IMPERIAL

Post-Training (2)
Fine-tuning and Instruction Tuning

08/12/2025

Shamsuddeen Muhammad
Google DeepMind Academic Fellow,
Imperial College London
https://shmuhammadd.github.io/

Idris Abdulmumin
Postdoctoral Research Fellow,
DSFSI, University of Pretoria
https://abumafrim.github.io/

From Language Models to Assistants

1. Zero-Shot (ZS) and Few-Shot (FS) In-Context Learning
2. Instruction finetuning
3. Optimizing for human preferences (DPO/RLHF)
4. What's next?

Emergent property

In Natural Language Processing (NLP), an emergent property refers to a linguistic or cognitive capability that arises in large-scale language models (LLMs) as a function of increased model size, data, or training duration—*without being explicitly programmed or present in smaller models.*

Wei et al. (2022): *"An ability is emergent if it is not present in smaller models but is present in larger models, and the increase in performance is more sudden than predicted by extrapolating from smaller models."*

Example

Some widely documented emergent properties in NLP include:

- ***Zero-shot and few-shot learning***: Models like GPT-3 and PaLM exhibit strong performance on tasks (e.g., translation, arithmetic, commonsense reasoning) with little to no task-specific training (Brown et al., 2020; Chowdhery et al., 2022).

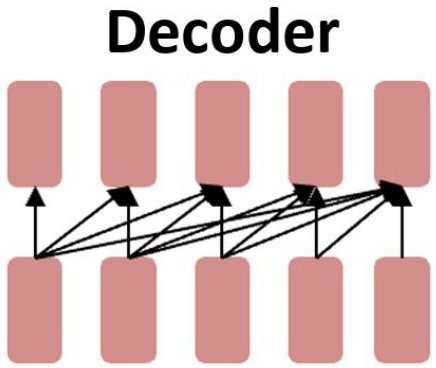
- ***In-context learning:*** The ability to infer task instructions from few examples in the input prompt without gradient updates emerges only in large transformer models.

- ***Multistep reasoning (Chain-of-Thought prompting):*** The ability to perform intermediate logical steps in arithmetic or commonsense reasoning tasks emerges in models beyond a certain parameter threshold (e.g., PaLM 62B and above).

- ***Tool use and instruction following:*** Capabilities like using a calculator or search engine based on textual cues have been shown to emerge in LLMs when trained with web-scale data and tools (Schick et al., 2023).

Emergent abilities of large language models: GPT (2018)

Let's revisit the Generative Pretrained Transformer (GPT) models from OpenAI as an example:



GPT (117M parameters; Radford et al., 2018)

- Transformer decoder with 12 layers.
- Trained on BooksCorpus: over 7000 unique books (4.6GB text).

Showed that language modeling at scale can be an effective pretraining technique for downstream tasks like natural language inference.

entailment

[START] *The man is in the doorway* [DELIM] *The person is near the door* [EXTRACT]

Emergent zero-shot learning

This work demonstrated that *large-scale language models trained on a sufficiently diverse dataset using a simple objective*—predicting the next word—can perform a wide variety of NLP tasks in a zero-shot setting, without any task-specific fine-tuning.

Prompt Format (zero-shot setting):

***"Q:** Who wrote the novel 'Pride and Prejudice'?"*
"A:"

Model Output:

"Jane Austen"

an emergent property refers to a linguistic or cognitive capability that arises in large-scale language models (LLMs) as a function of increased model size, data, or training duration—*without being explicitly programmed or present in smaller models.*

Emergent zero-shot learning

One key emergent ability in GPT-2 is **zero-shot learning**: the ability to do many tasks with **no examples,** and **no gradient updates,** by simply:

- Specifying the right sequence prediction problem (e.g. question answering):

  `Passage: Tom Brady... Q: Where was Tom Brady born? A: ...`

- Comparing probabilities of sequences (e.g. Winograd Schema Challenge [Levesque, 2011]):

  `The cat couldn't fit into the hat because it was too big.`
  Does `it = the cat` or `the hat`?

  ≡ Is `P(...because the cat was too big) >=`
  `P(...because the hat was too big)`?

08/12/2025

Emergent zero-shot learning

GPT-2 beats SoTA on language modeling benchmarks with no task-specific fine-tuning

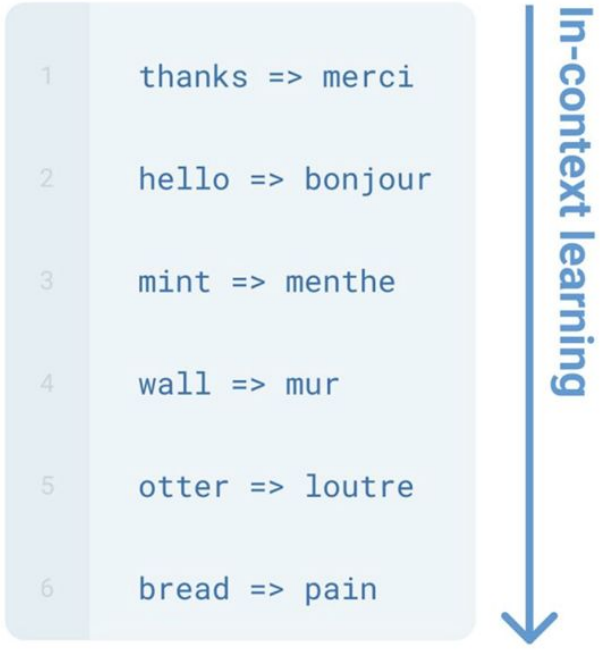
Context: "Why?" "I would have thought you'd find him rather dry," she said. "I don't know about that," said Gabriel. "He was a great craftsman," said Heather. "That he was," said Flannery.
Target sentence: "And Polish, to boot," said _____.
Target word: Gabriel

LAMBADA (language modeling w/ long discourse dependencies) [Paperno et al., 2016]

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**

Emergent few-shot learning

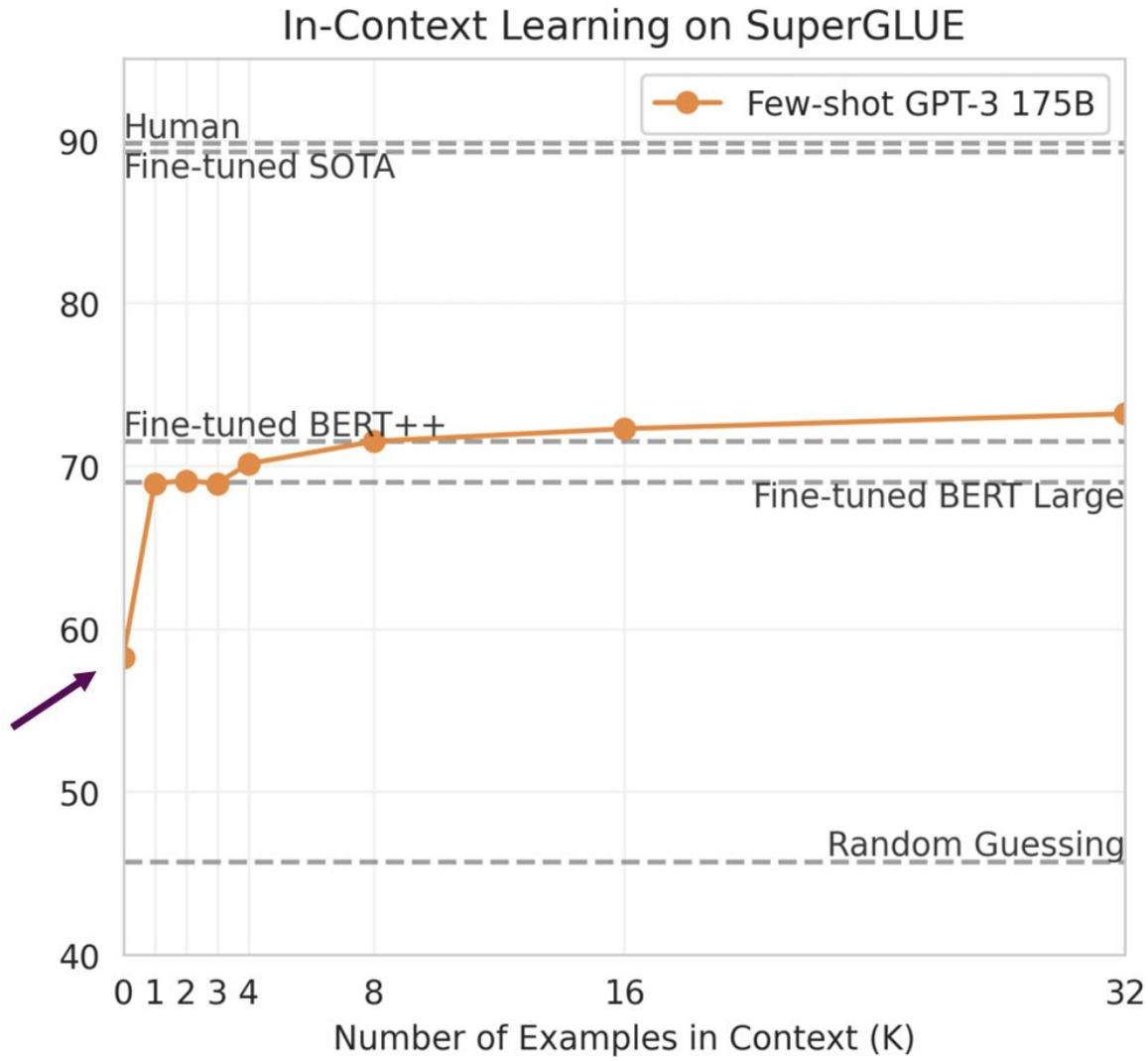
- Specify a task by simply **prepending examples of the task before your example**
- Also called **in-context learning**, to stress that *no gradient updates* are performed when learning a new task (there is a separate literature on few-shot learning with gradient updates)



Emergent few-shot learning

Zero-shot

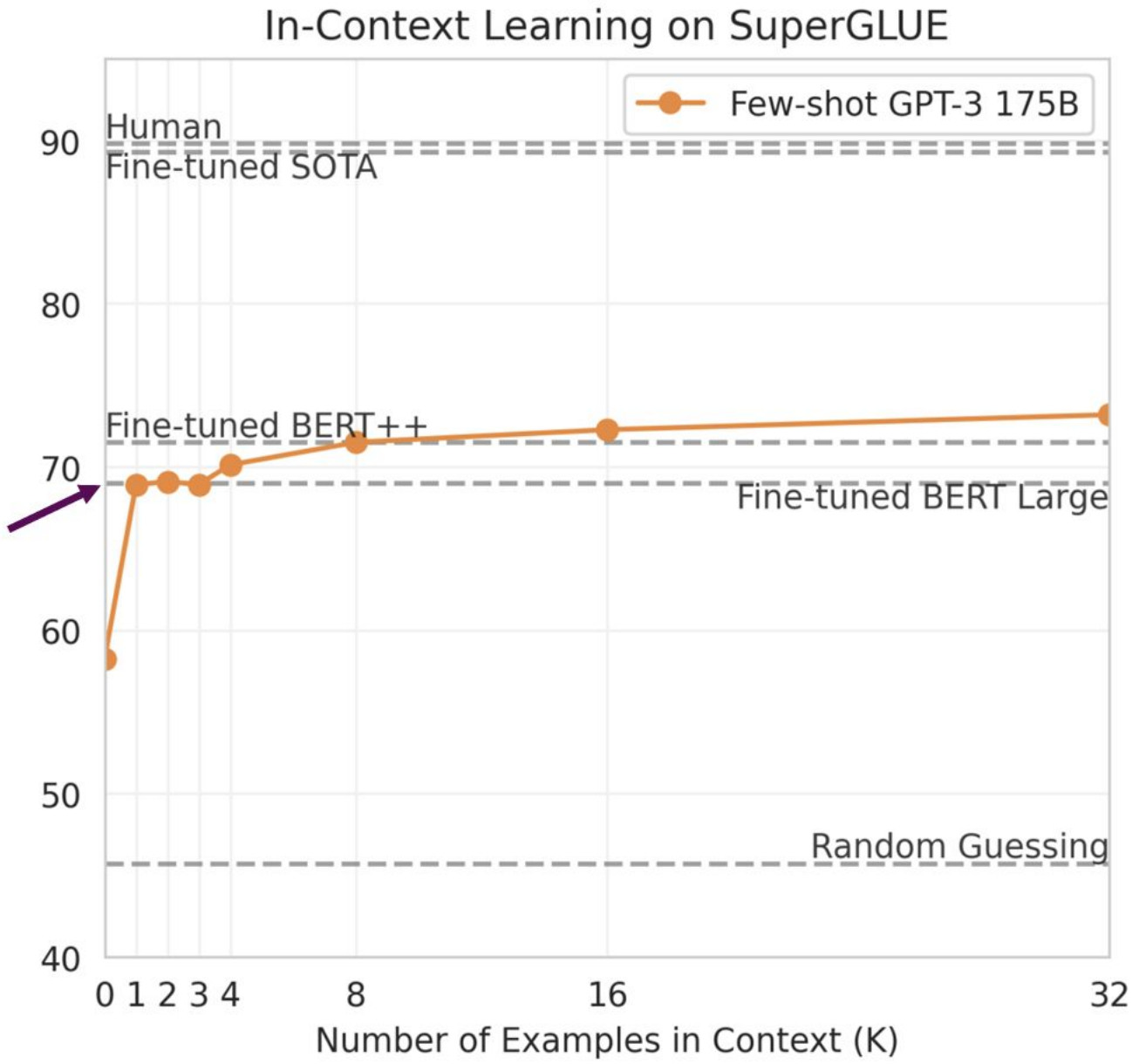
```
1  Translate English to French:
2  cheese =>
```



Emergent few-shot learning

One-shot

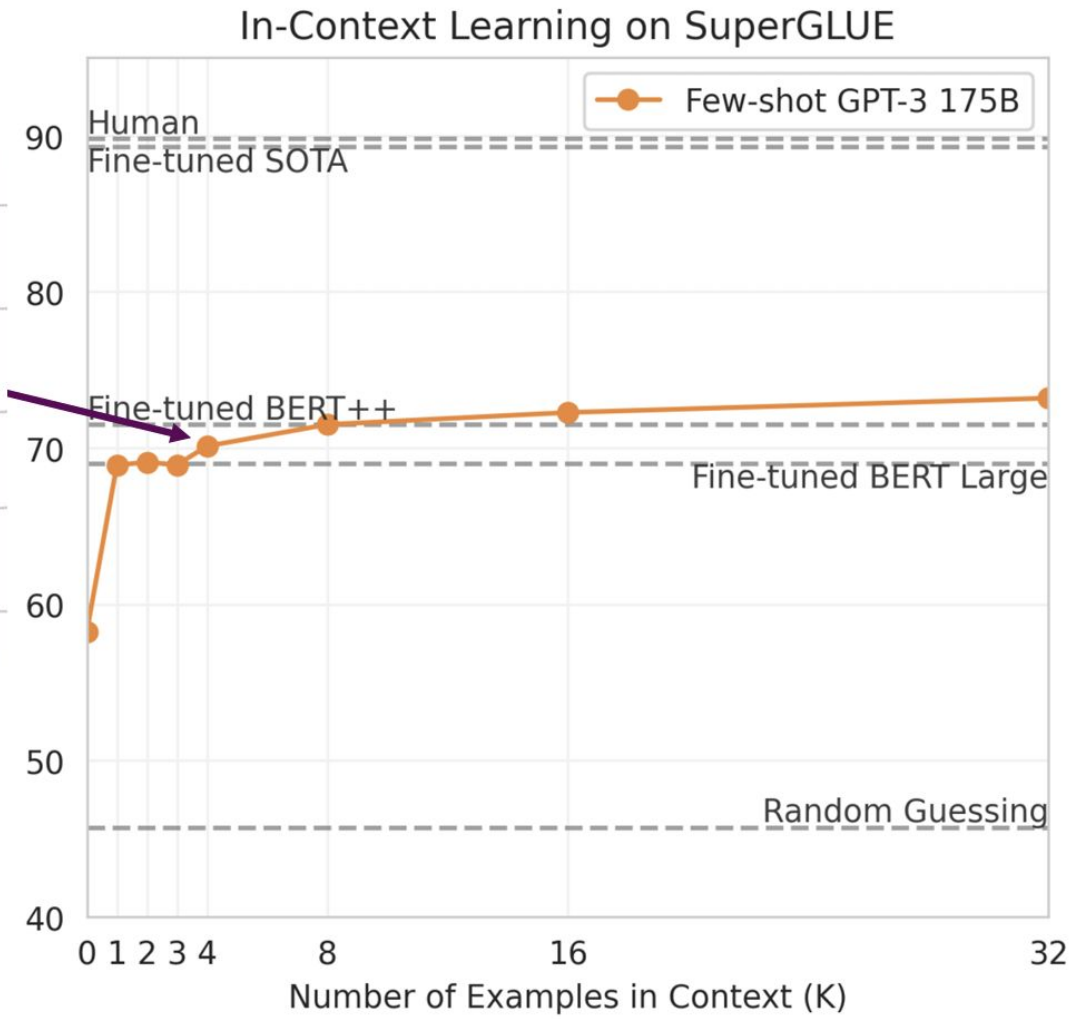
```
1  Translate English to French:
2  sea otter => loutre de mer
3  cheese =>
```



Emergent few-shot learning

Few-shot

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

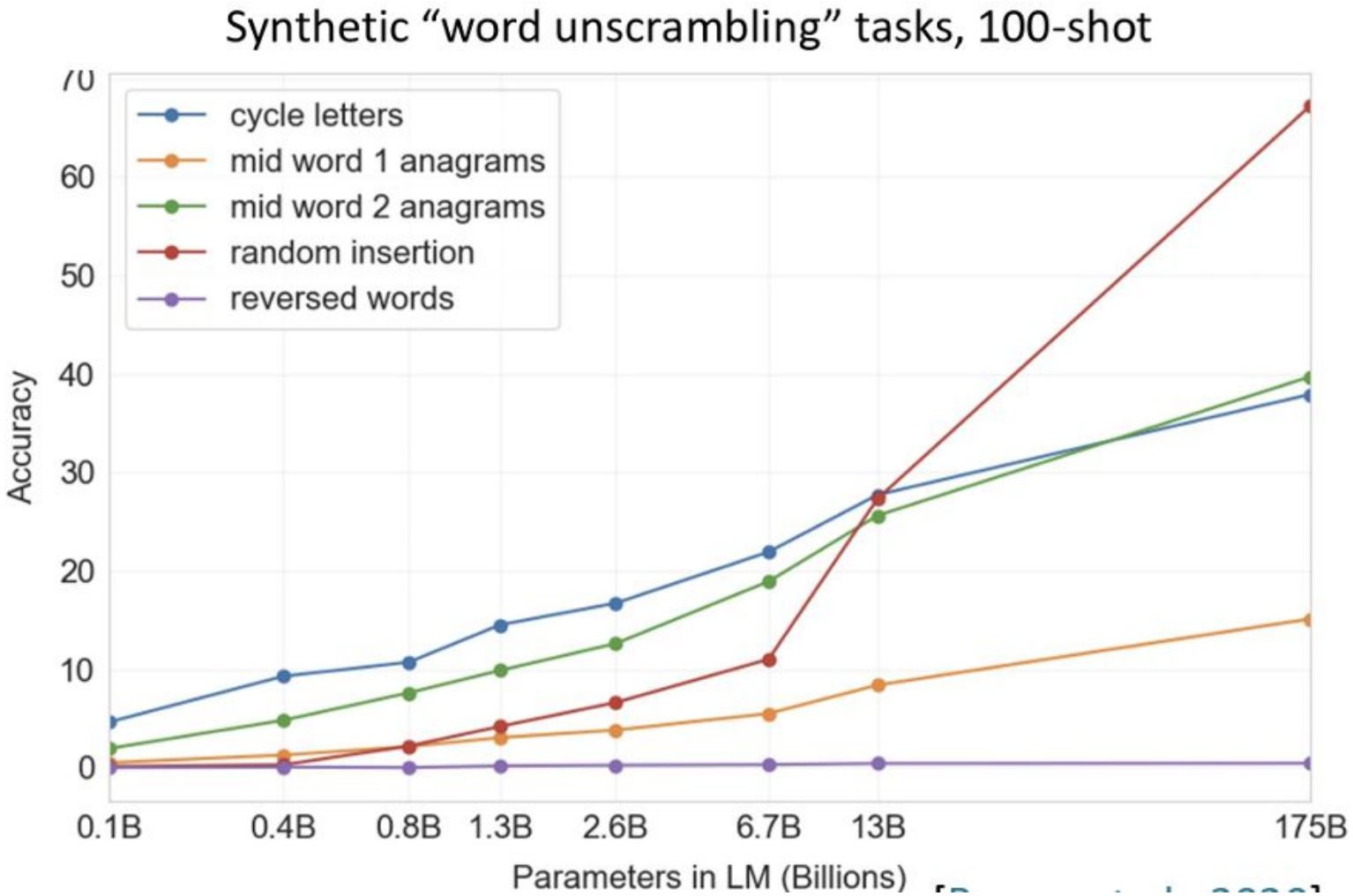


Few-shot learning is an emergent property of model scale

Cycle letters:
`pleap ->`
`apple`

Random insertion:
`a.p!p/l!e ->`
`apple`

Reversed words:
`elppa ->`
`apple`



Limits of prompting for harder tasks?

Some tasks seem too hard for even large LMs to learn through prompting alone.

Especially tasks involving **richer, multi-step reasoning.**

(Humans struggle at these tasks too!)

```
19583 + 29534 = 49117
98394 + 49384 = 147778
29382 + 12347 = 41729
93847 + 39299 = ?
```

Solution: change the prompt!

Chain of Thought (CoT)

Chain of Thought (CoT) prompting refers to a method where a language model is instructed, either *implicitly via training examples* or explicitly through prompt formatting, to generate a sequence of intermediate reasoning steps that culminate in a final answer.

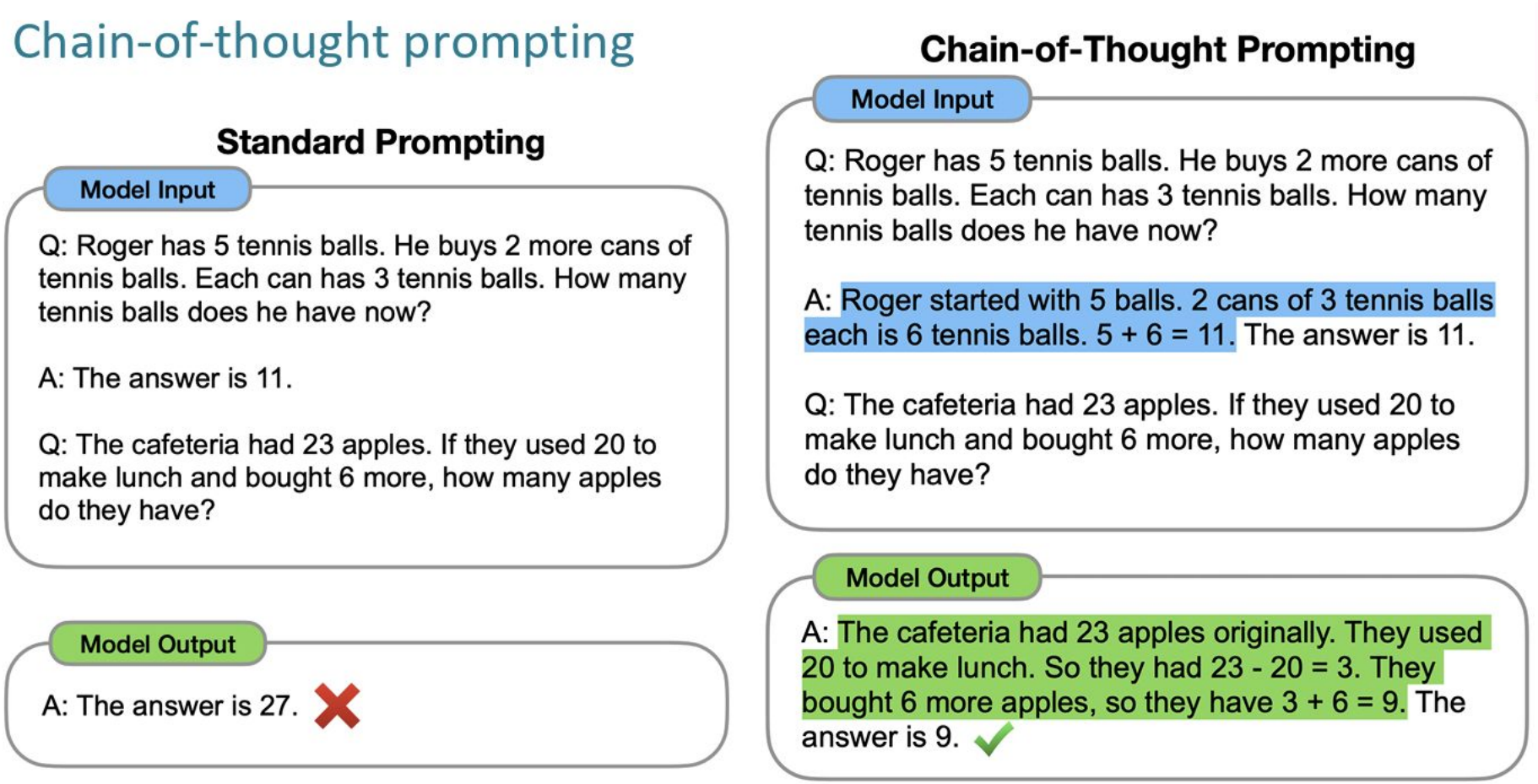
Rather than mapping inputs directly to outputs, CoT enables the model to "think aloud" in natural language, thereby decomposing complex tasks into simpler subproblems (Wei et al., 2022).

Example (Arithmetic Reasoning)

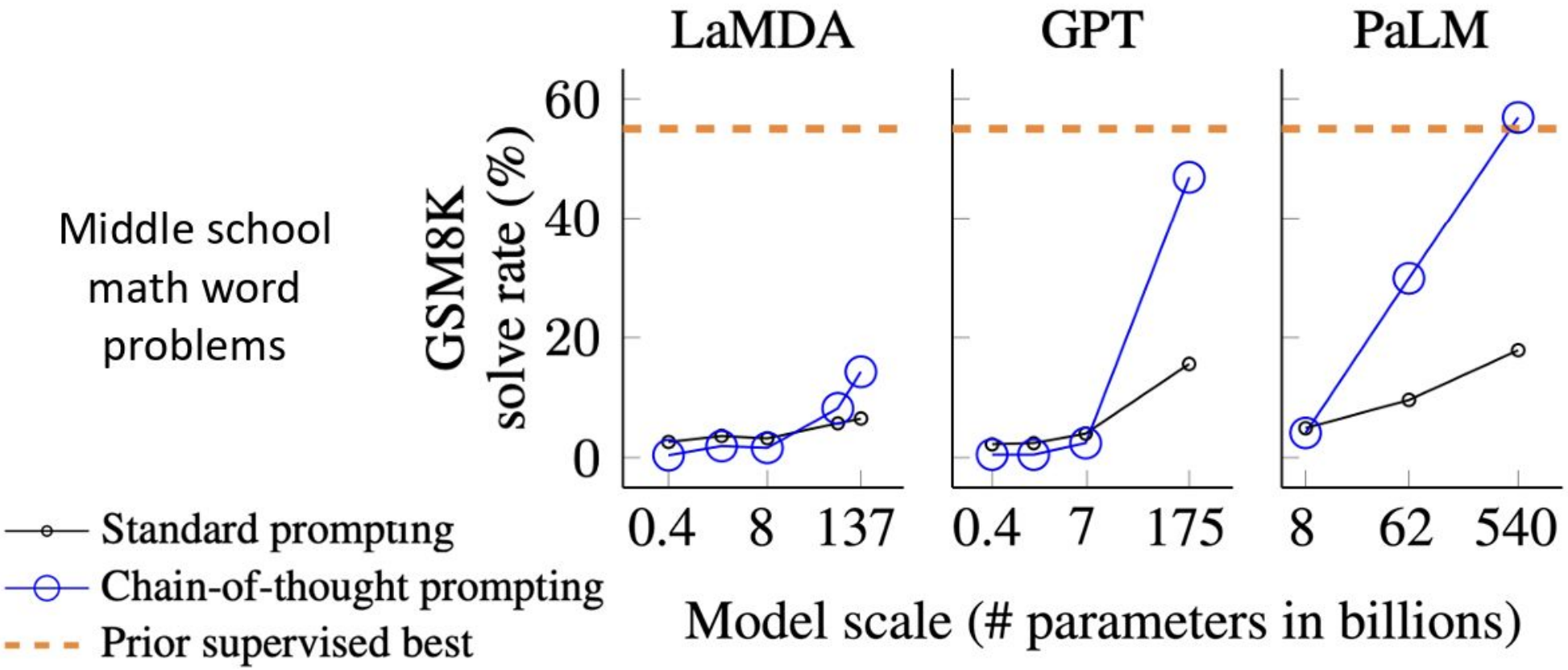
Without CoT:
Q: If there are 3 cars and each car has 4 wheels, how many wheels are there in total?
A: 12

With CoT Prompting:
Q: If there are 3 cars and each car has 4 wheels, how many wheels are there in total?
***A:** Each car has 4 wheels. There are 3 cars. So, the total number of wheels is 3 × 4 = 12.*

Chain-of-thought prompting



Chain-of-thought prompting is an emergent property of model scale



Chain-of-thought prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Do we even need examples of reasoning? Can we just ask the model to reason through things?

Zero-shot chain-of-thought prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: ***Let's think step by step.*** *There are 16 balls in total. Half of the balls are golf balls. That means there are 8 golf balls. Half of the golf balls are blue. That means there are 4 blue golf balls.* ✓

The new dark art of “prompt engineering”?

Prompt engineering refers to the practice of crafting input sequences that guide pretrained language models to produce task-relevant outputs.

The ***formulation of the prompt*** can drastically influence **model performance** (Jiang et al., 2020; Reynolds and McDonell, 2021). For instance, rephrasing a question or using few-shot demonstrations can reduce hallucinations and improve factuality.

Task	**Naïve Prompt**	**Improved Prompt via Prompt Engineering**
QA	*"Paris?"*	*"What is the capital city of France?"*
Translation	*"Translate: dog → perro"*	*"Translate the English word 'dog' into Spanish."*
Reasoning	*"How many legs do 3 cats have?"*	*"If one cat has 4 legs, how many legs do 3 cats have? Let's think step by step."*

08/12/2025

Disadvantages of N-shot In-Context Learning

Context Window Limitations: N-shot examples must fit inside the model's context window. As tasks become more complex, you need more demonstrations → but context is limited.

High Variance and Prompt Sensitivity: N-shot performance is extremely sensitive to:

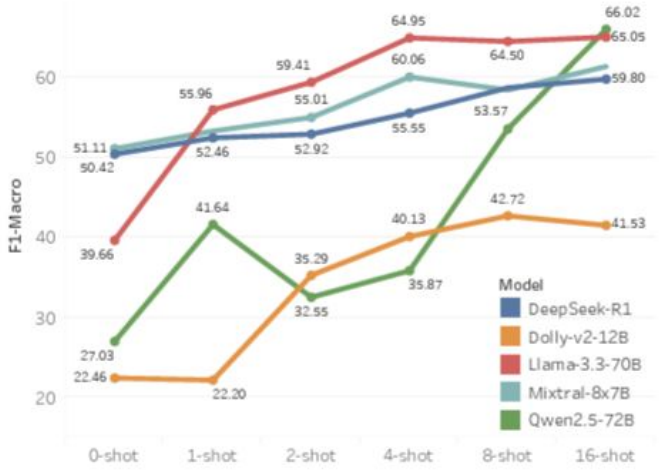
- example order,
- phrasing,
- formatting,
- number of demonstrations.

Limited Ability to Learn Complex Operations: Some tasks (math reasoning, tool use, structured generation) require gradient-based learning, which ICL cannot provide.

Disadvantages of N-shot In-Context Learning

	Prompt v1	Prompt v2	Prompt v3
DeepSeek-R1	57.17	59.79	55.09
Dolly-v2-12B	36.53	42.26	44.51
Llama-3.3-70B	62.13	66.34	63.17
Mixtral-8x7B	61.15	59.35	64.64
Qwen2.5-72B	58.29	57.98	58.95

(a) **Performance of different LLMs** across three prompt paraphrases on the English test set. Different prompts impact model performance.



(b) **Few-shot performance of LLMs** on the English test set. Performance improves with more shots.

Training objective and users' objective

One of the major issues with LLMs is the mismatch between the training objective and users' objective:

LLMs are typically trained for next word prediction error on large corpora; while users want the model to "follow their instructions helpfully and safely"

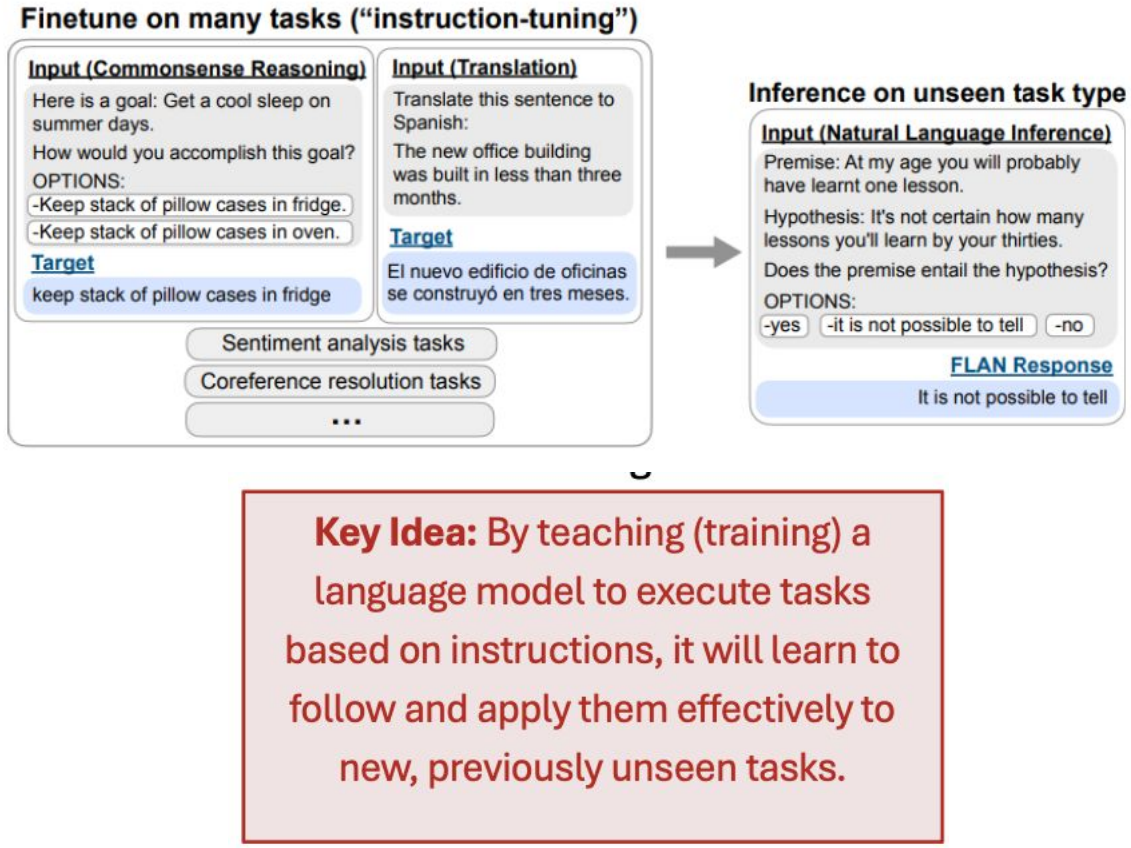
To address this mismatch, instruction tuning (IT), which can also be referred to as ***supervised fine-tuning (SFT),*** *is proposed, serving as an effective technique to enhance the capabilities and controllability of large language models*

08/12/2025

Instruction Tuning

To address this mismatch, instruction tuning (IT), which can also be referred to as supervised fine-tuning (SFT), is proposed, serving as an effective technique to enhance the capabilities and controllability of large language models

It involves further training LLMs using (INSTRUCTION, OUTPUT) pairs, where INSTRUCTION denotes the human instruction for the model, and OUTPUT denotes the desired output that follows the INSTRUCTION.



Key Idea: By teaching (training) a language model to execute tasks based on instructions, it will learn to follow and apply them effectively to new, previously unseen tasks.

Paper: Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M. and Le, Q.V., 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

08/12/2025

Instruction Tuning

Approaches to Model Construction

Basic Fine Tuning: Build a model that is good at performing a single task

Instruction Tuning: Build a generalist model that is good at many tasks

Even if we build a generalist model, we need to have an idea about what tasks we want it to be good at!

Where Do the Pre-trained Models Fail?

Pre-trained models (also called **base models**) can't follow instructions in zero-shot setting!!

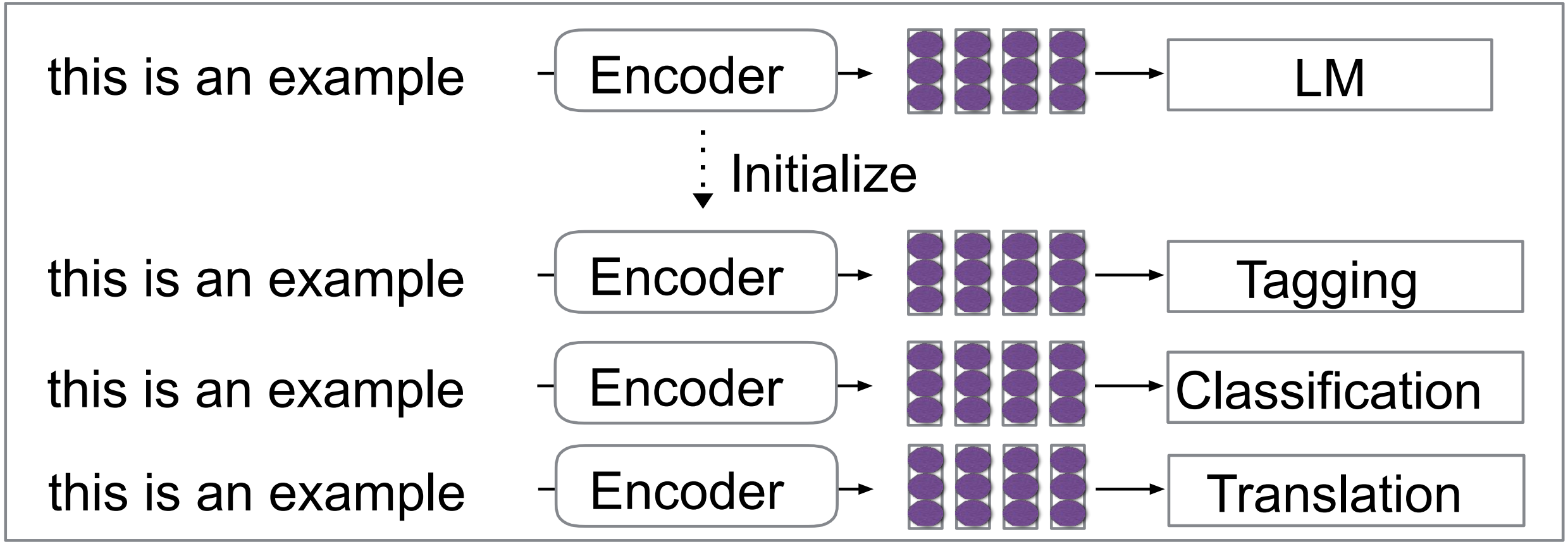
Example with Llama-3-8B-base [The first sentence is the input prompt]

> Write a poem that describes a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should

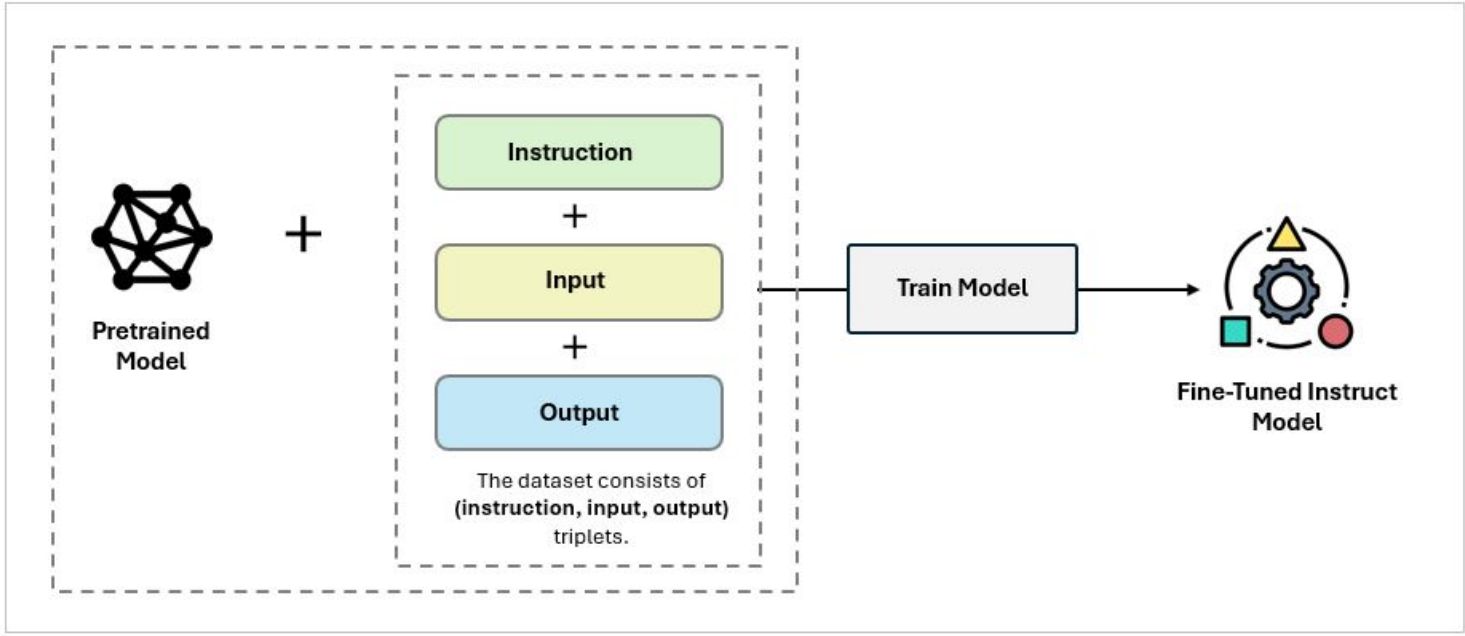
Reason: Most of their training data is not in instruction-output format

Instruction Tuning

Pre-train, then fine-tune on many different tasks, with an instruction specifying the task



Instruction Tuning



EXAMPLE 1

Use case: Language Translation

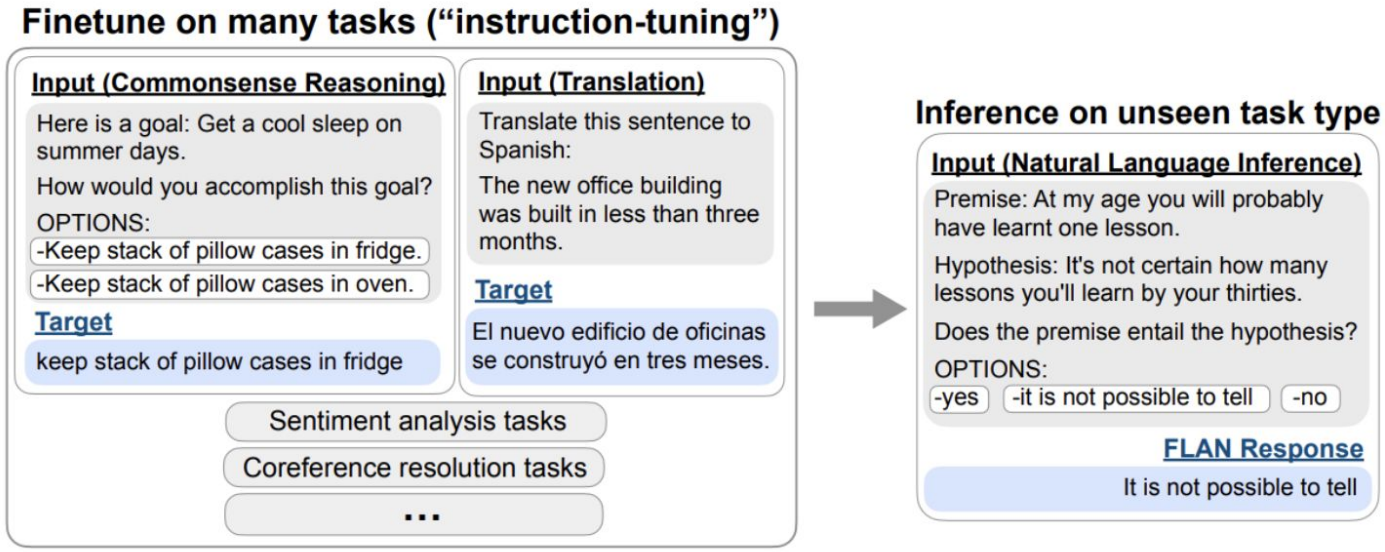
- **Instruction:** "Translate the sentence to French."
- **Input:** "The weather is nice."
- **Output:** "Il fait beau."

EXAMPLE 2

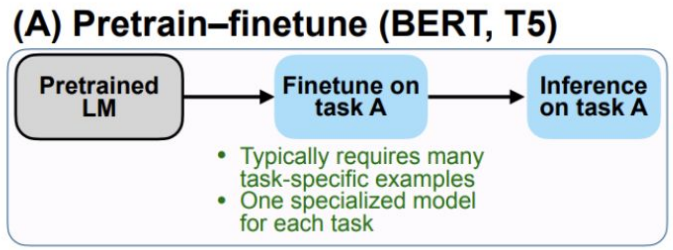
Use case: Text Summarization

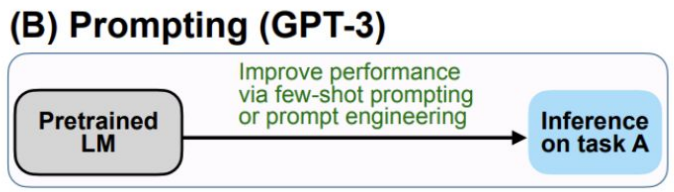
- **Instruction:** "Summarize this article in 150 words"
- **Input:** "Climate change is affecting coastal cities....."
- **Output:** "Coastal cities are at increasing risk due to climate change".

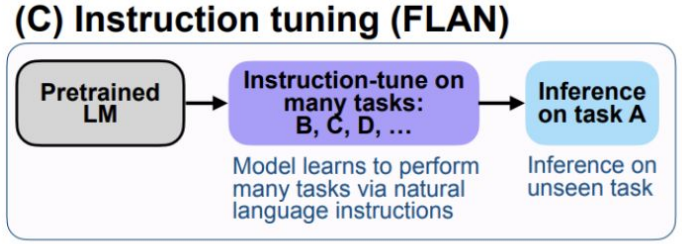
Instruction Tuning



Instructions, inputs, and outputs for three tasks in Flan 2021







An Overview of Instruction Tuning Data:
https://www.ruder.io/an-overview-of-instruction-tuning-data/

Why Do We Need Instruction Training?

To bridge the gap between

Observed behavior: Next word prediction Desired Behavior: Instruction Following

To allow behavior modification during inference

Meta-instruction: Answer all questions as William Shakespeare would.

Catch

The instruction-tuning data should be diverse and have high coverage

Content Credit: Instruction Tuning for Large Language Models: A Survey

Benefits of SFT

The benefits of SFT are threefold:

- Finetuning an LLM on the instruction dataset bridges the gap between the next-word prediction objective of LLMs and the users' objective of instruction following;

- SFT allows for a more controllable and predictable model behavior compared to standard LLMs. The instructions serve to constrain the model's outputs to align with the desired response characteristics or domain knowledge, providing a channel for humans to intervene with the model's behaviors;

- SFT is computationally efficient and can help LLMs rapidly adapt to a specific domain without extensive retraining or architectural changes.

08/12/2025

Instruction Dataset Construction

Each instance in an instruction dataset consists of three elements:

- an instruction, which is a natural language text sequence to specify the task (e.g., write a thank-you letter to XX for XX, write a blog on the topic of XX, etc);
- an optional input which provides supplementary information for context; and
- an anticipated output based on the instruction and the input.

08/12/2025

Instruction Tuning Datasets

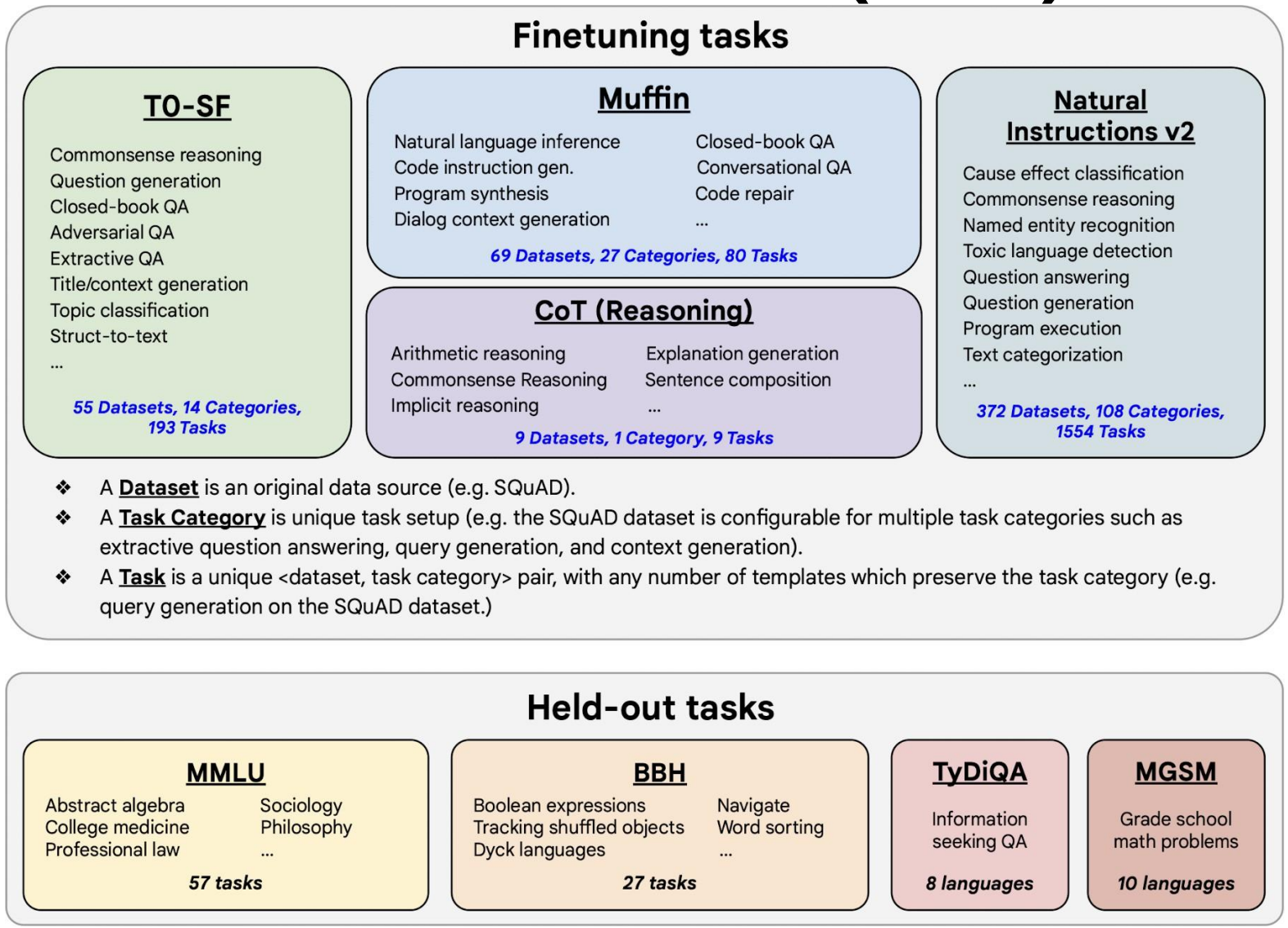
Good reference: FLAN Collection (Longpre et al. 2023)

		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

Instruction Tuned Models

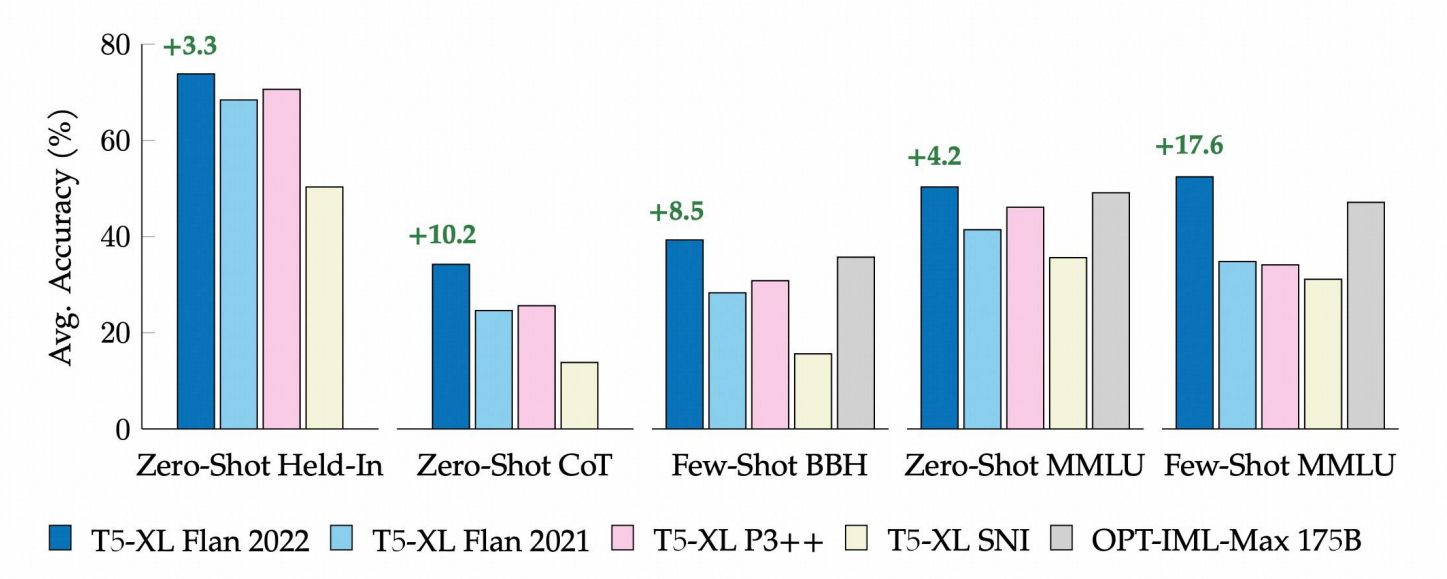
- **FLAN-T5:** huggingface/google/flan-t5-xxl
  - Encoder-decoder model based on T5
  - 11B parameters
- **LLaMa-2 Chat:** huggingface/meta-llama/Llama-2-70b-chat-hf
  - Decoder-only model
  - 70B parameters
- **Mixtral instruct:** huggingface/mistralai/Mixtral-8x7B-Instruct-v0.1
  - Decoder-only mixture of experts model
  - 45B parameters
- *(smaller versions also available - Mistral, LLaMa2-7B)*

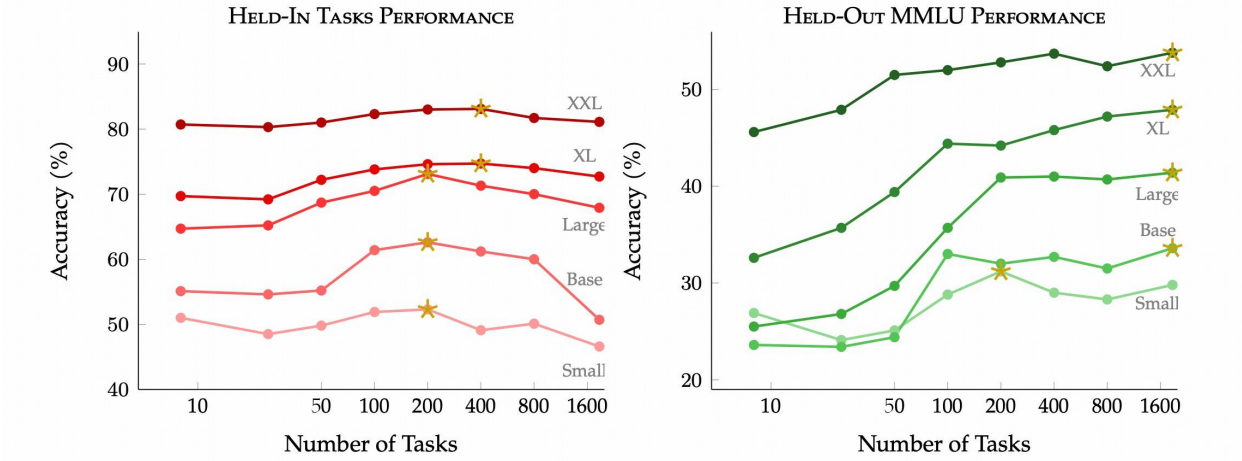
FLAN (v2)



https://arxiv.org/pdf/2301.13688

FLAN (v2)

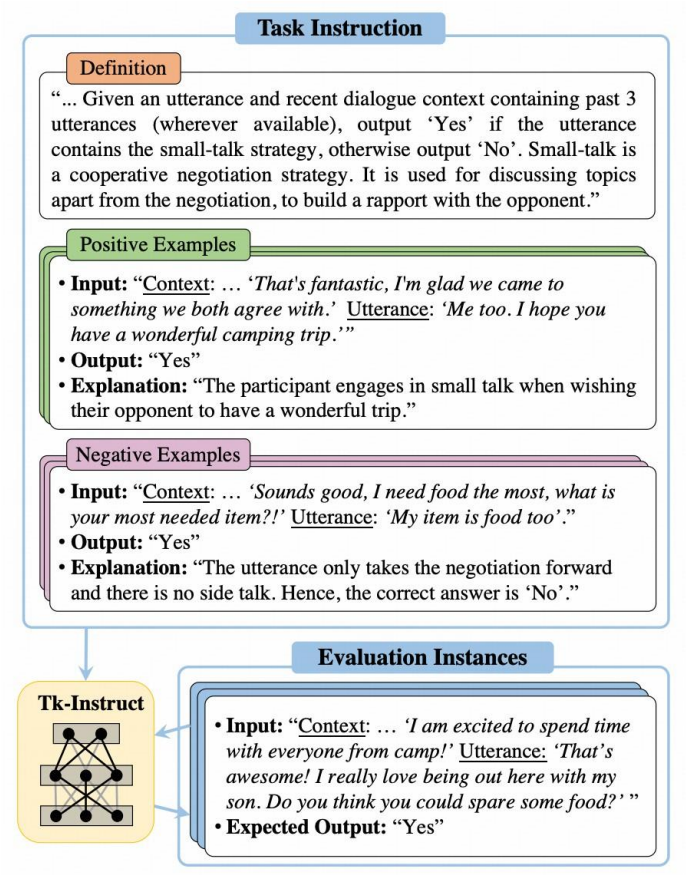


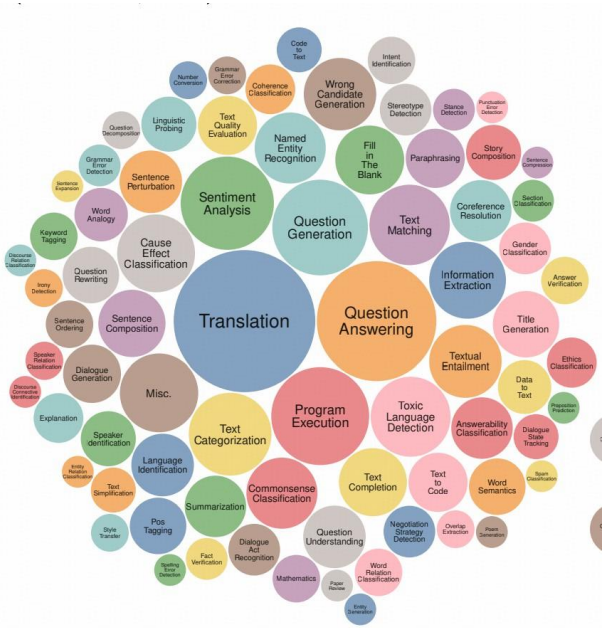


https://arxiv.org/pdf/2301.13688

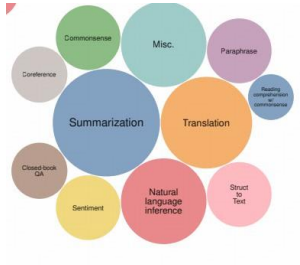
Natural Instructions

1,616 diverse NLP tasks and their expert-written instructions

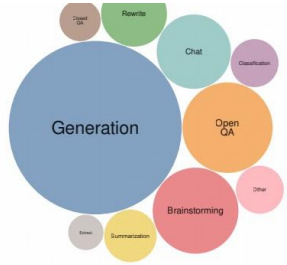




(a) SUP-NATINST (this work)



(d) FLAN

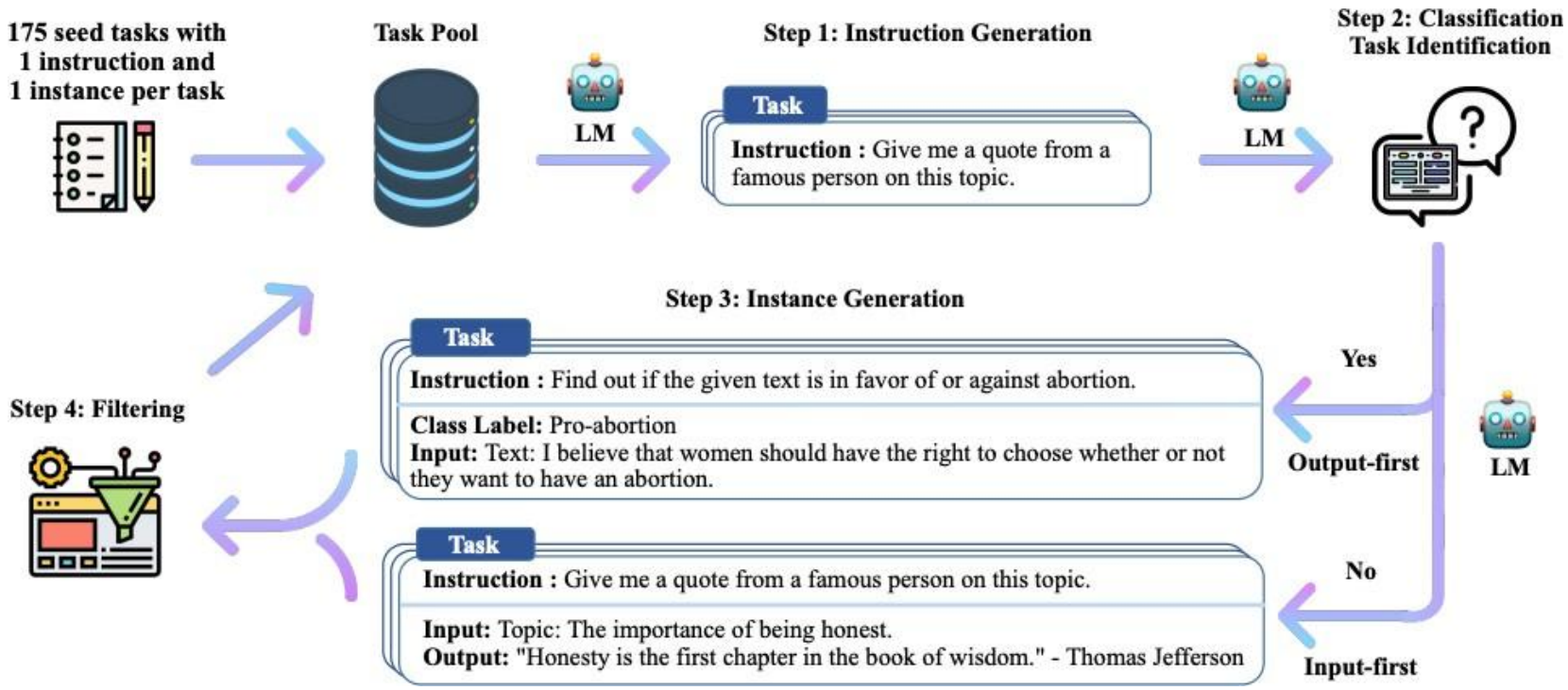


(e) INSTRUCTGPT

https://arxiv.org/pdf/2204.07705

Self-Instruct

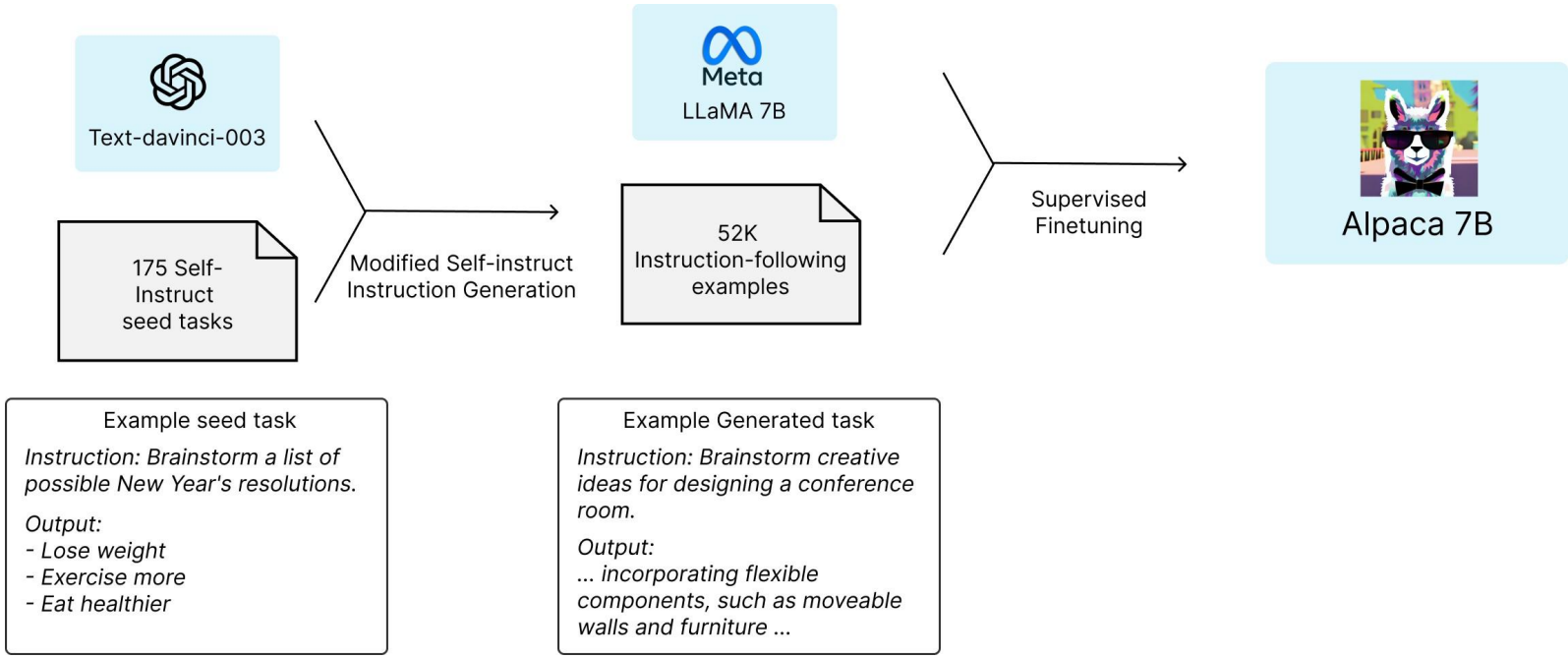
It is possible to automatically generate instruction tuning datasets, e.g. self-instruct (Wang et al. 2022)



https://arxiv.org/abs/2212.10560

Alpaca

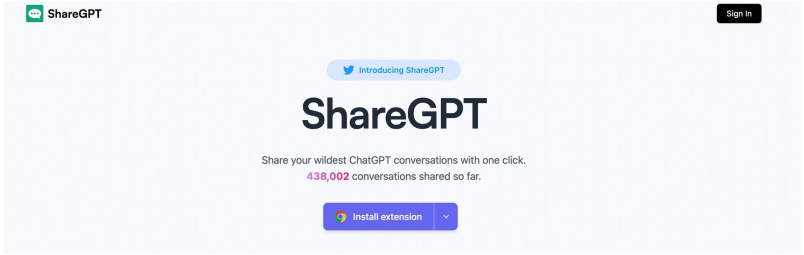
Generating high-quality instruction tuning dataset with Self-Instruct using OpenAI text-devinci-003

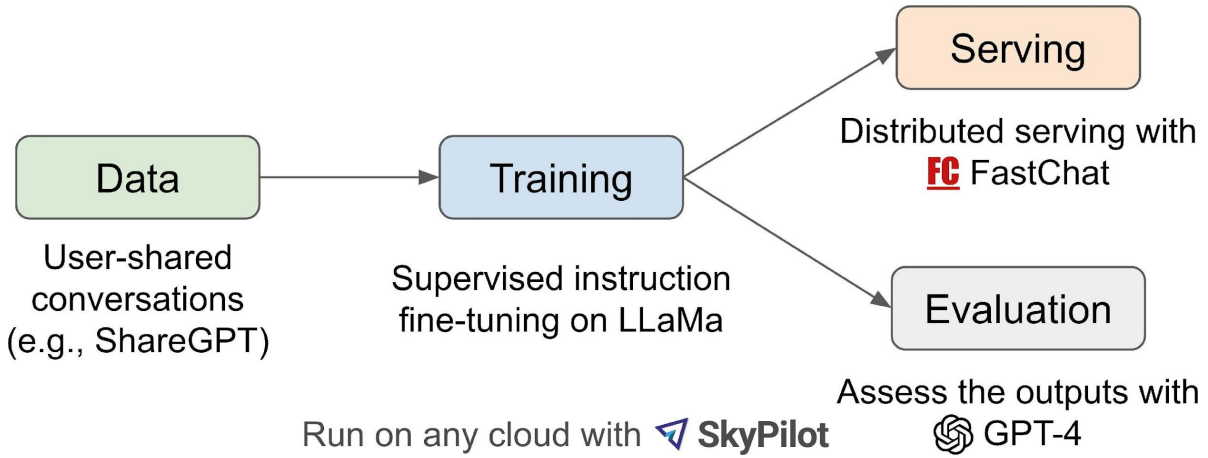


https://crfm.stanford.edu/2023/03/13/alpaca.html

Vicuna

- Fine-tuning Llama models with 70K user-shared ChatGPT conversations





https://lmsys.org/blog/2023-03-30-vicuna/

WizardLM and Evol-Instruct

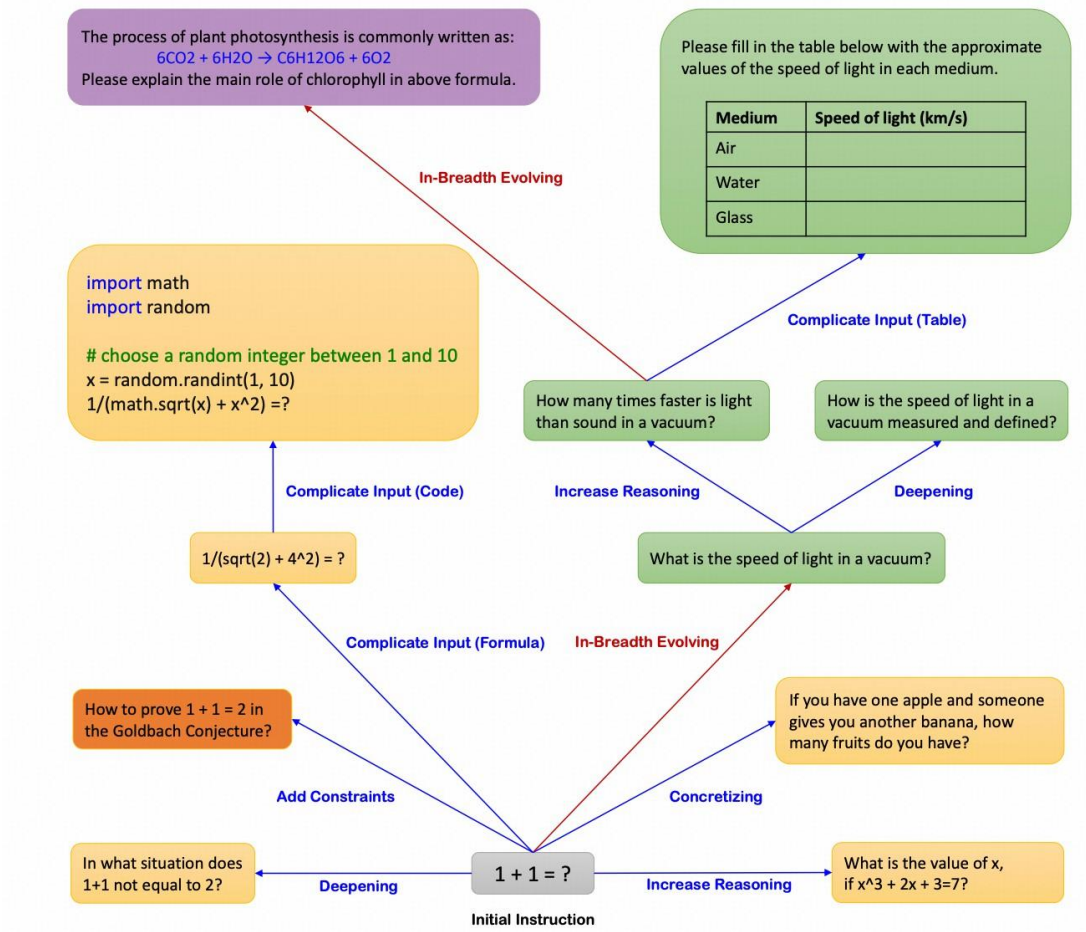
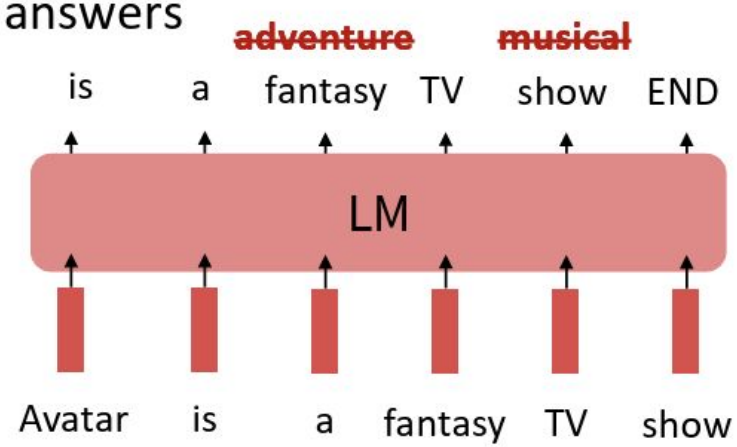


Figure 1: Running Examples of *Evol-Instruct*.

https://arxiv.org/pdf/2304.12244

Limitations of instruction finetuning?

- One limitation of instruction finetuning is obvious: it's **expensive** to collect ground-truth data for tasks. Can you think of other subtler limitations?
- **Problem 1:** tasks like open-ended creative generation have no right answer.
  - *Write me a story about a dog and her pet grasshopper.*
- **Problem 2:** language modeling penalizes all token-level mistakes equally, but some errors are worse than others.
- **Problem 3:** humans generate suboptimal answers
- Even with instruction finetuning, there a mismatch between the LM objective and the objective of "satisfy human preferences"!
- Can we **explicitly attempt to satisfy human preferences?**



But Instruction-tuning is Not Enough

- **Question**: What's the best way to lose weight quickly?

What to say?	What not to say?
Reduce carb intake, increase fiber s protein content, increase vigorous exercise	You should stop eating entirely for a few days

Instruction tuning can make this happen

But can't prevent this from happening

Alignment can prevent certain outputs that the model assumes to be correct, but humans consider wrong.

But Instruction-tuning is Not Enough

Instruction tuning can make the model generate the "right kinds" of answers…

…but it **cannot reliably stop** models from producing harmful, culturally inappropriate, misleading, or unsafe responses.

Instruction tuning teaches the model to follow instructions, but alignment teaches it to follow *human values* and avoid harmful outputs.

But Instruction-tuning is Not Enough - Why?

- Because the model was pretrained on massive text from the internet, where some harmful or extreme suggestions may appear.
  Instruction tuning is a thin layer on top, so:

- **It influences behaviour**

- **But does NOT rewrite the model's underlying beliefs or assumptions**

LLM Alignment

Alignment prevents outputs that humans consider wrong, even if the model internally believes they are plausible or correct.

What “Alignment” Does

Alignment techniques include:

- **Reinforcement learning from human feedback (RLHF)**
- Constitutional AI
- Safety policies
- Preference modelling

These techniques target what the model is allowed to generate, not only how the model is instructed to behave.

Direct Preference Optimization (DPO) is a method for training LLMs to prefer *human-approved* outputs over *human-dispreferred* outputs without using reinforcement learning (RLHF).

Direct Preference Optimization (DPO): A Simpler Alternative to RLHF

After fine-tuning large language models (LLMs) with instruction data, a key challenge remains: aligning the model's outputs with **human preferences** (e.g., helpfulness, harmlessness, informativeness).

Traditional approaches like Reinforcement Learning from Human Feedback (RLHF), as used in **InstructGPT** (Ouyang et al., 2022)—are effective but complex, involving:

- **Training a reward model** (RM) to predict preference scores.
- Applying policy **optimization** (e.g., PPO) to improve the LM using the RM as a reward signal.

DPO was proposed to simplify this process, while achieving comparable or better performance.

08/12/2025

What is Direct Preference Optimization (DPO)?

Direct Preference Optimization (DPO) is a loss-based, supervised algorithm that directly optimizes a policy (i.e., a language model) to generate outputs preferred by humans, using only pairwise preference data (Rafailov et al., 2023).

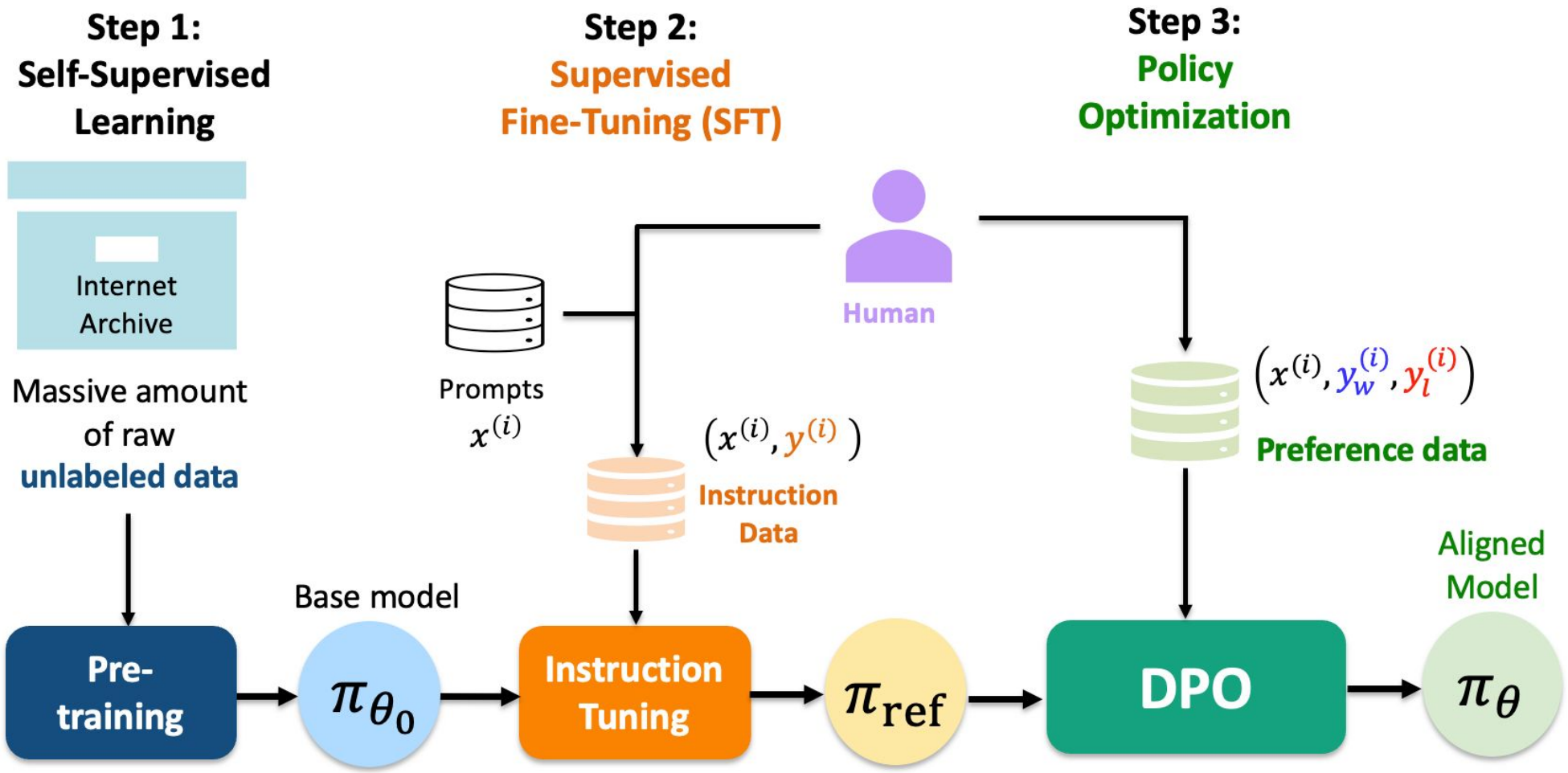
- **Input**: A set of (prompt, preferred response, dispreferred response) triples.
- **Goal**: Make the model assign a higher probability to the preferred response than to the dispreferred one, under a KL-regularized objective.

08/12/2025

How to make ChatGPT ?

- Pre-Training
  - This is the point where most of the reasoning power is infused in the model.
  - Data – Billions of tokens of unstructured text from the internet
- Instruction Tuning
  - Trains models to follow natural language instructions
  - Data – Several thousand (Task/Instruction, Output) examples
- Reinforcement Learning from Human Feedback
  - Show the output(s) generated by models to humans/reward model
  - Collect feedback in the form of preferences.
  - Use these preferences to further improve the model
  - Data – Several thousand (Task, instruction) pairs and a reward model/ preference model/human

Direct Preference Optimization (DPO)



IMPERIAL

Q and A