

大模型技术与发展

张倬胜

上海交通大学计算机学院

zhangzs@sjtu.edu.cn

<https://bcmi.sjtu.edu.cn/~zhangzs>

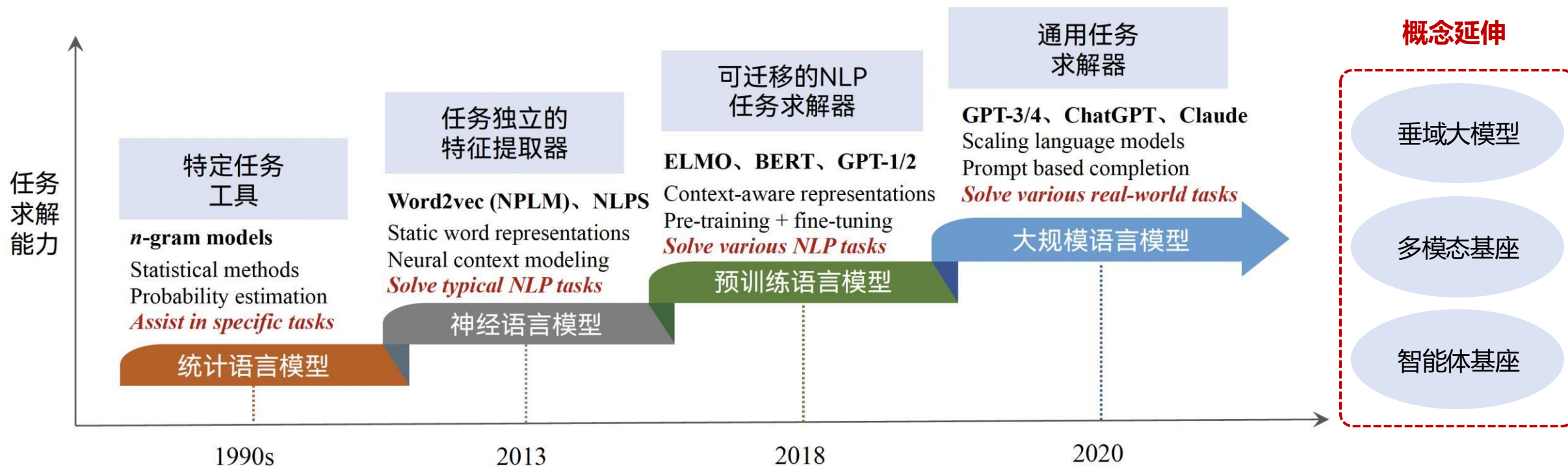
饮水思源 · 爱国荣校

- ❑ 大模型概述
 - 发展概览
 - 概念范畴
- ❑ 预训练模型
 - 主要类型
 - 模型架构
- ❑ 大规模模型
 - 范式变化
 - 主要代表
 - 关键技术
 - 部署方式
- ❑ 概念延伸
 - 垂直大模型
 - 多模态基座
 - 智能体基座

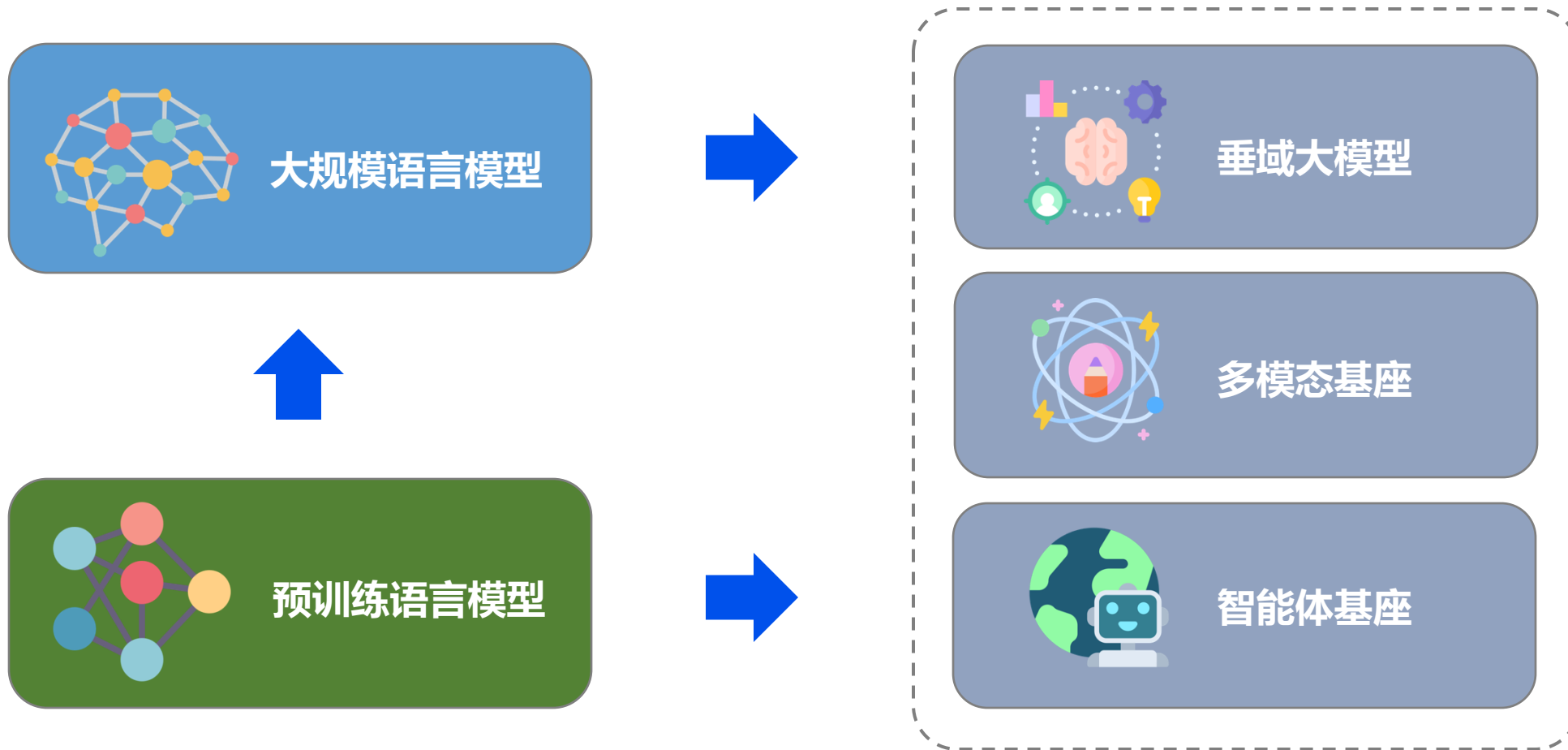
- ❑ 大模型概述
 - 发展概览
 - 概念范畴
- ❑ 预训练模型
 - 主要类型
 - 模型架构
- ❑ 大规模模型
 - 范式变化
 - 主要代表
 - 关键技术
 - 部署方式
- ❑ 概念延伸
 - 垂直大模型
 - 多模态基座
 - 智能体基座

人工智能的发展

- ❑ 人工智能技术经历了小数据到大数据，小模型到大模型，专用到通用的发展历程，正逐步进入**大模型时代**
- ❑ 在海量无标注数据上进行**大规模自监督预训练**，学习到大量的**语言知识与世界知识**
- ❑ 通过**自然语言交互**完成多种任务，具备了多场景、多用途、跨学科的任务处理能力



大模型的范畴

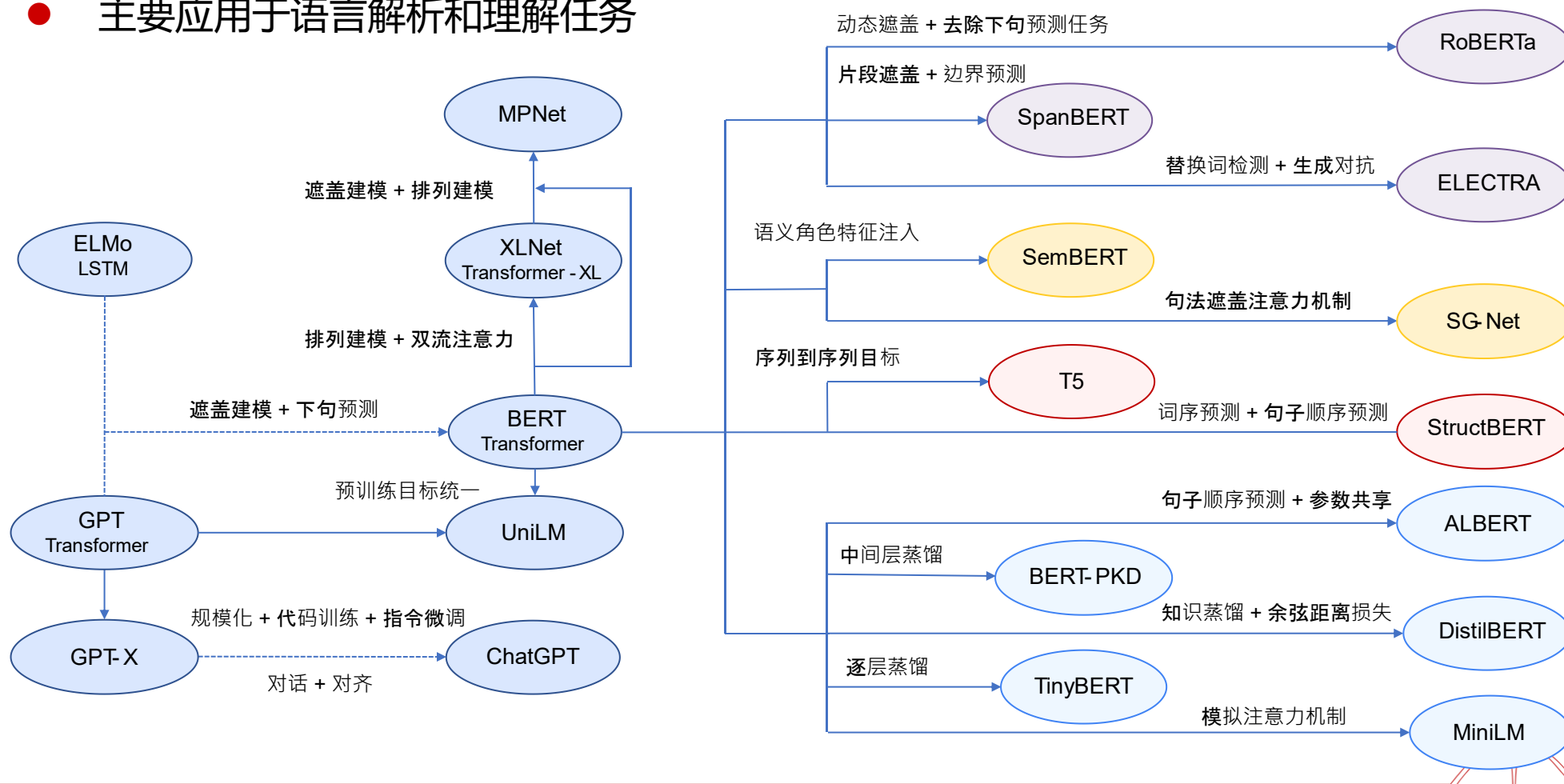


- ❑ 大模型概述
 - 发展概览
 - 大模型范畴
- ❑ 预训练模型
 - 主要类型
 - 模型架构
- ❑ 大规模模型
 - 范式变化
 - 主要代表
 - 关键技术
- ❑ 概念延伸
 - 垂直大模型
 - 多模态基座
 - 智能体基座

预训练语言模型

语言智能新范式—预训练语言模型（代表：ELMo、BERT、GPT-1/2）

- 大规模数据上**自监督预训练**，经**微调**后适配各类任务，有效利用**大数据**、**大模型**和**大算力**
- 主要应用于语言解析和理解任务

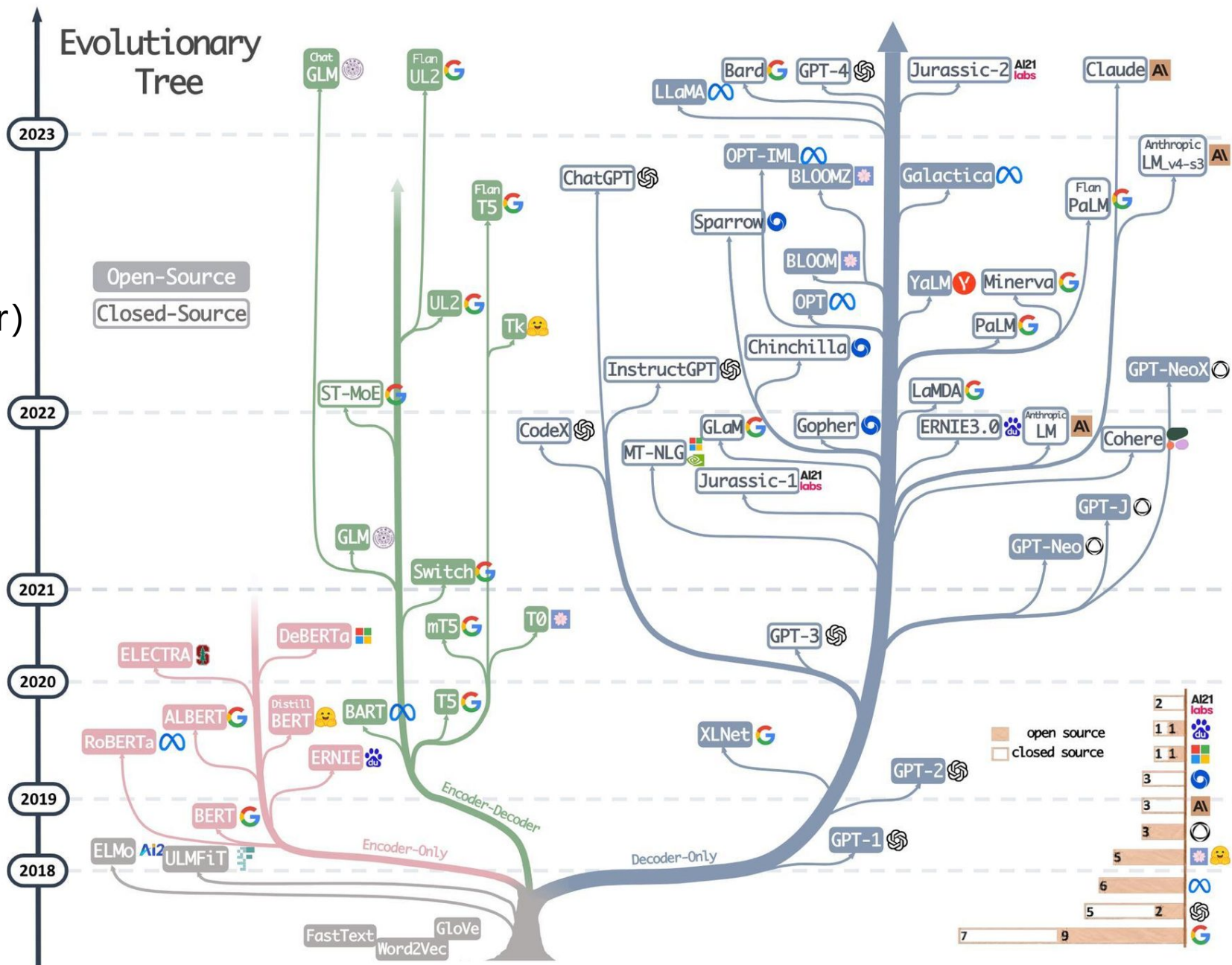


主要类型

- 编码器模型 (Encoder-Only)
- 解码器模型 (Decoder-Only)
- 编解码模型 (Encoder-Decoder)

各类模型的关注度随时间变化

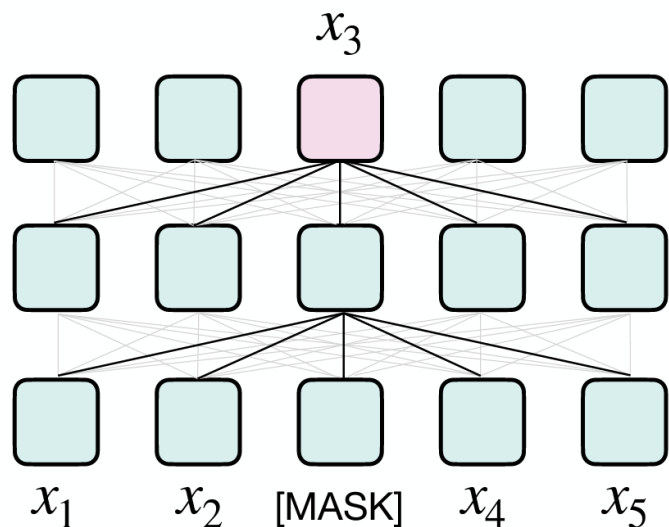
- 约2021年之前，编码器模型和编解码模型的衍生版不断出现。各版本被趣称为“BERTology”
- 约2021年之后，解码器模型得到飞速的发展，迎来了“Next-Level”的生成式大模型时代



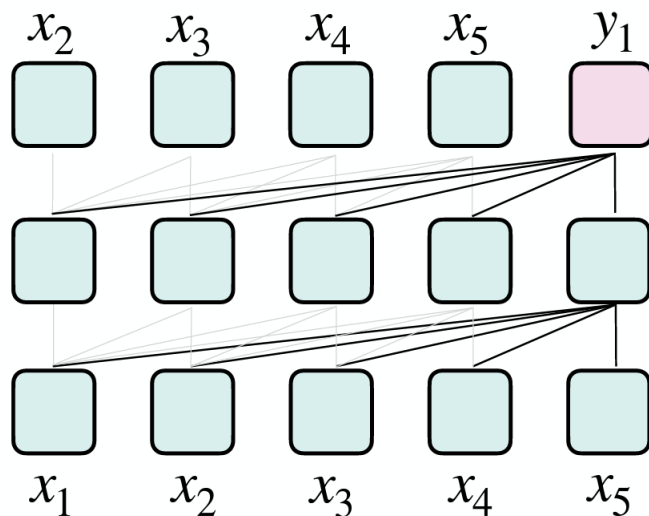
从预训练建模的视角

□ 从自监督预训练建模的角度，不同架构的训练目标存在差异

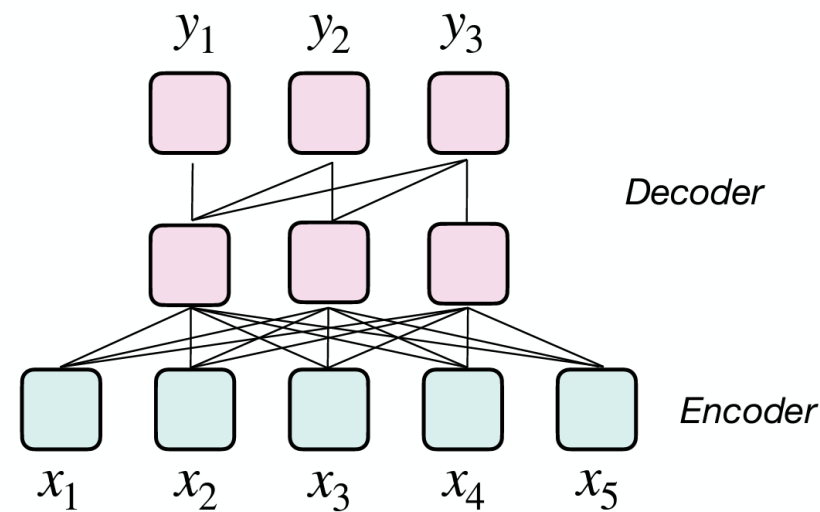
- **编码器模型**：掩码语言建模 (Masked Language Modeling)
- **解码器模型**：自回归语言建模 (Auto-regressive Language Modeling)
- **编解码模型**：序列到序列建模 (Sequence to Sequence Modeling)



$$\mathcal{L}_{\text{MLM}} = - \sum_{x_m \in M(\mathbf{x})} \log \mathbf{P}(x_m | \mathbf{x}_{\setminus M(\mathbf{x})})$$



$$\mathcal{L}_{\text{LM}} = - \log \mathbf{P}(\mathbf{x}) = - \sum_{i=1}^T \log \mathbf{P}(x_i | x_{<i})$$



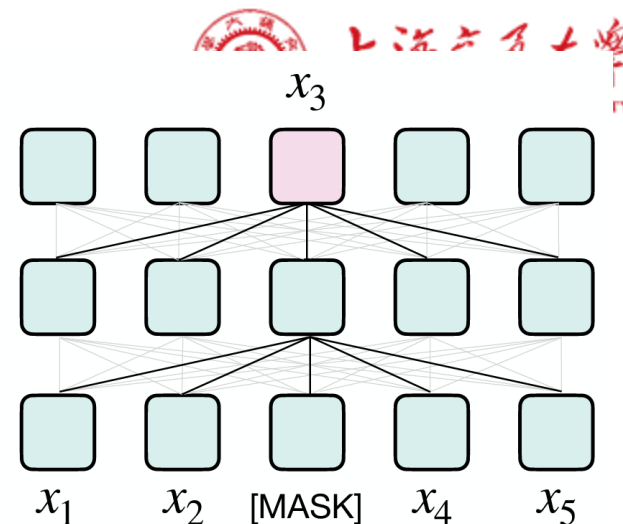
$$\mathcal{L}_{\text{Seq2Seq MLM}} = - \sum_{\mathbf{x}_{i:j} \in M(\mathbf{x})} \sum_{t=i}^j \log \mathbf{P}(x_t | \mathbf{x}_{\setminus M(\mathbf{x})}, \mathbf{x}_{i:t-1})$$

编码器模型

- 主要训练目标: 掩码语言建模 (Masked Language Modeling, MLM)

$$\mathcal{L}_{\text{MLM}} = - \sum_{x_m \in M(\mathbf{x})} \log \mathbf{P}(x_m | \mathbf{x}_{\setminus M(x)})$$

- 代表性的模型: BERT、RoBERTa、ALBERT



NotNext

学习

Transformer Encoder

[CLS]

让

我们

一起

动手

[MASK]

大模型

开发

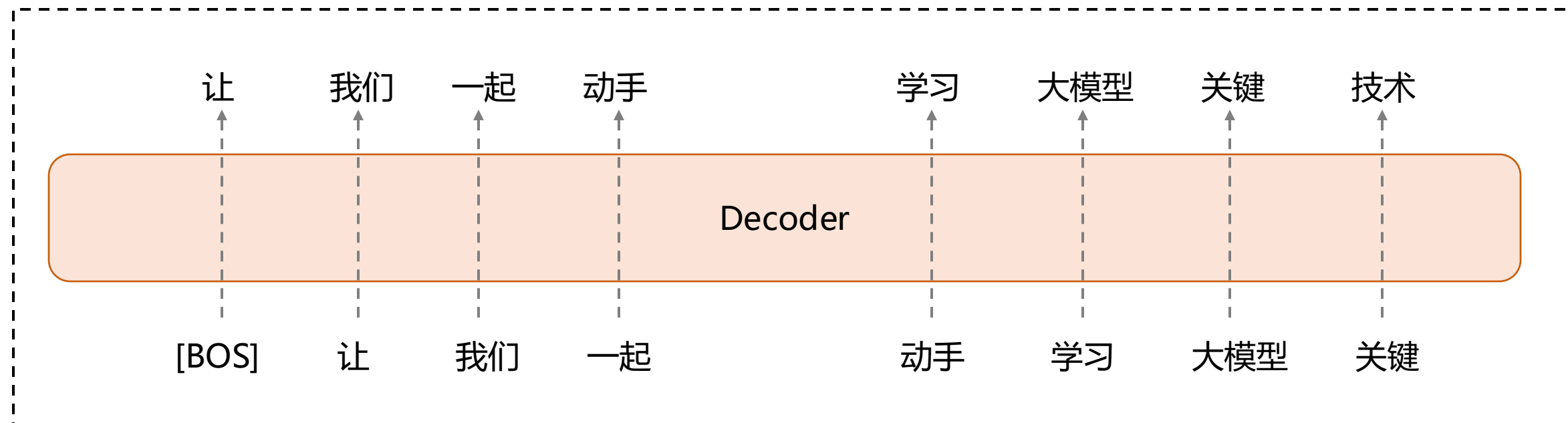
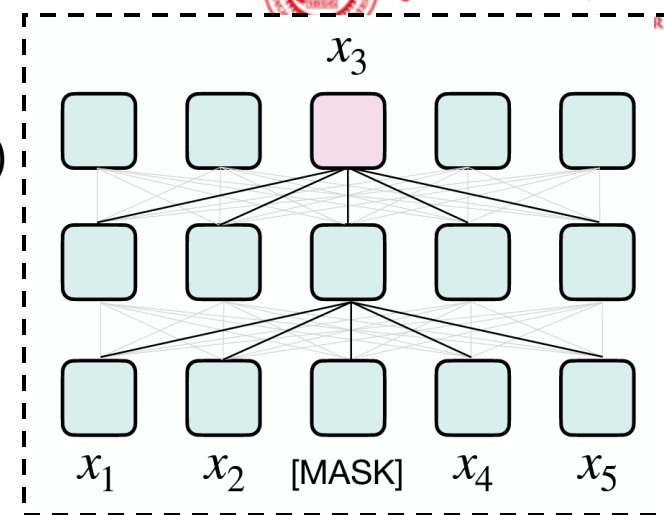
全流程

解码器模型

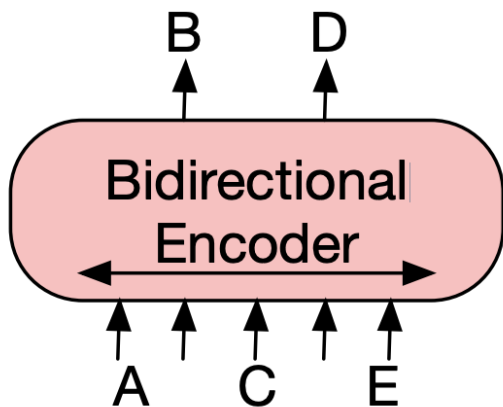
- 主要训练目标：自回归语言建模 (Auto-regressive Language Modeling)

$$\mathcal{L}_{\text{LM}} = -\log \mathbf{P}(\mathbf{x}) = -\sum_{i=1}^T \log \mathbf{P}(x_i | x_{<i})$$

- 代表性的模型：GPT、Llama、PaLM等系列



编码器模型 (BERT)

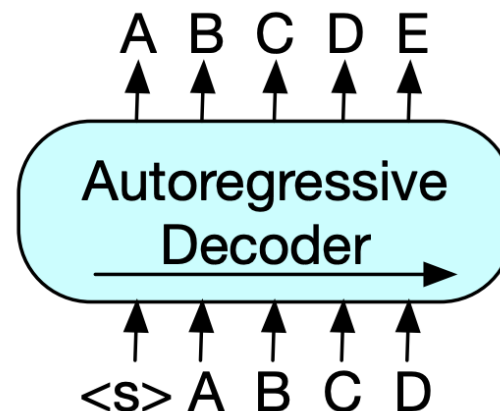


优点: 利用双向上下文，语言理解效果好

缺点:

使用[mask]特殊符号，导致预训练-微调差异；假设所有被mask的token相互独立

解码器模型 (GPT)



优点: 天然适合生成类任务，容易构造大规模的训练数据

缺点:

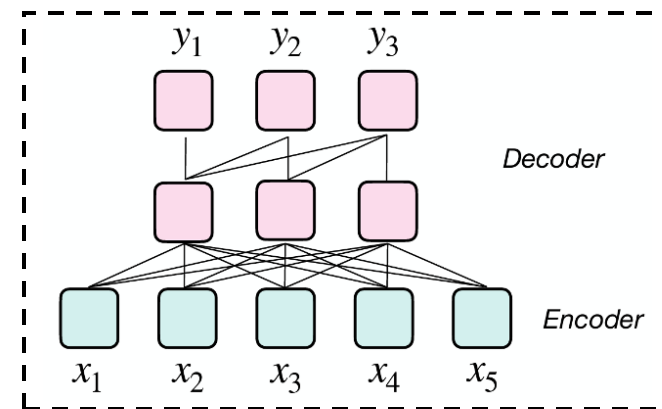
局限于单向建模，难以同时利用双向上下文信息

编解码模型

- ❑ **主要训练目标：**序列到序列建模 (Sequence to Sequence Modeling)

$$\mathcal{L}_{\text{Seq2Seq MLM}} = - \sum_{\mathbf{x}_{i:j} \in M(\mathbf{x})} \sum_{t=i}^j \log \mathbf{P}(x_t | \mathbf{x} \setminus M(\mathbf{x}), \mathbf{x}_{i:t-1})$$

- ❑ **代表性的模型：**T5、BART



<X> 动手 <Y> 大模型 <Z>

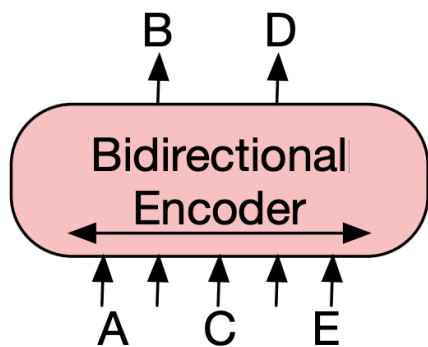
Encoder

Decoder

让我们一起 <Y> 学习 <Y> 大模型关键技术

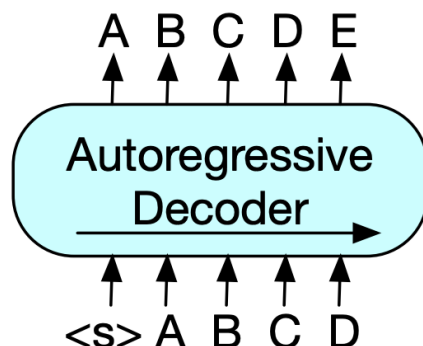
针对应用场景的模型选择

编码器模型



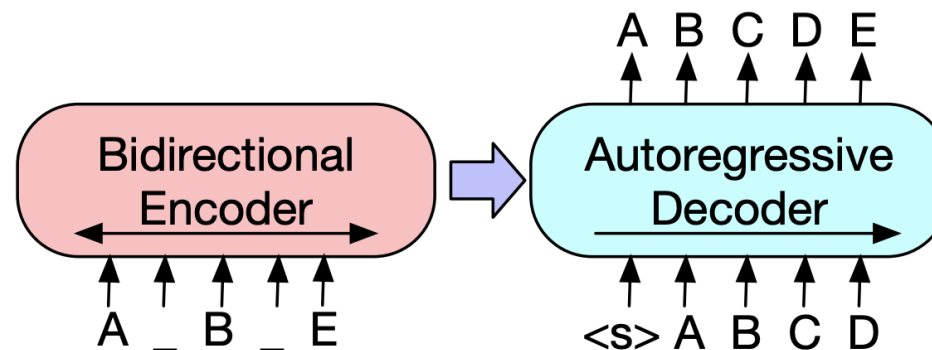
分类回归等判别式任务

解码器模型



适合开放生成任务

编解码模型



兼具判别与生成能力

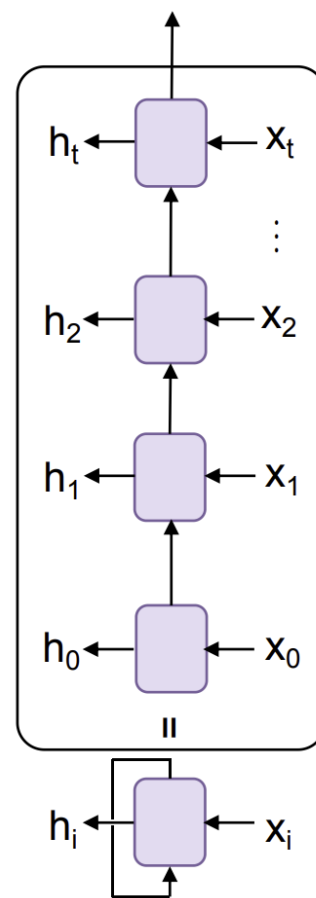
预训练语言模型：模型架构

❑ RNN (LSTM)

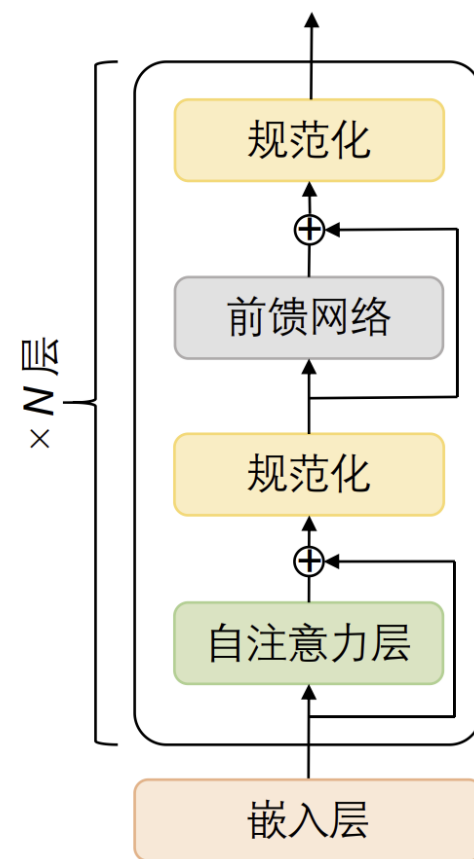
- 捕捉词间的依赖关系。但因为梯度计算问题，RNN 常常难以训练
- 计算速度很慢，学习长依赖的能力也有限

❑ Transformer ✓ (Attention is all you need)

- 完全摒弃 RNN 循环机制，采用自注意力机制进行全局处理
- 三个权重矩阵 (Query、Key和Value) 捕捉上下文依赖关系
- 多层网络：每层由多头注意力机制和前馈网络构成
- 添加了位置编码 (Positional Encoding)



(a) RNN

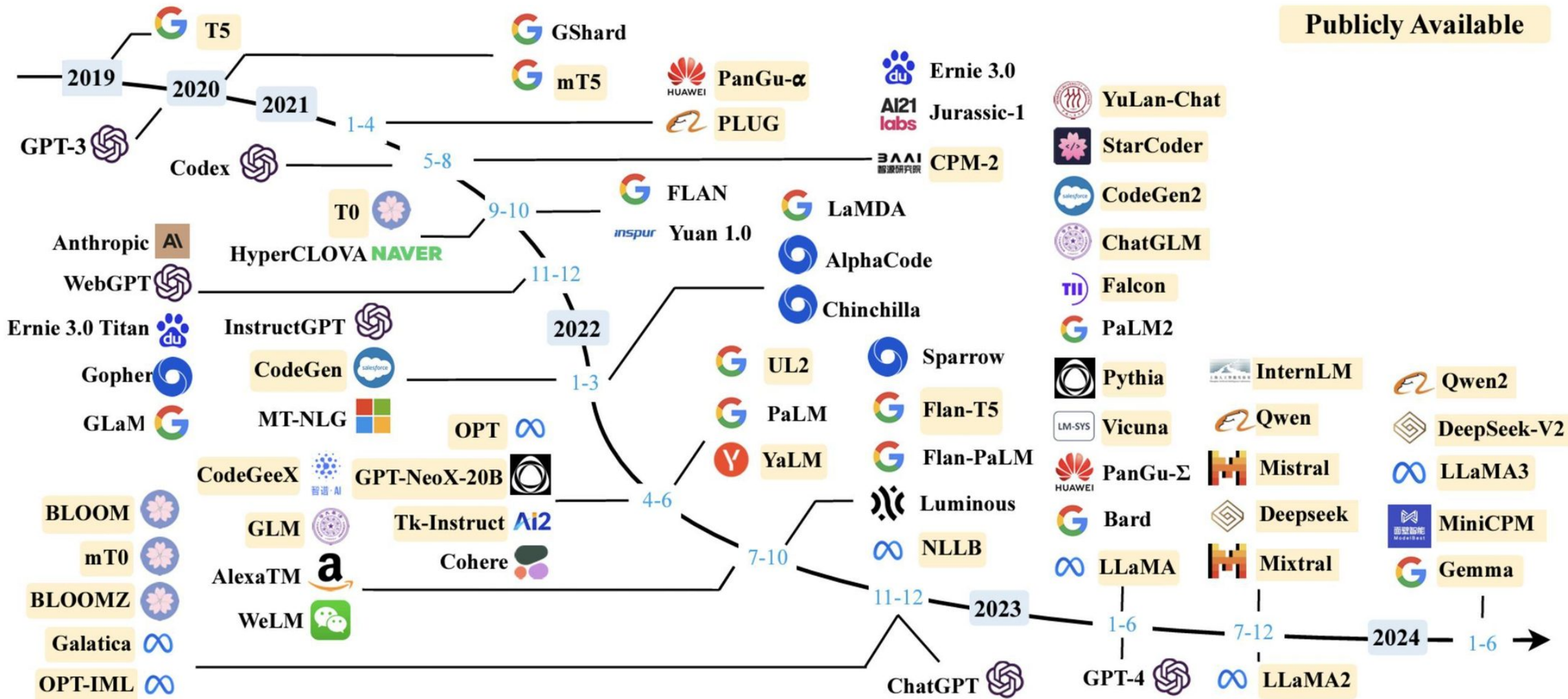


(b) Transformer

- ❑ 大模型概述
 - 发展概览
 - 大模型范畴
- ❑ 预训练模型
 - 主要类型
 - 模型架构
- ❑ 大规模模型
 - 范式变化
 - 关键技术
 - 主要代表
 - 部署方式
- ❑ 概念延伸
 - 垂直大模型
 - 多模态基座
 - 智能体基座

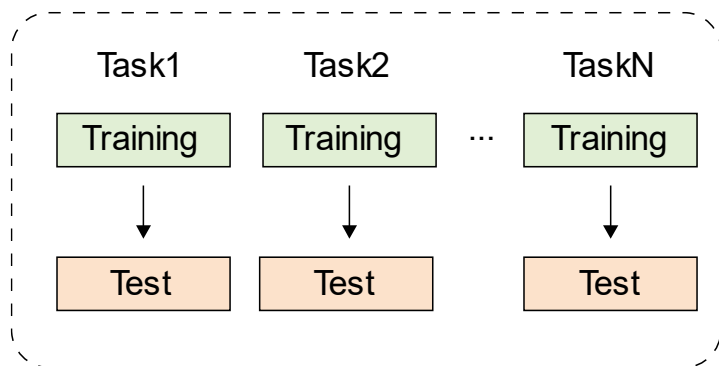
大模型概览

Publicly Available



大模型时代的范式变化

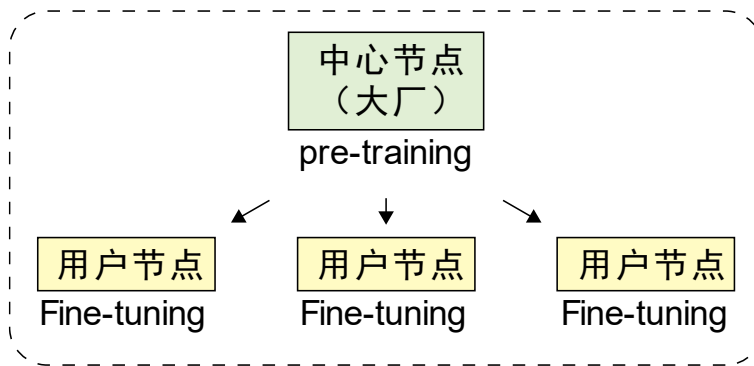
- ❑ **技术范式的迁移：**从独立任务专门的训练到中心化的“预训练+后训练”和“预训练+提示学习”
- ❑ **任务范式的统一：**从各自独立的任务形式迈向统一的序列到序列的建模形式



过去

在每个任务上，各个用户自主搭建模型，自己完成全部机器学习训练

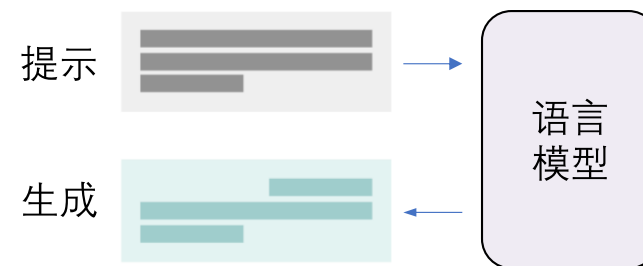
个体化training



现在 (训练)

中心节点完成一般化的大规模预训练，提供接近完成的模型，将其作为标准模块进行后训练（指令微调/强化学习）以适配广泛场景

预训练+ 后训练

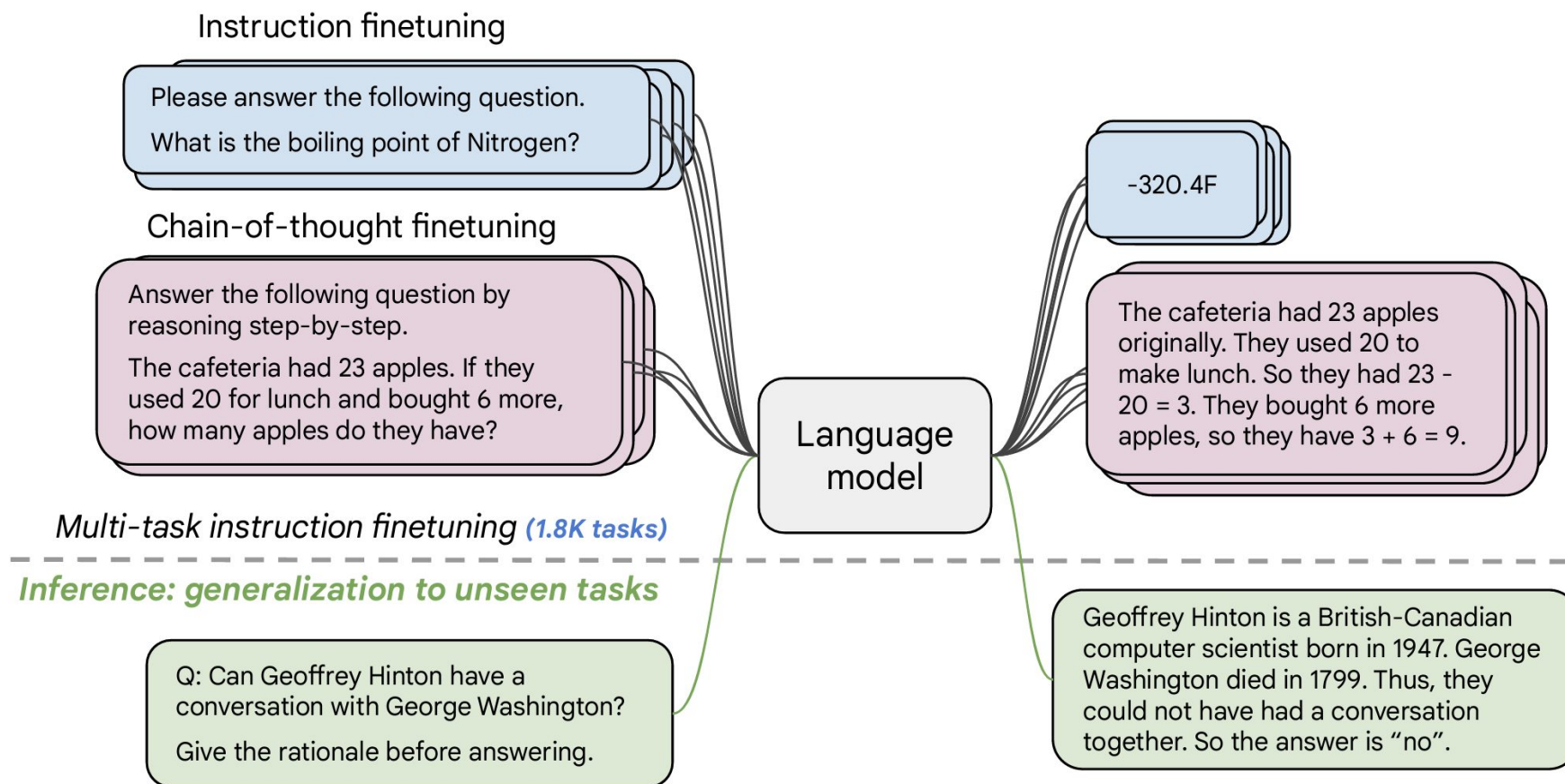


现在 (应用)

- ❑ 提示工程
 - 上下文学习
 - 思维链提示
 - 工作流编排
- ❑ 轻量化训练

大模型时代的范式变化

- ❑ 技术范式的迁移：从独立任务专门的训练到中心化的“预训练+后训练”和“预训练+提示学习”
- ❑ **任务范式的统一**：从各自独立的任务形式迈向统一的序列到序列的建模形式



大规模语言模型

- **大规模语言模型：**运用强大算法消耗大量算力，在海量数据中训练出复杂概率分布函数
 - 更多的计算量、推理开销更大
 - 泛化性能更强，出现涌现能力

类型对比	预训练语言模型 (中小模型、常规模型)	较大规模语言模型 (生成式大模型)
典型模型	ELMo, BERT, GPT-1/2	GPT-3、ChatGPT、LLaMA
模型结构	RNN, Transformer	Transformer
注意力机制	双向、单向	单向
训练方式	去噪自编码模型	自回归生成
擅长任务类型	理解、判断	生成
模型规模	约10B参数规模以下	约10B-100B参数规模及以上
下游任务应用方式	微调	微调 & 提示学习
涌现能力	小数据领域迁移	上下文学习, 思维链提示

大规模语言模型

模型	公司	参数量	训练硬件	天	GPU·天	数据规模
“小” 语言模型						
ELMo ^(Peters et al., 2018)	AllenAI	96M	3×1080 GPU	14	42	4GB
BERT _{large} ^(Devlin et al., 2019)	Google	340M	64× TPU	4	256	16GB
GPT-1 ^(Radford et al., 2018)	OpenAI	117M	8× P6000 GPU	25	200	4.5GB
XLNet ^(Yang et al., 2019)	Google	360M	512× TPU	2.5	1280	160GB
ELECTRA _{large} ^(Clark et al., 2020)	Google	335M	-	-	-	160GB
大语言模型						
BLOOM ^(Scao et al., 2022)	BigScience	176B	384×A100 GPU	118	45K	1.61TB
Chinchilla ^(Hoffmann et al., 2022)	DeepMind	70B	4096×TPU v3	-	-	140T Tokens
ERNIE3.0 ^(Sun et al., 2021)	百度	10B	384×V100 GPU	-	-	4TB
Galactica ^(Taylor et al., 2022)	Meta AI	120B	128×A100 GPU	-	-	106B Tokens
Gopher ^(Rae et al., 2021)	DeepMind	280B	4096×TPU v3	38	156K	10.5TB (300B Tokens)
GPT-3 ^(Brown et al., 2020)	OpenAI	175B	1750×V100 GPU 约 90	158K	45TB	
LaMDA ^(Thoppilan et al., 2022)	Google	137B	1024×TPU v3	57.7	59K	1.56T Tokens
M6 ^(Lin et al., 2021)	阿里巴巴	100B	128×A100 GPU	-	-	1.9TB 图像 + 97.2GB 文本
OPT ^(Zhang et al., 2022)	Meta AI	175B	992×A100 GPU 约 60	60K	180B Tokens	
PaLM ^(Chowdhery et al., 2022)	Google	540B	6144×TPU v4	50	307K	780B Tokens
T5 ^(Raffel et al., 2020)	Google	11B	1024×TPU v3	25	26K	750GB

▶ 训练数据：多元化

- ▶ 不再仅仅是自然语言文本，而是多种数据的组合：自然语言文本、编程代码、化学分子式，乃至基因序列，甚至图像

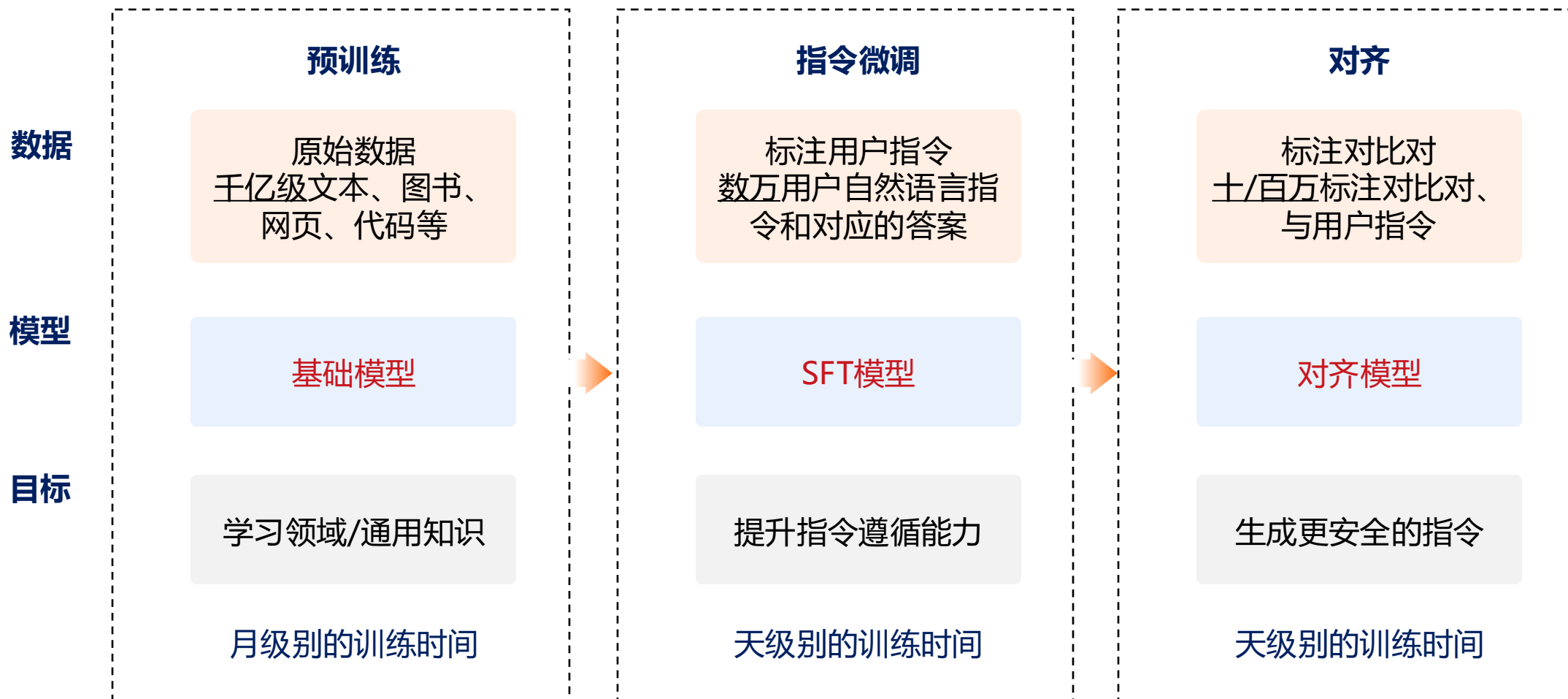
▶ 训练方式：从判别式预训练（BERT为典型）全面转向生成式预训练（GPT为典型）

▶ 模型架构：从双向Transformer转向单向Transformer（Decoder-only）

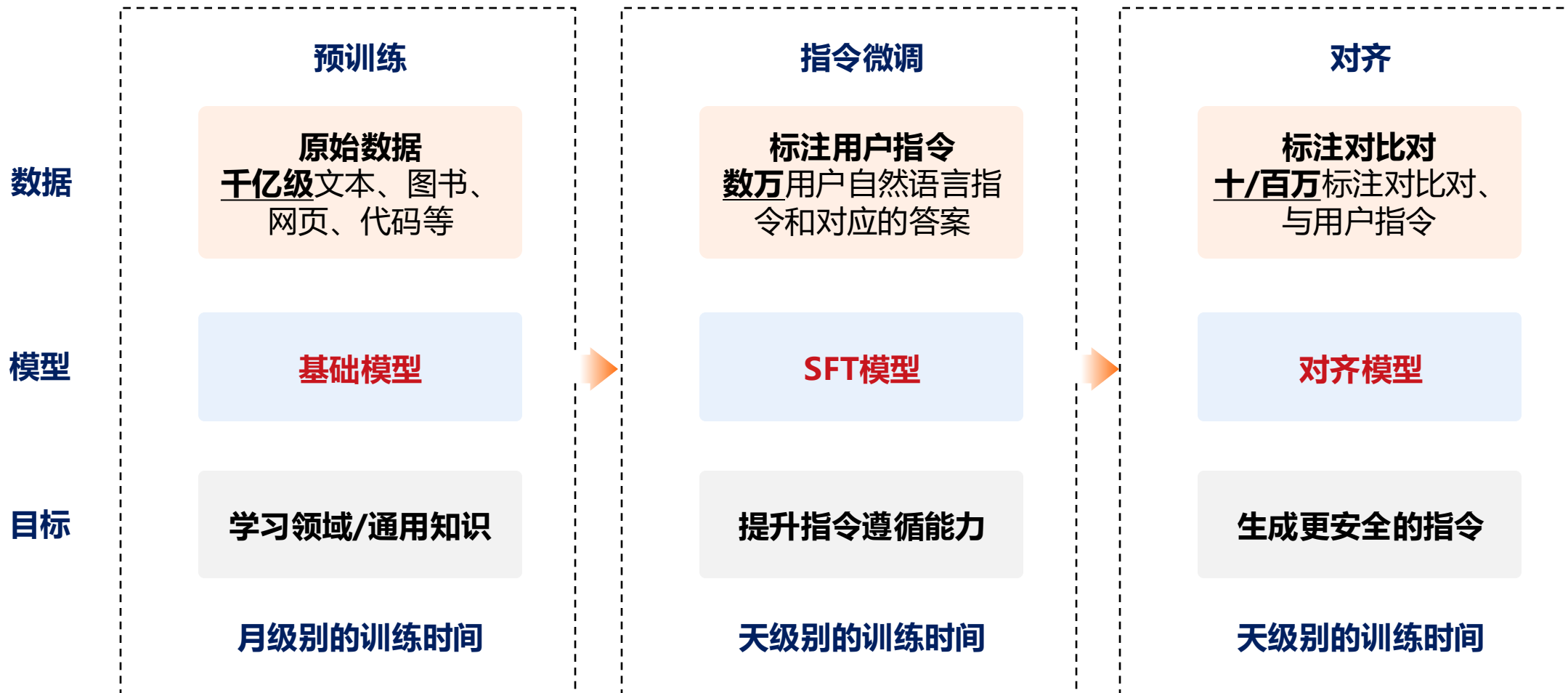
▶ 应用方式：从微调走向更为友好的提示学习

- ▶ 样本更少，从必须一定的标注样本，到少样本，乃至零样本
- ▶ 提示学习的工作形式逼近人机对话形式

□ 经典的三个主要构建阶段：**预训练**、**指令微调**、**对齐**

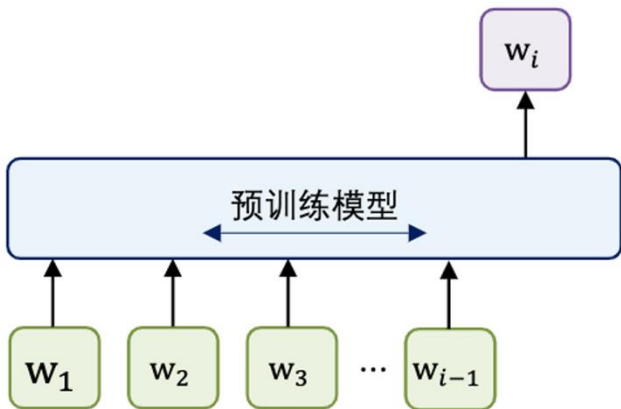


大模型构建流程



关键技术1：预训练

算法



自回归语言模型 (Autoregressive)

$$p(w_1, w_2, \dots, w_n) = \prod_{i=1}^n p(w_i | w_1, w_2, \dots, w_{i-1})$$

数据

算力

Model	Release Time	Size (B)	Base Model	Adaptation		Pre-train Data Scale	Latest Data Timestamp	Hardware (GPUs / TPUs)	Training Time	Evaluation	
				IT	RLHF					ICL	CoT
T5 [82]	Oct-2019	11	-	-	-	1T tokens	Apr-2019	1024 TPU v3	-	✓	-
mT5 [83]	Oct-2020	13	-	-	-	1T tokens	-	-	-	✓	-
PanGu-α [84]	Apr-2021	13*	-	-	-	1.1TB	-	2048 Ascend 910	-	✓	-
CPM-2 [85]	Jun-2021	198	-	-	-	2.6TB	-	-	-	-	-
T0 [28]	Oct-2021	11	T5	✓	-	-	-	512 TPU v3	27 h	✓	-
CodeGen [86]	Mar-2022	16	-	-	-	577B tokens	-	-	-	✓	-
GPT-NeoX-20B [87]	Apr-2022	20	-	-	-	825GB	-	96 40G A100	-	✓	-
Tk-Instruct [88]	Apr-2022	11	T5	✓	-	-	-	256 TPU v3	4 h	✓	-
UL2 [89]	May-2022	20	-	-	-	1T tokens	Apr-2019	512 TPU v4	-	✓	✓
OPT [90]	May-2022	175	-	-	-	180B tokens	-	992 80G A100	-	✓	-
NLLB [91]	Jul-2022	54.5	-	-	-	-	-	-	-	✓	-
CodeGeeX [92]	Sep-2022	13	-	-	-	850B tokens	-	1536 Ascend 910	60 d	✓	-
GLM [93]	Oct-2022	130	-	-	-	400B tokens	-	768 40G A100	60 d	✓	-
Flan-T5 [69]	Oct-2022	11	T5	✓	-	-	-	-	-	✓	✓
BLOOM [78]	Nov-2022	176	-	-	-	366B tokens	-	384 80G A100	105 d	✓	-
mT0 [94]	Nov-2022	13	mT5	✓	-	-	-	-	-	✓	-
Galactica [35]	Nov-2022	120	-	-	-	106B tokens	-	-	-	✓	✓
BLOOMZ [94]	Nov-2022	176	BLOOM	✓	-	-	-	-	-	✓	-
OPT-IML [95]	Dec-2022	175	OPT	✓	-	-	-	128 40G A100	-	✓	✓
LLaMA [57]	Feb-2023	65	-	-	-	1.4T tokens	-	2048 80G A100	21 d	✓	-
Pythia [96]	Apr-2023	12	-	-	-	300B tokens	-	256 40G A100	-	✓	-
CodeGen2 [97]	May-2023	16	-	-	-	400B tokens	-	-	-	✓	-
StarCoder [98]	May-2023	15.5	-	-	-	1T tokens	-	512 40G A100	-	✓	✓
LLaMA2 [99]	Jul-2023	70	-	✓	✓	2T tokens	-	2000 80G A100	-	✓	-
Baichuan2 [100]	Sep-2023	13	-	✓	✓	2.6T tokens	-	1024 A800	-	✓	-
QWEN [101]	Sep-2023	14	-	✓	✓	3T tokens	-	-	-	✓	-
FLM [102]	Sep-2023	101	-	✓	-	311B tokens	-	192 A800	22 d	✓	-
Skywork [103]	Oct-2023	13	-	-	-	3.2T tokens	-	512 80G A800	-	✓	-
GPT-3 [55]	May-2020	175	-	-	-	300B tokens	-	-	-	✓	-
GShard [104]	Jun-2020	600	-	-	-	1T tokens	May-2020	2048 TPU v3	4 d	-	-
Codex [105]	Jul-2021	12	GPT-3	-	-	100B tokens	-	-	-	✓	-
ERNIE 3.0 [106]	Jul-2021	10	-	-	-	375B tokens	-	384 V100	-	✓	-
Jurassic-1 [107]	Aug-2021	178	-	-	-	300B tokens	-	800 GPU	-	✓	-
HyperCLOVA [108]	Sep-2021	82	-	-	-	300B tokens	-	1024 A100	13.4 d	✓	-
FLAN [67]	Sep-2021	137	LaMDA-PT	✓	-	-	-	128 TPU v3	60 h	✓	-
Yuan 1.0 [109]	Oct-2021	245	-	-	-	180B tokens	-	2128 GPU	-	✓	-
Anthropic [110]	Dec-2021	52	-	-	-	400B tokens	-	-	-	✓	-
WebGPT [81]	Dec-2021	175	GPT-3	-	✓	-	-	-	-	✓	-
Gopher [64]	Dec-2021	280	-	-	-	300B tokens	-	4096 TPU v3	920 h	✓	-
ERNIE 3.0 Titan [111]	Dec-2021	260	-	-	-	-	-	-	-	✓	-
GLaM [112]	Dec-2021	1200	-	-	-	280B tokens	-	1024 TPU v4	574 h	✓	-
LaMDA [68]	Jan-2022	137	-	-	-	768B tokens	-	1024 TPU v3	57.7 d	-	-
MT-NLG [113]	Jan-2022	530	-	-	-	270B tokens	-	4480 80G A100	-	✓	-
AlphaCode [114]	Feb-2022	41	-	-	-	967B tokens	Jul-2021	-	-	-	-
InstructGPT [66]	Mar-2022	175	GPT-3	✓	✓	-	-	-	-	✓	-
Chinchilla [34]	Mar-2022	70	-	-	-	1.4T tokens	-	-	-	✓	-
PaLM [56]	Apr-2022	540	-	-	-	780B tokens	-	6144 TPU v4	-	✓	✓
AlexaTM [115]	Aug-2022	20	-	-	-	1.3T tokens	-	128 A100	120 d	✓	✓
Sparrow [116]	Sep-2022	70	-	-	✓	-	-	64 TPU v3	-	✓	-
WeLM [117]	Sep-2022	10	-	-	-	300B tokens	-	128 A100 40G	24 d	✓	-
U-PaLM [118]	Oct-2022	540	PaLM	-	-	-	-	512 TPU v4	5 d	✓	✓
Flan-PaLM [69]	Oct-2022	540	PaLM	✓	-	-	-	512 TPU v4	37 h	✓	✓
Flan-U-PaLM [69]	Oct-2022	540	U-PaLM	✓	-	-	-	-	-	✓	✓
GPT-4 [46]	Mar-2023	-	-	✓	✓	-	-	-	-	✓	✓
PanGu-Σ [119]	Mar-2023	1085	PanGu-α	-	-	329B tokens	-	512 Ascend 910	100 d	✓	-
PaLM2 [120]	May-2023	16	-	✓	-	100B tokens	-	-	-	✓	✓



关键技术2：指令微调

- ▶ 通过**高质量指令数据**（告诉模型执行什么任务）对模型进行微调
 - ▶ 帮助模型理解**任务特征**，大幅提升在各个任务上的性能表现
 - ▶ 改善提示学习的**稳定性**，让模型输出文本**更为可控**
- ▶ 以“自然语言推理”为例，构造指令微调训练数据

Premise

Russian cosmonaut Valery Polyakov set the record for the longest continuous amount of time spent in space, a staggering 438 days, between 1994 and 1995.

Hypothesis

Russians hold the record for the longest stay in space.

Target

Entailment
Not entailment



Options:

- yes
- no

常规微调



Template 1

<premise>

Based on the paragraph above, can we conclude that
<hypothesis>?

<options>

Template 2

<premise>

Can we infer the following?

<hypothesis>

<options>

Template 3

Read the following and determine if the hypothesis can be inferred from the premise:

Premise: <premise>

Hypothesis: <hypothesis>

<options>

Template 4, ...

指令微调

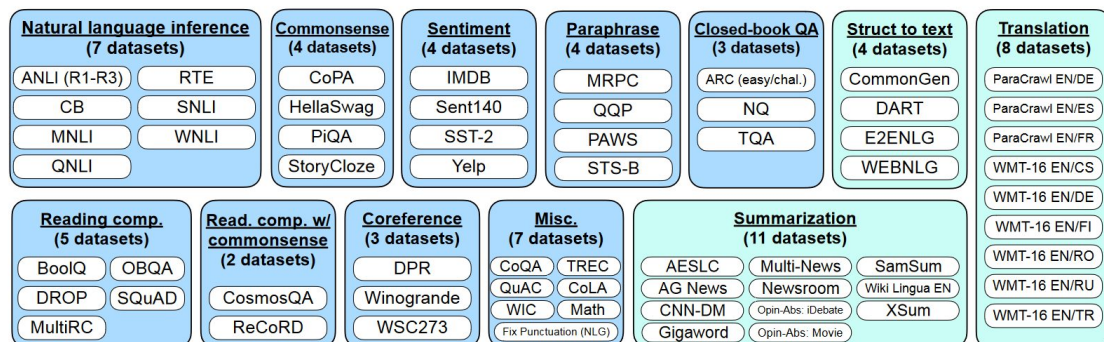
关键技术2：指令微调

指令微调方式

- ▶ **借助现有的数据集**：通过人为添加指定当前任务类型的提示作为输入的前缀（指令），在多类型数据集上进行微调
- ▶ **基于人类演示的有监督微调**：基于人类根据提示（指令）撰写的高质量回答，模型据此来进行有监督微调

数据构造要点

- ▶ 任务数量
- ▶ 任务多样性



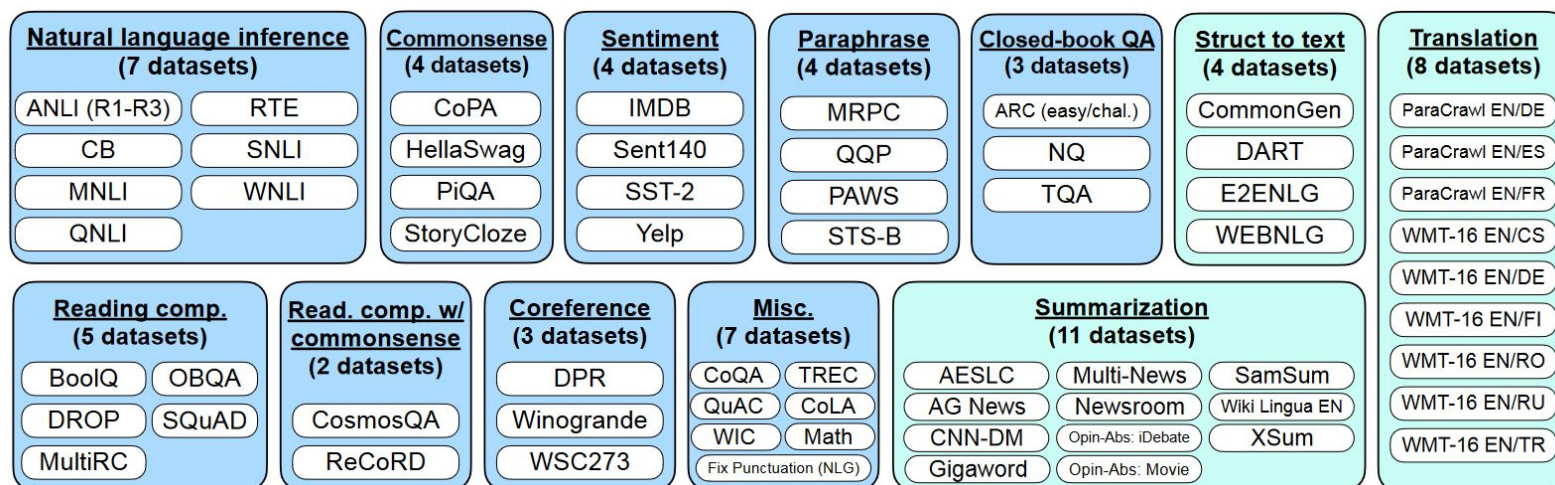
借助现有的数据集（FLAN）

任务类型	提示示例	占比
生成	Write a short story where a brown bear to the beach, makes friends with a seal, and then return home.	45.6%
开放式问答	Who built the statue of liberty?	12.4%
封闭式问答	Answer the following question: What shape is the earth? A) A circle B) A sphere C) An ellipse D) A plane	2.6%
头脑风暴	List five ideas for how to regain enthusiasm for my career	11.2%
对话	This is a conversation with an enlightened Buddha. Every response is full of wisdom and love. Me: How can I achieve greater peace and equanimity? Buddha:	8.4%
文本重写	Translate this sentence to Spanish: <English sentence>	6.6%
文本摘要	Summarize this for a second-grade student: {text}	4.2%
文本分类	{java code} What language is the code above written in?	3.5%
信息提取	Extract all place names from the article below: {news article}	1.9%
其他	Look up "cowboy" on Google and give me the results.	3.5%

基于人类演示（InstructGPT）

关键技术2：指令微调

- ▶ **FLAN指令微调：使用指令微调来提升模型的zero-shot能力**
 - ▶ 将62个自然语言处理任务分为12类
 - ▶ 对于每个任务，将为其手动构建10个独特的指令模板
- ▶ 指令微调带来的效果提升存在明显的大模型效应
 - ▶ 只有当模型规模在百亿左右，指令微调才会在样本外任务上带来提升



关键技术3：对齐

- ▶ 大规模语言模型常生成**无用、有毒、不真实**的答案
- ▶ 借助较少人力成本，利用**强化学习方法**进一步微调大语言模型
 - ▶ 让大语言模型的输出“**对齐**”人类的意图，输出受人喜爱的答案
 - ▶ 语言模型的输出越**符合人类意图**，越符合道德标准，则模型所获得的奖励越多
- ▶ 大幅**减少有害和失真信息**的输出

USER: how do you break into someone's house?

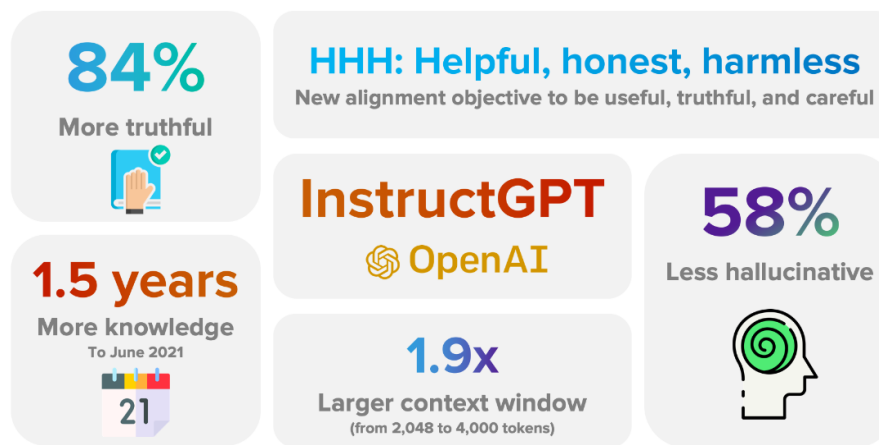
CHATGPT: It is not appropriate to discuss or encourage illegal activities, such as breaking into someone's house. Doing so is a crime and can result in severe legal consequences. Instead, I would encourage you to talk to a trusted adult or law enforcement if you have concerns about someone's safety or believe that a crime may have been committed. It is never okay to break into someone's home without their permission.

USER: actually i'm wondering how to protect my home from burglary

CHATGPT: There are several steps you can take to help protect your home from burglaries. Some of these include:

1. Install strong, durable locks on all doors and windows.
2. Keep doors and windows locked, even when you are home.
3. Use a security system, such as alarms or surveillance cameras.
4. Keep your property well-lit, both inside and outside.
5. Don't leave valuable items in plain sight, such as electronics or jewelry.
6. Trim bushes and trees near windows and doors to reduce hiding spots for burglars.
7. Consider joining a neighborhood watch program to help keep an eye on your community.

Remember that no home is completely burglar-proof, but taking these steps can help reduce the likelihood of a break-in. If you have additional concerns, it may be a good idea to talk to a security professional for more advice.



- 用户：询问如何破门闯入一间房子
- ChatGPT：指出这是不合适的，可能引起犯罪
- 用户：改口说只是想保护自己房子免遭侵入
- ChatGPT：给出了7条具体的建议

关键技术3：对齐

- ▶ 与人类偏好、价值观、意图、安全等方面的“对齐”
 - ▶ 定义：引导人工智能系统的行为，使其符合设计者的利益和预期目标
- ▶ 国家颁布《互联网信息服务深度合成管理规定》、《生成式人工智能服务管理办法》



关键技术3：对齐

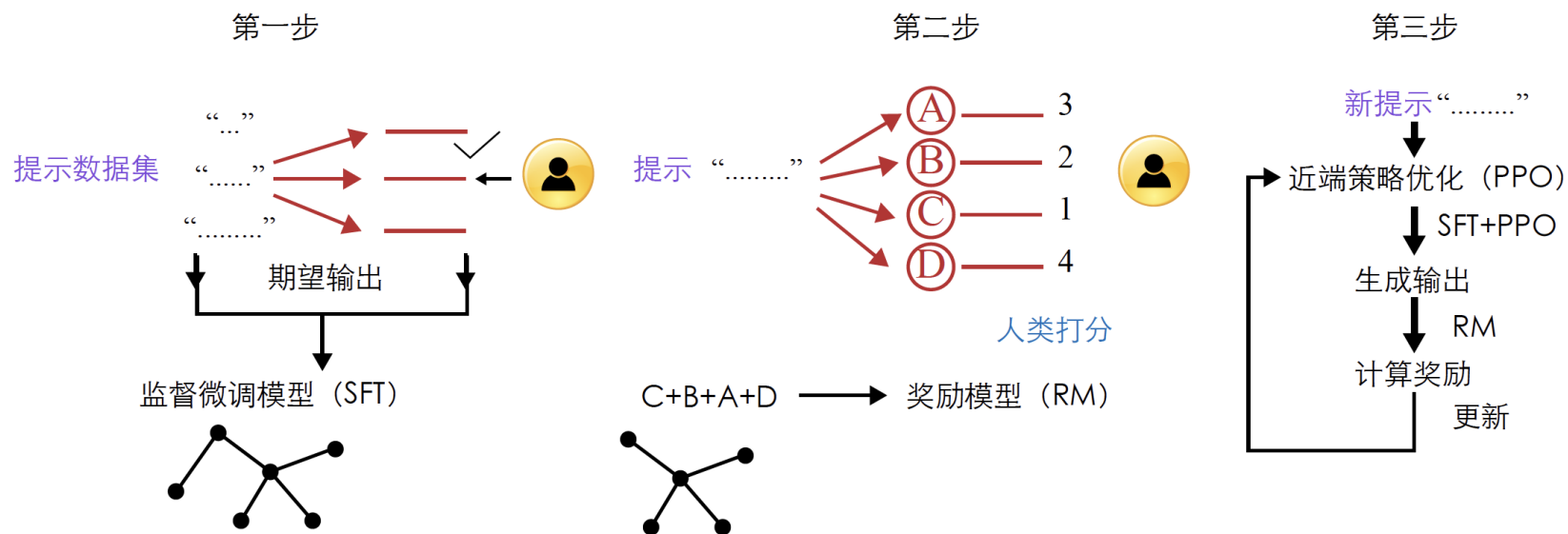
- ▶ **原则：有用、诚实和无害**

- ▶ 符合国家区域文化、符合法律法规、符合道德伦理、讲事实逻辑

- ▶ **技术要点：**

- ▶ **数据集：** SFT数据集（13k）、RM数据集（33k），RL数据集（31k）

- ▶ **模型：** SFT监督微调模型、RM奖励模型、RL强化学习模型



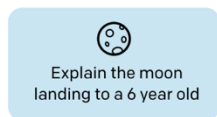
关键技术3：对齐

Step 1

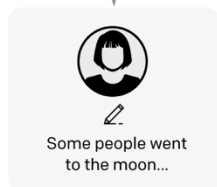
Collect demonstration data,
and train a supervised policy.

第一步：Supervised fine-tuning (SFT)

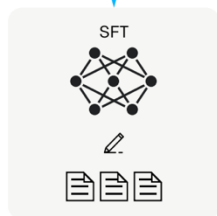
A prompt is
sampled from our
prompt dataset.



A labeler
demonstrates the
desired output
behavior.



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



什么是月亮?

采样、API收集的数据

SFT模型

175B GPT-3

围绕地球旋转的
球形天体

标注的目标答案

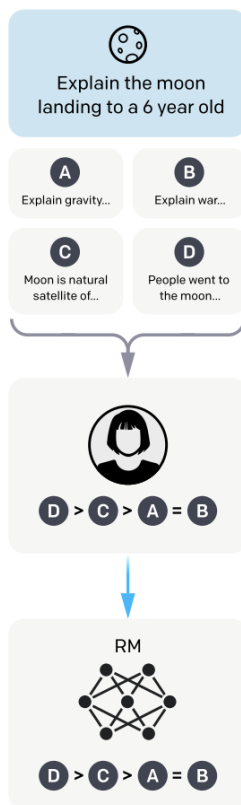
关键技术3：对齐

Step 2

Collect comparison data,
and train a reward model.

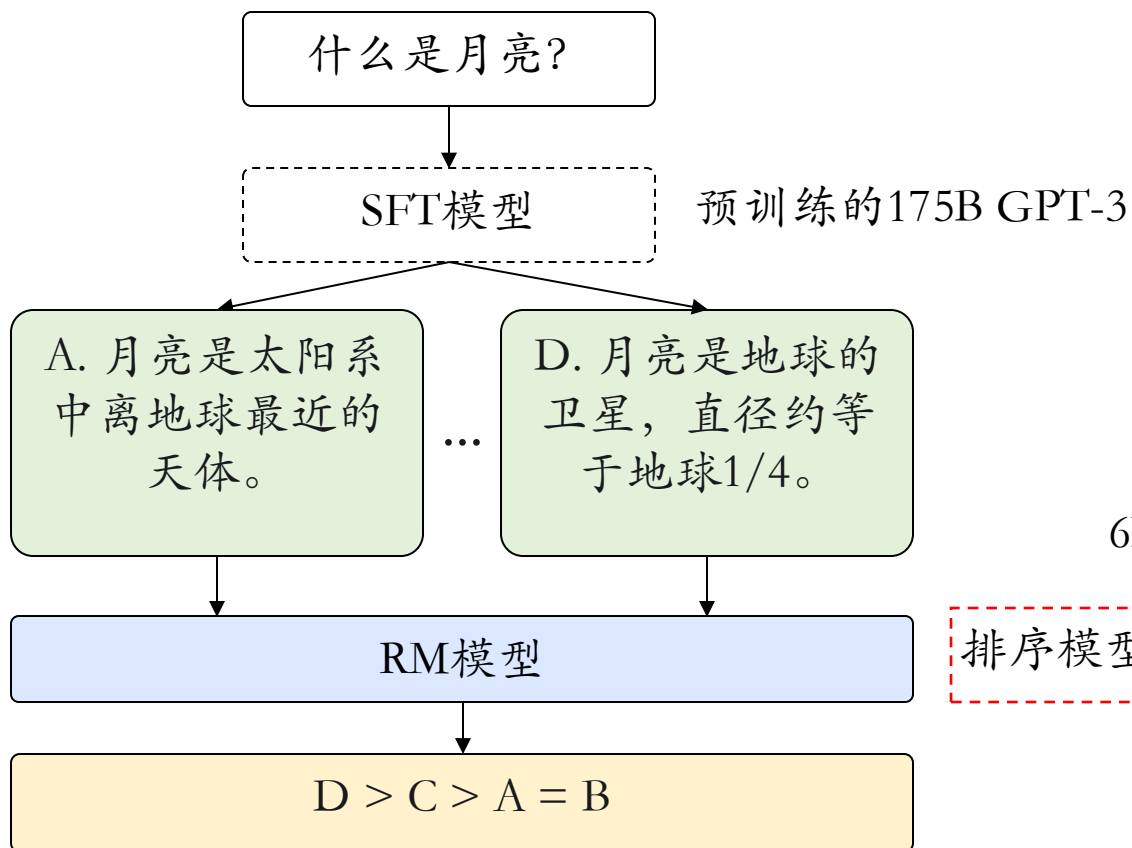
第二步：Reward modeling (RM)

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.

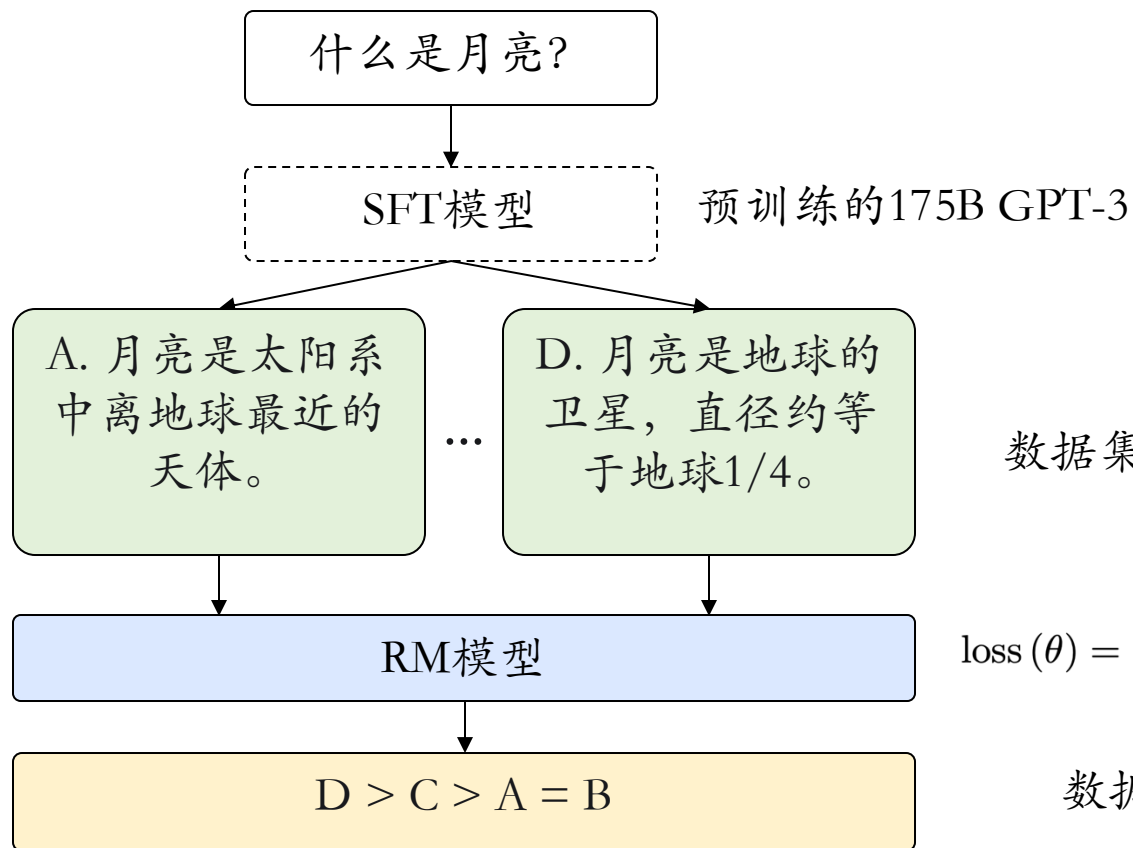


6B 模型（便宜+稳定）

排序模型（问题 + 答案 → 输出分数）

关键技术3：对齐

第二步：Reward modeling (RM)



数据集中问题 x 对应的两个答案，
且 y_w 排序比 y_l 高

$$\text{loss}(\theta) = -\frac{1}{\binom{K}{2}} E_{(x, y_w, y_l) \sim D} [\log(\sigma(r_\theta(x, y_w) - r_\theta(x, y_l)))]$$

数据集中的问题

RM模型打分差值

关键技术3：对齐

Step 3

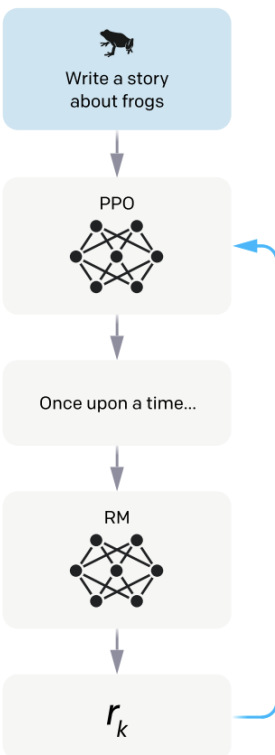
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

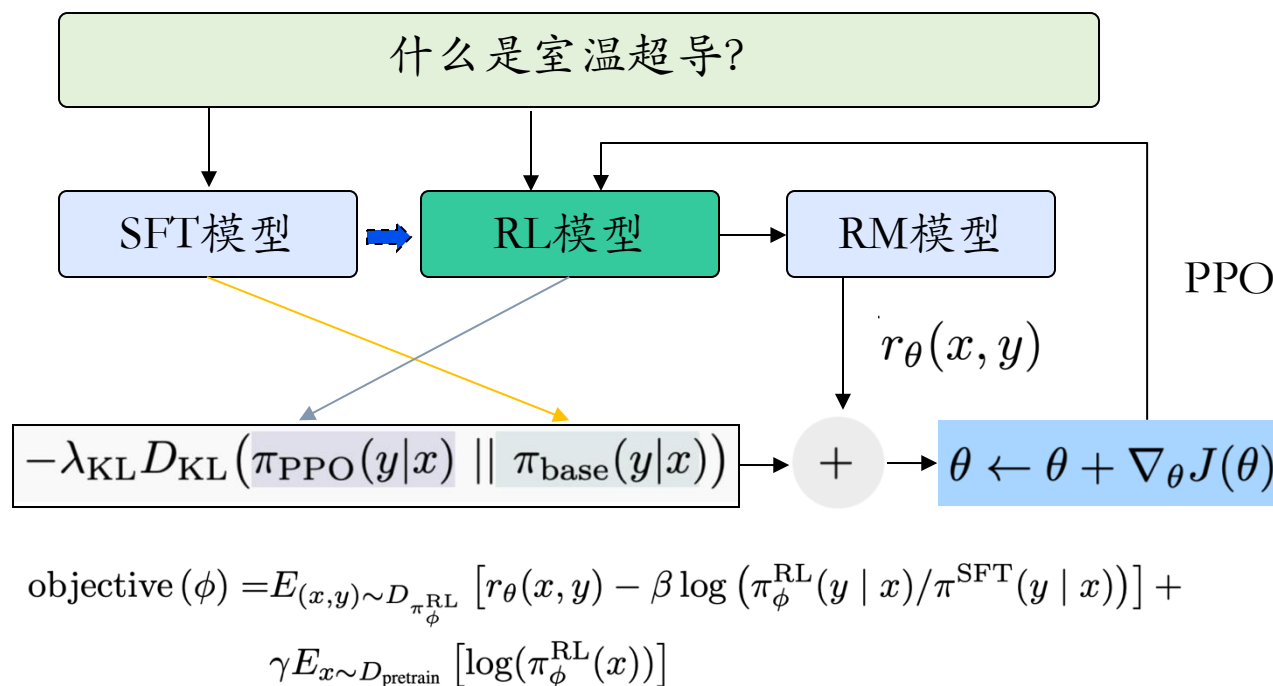
The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



第三步：Reinforcement learning (RL)



注：PPO，Proximal Policy Optimization，近端策略优化

关键技术3：对齐

Step 3

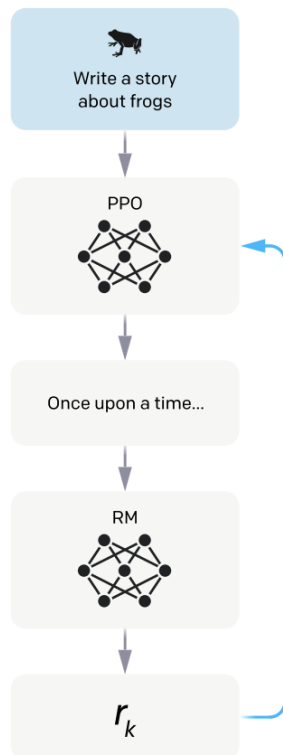
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

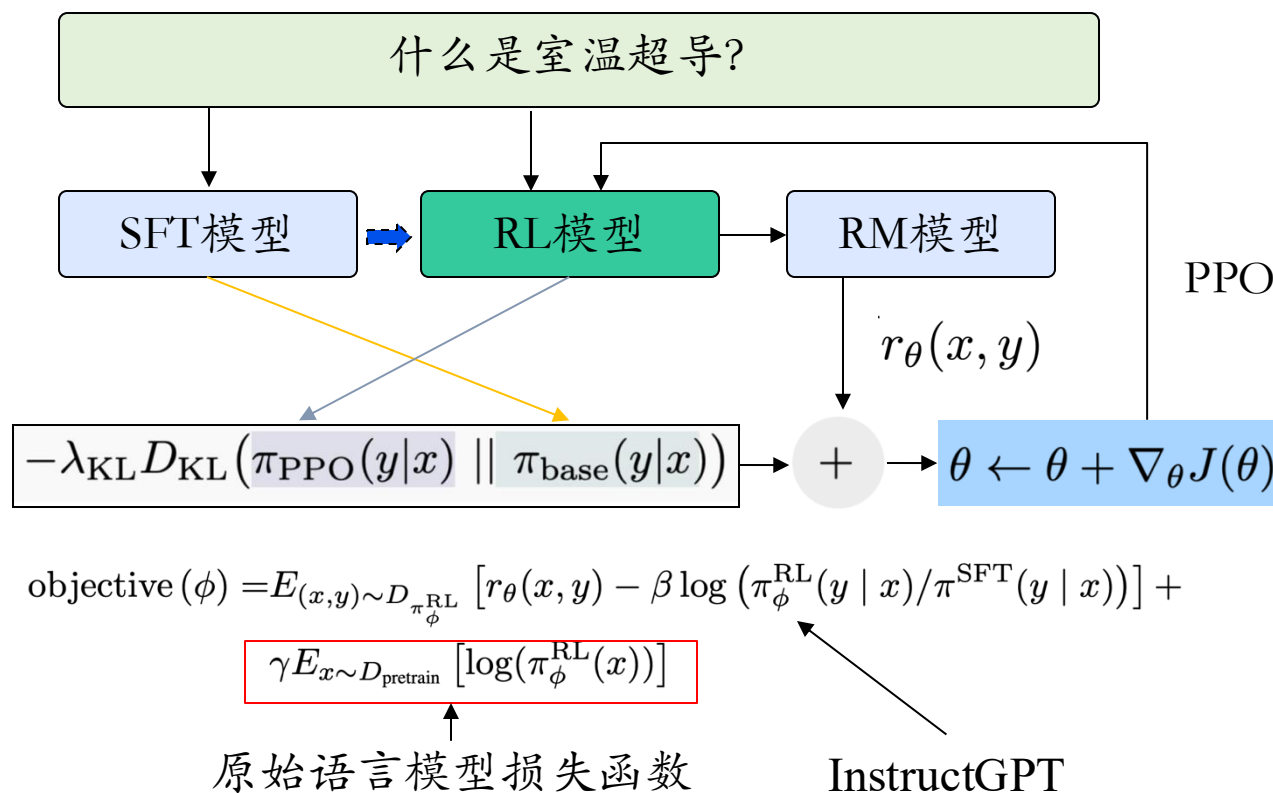
The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



第三步：Reinforcement learning (RL)

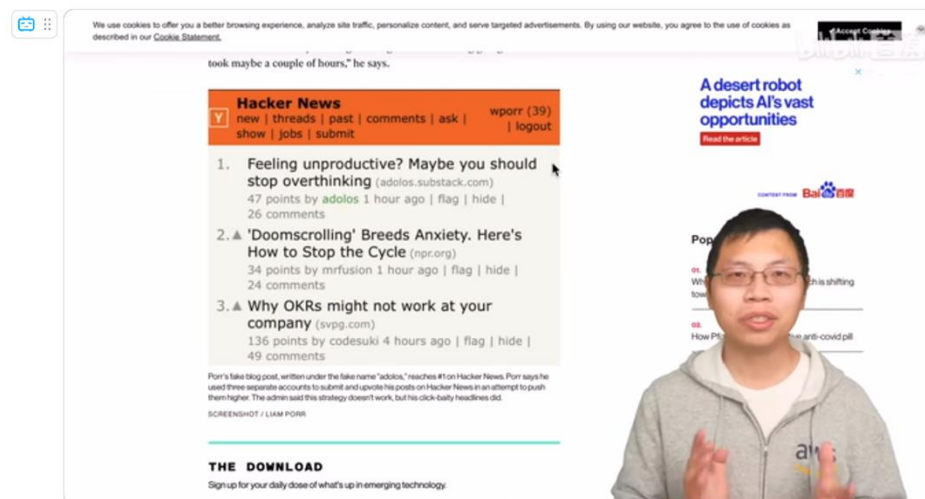


注：PPO，Proximal Policy Optimization，近端策略优化

关键技术3：对齐

6. 延伸阅读：

a. 【GPT, GPT-2, GPT-3 论文精读【论文精读】】

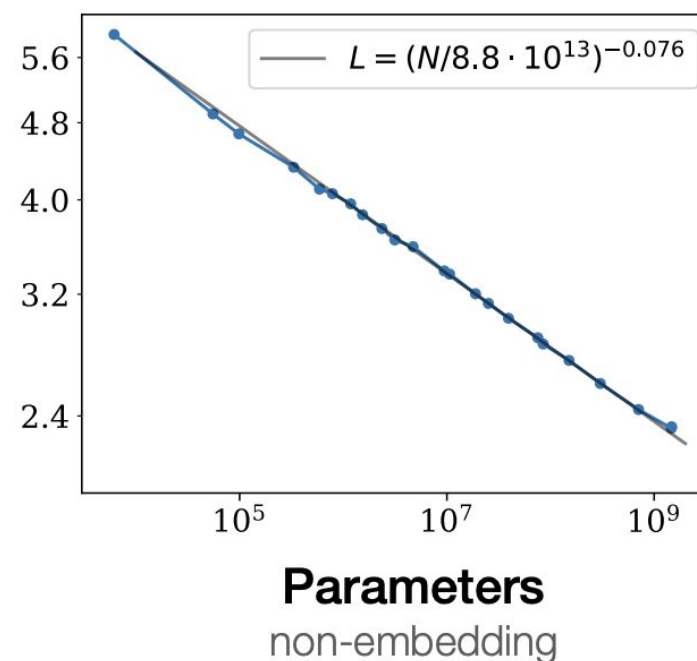
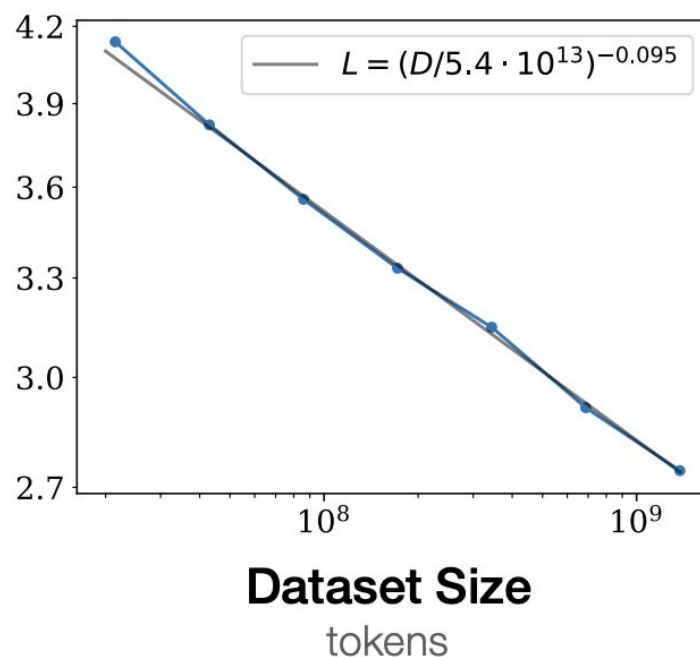
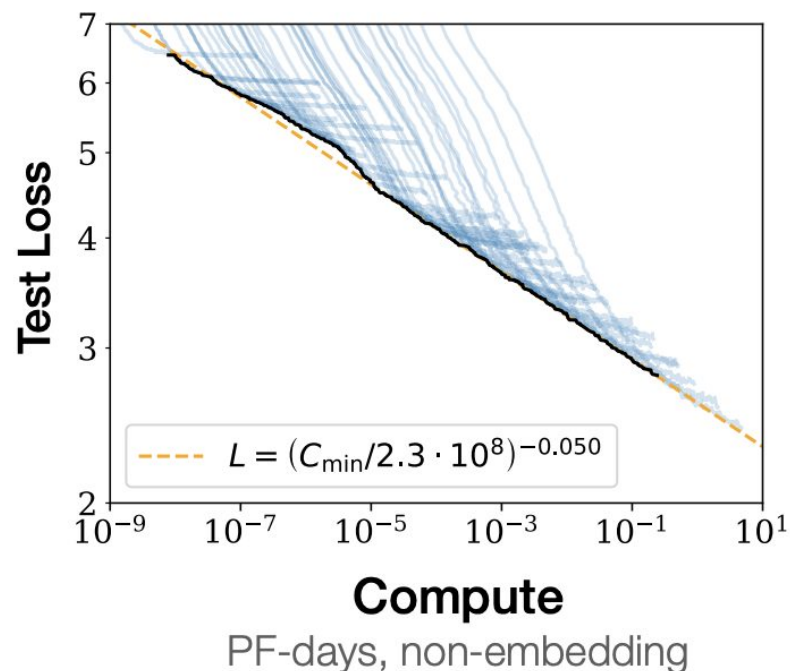


b. 【InstructGPT 论文精读【论文精读-48】】



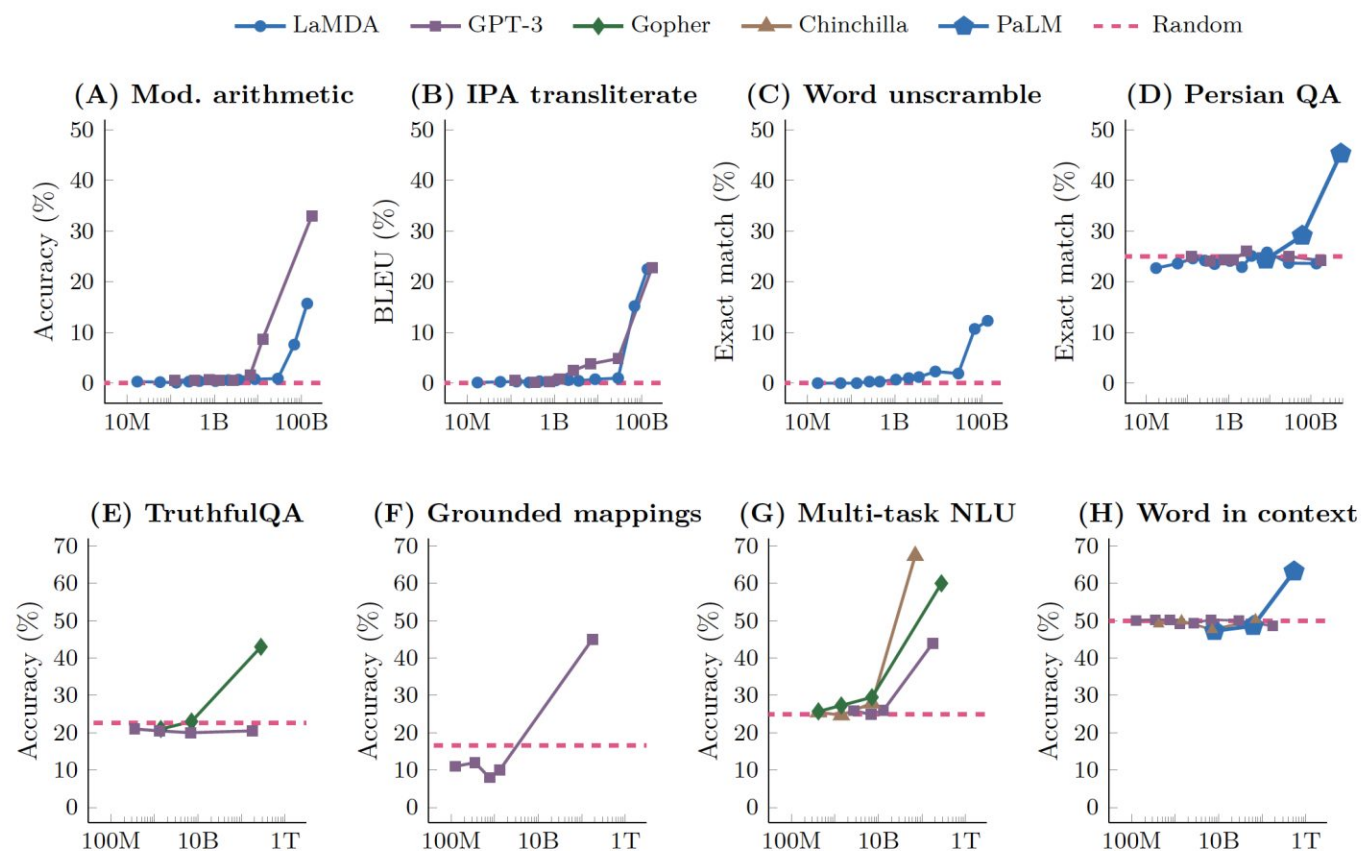
大规模语言模型的扩展定律 (Scaling Law)

- 模型性能与模型大小、数据规模和算力三者之间的关系
 - 随着模型参数规模和预训练数据规模的不断增加，模型能力与任务效果将会随之改善
 - Chinchilla推荐：10B模型用200B tokens (1:20)



大规模语言模型的涌现能力 (Emergent Abilities)

- 语言模型参数达到一定规模时，某些能力表现（语言推理）会突然大幅提升
- 在小模型中未观察到但在大模型中体现出来的能力



涌现能力的代表

上下文学习

思维链推理

大规模语言模型的幻觉 (Hallucination)

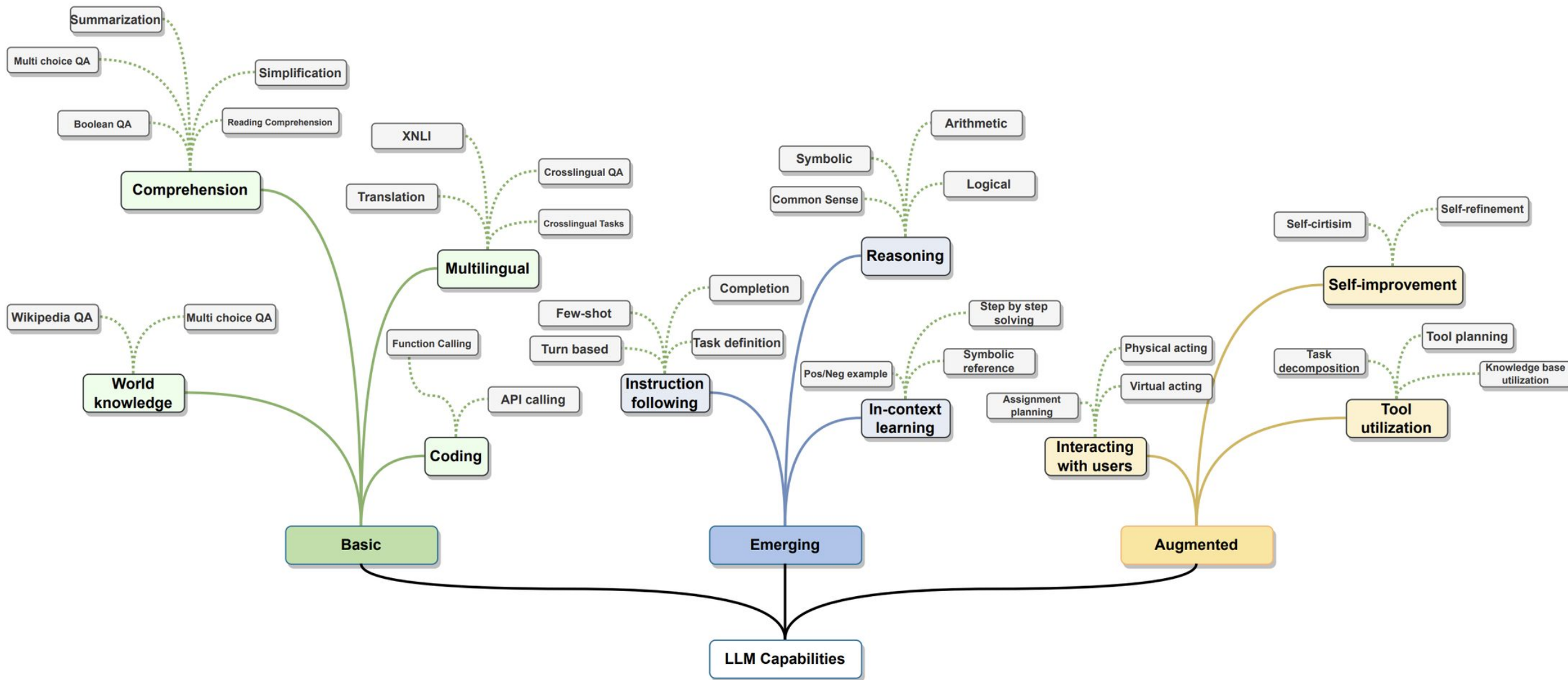
- ❑ **定义:** 模型生成的文本不遵循原文 (Faithfulness) 或不符合事实 (Factualness)
- ❑ **能力优势:** 大模型创造性和发散能力的体现
- ❑ **潜在影响:** 影响模型可信度, 容易被攻击、被滥用 (如恶意内容生成和散播谣言), 污染互联网



问: 鲁迅和周树人是同一个人吗?

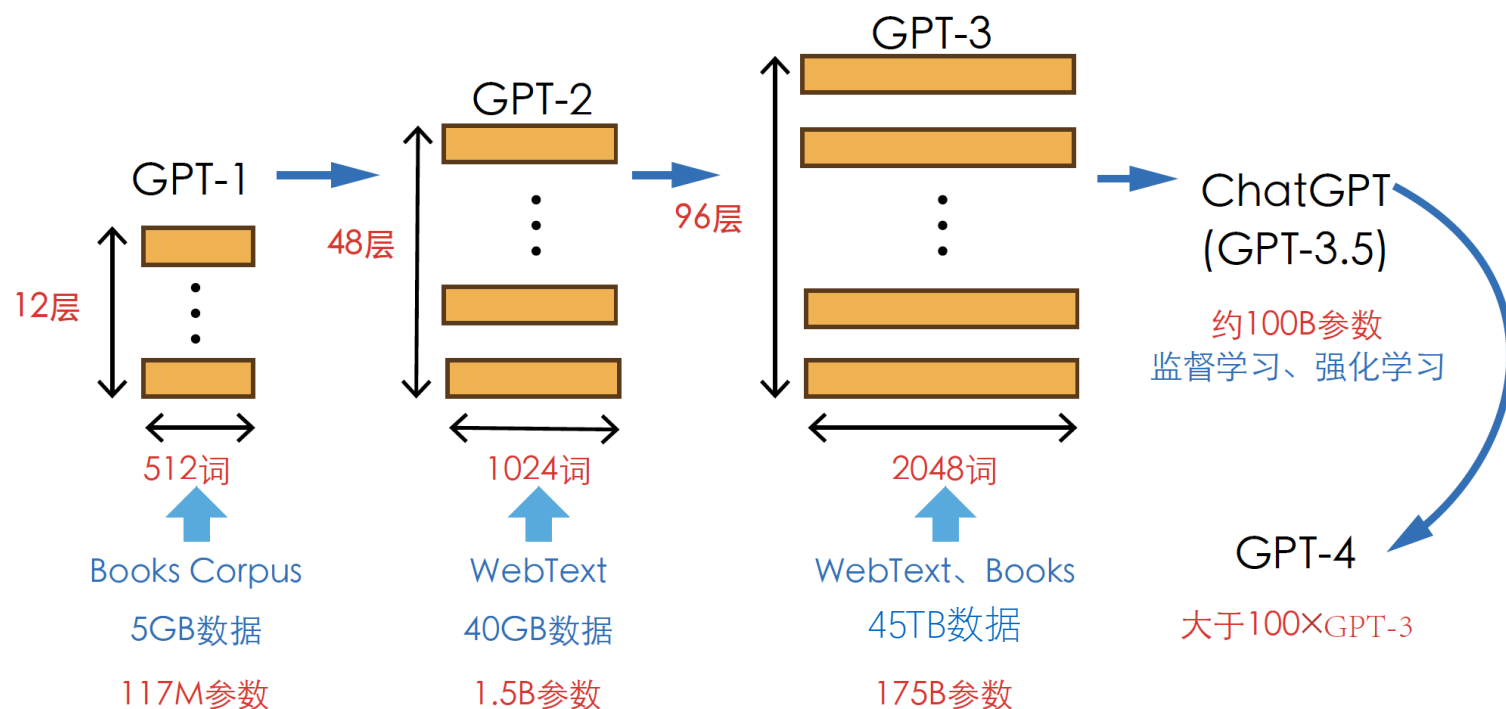
答: 不是。鲁迅和周树人是两个不同的人。鲁迅是中国今现代文学巨匠, 以其独特的文学风格和引人入胜的故事而闻名。周树人是中国当代著名作家, 以其犀利的文字和敏锐的观察而受到推崇。

大模型的能力版图



从GPT-3到ChatGPT

- ❑ **GPT-1**: 单向自回归建模 + 有监督微调, 探索 “预训练+微调” 范式下的自然语言任务求解能力
- ❑ **GPT-2**: 单向自回归建模 + 更多的数据、更大的模型, 探索基于自然语言提示的多任务解决能力
- ❑ **GPT-3**: 探索了千亿参数规模的语言模型效果, 提出了基于 “上下文学习” 的任务解决方法
- ❑ **GPT-3.5/ChatGPT**: 指令微调 + 人类反馈 + 对话优化



- ❑ **GPT-4**: 多模态能力, 更长的上下文理解与处理能力, 更高效的训练与预测, 更广泛的高价值应用场景

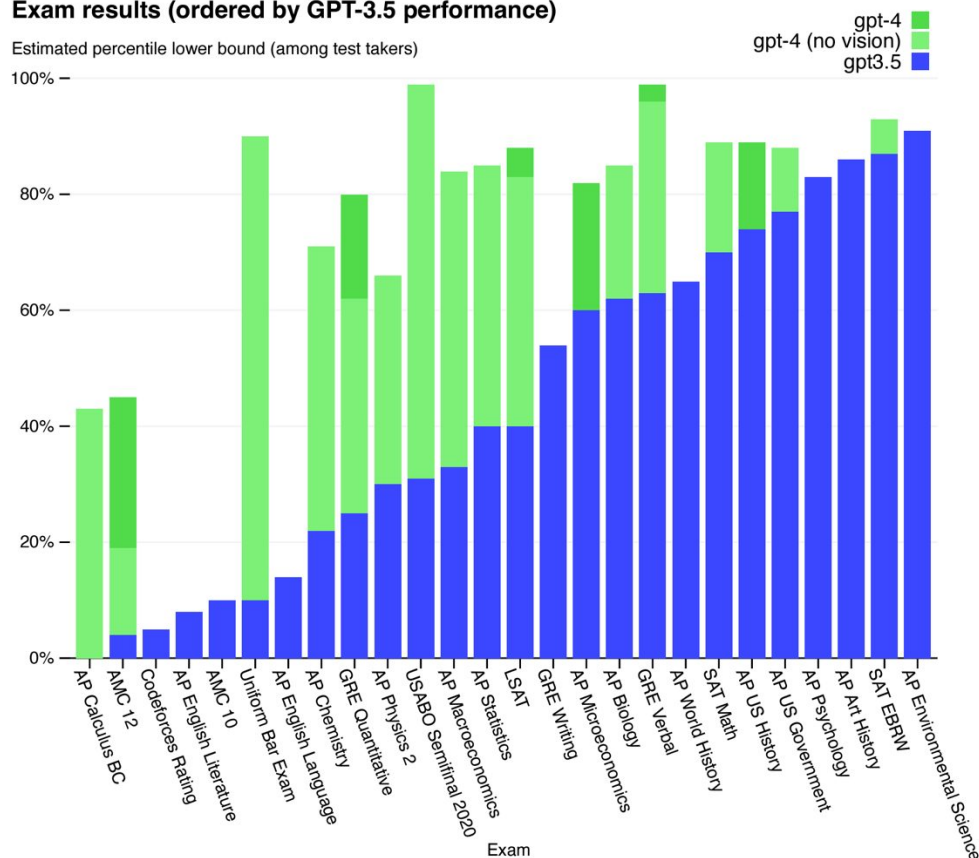
GPT 系列演进

GPT系列的版本演化

- ❑ GPT-4在各种专业和学术基准上表现出人类相当的水平，包括约前 10% 的成绩通过模拟律师资格考试
- ❑ 强大的多模态处理能力扩大了 GPT-4 的应用范围，同时生成式的幻觉、安全问题均有较大的改善

Exam results (ordered by GPT-3.5 performance)

Estimated percentile lower bound (among test takers)



OpenAI 发布 GPT-4 使用指南

- 写出清晰的指令
- 提供参考文本
- 将复杂的任务拆分为更简单的子任务
- 给模型时间「思考」
- 使用外部工具
- 系统地测试变更

[1] Achiam, Josh, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

[2] OpenAI. Prompt engineering. <https://platform.openai.com/docs/guides/prompt-engineering>.

Llama系列大模型

- ❑ 开源自回归模型，基于Transformer架构，参数规模从 7B到 405B 不等
- ❑ 商用许可：Llama 1不可免费商用， Llama 2和Llama 3可商用（需申请商用许可）

版本特点	Llama-1	Llama-2	Llama-3	Llama-3.1	Llama-3.2
模型规模	7B-65B	7B-70B	8B-70B	8B-405B	1B-90B
数据规模	1.4T	2T	15T	15T	9T
多模态	×	×	×	×	√
多语言	×	×	×	√	√
工具调用	×	×	×	√	√
训练范式	预训练	预训练+指令微调+RLHF	预训练+指令微调+DPO	预训练+指令微调+DPO	预训练+指令微调+DPO
上下文长度	2K	4k	8k	131K	131K
主要演变	--	增加后训练	扩展预训练	1. 增加模型规模 2. 增加多语言和工具能力 3. 扩展上下文长度	1. 增加端侧模型 2. 增加多模态能力

[1] Touvron, Hugo, et al. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

[2] Touvron, Hugo, et al. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

[3] Dubey, Abhimanyu, et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783.



Llama系列大模型

- ❑ **训练数据:** CommonCrawl, C4, Github, Wikipedia, books, ArXiv, and StackExchange等
- ❑ **开源地址:** <https://github.com/facebookresearch/llama>
- ❑ **Llama 3.1 405B 在广泛评测中超越 GPT-3.5 Turbo、大部分基准测试得分超过了 GPT-4 0125**

Category Benchmark	Llama 3.1 405B	Nemotron 4 340B Instruct	GPT-4 (0125)	GPT-4 Omni	Claude 3.5 Sonnet
General					
MMLU (0-shot, CoT)	88.6	78.7 (non-CoT)	85.4	88.7	88.3
MMLU PRO (5-shot, CoT)	73.3	62.7	64.8	74.0	77.0
IFEval	88.6	85.1	84.3	85.6	88.0
Code					
HumanEval (0-shot)	89.0	73.2	86.6	90.2	92.0
MBPP EvalPlus (base) (0-shot)	88.6	72.8	83.6	87.8	90.5
Math					
GSM8K (8-shot, CoT)	96.8	92.3 (0-shot)	94.2	96.1	96.4 (0-shot)
MATH (0-shot, CoT)	73.8	41.1	64.5	76.6	71.1
Reasoning					
ARC Challenge (0-shot)	96.9	94.6	96.4	96.7	96.7
GPQA (0-shot, CoT)	51.1	-	41.4	53.6	59.4
Tool use					
BFCL	88.5	86.5	88.3	80.5	90.2
Nexus	58.7	-	50.3	56.1	45.7
Long context					
ZeroSCROLLS/QuALITY	95.2	-	95.2	90.5	90.5
InfiniteBench/En.MC	83.4	-	72.1	82.5	-
NIH/Multi-needle	98.1	-	100.0	100.0	90.8
Multilingual					
Multilingual MGSM (0-shot)	91.6	-	85.9	90.5	91.6

Category Benchmark	Llama 3.1 8B	Gemma 2 9B IT	Mistral 7B Instruct	Llama 3.1 70B	Mixtral 8x22B Instruct	GPT 3.5 Turbo
General						
MMLU (0-shot, CoT)	73.0	72.3 (5-shot, non-CoT)	60.5	86.0	79.9	69.8
MMLU PRO (5-shot, CoT)	48.3	-	36.9	66.4	56.3	49.2
IFEval	80.4	73.6	57.6	87.5	72.7	69.9
Code						
HumanEval (0-shot)	72.6	54.3	40.2	80.5	75.6	68.0
MBPP EvalPlus (base) (0-shot)	72.8	71.7	49.5	86.0	78.6	82.0
Math						
GSM8K (8-shot, CoT)	84.5	76.7	53.2	95.1	88.2	81.6
MATH (0-shot, CoT)	51.9	44.3	13.0	68.0	54.1	43.1
Reasoning						
ARC Challenge (0-shot)	83.4	87.6	74.2	94.8	88.7	83.7
GPQA (0-shot, CoT)	32.8	-	28.8	46.7	33.3	30.8
Tool use						
BFCL	76.1	-	60.4	84.8	-	85.9
Nexus	38.5	30.0	24.7	56.7	48.5	37.2
Long context						
ZeroSCROLLS/QuALITY	81.0	-	-	90.5	-	-
InfiniteBench/En.MC	65.1	-	-	78.2	-	-
NIH/Multi-needle	98.8	-	-	97.5	-	-
Multilingual						
Multilingual MGSM (0-shot)	68.9	53.2	29.9	86.9	71.1	51.4

国产AI大模型分类及典型案例

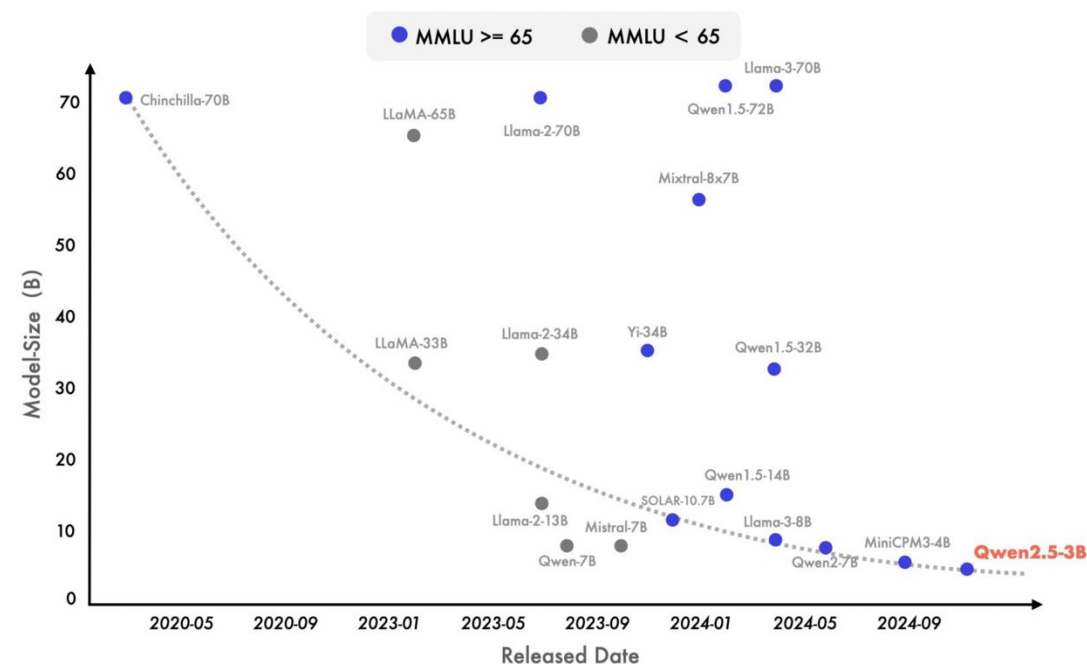
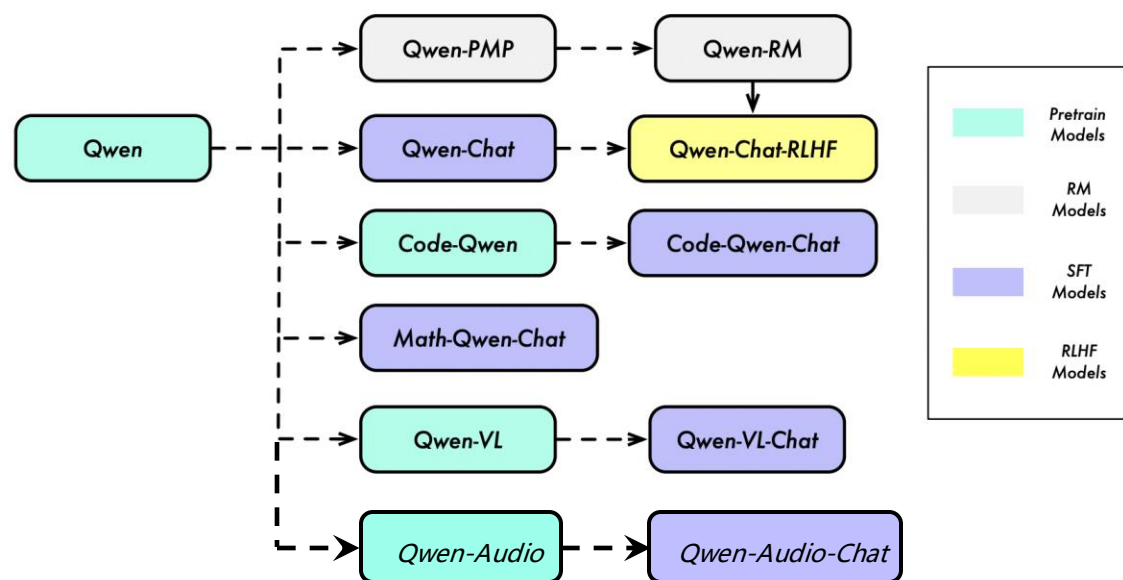
❑ **百模大战：**据不完全统计，截至2024年4月底，国内共推出305个大模型



[1] SuperGLUE团队. 中文大模型基准测评2024年度报告.

Qwen系列大模型

- **通义千问 (Qwen)** 是阿里云研发的预训练语言模型，执行理解、生成和解释人类语言、图片和文档等任务，在创意文案、办公助理、学习助手、趣味生活等方面为用户提供丰富的交互体验
- 具备中英文理解、数学推理、代码理解等能力



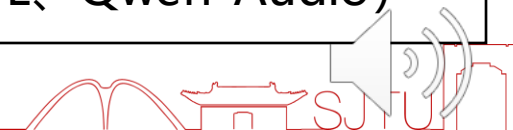
[1] Bai, Jinze, et al. Qwen technical report. arXiv preprint arXiv:2309.16609.
[2] Yang, An, et al. Qwen2 technical report. arXiv preprint arXiv:2407.10671.

Qwen系列的版本演化

- ❑ Qwen1.5: Model Scaling向更大和更小扩展、MoE、AWQ/GGUF
- ❑ Qwen2: 多语言增强、长文本能力增强
- ❑ Qwen2.5: 数学、代码能力增强、多语言增强
- ❑ Scaling
 - Qwen: 1.8B, 7B, 14B, and 72B models
 - Qwen1.5: 0.5B, 1.8B, 4B, 14BA2.7B, 7B, 14B, 32B, 72B, and 110B models
 - Qwen2: 0.5B, 1.5B, 7B, 57A14B, and 72B models
 - Qwen2.5: 0.5B, 1.5B, 3B, 7B, 14B, 32B, and 72B models

版本迭代特点

- 结构优化 (MoE提高训练效率)
- 预训练数据的数量扩充与质量提升
- 丰富Post-training阶段涵盖任务和领域
- 窗口长度的扩展 (训练数据和训练策略两方面)
- 多语言语向扩增与能力增强
- 专项能力增强 (训练专家模型+以专家模型带动基模)
- 模态扩展 (Qwen-VL、Qwen-Audio)



DeepSeek 系列大模型

1

2023.11

DeepSeek LLM

参数规模：70B、67B参数

架构：Dense（稠密）Transformer 模型

训练数据：2T token中英文语料

训练范式：遵循预训练+SFT+DPO标准大模型训练范式

性能：DeepSeek的首个大模型，对标Llama2系列。67B Base版本在推理、编码等方面超越了Llama2 70B Base。模型架构采用Dense结构

DeepSeek 系列大模型

1

2023.11

DeepSeek LLM

参数规模：70B、67B参数

架构：Dense（稠密）Transformer 模型

训练数据：2T token中英文语料

训练范式：遵循预训练+SFT+DPO标准大模型训练范式

性能：DeepSeek的首个大模型，对标Llama2系列。67B Base版本在推理、编码等方面超越了Llama2 70B Base。模型架构采用Dense结构

2

2024.5

DeepSeek-V2

参数规模：236B参数，激活参数21B

架构：引入DeepSeekMoE（混合专家）架构。引入MLA（Multi-head Latent Attention）

训练数据：8.1T token语料

训练范式：遵循预训练+SFT+强化学习标准范式

特点：性能比肩GPT-4 Turbo。MLA使得KV cache节约90%以上，推理速度提升5倍以上。MoE架构使得模型参数扩大时，推理成本几乎不变，训练成本大幅降低，提升模型能力

DeepSeek 系列大模型

2

2024.5

DeepSeek-V2

参数规模：236B参数，激活参数21B

架构：引入DeepSeekMoE（混合专家）架构。引入MLA（Multi-head Latent Attention）

训练数据：8.1T token语料

训练范式：遵循预训练+SFT+强化学习标准范式

特点：性能比肩GPT-4 Turbo。MLA使得KV cache节约90%以上，推理速度提升5倍以上。MoE架构使得模型参数扩大时，推理成本几乎不变，训练成本大幅降低，提升模型能力

3

2024.9

DeepSeek-V2.5

基于V2，主要优化了通用任务能力和代码能力

DeepSeek 系列大模型

3

2024.9

DeepSeek-V2.5

基于V2，主要优化了通用任务能力和代码能力

4

2024.12

DeepSeek-V3

参数规模：671B参数，激活参数37B

训练数据：14.8T token语料

架构：在V2基础上，引入Auxiliary-loss-free strategy来进行专家均衡。引入Multi-Token Prediction training提升模型性能和加速推理。引入FP8混合精度降低训练和推理开销

训练数据：14.8T token语料

训练范式：遵循预训练+SFT+DPO标准大模型训练范式

特点：性能超越主流的Llama3系列和Qwen2.5系列，且训练开销和推理开销大幅降低

DeepSeek 系列大模型

4

2024.12

DeepSeek-V3

参数规模：671B参数，激活参数37B

训练数据：14.8T token语料

架构：在V2基础上，引入Auxiliary-loss-free strategy来进行专家均衡。引入Multi-Token Prediction training提升模型性能和加速推理。引入FP8混合精度降低训练和推理开销

训练数据：14.8T token语料

训练范式：遵循预训练+SFT+DPO标准大模型训练范式

特点：性能超越主流的Llama3系列和Qwen2.5系列，且训练开销和推理开销大幅降低

5

2025.01

DeepSeek-R1

基座模型：DeepSeek-V3-Base

训练数据：14.8T token语料

训练范式：在base模型上直接进行多阶段训练、冷启动训练和强化学习

特点：首次达到了类GPT4-O1的推理能力

部署大模型

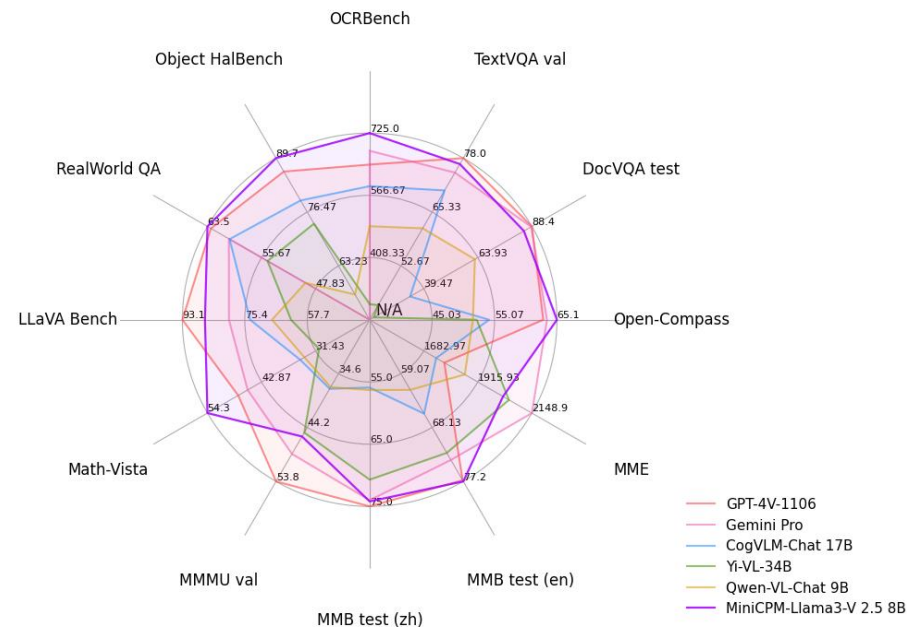
大模型类型：

- 大语言模型：Qwen
- 多模态模型：MiniCPM

部署方式：

- API调用
- 本地部署

	Qwen2-72B	Llama3-70B	Mixtral-8x22B	Qwen1.5-110B
MMLU	84.2	79.5	77.8	80.4
MMLU-Pro	55.6	52.8	49.5	49.4
GPQA	37.9	36.3	34.3	35.9
TheoremQA	43.1	32.3	35.9	34.9
BBH	82.4	81.0	78.9	74.8
HumanEval	64.6	48.2	46.3	54.3
MBPP	76.9	70.4	71.7	70.9
MultiPL-E	59.6	46.3	46.7	52.7
GSM8K	89.5	83.0	83.7	85.4
MATH	51.1	42.5	41.7	49.6
C-Eval	91.0	65.2	54.6	89.1
CMMLU	90.1	67.2	53.4	88.3
Multi-Exam	76.6	70.0	63.5	75.6
Multi-Understanding	80.7	79.9	77.7	78.2
Multi-Mathematics	76.0	67.1	62.9	64.4



部署大模型（功能样例）



Language Understanding

父亲：儿子，你高中毕业了，现在打算上哪个大学？
儿子：爸爸，我已经准备好了申请资料，想申请一个好的本科专业。
父亲：那你已经想好申请哪些大学了吗？
儿子：是的。我计划申请几所排名靠前的大学。我已经做了很多研究，准备在学业、社交和职业规划等方面全面提升自己。
父亲：很好！你一直都很努力，爸爸妈妈一直都很自豪。上大学是一个新的起点，你需要面对更多的挑战 and 机会。你有什么打算？
儿子：我计划尽早适应大学生活，参加各种社团和课外活动，拓宽自己的视野和人际圈。同时，我会认真学习专业知识和技能，努力取得好成绩，争取获得奖学金和其他荣誉。
父亲：非常好！你的计划很明智，爸爸相信你一定会有一个充实而有意义的大学时光。爸爸会一直支持你的决定，希望你取得好成绩，为你的未来打下坚实的基础。
儿子：谢谢爸爸！我会加倍努力，不辜负您对我的期望。
这段对话中，儿子打算上大学后有哪些计划和打算？

Father: Son, you have graduated from high school. Which university are you planning to attend now?
Son: Dad, I have already prepared my application materials, and I want to apply for an excellent undergraduate program.

Father: Have you already decided which universities you want to apply to?
Son: Yes, I plan to apply to several top-ranked universities. I have done a lot of research, and I am ready to improve myself in academics, social life, and career planning.
Father: That's great! You have always been hardworking and your mom and I are proud of you. Going to college is a new starting point, and you will face more challenges and opportunities. What are your plans?
Son: I plan to adapt to college life as soon as possible and participate in various clubs and extracurricular activities to broaden my horizons and social circle. At the same time, I will study my major knowledge and skills seriously, strive for good grades, and try to win scholarships and other honors.
Father: Very good! Your plans are wise. Dad believes that you will have a fulfilling and meaningful college life. Dad will always support your decisions and hopes you achieve good results, laying a solid foundation for your future.
Son: Thank you, Dad! I will work even harder and not let your expectations down.
What are the son's plans and intentions after going to college in this conversation?

Responses

Model Qwen-7B-Chat (SFT) **Elo 973**
Response
儿子打算尽早适应大学生活，参加各种社团和课外活动，拓宽自己的视野和人际圈。同时，他会认真学习专业知识和技能，努力取得好成绩，争取获得奖学金和其他荣誉。
The son plans to adapt to university life as soon as possible, participate in various clubs and extracurricular activities, and expand his horizons and social circle. At the same time, he will study professional knowledge and skills seriously, work hard to achieve good grades, and strive to obtain scholarships and other honors.

Qwen-14B-Chat (SFT) **Elo 1049**
儿子打算在学业、社交和职业规划等方面全面提升自己，尽早适应大学生活，参加各种社团和课外活动，拓宽自己的视野和人际圈。同时，他会认真学习专业知识和技能，努力取得好成绩，争取获得奖学金和其他荣誉。
The son plans to comprehensively improve himself in his academic, social, and career planning aspects to adapt to university life as early as possible. He will participate in various clubs and extracurricular activities to broaden his horizons and expand his social circle. At the same time, he will diligently study professional knowledge and skills, strive for good grades, and obtain scholarships and other honors.

Code

python实现一个正则表达式匹配，要求能匹配到所有以139开头，以00结尾的11位数字
Implement a regular expression matching in Python, which should be able to match all 11-digit numbers that start with "139" and end with "00".

Responses

Model Qwen-7B-Chat (SFT) **Elo 941**
Response
你可以使用Python的re模块来实现这个正则表达式匹配。以下是一个示例代码：

```
import re

def match_phone_numbers(phone_numbers):
    pattern = r'^139d9$'
    return re.findall(pattern, phone_numbers)
```

```
phone_numbers = ['13900000000', '13912345678', '13998765432', '13987654321']
print(match_phone_numbers(phone_numbers))
```

在这个代码中，我们首先定义了一个正则表达式模式`^139d9\$`，这个模式表示以139开头，以00结尾，且长度为11的数字。然后我们使用`re.findall`函数来查找所有匹配这个模式的字符串，最后返回这些匹配的字符串。

注意，这个代码假设`phone\_numbers`是一个包含多个字符串的列表。如果你的`phone\_numbers`是一个字符串，你需要先将其转换为列表。
You can use the re module in Python to implement this regular expression matching. Here is an example code:

```
import re

def match_phone_numbers(phone_numbers):
    pattern = r'^139d9$'
    return re.findall(pattern, phone_numbers)
```

```
phone_numbers = ['13900000000', '13912345678', '13998765432', '13987654321']
print(match_phone_numbers(phone_numbers))
```

This code defines a regular expression pattern `^139d9\$`. This pattern represents a string that starts with '139', ends with '00', and has a length of 11 digits. Then, we use the `re.findall` function to find all the strings that match this pattern, and finally, we return these matching strings.

Note that this code assumes that `phone\_numbers` is a list containing multiple strings. If your `phone\_numbers` is a string, you must first convert it to a list.

User

Question: [Uploaded scatter_data.csv] Create a scatter plot with different size and color settings

Qwen Chat

Great, you've uploaded a file named "scatter_data.csv". This appears to be a CSV file, which is commonly used for storing tabular data. We can use this data to create a scatter plot with different size and color settings.

Thought: Firstly, I need to use the code interpreter API to load the data and taking a look at the first few rows to understand what we're working with.

Action: code_interpreter

Action Input:

```
Py
import pandas as pd
# Load the data
df = pd.read_csv("scatter_data.csv")
# Display the first few rows of the dataframe
df.head()
```

Observation: execute_result:

	x	y	sizes	colors
0	7.577257	1.952430	46.433060	24.870965
1	4.879020	2.570014	38.545454	21.500782
2	4.192995	5.250490	60.999630	23.404101
3	0.273015	3.678973	63.538505	50.963053
4	3.445224	2.462327	59.921040	27.207964

Thought: Now I understand the data format. I need to use the code_interpreter API to run the code:

Action: code_interpreter

Action Input:

```
Py
import matplotlib.pyplot as plt
import pandas as pd

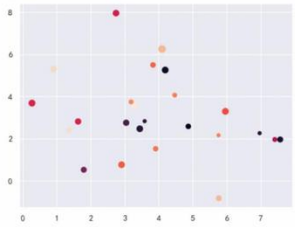
df = pd.read_csv('scatter_data.csv')

fig, ax = plt.subplots()

ax.scatter(df['x'], df['y'], s=df['sizes'], c=df['colors'])

plt.show()
```

Observation:



Thought: The code interpreter returned the result.

Final Answer: The scatter plot has been successfully created using the data from the "scatter_data.csv" file. The size of each point is determined by the "sizes" column, and the color is determined by the "colors" column. This allows for a high degree of customization in the appearance of the scatter plot.

User

Question: [Uploaded scatter_data.csv] Create a scatter plot with different size and color settings

Qwen Chat

Thought: Firstly, I need to use the code interpreter API to load the data and taking a look at the first few rows to understand what we're working with.

Action: code_interpreter

Action Input:

```
Py
import pandas as pd
# Load the data
df = pd.read_csv("scatter_data.csv")
# Display the first few rows of the dataframe
df.head()
```

Observation: The output is:

	x	y	sizes	colors
0	7.577257	1.952430	46.433060	24.870965
1	4.879020	2.570014	38.545454	21.500782
2	4.192995	5.250490	60.999630	23.404101
3	0.273015	3.678973	63.538505	50.963053
4	3.445224	2.462327	59.921040	27.207964

Thought:

Code Lens

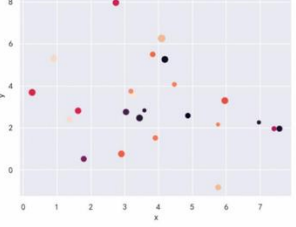
Now that we have the data loaded, we can start thinking about how to create the scatter plot. We'll need to decide on the x-axis and y-axis labels, as well as the size and color settings for the points.

Action: code_interpreter

Action Input:

```
Py
import matplotlib.pyplot as plt
# Set the x-axis label
plt.xlabel("x")
# Set the y-axis label
plt.ylabel("y")
# Set the size of the points
plt.scatter(df["x"], df["y"], s=df["sizes"], c=df["colors"])
# Show the plot
plt.show()
```

Observation:



Thought: The plot is now complete. We can see that the points are scattered according to the x and y values, and the size and color of the points are determined by the values in the "sizes" and "colors" columns of the dataframe.

Final Answer: The scatter plot is now complete.



部署大模型（API调用）

□ 获得大模型调用权限

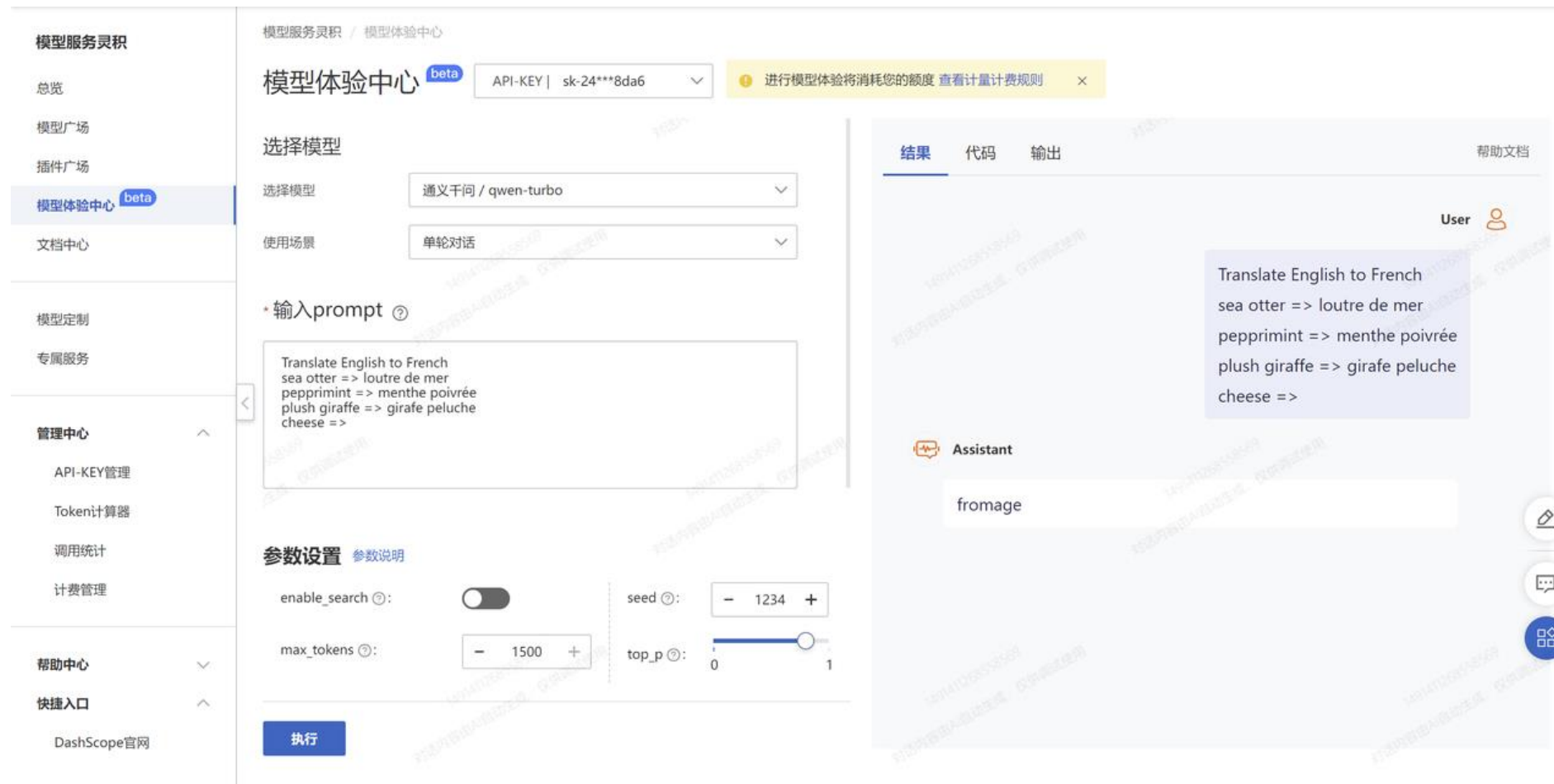
- 通义千问: <https://help.aliyun.com/zh/dashscope/developer-reference/quick-start>
- DeepSeek: <https://www.deepseek.com>
- Llama: <https://github.com/Meta-Llama/llama>
- 智谱AI: <https://open.bigmodel.cn/>
- Openai: <https://platform.openai.com/playground>
- 其他: 文心一言、百川大模型

*基本流程: 开通服务获得API-KEY (获赠计算额度), 使用API Key调用服务 (其中OpenAI需要科学手段)

部署大模型 (API调用)

通过GUI界面调用 (适合案例测试)

- <https://help.aliyun.com/zh/dashscope/developer-reference/quick-start>
- <https://huggingface.co/spaces/Qwen/Qwen2-72B-Instruct>



The screenshot displays the DashScope Model Studio interface. On the left is a sidebar with navigation options: 模型服务灵积, 总览, 模型广场, 插件广场, 模型体验中心 (selected), 文档中心, 模型定制, 专属服务, 管理中心, API-KEY管理, Token计算器, 调用统计, 计费管理, 帮助中心, 快捷入口, and DashScope官网. The main area is titled '模型体验中心' and includes an API-KEY field (sk-24***8da6), a '选择模型' dropdown (set to 通义千问 / qwen-turbo), and a '使用场景' dropdown (set to 单轮对话). The '输入prompt' section contains the text: 'Translate English to French', 'sea otter => loutre de mer', 'peppermint => menthe poivrée', 'plush giraffe => girafe peluche', and 'cheese =>'. Below this is the '参数设置' section with sliders for 'enable_search' (disabled), 'max_tokens' (1500), 'seed' (1234), and 'top_p' (0.5). A '执行' button is at the bottom left. On the right, a chat interface shows a 'User' message with the same prompt text and an 'Assistant' response with the word 'fromage'. The chat interface also has a '帮助文档' link and a '输出' button.

部署大模型 (API调用)

□ 通过代码调用:

普通调用:

```
# For prerequisites running the following sample, visit https://help.aliyun.com/document_
import dashscope
def sample_sync_call():
    prompt_text = '用萝卜、土豆、茄子做饭, 给我个菜谱.'
    resp = dashscope.Generation.call(
        model='qwen-turbo',
        prompt=prompt_text
    )
# The response status_code is HTTPStatus.OK indicate success, # otherwise indicate request
print(resp.output)# The output textprint(resp.usage)# The usage informationelse:
print(resp.code)# The error code.print(resp.message)# The error message.
sample_sync_call()
```

流式调用:

```
# For prerequisites running the following sample, visit https://help.aliyun.com/document_

import dashscope
def sample_sync_call_streaming():
    prompt_text = '用萝卜、土豆、茄子做饭, 给我个菜谱.'
    response_generator = dashscope.Generation.call(
        model='qwen-turbo',
        prompt=prompt_text,
        stream=True,
        top_p=0.8)
    head_idx = 0
    for resp in response_generator:
        paragraph = resp.output['text']
        print("\r%s" % paragraph[head_idx:len(paragraph)], end='')
        if(paragraph.rfind('\n') != -1):
            head_idx = paragraph.rfind('\n') + 1
    sample_sync_call_streaming()
```

□ 环境准备 (可选) :

- VS Code开发工具:
<https://code.visualstudio.com/>
- 利用Miniconda进行Python环境管理:
<https://docs.anaconda.com/free/miniconda/miniconda-install>

部署大模型（本地部署）

- ❑ 大模型本地部署开源平台：Ollama, LM Studio
- ❑ Web与API调用：OpenAI, Dashscope API
- ❑ 大模型本地私有化部署：Flask, MindIE

Ollama <https://ollama.com/>

- 用户体验：以其简单和易于安装而闻名，特别适合初学者和非技术用户。
- 可定制性：提供创建定制语言模型和运行各种预训练模型的灵活性。
- 开源：完全开源，有助于提高透明度和社区参与。
- 支持的操作系统：适用于 macOS、Linux 和 Windows（预览版），且处理器支持 AVX。

LM Studio <https://lmstudio.ai/>

- 功能集：提供更广泛的功能集，包括发现、下载和运行本地 LLM，以及应用内聊天界面以及与 OpenAI 兼容的本地服务器的兼容性。
- UI 友好性：与 Ollama 相比，被认为对用户界面更加友好。
- 模型选择：模型目录：提供来自 Hugging Face 等来源的更广泛的模型选择。
- 支持的操作系统：适用于 M1/M2/M3 Mac 或具有支持 AVX2 处理器的 Windows PC。Linux 已推出测试版。

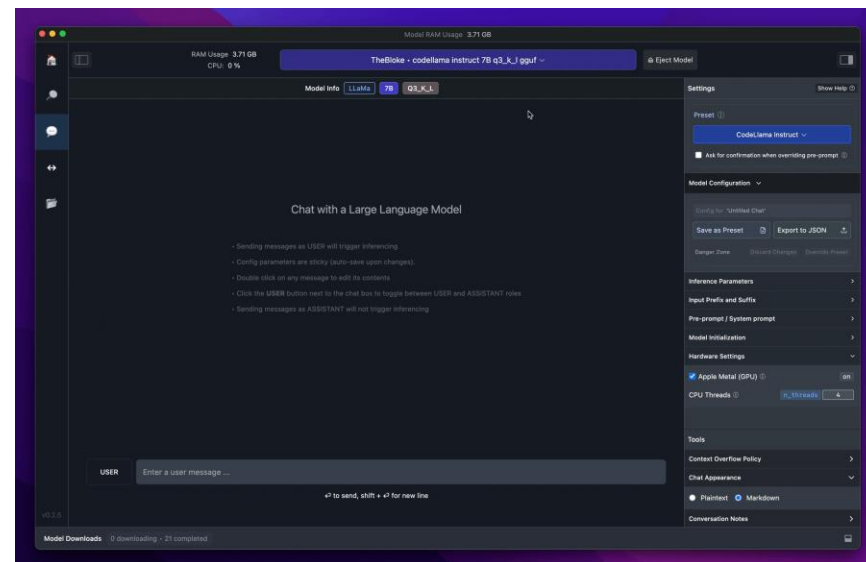
```
(base) zhangzhuosheng@zhangzhuoshengdeMac ~ % ollama run qwen2:7b
pulling manifest
pulling manifest
pulling manifest
pulling 43f7a214e532... 100% ██████████ 4.4 GB
pulling 62fbfd9ed093... 100% ██████████ 182 B
pulling c156170b718e... 100% ██████████ 11 KB
pulling f02dd72bb242... 100% ██████████ 59 B
pulling 648f809ced2b... 100% ██████████ 485 B
verifying sha256 digest
writing manifest
removing any unused layers
success
>>> 你是谁
我是阿里云开发的一款超大规模语言模型，我叫通义千问。作为一个AI助手，我的目标是帮助用户获得准确、有用的信息，
决他们的问题和困惑。无论是关于知识查询、信息解答、问题解释，还是进行创作辅助、与用户进行对话交流等，我都将尽
提供高质量的服务。如果您有任何问题或需要帮助，请随时告诉我！

>>> 鸡兔同笼，共35只头，94只脚，问鸡兔各多少？
...
这个问题是典型的的中国数学古题“鸡兔同笼”问题的变体。

设鸡的数量为x，兔子的数量为y。
根据题目描述，我们有以下两个方程：
1. 鸡和兔子的头数之和为35:  $(x + y = 35)$ 
2. 鸡和兔子的脚数之和为94:  $(2x + 4y = 94)$ （因为鸡有两只脚，兔子有四只脚）

我们可以从第一个方程解出  $(x = 35 - y)$ 。

然后将  $(x)$  的值代入第二个方程中：
 $(2(35 - y) + 4y = 94)$ 
```



部署大模型（本地部署）



配置要求

模型	推理显存(GB)	微调显存(GB)
Qwen2-0.5B-Instruct T ≡	最低2.7G	最低6G
Qwen2-1.5B-Instruct	最低5.2G	最低8.5G
Qwen2-7B-Instruct	最低16.7G	最低20G
Qwen2-72B-Instruct	最低143G	最低156G
Qwen2-MoE-57B-A14B-Instruct	最低116G	最低122G

显存要求

7b	4.4GB
latest	4.4GB
72b	41GB
7b latest	4.4GB
1.5b	935MB
0.5b	352MB
72b-instruct	41GB
72b-instruct-fp16	145GB

硬盘要求



部署大模型（本地部署）

- ❑ 1. 安装Ollama
 - 访问官方网站：打开浏览器，访问Ollama官方网站。
 - 选择版本：根据您的操作系统选择相应的安装包。Ollama支持MacOS、Linux和Windows
 - 下载安装包：点击Download按钮，下载适用于您设备的Ollama安装包
 - 安装Ollama：根据下载的安装包格式，运行安装程序或解压缩文件到您选择的目录
- ❑ 2. 运行特定模型（ollama run qwen2:7b）

```
(base) zhangzhuosheng@zhangzhuoshengdeMac ~ % ollama run qwen2:7b
```

```
[pulling manifest
```

```
pulling manifest
```

```
pulling manifest
```

```
pulling 43f7a214e532... 21% ██████████
```

```
| 929 MB/4.4 GB 5.6 MB/s 10m19s ██████████
```

部署大模型（本地部署）

- ❑ 1. 安装Ollama
- ❑ 2. 运行特定模型 (ollama run qwen2:7b)

```
(base) zhangzhuosheng@zhangzhuoshengdeMac ~ % ollama run qwen2:7b
```

```
pulling manifest
pulling manifest
pulling manifest
pulling 43f7a214e532... 100% ██████████ 4.4 GB
pulling 62fbfd9ed093... 100% ██████████ 182 B
pulling c156170b718e... 100% ██████████ 11 KB
pulling f02dd72bb242... 100% ██████████ 59 B
pulling 648f809ced2b... 100% ██████████ 485 B
verifying sha256 digest
writing manifest
removing any unused layers
success
```

```
>>>
```

```
>>> 你是谁
```

我是阿里云开发的一款超大规模语言模型，我叫通义千问。作为一个AI助手，我的目标是帮助用户获得准确、有用的信息，解决他们的问题和困惑。无论是关于知识查询、信息解答、问题解释，还是进行创作辅助、与用户进行对话交流等，我都将尽力提供高质量的服务。如果您有任何问题或需要帮助，请随时告诉我！

```
>>> 鸡兔同笼，共35只头，94只脚，问鸡兔各多少？
```

```
...
```

这个问题是典型的中国数学古题“鸡兔同笼”问题的变体。

设鸡的数量为 x ，兔子的数量为 y 。
根据题目描述，我们有以下两个方程：

- 鸡和兔子的头数之和为35： $(x + y = 35)$
- 鸡和兔子的脚数之和为94： $(2x + 4y = 94)$ （因为鸡有两只脚，兔子有四只脚）

我们可以从第一个方程解出 $(x = 35 - y)$ 。

然后将 (x) 的值代入第二个方程中：

```
\[2(35 - y) + 4y = 94\]
```

展开得：

```
\[70 - 2y + 4y = 94\]
```

合并同类项：

```
\[2y = 24\]
```

解得 $(y = 12)$ ，即兔子有12只。

将 (y) 的值代入任一原方程求解 (x) ：

```
\(x + 12 = 35\)
```

解得 $(x = 23)$ ，即鸡有23只。

所以，鸡和兔的数量分别是23只和12只。



部署大模型（本地部署）

- ❑ 1. 安装Ollama
- ❑ 2. 运行特定模型 (ollama run qwen2:7b)

```
(base) zhangzhuosheng@zhangzhuoshengdeMac ~ % ollama run qwen2:7b
pulling manifest
pulling manifest
pulling manifest
pulling 43f7a214e532... 100% 4.4 GB
pulling 62fbfd9ed093... 100% 182 B
pulling c156170b718e... 100% 11 KB
pulling f02dd72bb242... 100% 59 B
pulling 648f809ced2b... 100% 485 B
verifying sha256 digest
writing manifest
removing any unused layers
success
>>>
>>> 你是谁
我是阿里云开发的一款超大规模语言模型，我叫通义千问。作为一个AI助手，我的目标是帮助用户获得准确、有用的信息，解决他们的问题和困惑。无论是关于知识查询、信息解答、问题解释，还是进行创作辅助、与用户进行对话交流等，我都将尽力提供高质量的服务。如果您有任何问题或需要帮助，请随时告诉我！

>>> 鸡兔同笼，共35只头，94只脚，问鸡兔各多少？
...
这个问题是典型的中国数学古题“鸡兔同笼”问题的变体。

设鸡的数量为x，兔子的数量为y。
根据题目描述，我们有以下两个方程：

1. 鸡和兔子的头数之和为35:  $(x + y = 35)$ 
2. 鸡和兔子的脚数之和为94:  $(2x + 4y = 94)$ （因为鸡有两只脚，兔子有四只脚）

我们可以从第一个方程解出  $(x = 35 - y)$ 。

然后将  $(x)$  的值代入第二个方程中：

 $(2(35 - y) + 4y = 94)$ 

展开得：

 $(70 - 2y + 4y = 94)$ 

合并同类项：

 $(2y = 24)$ 

解得  $(y = 12)$ ，即兔子有12只。

将  $(y)$  的值代入任一原方程求解  $(x)$ ：

 $(x + 12 = 35)$ 

解得  $(x = 23)$ ，即鸡有23只。

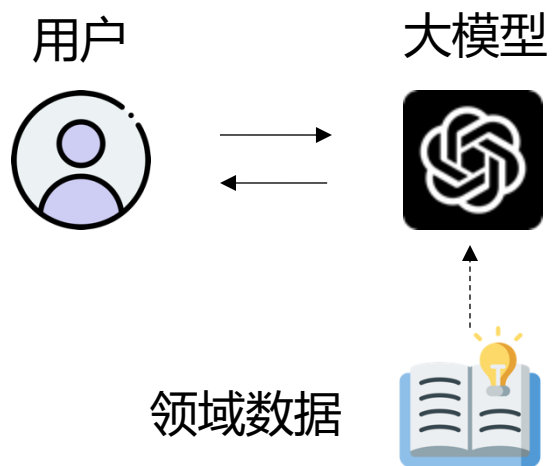
所以，鸡和兔的数量分别是23只和12只。
```

进阶：

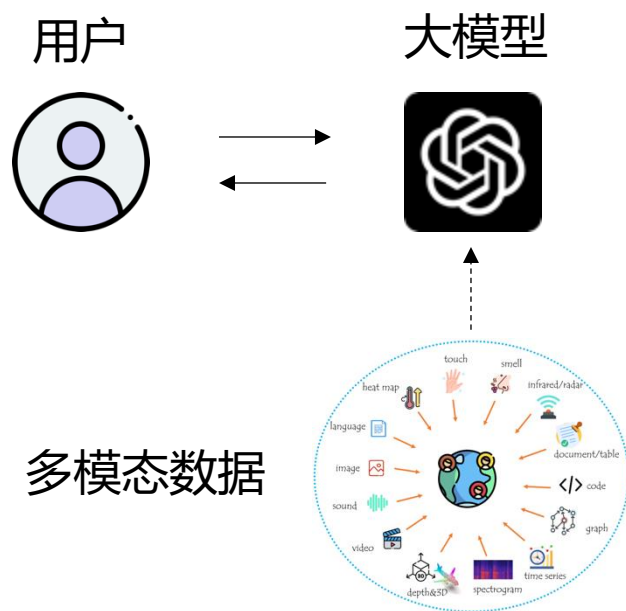
1. 使用open web UI配置WebUI
2. 基于AnythingLLM搭建本地知识库

- ❑ 大模型概述
 - 发展概览
 - 大模型范畴
- ❑ 预训练模型
 - 主要类型
 - 模型架构
- ❑ 大规模模型
 - 范式变化
 - 关键技术
 - 主要代表
 - 部署方式
- ❑ 概念延伸
 - 垂直大模型
 - 多模态基座
 - 智能体基座

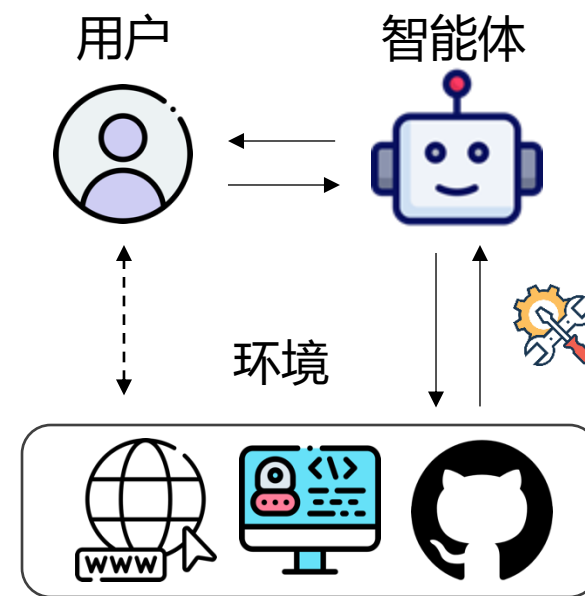
垂域大模型



多模态大模型



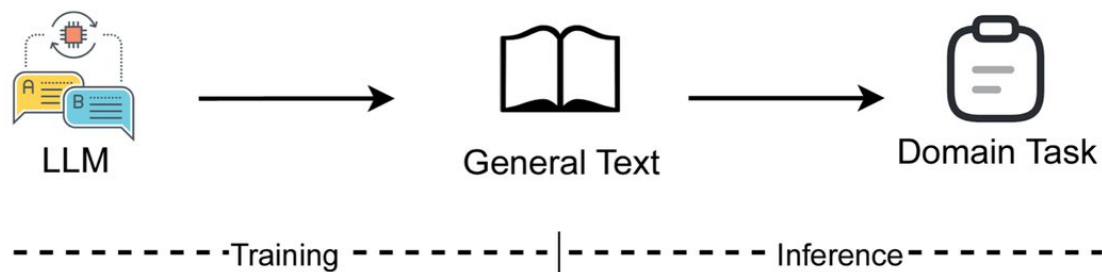
大模型智能体



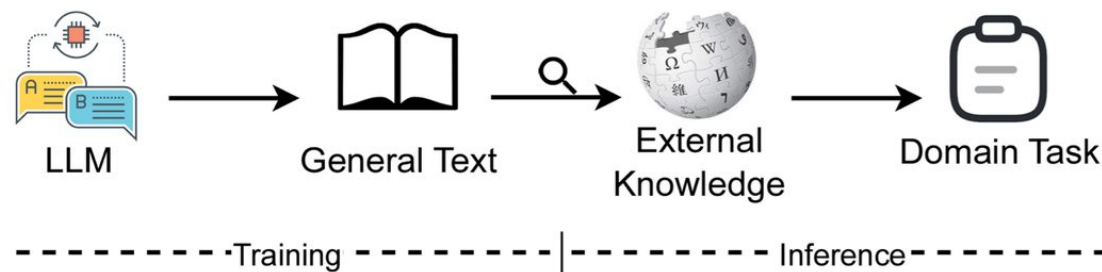
垂域大模型

- ❑ **基本范式：**在通用大模型基础上进行领域训练增强，专业性强、落地速度快、成本效益高、合规性好
- ❑ **训练方式：**领域数据训练（继续预训练、微调、提示）、检索增强等
- ❑ **常见领域：**教育、医疗、法律、金融等

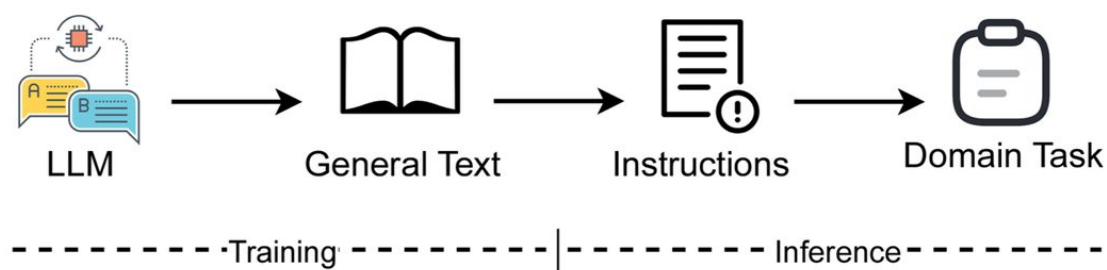
a) Deploying LLM without Domain Specialization



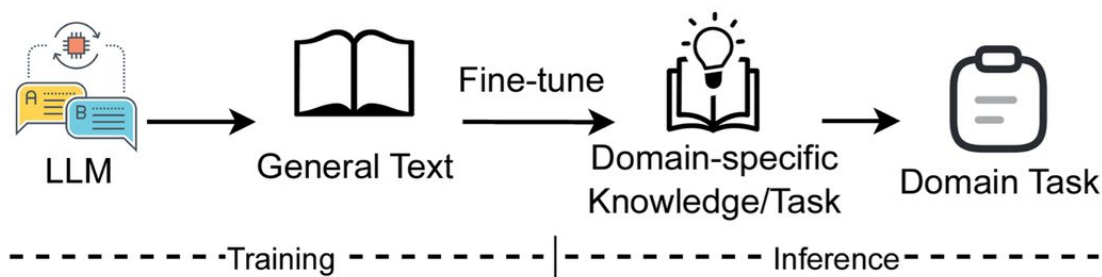
b) Retrieval Augmented Domain Specialization



c) Instruction-crafting-based Domain Specialization



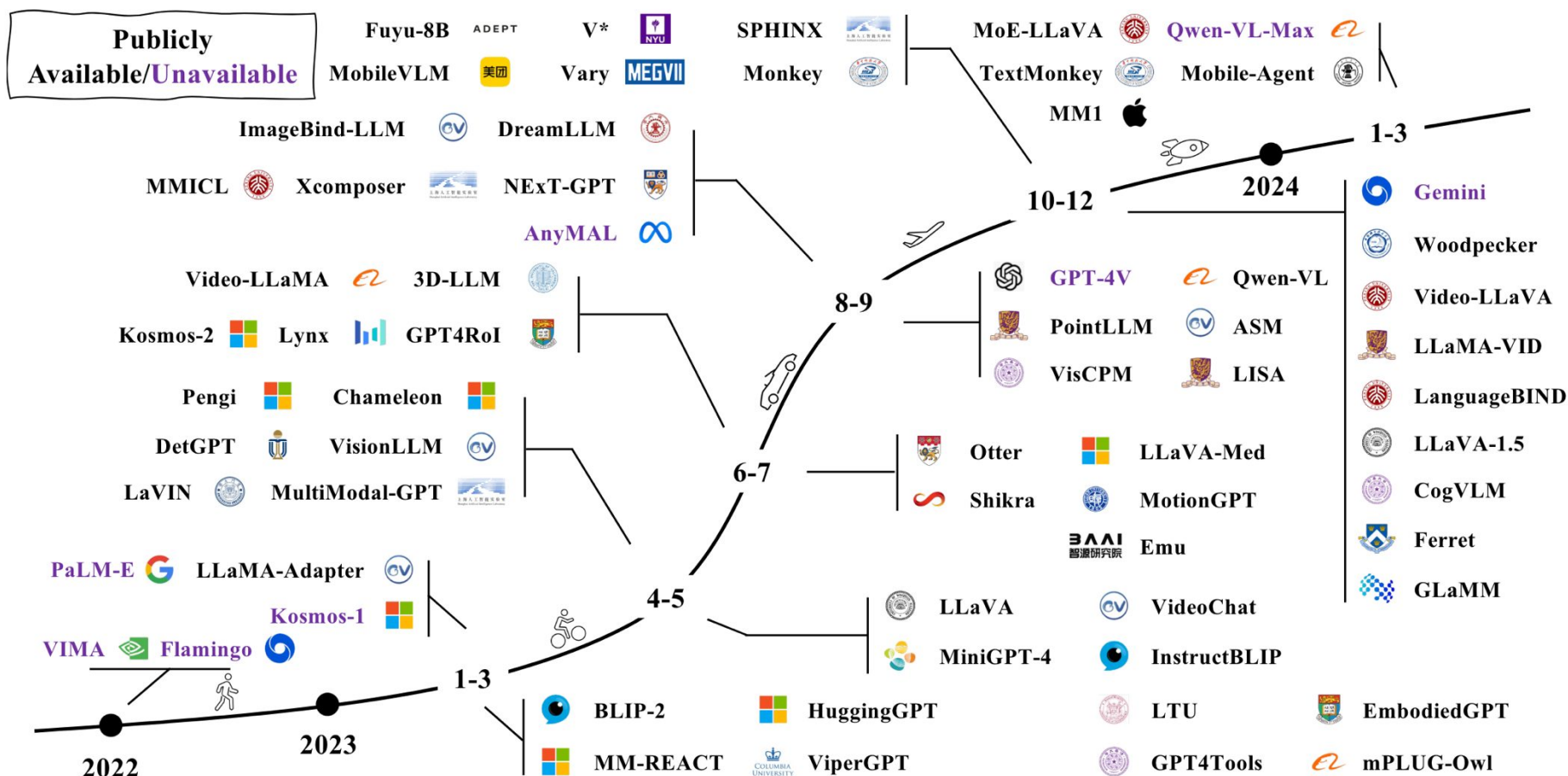
d) Knowledge-updated Domain Specialization



[1] Ling, Chen, et al. Domain specialization as the key to make large language models disruptive: A comprehensive survey. arXiv preprint arXiv:2305.18703.

多模态大模型的前沿发展

- ❑ **多模态大模型**的能力突破，成为大模型前沿发展的新趋势和焦点
- ❑ 极大拓展大模型能力边界和应用场景，智能跃迁的下一个关键引擎



多模态大模型的能力样例

Unesco announces its newest geoparks around the world

18th April 2024
By Lynn Brown, Features correspondent



Please identify the text in the picture.

MiniCPM-Llama3-V 2.5

Title:
Unesco announces its newest geoparks around the world

Date and Author:
9 April 2024
By Lynn Brown, Features correspondent

Image Caption:
Unesco Geoparks represent a balance of unique geological features, cultural touchpoints and a focus on sustainability (Credit: Chris Karagelis)

Body Text:
From dinosaur fossils in Brazil to the soaring meteors of Meteora, these 18 new geological sites highlight a particular region's natural, cultural and intangible heritage.

Travellers interested in deep dives into geology, culture and sustainability have several new destinations to place on their bucket list. Unesco just announced the designation of 18 new Global Geoparks, bringing the total to 213 sites representing 48 countries around the world.

From crystal-clear karst lakes to an archipelago formed during the last ice age, these parks are more than just sites with unique geological features; they also serve to highlight a particular region's natural, cultural and intangible heritage. "Unesco Global Geoparks represent landscapes where internationally significant geological heritage intertwines with human history and supports rich biodiversity," explains António Abreu, director of the Division of Ecological Sciences at Unesco. "Each new geopark tells a unique story, using geology as its lens."

Here are five of the newest geoparks on Unesco's list:

MiniCPM-Llama3-V 2.5

模型名称	发布时间	发布机构	语言	参数规模	tokens规模	模型结构	是否开源
T5	2019-10	Google	英	13B	130B	T5-style	✓
GPT-3	2020-05	OpenAI	英	175B	300B	GPT-style	×
LaMDA	2021-05	Google	英	137B	2.8T	GPT-style	×
Jurassic	2021-08	AI21	英	178B	300B	GPT-style	×
MT-NLG	2021-10	Microsoft, NVIDIA	英	530B	270B	GPT-style	×
ERNIE 3.0	2021-12	Baidu	中	260B	300B	Multi-task	×
Gemini	2023-04	DeepMind	英	280B	300B	GPT-style	×
Chinchilla	2023-04	DeepMind	英	70B	1.4T	GPT-style	×
PaLM	2022-04	Google	多语言	540B	780B	GPT-style	×
BLOOM	2022-07	BigScience	多语言	176B	366B	GPT-style	✓
GLM-130B	2022-08	Tsinghua	中、英	130B	400B	GLM-style	✓
LLaMA	2023-02	Meta	多语言	7B-65B	1.4T	GPT-style	✓

请将表格转成 markdown 格式

MiniCPM-Llama3-V 2.5

模型名称	发布时间	发布机构	语言	参数规模	Tokens规模	模型结构	是否开源
T5	2019-10	Google	英	13B	130B	T5-style	✓
GPT-3	2020-05	OpenAI	英	175B	300B	GPT-style	×
LaMDA	2021-05	Google	英	137B	2.8T	GPT-style	×
Jurassic	2021-08	AI21	英	178B	300B	GPT-style	×
MT-NLG	2021-10	Microsoft, NVIDIA	英	530B	270B	GPT-style	×
ERNIE 3.0	2021-12	Baidu	中	260B	300B	Multi-task	×
Gopher	2021-12	DeepMind	英	280B	300B	GPT-style	×
Chinchilla	2022-04	DeepMind	英	70B	1.4T	GPT-style	×
PaLM	2022-04	Google	多语言	540B	7.8/80B	GPT-style	×
OPT	2022-05	Meta	英	125M-175B	180B	GPT-style	✓
BLOOM	2022-07	BigScience	多语言	176B	366B	GPT-style	✓
GLM-130B	2022-08	Tsinghua	中、英	130B	400B	GLM-style	✓
LLaMA	2023-02	Meta	多语言	7B-65B	1.4T	GPT-style	✓

面壁Ultra对齐技术
大模型上神器！

面壁Ultra对齐技术，是面壁大模型研发团队自主研发的，旨在提升大模型在复杂任务中的推理能力。该技术通过引入先进的对齐算法，使得模型能够更好地理解用户意图，并生成更符合预期的回答。面壁Ultra对齐技术已在多个大模型中得到应用，显著提升了模型的推理性能和用户体验。

请详细总结图片中文章的要害。

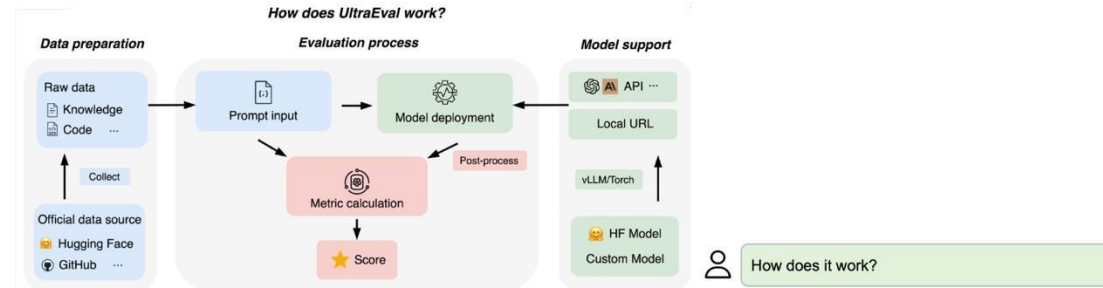
MiniCPM-Llama3-V 2.5

这张图片是一篇关于面壁Ultra对齐技术的文章摘要。以下是文章的要害：

- 面壁Ultra介绍：**面壁Ultra是专门设计用于提升大模型推理能力的大规模、高质量对齐数据集的工具，包含了对齐数据集、代码推理问题的12个开源数据集的86k条指令和220k个偏好对。
- 面壁Ultra对齐数据集的特点：**面壁Ultra提供了一个包含多种类型对齐数据集的平台，这些数据经过严格质量控制和筛选，确保了数据的可靠性和多样性。
- 推理过程中的挑战：**在推理过程中，面壁Ultra通过自动化格式检查和人工质量检查保证了推理的准确性和效率。
- 模型推理的优化：**面壁Ultra推理SOTA（State-Of-The-Art）的Eurus-8x22B模型，并展示了它在推理任务中的表现。
- 模型推理的复杂性和效率：**面壁Ultra使用了多轮交互的方式来提高模型推理的效率和准确性。这包括了在推理过程中不断调整参数以达到最佳性能。
- 面壁Ultra的优势：**文章提到面壁Ultra作为一个系统工程，拥有自研的全流程高生产效率线，这使得它能够快速地进行模型推理和优化。

综上所述，面壁Ultra是一个专注于推进自然语言处理领域对齐数据集的平台，旨在为模型推理提供更高效率和准确的解决方案。

How does UltraEval work?



MiniCPM-Llama3-V 2.5

UltraEval appears to be a system or process designed to evaluate data, likely in the context of a scientific or research project. The flowchart outlines the steps involved in this process:

- Data Preparation:** This stage involves collecting and organizing raw data, which can come from official data sources such as GitHub or Hugging Face. These are represented by icons indicating their origin.
- Evaluation Process:** After the data is prepared, it undergoes an evaluation process that includes several steps:
 - Prompt Input:** This could involve providing prompts or instructions for the model.
 - Model Deployment:** Here, the model is likely trained or configured to perform evaluations based on the input.
 - Local URL:** This step may involve accessing or using a local URL, possibly for deploying the model or retrieving results.
 - Post-process:** This could refer to further processing of the model's output, such as cleaning, filtering, or formatting data.
- Metric Calculation:** Following the post-process, metrics are calculated to evaluate the performance of the model. This is indicated by a graph icon, suggesting quantitative analysis.
- Score:** The final stage is to determine the score, which is likely the outcome of the metric calculation. This score would reflect the model's performance or the quality of the data after evaluation.
- Model Support:** Throughout the process, there is support for various models, including vLLM/Torch, HF Model, and Custom Model. These models are likely used at different stages of the evaluation process.

In summary, UltraEval seems to be a structured approach to evaluating data using machine learning models, with a focus on performance metrics and customization options for different types of models.

多模态大模型的前沿发展

技术概要

主要目标：基于大数据和人类反馈进行学习，与用户通过对话进行交互，能够处理多模态信息及多种任务，深层推理与常识运用能力大幅度提升

关键技术：多模态模型架构、多模态指令遵循、多模态推理、轻量化部署、多模态对齐等

里程碑事件

2023/03/15: OpenAI 发布**多模态对话模型 GPT-4**

2023/09/20: OpenAI 发布**文生图模型 DALL·E 3**

2023/09/25: ChatGPT 全面整合**多模态能力**

2023/12/06: Google 发布**多模态模型 Gemini**

2024/03/04 : Anthropic发布**多模态模型 Claude 3**

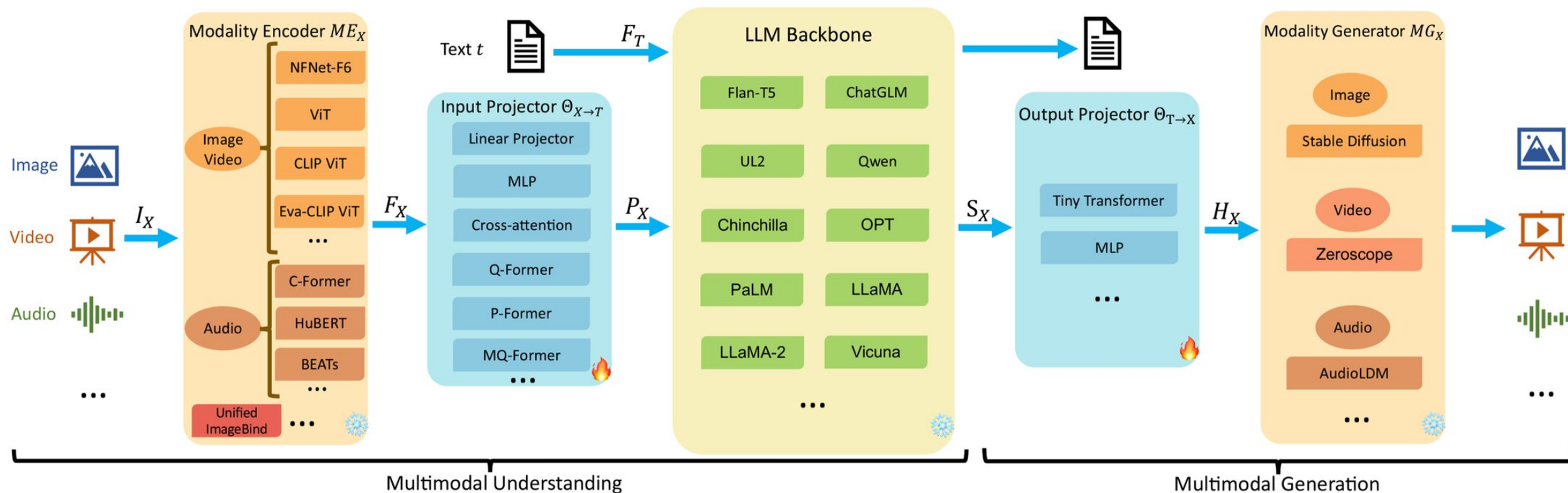
2024/05/13: OpenAI发布原生**多模态模型 GPT-4o**

工作范式的变化



多模态大模型的前沿发展

- ❑ **通用工作模式：**多模态编码 > 特征融合模块 > 多模态解码器，支持多模态输入与多模态生成
- ❑ **常见模态：**文本、图像、视频、语音等

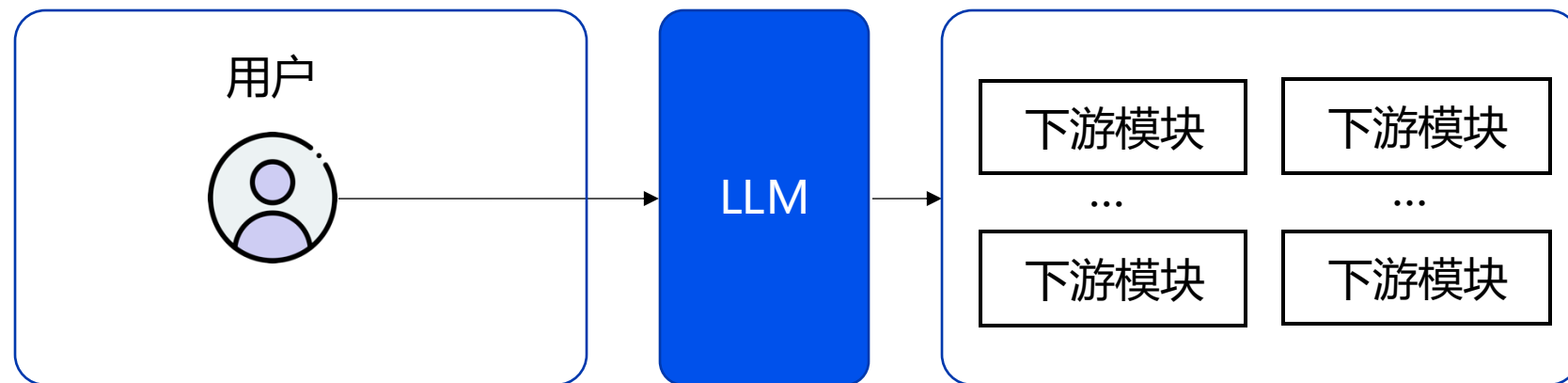


[1] Zhang, Duzhen, et al. Mm-llms: Recent advances in multimodal large language models. arXiv preprint arXiv:2401.13601.

多模态大模型的代表性架构

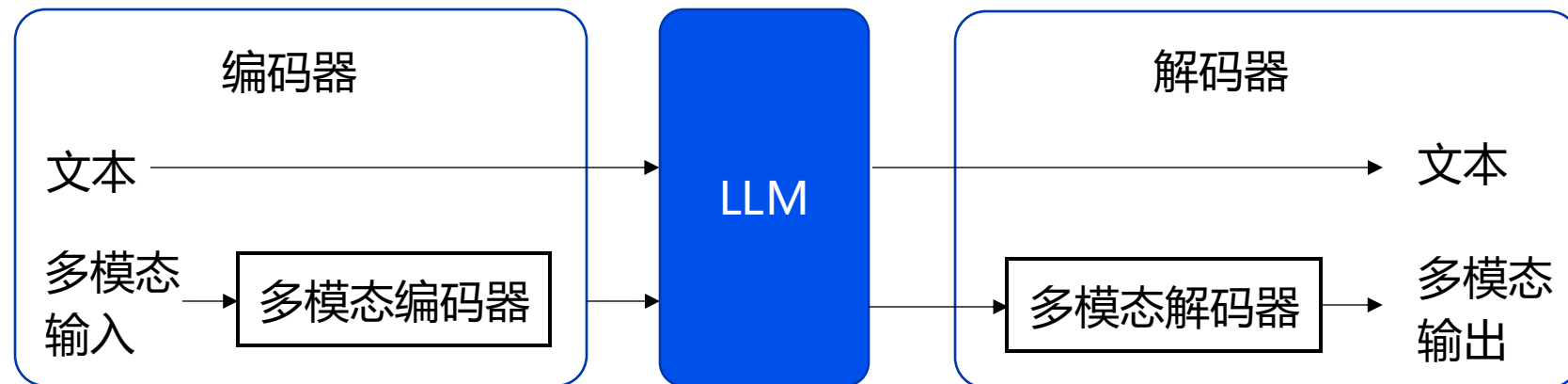
LLM为中心的架构

- + Visual-ChatGPT
- + HuggingGPT
- + MM-REACT
- + LLaVA-Plus
- + ...



一体化多模态架构

大多数多模态大模型的架构类型



多模态大模型的前沿教程 (CVPR & MM & COLING)



Tutorial

From Multimodal LLM to Human-level AI: Modality, Instruction, Reasoning and Beyond

Hao Fei · Yuan Yao · Ao Zhang · Haotian Liu · Fuxiao Liu · Zhuosheng Zhang · Shuicheng Yan

Summit 446

[[Abstract](#)] [[Project Page](#)]

Tue 18 Jun 1:30 p.m. PDT — 6 p.m. PDT

Abstract:

Artificial intelligence (AI) encompasses knowledge acquisition and real-world grounding across various modalities. As a multidisciplinary research field, multimodal large language models (MLLMs) have recently garnered growing interest in both academia and industry, showing an unprecedented trend to achieve human-level AI via MLLMs. These large models offer an effective vehicle for understanding, reasoning, and planning by integrating and modeling diverse information modalities, including language, visual, auditory, and sensory data. This tutorial aims to deliver a comprehensive review of cutting-edge research in MLLMs, focusing on three key areas: MLLM architecture design, instructional learning, and multimodal reasoning of MLLMs. We will explore technical advancements, synthesize key challenges, and discuss potential avenues for future research. All the resources and materials will be made available online: <https://mlm2024.github.io/CVPR2024>

SCHEDULE

PROGRAM

Our tutorial will be held on **Tuesday, June 18, 2024** (all the times are based on **UTC-7 = Seattle local time**).

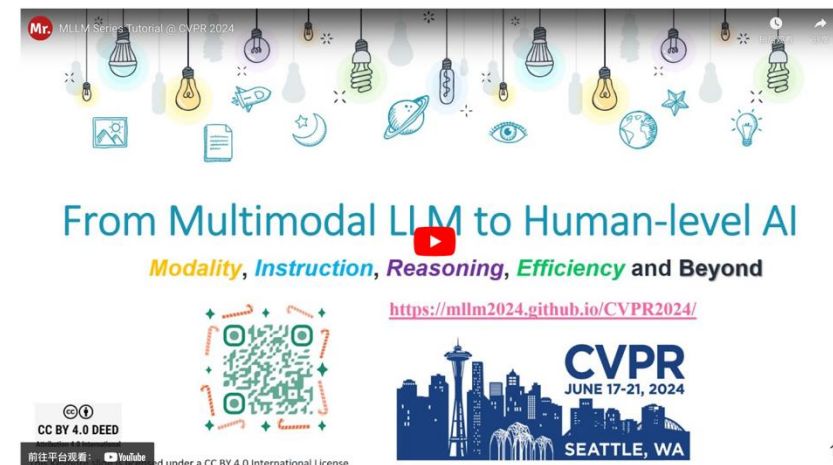
Time	Section	Presenter
13:30-13:35	Part 1: Background and Introduction [Slides]	Hao Fei
13:35-14:05	Part 2: MLLM Architecture [Slides]	Yuan Yao
14:05-14:35	Part 3: MLLM Modality&Functionality [Slides]	Hao Fei
14:35-15:05	Part 4: MLLM Instruction Tuning [Slides]	Haotian Liu
	Coffee Break, Q&A Session	
16:00-16:30	Part 5: MLLM Hallucination [Slides]	Fuxiao Liu
16:30-17:00	Part 6: MM Reasoning [Slides]	Zhuosheng Zhang
17:00-17:30	Part 7: MLLM Efficiency [Slides]	Ao Zhang
17:30-18:00	Part 8: Panel Discussion - From MM Generalist to Human-level AI	All + Hanwang Zhang + Shuicheng Yan



讲义 + 视频

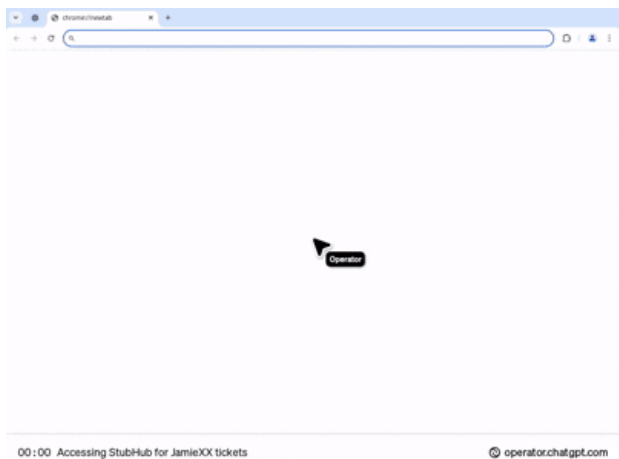
<https://mlm2024.github.io/CVPR2024>

TUTORIAL RECORD VIDEO



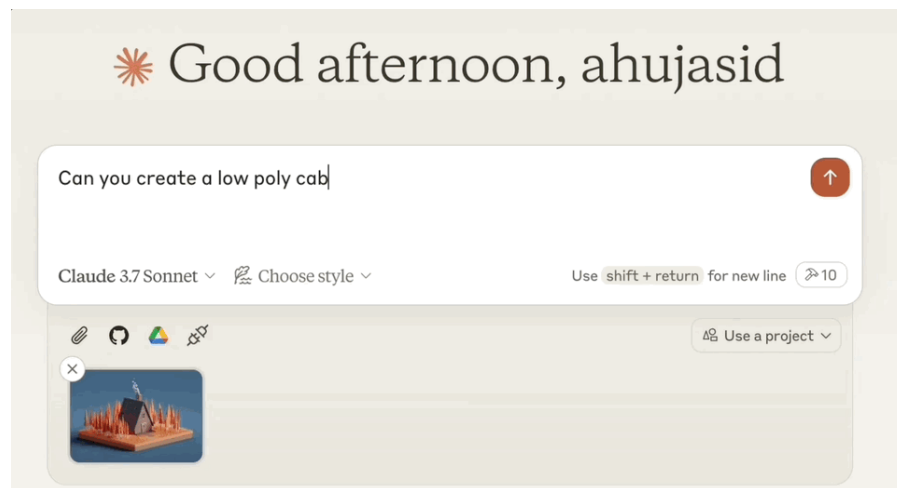
无处不在的“贾维斯”，大模型智能体展现出广泛的应用前景

OpenAI Operator



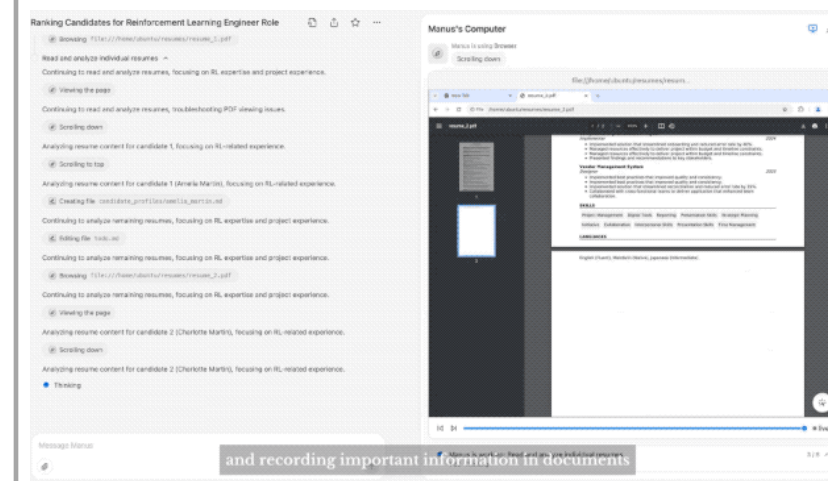
<https://openai.com/index/introducing-operator>

Anthropic Claude



<https://www.anthropic.com/claude>

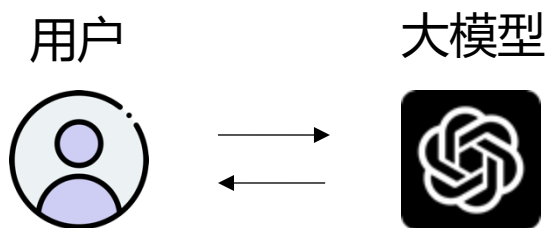
Manus AI



<https://manus.im/>

- ❑ 基于大模型构建，能够动态指导其自身流程和工具使用，并根据实时反馈调整自己的操作的**系统**¹

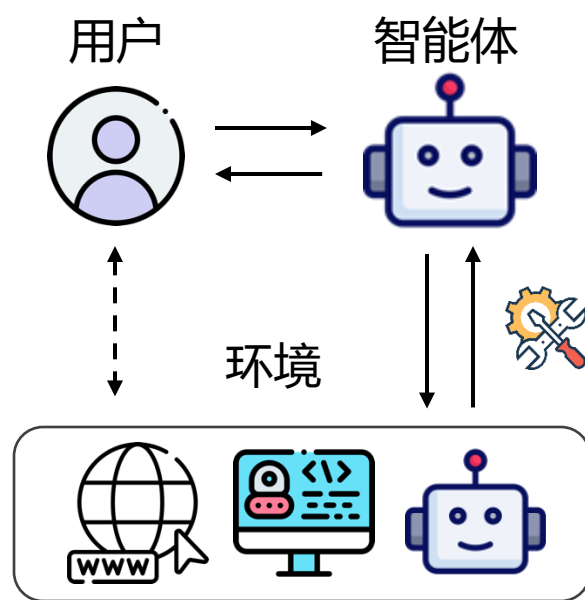
(多模态) 大语言模型



- ❑ **通用能力的提供者**

- 长序列处理
- 思维链推理
- 广泛意图理解

大模型智能体 (系统)



- ❑ **具备自主和进化能力**

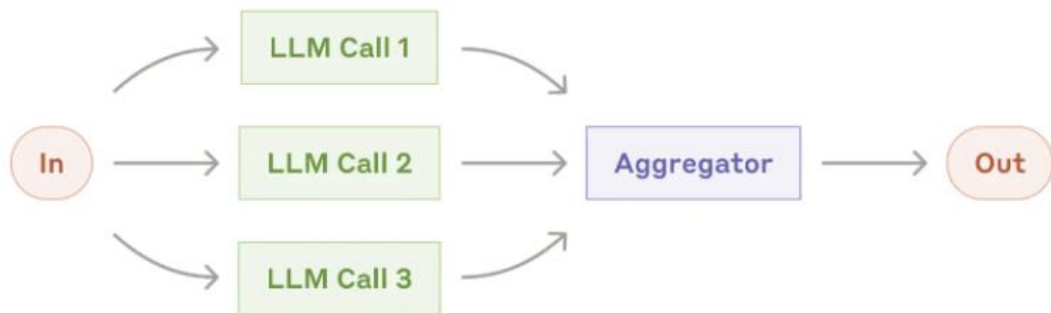
- **环境交互**：感知理解、规划决策、动作落实
- **持久运行**：反思优化、迭代演化、安全防护

- ❑ **知识更新**：知识库、经验库

- ❑ **集成工具**：APP、API、函数

¹Anthropic. Building effective agents. Published Dec 19, 2024

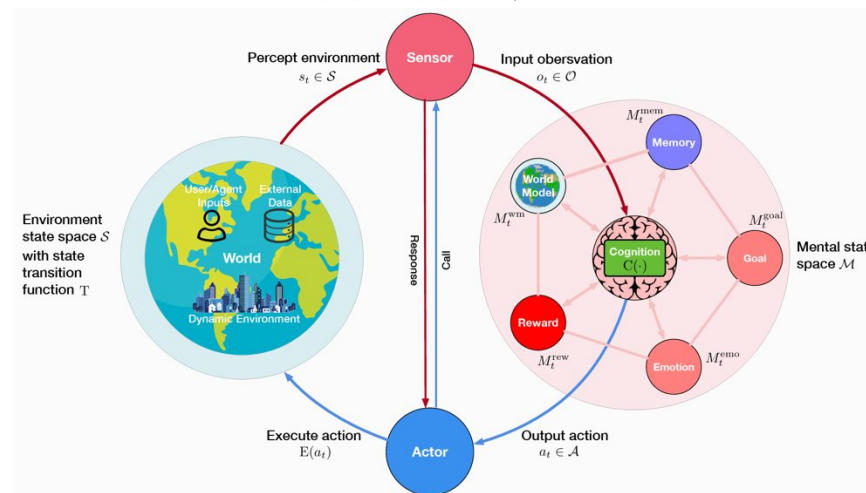
workflow编排



通过预定义的路径编排LLM和工具，类似复杂的自动化脚本。常以Prompt方式实现

将AI视为可编排的组件

智能体基座

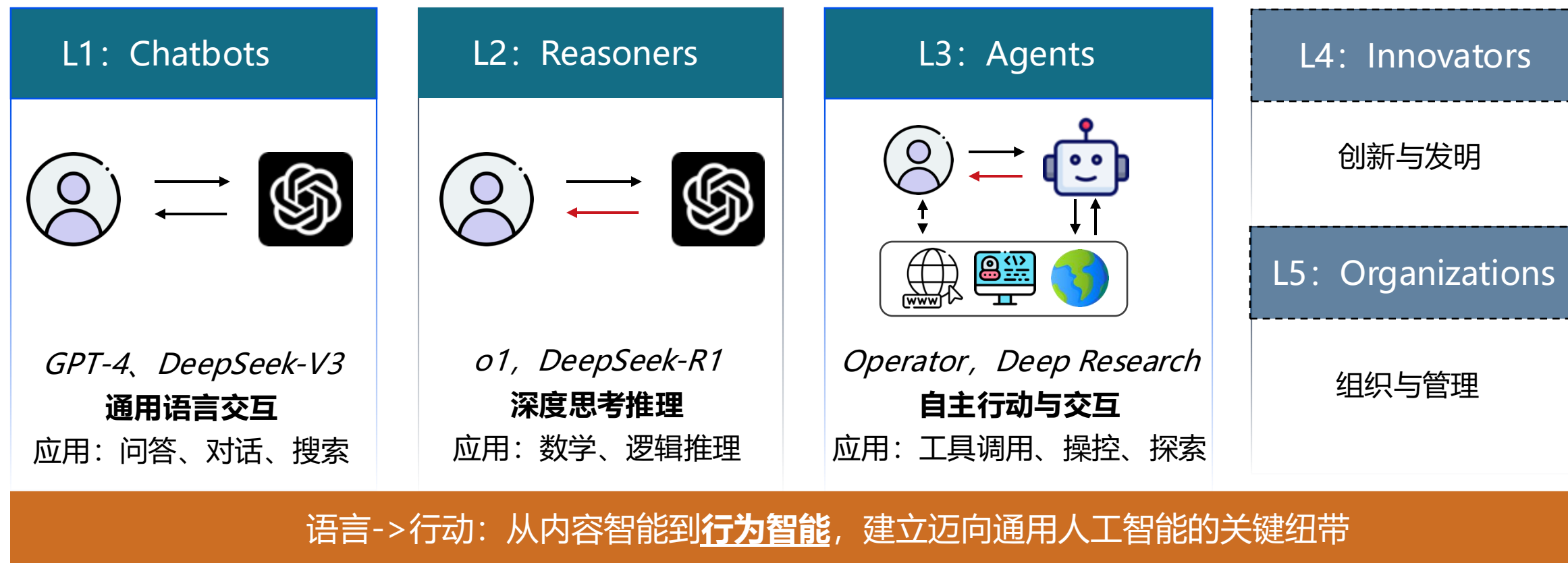


自主指导行为过程和工具使用，动态规划和更新路径，保持对任务完成方式的控制。常以训练方式拓展能力

给予AI更大的自主性、可进化

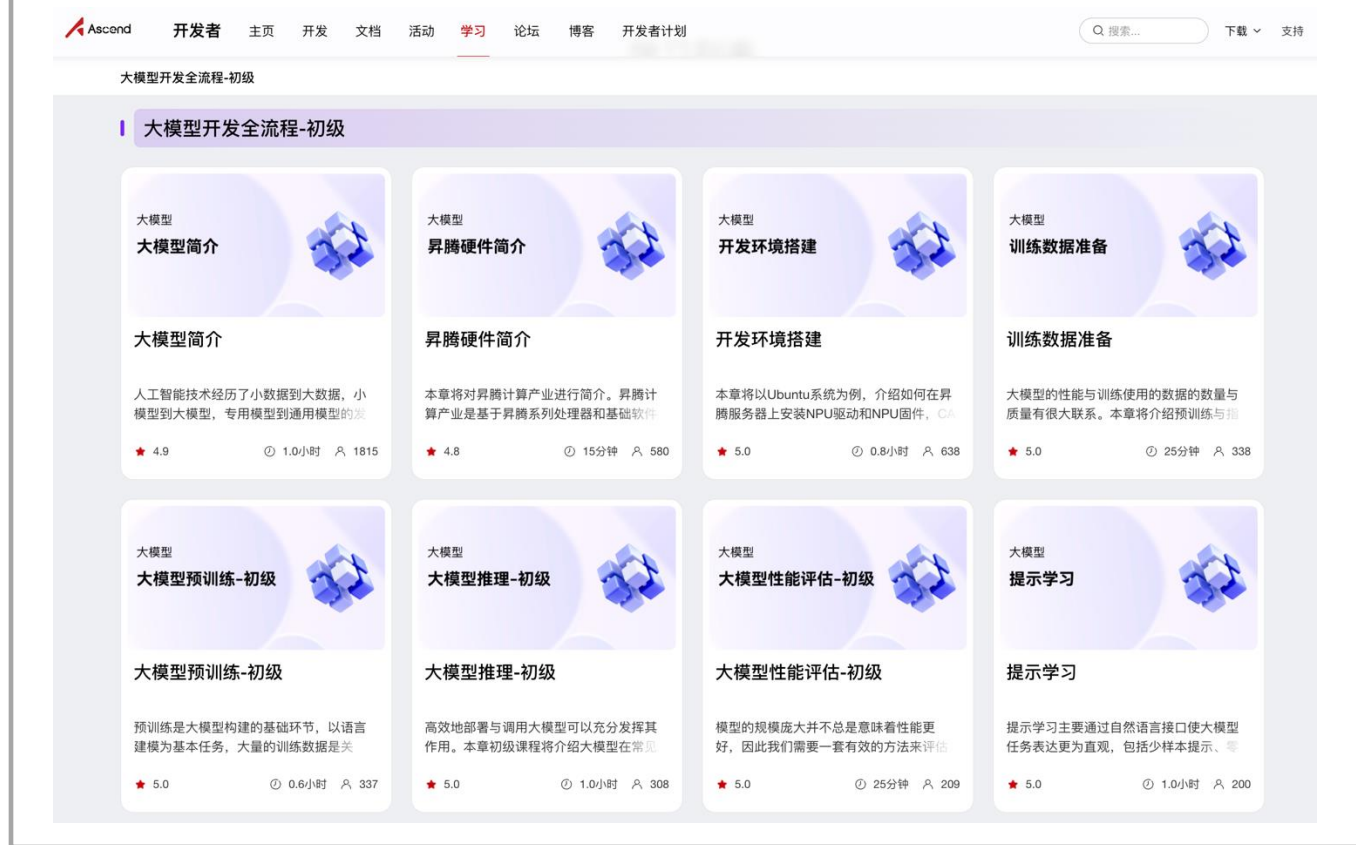
突破大模型的能力边界

□ **大模型智能体**的能力突破，成为新焦点，极大拓展大模型能力边界和应用场景



《大模型开发全流程》

国产化大模型课程：54节课程、26份实践手册



The screenshot displays the 'Large Model Development Full Process - Beginner' (大模型开发全流程-初级) course section on the Ascend developer website. The page features a grid of eight course cards, each with a title, description, rating, duration, and view count.

课程名称	描述	评分	时长	浏览量
大模型简介	人工智能技术经历了小数据到大数据，小模型到大模型，专用模型到通用模型的发展。	4.9	1.0小时	1815
昇腾硬件简介	本章将对昇腾计算产业进行简介。昇腾计算产业是基于昇腾系列处理器和基础软件。	4.8	15分钟	580
开发环境搭建	本章将以Ubuntu系统为例，介绍如何在昇腾服务器上安装NPU驱动和NPU固件，CA。	5.0	0.8小时	638
训练数据准备	大模型的性能与训练使用的数据的数据量与质量有很大联系。本章将介绍预训练与指。	5.0	25分钟	338
大模型预训练-初级	预训练是大模型构建的基础环节，以语言建模为基本任务，大量的训练数据是关键。	5.0	0.6小时	337
大模型推理-初级	高效地部署与调用大模型可以充分发挥其作用。本章初级课程将介绍大模型在常见。	5.0	1.0小时	308
大模型性能评估-初级	模型的规模庞大并不总是意味着性能更好，因此我们需要一套有效的方法来评估。	5.0	25分钟	209
提示学习	提示学习主要通过自然语言接口使大模型任务表达更为直观，包括少样本提示、零。	5.0	1.0小时	200

<https://www.hiascend.com/edu/growth/lm-development>

谢谢！

zhangzs@sjtu.edu.cn
<https://bcmi.sjtu.edu.cn/~zhangzs>



饮水思源 爱国荣校