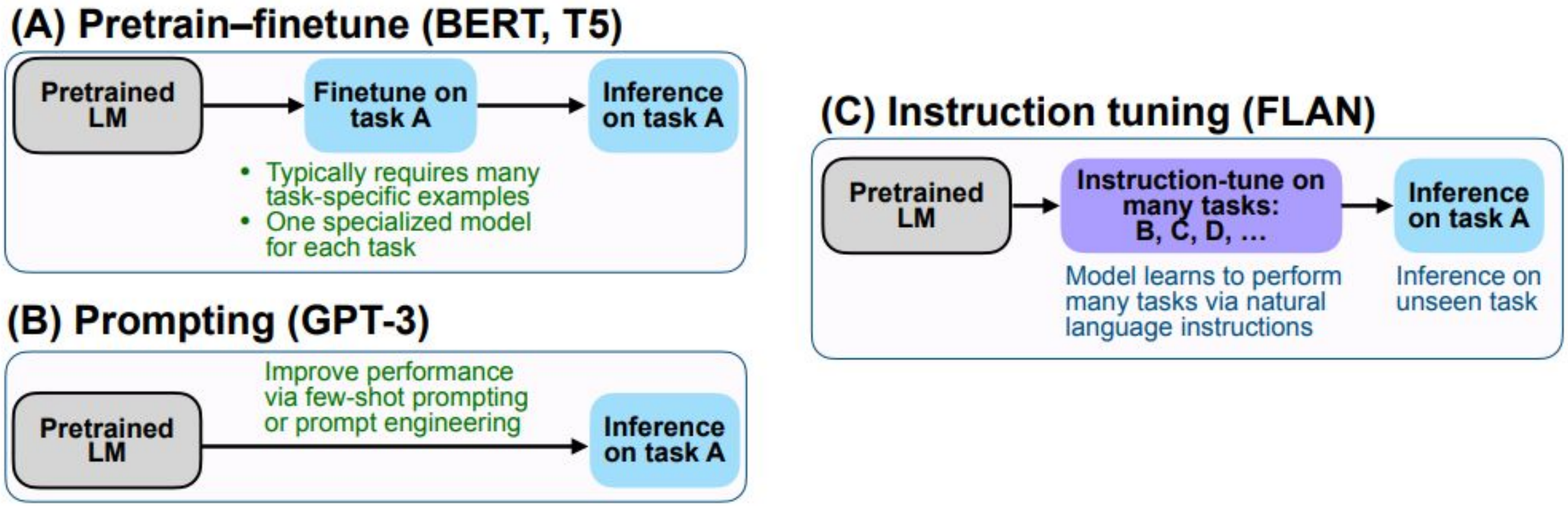


Figure 2: Comparing instruction tuning with pretrain–finetune and prompting.

Finetuned Language Models are Zero-Shot Learners (Wei et al., 2022)

Transfer Learning: InstructGPT

Prompt *Explain the moon landing to a 6 year old in a few sentences.*

Completion GPT-3

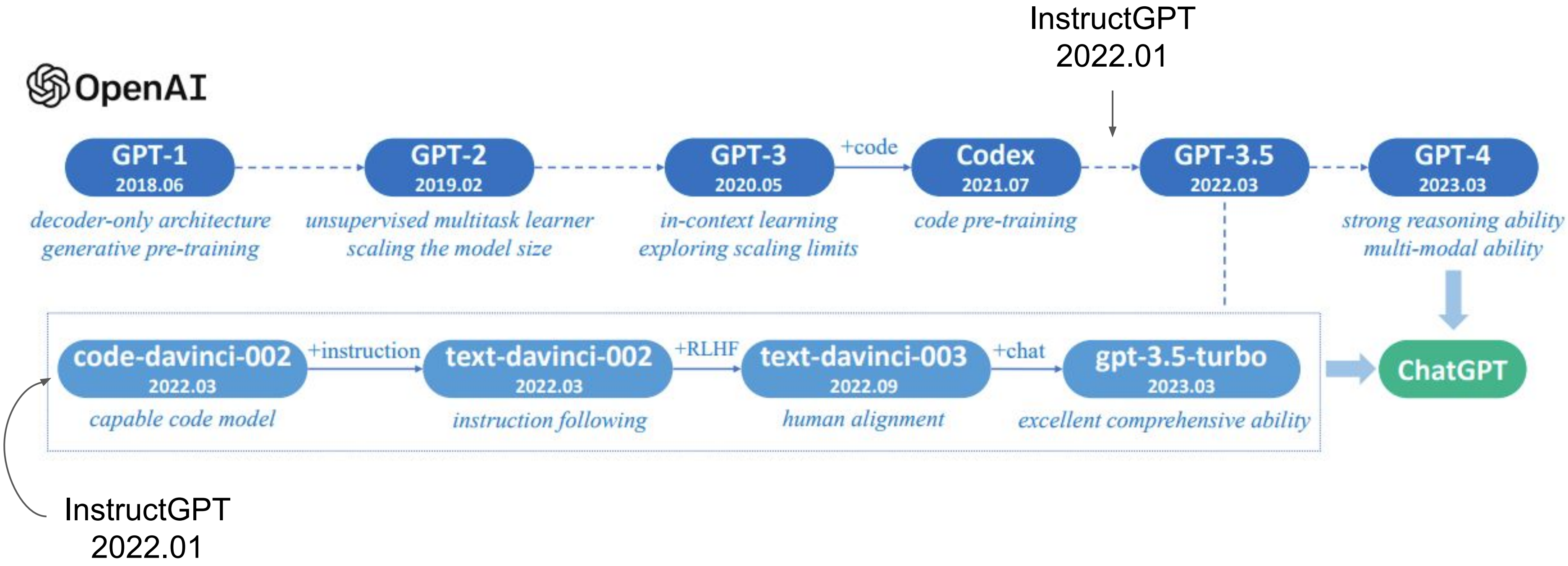
```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

InstructGPT

```
People went to the moon, and they took pictures of what they saw, and sent them
back to the earth so we could all see them.
```

Training language models to follow instructions with human feedback (Ouyang et al., 2022)

Transfer Learning: From OpenAI Models



“A Survey of Large Language Models” (Zhao et al., 2023)

Instruction Fine-Tuning (Dataset Curation Process)

There are three dominant approaches:

Instruction Fine-Tuning (Dataset Curation Process)

There are three dominant approaches:

1. **Data integration from annotated natural language datasets**

   Examples: Flan (Longpre et al., 2023), P3 (Sanh et al., 2021)

Instruction Fine-Tuning (Dataset Curation Process)

There are three dominant approaches:

1. **Data integration from annotated natural language datasets**

   Examples: Flan (Longpre et al., 2023), P3 (Sanh et al., 2021)

2. **Generating outputs using LLMs**

   Examples: InstructWild (Xue et al., 2023), Self-Instruct (Wang et al., 2022)

Instruction Fine-Tuning (Dataset Curation Process)

There are three dominant approaches:

1. **Data integration from annotated natural language datasets**

   Examples: Flan (Longpre et al., 2023), P3 (Sanh et al., 2021)

2. **Generating outputs using LLMs**

   Examples: InstructWild (Xue et al., 2023), Self-Instruct (Wang et al., 2022)

3. **Hybrid LLM + human combination**

   Examples: OIG (LAION.ai, 2023)

Instruction Fine-Tuning: Natural Instructions

193K instances, coming from 61 distinct NLP tasks

Example task instances

Instance

- **Input:** Sentence: It's hail crackled across the comm, and Tara spun to retake her seat at the helm.
- **Expected Output:** How long was the storm?

⋮

Instance

- **Input:** Sentence: During breakfast one morning, he seemed lost in thought and ignored his food.
- **Expected Output:** How long was he lost in thoughts?

Instructions for MC-TACO question generation task

- **Title:** Writing questions that involve commonsense understanding of "event duration".
- **Definition:** In this task, we ask you to write a question that involves "event duration", based on a given sentence. Here, event duration is defined as the understanding of how long events typically last. For example, "brushing teeth", usually takes few minutes.
- **Emphasis & Caution:** The written questions are not required to have a single correct answer.
- **Things to avoid:** Don't create questions which have explicit mentions of answers in text. Instead, it has to be implied from what is given. In other words, we want you to use "instinct" or "common sense".

Positive Example

- **Input:** Sentence: Jack played basketball after school, after which he was very tired.
- **Output:** How long did Jack play basketball?
- **Reason:** the question asks about the duration of an event; therefore it's a temporal event duration question.

Negative Example

- **Input:** Sentence: He spent two hours on his homework.
- **Output:** How long did he do his homework?
- **Reason:** We DO NOT want this question as the answer is directly mentioned in the text.
- **Suggestion:** -

- **Prompt:** Ask a question on "event duration" based on the provided sentence.

Cross-Task Generalization via Natural Language Crowdsourcing Instructions (Mishra et al., 2021)

Instruction Fine-Tuning

Datasets

Type	Dataset Name	# of Instances	# of Tasks	# of Lang	Construction	Open-source
Generalize to unseen tasks	UnifiedQA (Khashabi et al., 2020)[1]	750K	46	En	human-crafted	Yes
	OIG (LAION.ai, 2023)[2]	43M	30	En	human-model-mixed	Yes
	UnifiedSKG (Xie et al., 2022)[3]	0.8M	-	En	human-crafted	Yes
	Natural Instructions (Honovich et al., 2022)[4]	193K	61	En	human-crafted	Yes
	Super-Natural Instructions (?)[5]	5M	76	55 Lang	human-crafted	Yes
	P3 (Sanh et al., 2021)[6]	12M	62	En	human-crafted	Yes
	xP3 (Muennighoff et al., 2022)[7]	81M	53	46 Lang	human-crafted	Yes
	Flan 2021 (Longpre et al., 2023)[8]	4.4M	62	En	human-crafted	Yes
	COIG (Zhang et al., 2023a)[9]	-	-	-	-	Yes
Follow users' instructions in a single turn	InstructGPT (Ouyang et al., 2022)	13K	-	Multi	human-crafted	No
	Unnatural Instructions (Honovich et al., 2022)[10]	240K	-	En	InstructGPT-generated	Yes
	Self-Instruct (Wang et al., 2022c)[11]	52K	-	En	InstructGPT-generated	Yes
	InstructWild (Xue et al., 2023)[12]	104K	429	-	model-generated	Yes
	Evol-Instruct (Xu et al., 2023a)[13]	52K	-	En	ChatGPT-generated	Yes
	Alpaca (Taori et al., 2023)[14]	52K	-	En	InstructGPT-generated	Yes
	LogiCoT (Liu et al., 2023a)[15]	-	2	En	GPT-4-generated	Yes
	Dolly (Conover et al., 2023a)[16]	15K	7	En	human-crafted	Yes
	GPT-4-LLM (Peng et al., 2023)[17]	52K	-	En&Zh	GPT-4-generated	Yes
	LIMA (Zhou et al., 2023)[18]	1K	-	En	human-crafted	Yes
Offer assistance like humans across multiple turns	ChatGPT (OpenAI, 2022)	-	-	Multi	human-crafted	No
	Vicuna (Chiang et al., 2023)	70K	-	En	user-shared	No
	Guanaco (JosephusCheung, 2021)[19]	534,530	-	Multi	model-generated	Yes
	OpenAssistant (Köpf et al., 2023)[20]	161,443	-	Multi	human-crafted	Yes
	Baize v1 (?)[21]	111.5K	-	En	ChatGPT-generated	Yes
	UltraChat (Ding et al., 2023a)[22]	675K	-	En&Zh	model-generated	Yes

Instruction Fine-Tuning

Models

Instruction fine-tuned LLMs	# Params	Base Model	Fine-tuning Trainset		
			Self-build	Dataset Name	Size
Instruct-GPT (Ouyang et al., 2022)	176B	GPT-3 (Brown et al., 2020b)	Yes	-	-
BLOOMZ (Muennighoff et al., 2022)[1]	176B	BLOOM (Scao et al., 2022)	No	xP3	-
FLAN-T5 (Chung et al., 2022)[2]	11B	T5 (Raffel et al., 2019)	No	FLAN 2021	-
Alpaca (Taori et al., 2023)[3]	7B	LLaMA (Touvron et al., 2023a)	Yes	-	52K
Vicuna (Chiang et al., 2023)[4]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	70K
GPT-4-LLM (Peng et al., 2023)[5]	7B	LLaMA (Touvron et al., 2023a)	Yes	-	52K
Claude (Bai et al., 2022b)	-	-	Yes	-	-
WizardLM (Xu et al., 2023a)[6]	7B	LLaMA (Touvron et al., 2023a)	Yes	Evol-Instruct	70K
ChatGLM2 (Du et al., 2022)[7]	6B	GLM (Du et al., 2022)	Yes	-	1.1 Tokens
LIMA (Zhou et al., 2023)	65B	LLaMA (Touvron et al., 2023a)	Yes	-	1K
OPT-IML (Iyer et al., 2022)[8]	175B	OPT (Zhang et al., 2022a)	No	-	-
Dolly 2.0 (Conover et al., 2023a)[9]	12B	Pythia (Biderman et al., 2023)	No	-	15K
Falcon-Instruct (Almazrouei et al., 2023a)[10]	40B	Falcon (Almazrouei et al., 2023b)	No	-	-
Guanaco (JosephusCheung, 2021)[11]	7B	LLaMA (Touvron et al., 2023a)	Yes	-	586K
Minotaur (Collective, 2023)[12]	15B	Starcoder Plus (Li et al., 2023f)	No	-	-
Nous-Hermes (NousResearch, 2023)[13]	13B	LLaMA (Touvron et al., 2023a)	No	-	300K+
TÜLU (Wang et al., 2023c)[14]	6.7B	OPT (Zhang et al., 2022a)	No	Mixed	-
YuLan-Chat (YuLan-Chat-Team, 2023)[15]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	250K
MOSS (Tianxiang and Xipeng, 2023)[16]	16B	-	Yes	-	-
Airoboros (Durbin, 2023)[17]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	-
UltraLM (Ding et al., 2023a)[18]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	-

Instruction Fine-Tuning Challenges (superficial alignment hypothesis)

The Superficial Alignment Hypothesis posits that:

a model's knowledge and capabilities are learned almost entirely during pretraining

while alignment teaches it which subdistribution of formats should be used when interacting with users. This suggests that

focusing on data quality and diversity

rather than just quantity

leads to better alignment and performance

LIMA: Less Is More for Alignment (Zhou et al., 2023)

Instruction Fine-Tuning Challenges (superficial alignment hypothesis)

- LIMA (65B): Fine-tuned LLaMA (65B) (Touvron et al., 2023a) model based on superficial alignment hypothesis
- Dataset: 1,000 examples (750 from community forums, 250 manually written)
- Comparison: Outperforms RLHF-trained DaVinci003 and 65B Alpaca
- Human preference: LIMA equal or preferable in 43% (GPT-4), 46% (Claude), 58% (Bard) cases
- Response quality: 88% meet prompt requirements, 50% considered excellent

LIMA: Less Is More for Alignment (Zhou et al., 2023)

Instruction Fine-Tuning Challenges

- Increasing concern that IT only improves on tasks that are heavily supported in the IT training dataset (Gudibande et al., 2023)
- Criticism that IT only captures surface-level patterns and styles (e.g., the output format) rather than comprehending and learning the task (Kung and Peng, 2023)

Alignment

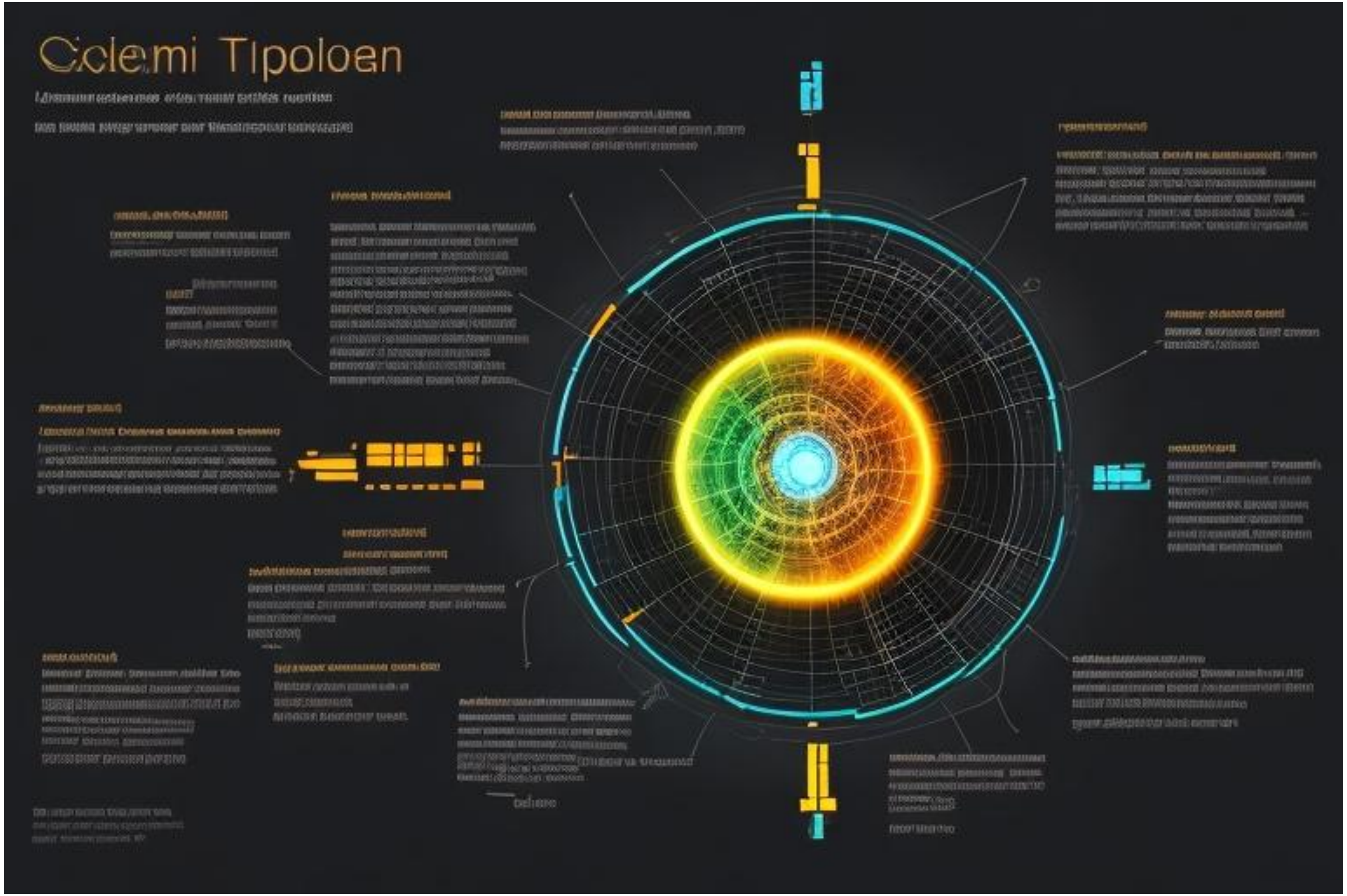
Source: leonardo.ai

Prompt: “Title: Teaching artificial intelligence to comply with human values”

Appendix

Source: leonardo.ai

Prompt: “Title: Teaching artificial intelligence to comply with human values”



Stages of Learning

Source: [State of GPT, Andrej Karpathy (2023)]

Stage	Pretraining	Supervised Finetuning	Reward Modeling	Reinforcement Learning
Dataset	**Raw internet** text trillions of words low-quality, large quantity	**Demonstrations** Ideal Assistant responses, ~10-100K (prompt, response) written by contractors low quantity, high quality	**Comparisons** 100K –1M comparisons written by contractors low quantity, high quality	**Prompts** ~10K-100K prompts written by contractors low quantity, high quality
Algorithm	**Language modeling** predict the next token	**Language modeling** predict the next token	**Binary classification** predict rewards consistent w preferences	**Reinforcement Learning** generate tokens that maximize the reward
		init from	init from	init from SFT use RM
Model	**Base model**	**SFT model**	**RM model**	**RL model**
Notes	1000s of GPUs months of training ex: GPT, LLaMA, PaLM can deploy this model	1-100 GPUs days of training ex: Vicuna-13B can deploy this model	1-100 GPUs days of training	1-100 GPUs days of training ex: ChatGPT, Claude can deploy this model

Instruction Fine-Tuning (Possible Sources of Hallucination)

1. e: Writing questions that involve commonsense understanding of "event duration". - Definition: In this task, we ask you to write a question that involves ?event duration", based on a given sentence. Here, event duration is defined as the understanding of how long events typically last. For example, ?brushing teeth?, usually takes few minutes. - Emphasis & Caution: The written questions are not required to have a single correct answer. - Things to avoid: Don't create questions which have explicit mentions of answers in text. Instead, it has to be implied from what is given. In other words, we want you to use "instinct" or "common sense". -Input: Sentence: Jack played basketball after school, after which he was very tired. -Output: How long did Jack play basketball? -Reason: the question asks about the duration of an event; therefore it's a temporal event duration question. Positive Example -Input: Sentence: He spent two hours on his homework. -Output: How long did he do his homework? -Reason: We DONOT want this question as the answer is directly mentioned in the text. -Suggestion: - Negative Example - Prompt: Ask a question on "event duration" based on the provided sentence.

Instructions for MC-TACO question generation task

- **Title:** Writing questions that involve commonsense understanding of "event duration".
- **Definition:** In this task, we ask you to write a question that involves “event duration", based on a given sentence. Here, event duration is defined as the understanding of how long events typically last. For example, “brushing teeth”, usually takes few minutes.
- **Emphasis & Caution:** The written questions are not required to have a single correct answer.
- **Things to avoid:** Don't create questions which have explicit mentions of answers in text. Instead, it has to be implied from what is given. In other words, we want you to use "instinct" or "common sense".

Positive Example

- **Input:** Sentence: Jack played basketball after school, after which he was very tired.
- **Output:** How long did Jack play basketball?
- **Reason:** the question asks about the duration of an event; therefore it's a temporal event duration question.

Negative Example

- **Input:** Sentence: He spent two hours on his homework.
- **Output:** How long did he do his homework?
- **Reason:** We DO NOT want this question as the answer is directly mentioned in the text.
- **Suggestion:** -

- **Prompt:** Ask a question on "event duration" based on the provided sentence.

Mishra et al (2021)

Stats on Training Models

“These models are hard to run on easily accessible devices. For example, just to do inference on BLOOM-176B, you would need to have 8x 80GB A100 GPUs (~$15k each). To fine-tune BLOOM-176B, you'd need 72 of these GPUs! Much larger models, like PaLM would require even more resources.”

“During fine-tuning, FLAN-T5 adapts the JAXbased T5X framework and selects the best model evaluated on the held-out tasks every 2k step. Compared with T5’s pre-training stage, fine-tuning costs 0.2% computational resources (approximately 128 TPU v4 chips for 37 hours).”

“WizardLM (7B) (Xu et al., 2023a) is a language model trained by fine-tuning LLaMA (7B) (Touvron et al., 2023a) on the instruction dataset Evol-Instruct generated by ChatGPT (details see Section 3.7). The fine-tuning process takes approximately 70 hours on 3 epochs based on an 8 V100 GPU with the Deepspeed Zero-3 (Rasley et al., 2020) technique.” -

“GPT-4-LLM (7B) (Peng et al., 2023) is a language model trained by fine-tuning LLaMA (7B) (Touvron et al., 2023a) on the GPT-4 (OpenAI, 2023) generated instruction dataset. The fine-tuning process takes approximately three hours on an 8*80GB A100 machine with mixed precision and fully shared data parallelism.”

“Alpaca (7B) (Taori et al., 2023) is a language model trained by fine-tuning LLaMA (7B) (Touvron et al., 2023a) on the constructed instruction dataset generated by InstructGPT (175B, text-davinci003) (Ouyang et al., 2022). The fine-tuning process takes around 3 hours on an 8-card 80GB A100 device with mixed precision training and fully shared data parallelism.”