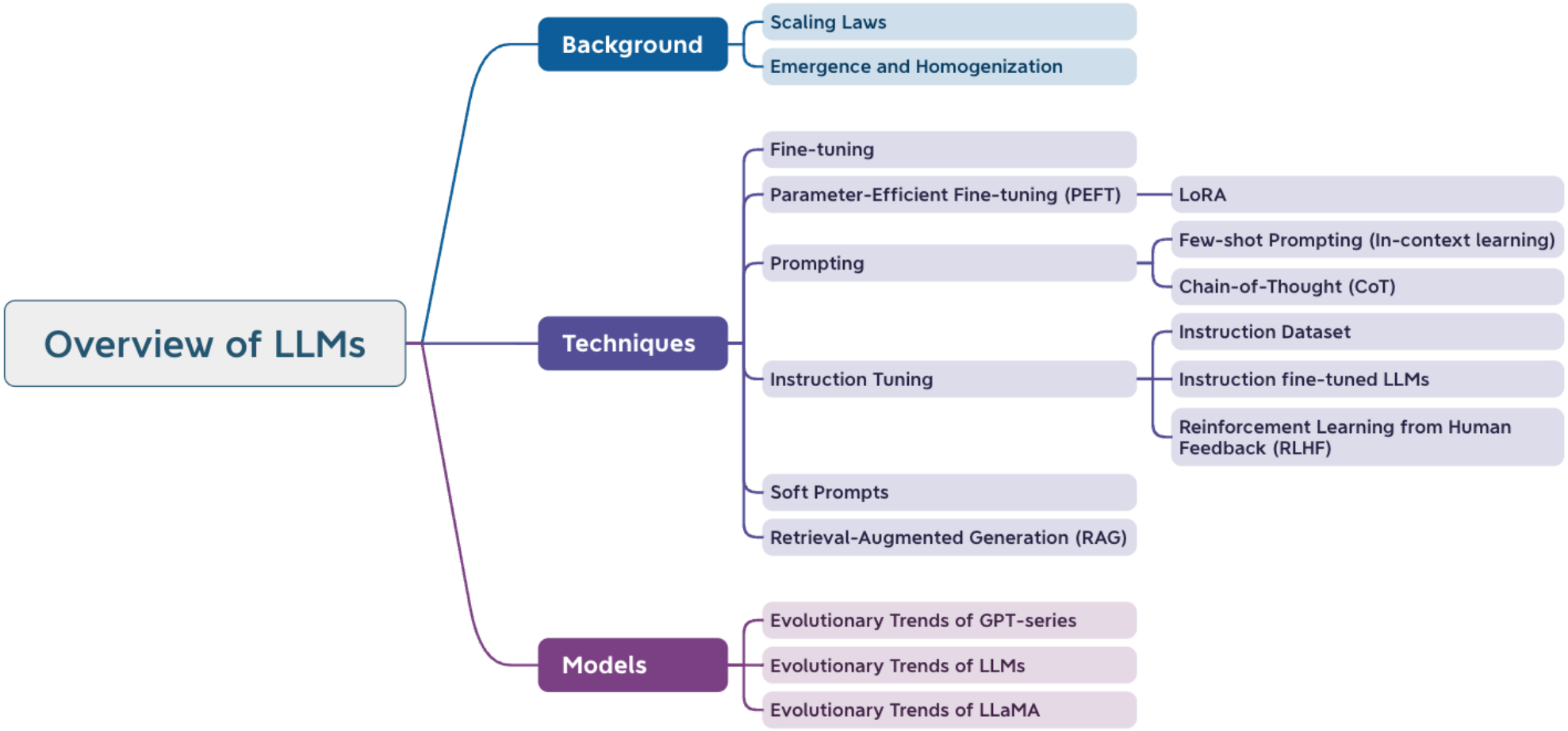
Updated: 6 Oct 2023

Overview of Large Language Models

Jean Lee

University of Sydney NLP Group

Table of Contents



THE UNIVERSITY OF SYDNEY

Research Trends of Large Language Models (LLMs)

- LLMs : rapidly growing research areas
- Stimulated domain-specific research and applications (e.g. science, healthcare, finance, and education)

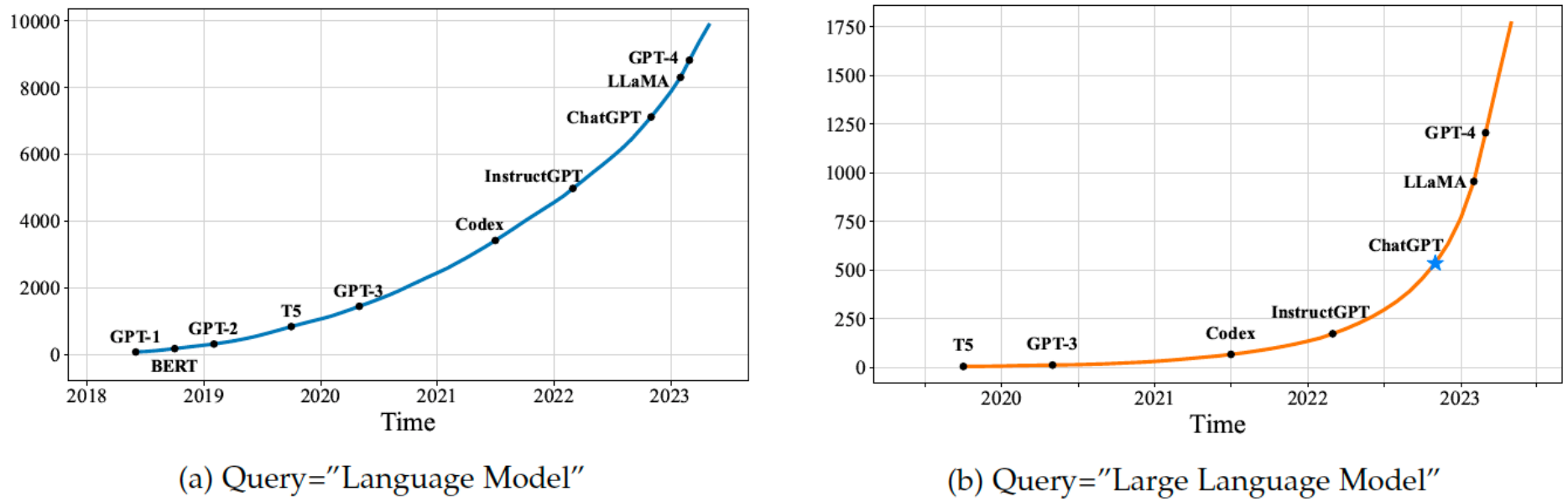


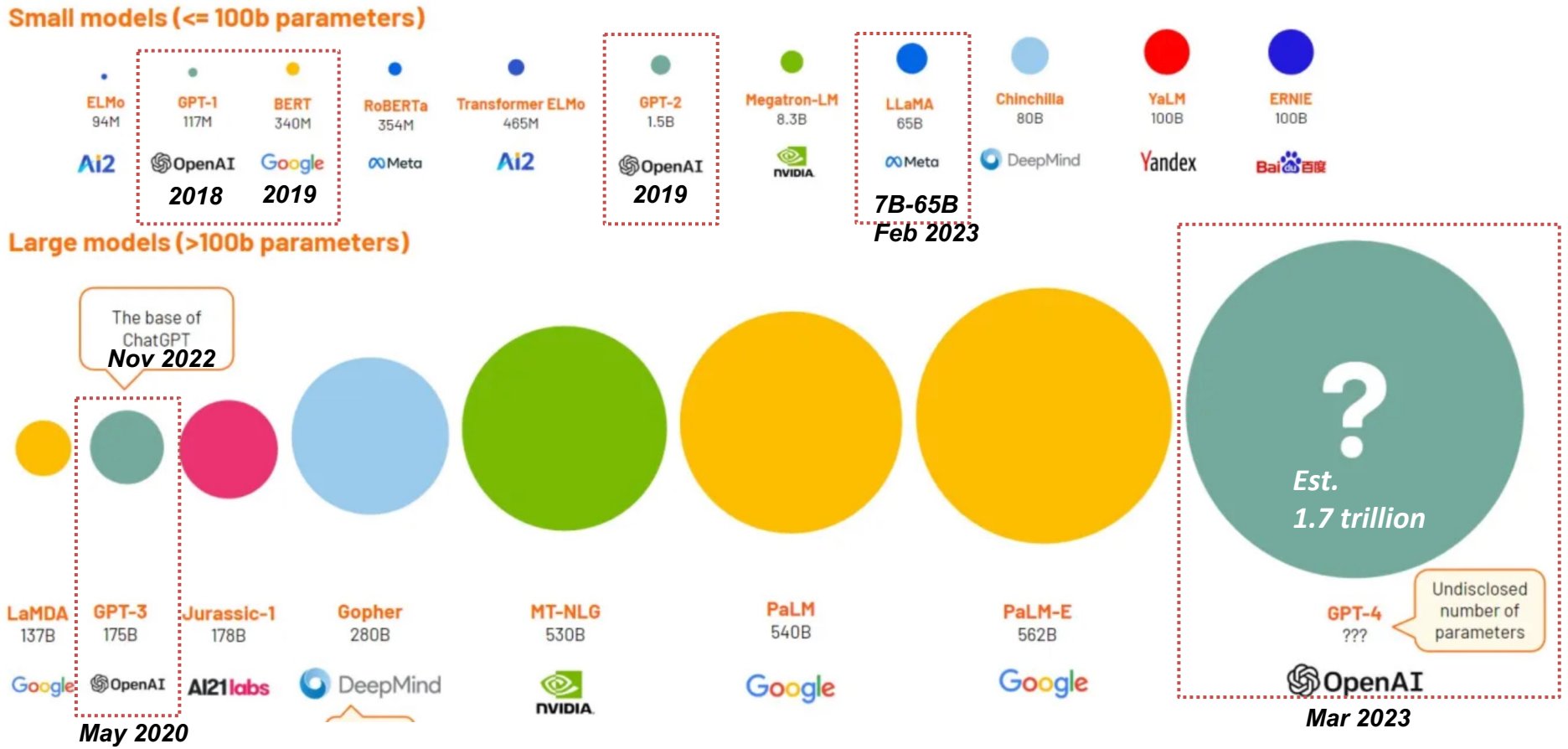
Fig : The trends of the cumulative numbers of arXiv papers that contain the keyphrases "language model" (since June 2018) and "large language model" (since October 2019), respectively. A sharp increase occurs after the release of ChatGPT. The average number of published arXiv papers that contain "large language model" in title or abstract goes from 0.40 per day to 8.58 per day.

- Fig: A Survey of Large Language Models (WX Zhao et al., Sep 2023) [paper] [github]

THE UNIVERSITY OF SYDNEY

Large Language Models

- **Scale** : a billion to a trillion parameters (PLMs -> large parameter size) + **Architecture** (Transformer Variants)



Recently,

- LLaMA 2
*(**18 Jul 2023**,*
7B, 13B, 70B
Open source –
Meta & HuggingFace)

- GPT-4V
*(**25 Sep 2023**,*
ChatGPT-plus)

- Fig: https://thelowdown.momentum.asia/the-emergence-of-large-language-models-llms/
- LLaMA2 download (Jul 2023) : https://ai.meta.com/llama/
- Sparks of Artificial General Intelligence: Early experiments with GPT-4 (S Bubeck et al., Microsoft Research, Apr 2023) [paper]

Scaling Laws – LLMs performance

- Scaling up language models improves the **model performance** -> "**What factors to be scaled**?"
- OpenAI team (Jan 2020) ***: Model size (parameters), Dataset size, and Amount of compute*** -> lead to larger model (e.g. GPT-3)
- DeepMind team (Mar 2022): Model size and the number of training tokens should be scaled equally. -> Chinchilla 70B outperforms.
- Human brain : Est. Average 86B neurons.

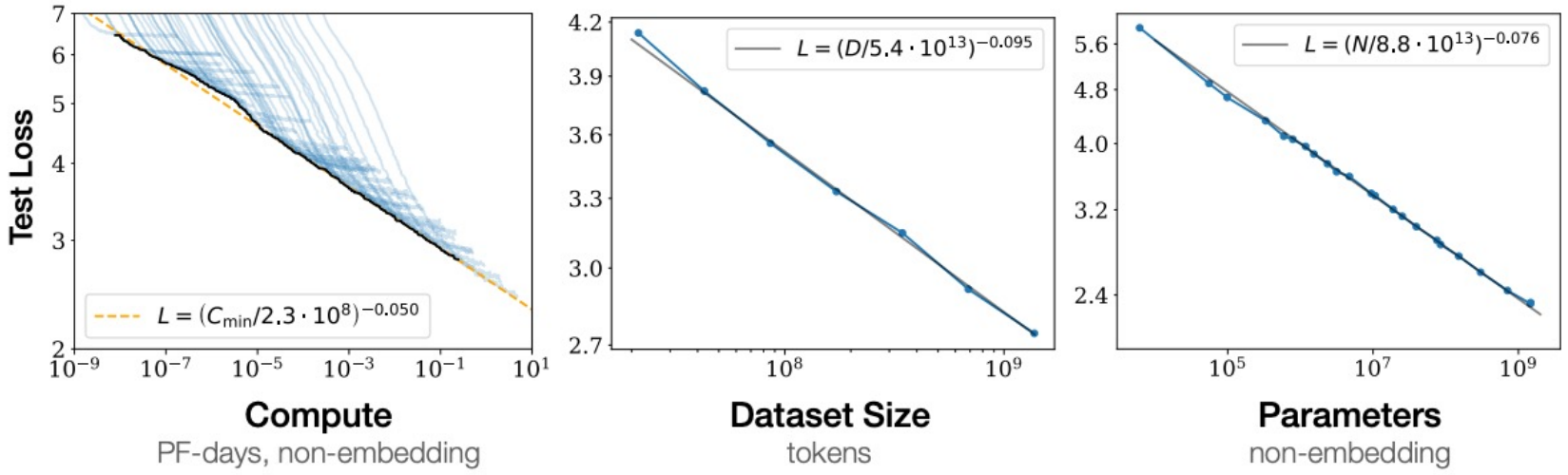


Fig : Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

- Fig: Scaling Laws for Neural Language Models (J Kaplan et al., OpenAI, Jan 2020) [paper]
- [Chinchilla] Training Compute-Optimal Large Language Models (J Hoffmann et al., DeepMind, Mar 2022) [paper]
- A new Moore's Law? https://huggingface.co/blog/large-language-models

Emergence and Homogenization – LLMs characters

- **Emergence** : the behavior of an AI system is **implicitly induced** rather than explicitly constructed
  - Some ability of LM is not present in smaller models but is present in larger models (e.g. GPT-3 : in-context learning)
- **Homogenization** : the consolidation of methodologies for building AI systems across **a wide range** of applications
  - by **tuning**: Generalist (intermediate) -> Specialist (task-specific, domain-specific)
- **Foundation Models** : AI models that surpass language-centric capabilities (e.g. multi-modal)

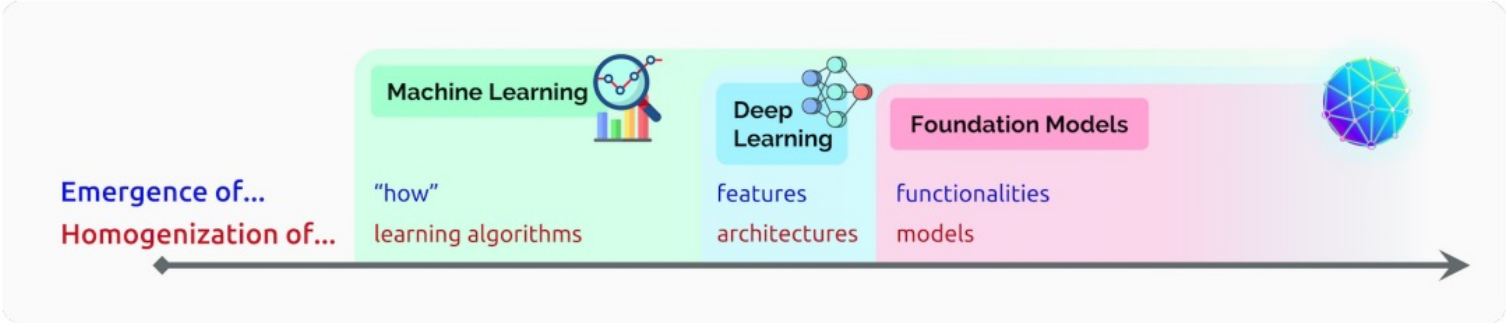


Fig : The story of AI has been one of increasing **emergence and homogenization**. With the introduction of machine learning, **how a task is performed emerges (is inferred automatically) from examples**; with deep learning, the high-level features used for prediction emerge; and with foundation models, even advanced functionalities such as in-context learning emerge. At the same time, machine learning homogenizes learning algorithms (e.g., logistic regression), deep learning homogenizes model architectures (e.g., Convolutional Neural Networks), and **foundation models homogenizes the model itself** (e.g., GPT-3).

On the Opportunities and Risks of Foundation Models

Rishi Bommasani* Drew A. Hudson Ehsan Adeli Russ Altman Simran Arora
Sydney von Arx Michael S. Bernstein Jeannette Bohg Antoine Bosselut Emma Brunskill
Erik Brynjolfsson Shyamal Buch Dallas Card Rodrigo Castellon Niladri Chatterji
Annie Chen Kathleen Creel Jared Quincy Davis Dorottya Demszky Chris Donahue
Moussa Doumbouya Esin Durmus Stefano Ermon John Etchemendy Kawin Ethayarajh
Li Fei-Fei Chelsea Finn Trevor Gale Lauren Gillespie Karan Goel Noah Goodman
Shelby Grossman Neel Guha Tatsunori Hashimoto Peter Henderson John Hewitt
Daniel E. Ho Jenny Hong Kyle Hsu Jing Huang Thomas Icard Saahil Jain
Dan Jurafsky Pratyusha Kalluri Siddharth Karamcheti Geoff Keeling Fereshte Khani
Omar Khattab Pang Wei Koh Mark Krass Ranjay Krishna Rohith Kuditipudi
Ananya Kumar Faisal Ladhak Mina Lee Tony Lee Jure Leskovec Isabelle Levent
Xiang Lisa Li Xuechen Li Tengyu Ma Ali Malik Christopher D. Manning
Suvir Mirchandani Eric Mitchell Zanele Munyikwa Suraj Nair Avanika Narayan
Deepak Narayanan Ben Newman Allen Nie Juan Carlos Niebles Hamed Nilforoshan
Julian Nyarko Giray Ogut Laurel Orr Isabel Papadimitriou Joon Sung Park Chris Piech
Eva Portelance Christopher Potts Aditi Raghunathan Rob Reich Hongyu Ren
Frieda Rong Yusuf Roohani Camilo Ruiz Jack Ryan Christopher Ré Dorsa Sadigh
Shiori Sagawa Keshav Santhanam Andy Shih Krishnan Srinivasan Alex Tamkin
Rohan Taori Armin W. Thomas Florian Tramèr Rose E. Wang William Wang Bohan Wu
Jiajun Wu Yuhuai Wu Sang Michael Xie Michihiro Yasunaga Jiaxuan You Matei Zaharia
Michael Zhang Tianyi Zhang Xikun Zhang Yuhui Zhang Lucia Zheng Kaitlyn Zhou
Percy Liang*[1]

Center for Research on Foundation Models (CRFM)
Stanford Institute for Human-Centered Artificial Intelligence (HAI)
Stanford University

- Fig: On the Opportunities and Risks of Foundation Models (CRFM, HAI at Stanford, v1-Aug 2021) [paper]

THE UNIVERSITY OF SYDNEY

Fine-tuning

- Fine-tuning is an approach to transfer learning by **modifying the weights** of a pre-trained model (LLMs) to help it perform better on a **specific task or set of tasks**.
- **Full fine-tuning** : update all model parameters, use a smaller dataset

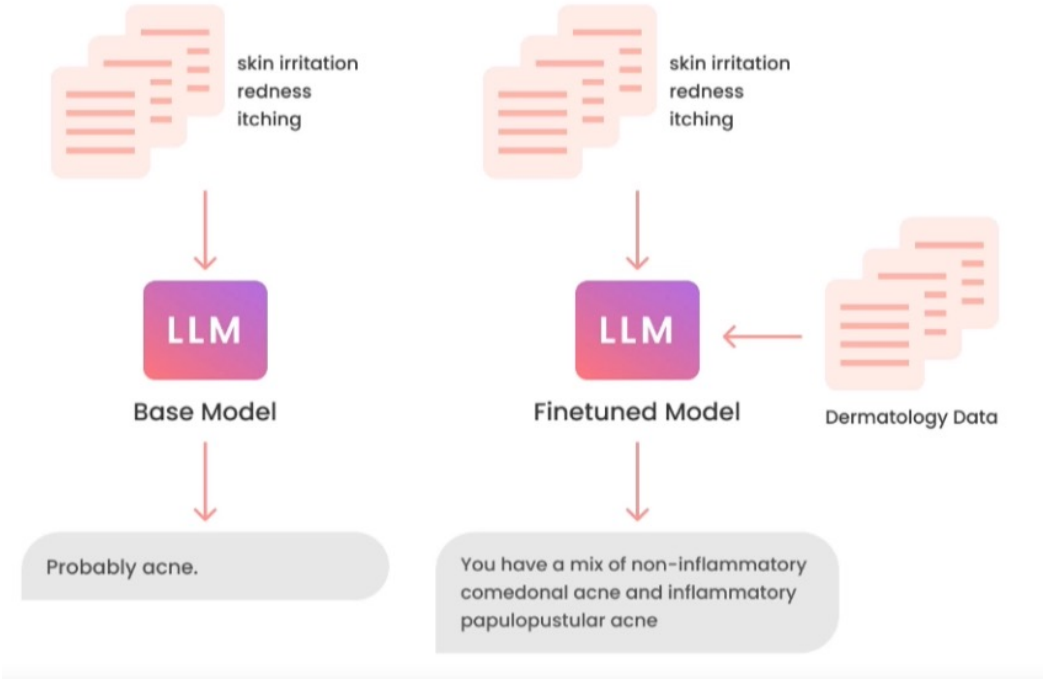


Fig 1: Pretrain-finetune Flow

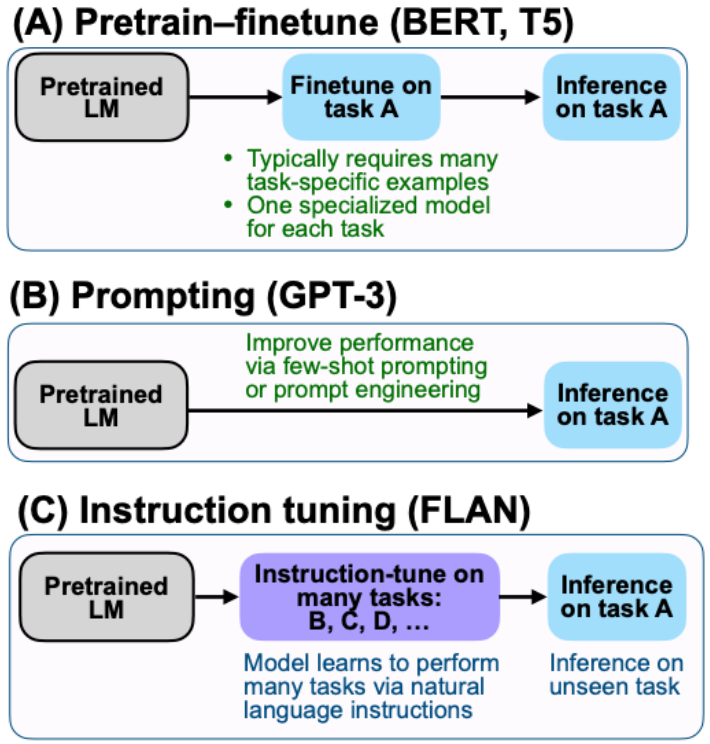


Fig 2: Comparing pretrain–finetune, prompting and instruction tuning

- *Fig 1: DeepLearningAI: https://www.deeplearning.ai/short-courses/finetuning-large-language-models/*
- *Fig 2: [FLAN] Finetuned Language Models Are Zero-Shot Learners (J Wei et al., Google Research, v1-Sep 2021, ICLR 2021) [paper]*
- *Ludwig (Low code framework) : https://ludwig.ai/latest/user_guide/llms/finetuning*

Parameter-Efficient Fine-tuning (PEFT)

- Why fine-tune all the parameters? -> find a method for **efficiently adapting LLMs** to various downstream applications.
- PEFT : **only fine-tune a small number of (extra) model parameters**, significantly decreasing computational and storage cost.

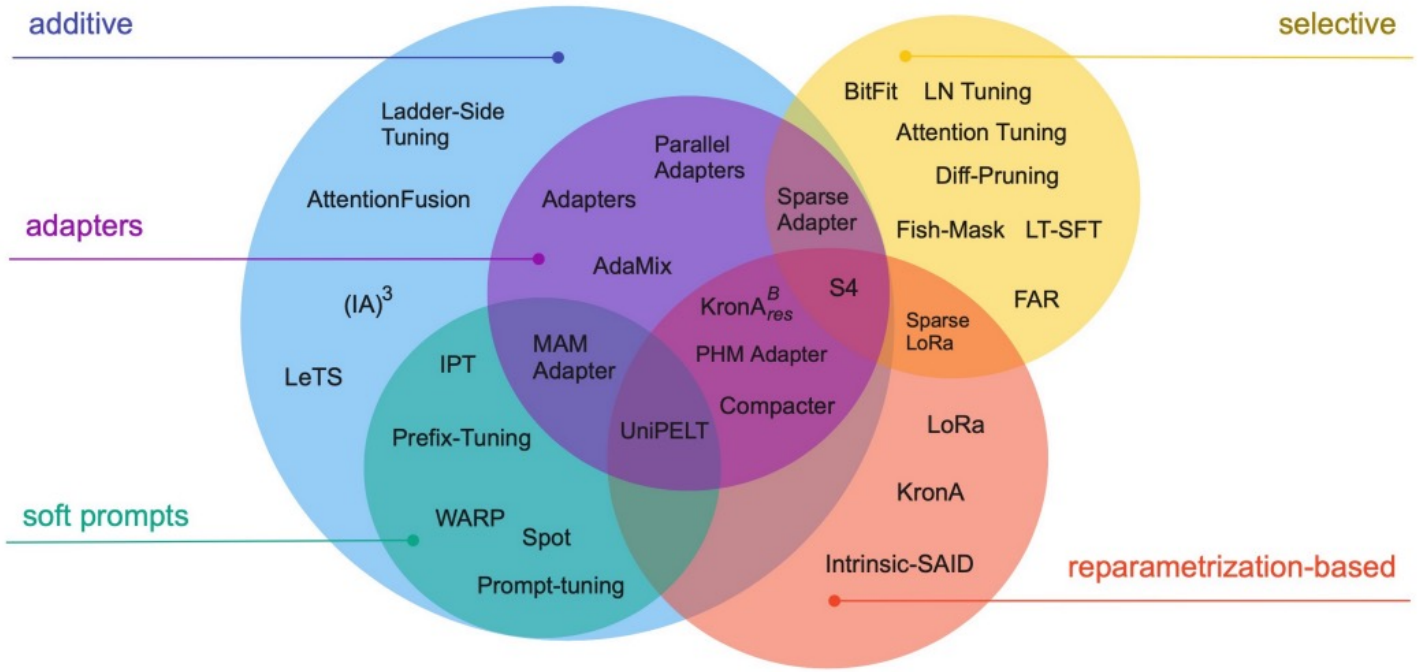


Fig : **Parameter-efficient fine-tuning (PEFT)** methods taxonomy.

- **Addition-based**: adding extra parameters or layers and training only the newly added parameters.
  - Adapters
  - Soft Prompts (E.g. Prompt Tuning)
- **Selection-based**: fine-tuning only a few top layers of a network (E.g. BitFit)
- **Reparametrization-based**: leveraging low-rank representation to minimize the number of trainable parameters (E.g. **LoRA**)

- Fig: Scaling Down to Scale Up: A Guide to Parameter-Efficient Fine-Tuning (V Lialin et al., Mar 2023) [paper]
- HuggingFace PEFT: https://huggingface.co/docs/peft/index

Low-Rank Adaptation (LoRA)

- An approach to represent the weight updates with two smaller matrices (called **update matrices**) through low-rank decomposition.
- LM has low "instrisic dimension" and can still learn efficiently despite a random projection to a smaller subspace.
- By updating a much smaller number of parameters, we reduce the computational and memory requirements.

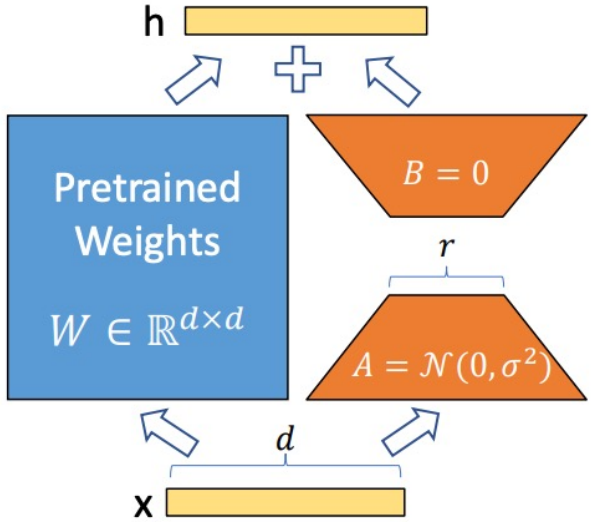


Fig: **Reparametrization**. LoRA only trains A and B..

Methods

- $W_0 + \Delta W = W_0 + BA$ (x = input)
- **Original weight matrix** $W_0 \in \mathbb{R}^{d \times d}$: frozen and doesn't receive any further adjustments
- **New low-rank matrix** $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times d}$: reparametrize the original weight matrix into two matrices, A and B, each low rank r.
- **Final results** $h = W_0 x + \Delta W x = W_0 x + BAx = (W_0 + BA)x$: both the original and the adapted weights are summed coordinate-wise.

Advantages

Efficient Fine-Tuning

- Fewer trainable parameters : For GPT3, 10,000x less
- Less GPU memory : for GPT3, 3x less
- Slightly below accuracy to finetuning

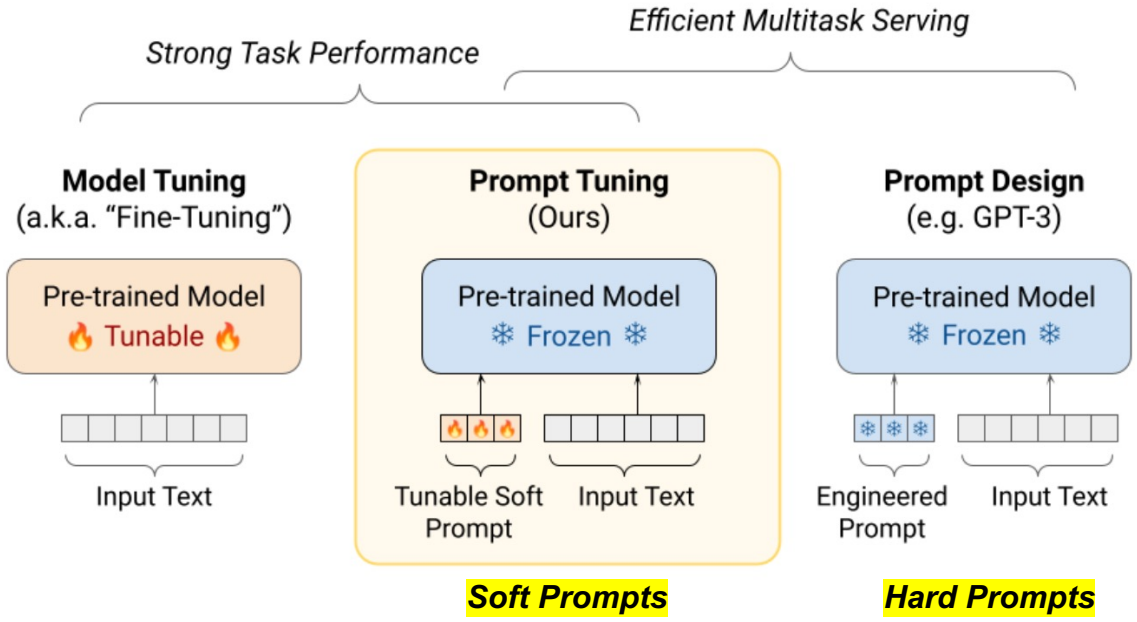
No Additional inference latency

- For another downstream task, recover original weight (W_0) and adding a different new LoRA ($B'A'$)
- A quick option with very little memory overhead.

- Fig: [LoRA] Low-Rank Adaptation of Large Language Models (EJ Hu et al., Microsoft, Jun 2021, ICLR 2021) [paper]
- QLoRA: Efficient Finetuning of Quantized LLMs (T Dettmers et al., U of Washington, May 2023) [paper]
- HuggingFace PEFT: https://huggingface.co/docs/peft/conceptual_guides/lora
- Efficient Fine-Tuning for Llama-v2-7b (Ludwig): https://www.youtube.com/watch?v=g68qlo9Izf0

Prompting

- **Prompting** : frozen LLMs for a specific downstream task by including a (text) prompt that describes the task or even demonstrates an example of the task.
- **Prompt Engineering** : involves human writing, refining and optimizing prompts in a structured way.



Hard Prompts (a.k.a. Prompt Engineering)

- manually handcrafted **text prompts** with discrete input tokens;
- (-) it requires a lot of effort to create a good prompt.
- (e.g. **Few-shot, Chain of Thought, Instruction tuning**)

> I want you to act as a spoken English teacher. I will speak to you in English and you will reply to me in English to practice my spoken English. I want you to keep your reply neat, limiting the reply to 100 words. I want you to strictly correct my grammar mistakes and typos. I want you to ask me a question in your reply. Now let's start practicing, you could ask me a question first. Remember, I want you to strictly correct my grammar mistakes, typos and factual errors.

Soft Prompts (a.k.a. Parameter-Efficient Prompt Tuning)

- are **learnable tensors** concatenated with the input embeddings that can be optimized to a dataset
- (-) they aren't human readable.
- (e.g. **Prompt Tuning, Prefix Tuning, P-tuning**)

- What is Prompt Tuning? (IBM) : https://www.youtube.com/watch?v=yu27PWzJI_Y
- Fig (Google Research) : https://blog.research.google/2022/02/guiding-frozen-language-models-with.html
- HuggingFace PEFT: https://huggingface.co/docs/peft/conceptual_guides/prompting

- Prompt Engineering Guide: https://www.promptingguide.ai
- Prompt Engineering Tutorial : https://www.youtube.com/watch?v=_ZvnD73m40o
- GPT (prompt) best practices: https://platform.openai.com/docs/guides/gpt-best-practices

Few-shot Prompting (In-context learning)

- Human can generally perform a new language task from only a few examples -> "**can LMs learn from a few-shot examples?**"
- **GPT-3 175B** : scaling up autoregressive LM -> emergent ability of **"in-context learning"**
- **In-context learning :** the model develops a broad set of skills at training time, and then uses those abilities at inference time to rapidly adapt to the desired task -> ***an ability that is non-existent in small models but rapidly improves performance.***

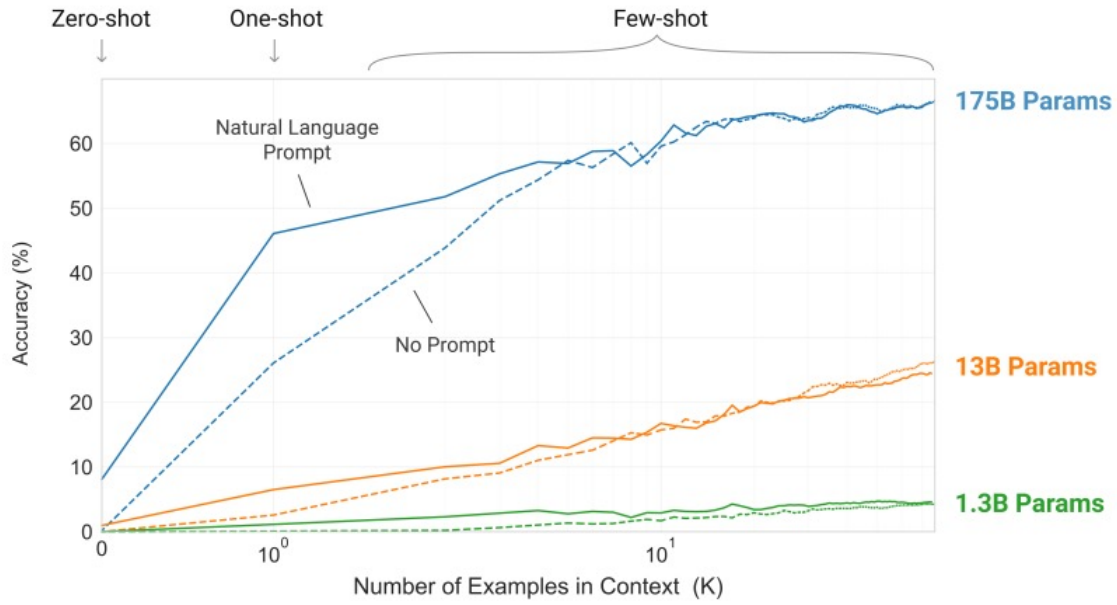


Fig : **Larger models make increasingly efficient use of in-context information.** The steeper "in-context learning curves" for large models demonstrate improved ability to learn a task from contextual information. We see qualitatively similar behavior across a wide range of tasks.

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
Translate English to French:        ← task description
sea otter => loutre de mer          ← examples
peppermint => menthe poivrée        ←
plush girafe => girafe peluche      ←
cheese =>                           ← prompt
```

No gradient updates
- **Zero-shot** : task description, prompt
- **One-shot** : + one example
- **Few-shot** : + a few examples

Gradient updates
- **fine-tuning**

- Fig: [GPT 3] Language Models are Few-Shot Learners (TB Brown et al., OpenAI, May 2020, NIPS 2020) [paper]

Chain-of-Thought (CoT) Prompting

- **Chain-of-thought (CoT)**: enables complex reasoning capabilities through intermediate reasoning steps (Few-shot-CoT).
- **Zero-shot-CoT**: just add "***Let's think step by step***" to the original prompt (Not so bad, LM-designed outperforms human-designed).

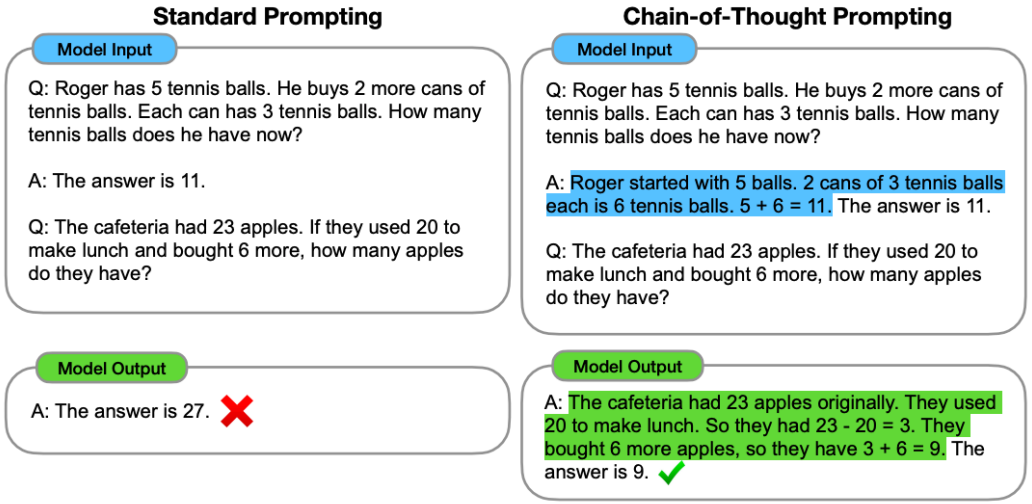


Fig 1: **Chain-of-thought prompting** enables large language models to tackle complex **arithmetic, commonsense, and symbolic reasoning** tasks. Chain-of-thought reasoning processes are highlighted. (using step-by-step reasoning examples per task)

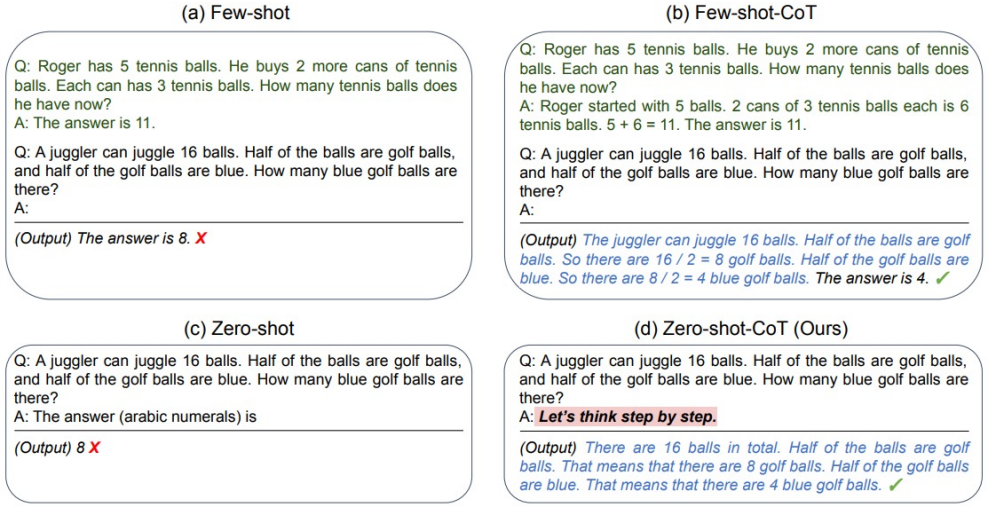


Fig 2: (a) standard Few-shot ([Brown et al., 2020]), (b) Few-shot-CoT ([Wei et al., 2022]), (c) standard Zero-shot, and (d) ours (**Zero-shot-CoT**). Zero-shot-CoT does not need any examples and just uses **the same prompt "Let's think step by step" across all tasks** (arithmetic, symbolic, commonsense, and other logical reasoning tasks)

- *Fig 1: Chain-of-Thought Prompting Elicits Reasoning in Large Language Models (J Wei et al., Google Research, v1- Jan 2022, NIPS 2022) [paper]*
- *Fig 2: Large Language Models are Zero-Shot Reasoners (T Kojima et al., U of Tokyo, v1-May 2022, NIPS 2022) [paper]*
- *Automatic Chain of Thought Prompting in Large Language Models (Z Zhang et al., Oct 2022, ICLR 2022) [paper]*
- *Towards Reasoning in Large Language Models: A Survey (J Huang and KCC Chang, U of Illinois, May 2023, ACL 2023 Findings) [paper]*

Instruction Tuning – Instruction Dataset

- Instruction Tuning : 1) Constructing Instruction **dataset**, 2) Fine-tuning base LLMs -> Instruction fine-tuned **LLMs**

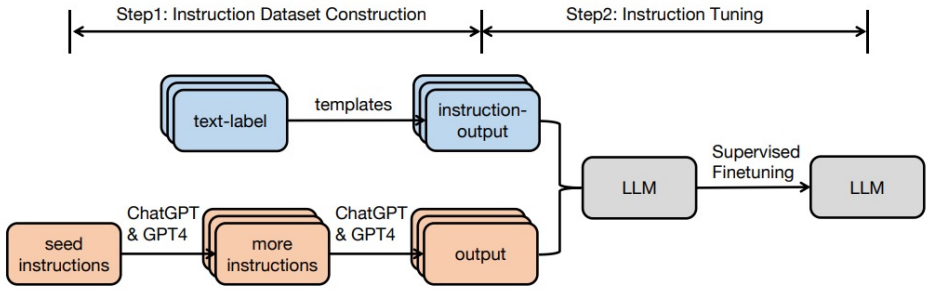


Fig 1: General pipeline of instruction tuning

Step1. Instruction Dataset Construction

1) Data integration from annotated existing dataset

- Using templates (instruction, output) pairs
- (e.g.) Super-Natural Instructions, P3, xP3, FLAN, etc.

2) Generating instruction/output using LLMs

- Manually collecting seed instructions
- Expanding instructions based on a small seed instructions using LLMs (e.g. GPT 3.5 or GPT4)
- Generating outputs using LLMs
- (e.g.) Self-Instruct, Alpaca, GPT-4-LLM, etc.

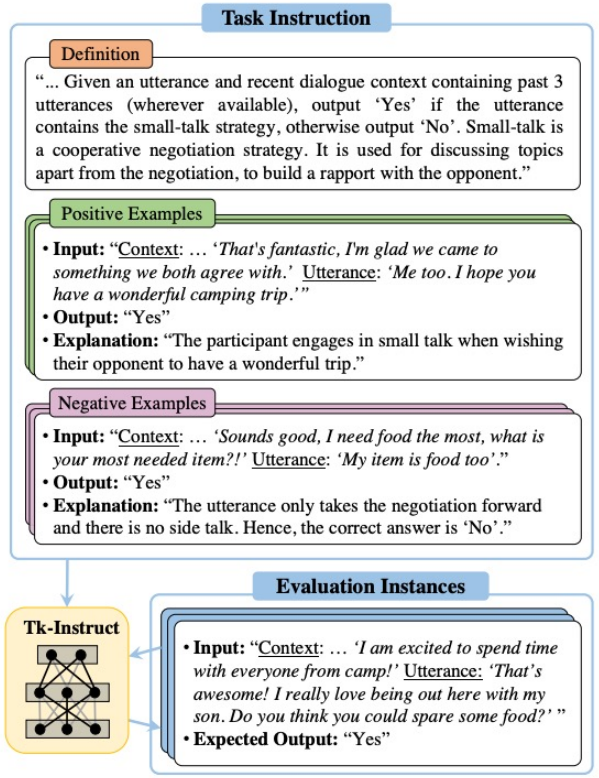


Fig 2: An example task from SUP-NATINST

Instruction Dataset Elements

- **Instruction**: natural language text sequence to specify the task. (e.g. "*determine the sentiment of the sentence*.")
- **Context (optional)** : external information or additional context that can steer the model to better responses (i.e. positive exs., negative exs., and constrains)
- **Input**: the input or question that we are interested to find a response for (e.g. "*He likes the cat." Is positive or negative*?)
- **Output**: the type or format of the anticipated output based on the instruction and the input. (e.g. "*Positive*")

- Fig 1: Instruction Tuning for Large Language Models: A Survey (S Zhang et al., Sep 2023) [paper]
- Fig 2: Super-Natural Instructions: Generalization via Declarative Instructions on 1600+ NLP Tasks (Y Wang et al., Apr 2022, EMNLP 2022) [paper]
- [Self-Instruct]: Aligning Language Models with Self-Generated Instructions (Y Wang et al., May 2023, ACL 2023) [paper]
- [GPT-4-LLM] Instruction Tuning with GPT-4 (B Peng et al., Microsoft Research, Apr 2023) [paper]

Instruction Tuning – Instruction Tuned LLMs

Step 2. Instruction tuning: **Pretrained LLMs** -> **Instruction Dataset** Construction-> Full fine-tuning -> **Instruction fine-tuned LLMs**

- *(e.g.)* ***PaLM (540B)*** + ***FLAN 2022*** *(1.8k tasks | 15M examples | zero-, few-, CoT)* -> ***FLAN-PaLM (540B)***
- *FLAN-PaLM (540B) outperforms PaLM (540B) by a large margin (+9.4% on average)*

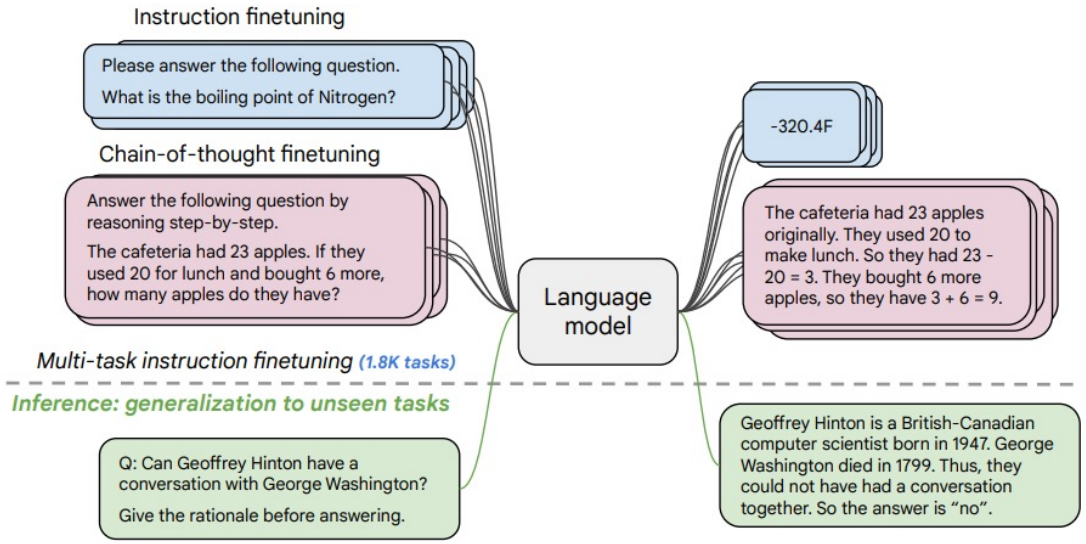


Fig 1: The overview framework of FLAN PaLM. It is fine-tuned by both with and without exemplars (i.e., zero-shot and few-shot) and with and without chain-of-thought, enabling generalization across a range of evaluation scenarios.

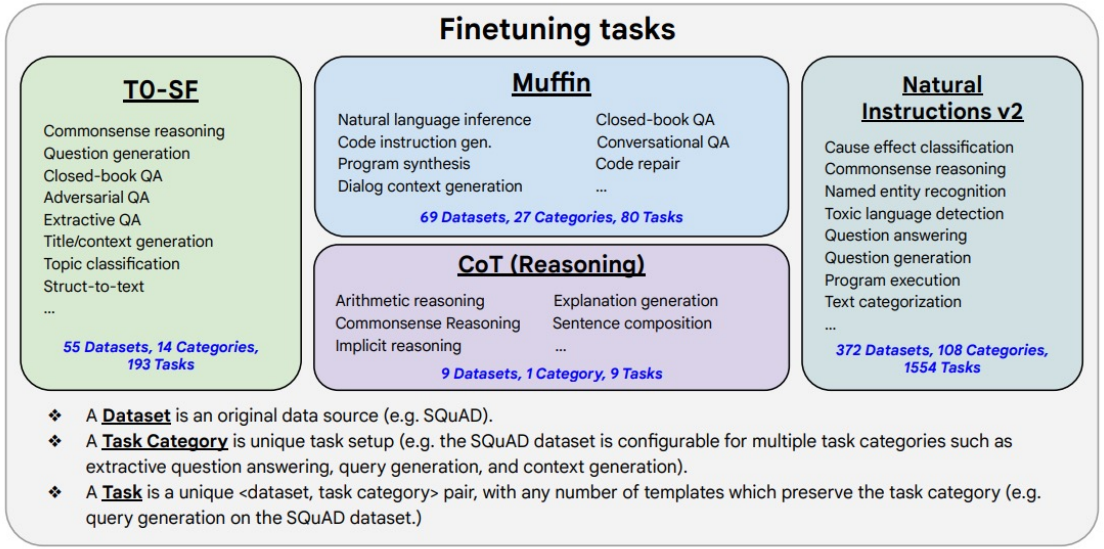


Fig 2: FLAN PaLM's finetuning data comprises 473 datasets, 146 task categories, and 1,836 total tasks.

- *[FLAN] Finetuned Language Models Are Zero-Shot Learners (J Wei et al., Google Research, v1-Sep 2021, ICLR 2021) [paper][github]*
- *[FLAN 2022] The Flan Collection: Designing Data and Methods for Effective Instruction Tuning (S Longre et al., Google Research, Feb 2023) [paper][github]*
- *[PaLM]: Scaling Language Modeling with Pathways (A Chowdhery et al., Google Research, v1- Apr 2022, JMLR 2023) [paper]*
- *Fig 1, 2: [FLAN PaLM] Scaling Instruction-Finetuned Language Models (HW Chung et al., Google, v1-Oct 2022) [paper]*

Instruction Tuning – Instruction Tuned LLMs

		Model Details				Data Collection & Training Details			
Release	Collection	Model	Base	Size	Public?	Prompt Types	Tasks in Flan	# Exs	Methods
2020 05	UnifiedQA	UnifiedQA	RoBerta	110-340M	P	ZS	46 / 46	750k	
2021 04	CrossFit	BART-CrossFit	BART	140M	NP	FS	115 / 159	71.M	
2021 04	Natural Inst v1.0	Gen. BART	BART	140M	NP	ZS / FS	61 / 61	620k	+ Detailed k-shot Prompts
2021 09	Flan 2021	Flan-LaMDA	LaMDA	137B	NP	ZS / FS	62 / 62	4.4M	+ Template Variety
2021 10	P3	T0, T0+, T0++	T5-LM	3-11B	P	ZS	62 / 62	12M	+ Template Variety + Input Inversion
2021 10	MetaICL	MetaICL	GPT-2	770M	P	FS	100 / 142	3.5M	+ Input Inversion + Noisy Channel Opt
2021 11	ExMix	ExT5	T5	220M-11B	NP	ZS	72 / 107	500k	+ With Pretraining
2022 04	Super-Natural Inst.	*Tk*-Instruct	T5-LM, mT5	11-13B	P	ZS / FS	1556 / 1613	5M	+ Detailed k-shot Prompts + Multilingual
2022 10	GLM	GLM-130B	GLM	130B	P	FS	65 / 77	12M	+ With Pretraining + Bilingual (en, zh-cn)
2022 11	xP3	BLOOMz, mT0	BLOOM, mT5	13-176B	P	ZS	53 / 71	81M	+ Massively Multilingual
2022 12	Unnatural Inst.†	T5-LM-Unnat. Inst.	T5-LM	11B	NP	ZS	~20 / 117	64k	+ Synthetic Data
2022 12	Self-Instruct†	GPT-3 Self Inst.	GPT-3	175B	NP	ZS	Unknown	82k	+ Synthetic Data + Knowledge Distillation
2022 12	OPT-IML Bench†	OPT-IML	OPT	30-175B	P	ZS + FS, CoT	~2067 / 2207	18M	+ Template Variety + Input Inversion + Multilingual
2022 10	Flan 2022 (ours)	Flan-T5, Flan-PaLM	T5-LM, PaLM	10M-540B	P NP	ZS + FS, CoT	1836	15M	+ Template Variety + Input Inversion + Multilingual

Fig : A Timeline of **Public Instruction Tuning Dataset** and detailed information on the **Instruction fine-tuned LLMs**

Instruction Tuned LLM examples

FLAN-PaLM (540B) - Google
- Pretrained LLMs : **PaLM (540B)**
- Instruction dataset : **FLAN 2022** (1.8k tasks)

InstructGPT (175B) - OpenAI
- Pretrained LLMs : **GPT3 (175B)**
- Instruction dataset : **human-crafted, not public**
- ***Using RLHF***

BLOOMZ (176B) – Huggingface ++ *[github]*
- Pretrained LLMs : **BLOOM (176B)** *[paper]*
- Instruction dataset : **xP3** (53 tasks, 46 languages) *[paper]*

Alpaca (7B) – Stanford *[Link][github]*
- Pretrained LLMs : **LLaMA (7B)**
- Instruction dataset : **Self-instruct** from davinci-003 API (52K examples)**, public**

Vicuna (13B) – UC Berkley *[Link]*
- Pretrained LLMs : **LLaMA (13B)**
- Instruction dataset : **User-shared** conversations (70K examples)**, not public**

- Fig: [FLAN 2022] The Flan Collection: Designing Data and Methods for Effective Instruction Tuning (S Longre et al., Google Research, Feb 2023) [paper][github]

THE UNIVERSITY OF SYDNEY

Reinforcement Learning from Human Feedback (RLHF)

- **RLHF** : Reinforcement Learning (RL) techniques by integrating human feedback to define the reward signal.
- **InstructGPT** : aligning LLMs with user intent on a wide range of tasks by fine-tuning with human feedback.

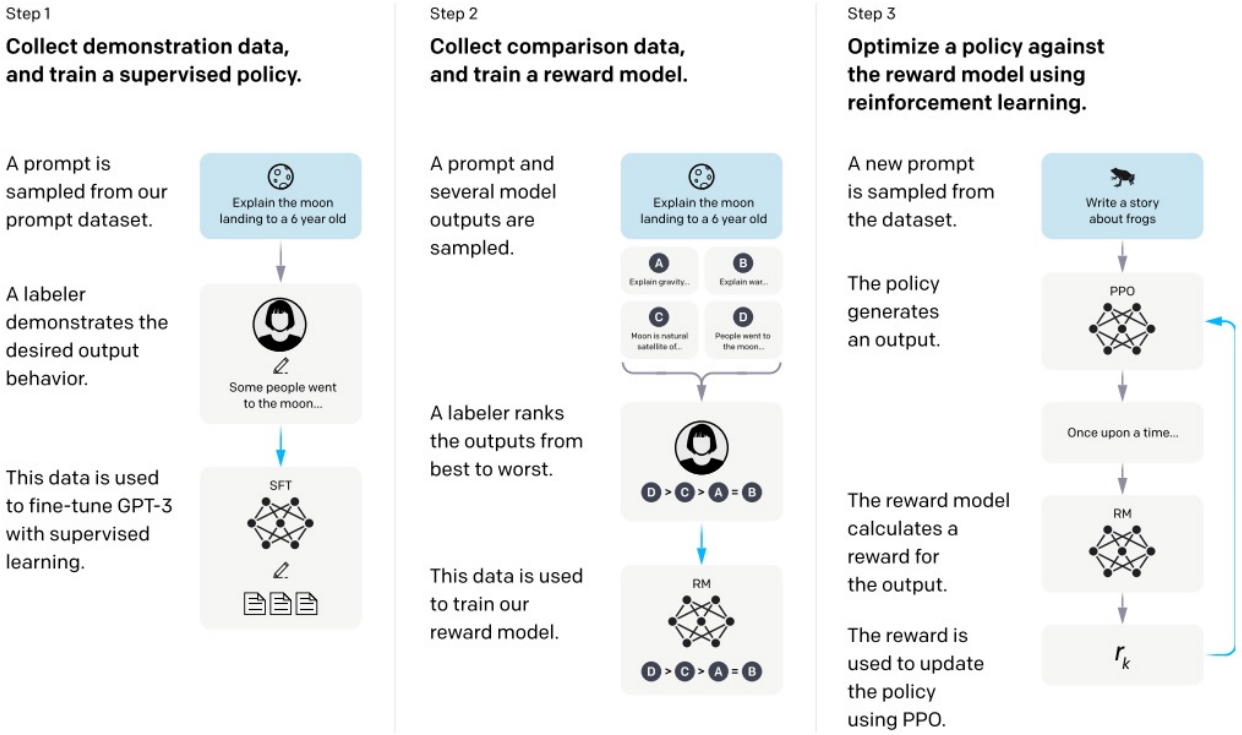


Fig: A procedure illustration of Reinforcement Learning from Human Feedback (RLHF)

Steps

1. **Supervised fine-tuning (SFT)** : An initial model is trained on the human-filtered instruction dataset using supervised learning. (e.g. fine-tune GPT3)
2. **Reward Model (RM) Training based on Human Feedback :** By sampling multiple responses for one instruction, human evaluators rank them from the best to the worst.
3. **Update SFT Model using RM as a reward function :** Use proximal policy optimization (PPO) Reinforcement learning techniques to maximize a reward.
4. **Iteration**: Steps 1 to 3 are repeated multiple times until the model performance does not significantly improve.

- Learning to summarize from human feedback (N Stiennon et al., OpenAI, Feb 2022, NIPS 2020) [paper]
- Fig: [InstructGPT] Training language models to follow instructions with human feedback (L Ouyang et al., OpenAI, Mar 2022, NIPS 2022) [paper]

Soft Prompts (Parameter-Efficient Prompt Tuning)

- Prompt Tuning Strategy : To find a lightweight alternative to fine-tuning -> **efficiency + performance**

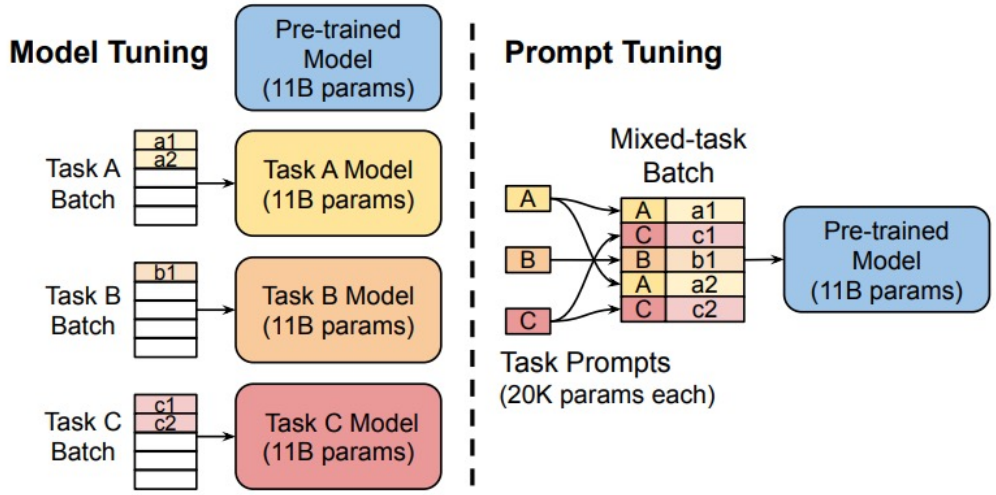


Fig1: [Prompt Tuning] **Model tuning** requires making a task-specific copy of the entire pre-trained model for each downstream task and inference must be performed in **separate batches**. **Prompt tuning** only requires **storing a small task-specific prompt** for each task and enables **mixed-task inference** using the original pretrained model.

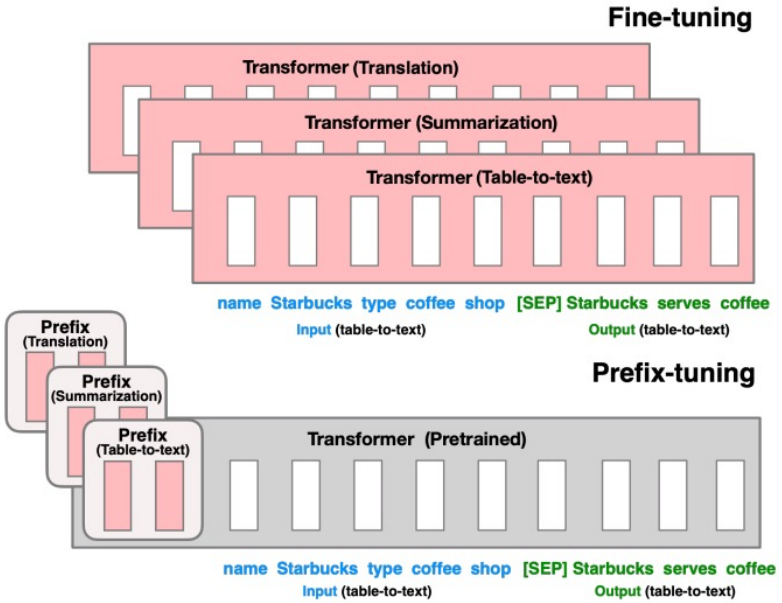


Fig2: [Prefix Tuning] Fine-tuning (top) updates all Transformer parameters (the red Transformer box) and requires storing a full model copy for each task. We propose **prefix-tuning (bottom)**, which freezes the Transformer parameters and **only optimizes the prefix (the red prefix blocks)**. Consequently, we only need to store the prefix for each task, making **prefix-tuning modular and space-efficient**.

- *Fig1: [Prompt Tuning] The Power of Scale for Parameter-Efficient Prompt Tuning (B Lester et al., Google Research, Sep 2021, EMNLP 2021) [paper]*
- *Fig 2: [Prefix Tuning] Prefix-Tuning: Optimizing Continuous Prompts for Generation (XL Li and P Liang, Stanford U, Jan 2021, ACL 2021) [paper]*
- *Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing (P Liu et al., CMU, Jul 2021 CSUR 2023) [paper]*

Retrieval-Augmented Generation (RAG)

- RAG : improving the quality of LLM-generated responses by grounding the model on external sources of knowledge.
- An ”open book” approach to answering tough questions
- Model has access to the most current and reliable facts -> Build model trust and mitigate “hallucinations” issues.

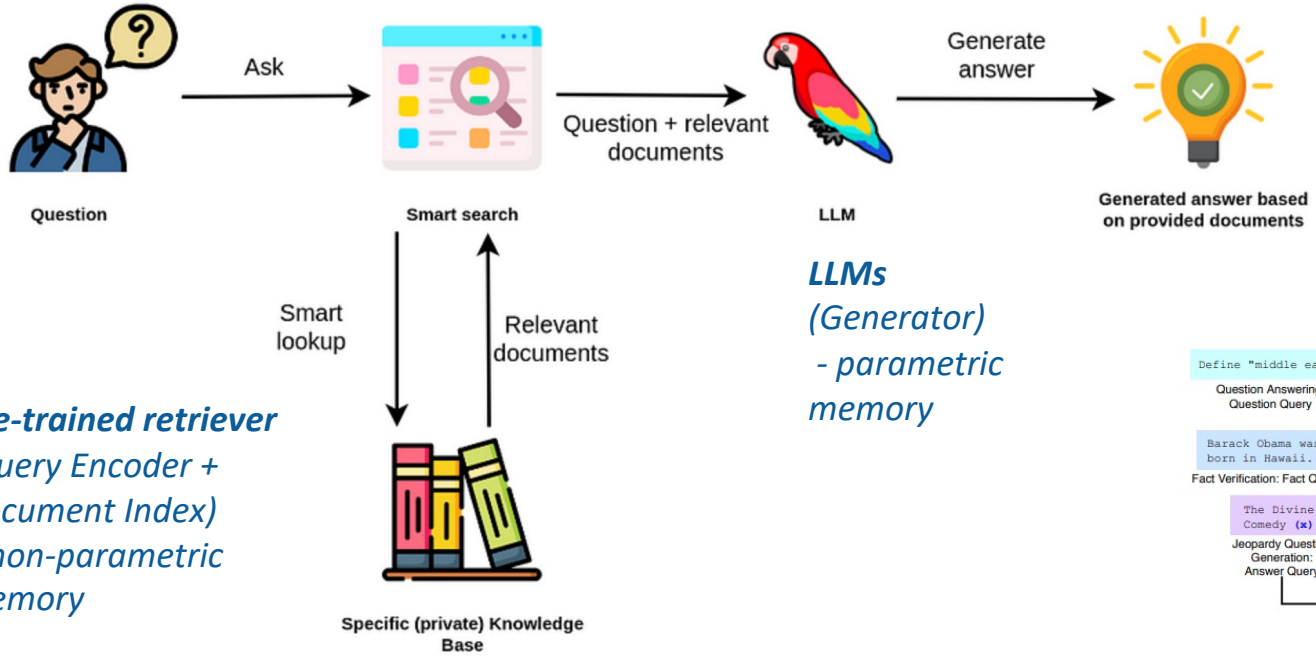


Fig 1: RAG conceptual illustration

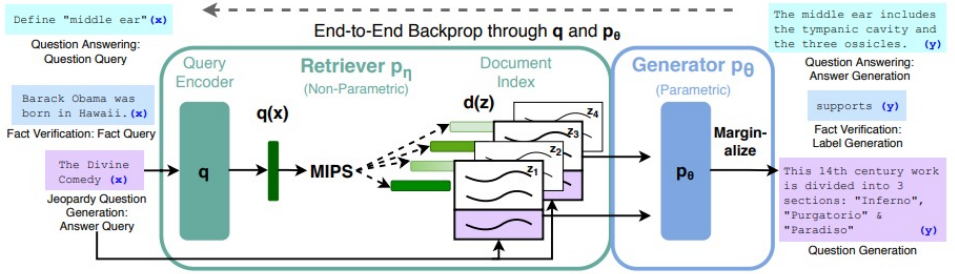


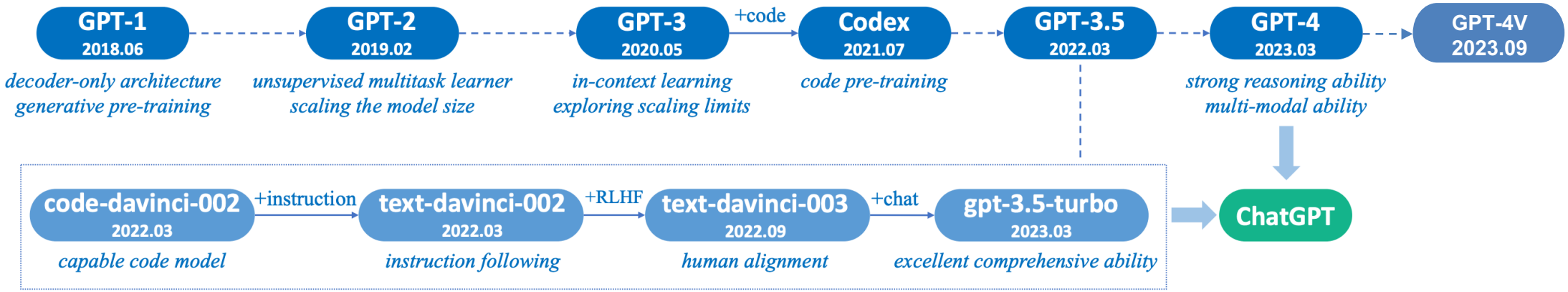
Fig 2: Overview of RAG

- *Fig 2: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (P Lewis et al., v1-May 2020, Facebook AI, NIPS 2020) [paper]*
- *Fig 1: RAG (Neo4j) https://neo4j.com/developer-blog/fine-tuning-retrieval-augmented-generation/*
- *What is RAG? (IBM) : https://www.youtube.com/watch?v=T-D1OfcDW1M*
- *Retrieval doc (LangChain): https://python.langchain.com/docs/modules/data_connection/*

Evolutionary Trends of GPT-series

- **GPT1 (2018, 0.1B)** : Uni-directional Transformer (generative, decoder only), fine-tune on each specific task.
- **GPT2 (2019, 1.5B)** : scale, a probabilistic form for multi-task solving, i.e., *p(output|input, task)*
- **GPT3 (2020, 175B)** : LLM (scaling laws), In-context learning (zero-, one-, few-shot learning)
- **Codex (2021, 12B)** : Code dataset 159GB -> GitHub Copilot (12B), code-davinci-002 (175B)
- **InstructGPT (2022, 175B)** : human alignment (RLHF) | GPT3 + Instruct GPT + Codex -> chatGPT application
- **GPT4 (2023, Est. 1700B = 1.7T)** : human-level performance (e.g. passing bar exam), multi-modal ability

Generative task Scaling



- *[GPT 1] Improving Language Understanding by Generative Pre-Training (A Radford et al., OpenAI, Jun 2018)[paper]*
- *[GPT 2] Language models are unsupervised multitask learners (A Radford et al., OpenAI, Feb 2019) [paper]*
- *[GPT 3] Language Models are Few-Shot Learners (TB Brown et al., OpenAI, May 2020, NIPS 2020) [paper]*
- *[Codex] Evaluating Large Language Models Trained on Code (M Chen et al, OpenAI, Jul 2021) [paper]*
- *[InstructGPT] Training language models to follow instructions with human feedback (L Ouyang et al., OpenAI, Mar 2022, NIPS 2022) [paper]*
- *[GPT 4] GPT-4 Technical Report (OpenAI, Mar 2023) [paper]*
- *Sparks of Artificial General Intelligence: Early experiments with GPT-4 (S Bubeck et al., Microsoft Research, Apr 2023) [paper]*
- *Fig : A Survey of Large Language Models (WX Zhao et al., Sep 2023) [paper] [github]*

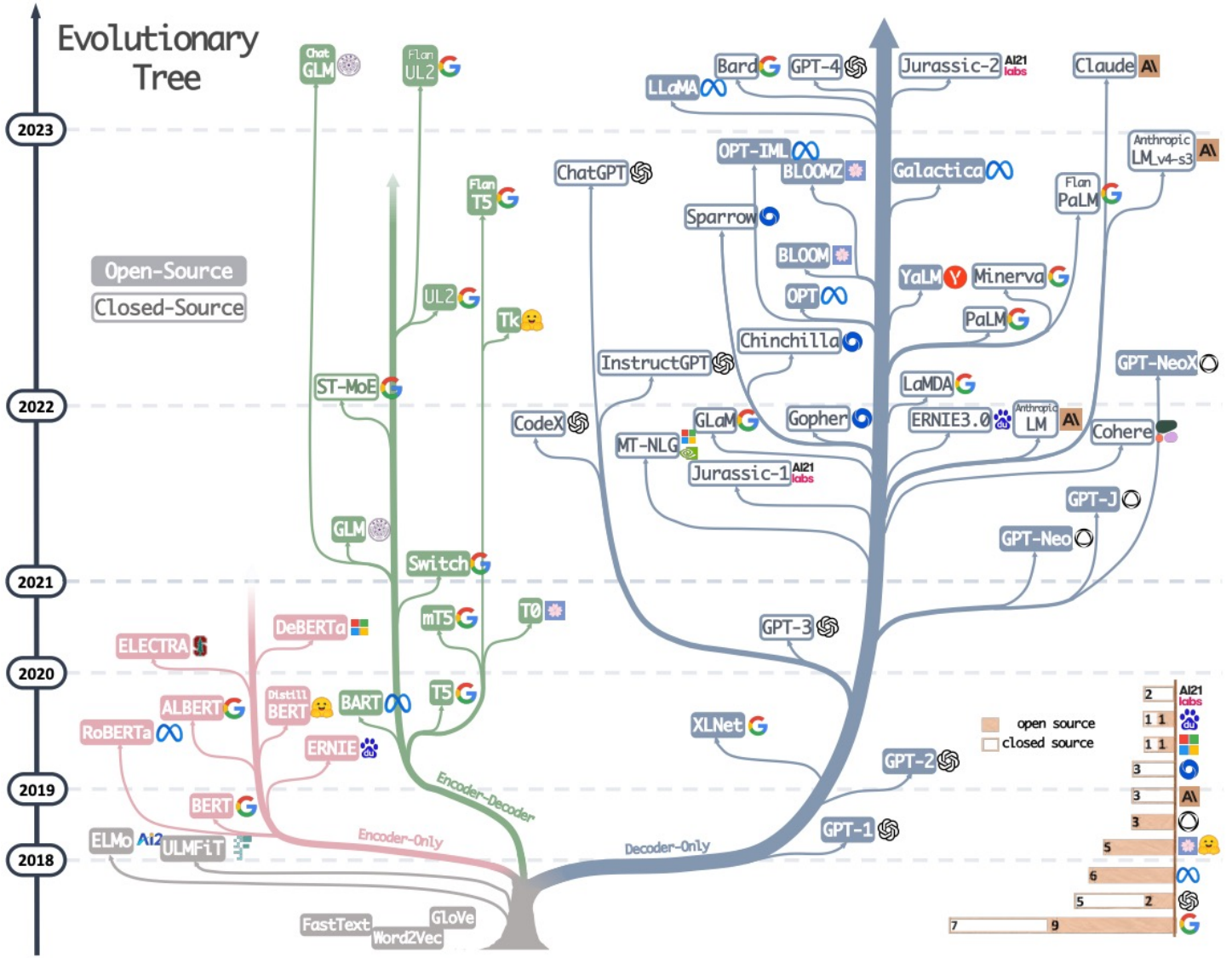
Evolutionary Trends of LLMs

Encoder-Decoder or Encoder-only (BERT-style)

- *Training : **Masked LMs***
- *Model type : **Discriminative***
- *Pretrain task : **Predict masked words***
- ***Models**: ELMo, BERT, RoBERTa, DistilBERT, ALBERT, XLM, ELECTRA, T5, etc.*

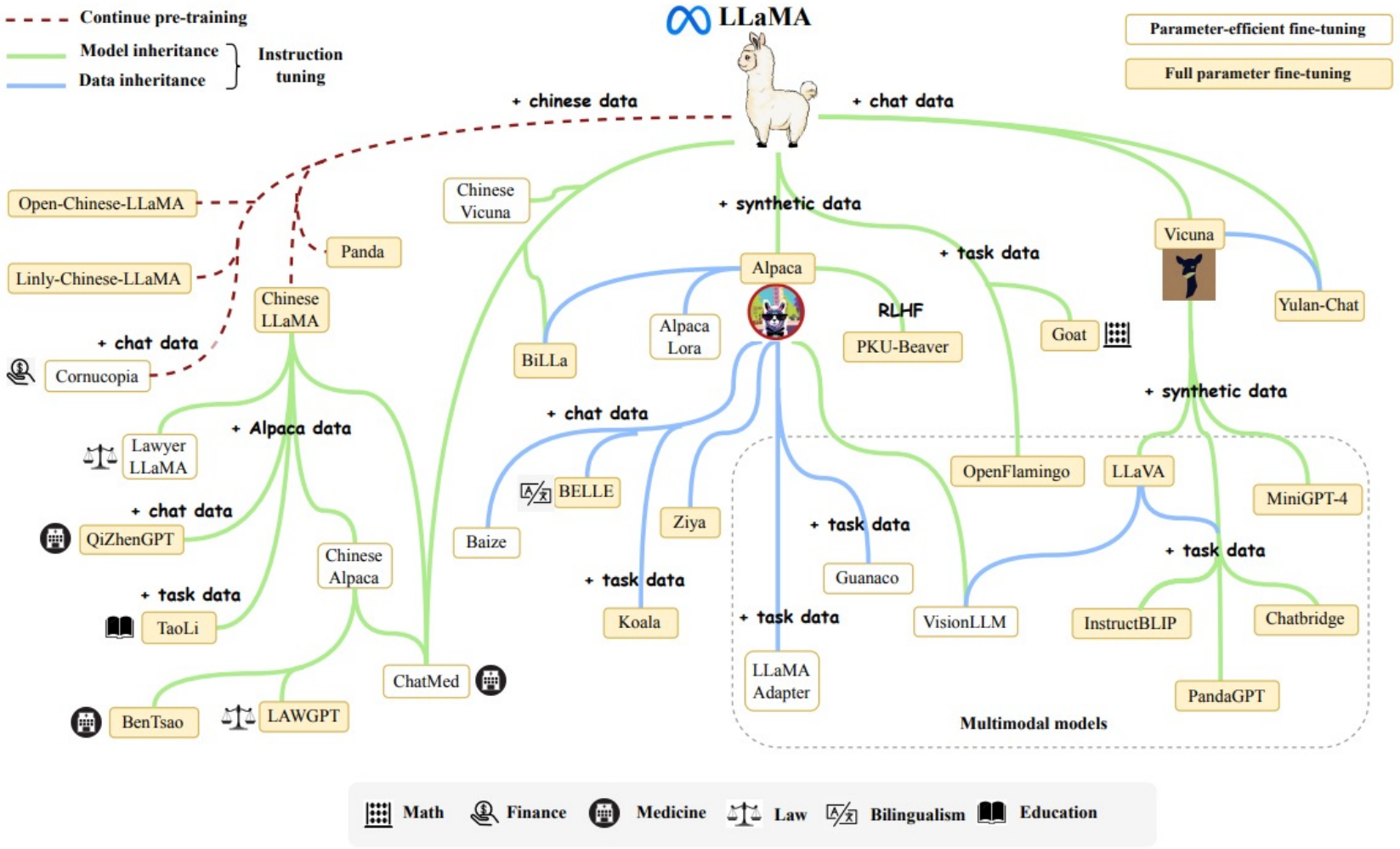
Decoder-only (GPT-style)

- *Training : **Autoregressive LMs***
- *Model type : **Generative***
- *Pretrain task : **Predict next word***
- ***Models**: GPT-3, OPT, PaLM, BLOOM, MT-NLG, Gopher, Chinchilla, LaMDA, LLaMA, etc.*



- Fig: Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond (J Yang et al., Apr 2023)[paper][github]

Evolutionary Trends of LLaMA



- *Fig : A Survey of Large Language Models (WX Zhao et al., Sep 2023) [paper] [github]*
- *LLaMA: Open and Efficient Foundation Language Models (H Touvron et al., MetaAI, Feb 2023) [paper]*
- *LLaMA2 download (Jul 2023) : https://ai.meta.com/llama/*

References

Surveys

- *On the Opportunities and Risks of Foundation Models (CRFM, HAI at Stanford, v1-Aug 2021) [paper]*
- *A Survey of Large Language Models (WX Zhao et al., Sep 2023) [paper] [github]*
- *Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond (J Yang et al., Apr 2023)[paper][github]*
- *Towards Reasoning in Large Language Models: A Survey (J Huang and KCC Chang, U of Illinois, May 2023, ACL 2023 Findings) [paper]*
- *Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing (P Liu et al., CMU, Jul 2021 CSUR 2023) [paper]*
- *Instruction Tuning for Large Language Models: A Survey (S Zhang et al., Sep 2023) [paper]*

Techniques & Models & Datasets

OpenAI

- *[GPT 1] Improving Language Understanding by Generative Pre-Training (A Radford et al., OpenAI, Jun 2018)[paper]*
- *[GPT 2] Language models are unsupervised multitask learners (A Radford et al., OpenAI, Feb 2019) [paper]*
- *Scaling Laws for Neural Language Models (J Kaplan et al., OpenAI, Jan 2020) [paper]*
- *[GPT 3] Language Models are Few-Shot Learners (TB Brown et al., OpenAI, May 2020, NIPS 2020) [paper]*
- *[Codex] Evaluating Large Language Models Trained on Code (M Chen et al, OpenAI, Jul 2021) [paper]*
- *Learning to summarize from human feedback (N Stiennon et al., OpenAI, Feb 2022, NIPS 2020) [paper]*
- *[InstructGPT] Training language models to follow instructions with human feedback (L Ouyang et al., OpenAI, Mar 2022, NIPS 2022) [paper]*
- *[GPT 4] GPT-4 Technical Report (OpenAI, Mar 2023) [paper]*

Google Research & DeepMind

- *[Prompt Tuning] The Power of Scale for Parameter-Efficient Prompt Tuning (B Lester et al., Google Research, Sep 2021, EMNLP 2021) [paper]*
- *[CoT] Chain-of-Thought Prompting Elicits Reasoning in Large Language Models (J Wei et al., Google Research, v1- Jan 2022, NIPS 2022) [paper]*
- *[Chinchilla] Training Compute-Optimal Large Language Models (J Hoffmann et al., DeepMind, Mar 2022) [paper]*
- *[FLAN] Finetuned Language Models Are Zero-Shot Learners (J Wei et al., Google Research, v1-Sep 2021, ICLR 2021) [paper][github]*
- *[FLAN 2022] The Flan Collection: Designing Data and Methods for Effective Instruction Tuning (S Longre et al., Google Research, Feb 2023) [paper][github]*
- *[PaLM]: Scaling Language Modeling with Pathways (A Chowdhery et al., Google Research, v1- Apr 2022, JMLR 2023) [paper]*
- *[FLAN PaLM] Scaling Instruction-Finetuned Language Models (HW Chung et al., Google, v1-Oct 2022) [paper]*

Meta AI

- *[RAG] Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (P Lewis et al., v1-May 2020, Facebook AI, NIPS 2020) [paper]*
- *[LLaMA]: Open and Efficient Foundation Language Models (Touvron et al., MetaAI, Feb 2023) [paper]*

THE UNIVERSITY OF SYDNEY

References

Techniques & Models & Datasets (Cont.)

Microsoft

- *[LoRA] Low-Rank Adaptation of Large Language Models (EJ Hu et al., Microsoft, Jun 2021, ICLR 2021) [paper]*
- *[GPT-4-LLM] Instruction Tuning with GPT-4 (B Peng et al., Microsoft Research, Apr 2023) [paper]*
- *Sparks of Artificial General Intelligence: Early experiments with GPT-4 (S Bubeck et al., Microsoft Research, Apr 2023) [paper]*

Various Institutions

- *Large Language Models are Zero-Shot Reasoners (T Kojima et al., U of Tokyo, v1-May 2022, NIPS 2022) [paper]*
- *Automatic Chain of Thought Prompting in Large Language Models (Z Zhang et al., Oct 2022, ICLR 2022) [paper]*
- *[Prefix Tuning] Prefix-Tuning: Optimizing Continuous Prompts for Generation (XL Li and P Liang, Stanford U, Jan 2021, ACL 2021) [paper]*
- *[QLoRA] Efficient Finetuning of Quantized LLMs (T Dettmers et al., U of Washington, May 2023) [paper]*
- *Scaling Down to Scale Up: A Guide to Parameter-Efficient Fine-Tuning (V Lialin et al., Mar 2023) [paper]*
- *[Super-Natural Instructions]: Generalization via Declarative Instructions on 1600+ NLP Tasks (Y Wang et al., Apr 2022, EMNLP 2022) [paper]*
- *[Self-Instruct]: Aligning Language Models with Self-Generated Instructions (Y Wang et al., May 2023, ACL 2023) [paper]*

Others

- *LLM parameter sizes: https://thelowdown.momentum.asia/the-emergence-of-large-language-models-llms/*
- *A new Moore's Law? https://huggingface.co/blog/large-language-models*
- *LLaMA2 download (Jul 2023) : https://ai.meta.com/llama/*
- *HuggingFace PEFT: https://huggingface.co/docs/peft/index*
- *DeepLearningAI: https://www.deeplearning.ai/short-courses/finetuning-large-language-models/*
- *What is Prompt Tuning? (IBM) : https://www.youtube.com/watch?v=yu27PWzJI_Y*
- *Soft Prompts (Google Research) : https://blog.research.google/2022/02/guiding-frozen-language-models-with.html*
- *Ludwig AI (Low code framework) : https://ludwig.ai/latest/user_guide/llms/finetuning*
- *Efficient Fine-Tuning for Llama-v2-7b (Ludwig): https://www.youtube.com/watch?v=g68qlo9Izf0*
- *Prompt Engineering Guide: https://www.promptingguide.ai*
- *Prompt Engineering Tutorial : https://www.youtube.com/watch?v=_ZvnD73m40o*
- *GPT (prompt) best practices: https://platform.openai.com/docs/guides/gpt-best-practices*
- *What is RAG? (IBM) : https://www.youtube.com/watch?v=T-D1OfcDW1M*
- *RAG (Neo4j) https://neo4j.com/developer-blog/fine-tuning-retrieval-augmented-generation/*
- *Retrieval doc (LangChain) : https://python.langchain.com/docs/modules/data_connection/*

THE UNIVERSITY OF SYDNEY

Thank you!

If you have any questions or need updates,
please feel free to reach out to me.
jean.lee@sydney.edu.au

THE UNIVERSITY OF SYDNEY