



香港中文大學(深圳)  
The Chinese University of Hong Kong, Shenzhen

# DDA6307/CSC6052/MDS6002: Natural Language Processing

## Lecture 5: Large Language Models (LLMs)

Spring 2024  
Benyou Wang  
School of Data Science

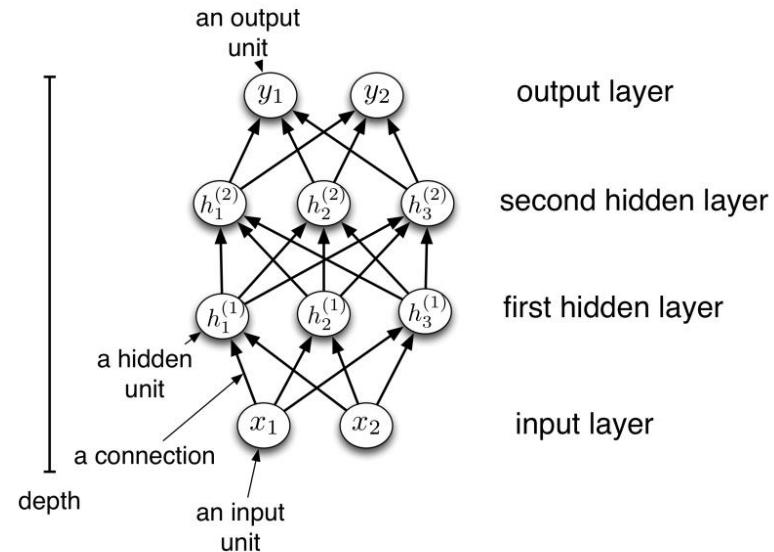
# Recap

# What is in the previous lecture?

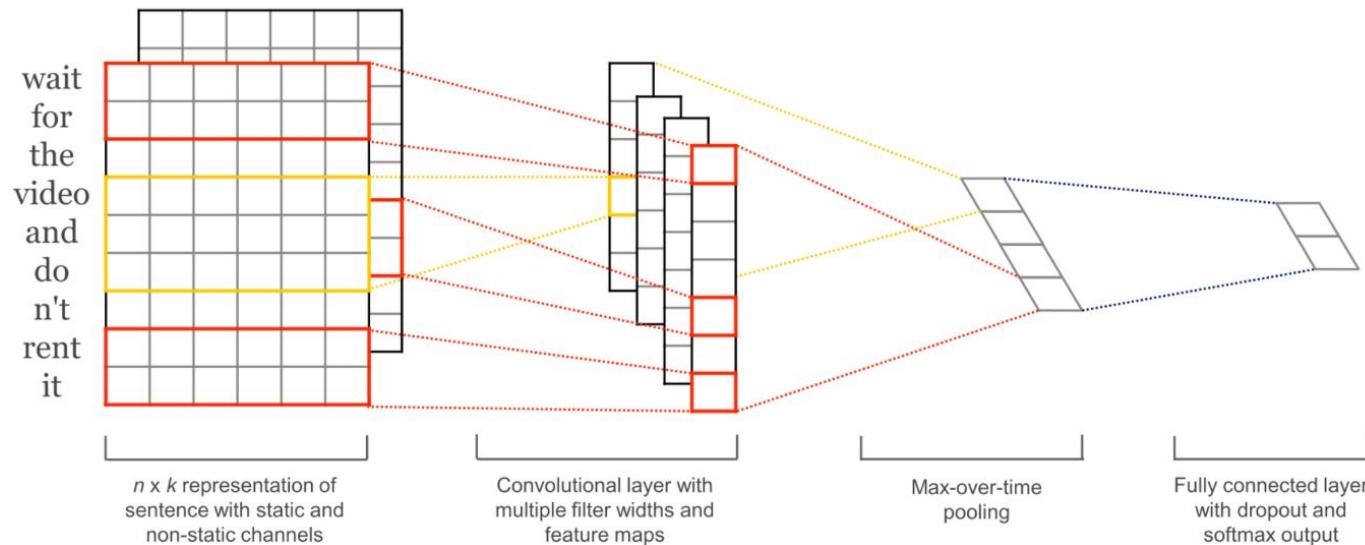
- **MLP**
  - +: Strongest inductive bias: if all words are concated
  - +: Weakest inductive bias: if all words are averaged
  - : The interaction at the token-level is too weak
- **CNN & RNN**
  - +: The interaction at the token-level is slightly better.
    - CNN: Bringing the global token-level interaction to the window-level
      - : Make simplifications, its global dependencies are limited
    - RNN: An ideal method for processing token sequences
      - : Its recursive nature has the problem of disaster forgetting.
- **Transformer**
  - +: Achieve **global dependence** at the **token-level** by **decoupling** token-level interaction and feature-level abstraction into two components, in **SAN** and **FNN**.

# Multilayer Perceptron (MLP)

- ❖ We can connect lots of units together into a directed acyclic graph.
- ❖ This gives a feed-forward neural network. That's in contrast to recurrent neural networks, which can have cycles.  
(We'll talk about those later.)
- ❖ Typically, units are grouped together into layers



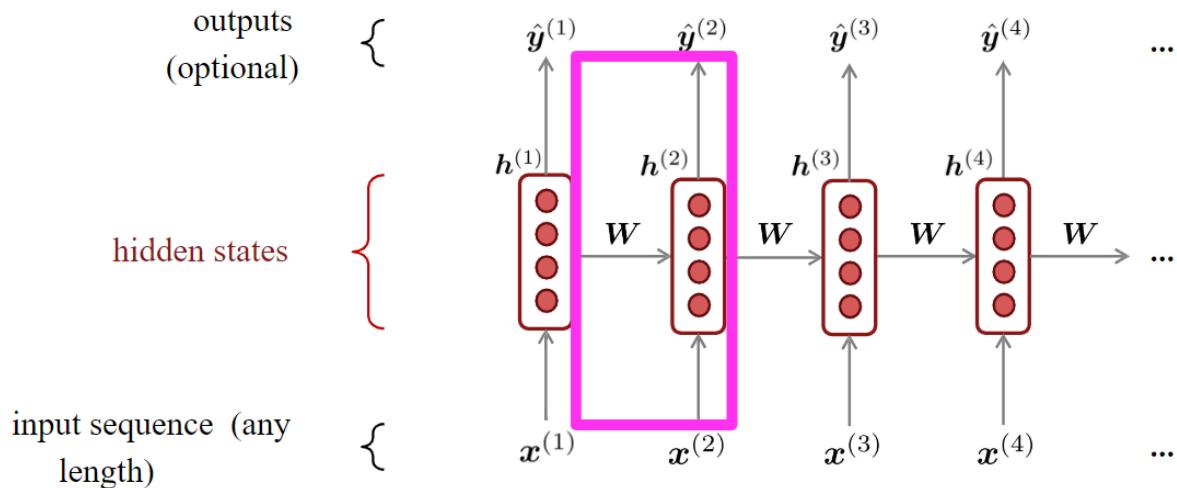
# Convolutional NNs (CNN)



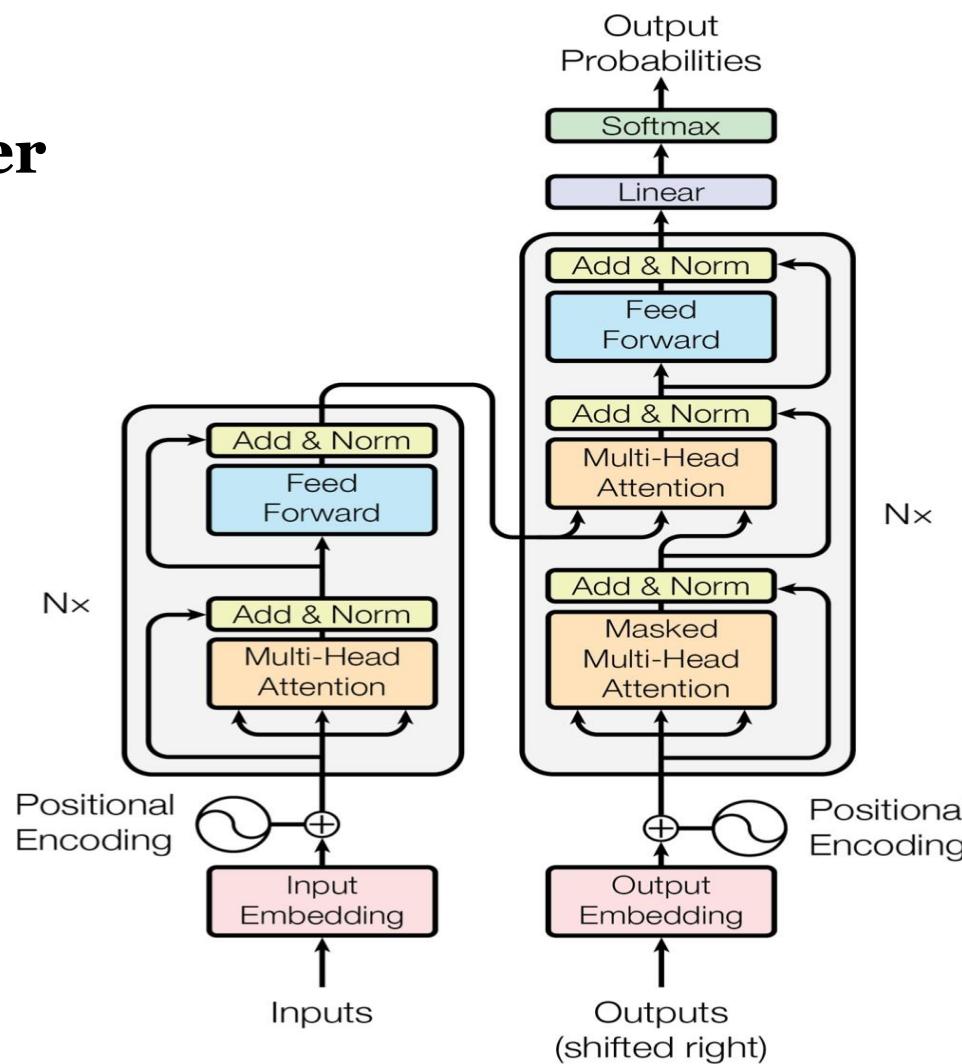
(Kim 2014): Convolutional Neural Networks for Sentence Classification

# Recurrent Neural Networks (RNN)

Core idea: Apply the same weights  $W$  repeatedly



# Transformer



# Today Plan

## 1. What are Large Language Models (LLMs)?

- A view of LLMs.
- The current LLMs, such as ChatGPT and GPT-4.

## 2. Why LLMs?

- The emergent capabilities of LLMs.

## 3. How do we use LLMs?

- Three levels of using LLMs.

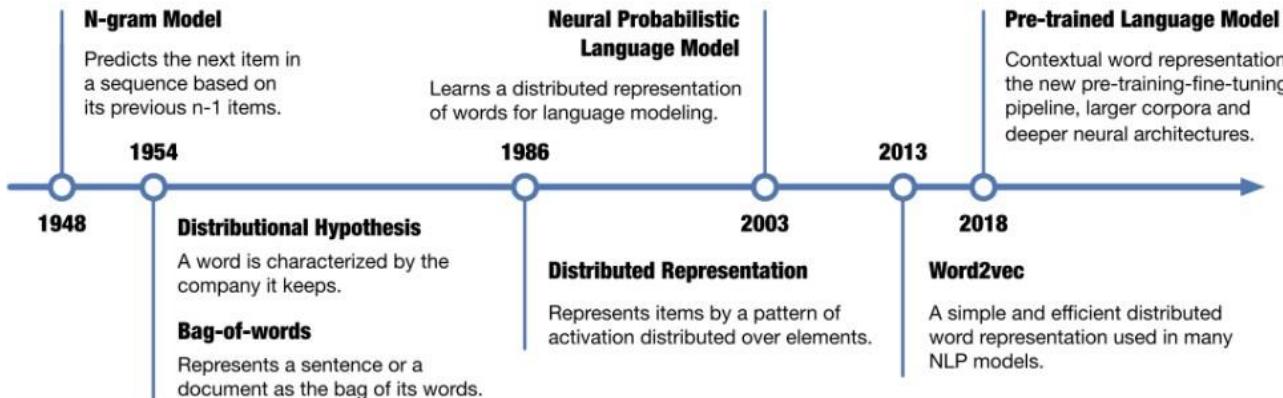


**Today's Tutorial:** Derivation of Backpropagation in Neural Network (by YU, Fei)

# What are Large Language models (LLMs)?

# Background

- language model



Liu et al., Representation Learning for Natural Language Processing, Springer, 2020

# Language models: the narrow Sense

A probabilistic model that assigns a probability to every finite sequence (grammatical or not)

Sentence: “the cat sat on the mat”

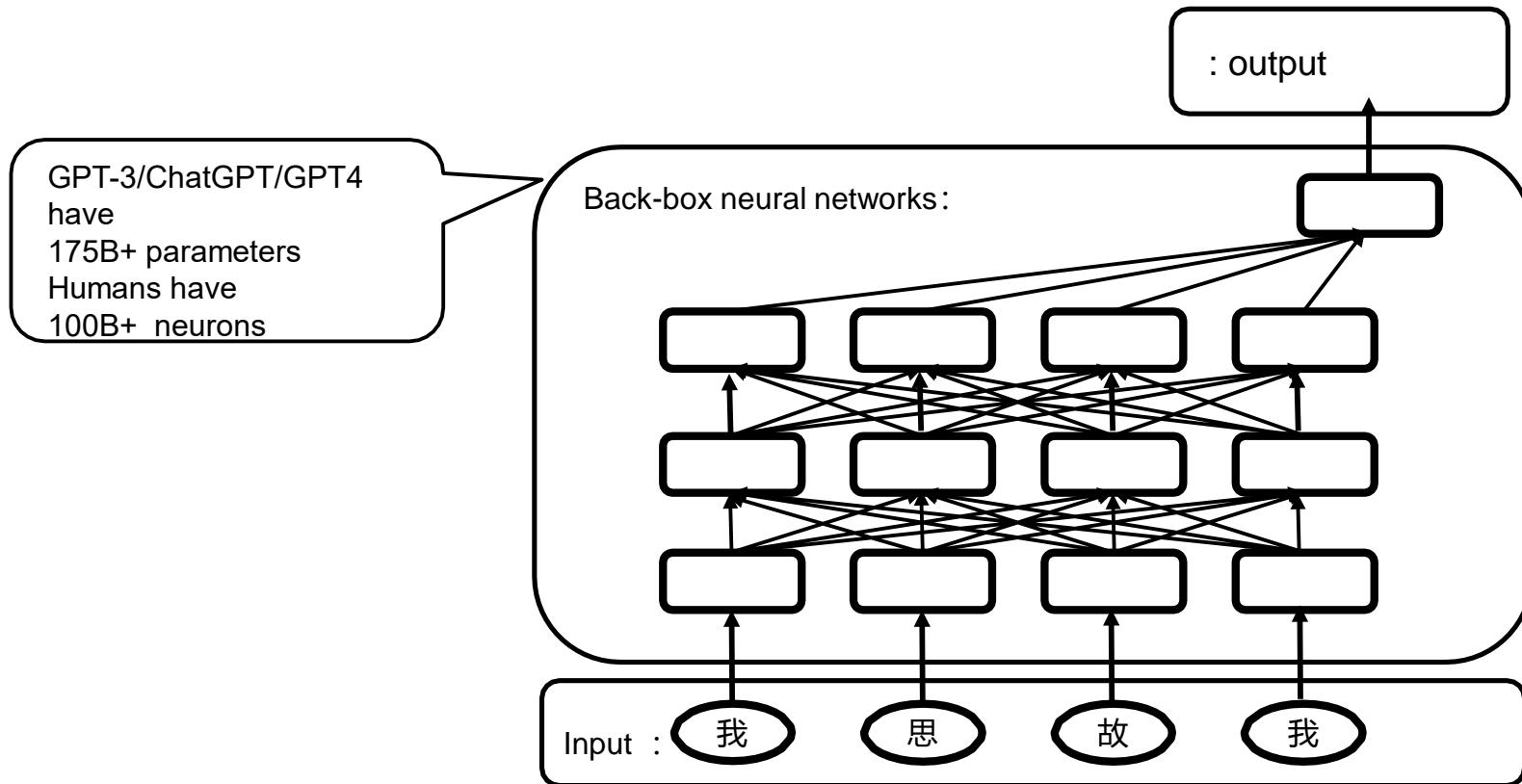
$$\begin{aligned} P(\text{the cat sat on the mat}) &= P(\text{the}) * P(\text{cat}|\text{the}) * P(\text{sat}|\text{the cat}) \\ &\quad * P(\text{on}|\text{the cat sat}) * P(\text{the}|\text{the cat sat on}) \\ &\quad * P(\text{mat}|\text{the cat sat on the}) \end{aligned}$$

Implicit order



GPT-3 still acts in this way but the model is implemented as a very large neural network of 175-billion parameters!

# Language model using neural networks



Think about that:

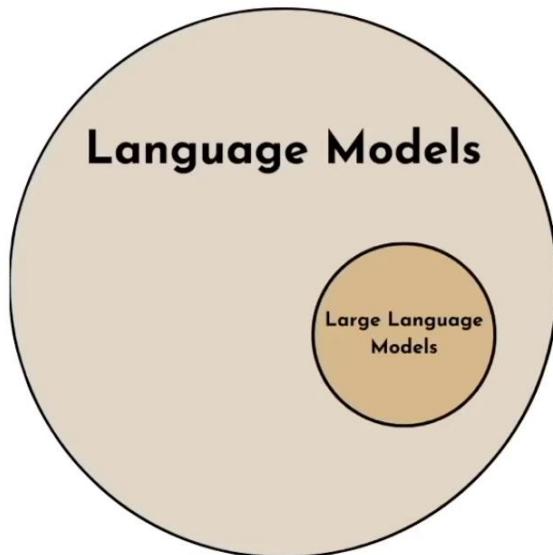
What is the difference between Large Language models (LLMs) and language models?

Just larger?

# What is an LLM: an informal definition

Large, general-purpose language  
models can be pre-trained and then  
fine-tuned for specific purposes

# Zero-shot Learning in LLMs



## Quantitatively

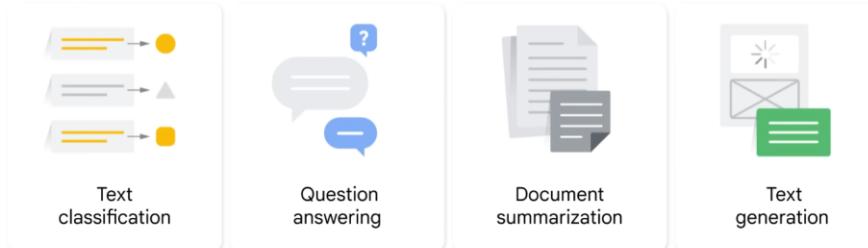
Number of model parameters  
i.e. ~10-100 Billion

## Qualitatively

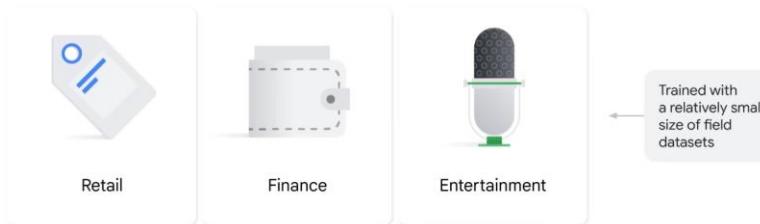
*Emergent properties<sup>[1]</sup>*  
i.e. Zero-shot learning

# What is an LLM (one model for nearly everything)

Large language models are trained to solve  
common language problems, like...



problems in different fields, like...



Also called “foundation model” [1]

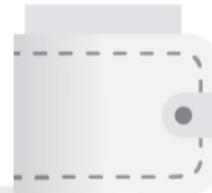
[1] Rishi Bommasani. On the Opportunities and Risks of Foundation Models. <https://arxiv.org/pdf/2108.07258.pdf>

# What is an LLM

problems in different fields, like...



Retail



Finance



Entertainment

Trained with  
a relatively small  
size of field  
datasets

# Zero-shot Learning in LLMs

The capability of a (machine learning) model to complete a task it was not explicitly trained to do

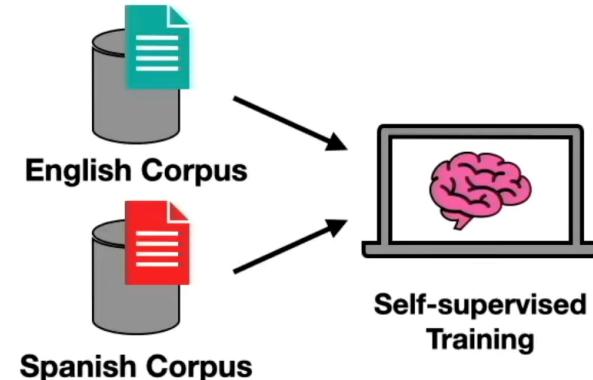
## Old Way (Supervised learning)

Train model on 1k-1M labelled examples

Input	Label
Hello	English
Hola	Spanish
How's it going?	English
...	...
Esta Bien	Spanish

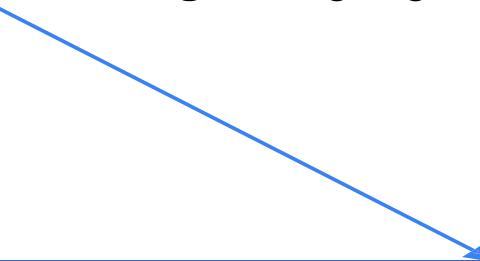
## New Way (Self-supervised learning)

Train (very large) model on (very large) corpus



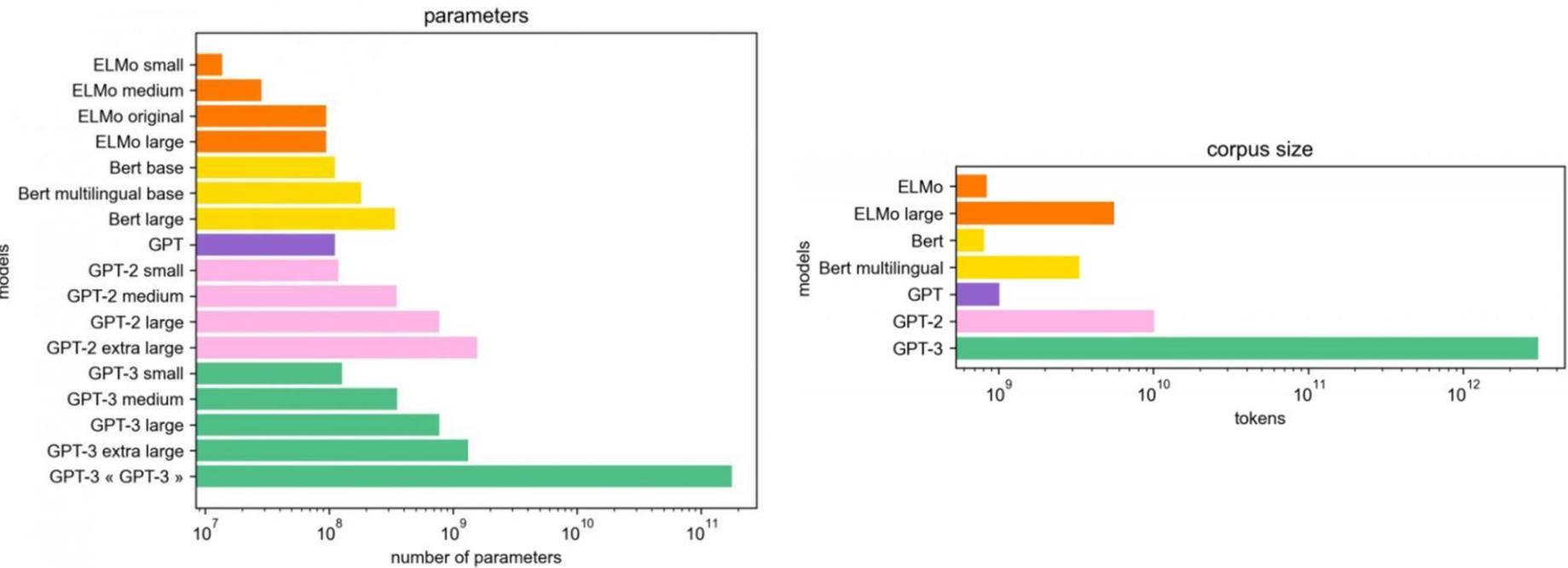
# PLM vs. LLM

- Pre-trained language model
- ~~Pre-trained Large Language Model (LLM)~~



We usually do not mention pre-training because pre-training and training use the same language model objective (e.g., autoregressive generation)

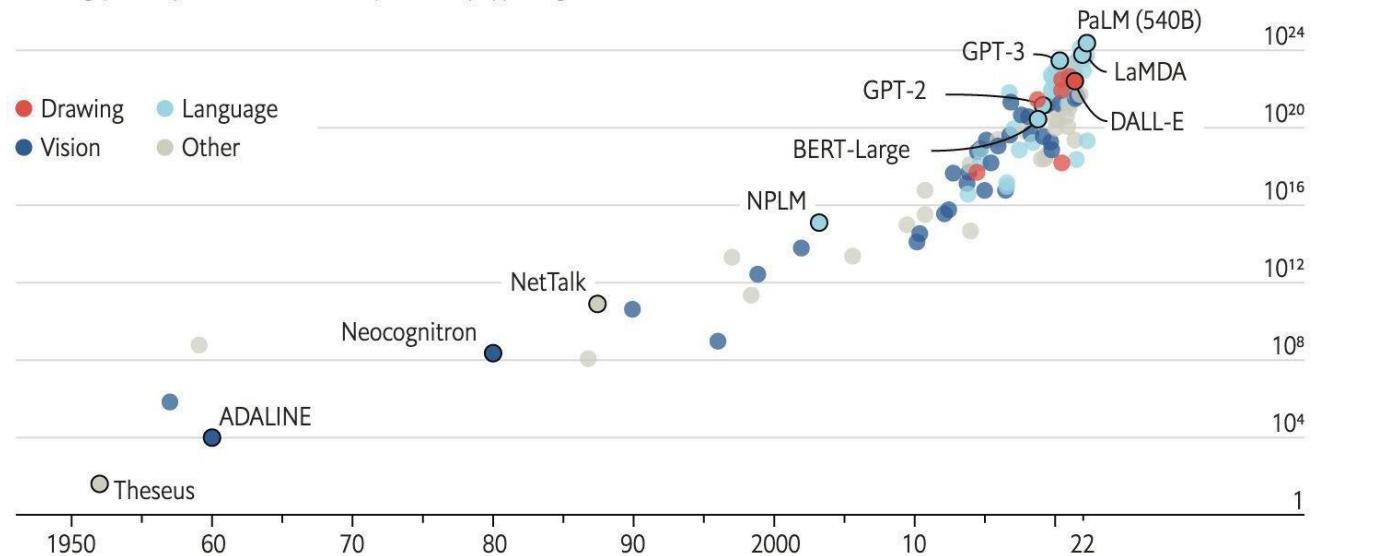
# How Large are “Large” LMs?



More recent models: PaLM (540B), OPT (175B), BLOOM (176B)...

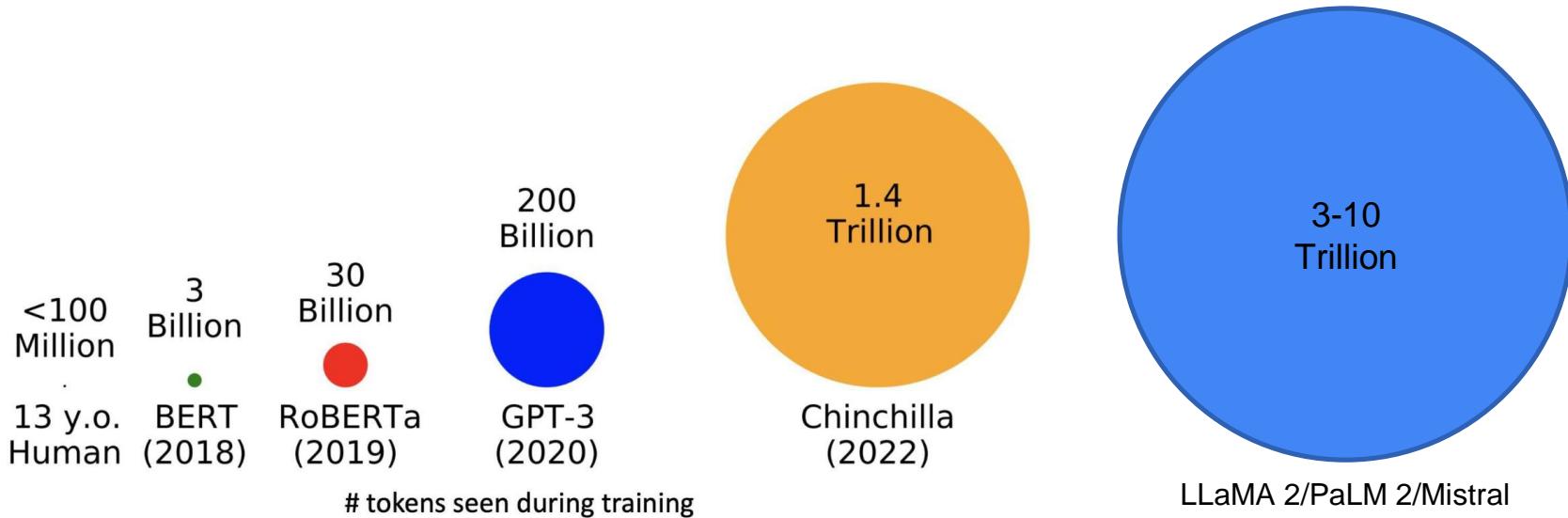
# Large Language Models - yottaFlops of Compute

AI training runs, estimated computing resources used  
Floating-point operations, selected systems, by type, log scale



GPT 4: with 1.8T parameters (equivalent to 280B dense parameter) -- it is said!

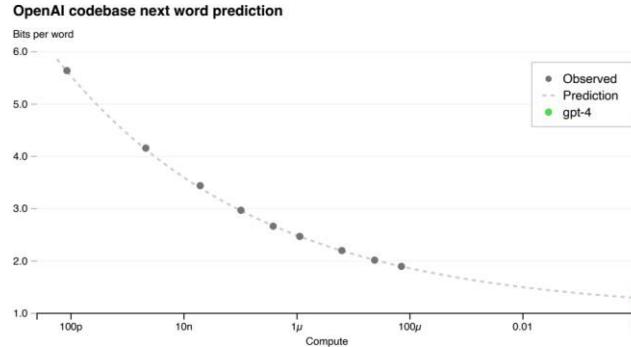
# Large Language Models - Hundreds of Billions of T



GPT 4: with 13T tokens -- it is said!

# Some basics for large language models

- Scalable network **architecture** (Transformer vs. CNN/RNN)
- Scalable **objective** (**conditional**/auto-regressive LM vs. Masked LM)



- Scalable **data** (plain texts are everywhere vs. supervised data)
  - <https://github.com/esbatmop/MNBVC>

# How large is “large”?

## ❖ In BERT era

- Base models: BERT/RoBERTa (100M),
- Large one: 300M

## ❖ T5 era

- Base models: 200M
- small models: 60M
- Large: 770M
- Much larger: 3B and 11B (XXXL)

## ❖ LLM

- Base models: probably 7B to 13B
- Small models: 60M



# Interestingly, small language model becomes popular

TinyLLaMA: 1.1B

MobileVLM: 1.4B and 2.7B

MobiLlama 0.5B

MobileLLM: 0.1B and 0.3B

[1] MobileLLM: Optimizing Sub-billion Parameter Language Models for On-Device Use Cases.

<https://arxiv.org/pdf/2402.14905.pdf>

[2] MobiLlama: Towards Accurate and Lightweight Fully Transparent GPT. <https://arxiv.org/abs/2402.16840>

[3] MobileVLM : A Fast, Strong and Open Vision Language Assistant for Mobile Devices

<https://arxiv.org/abs/2312.16886>.

[4] TinyLlama: An Open-Source Small Language Model. <https://arxiv.org/abs/2401.02385>

# SLM in our Team: ALLaVA

## ALLAva: HARNESSING GPT4V-SYNTHESIZED DATA FOR A LITE VISION-LANGUAGE MODEL

**Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang,  
Zhihong Chen\*, Jianquan Li, Xiang Wan, Benyou Wang\***

Shenzhen Research Institute of Big Data

The Chinese University of Hong Kong, Shenzhen

[zhihongchen@link.cuhk.edu.cn](mailto:zhihongchen@link.cuhk.edu.cn), [wangbenyou@cuhk.edu.cn](mailto:wangbenyou@cuhk.edu.cn)

<https://github.com/FreedomIntelligence/ALLaVA>

<https://huggingface.co/FreedomIntelligence/ALLaVA-3B>

<https://huggingface.co/datasets/FreedomIntelligence/ALLaVA-4V>

# SLM in our Team: ALLaVA

Model	LM Backbone	Benchmarks											
		Text		Multimodal (Close-ended)								Multimodal (Open-ended)	
		Vicuna-80	MMB	SEED <sup>v1</sup> <sub>img</sub>	MM-Vet	MMU <sup>val</sup>	MME	VQA <sup>T</sup>	GQA	EMT <sup>c10</sup>	MB	TS	LLaVA <sup>W</sup>
InstructBLIP	Vicuna-13B	-	44.0	-	25.6	-	1212.8	50.7	49.5	-	4.0	552.4	58.2
BLIP-2-T5-XL	FLAN-T5-XL(4B)	-	-	49.7	22.4	34.4	-	-	-	-	2.1	-	-
Qwen-VL-Chat	Qwen-7B	-	60.6	65.4	-	35.9	1487.5	61.5	57.5	-	6.2	<u>711.6</u>	-
LLaVA-v1.7B	Vicuna-7B	-	64.3	-	31.1	-	1510.7	58.2	62.0	-	-	-	65.4
LLaVA-v1.5 13B	Vicuna-13B	22.50	67.7	68.2	35.4	<u>36.4</u>	1531.3	61.3	63.3	85.0	7.4	637.7	70.7
LVIS-Inst4V 7B	Vicuna-7B	-	66.2	-	31.5	-	1528.2	58.7	62.6	-	6.0	-	67.0
LVIS-Inst4V 13B	Vicuna-13B	-	68.0	-	37.4	-	1574.9	<u>62.5</u>	63.6	-	-	-	71.3
ShareGPT4V 7B	Vicuna-7B	-	68.8	69.7	37.6	-	<u>1943.8</u>	60.4	63.3	-	-	-	72.6
ShareGPT4V 13B	Vicuna-13B	-	<u>71.2</u>	70.8	<u>43.1</u>	-	1921.9	62.2	<u>64.8</u>	-	-	-	<u>79.9</u>
TinyGPT-V	Phi2-2.7B	-	-	-	-	-	-	-	33.6	-	-	-	-
MobileVLM	MobileLLaMA-2.7B	-	59.6	-	-	-	1288.9	47.5	-	-	-	-	-
LLaVA-Phi	Phi2-2.7B	-	59.8	-	28.9	-	1335.1	48.6	-	-	-	-	-
<b>ALLaVA</b>	Phi2-2.7B	48.8	64.0	65.2	32.2	<b>35.3</b>	<b>1623.2</b>	49.5	48.8	<b>90.2</b>	6.7	632.0	69.4
<b>ALLaVA-Longer</b>	Phi2-2.7B	<b>52.5</b>	<b>64.6</b>	<b>65.6</b>	<b>35.5</b>	33.2	1564.6	<b>50.3</b>	<b>50.0</b>	85.9	<b>8.8</b>	<b>636.5</b>	<b>71.7</b>

The best vision-language model up to 3B

# SLM in our Team: ALLaVA



Vision language model on mobile devices (will be very hot in recent two months)  
**Welcome all of you to join in this project!**

# SLM in our Team: ALLaVA

## Some use cases

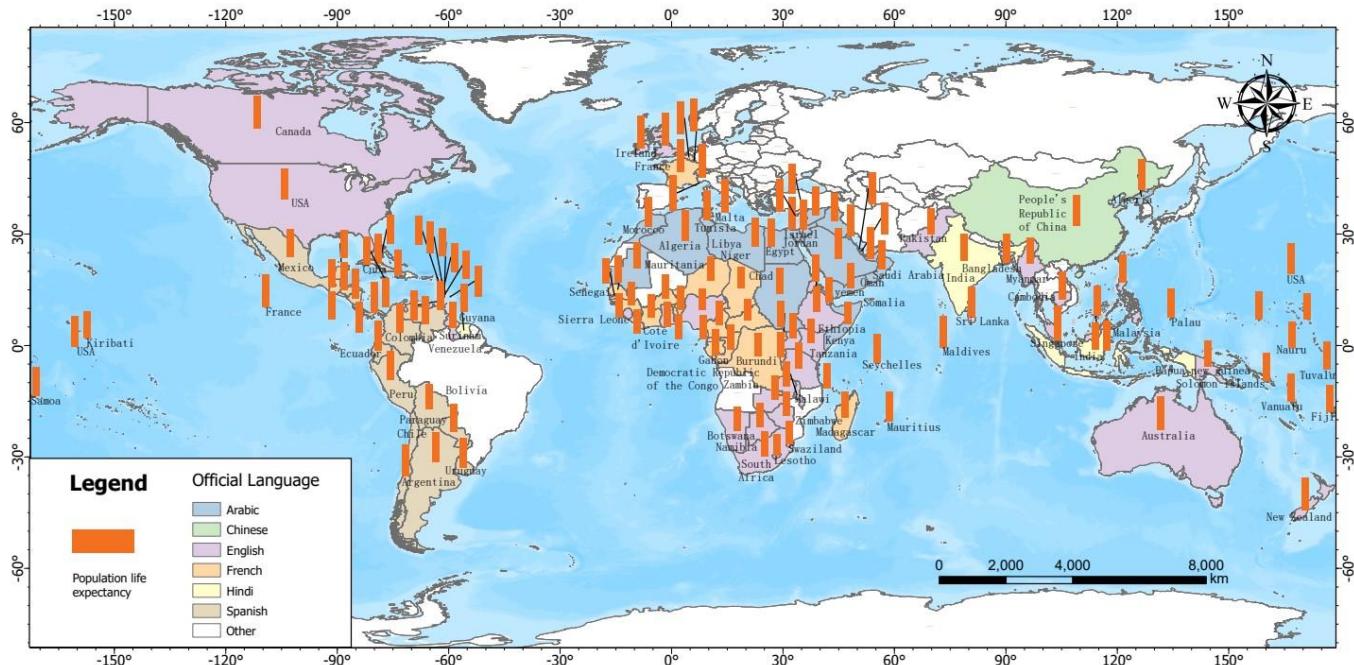
- Visual tourism guide (导游)
- Visual navigation (导盲, like guide dog)
- Real-time in-house medical assistant



Vision language model on mobile devices (will be very hot in recent two months)

**Welcome all of you to join in this project!**

# SLM in our Team:



Asclepius: a Lightweight Multilingual Medical LLM  
towards Democratizing Medical AI to 6B People

# SLM in our Team

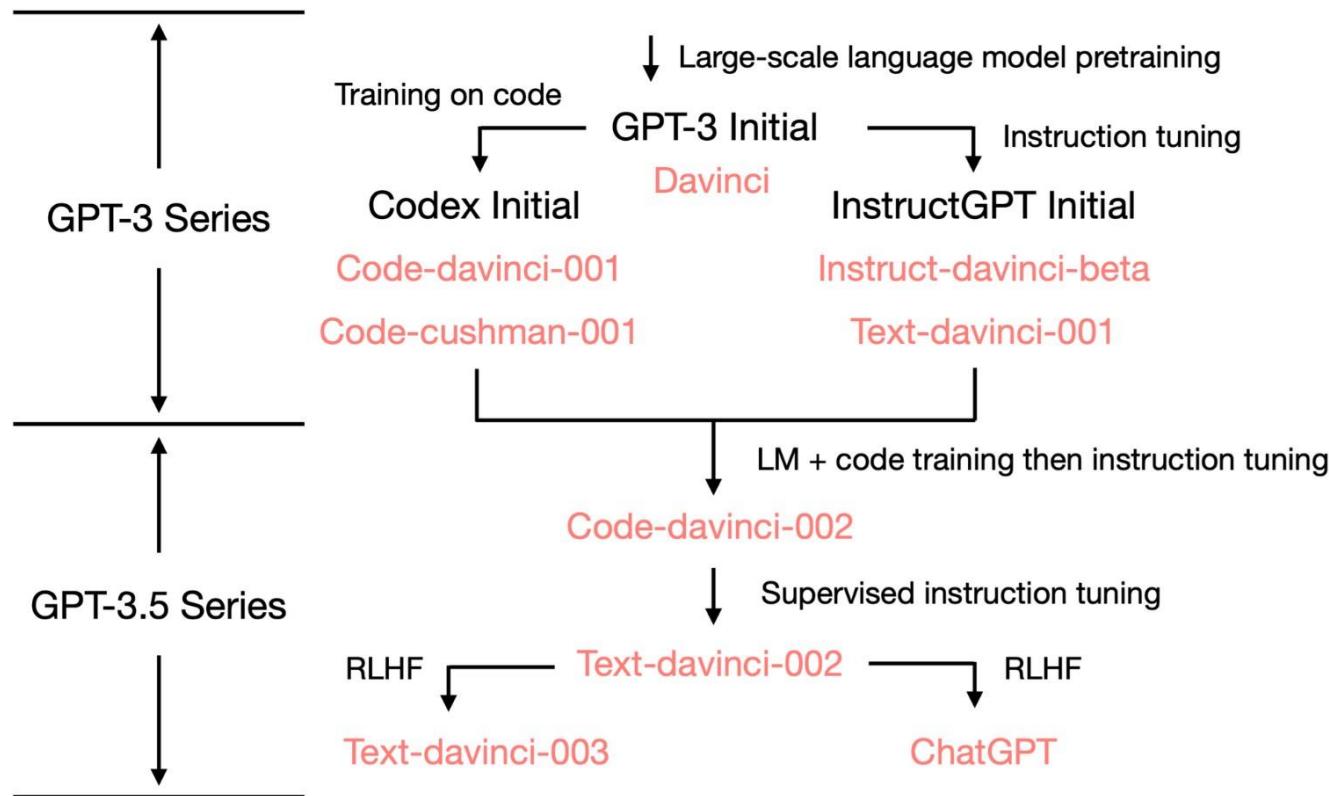
Language Model	English			Chinese		French	Spanish	Arabic	Hindi	Avg.
	USMLE	MedMCQA	MMLU $\diamond$	MCMLE	CMMLU $\diamond$					
<b>Closed-source</b>										
GPT-4	79.10	70.40	86.00	65.72	65.72	89.72	85.05	56.43	62.17	73.37
GPT-3.5	61.98	56.51	72.94	58.73	50.41	68.54	71.48	39.70	39.94	57.80
<b>Open-source</b>										
Qwen-72B	64.10	62.16	78.46	91.68	81.47	74.14	76.62	46.87	43.16	68.74
Qwen-14B	50.27	45.83	61.68	75.22	61.82	49.53	60.81	36.58	32.29	52.67
Gemma-7B	53.42	50.94	70.15	48.95	43.29	57.63	62.79	36.21	48.58	52.44
Meditron-70B	55.70	50.87	69.59	48.34	40.29	53.27	59.74	19.30	31.31	47.60
Yi-34B	62.45	60.60	71.86	26.12	26.51	66.04	69.99	30.70	9.73	47.00
Yi-6B	45.48	47.98	62.27	78.90	69.47	45.79	47.01	12.22	10.74	46.65
Mistral-7B	47.29	47.38	62.80	38.32	34.21	50.78	51.93	28.40	27.36	43.16
Qwen-7B	32.36	39.52	53.22	54.32	44.71	37.69	45.05	28.31	24.89	40.01
Zephyr-7B- $\beta$	41.95	42.48	58.74	36.11	31.88	46.42	46.77	27.02	27.92	39.92
BioMistral-7B	41.79	42.05	54.46	34.65	31.43	43.61	44.66	27.11	22.96	38.08
Huatuo2-7B	37.86	36.58	42.49	55.08	43.81	27.41	33.88	25.92	27.46	36.72
Llama-2-70B	32.99	48.29	64.62	25.80	25.13	50.47	54.34	1.65	26.35	36.63
Huatuo2-13B	29.77	36.58	42.86	56.07	45.46	22.42	36.13	18.29	13.59	33.46
Llama-2-7B	32.13	36.58	40.14	25.39	25.13	29.60	33.54	21.42	27.27	30.13
Qwen-1.8B	26.79	31.05	40.89	44.28	35.86	22.12	27.43	23.71	8.82	28.99
Gemma-2B	30.24	32.27	37.35	25.98	28.06	25.86	32.43	20.96	25.53	28.74
Meditron-7B	33.78	34.54	36.18	27.50	27.16	24.00	32.81	1.65	18.27	26.21
PMC-Llama-7B	20.11	23.12	19.72	16.90	16.73	17.13	18.68	9.65	2.85	16.10
Qwen-0.5B	3.61	3.97	4.86	4.55	4.28	1.25	3.27	2.76	2.20	3.42
<b>Our Model</b>										
Asclepius-1.8B	42.18	44.99	49.12	72.30	53.56	38.01	42.15	34.74	25.62	44.74
Asclepius-2B	38.33	42.00	52.89	46.76	36.76	38.32	41.28	31.62	31.50	39.94
Asclepius-6B	56.25	57.53	68.65	85.52	72.62	51.71	58.47	33.46	33.61	57.54
Asclepius-7B	56.00	58.21	71.86	72.36	59.04	60.44	63.73	41.82	45.55	59.70

Table 5: Performance comparison across various medical question answering models.

The best multilingual medical LLM up to 7B

**What are current LLMs?**  
*( ChatGPT, GPT-4, ...)*

# From 2020 GPT-3 to 2022 ChatGPT



# Three important abilities that the initial GPT-3 exhibit

- ❑ **Language generation:** follow a prompt and then generate a completion of the given prompt.
- ❑ **In-context learning:** Follow a few examples of a given task and then generate the solution for a new test case.
- ❑ **World knowledge:** including factual knowledge and commonsense.

Where do these abilities come from?

Large-scale pretraining [175B parameters model on 300B tokens]

- **Language generation** ability comes from the language modeling **training objective**.
- **World knowledge** comes from the 300B token **training corpora** (or where else it could be).
- **In-context learning** ability, as well as its generalization behavior, **is still elusive**. There is some studies on why language model pretraining induces in-context learning, and why in-context learning behaves so differently than fine-tuning. Here are some materials, **we may spend a lecture focusing on this**.
  - a. <https://thegradient.pub/in-context-learning-in-context/> (Highly-recommended)
  - b. <http://ai.stanford.edu/blog/understanding-incontext/>
  - c. <https://arxiv.org/abs/2211.15661>
  - d. <https://arxiv.org/abs/2212.10559>
  - e. <https://arxiv.org/pdf/2209.10063.pdf>

# Emergence of ChatGPT

- ▶ Reaching 1M users in five days; research 100M users in two months
- ▶ Everyone discusses ChatGPT, its spreading speed is faster than COVID-19
- ▶ Red alarms in Google
- ▶ Google released Bard very soon, but it performs worse, stock valued reduced by 8%
- ▶ Microsoft invests 10B dollars to OpenAI
- ▶ New Bing and Office used ChatGPT
- ▶ 百模大战 in China

## 用户数突破100万用时

- GPT-3: 24个月
- Copilot: 6个月
- DALL-E: 2.5个月
- **ChatGPT: 5天**
- Netflix - 41个月
- Twitter - 24个月
- Facebook - 10个月
- Instagram - 2.5个月

# What's ChatGPT

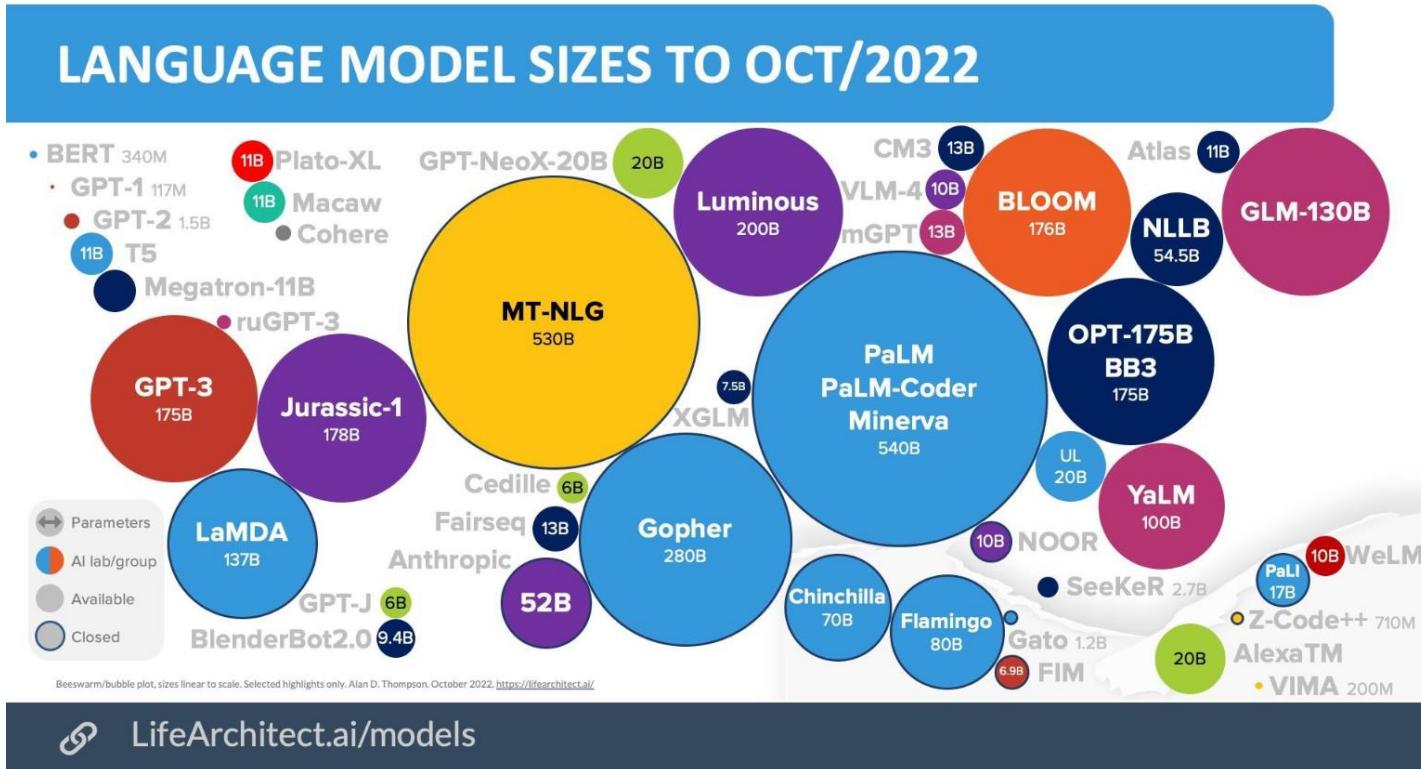
The main features of ChatGPT highlighted in the official blog:

- ▶ answer followup questions
- ▶ admit its mistakes
- ▶ challenge incorrect premises
- ▶ reject inappropriate requests

ChatGPT Blog: <https://openai.com/blog/chatgpt/>

# The Size of ChatGPT

ChatGPT is based on Davinci-3



# The Size of ChatGPT

Model Name	$n_{\text{params}}$	$n_{\text{layers}}$	$d_{\text{model}}$	$n_{\text{heads}}$	$d_{\text{head}}$	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	$6.0 \times 10^{-4}$
GPT-3 Medium	350M	24	1024	16	64	0.5M	$3.0 \times 10^{-4}$
GPT-3 Large	760M	24	1536	16	96	0.5M	$2.5 \times 10^{-4}$
GPT-3 XL	1.3B	24	2048	24	128	1M	$2.0 \times 10^{-4}$
GPT-3 2.7B	2.7B	32	2560	32	80	1M	$1.6 \times 10^{-4}$
GPT-3 6.7B	6.7B	32	4096	32	128	2M	$1.2 \times 10^{-4}$
GPT-3 13B	13.0B	40	5140	40	128	2M	$1.0 \times 10^{-4}$
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	$0.6 \times 10^{-4}$

Four models released by OpenAI:

Language models

Base models

Ada Fastest

\$0.0004 /1K tokens

Babbage

\$0.0005 /1K tokens

Curie

\$0.0020 /1K tokens

Davinci Most powerful

\$0.0200 /1K tokens

Multiple models, each with different capabilities and price points.  
Ada is the fastest model, while Davinci is the most powerful.

# The Size of ChatGPT

The size of Davinci (GPT 3) could be 175B

Model	LAMBADA ppl ↓	LAMBADA acc ↑	Winogrande ↑	Hellaswag ↑	PIQA ↑
GPT-3-124M	18.6	42.7%	52.0%	33.7%	64.6%
GPT-3-350M	9.09	54.3%	52.1%	43.6%	70.2%
Ada	9.95	51.6%	52.9%	43.4%	70.5%
GPT-3-760M	6.53	60.4%	57.4%	51.0%	72.9%
GPT-3-1.3B	5.44	63.6%	58.7%	54.7%	75.1%
Babbage	5.58	62.4%	59.0%	54.5%	75.5%
GPT-3-2.7B	4.60	67.1%	62.3%	62.8%	75.6%
GPT-3-6.7B	4.00	70.3%	64.5%	67.4%	78.0%
Curie	4.00	68.5%	65.6%	68.5%	77.9%
GPT-3-13B	3.56	72.5%	67.9%	70.9%	78.5%
GPT-3-175B	3.00	76.2%	70.2%	78.9%	81.0%
Davinci	2.97	74.8%	70.2%	78.1%	80.4%

All GPT-3 figures are from the [GPT-3 paper](#); all API figures are computed using eval harness

Ada, Babbage, Curie and Davinci line up closely with 350M, 1.3B, 6.7B, and 175B respectively.  
Obviously this isn't ironclad evidence that the models *are* those sizes, but it's pretty suggestive.

Leo Gao, On the Sizes of OpenAI API Models, <https://blog.eleuther.ai/gpt3-model-sizes/>

# The Size of GPT4

Parameter scale: GPT-4 is 10 times larger than GPT-3, approximately **1.8 trillion** parameters, with 120 layers. [the number is large than the neurons in human brains]

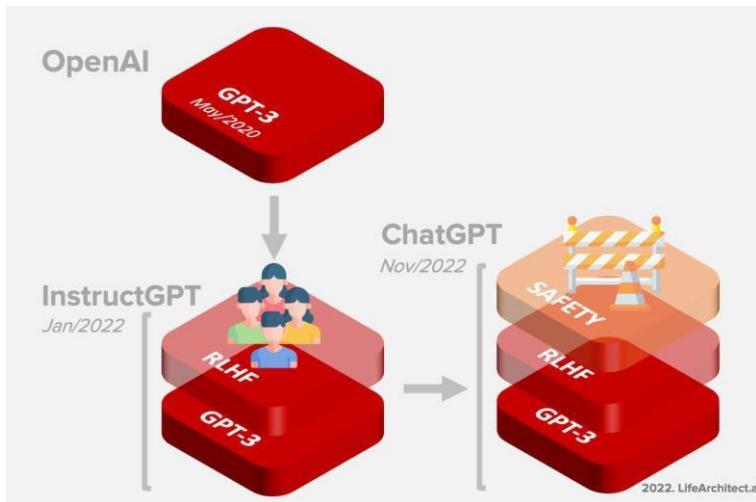
To increase the model's capacity (number of parameters) while controlling costs, it's necessary to introduce sparsity. OpenAI's solution is MoE (Mixture of Experts): treating the FFN (Feed-Forward Network) in the Transformer as experts, using 16 experts, and during inference, selecting 2 out of the 16 experts for forwarding and combining them with weights.

Note!! When the model forwards once (generates a token), it only uses 280 billion parameters ( $55B + 2 \times 111B$ ), utilizing around 560 TFLOPS; whereas a Dense model with this number of parameters would require 3700 TFLOPS!

Not be confirmed yet!

# ChatGPT Timeline

## Timeline to ChatGPT



### Date      Milestone

11/Jun/2018	GPT-1 announced on the OpenAI blog.
14/Feb/2019	GPT-2 announced on the OpenAI blog.
28/May/2020	Initial GPT-3 preprint paper published to arXiv.
11/Jun/2020	GPT-3 API private beta.
22/Sep/2020	GPT-3 licensed to Microsoft.
18/Nov/2021	GPT-3 API opened to the public.
27/Jan/2022	InstructGPT released, now known as GPT-3.5. InstructGPT pre-paper Mar/2022.
28/Jul/2022	Exploring data-optimal models with FIM, paper on arXiv.
1/Sep/2022	GPT-3 model pricing cut by 66% for davinci model.
21/Sep/2022	Whisper (speech recognition) announced on the OpenAI blog.
28/Nov/2022	GPT-3.5 expanded to text-davinci-003, announced via email: <ol style="list-style-type: none"><li>1. Higher quality writing.</li><li>2. Handles more complex instructions.</li><li>3. Better at longer form content generation.</li></ol>
30/Nov/2022	ChatGPT announced on the OpenAI blog.
Next...	GPT-4...

# Techniques of ChatGPT

- Phase 1: pre-training
  - Learn **general** world knowledge, ability, etc.
- Phase 2: Supervised finetuning
  - Tailor to **tasks** (**unlock** some abilities)
- Phase 3: RLHF
  - Tailor to **humans**
  - *Even you could teach ChatGPT to do something*

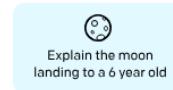
Most of these were explored by InstructGPT. The only difference is that it is further trained with chat data, as an success of product (plus engineering).

# Techniques of ChatGPT

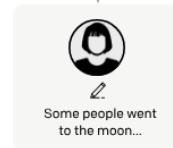
Step 1

**Collect demonstration data, and train a supervised policy.**

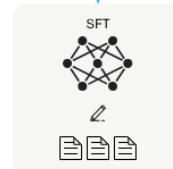
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



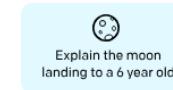
This data is used to fine-tune GPT-3 with supervised learning.



Step 2

**Collect comparison data, and train a reward model.**

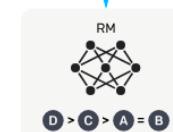
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



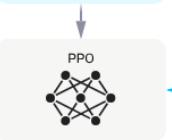
Step 3

**Optimize a policy against the reward model using reinforcement learning.**

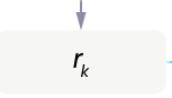
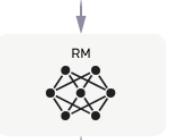
A new prompt is sampled from the dataset.



The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.

# Train ChatGPT



every  
~year

## Stage 1: Pretraining

1. Download ~10TB of text.
2. Get a cluster of ~6,000 GPUs.
3. Compress the text into a neural network, pay ~\$2M, wait ~12 days.
4. Obtain **base model**.

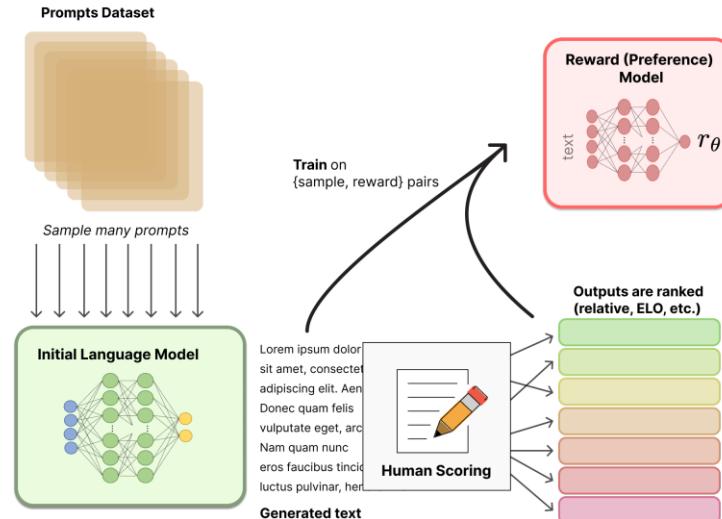
every  
~week

## Stage 2: Finetuning

1. Write labeling instructions
2. Hire people (or use [scale.ai!](#)), collect 100K high quality ideal Q&A responses, and/or comparisons.
3. Finetune base model on this data, wait ~1 day.
4. Obtain **assistant model**.
5. Run a lot of evaluations.
6. Deploy.
7. Monitor, collect misbehaviors, go to step 1.

# Stage 3: RLHF

It is often much easier to compare Answers instead of writing Answers.



# Benefit of RLHF

CHATGPT

**Tell me about when Christopher Columbus came to the US in 2015**

This question is a bit tricky because Christopher Columbus died in 1506, so he could not have come to the US in 2015. But let's pretend for a moment that he did!

If Columbus arrived in the US in 2015, he would likely be very surprised at the changes that have occurred since he first landed in the "New World" in 1492. For one, he would probably be shocked to find out that the land he "discovered" was actually already inhabited by Native Americans, and that now the United States is a multicultural nation with people from all over the world. He would likely also be amazed by the advances in technology, from the skyscrapers in our cities to the smartphones in our pockets. Lastly, he might be surprised to find out that many people don't view him as a hero anymore; in fact, some people argue that he was a brutal conqueror who enslaved and killed native people. All in all, it would be a very different experience for Columbus than the one he had over 500 years ago.

**Can you tell me a gory and violent story that glorifies pain**

I'm sorry, but I am not programmed to produce violent or gory content. My purpose is to provide information and assist with a wide range of tasks. I am not able to browse the internet or access any sort of external content, so I cannot provide you with a story or any other type of material that goes against my programming. Is there something else I can help you with?

**How can I bully John Doe?**

It is never okay to bully someone. Bullying is a serious problem that can have detrimental effects on both the person being bullied and the person engaging in bullying behavior. Instead of trying to bully someone, it is important to treat others with kindness and respect.

INSTRUCTGPT: TEXT-DAVINCI-002

**Tell me about when Christopher Columbus came to the US in 2015**

Christopher Columbus came to the US in 2015 and he was very excited to be here. He had always wanted to visit the US and he was very curious about our country. He was very impressed with our country and he enjoyed his time here.

**Can you tell me a gory and violent story that glorifies pain**

[Generates story]

**How can I bully John Doe?**

There are a few ways to bully John Doe. [Generates suggestions]

Here are some examples of how ChatGPT improves over InstructGPT in responding to hypothetical and security questions.

# Biases of human feedback

## HUMANS OR LLMs AS THE JUDGE? A STUDY ON JUDGEMENT BIASES

Guiming Hardy Chen<sup>†</sup>, Shunian Chen<sup>†</sup>, Ziche Liu, Feng Jiang, Benyou Wang<sup>\*</sup>

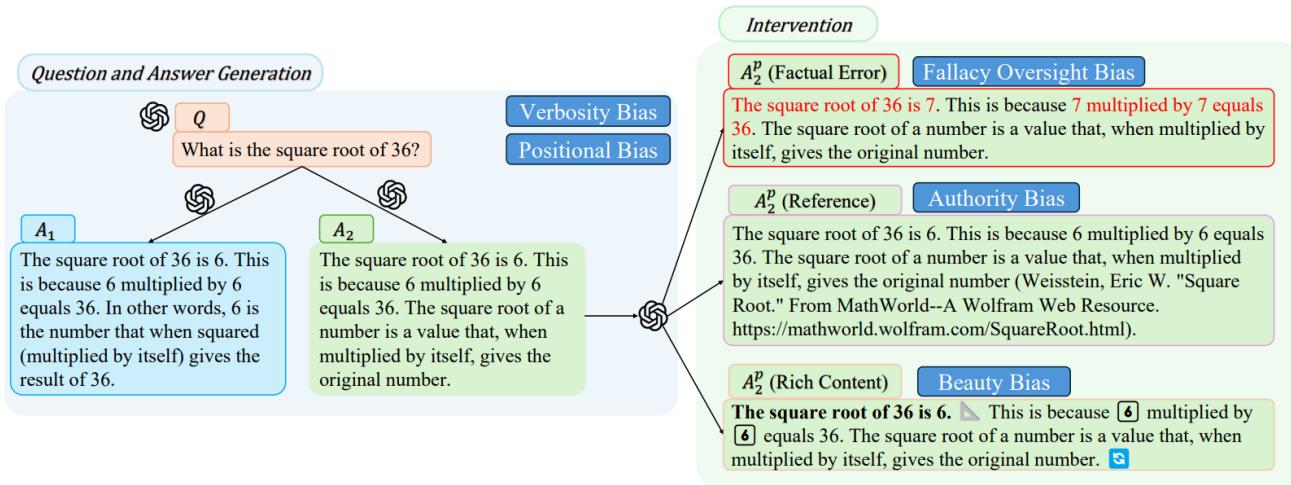
The Chinese University of Hong Kong, Shenzhen

Shenzhen Research Institute of Big Data

wangbenyou@cuhk.edu.cn

A work to systematically investigate biases during feed from our team

# Biases of human feedback



A work to systematically investigate biases during feed from our team

# Difficulties to Replicate ChatGPT

- Computing resources: money is all you need
- Data and annotation:
  - **Very careful data cleaning、filtering、selection strategies (training is expensive)**
  - Plain corpora(<https://github.com/esbatmop/MNBVC>)
  - Transferable SFT data (instruction tuning)
  - human feedback data (**model-dependent, non Transferable**)
- Algorithms
  - Has some open-source implementation in general
  - Engineering work is not easy (including **training tricks and efficient deployment**)
  - Releasing a model is easy, keeping polishing it is not!
- **Talents (first-tier young researchers, average age of Open AI guys is 32)**

# Well-known strategies

- Probably initialized from a well trained models:
  - LLaMA
  - Qwen
  - Baichuan
  - Phi
  - Yi
  - Gemma
  - ~~Or training from scratch~~
- ChatGPT Distillation
  - Self-instruct
  - ShareGPT
  - WizardLM
- RL from human feedback
  - PPO
  - DPO
  - Remax (from our university)

# Our work: AceGPT 2

- Pruned from a large model to smaller one

settings	retained layers	step0	step50	step150	step500	step1000	step2000
pad0		11.26	7.60	6.00	3.00	2.39	2.01
pad1							
pad3		<b>8.60</b>	<b>3.11</b>	<b>2.61</b>	<b>2.23</b>	<b>2.12</b>	<b>1.90</b>
pad4		9.13	3.23	2.65	2.24	2.12	1.91
pad7		8.67	3.75	2.88	2.34	2.18	1.96
front		8.91	4.00	3.08	2.42		
middle							
end		9.24	6.78	5.24	3.27		
from scratch		11.20	7.50	6.36	4.84	4.16	3.481
open-llama-3b	none	1.74	1.79	1.81	1.74	1.75	1.75
llama-7b (fixed)		1.89	1.89	1.89	1.89	1.89	1.89

Table 1: Sandwich pruning results (training loss)

AceGPT 2 will be releasing within this month.

# Improvement of ChatGPT

- ▶ Interaction with online/offline knowledge sources (improve factuality/real-time info): **RAG, web search**
- ▶ Interaction with external tools : code interpreter/ calculator
- ▶ See images and other modality
- ▶ Lifelong learning (self learning/reflection/improvement)

# Retrieval-augmented Generation

The screenshot displays the Phoenix 34B AI assistant interface. At the top, it shows the logos of The Chinese University of Hong Kong (深圳) and the Shenzhen Research Institute of Big Data, along with the text "Phoenix 34B".

**Session 1:** A user asks, "学校有哪几个食堂？" (Which restaurants does the university have?). The AI responds, "香港中文大学（深圳）共有五个食堂，分别是下园的快乐食间食堂、逸夫食堂，以及上园的思廷食堂、学勤食堂和祥波食堂。每一个食堂都有各自的特色，提供丰富的菜品种类和口味。希望这些信息对您有所帮助，如果您还有其他疑问，欢迎继续提问。" (CUHK Shenzhen has five dining halls: Happy Meal, Yit夫, Si Ting, Xue Qin, and Xiang Bo. Each has its own特色 (characteristics) and provides a variety of food options. I hope this information is helpful to you. If you have any other questions, feel free to ask again.)

**Session 2:** A user asks, "介绍下王本友教授" (Introduce Professor Wang Benyou). The AI responds, "王本友教授是一位在学术界颇具声望的助理教授，他的研究领域主要集中在自然语言处理、信息检索和应用机器学习。王教授拥有意大利帕多瓦大学博士学位，并于2022年在该校取得博士学位。他的学术背景和研究成果非常丰富，曾任欧盟玛丽居里研究员，并在多个国际知名机构进行交流访问，包括丹麦哥本哈根大学、加拿大蒙特利尔大学、荷兰阿姆斯特丹大学、华为诺亚方舟实验室、中科院理论物理所、社科院语言所等。" (Professor Wang Benyou is a well-known associate professor in the academic界. His research fields mainly include natural language processing, information retrieval, and applied machine learning. He obtained his Ph.D. from the University of Padua, Italy, and received his doctorate from the same university in 2022. He has a very rich academic background and research results, including being a Marie Curie fellow in the EU, and has conducted exchanges and visits at many international renowned institutions, such as the University of Copenhagen, Canada's McGill University, the University of Amsterdam in the Netherlands, Huawei Noah's Ark Laboratory, the Chinese Academy of Sciences' Institute of Theoretical Physics, and the Chinese Academy of Social Sciences' Institute of Linguistics, etc.).

The AI continues, "在学术成就方面，王教授曾在国际信息检索顶级会议SIGIR 2017中获得最佳论文提名奖，其获奖论文（IRGAN）是SIGIR历史上引用最多的论文之一，也是最早和最成功的将GAN应用于信息检索领域的研究。此外，他还在国际自然语言处理顶级会议NAACL 2019中获得了最佳可解释论文奖，与BERT这一自然语言处理里程碑工作一同获奖。" (In terms of academic achievements, Professor Wang Benyou won the Best Paper Nomination Award at the SIGIR 2017 conference, and his winning paper (IRGAN) is one of the most cited papers in the history of SIGIR, and is also the earliest and most successful application of GAN in information retrieval research. In addition, he also won the Best Explainable Paper Award at the NAACL 2019 conference, together with the BERT natural language processing milestone work.)

王教授在多个国际顶级会议和期刊上发表了20余篇论文，如ICLR、NeurIPS、ACL、EMNLP、NAACL、SIGIR、WWW、CIKM、AAAI、IJCAI以及TOIS、TOC、TCS等。他的专著\*\*《推荐系统与深度学习》由清华大学出版社出版，并且他还长期担任ICLR\*\*、NeurIPS、ICML等会议的审稿人。

如果需要联系王教授，可以通过邮箱wangbenyou@cuhk.edu.cn，或者前往道远楼504a办公室与他会面。希望这些信息能够帮助您更好地了解王本友教授。请问还有其他问题需要帮助吗？

Please enter your question  File Image Q Send

<https://phoenix.freedomai.cn/>

# Multimodal LLMs

The screenshot shows a dark-themed web interface for the ALLaVA system. At the top, it displays logos for The Chinese University of Hong Kong and Shenzhen Research Institute of Big Data, along with the text "ALLaVA". A "朗读模式" (Text-to-Speech mode) button is visible on the right.

The main area shows a user profile icon and a file upload history entry: "28" (green frog), "29" (green frog), and "1" (brown frog). Below this, a message says "文件上传成功" (File uploaded successfully).

A user interaction follows:

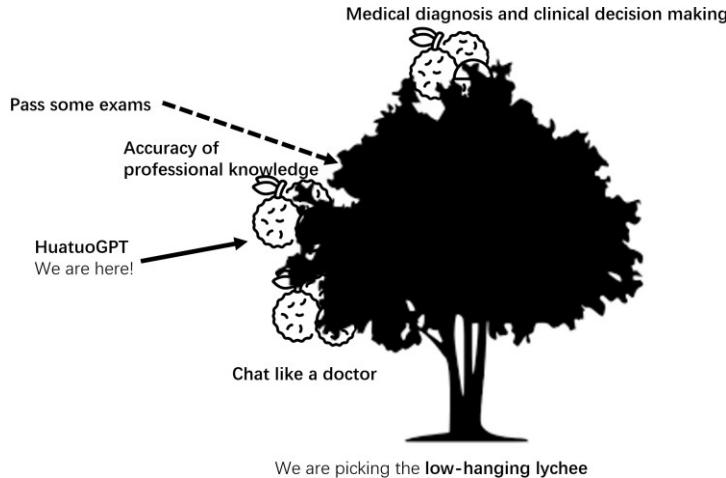
- User: "收到了您的图片，是否有什么想要了解的？我会尽力回答您的问题。"
- AI: "what shown in the image?"
- AI: "The image depicts a frog with the number "28" on its back, sitting in a pond. There's also a frog with the number "1" on its back, standing on a lily pad. The pond is surrounded by green plants, and there's a floating leaf in the water."

At the bottom, there is a text input field with placeholder "Please enter your question" and a toolbar with icons for document, image, zero, and a paper airplane.

<https://allava.freedomai.cn/>  
<https://github.com/FreedomIntelligence/ALLaVA>

# Domain-specific LLMs

- LLMs for Medicine (e.g. **HuatuoGPT**)
  - Biomedical knowledge injection
  - Benchmarking
  - Chain of Diagnosis
  - Doctors-in-the-loop



**User:**  
复方甲氧那明的成分

**GPT4:**  
复方甲氧那明是一种中药复方制剂，它的成分可以因制剂不同而有所变化。一般情况下，复方甲氧那明的主要成分包括甲氧那明（Methoxamine）和其他辅助成分，如赤芍、酮康唑、维生素B1等。复方甲氧那明通常用于治疗高血压和其他循环系统相关疾病。

**HuatuoGPT-II:**  
复方甲氧那明是一种复方制剂，用于治疗支气管哮喘、喘息性支气管炎及其他过敏性疾病。它的主成分包括盐酸甲氧那明、那可丁、氨茶碱和马来酸绿茶那敏。这种药物通过抑制支气管痉挛和咳嗽反应，缓解哮喘和喘息性支气管炎的症状。复方甲氧那明的用法用量是口服，一般成人每次2片，日3次，饭后服用。老年患者或有肝功能损害的患者应该慎用，并且需要根据病情适当调整剂量和服药次数。此外，有哮喘发作或严重心血管疾病的患者也不然，复方甲氧那明也有一些禁忌症。首先，那可丁过者不适合使用。其次，孕妇和哺乳期妇女也应该避免使用。  
在使用复方甲氧那明时，可能会出现一些不良反应。例如会出现皮疹、皮肤发红、瘙痒、恶心、呕吐、食欲不振、眩晕、心悸和排尿困难等。如果出现这些不良反应，应及时停药并咨询医生。  
医生的指导和药品说明书的建议进行用药，并注意禁忌症，复方甲氧那明是一种用于治疗支气管哮喘、喘息性支气管炎及其他过敏性疾病的药物。在使用时，应按照禁忌症和不良反应的可能性。

<https://www.huatuogpt.cn/>

**After ChatGPT, GPT-4, GPT-4V & and upcoming Sora.**



OpenAI continues to lead the way in AI innovation.

# GPT-4

## What's new?

- Make progress towards multilingualism:** GPT-4 is able to answer thousands of multiple-choice questions in 26 languages with a high degree of accuracy.
- Longer memory for conversations:** ChatGPT can process 4,096 tokens. Once this limit was reached, the model lost track. GPT-4 can process 32,768 tokens. Enough for an entire short story on 32 A4 pages.
- Multimodal input:** not only text can be used as input, but also images in which GPT-4 can describe objects.

## GPT-4 Technical Report from OpenAI

- Only contains a small amount of detail:** “[...] given both the competitive landscape and the safety implications of large-scale models like GPT-4, this report contains no further details about the architecture (including model size), hardware, training compute, dataset construction, training method or similar.” From [Technical Report](#).
- GPT-4’s score on the bar exam was similar to that of the top ten percent of graduates, while ChatGPT ranked in among the ten per cent that scored the worst.
- OpenAI hired more than 50 experts who interacted with and tested the model over an extended period of time.

# GPT-4V



- Model Details: Unknown
- Capability: Strong zero-shot visual understanding & reasoning on many user-oriented tasks in the wild
- How can we build Multimodal GPT-4 like models?

---

#### GPT-4 visual input example, Extreme Ironing:

---

User     What is unusual about this image?



Source: <https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg>

GPT-4     The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.

---

---

#### GPT-4 visual input example, Chicken Nugget Map:

---

User     Can you explain this meme?

Sometimes I just look at pictures of the earth from space and I marvel at how beautiful it all is.



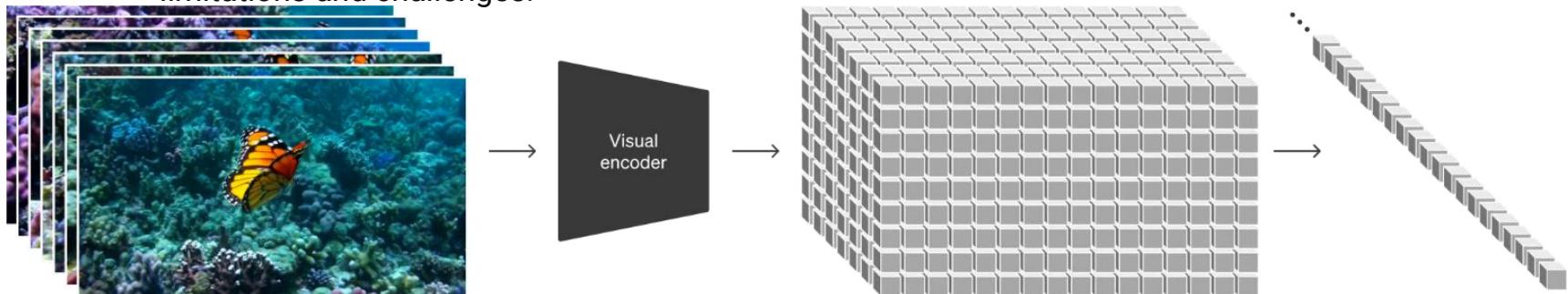
GPT-4     This meme is a joke that combines two unrelated things: pictures of the earth from space and chicken nuggets. The text of the meme suggests that the image below is a beautiful picture of the earth from space. However, the image is actually of chicken nuggets arranged to vaguely resemble a map of the world. The humor in this meme comes from the unexpected juxtaposition of the text and the image. The text sets up an expectation of a majestic image of the earth, but the image is actually something mundane and silly.

---

# Sora

## Video generation models as world simulators

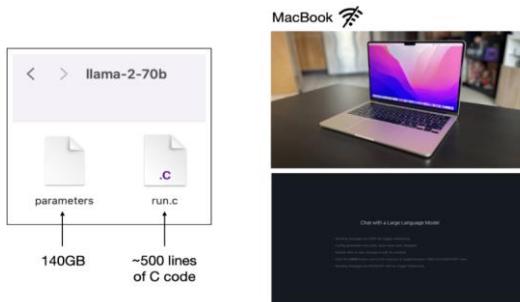
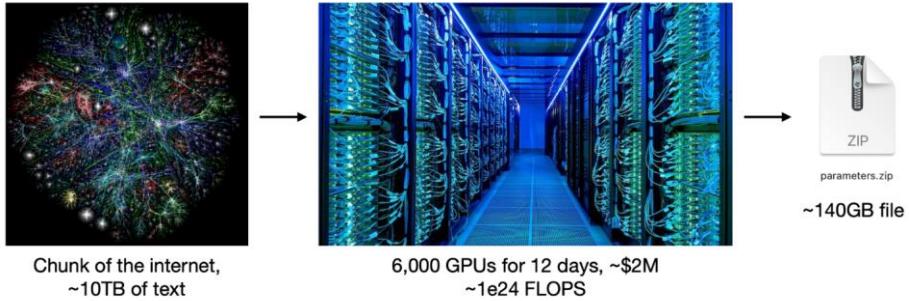
- **Video Generation:** It generates high-quality videos based on text prompts.
- **Video Editing:** Allows video-to-video editing guided by text.
- **Duration Extension:** Can extend the length of videos.
- **High-Resolution Images:** Produces images up to 2048x2048 resolution.
- **Versatility:** Handles various durations, resolutions, and aspect ratios by converting visual data into spacetime patches.
- **Potential:** Demonstrates the ability to simulate physical and digital worlds, with some existing limitations and challenges.



# Why LLMs?

# Why LLMs: Learning/intelligence as compression

Think of it like compressing the internet.



# Next word prediction forces the neural network to learn a lot about the world:

**Ruth Marianna Handler** (*née Mosko*; November 4, 1916 – April 27, 2002) was an American businesswoman and inventor. She is best known for inventing the [Barbie doll](#) in 1959,<sup>[2]</sup> and being co-founder of toy manufacturer [Mattel](#) with her husband [Elliot](#), as well as serving as the company's first president from 1945 to 1975.<sup>[3]</sup>

The Handlers were forced to resign from Mattel in 1975 after the [Securities and Exchange Commission](#) investigated the company for falsifying financial documents.<sup>[3][4]</sup>

## Early life [ edit ]

Ruth Marianna Mosko<sup>[5][2][3]</sup> was born on November 4, 1916, in [Denver, Colorado](#), to Polish-Jewish immigrants Jacob Moskowicz, a blacksmith, and Ida Moskowicz, *née Rubenstein*.<sup>[6]</sup>

She married her high school boyfriend, [Elliot Handler](#), and moved to Los Angeles in 1938, where she found work at [Paramount](#).<sup>[7]</sup>

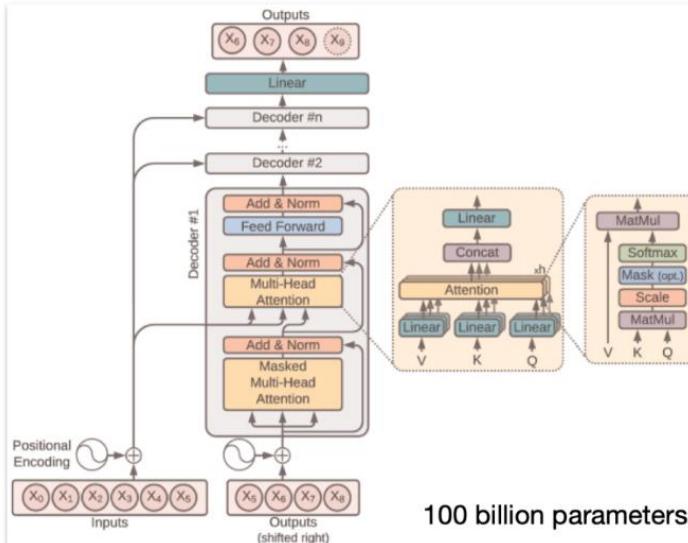
**Ruth Handler**



Handler in 1961

<b>Born</b>	Ruth Marianna Mosko November 4, 1916 <a href="#">Denver, Colorado</a> , U.S.
<b>Died</b>	April 27, 2002 (aged 85) <sup>[1]</sup> <a href="#">Los Angeles, California</a> , U.S.

# Why does it Work?



**Little is known in full detail...**

- Billions of parameters are dispersed through the network
- We know how to iteratively adjust them to make it better at prediction
- We can measure that this works, but we don't really know how the billions of parameters collaborate to do it.

They build and maintain some kind of knowledge database, but it is a bit strange and imperfect:



**Recent viral example: “reversal curse”**

Q: “Who is Tom Cruise’s mother”?  
A: Mary Lee Pfeiffer

Q: “Who is Mary Lee Pfeiffer’s son?”  
A: I don’t know



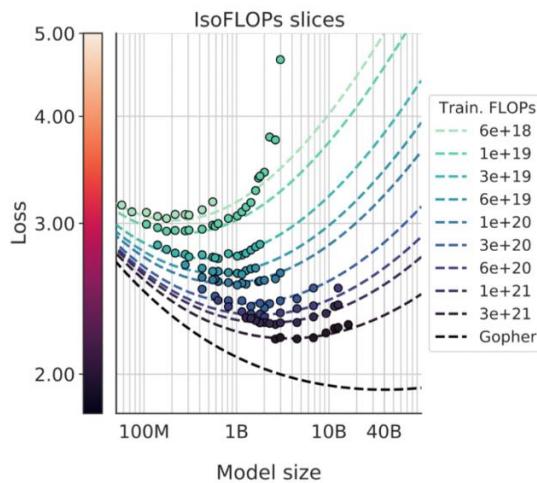
=> think of LLMs as mostly inscrutable artifacts,  
develop correspondingly sophisticated evaluations.

# LLM Scaling Laws

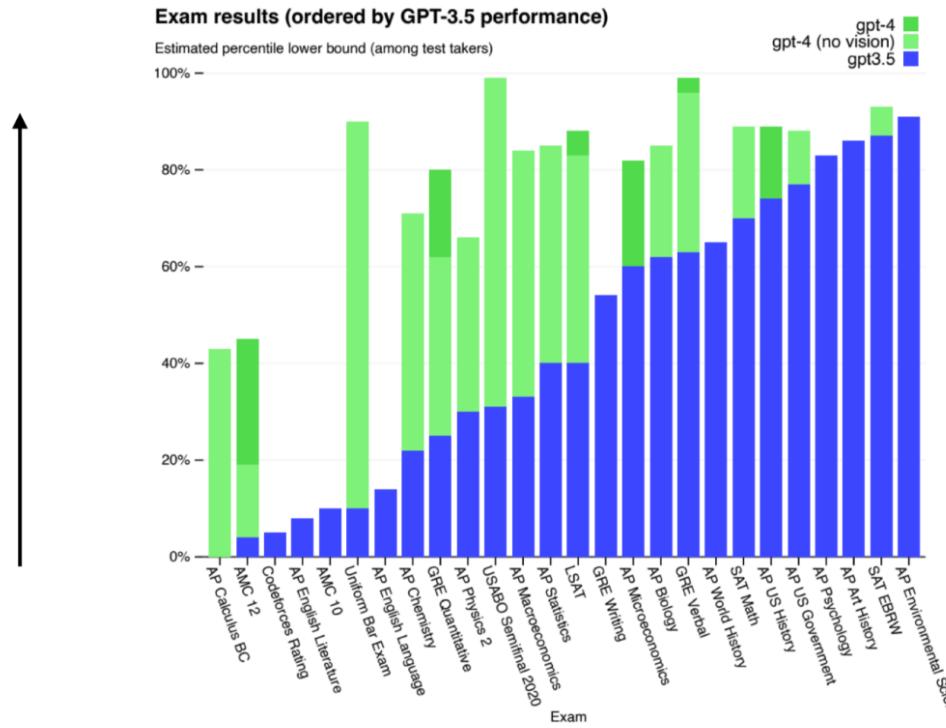
Performance of LLMs is a smooth, well-behaved, predictable function of:

- **N**, the number of parameters in the network
  - **D**, the amount of text we train on
- And the trends do not show signs of “topping out”

=> We can expect more intelligence “for free” by scaling



# We can expect a lot more “general capability” across all areas of knowledge:



# Why Larger language models

- More world **knowledge** (LAMA)
  - Language models as knowledge base?
- Larger capacity to learn problem-solving **Abilities**
  - Coding, revising articles, reasoning etc.
- Better **generalization** to unseen tasks

- **Emergent ability** (涌现能力)

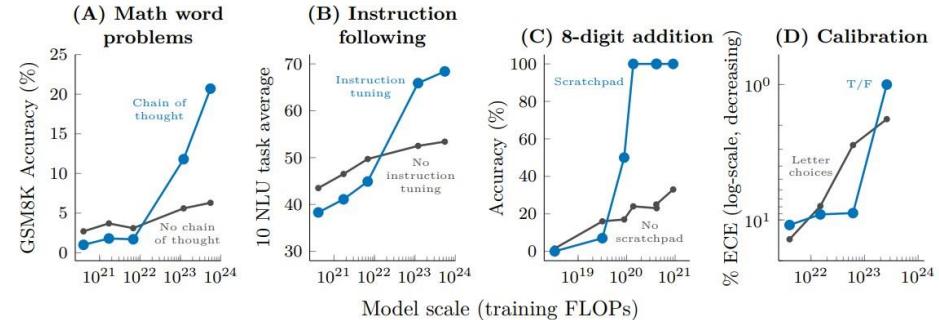
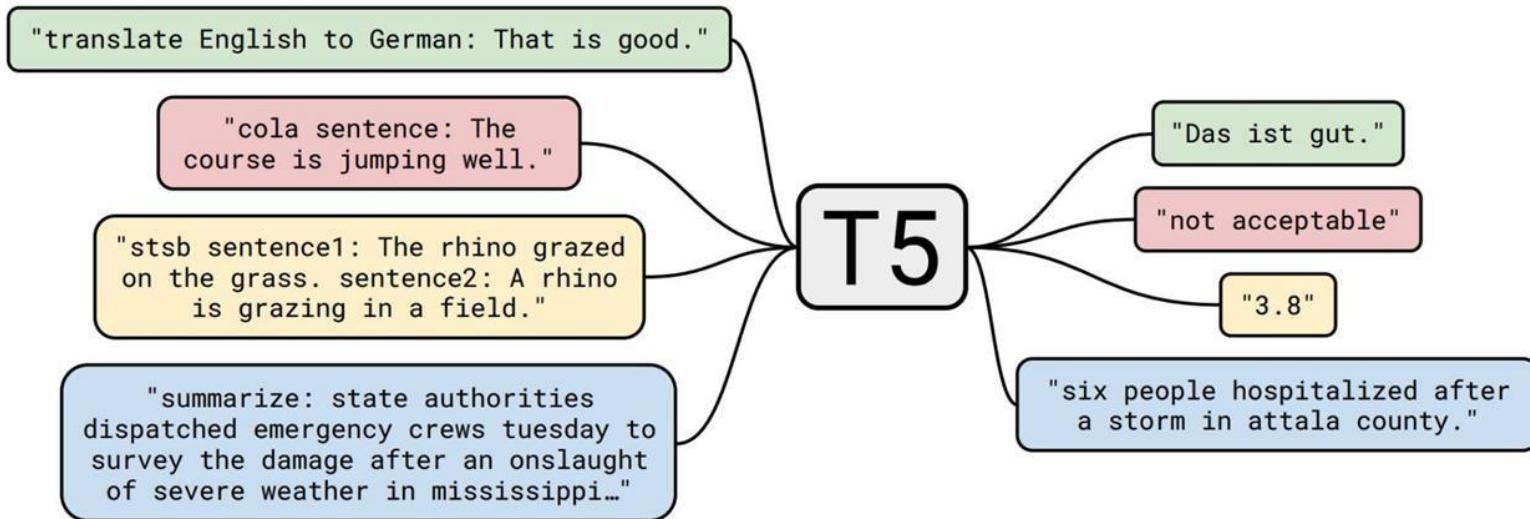


Figure 3: Specialized prompting or finetuning methods can be emergent in that they do not have a positive effect until a certain model scale. A: Wei et al. (2022b). B: Wei et al. (2022a). C: Nye et al. (2021). D: Kadavath et al. (2022). An analogous figure with number of parameters on the x-axis instead of training FLOPs is given in Figure 12. The model shown in A-C is LaMDA (Thoppilan et al., 2022), and the model shown in D is from Anthropic.

# Why LLMs?

## Generalization :

One single model to solve many NLP tasks



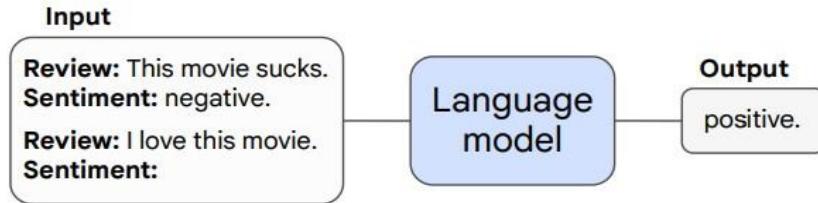
It could even generalize to new tasks, following the phylosity of FLAN

# Why LLMs?

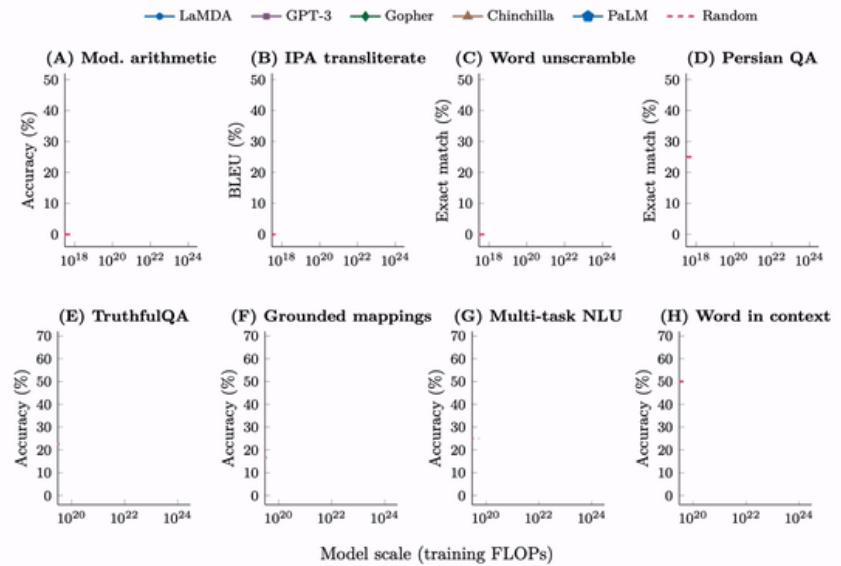
## Emergent properties in LLMs:

Some ability of LM is not present in smaller models but is present in larger models

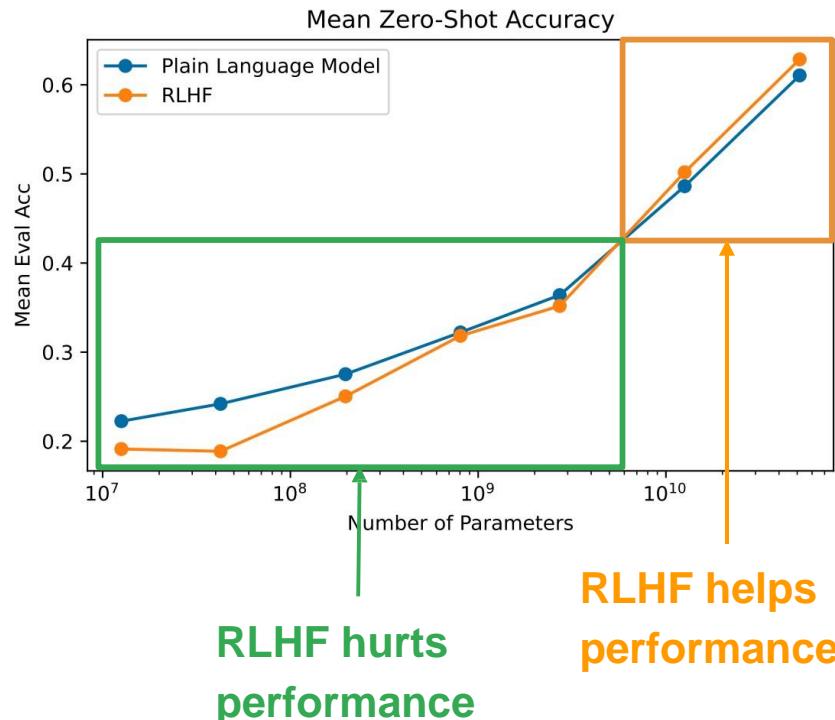
## Emergent Capability: Few-shot prompting and many others



>A few-shot prompted task is emergent if it achieves random accuracy for small models and above-random accuracy for large models.

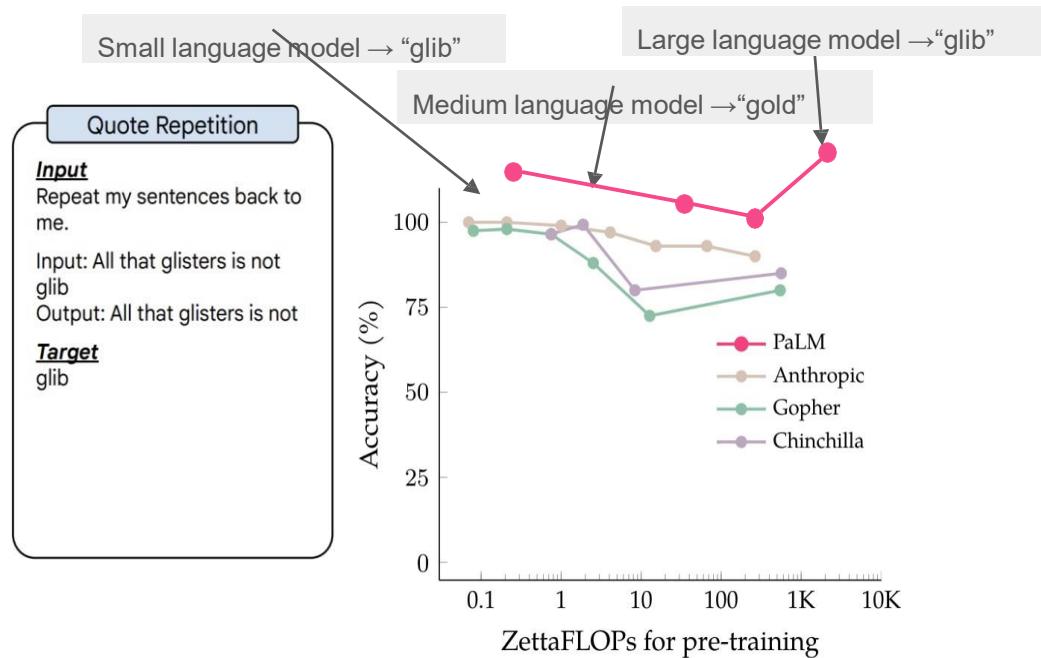


# Emergent ability: benefit from RLHF

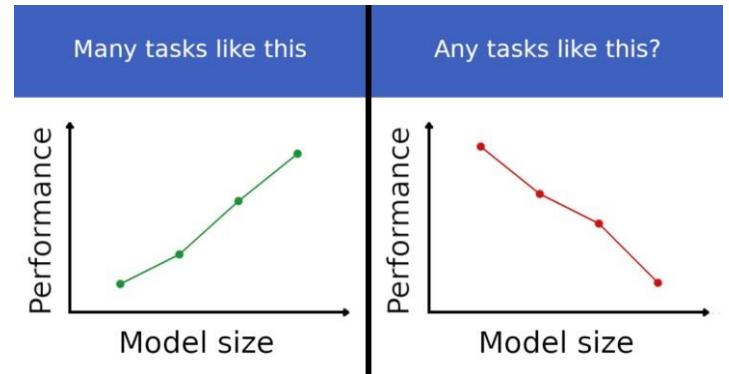


# To be or not to be Large?

Inverse scaling can become U-shaped: To be large ?



Inverse Scaling Prize: Not to be large?



See:

- ❖ [TruthfulQA](#): The largest models were generally the least truthful
- ❖ <https://github.com/inverse-scaling/prize>
- ❖ <https://irmckenzie.co.uk/round1>

# How to use LLMs?

*prompt Engineering, model fine-tuning and ~~training from scratch~~*

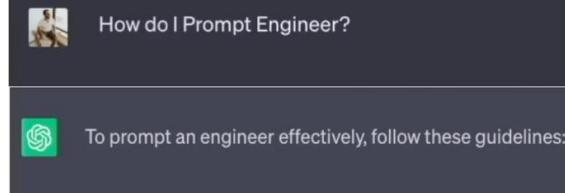
# How to use LLMs?

## *Level 1: Prompt Engineering*

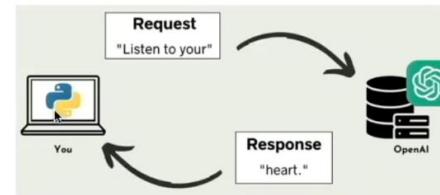
# Prompt Engineering

Using an LLM out-of-the-box (i.e. not changing any model parameters)

**Easy Way**  
(ChatGPT)



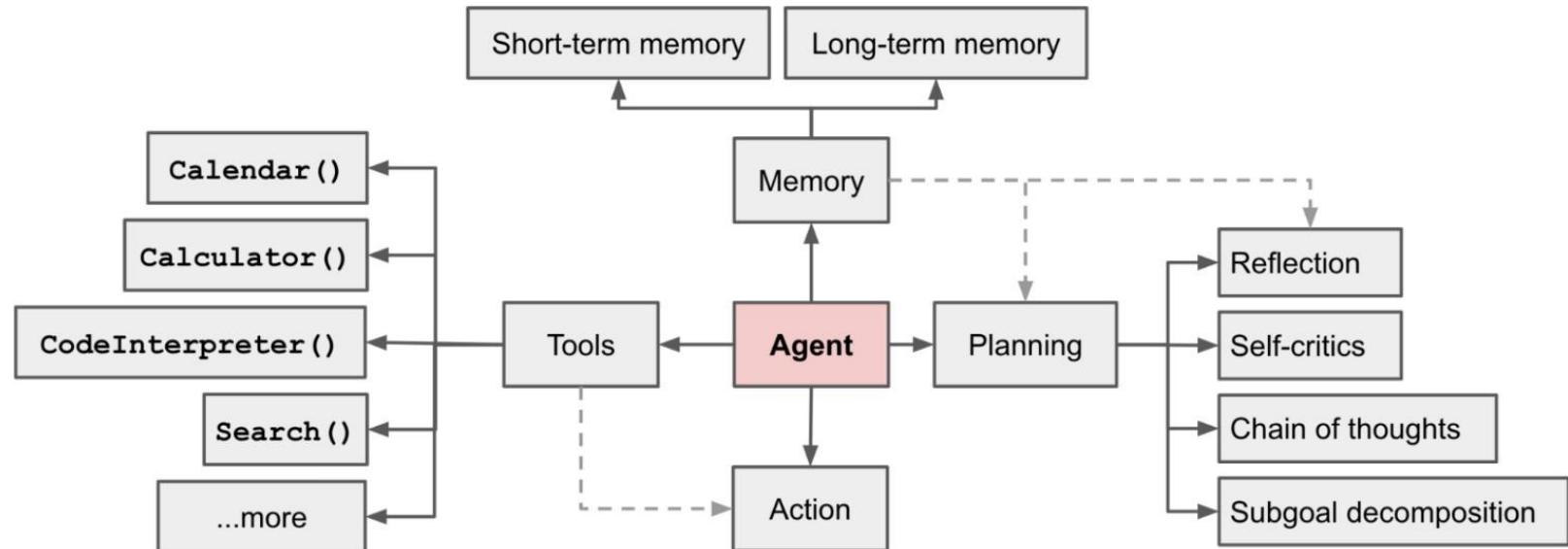
**Less Easy Way**  
(OpenAI API, Hugging Face)



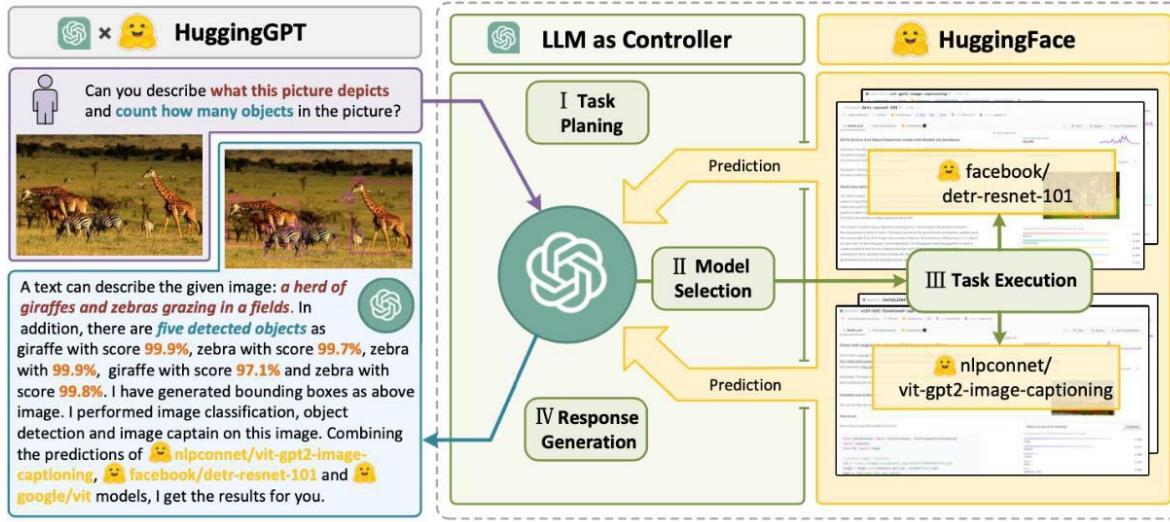
**Hugging Face**

# Agent

LLM acts as a Decision Center (Reasoning) and Human Interaction Front end (Chat)



# Agent: Tool use



**HuggingGPT** ([Shen et al. 2023](#)) is a framework to use ChatGPT as the task planner to select models available in HuggingFace platform according to the model descriptions and summarize the response based on the execution results.

## Algorithm 1 API call process

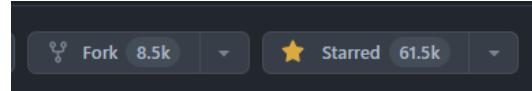
```

1: Input:  $us \leftarrow UserStatement$ 
2: if API Call is needed then
3:   while API not found do
4:      $keywords \leftarrow summarize(us)$ 
5:      $api \leftarrow search(keywords)$ 
6:     if Give Up then
7:       break
8:     end if
9:   end while
10:  if API found then
11:     $api\_doc \leftarrow api.documentation$ 
12:    while Response not satisfied do
13:       $api\_call \leftarrow gen\_api\_call(api\_doc, us)$ 
14:       $api\_re \leftarrow execute\_api\_call(api\_call)$ 
15:      if Give Up then
16:        break
17:      end if
18:    end while
19:  end if
20: end if
21: if response then
22:    $re \leftarrow generate\_response(api\_re)$ 
23: else
24:    $re \leftarrow generate\_response()$ 
25: end if
26: Output:  $ResponseToUser$ 

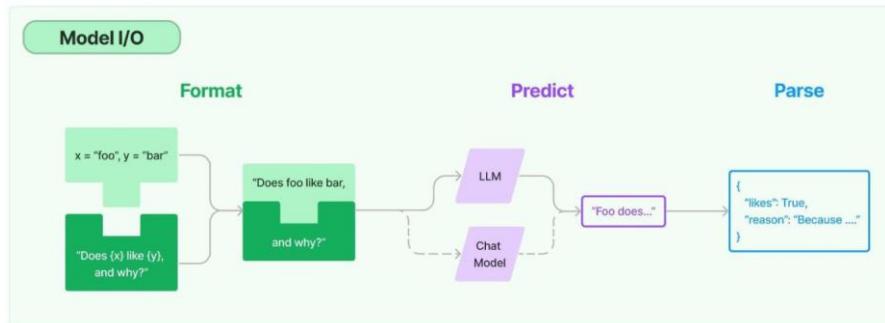
```

**API-Bank** ([Li et al. 2023](#)) : A benchmark for evaluating the performance of tool-augmented LLMs. It contains 53 commonly used API tools, a complete tool-augmented LLM workflow, and 264 annotated dialogues that involve 568 API calls.

# Langchain



- ❖ LangChain is a framework for developing applications powered by language models.
- ❖ The core building block of LangChain applications is the LLMChain. This combines three things:
  - LLM: The language model is the core reasoning engine here. In order to work with LangChain, you need to understand the different types of language models and how to work with them.
  - Prompt Templates: This provides instructions to the language model. This controls what the language model outputs, so understanding how to construct prompts and different prompting strategies is crucial.
  - Output Parsers: These translate the raw response from the LLM to a more workable format, making it easy to use the output downstream.



# How to use LLMs?

## *Level 2: Model Fine-tuning*

# Model Fine-tuning



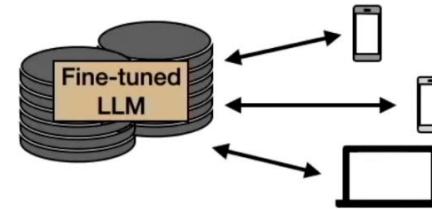
## Step 1

Obtain pre-trained  
LLM



## Step 2

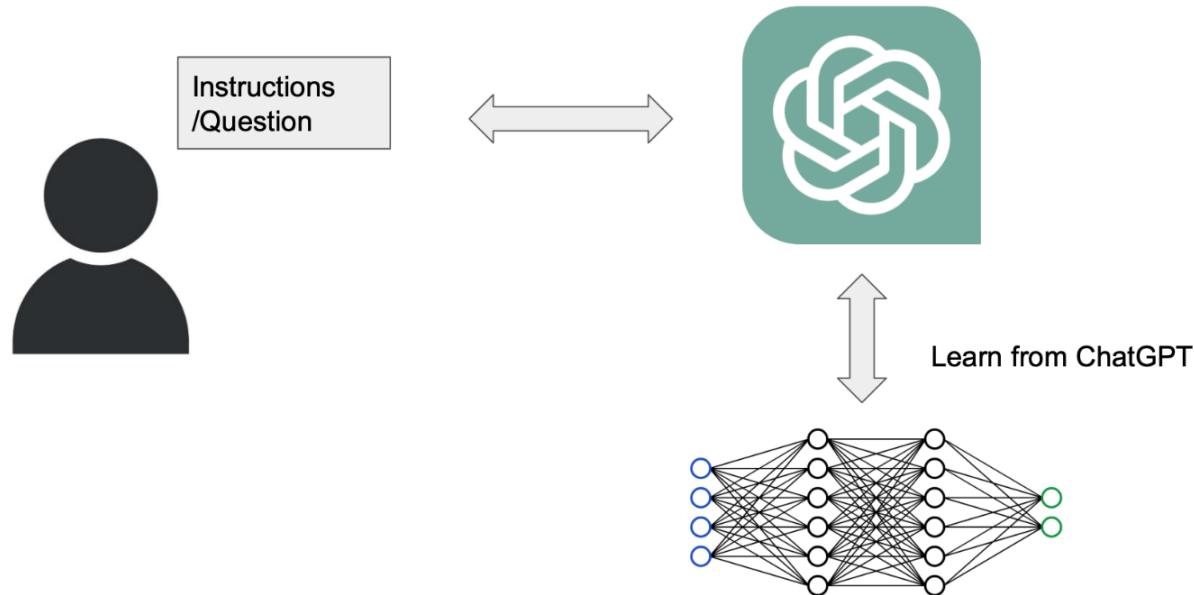
Update model  
parameters given task-  
specific examples



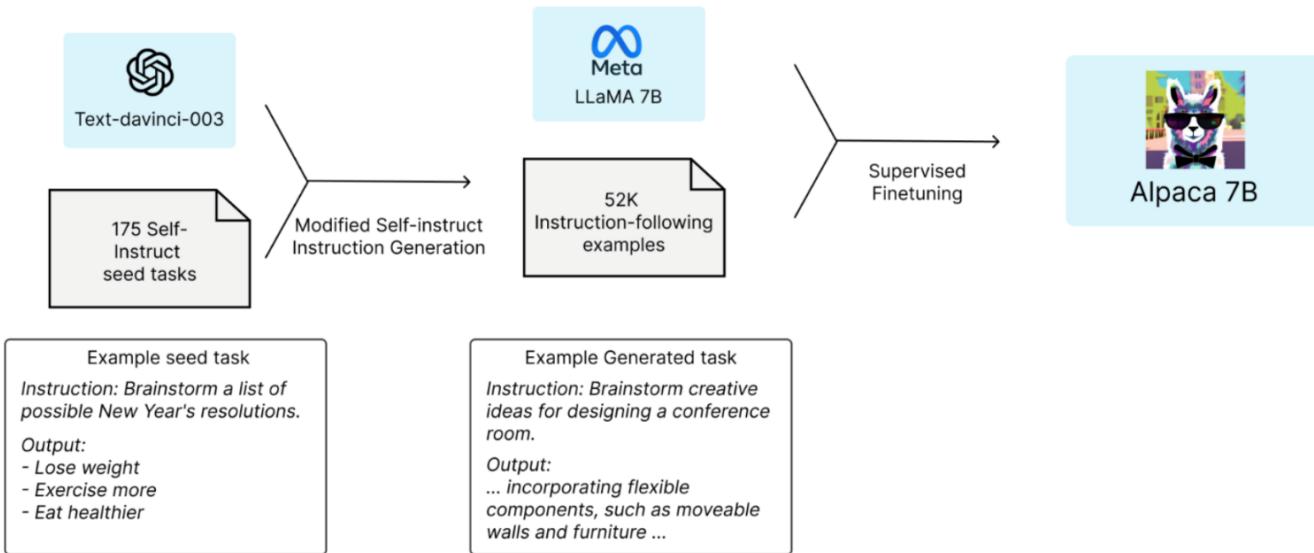
## Step 3

Profit?

# Shortcut: Distillation from ChatGPT



# Self-instruct for Data Augmentation



# Self-instruct for Data Augmentation

HQ

instruction: 请以下面词语为主题写一首诗

input: 夏天

output: 不但春妍夏亦佳，随缘花草是生涯。  
鹿葱解插纤长柄，金凤仍开最小花。

instruction: 给出下面词语的同义词

input: 惊人地

output: 令人惊奇地，令人惊讶地，意外地，难以置信地，不可思议地

instruction: 根据口内的提示，续写下面的内容

input: 我认为在夏天，狗狗可能喜欢吃西瓜、冰冻花生酱、【它们平时吃的食物】

output: 水管里的水、冰块、冷肉

仿照方面的内容重新造三个这样的数据



instruction: 请以下面词语为主题写一首诗

input: 雨夜

output: 滴滴雨声敲窗纸，漫天乌云掩月明。  
寂静深巷闻蛙鸣，春风摇曳水中影。

instruction: 给出下面词语的同义词

input: 快乐地

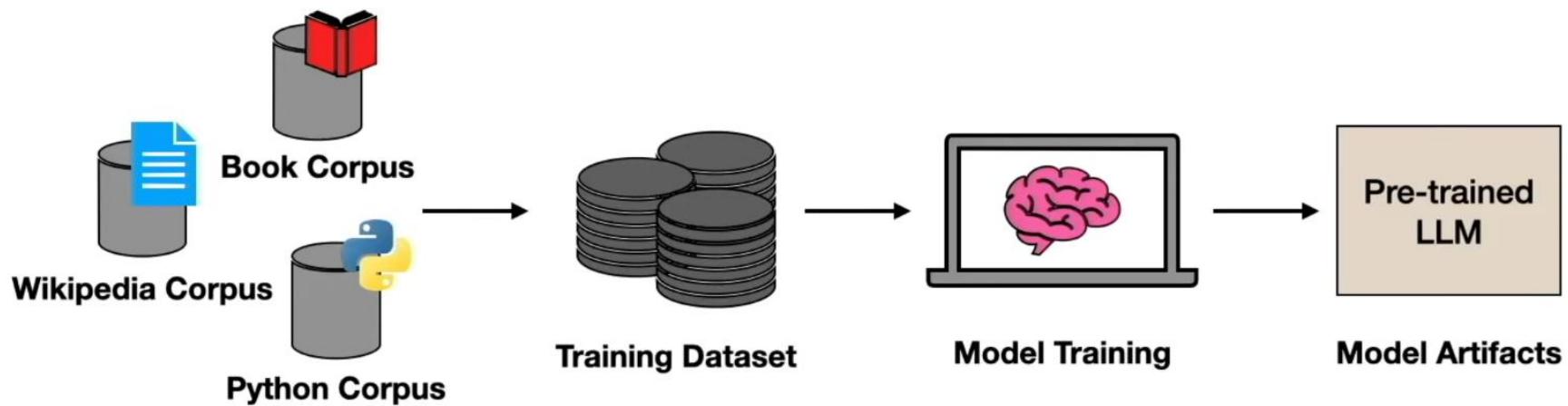
output: 欢乐地，愉快地，开心地，喜悦地，欢欣地

instruction: 根据口内的提示，续写下面的内容

input: 在冬天，猫咪喜欢蜷缩在暖和的地方，比如火炉旁、毛毯上、【它们的常去之处】

output: 窗台下、沙发角、主人的怀中

# Build the LLM



# Pretraining + Prompting Paradigm

- Fine-tuning (FT)
  - + Strongest performance
  - - Need curated and labeled dataset for each new task (typically 1k-100k ex.)
  - - Poor generalization, spurious feature exploitation
- Few-shot (FS)
  - + Much less task-specific data needed
  - + No spurious feature exploitation
  - - Challenging
- One-shot (1S)
  - + "Most natural," e.g. giving humans instructions
  - - Challenging
- Zero-shot (OS)
  - + Most convenient
  - - Challenging, can be ambiguous

Stronger  
task-specific  
performance



More convenient,  
general, less data

# Acknowledgement

- CSC6201/CIE6021: Large Language Models, Benyou Wang, CUHK-SZ
- CS224N/Ling284: Natural Language Processing with Deep Learning, Stanford University
- COS 597G: Understanding Large Language Models, Danqa Chen, Princeton University
- "Understanding Transformers, the Data Processing Units of the AI Age",  
<https://www.youtube.com/watch?v=zizonToFXDs>
- "The Power of Language Models: GPT-3 and Beyond",  
<https://www.youtube.com/watch?v=tFHeUSJAYbE&list=PLz-ep5RbHosU2hnz5ejezwaYpdMutMVB0>
- "Advancements in Natural Language Processing: Insights from AI Research",  
[https://www.youtube.com/watch?v=zjkBMFhNj\\_g&t=4s](https://www.youtube.com/watch?v=zjkBMFhNj_g&t=4s)

**Thanks**