基础语言模型

代季峰

课程目录

- **预训练语言模型**
- 大语言模型的应用

This chapter is revised from CS 404/504 & CS231N

词结构和子词（subword）模型

- 让我们看一下我们对语言词汇所做的假设.
- 我们假设根据训练集构建了一个有数万个单词的固定词汇表.
- 测试时看到的所有新词都被映射到UNK

	word		vocab mapping	embedding
Common words	hat	→	hat	
	learn	→	learn	
Variations	taaaaasty	→	UNK	
misspellings	laern	→	UNK	
novel items	Transformerify	→	UNK	

字节对编码（Byte-Pair Encoding，BPE）算法

背景介绍

- 在自然语言处理（NLP）中，词汇建模通常需要对词语进行更细粒度的处理，如将单词拆分为子词、字符或字节等更小的单位
- 现代主流范式采用 子词标记（parts of words） 的词汇表来进行学习
- 在训练和测试过程中，每个单词被拆分成一系列预定义的子词单元

字节对编码（Byte-Pair Encoding，BPE）算法概述

BPE是定义子词词汇表的一种简单有效的策略，步骤如下：

1. 以仅包含字符和“词尾”符号的词汇表开始.
2. 使用文本语料库，找到最常见的相邻字符 “a,b”; 添加 “ab” 作为子词.
3. 用新的子词替换字符对的例子; 一直重复直到所需的词汇量

字节对编码（Byte-Pair Encoding，BPE）算法

假设语料如下：

lua 复制 编辑

```lua
low, lower, newest, widest
```

初始字符表示（以空格分隔）：

bash 复制 编辑

```bash
l o w</w>
l o w e r</w>
n e w e s t</w>
w i d e s t</w>
```

第一步，统计相邻字符对出现频率：

- `l o`、`o w`、`w</w>`、`w e` 等

然后找到最频繁的一对（比如 `e s`），合并为新符号 `es`，更新所有词中出现的位置，继续下一轮。

最终可能会得到如 `low`，`lower`，`newest`，`widest` 被分成的子词如：

cpp 复制 编辑

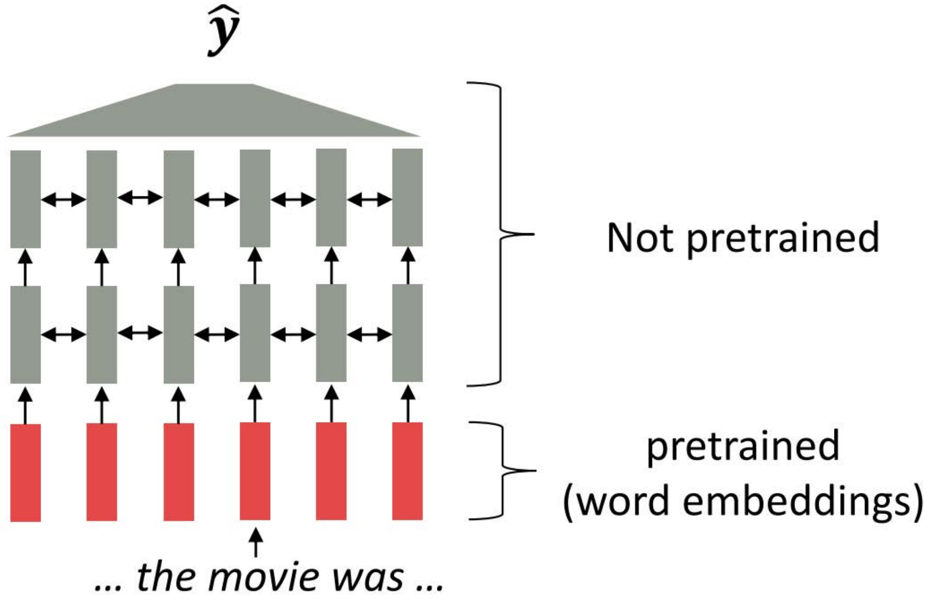
```cpp
low, er, new, est, wid, est
```

词结构和子词（subword）模型

- 常用词最终成为子词词汇表的一部分，而罕见的词被分割
- 成为 (有时直观，有时不直观) 的多个部分.
- 在最坏的情况下，单词会被分割成与其字符数量一样多的子单词.

预训练词嵌入

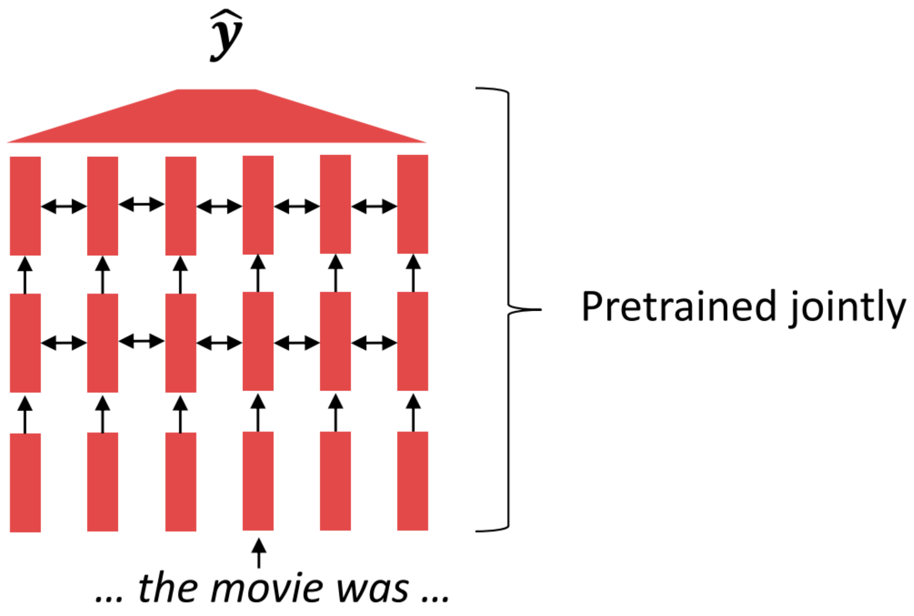
- 预训练词库中所有词的嵌入表示
- 一些难以解决的问题:
  - 下游任务(例如问答)可能涉及特定的术语或表达方式，这在预训练的语料库中可能并不常见。为下游任务提供的训练数据需要涵盖任务相关的所有语境特性。
  - 另外，网络中的大多数参数都将是随机初始化的!



[Recall, *movie* gets the same word embedding, no matter what sentence it shows up in]

预训练整个模型

- 在现代的NLP实践中
  - NLP网络中的所有（或者几乎所有）的参数都是通过预训练来初始化
  - 比如预训练方法可以从模型中隐藏部分输入，并训练模型来重建这些部分
- 有以下几方面的强大能力:
  - 不依赖额外标注，容易把规模做大
  - 蕴含丰富的语言表征和模式
  - 强大NLP模型的参数初始化：为后续特定任务的微调提供了有效的参数起点
  - 建模语言概率分布：用于文本生成

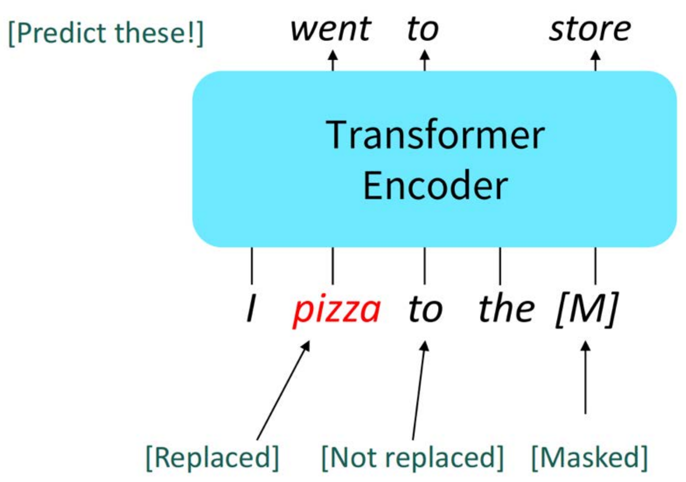


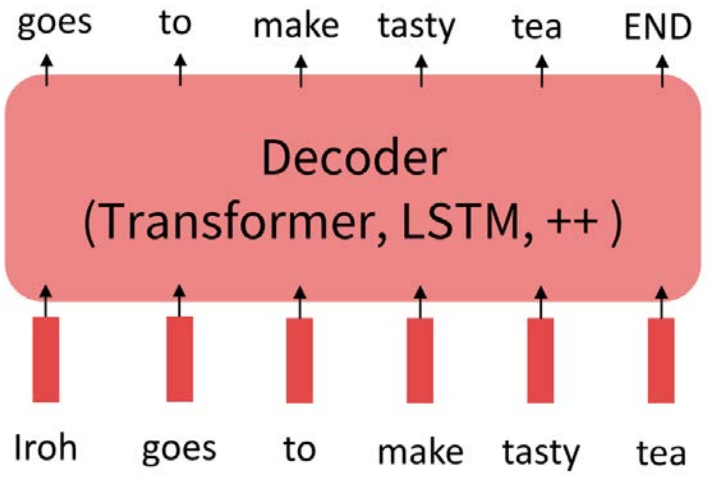
[This model has learned how to represent entire sentences through pretraining]

训练方式：语言建模

- 语言建模任务或其变种
  - 针对Encoder，预测被掩盖的单词
  - 使用Decoder，预测下一个单词
  - （接下来会详细阐述）

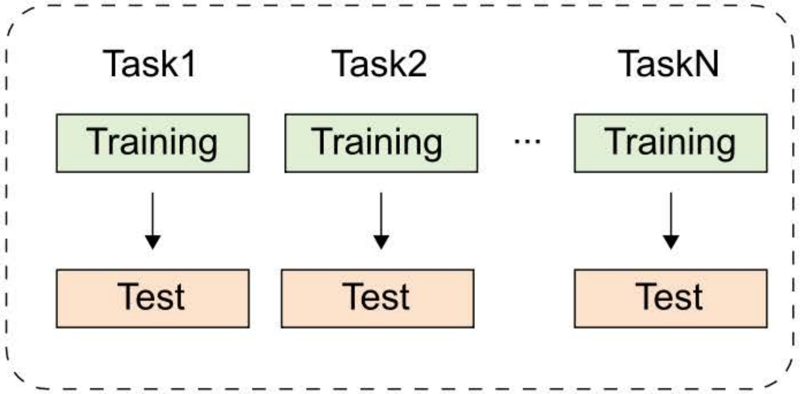
- 以语言建模为目标，在大量文本上进行预训练

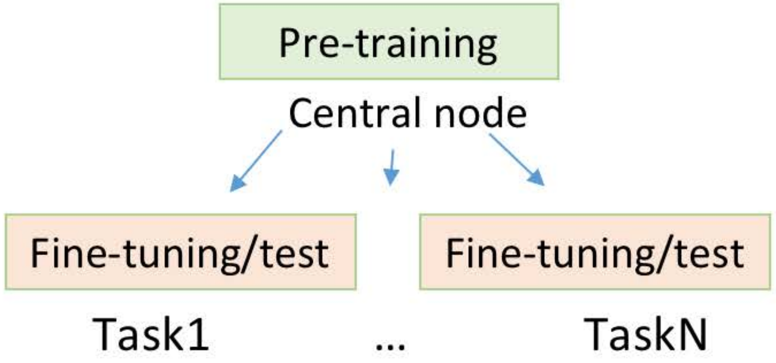




Dai A M, Le Q V. Semi-supervised sequence learning[J]. Advances in neural information processing systems, 2015, 28.

预训练/微调范式





Past

Develop the individual model for each task and finish both the training and test.

Now

The central node completes the large-scale pre-training of the general language model.
Other users borrow the existing pre-trained model as the standard module for further fine-tuning.

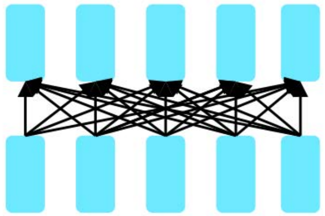
Individual training → Centralized pre-training + individual fine-tuning

随机梯度下降和预训练/微调

- 从“训练神经网络”角度来看，为什么预训练和微调会有好处?
- 预训练提供一个好的起始估计 $\hat{\theta}$
  - 预训练损失 $\min_{\theta} \mathcal{L}_{\text{pretrain}}(\theta)$
- 然后从起始估计 $\hat{\theta}$ 开始微调
  - 微调损失 $\min_{\theta} \mathcal{L}_{\text{finetune}}(\theta)$
- 预训练可能很重要，因为微调时，在随机梯度下降的作用下，参数将保持在 $\hat{\theta}$ 附近
  - 微调时得到的局部极小值点泛化得较好
  - 或者说微调损失$\mathcal{L}_{\text{finetune}}$ 在 $\hat{\theta}$ 附近的梯度可能“传播得很好”

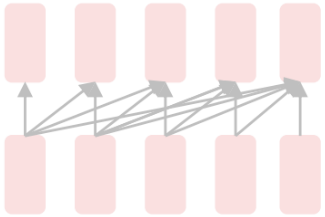
两种架构的预训练

神经网络的结构将影响预训练的方式以及模型的适用场景。



Encoders

- 获取双向上下文 – 能够以未来为条件: BERT



Decoders

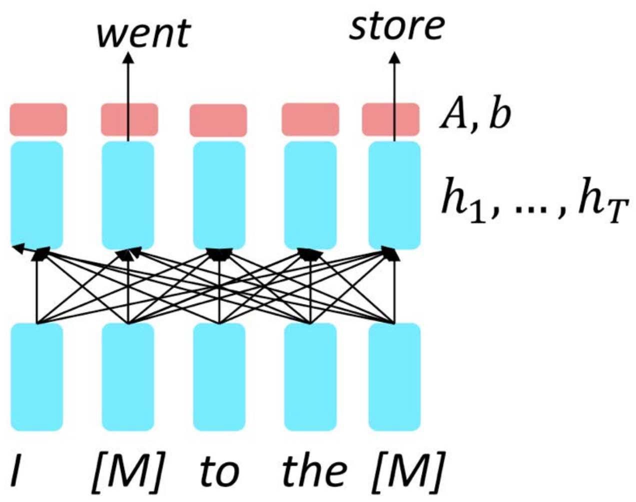
- 目前最常见的模型：GPT

预训练编码器: 预训练目标

- 我们主要研究语言模型的预训练。
- 但编码器获得了双向的上下文信息，因此无法进行语言建模。

- 想法：用特殊的[MASK]标记替换输入中的部分单词，然后预测这些单词

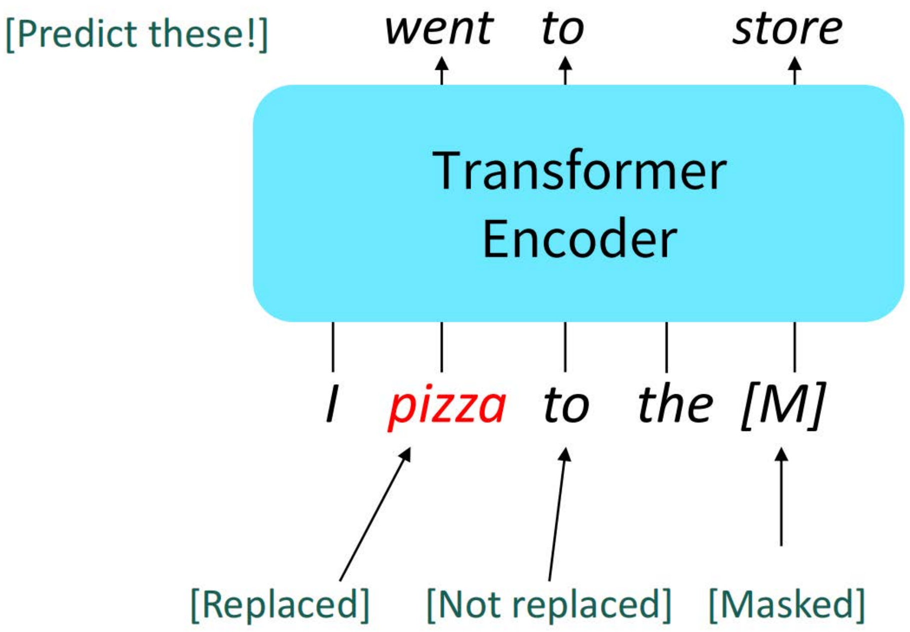
- 只从被“掩盖”的词计算损失：如果 $\tilde{x}$ 是 x 做掩盖之后的版本，我们学习的是 $p_\theta(x|\tilde{x})$, 这称作 Masked Language Model (Masked LM).



BERT：Transformers双向编码器

- Devlin等人在 2018年提出 “Masked LM” ，并发布预训练Transformer模型，称为 BERT

- BERT的细节:
- 预测随机选择的15%数量的单词token
  - 80%的情况下用[MASK]替换输入token
  - 10%的情况下用随机token替换
  - 10%的情况下保持输入token不变(但是仍然预测它)
- 设置后20%情况的目的：迫使模型在没有[MASK]标记提示的情况下，也能注意上下文信息 —— 在进行预测时，实际的文本中并没有[MASK]标记



[Devlin et al., 2018]

BERT Results

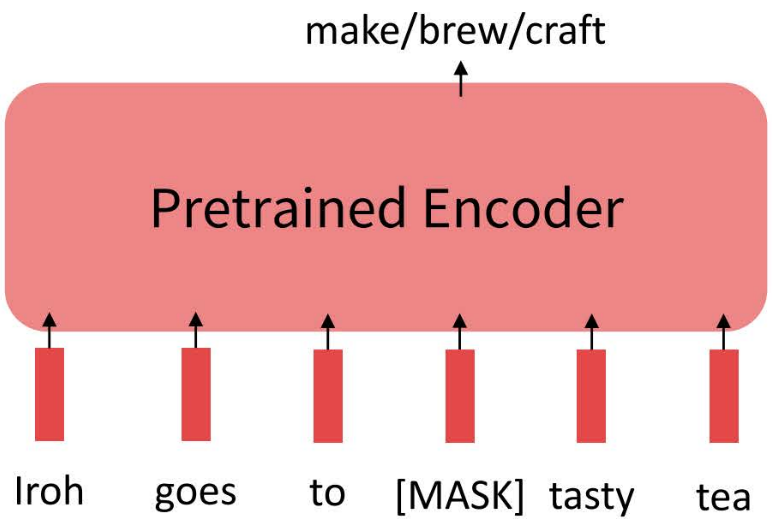
对BERT进行微调，在多种任务上得到了SOTA结果。

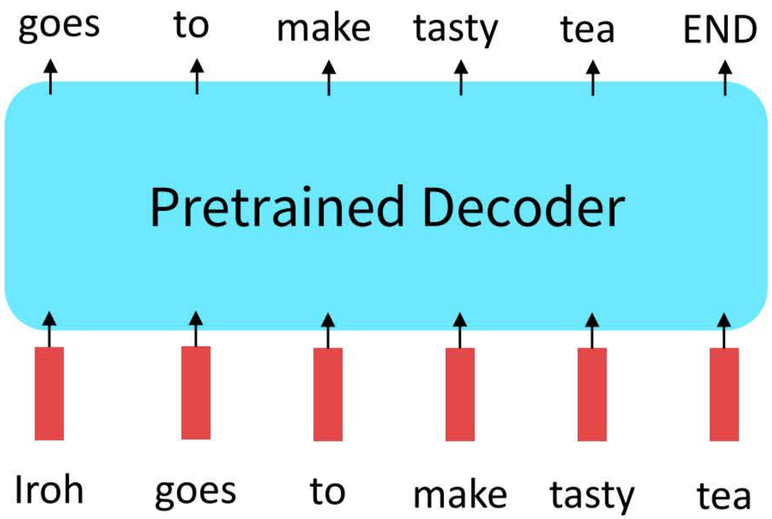
- Quora 问题对
- QNLI: 基于问答数据的自然语言推理
- SST-2: 情感分析
- CoLA: 语言可接受性语料库（检测句子是否符合语法）
- STS-B: 语义文本相似性
- MRPC: 微软改述语料库
- RTE: 一个小型的自然语言推理语料库

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

预训练编码器的局限性

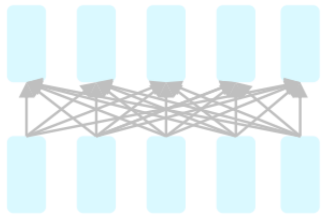
- BERT的结果（在当时看）非常优秀，但预训练编码器可以解决所有事情吗?
- 如果任务涉及生成序列，需要使用**预训练解码器**
- BERT和其他预训练编码器不能自然地适应自回归（一次一个单词）生成方法。





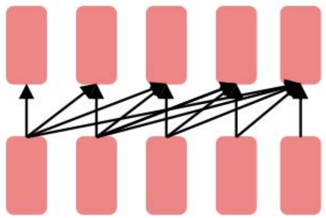
两种架构的预训练

神经网络的结构将影响预训练的方式以及模型的适用场景。



Encoders

- 获取双向上下文-可以以未来为条件!
- 我们如何训练它们建立有效的表征?
- BERT

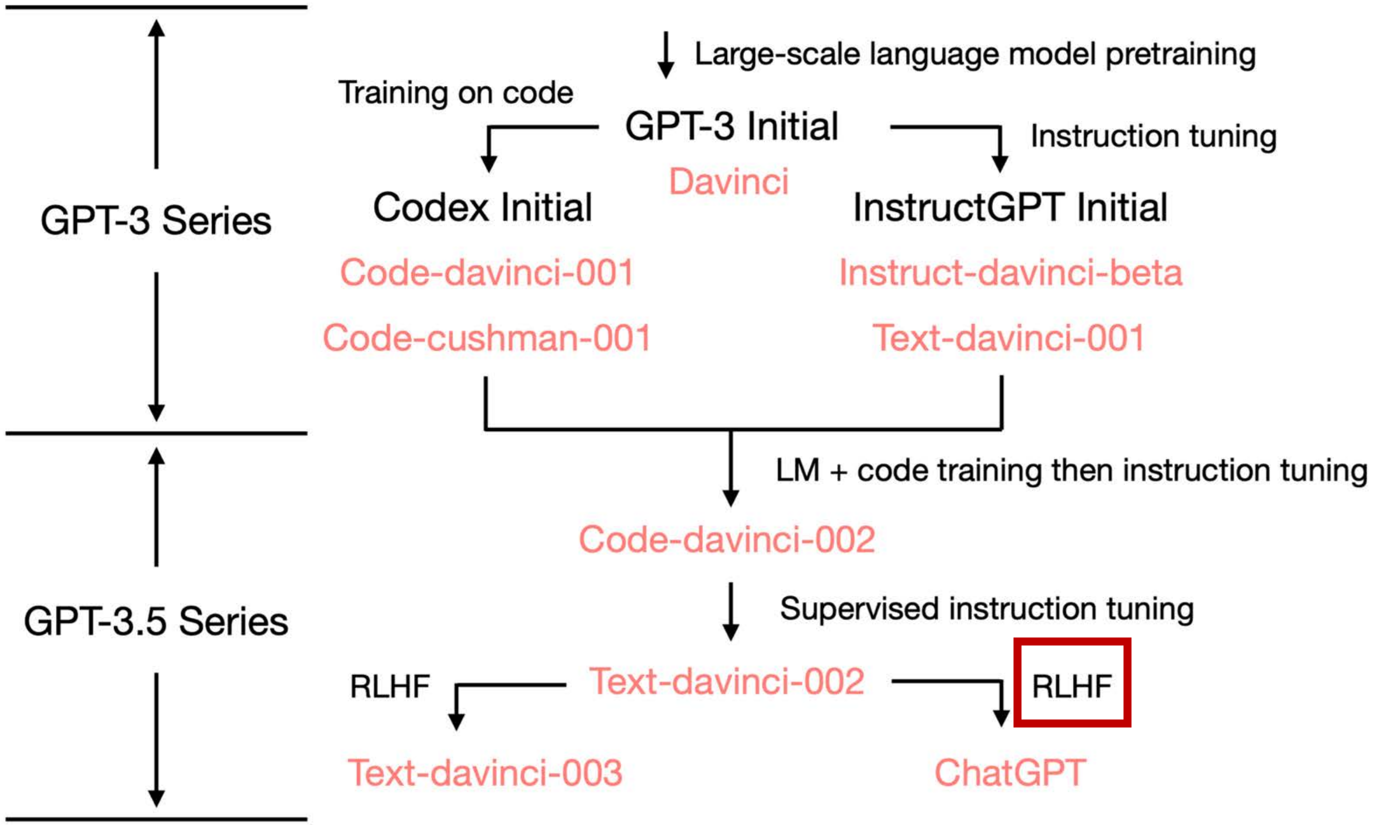


Decoders

- 目前最常见的模型: GPT

GPT家族：时间线

时间	事件
18年6月	GPT-1 在OpenAI博客发布
19年2月	GPT-2 在OpenAI博客发布
20年5月	GPT-3 论文发布在arXiv
20年5月	GPT-3 API Beta 测试
20年9月	GPT-3 授权给Microsoft
21年11月	GPT-3 API 开放
22年1月	InstructGPT发布（GPT-3.5）
22年11月28日	GPT-3.5拓展到davinci-003
22年11月28日	ChatGPT在OpenAI博客发布
23年3月14日	GPT-4发布
23年9月25日	GPT-4V发布



Generative Pretrained Transformer (GPT-1)

2018年， GPT-1 取得了巨大的成功

- 是一个12层, 117M参数的 Transformer 解码器
- 768维隐藏状态
- 使用12个注意力头
- 文本输入序列最大长度为512
- 在 BooksCorpus进行训练: 包括超过 7000 种独特的书籍
- 数据中包含了大量长跨度的连续文本，用于学习长距离依赖性.

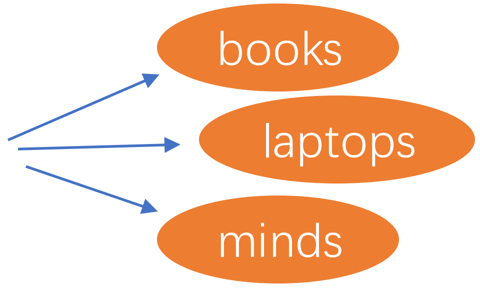
Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training[J]. 2018.

Generative Pretrained Transformer (GPT-1)

- GPT-1的预训练目标即为学习语言建模

- 语言建模的目标：给定一段文本，预测下一个单词是什么

The students opened their ________.



- 用公式来表示：

  给定一段文本$x^{(1)}, x^{(2)}, \dots, x^{(t)}$，我们需要计算下一个单词 $x^{(t+1)}$的分布

$$\boldsymbol{P}(\,\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)}, \dots, \boldsymbol{x}^{(1)})$$

  其中 $x^{(t+1)}$是词库$\mathcal{V} = \{w_1, \dots, w_{|V|}\}$中的任意一个词

- 完成这种任务的系统叫做**语言模型**

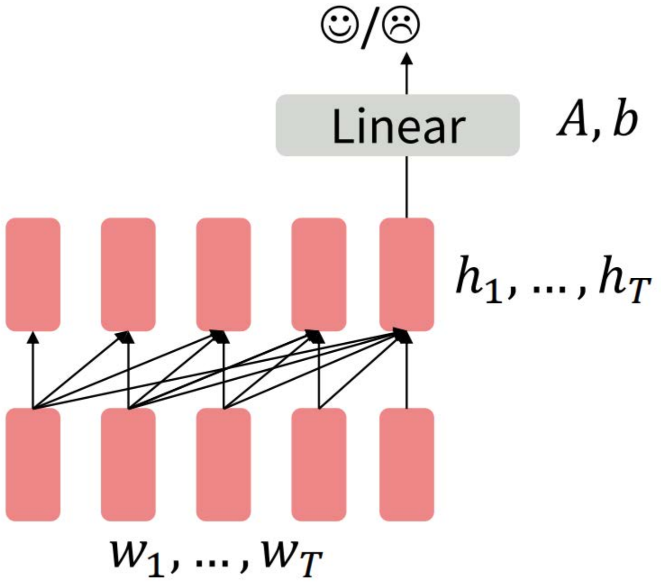
预训练解码器

- 在RNN章节中已经提到，以语言建模为目标预训练的解码器，可以用在各种序列生成任务上

- 如何解决其他任务呢？例如对整个句子进行某种分类

$$h_1, \ldots, h_T = \text{Decoder}(w_1, \ldots, w_T)$$
$$y \sim Ah_T + b$$

- 可以通过在最后一个词的隐藏状态上加上一个分类器来进行微调
- 其中A 和 b 是随机初始化的，并且由下游任务指定
- 梯度会在整个网络中反向传播



[Note how the linear layer hasn't been pretrained and must be learned from scratch.]

Generative Pretrained Transformer (GPT-1)

GPT-1针对特定任务的微调——如何设计**输入格式**?

以自然语言推理任务为例: 需要将句子对判断为蕴含/矛盾/或无关

Premise: *The man is in the doorway*

Hypothesis: *The person is near the door*

$\}$ **entailment**

以下是输入的大致格式

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

Token [DELIM]：用于分隔

Token [EXTRACT]：微调时，把[EXTRACT] token对应位置的特征表示接上线性分类器用于预测三类输出

GPT-1的结构，和针对更多种任务设计的输入格式

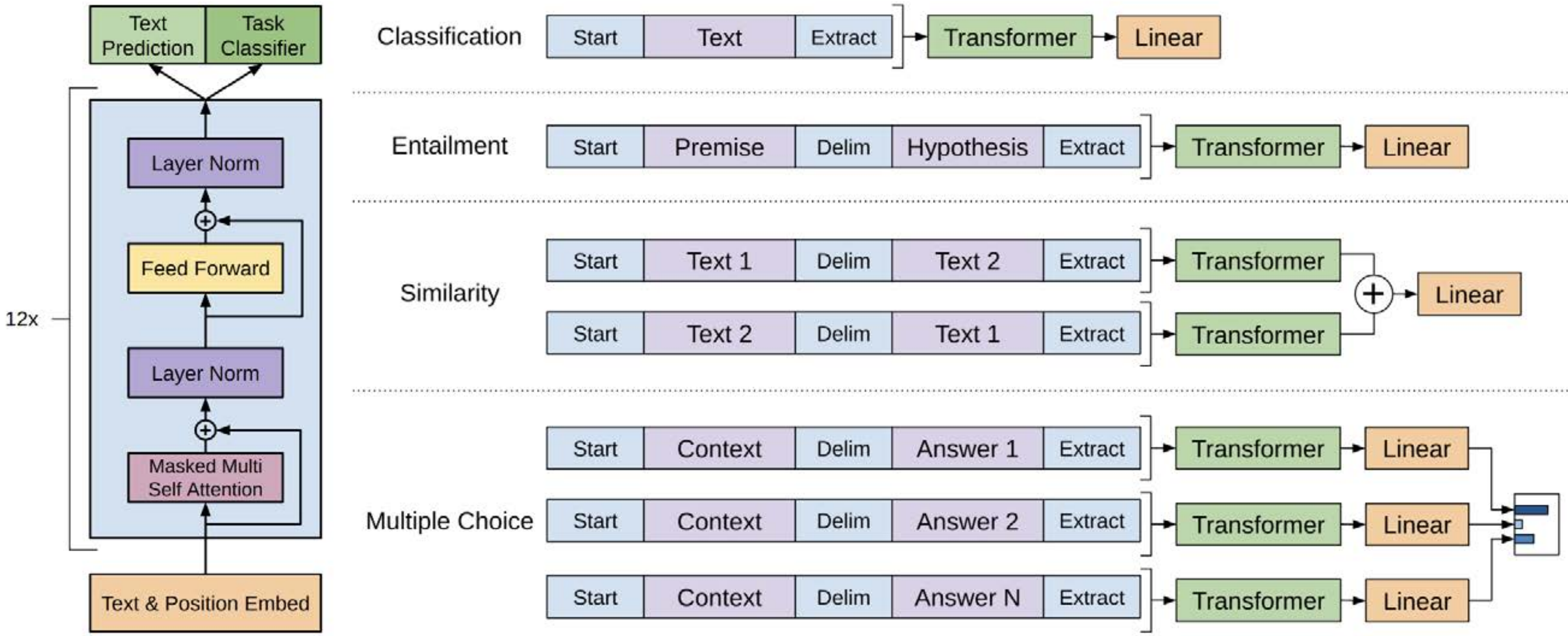


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

GPT-2: Language Models are Unsupervised Multitask Learners

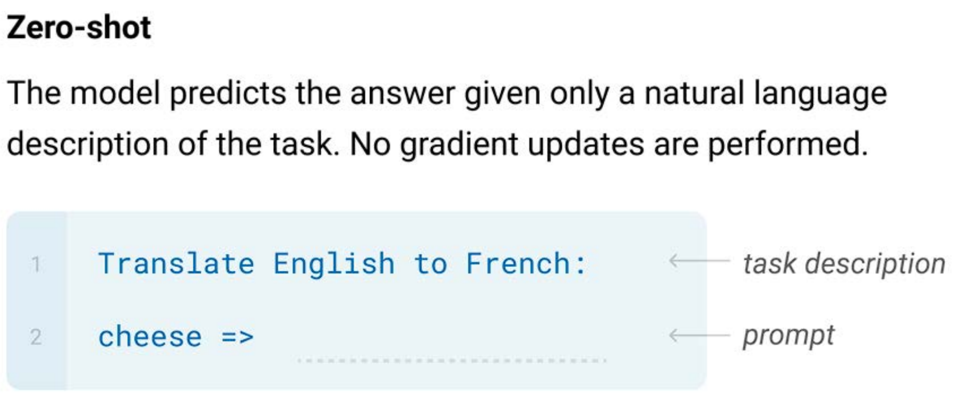
GPT-2 是 GPT 的更大版本，得到了相当大的性能提升

- 模型参数117M -> 1.5B参数，数据量约5GB -> 约40GB
- 模型结构并没有变化。
- ——大力出奇迹？并非如此简单

Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners[J]. OpenAI blog, 2019, 1(8): 9.

GPT-2: 如何完成零样本任务

- 在GPT-1中，为了针对特定任务微调，需要引入**特殊token**，例如[DELIM], [EXTRACT]
- 如果要进行零样本任务，则不能进行梯度更新，那么模型认识这些特殊token吗?
- GPT-2 通过**大规模无监督学习**来提高模型的泛化能力
- 引入了**提示(prompt)**的概念，完全以自然语言指令描述任务，不再为任务设计特殊token
- GPT-2 即使没有进行过特定任务的微调，也能够通过任务提示来理解并执行任务
- 因此，GPT-2的论文标题是 **Language Models are Unsupervised Multitask Learners**



零样本任务

GPT3: In-context Learning，超大模型

- 继续增大模型规模会怎么样？
- 到目前为止，我们介绍了使用预训练模型的两种办法：
  - 从模型确定的分布中采样 (可能提供零样本的prompt)
  - 针对我们关心的任务进行微调
- 研究者还发现：非常庞大的语言模型能够表现出从**上下文**中提供的**示例**进行学习的能力
- GPT-3论文标题： “ **Language Models are Few-Shot Learners** ”
- GPT-3同样没有改变模型结构，但参数量由GPT-2的1.5B增加至175B，数据量由40G增加至约45TB(>1000倍提高！)

Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877-1901.

GPT3: In-context Learning，超大模型

COMMON CRAWL

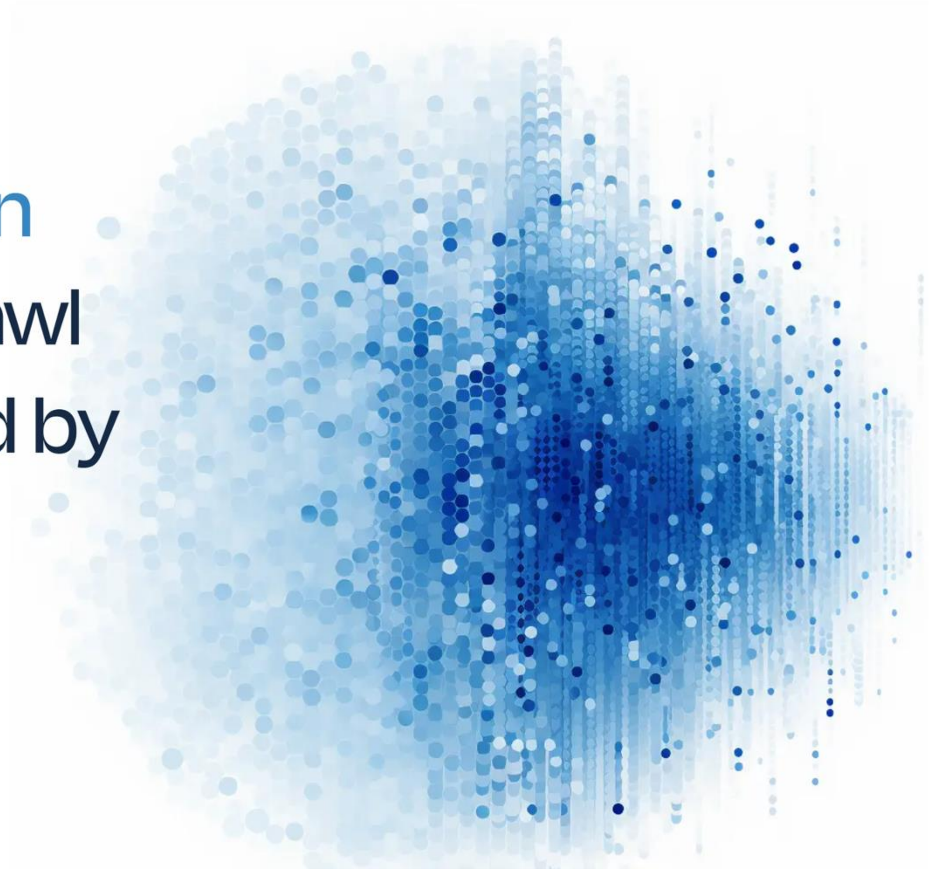
The Data | Resources | Community | About | Search | Contact Us

Common Crawl maintains a free, open repository of web crawl data that can be used by anyone.

Common Crawl is a 501(c)(3) non–profit founded in 2007.

We make wholesale extraction, transformation and analysis of open web data accessible to researchers.

Overview



Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877-1901.

GPT3: In-context Learning，超大模型

In-context Learning

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

`Translate English to French:`	*task description*
`cheese =>`	*prompt*

One-shot

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

`Translate English to French:`	*task description*
`sea otter => loutre de mer`	*example*
`cheese =>`	*prompt*

Few-shot

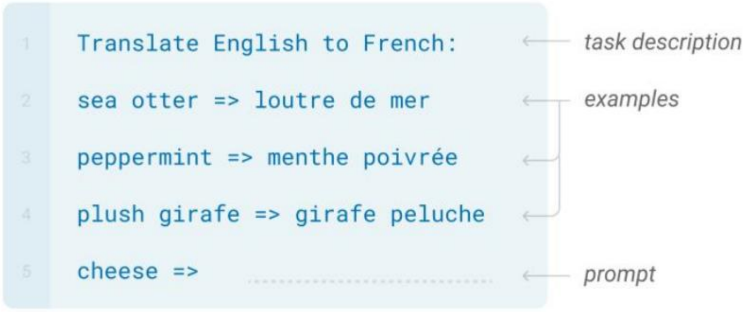
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

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

GPT3: In-context Learning，超大模型

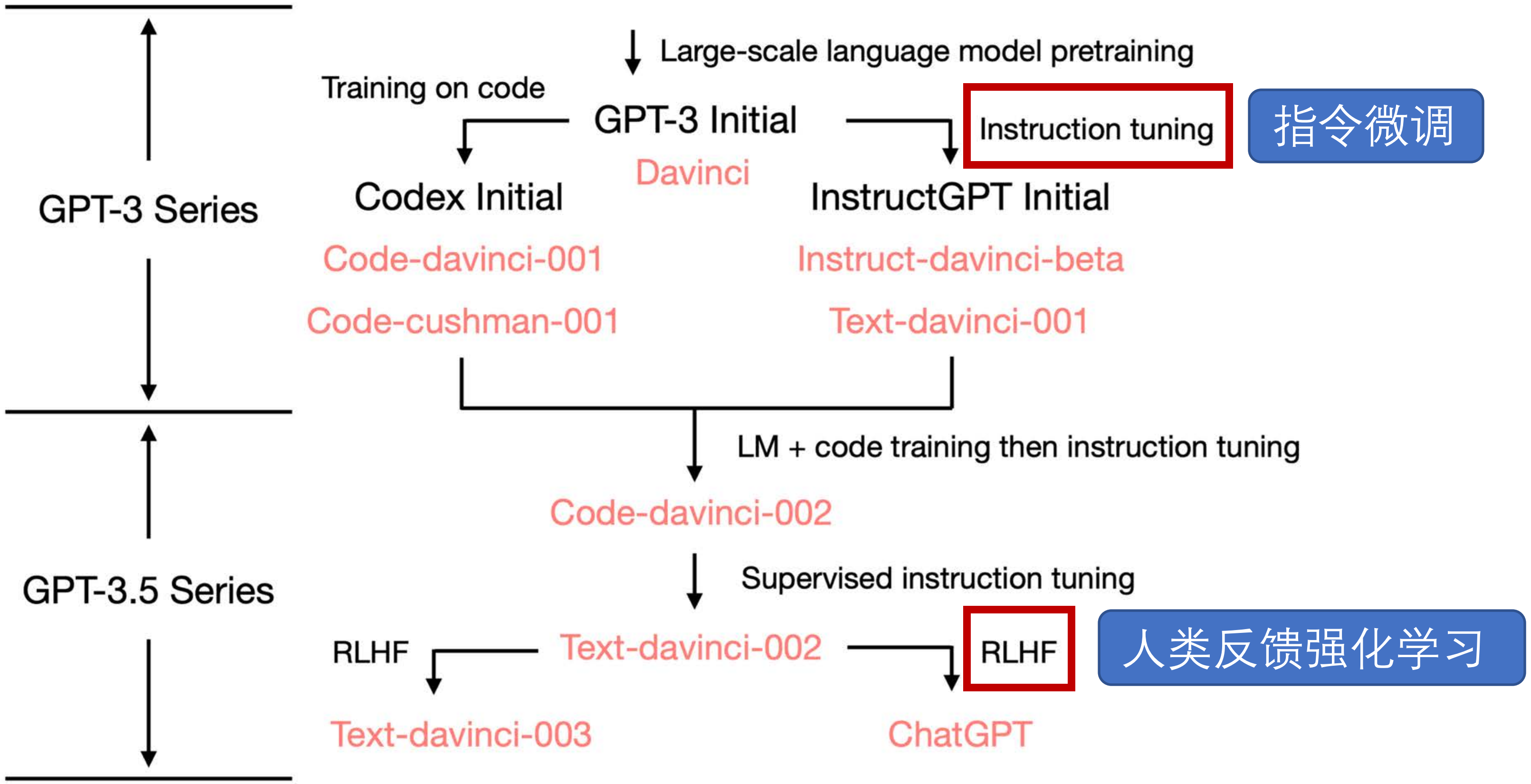
GPT-2 Style Few-shot Learning

GPT-3 Style Few-shot Learning



GPT家族：时间线

18年6月	19年2月	20年5月	20年5月	20年9月	21年11月	22年1月	22年11月28日	22年11月28日	23年3月14日	23年9月25日
GPT-1 在OpenAI博客发布	GPT-2 在OpenAI博客发布	GPT-3 论文发布在arXiv	GPT-3 API Beta 测试	GPT-3 授权给Microsoft	GPT-3 API 开放	InstructGPT发布（GPT-3.5)	GPT-3.5拓展到davinci-003	ChatGPT在OpenAI博客发布	GPT-4发布	GPT-4V发布



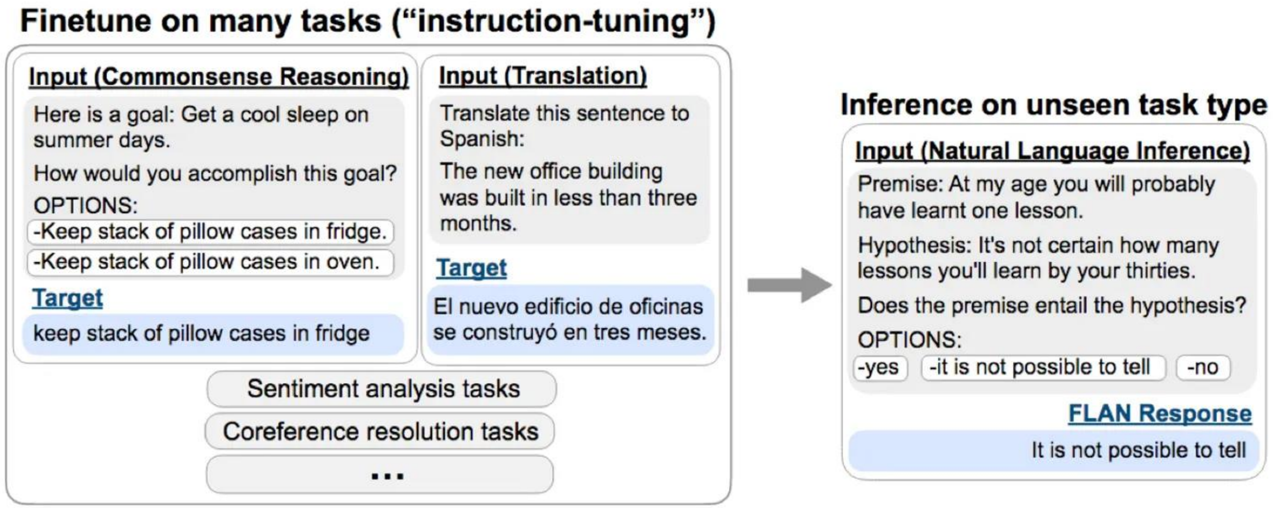
ChatGPT: 在GitHub所有代码上学习

- 在GitHub所有代码上学习，极大的增强的GPT模型的逻辑性和推理能力

InstructGPT: 指令微调技术

指令微调（Instruction tuning)

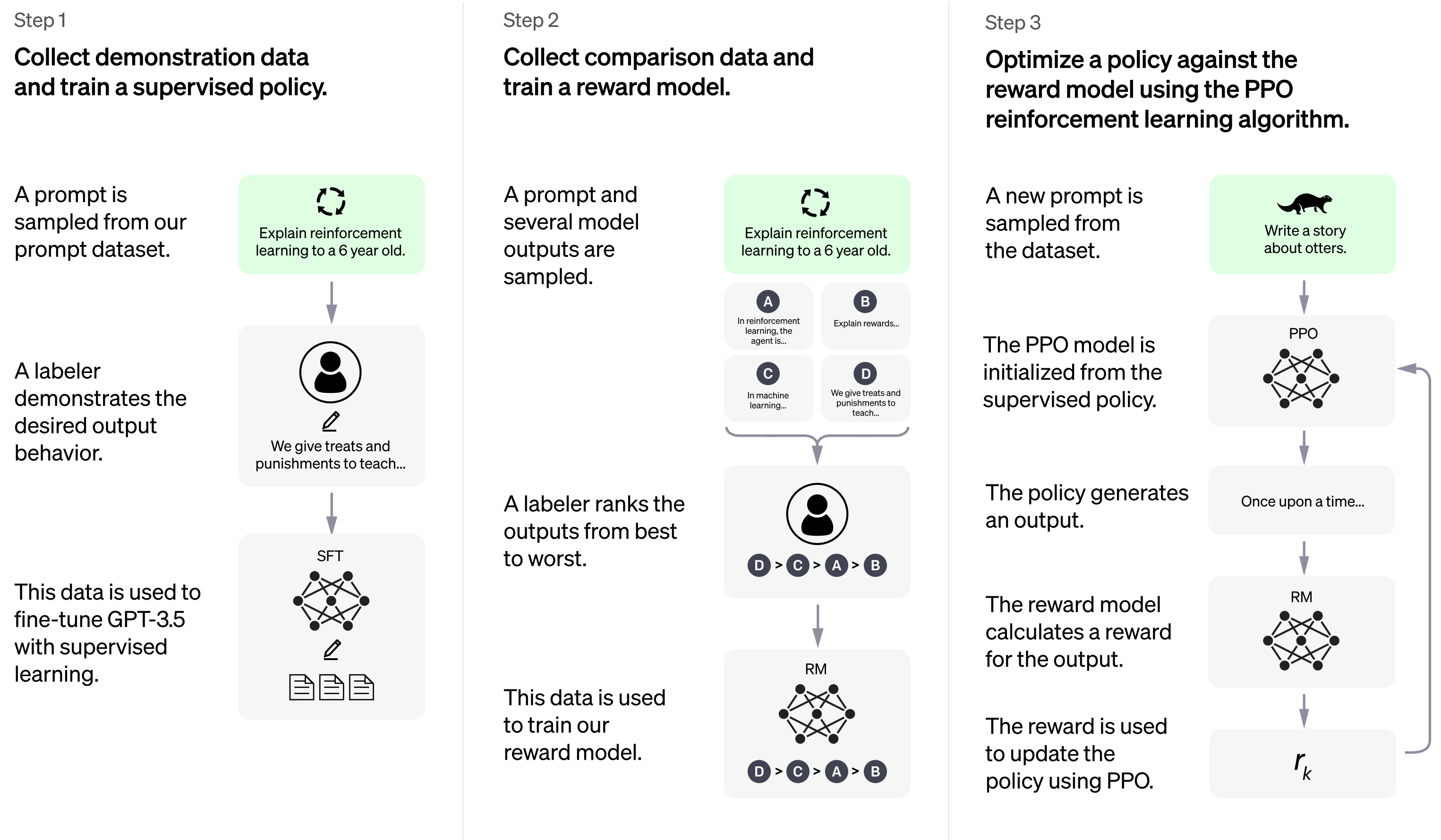
- 假设有很多NLP任务，研究人员针对每个NLP任务设计多种提示模板作为指令，并使用这些指令以及相应的数据对模型进行微调。
- 训练完成后，给大语言模型一个它从未见过的全新任务的指令，即让大语言解决零样本(zero-shot)任务。零样本上任务上的表现体现了模型的泛化能力



ChatGPT: RLHF技术

人类反馈强化学习（Reinforcement Learning from Human Feedback）

- 三个阶段：有监督微调 -> 奖励模型 -> 策略优化
- 主要特点：更面向人类用户真实需求。Prompt从大量用户提交的真实请求中抽样而来，而不是固定好研究任务的范围让研究人员给出任务描述



GPT-4V：最新进展

- 2023年9月25日，OpenAI宣布推出两项新功能，扩展用户与GPT-4的交互方式：

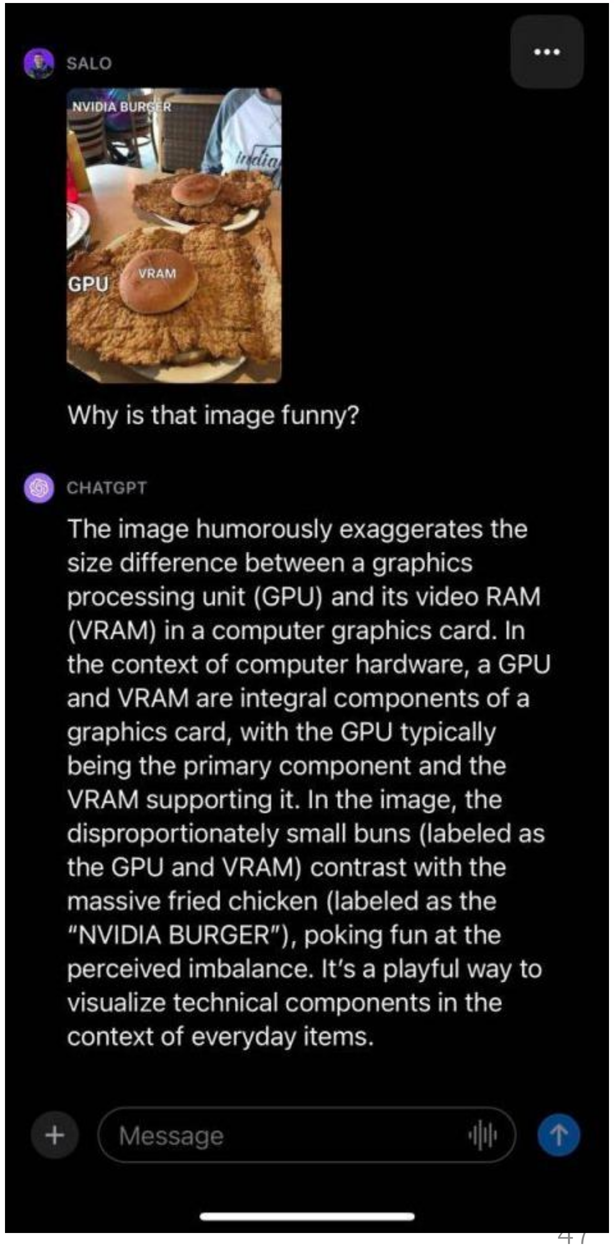
  支持语音和图像输入！

- 对应的多模态大模型被称为 **GPT-4V**

- 9月29日，微软发布了长达166页的GPT-4V测评报告：

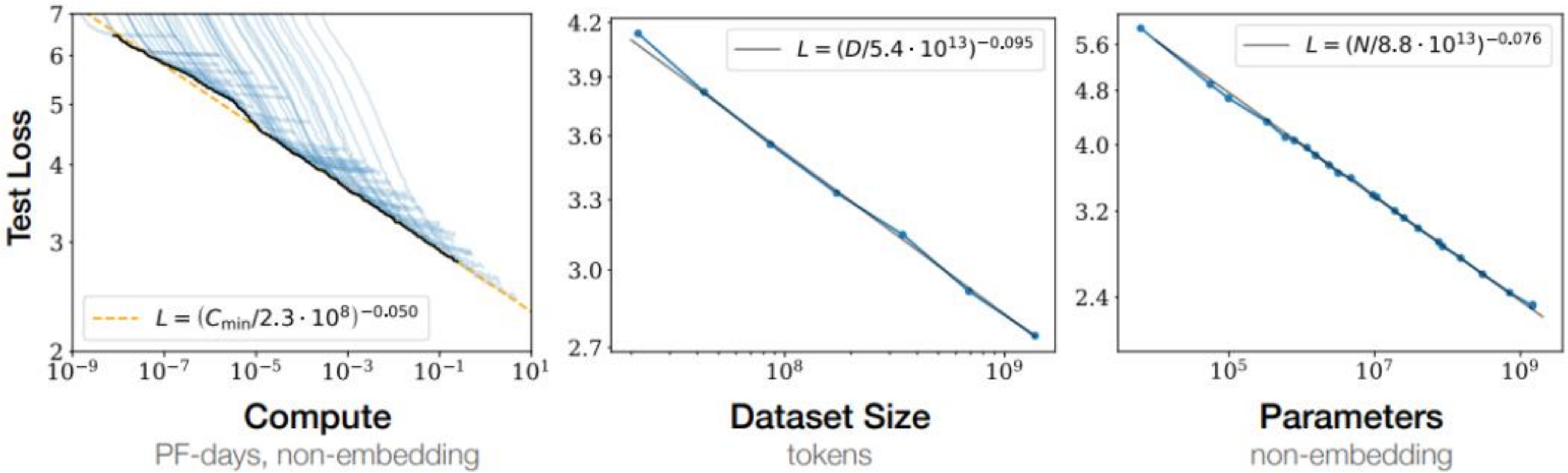
  The Dawn of LMMs:Preliminary Explorations with GPT-4V(ision)

https://arxiv.org/abs/2309.17421



缩放法则（Scaling Laws）

- **模型的性能强烈依赖于模型的规模**
- 包括：计算量、数据集大小和参数数量
- 最后的模型的效果（图中表现为 loss 值降低）会随着三者的指数增加而线性提高。



Kaplan J, McCandlish S, Henighan T, et al. Scaling laws for neural language models.

缩放法则（Scaling Laws）

缩放法则“失效”

- 在一些复杂任务上，当模型小于某一个规模时，模型的性能接近随机
- 当规模超过某个临界的阈值时，性能会显著提高到高于随机。
- 尤其是逻辑推理、数学推理或其他需要多步骤的复杂任务

能力涌现

涌现能力（Emergent Abilities）

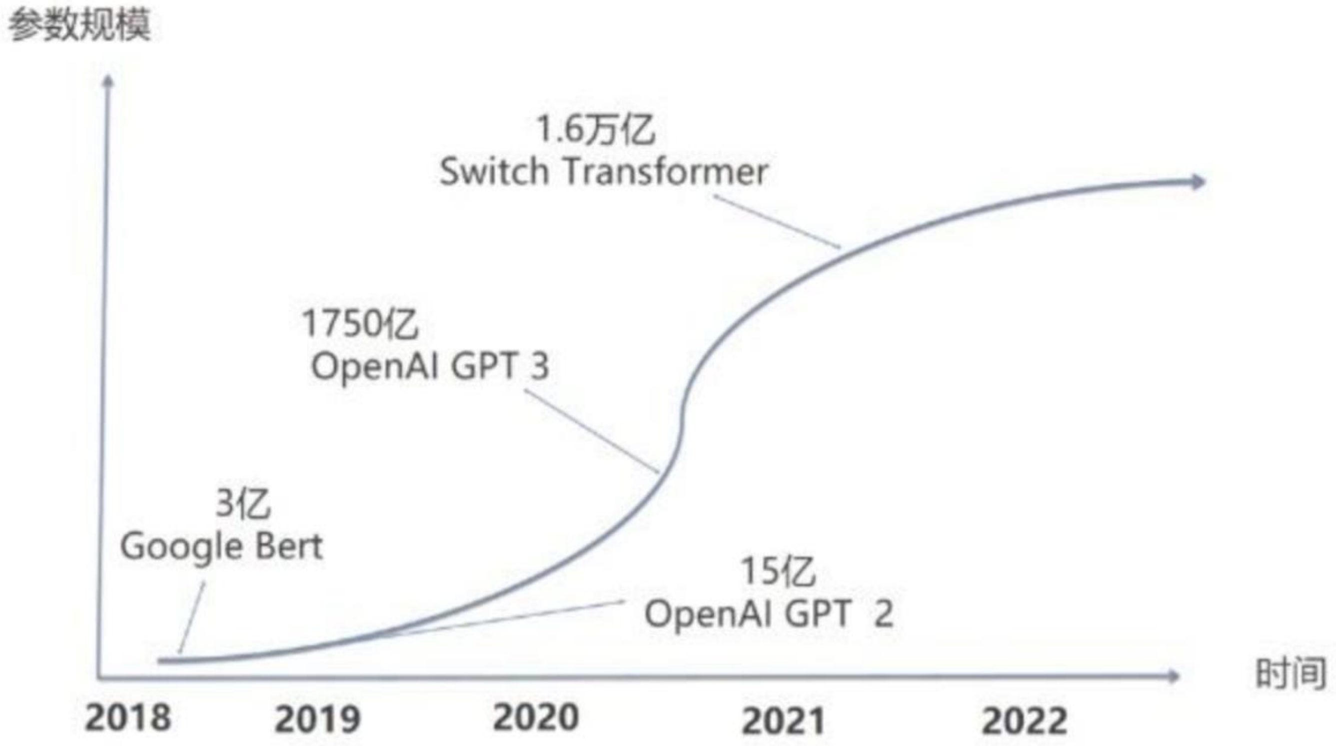
- 在宏观层面上展现出微观个体无法解释的特殊现象

“量变引起质变”

涌现能力（Emergent Abilities）

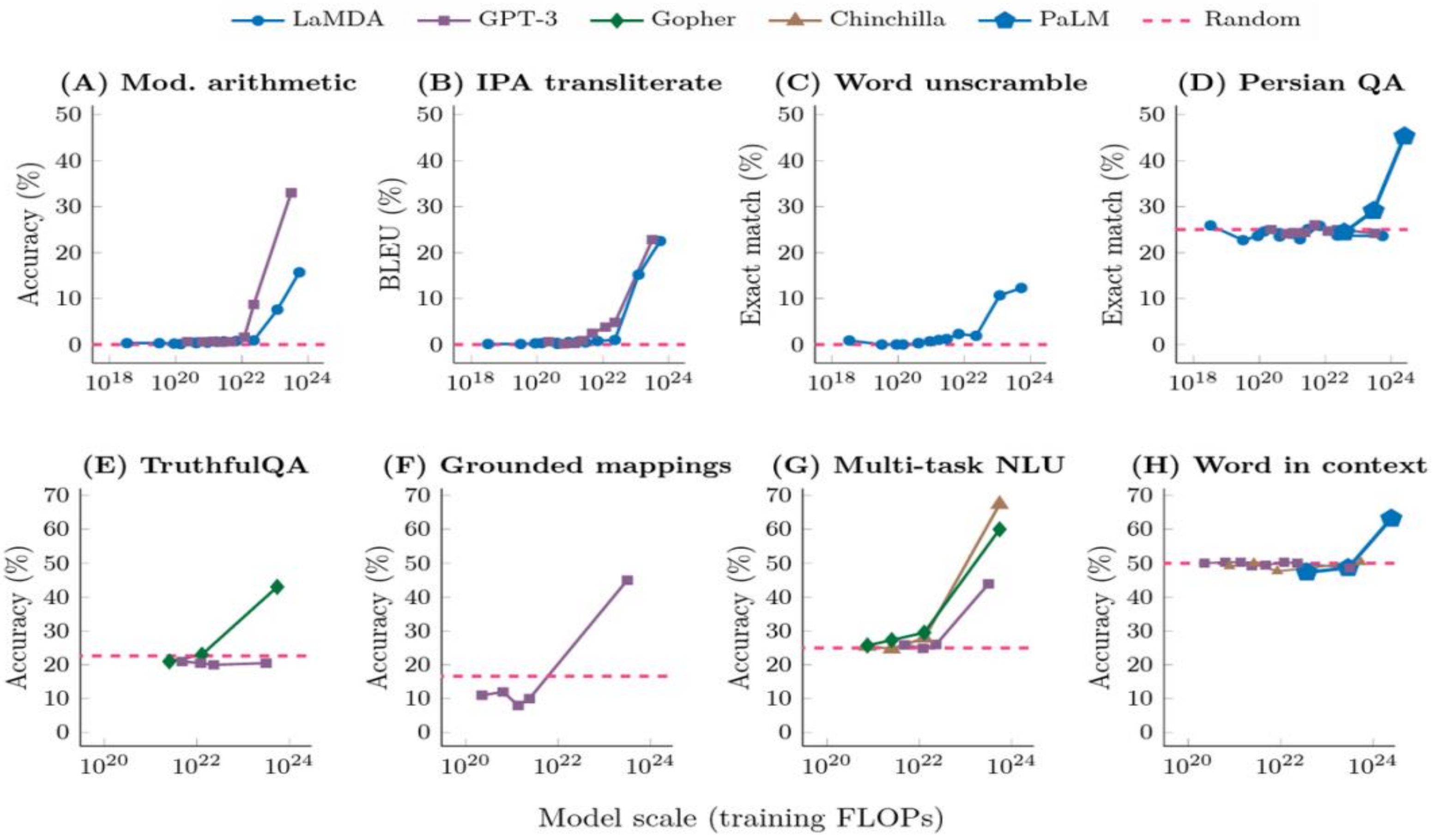
- 近年来，语言模型的参数规模快速增长

模型名称	参数规模（B）
GPT-3	175
GPT-3.5	175
LaMDA	130
Gopher	280
PaLM	540
PaLM-E	566
GPT-4	1800



涌现能力（Emergent Abilities）

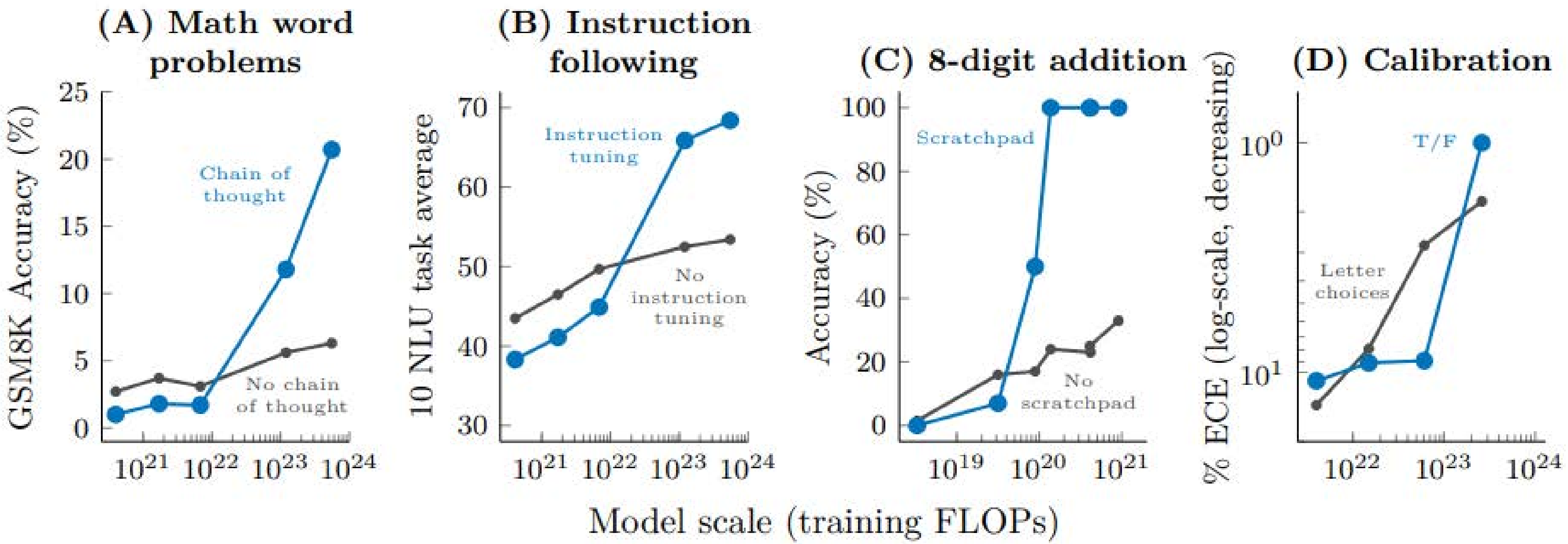
- 在小模型中没有表现出来，但是在大模型中表现出来的能力



Wei J, Tay Y, Bommasani R, et al. Emergent abilities of large language models.

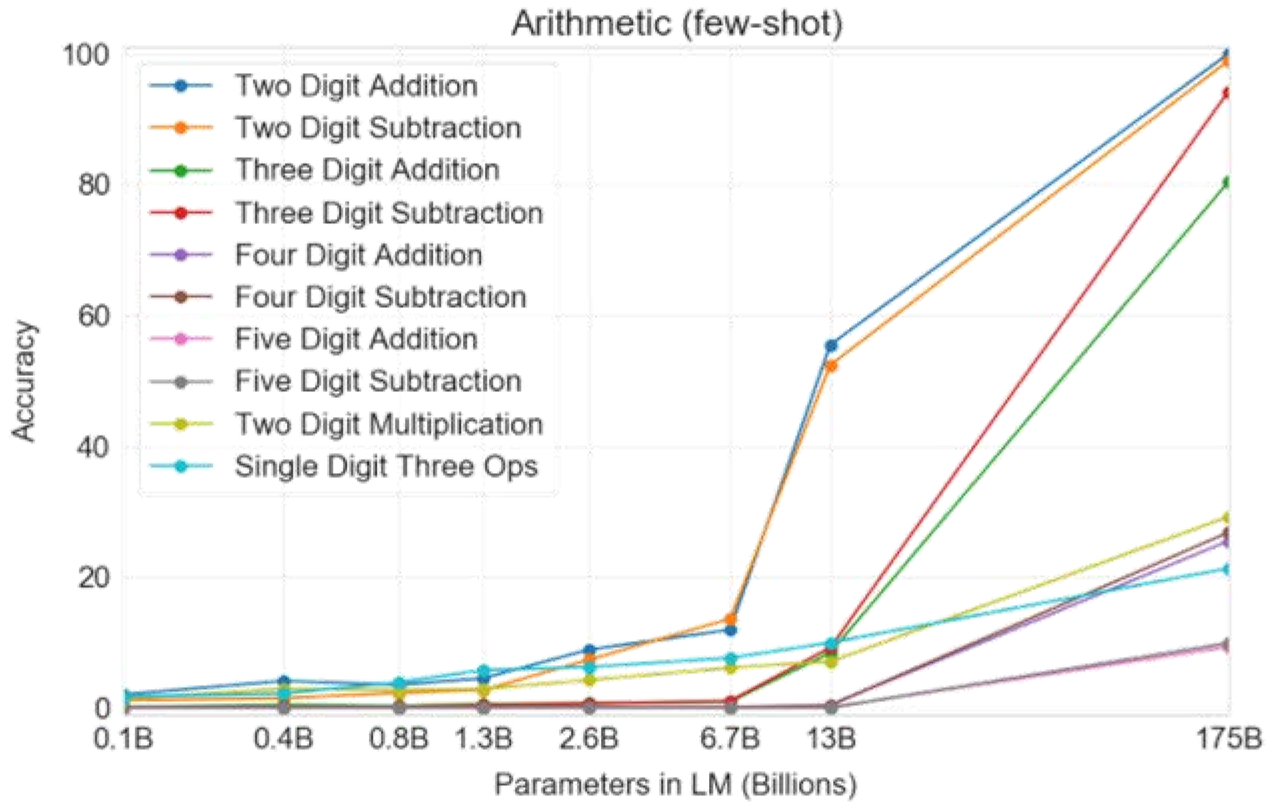
涌现能力（Emergent Abilities）

- 当模型规模在一定范围内时，模型的能力并没有随着模型规模的提升而提高。
- 但当模型超过一个临界值时，效果会马上提升。
- 这种提升和模型的结构如何并没有明显的关系。



涌现能力（Emergent Abilities）

- 图为不同规模的 GPT-3 模型在少样本学习设定下的所有 10 项算术任务的结果。
- 从第二大模型（GPT-3 13B）到最大模型（GPT-3 175B）的准确率有显著的跃升。



Brown, T. B. et. al. 2020. Language Models are Few-Shot Learners.

压缩即智能：为什么 ChatGPT 拥有智能?

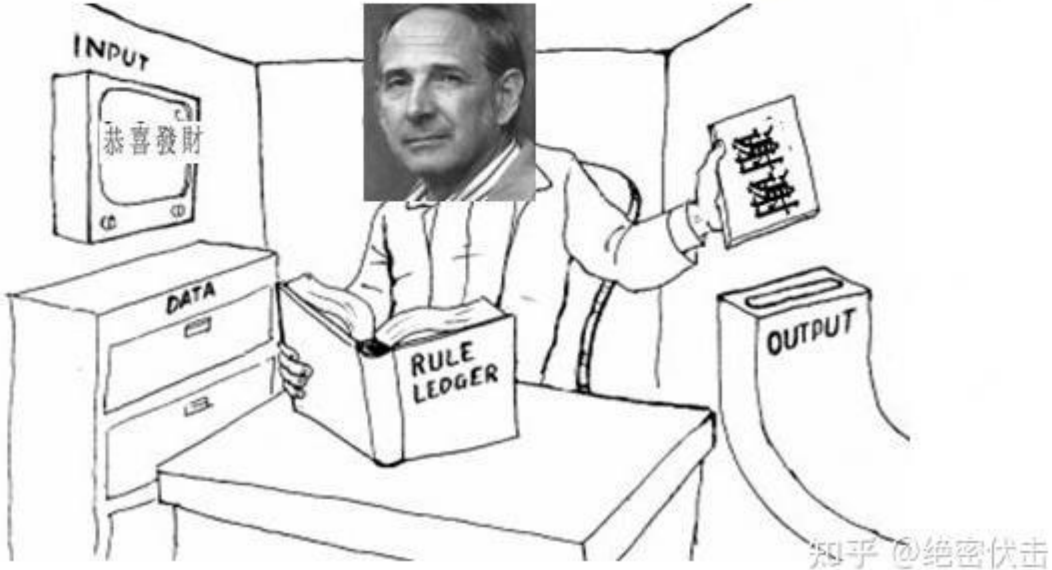
- AGI 基础模型的目标是实现对有效信息最大限度的无损压缩
- 压缩即智能
  - LLM = Compression （GPT 的 Next Token Prediction 本质上是对训练数据的无损压缩）

https://www.youtube.com/watch?v=dO4TPJkeaaU

压缩即智能：为什么 ChatGPT 拥有智能?

- 《中文房间》思想实验
  - 1980年，John Searle提出了一个著名的思想实验《中文房间》。实验过程可以表述如下：将一个对中文毫无了解，只会说英语的人关在一个只有一个小窗的封闭房间里。房间里有一本记录着中英文翻译的手册。房间里还有足够的稿纸、铅笔。同时，写着中文的纸片通过小窗 口被送入房间中。房间中的人可以使用他的书来翻译这些文字并用中文回复。 虽然他完全不会中文，但通过这个过程，房间里的人可以让任何房间外的人以 为他会说流利的中文。

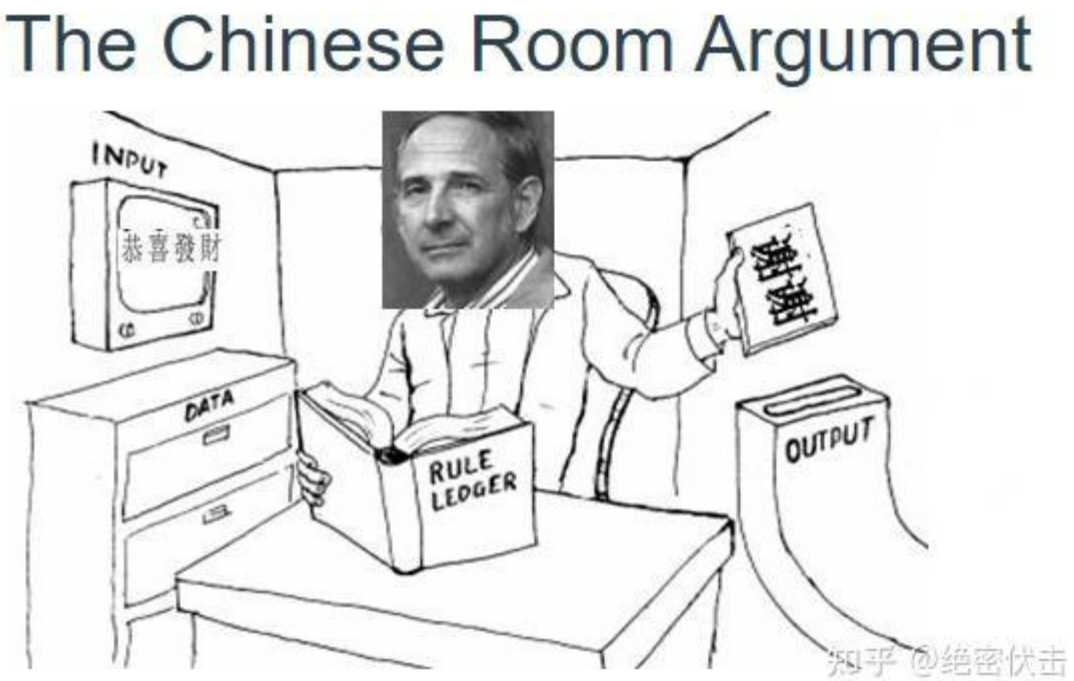
The Chinese Room Argument



https://www.youtube.com/watch?v=dO4TPJkeaaU

压缩即智能：为什么 ChatGPT 拥有智能?

- 《中文房间》思想实验
  - 如果我们能够从大量的数据中提取出一些语法和规则，那么手册可能会变得更加精简，但是系统的智能水平将会更高（泛化能力更强）。
  - 手册越厚，智能越弱；手册越薄，智能越强。就好像公司雇一个人好像能力越强的人，你需要解释得越少，能力越弱，你需要解释得越多。
  - GPT 的训练过程本质上就是对全世界语料的无损压缩



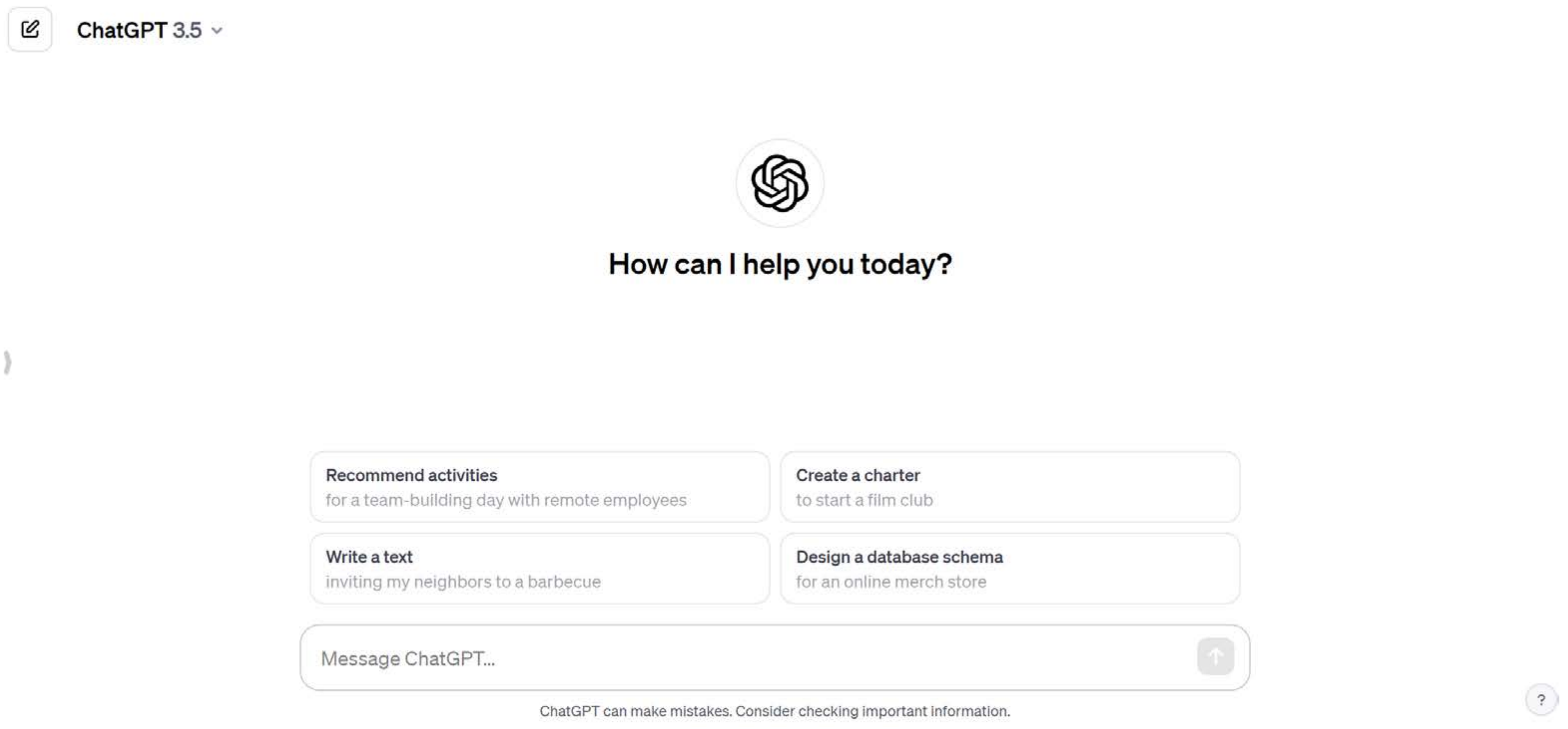
https://www.youtube.com/watch?v=dO4TPJkeaaU

课程目录

- 预训练语言模型
- **大语言模型的应用**

开始使用 ChatGPT

- 网址：https://chat.openai.com/



开始使用 ChatGPT

- **使用技巧 1**：在复杂问题上，用英文效果远比中文效果好。

- 英文提问

You

Please outline the steps for the Tower of Hanoi problem with 3 disks.

- 中文提问

You

请列出3个圆盘的汉诺塔问题的解法步骤。

开始使用 ChatGPT

- **使用技巧 1**：在复杂问题上，用英文效果远比中文效果好。

- 英文回答

1. **Initial Configuration:**
   - Rod A: 3 2 1
   - Rod B: (empty)
   - Rod C: (empty)
2. **Move Disk 1 from A to C:**
   - Rod A: 3 2
   - Rod B: (empty)
   - Rod C: 1
3. **Move Disk 2 from A to B:**
   - Rod A: 3
   - Rod B: 2
   - Rod C: 1
4. **Move Disk 1 from C to B:**
   - Rod A: 3
   - Rod B: 2 1
   - Rod C: (empty)
5. **Move Disk 3 from A to C:**
   - Rod A: (empty)
   - Rod B: 2 1
   - Rod C: 3
6. **Move Disk 1 from B to A:**
   - Rod A: 1
   - Rod B: 2
   - Rod C: 3
7. **Move Disk 2 from B to C:**
   - Rod A: 1
   - Rod B: (empty)
   - Rod C: 3 2
8. **Move Disk 1 from A to C:**
   - Rod A: (empty)
   - Rod B: (empty)
   - Rod C: 3 2 1

恰好 7 步，完美解决！

开始使用 ChatGPT

- **使用技巧 1**：在复杂问题上，用英文效果远比中文效果好。

- 中文回答

1. **将一个圆盘从柱子A移动到柱子C**:
   - 将编号为1的圆盘从柱子A移动到柱子C。
2. **将两个圆盘从柱子A移动到柱子B**:
   - 将编号为1的圆盘从柱子A移动到柱子B。
   - 将编号为2的圆盘从柱子A移动到柱子C。
   - 将编号为1的圆盘从柱子B移动到柱子C。
3. **将三个圆盘从柱子A移动到柱子C**:
   - 将编号为1的圆盘从柱子A移动到柱子B。
   - 将编号为2的圆盘从柱子A移动到柱子C。
   - 将编号为1的圆盘从柱子B移动到柱子C。
   - 将编号为3的圆盘从柱子A移动到柱子B。
   - 将编号为1的圆盘从柱子C移动到柱子A。
   - 将编号为2的圆盘从柱子C移动到柱子B。
   - 将编号为1的圆盘从柱子A移动到柱子B。
   - 将编号为3的圆盘从柱子B移动到柱子C。
   - 将编号为1的圆盘从柱子B移动到柱子A。
   - 将编号为2的圆盘从柱子B移动到柱子C。
   - 将编号为1的圆盘从柱子A移动到柱子B。
   - 将编号为3的圆盘从柱子C移动到柱子A。
   - 将编号为1的圆盘从柱子B移动到柱子C。
   - 将编号为2的圆盘从柱子B移动到柱子A。
   - 将编号为1的圆盘从柱子C移动到柱子A。
   - 将编号为3的圆盘从柱子A移动到柱子B。
   - 将编号为1的圆盘从柱子A移动到柱子C。

很多步仍未解决
陷入了混乱！

开始使用 ChatGPT

- **使用技巧 2**：多使用 Regenerate 功能，回答往往会更好。

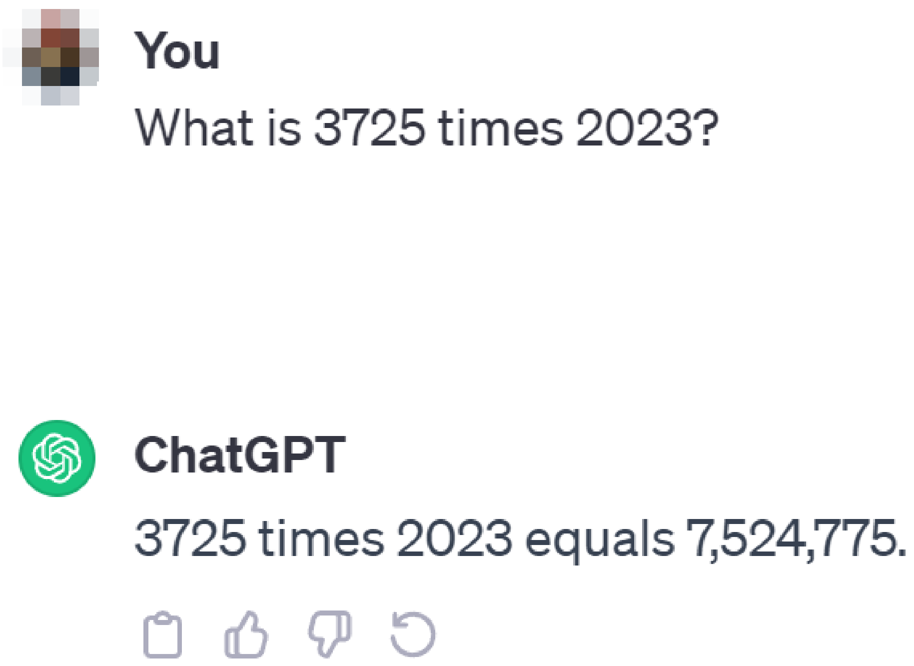
- 第二次中文回答

1. **将A柱上的2个盘子借助C柱移到B柱：**
   - 将A柱上的最大盘子（编号为3）移到C柱。
   - 将A柱上的中等盘子（编号为2）移到B柱。
   - 将C柱上的最大盘子（编号为3）移到B柱。
2. **将A柱上的最大盘子移到C柱：**
   - 将A柱上的最小盘子（编号为1）移到B柱。
   - 将A柱上的最大盘子（编号为3）移到C柱。
   - 将B柱上的最小盘子（编号为1）移到C柱。
3. **将B柱上的2个盘子借助A柱移到C柱：**
   - 将B柱上的中等盘子（编号为2）移到A柱。
   - 将B柱上的最小盘子（编号为1）移到C柱。
   - 将A柱上的中等盘子（编号为2）移到C柱。

解法仍然不正确
但逻辑正确了很多！

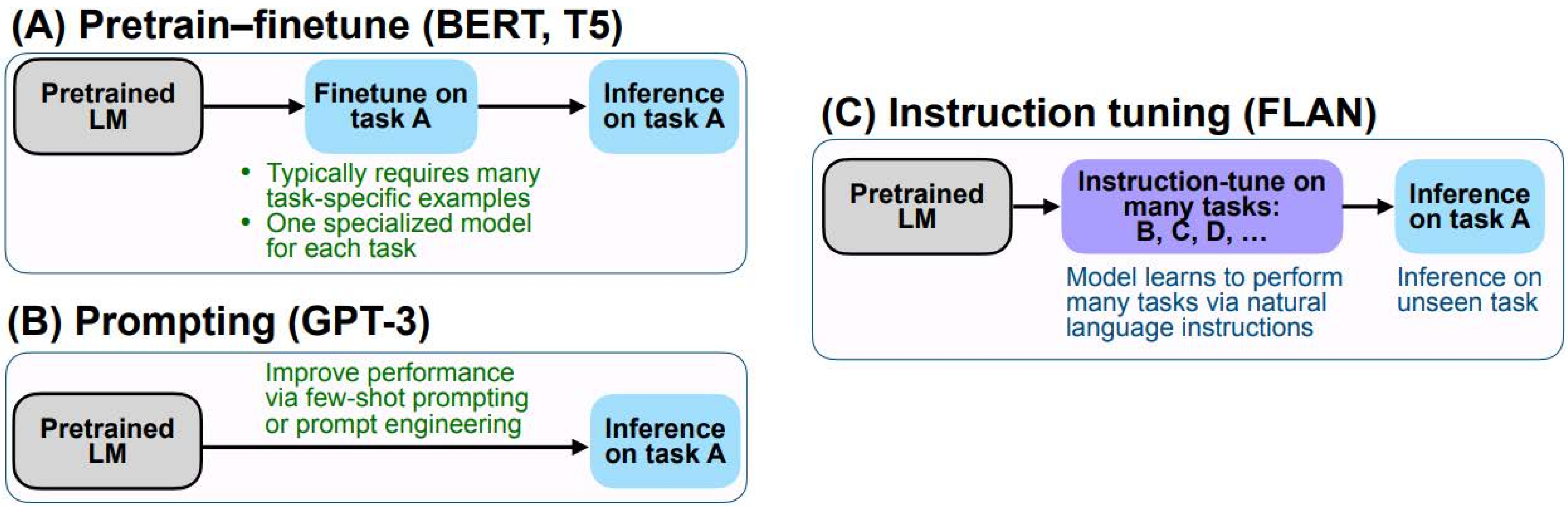
开始使用 ChatGPT

- **使用技巧 3**：大模型并不擅长数学计算，在必要的时候使用其他工具辅助它。



大语言模型的使用方式

- **提示（Prompting）**：无需训练，通过少量提示提升性能
  - 零样本提示（Zero-shot Prompting）/ 指令提示（Instruction Prompting）
  - 少样本提示（ Few-shot Prompting）
- **指令微调（Instruction tuning）**：通过语言指令让模型学习较多任务，从而完成未见过的任务
- **微调（Fine-tuning）**：需要尽可能多的任务相关的数据；每一个任务需要单独训练一个模型



提示工程（Prompt Engineering）

什么是提示工程（Prompt Engineering）？

- 从本质上讲，提示工程指精心设计的提示来引导人工智能模型，尤其是大型语言模型(LLM)，实现特定的输出。
- 类似于家长通过问题来教育孩子，利用措辞巧妙的问题引导孩子思考的过程

提示工程（Prompt Engineering）的定义和核心概念

- 提示工程是设计和完善提示（问题或指令）以引发人工智能模型特定响应的实践，可以将其视为人类意图和机器输出之间的接口。
- 和 Siri 等语音助手互动，就像是提示工程的基本形式。不同表达请求的方式会产生截然不同的结果：

请介绍一些著名的篮球运动员

请介绍一下科比·布莱恩特

零样本提示（Zero-Shot Prompting)

零样本提示（Zero-shot Prompting）：直接将任务描述输入模型

Zero-shot learning is to simply feed the task text to the model and ask for results.

(All the sentiment analysis examples are from SST-2)

```
Text: i'll bet the video game is a lot more fun than the film.
Sentiment:
```

早期的Zero shot prompting，实际上在不知道如何表达一个任务的情况下，反复更换提示词尝试好的任务表达方式，这种做法已经被证明是在**拟合训练数据的分布**。

零样本提示（Zero-Shot Prompting）

GPT-4V示例：

Sec. 4.1 Image Description on Diverse Domains

Scene Understanding

Prompt:
Describe the image

GPT-4V:
The image is a screenshot from a dashcam, taken from the inside of a car, showing the view of the road ahead. The car is driving on a two-lane road, with a speed limit sign of 35 mph visible on the right side of the road. There are trees on both sides of the road, and the sky is filled with dark clouds, suggesting that it might rain soon. Orange cones are visible on the left side of the road, indicating a construction zone. In the distance, there is a bridge overpass.

The Dawn of LMMs: Preliminary Explorations with GPT-4V(ision)

Sec. 4.1 Image Description on Diverse Domains

In-the-Wild Logo Recognition and Description

Prompt:
Describe both the image and logo in details

GPT-4V:
The image is of a McDonald's french fry container, filled with golden brown french fries. The container is red, and has the iconic McDonald's "M" logo on it. The background is a bright yellow, making the red container and fries stand out.
The logo is the McDonald's "M" which is a large golden arch, forming the letter "M" for McDonald's. It is one of the most recognized logos in the world, and is often associated with the fast food chain.

Prompt:
Describe both the image and logo in details

GPT-4V:
The image shows two Starbucks cups, one is a clear plastic cup with a lid and straw, and the other is a white paper cup with a lid. The clear plastic cup contains a cold drink with ice and whipped cream on top, while the white paper cup contains a hot drink. The cups are placed on a wooden table with a blurred background of a Starbucks store.
The logo on the cups is the Starbucks logo, which is a green circle with a white mermaid or siren in the center

Prompt:
Describe both the image and logo in details

GPT-4V:
The image shows a woman in traditional Japanese clothing standing in front of the Olympic rings logo. The woman is wearing a purple kimono with a floral pattern and a red obi belt. She is smiling and posing for the photo.
The Olympic rings logo is made up of five interlocking rings of different colors: blue, yellow, black, green, and red. The rings are arranged in a way that they form a flower-like shape. The logo is displayed on a grassy area with trees in the background.

Prompt:
Describe both the image and logo in details

GPT-4V:
The image shows two Coca-Cola bottles lying on a bed of ice cubes. The bottles are wet, indicating that they are chilled and ready to drink. The bottle on the left is a new grip bottle, as mentioned on the label. The bottle on the right is a regular Coca-Cola bottle. Both bottles have the iconic Coca-Cola logo on them.
The logo is written in a cursive font in white color, with a red background. The first "C" in Coca is larger than the rest of the letters, and the "o" in Cola is elongated. The logo is instantly recognizable and is synonymous with the brand.

零样本提示（Zero-Shot Prompting）

GPT-4V示例：

Sec. 4.3 Multimodal Knowledge and Commonsense

Multimodal Commonsense

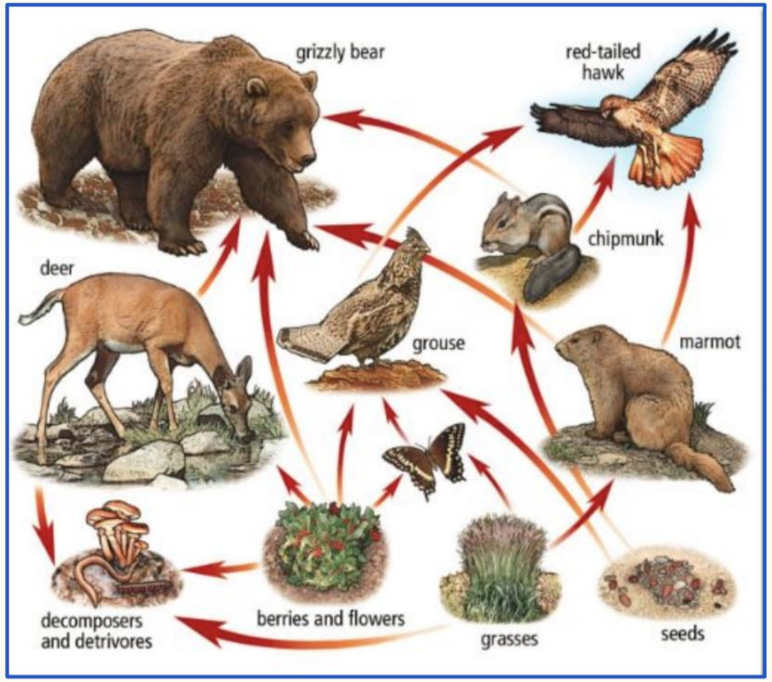
Prompt:
What is [person3] doing?

GPT-4V:
[person3] is carrying a plate of food, likely serving it to the customers at the table. It appears that they are working as a waiter or server in a restaurant.

Prompt:
What are [person1] and [person2] doing?

GPT-4V:
[person1] and [person2] are walking down the aisle at a wedding ceremony.

Prompt:
Which of the organisms is the producer in this food web?



GPT-4V:
The producers in this food web are the berries and flowers, grasses, and seeds. These are the organisms that use energy from the sun to produce their own food through photosynthesis.

零样本提示（Zero-Shot Prompting）

GPT-4V示例:

零样本提示（Zero-Shot Prompting）

GPT-4V示例：

Why is this picture funny?

This picture appears to be taken in a light-hearted moment showing a man (who, based on his appearance and the reactions of those around him, is likely someone of significance) playfully posing as if he's being measured for his height against a height chart. The humor in the photo likely arises from a few elements:

1. **Juxtaposition**: The situation is a juxtaposition of a formal setting, where individuals are dressed in suits and seem to be engaged in serious business, against a casual, almost childlike activity of measuring one's height.
2. **Expressions**: The smiling and laughing expressions of the individuals in the picture suggest they are sharing in a joke or a moment of levity.
3. **Context**: If the individual being measured is indeed a high-profile figure, the whimsy of participating in such an ordinary activity might contrast with the usual gravitas expected from someone in their position.
4. **Incongruity**: The height chart is a tool typically used in schools or at medical check-ups for children, so an adult, particularly a dignified one, engaging in this activity is an unexpected and humorous use of the chart.
5. **Surprise**: The moment captured seems spontaneous, and the man's playful stance, along with the reactions of those around him, indicates that this is a break from what might have been the routine or expected behavior, adding an element of surprise which often contributes to humor.

It's the combination of these elements—unexpected behavior, contrasting contexts, and shared enjoyment—that gives this picture a humorous quality.

GPT-4

Y

what's funny in this image

The humor in this image seems to come from the playful interaction between the two men at the scale. The man on the scale is looking down at it with a somewhat serious expression, holding a clipboard that suggests he might be recording his weight. Meanwhile, the other man, who appears to be in a leadership role, is playfully stepping onto the scale from behind, which would add his weight to the reading, causing the numbers to be higher than expected. This playful gesture seems to be causing laughter among the observers, adding a light-hearted moment to what might otherwise be a routine or mundane activity.

指令提示（Instruction Prompting ）

指令提示（Instruction Prompting）：同样不为模型提供样本，但具体做法有所不同

- 零样本提示：给出描述性的提示或问题

```
"""
Text: I'll bet the video game is a lot more fun than the film.
Sentiment:
"""
```

- **指令提示**：提供明确的指令或步骤，以执行特定的任务
  - 直接的指令， 告诉模型该做什么
  - 可以包括格式化的输出示例

```
"""
Please label the sentiment towards the movie of the given movie review. The sentiment label
should be "positive" or "negative".
"""
```

指令提示（Instruction Prompting ）

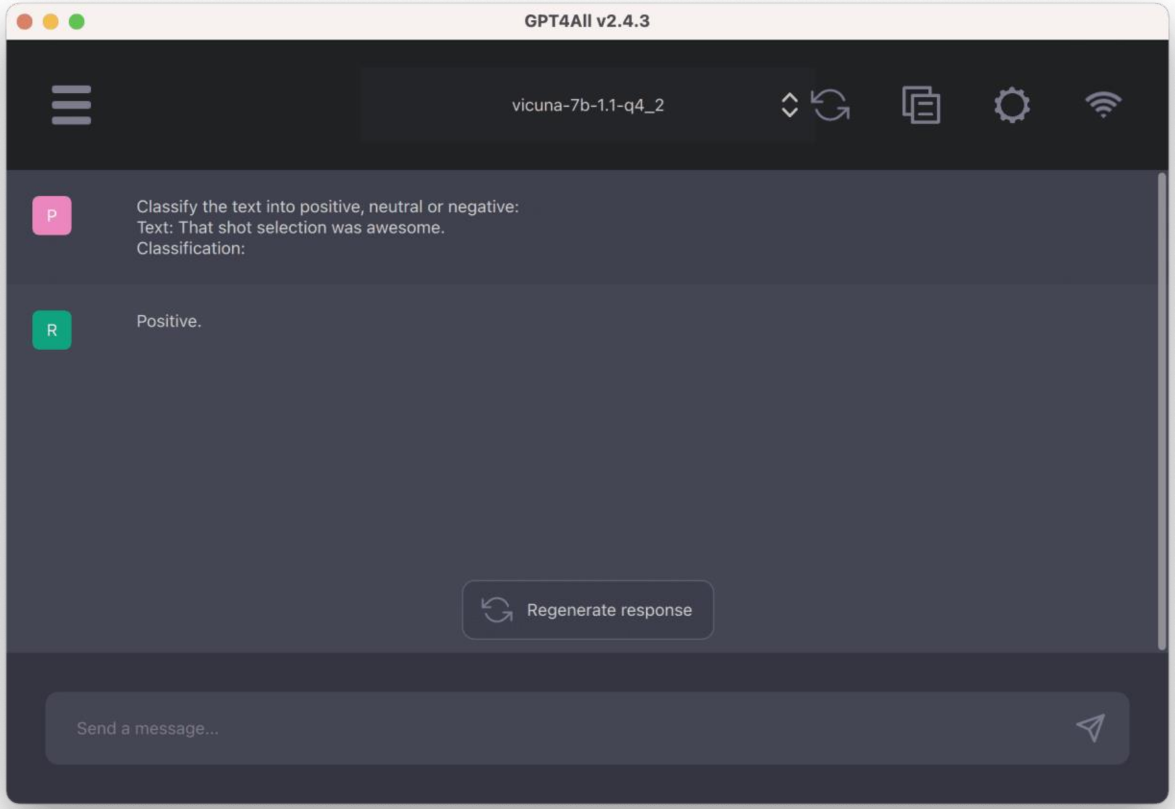
零样本提示和指令提示使得模型能够在未经特定任务训练的情况下理解并执行新的任务类型。这种灵活性是大型语言模型全能性的关键体现。

提示：

```
1 Classify the text into positive, neutral or negative:
2 Text: That shot selection was awesome.
3 Classification:
```

输出：

```
1 positive
```



指令提示（Instruction Prompting ）

GPT-4V示例:

Prompt:

Please read the text in this image and return the information in the following JSON format (note xxx is placeholder, if the information is not available in the image, put "N/A" instead).
{"Surname": xxx, "Given Name": xxx, "USCIS #": xxx, "Category": xxx, "Country of Birth": xxx, "Date of Birth": xxx, "SEX": xxx, "Card Expires": xxx, "Resident Since": xxx}

GPT-4V:

{"Surname": "STEVENS", "Given Name": "ICHIKA", "USCIS #": "000-000-000", "Category": "IR1", "Country of Birth": "Japan", "Date of Birth": "18 AUG 1988", "SEX": "F", "Card Expires": "07/04/2027", "Resident Since": "07/05/2017"}

指令提示（Instruction Prompting ）

GPT-4V示例：

Prompt:
Count the number of apples in the image. First count how many rows of apples are there, and then count the apples in each row, and finally sum them up to get the total number.

Note: Same query image as shown above, resized to save space

GPT-4V:
There are 3 rows of apples in the image.
The top row has 5 apples.
The middle row has 4 apples.
The bottom row has 6 apples.
Therefore, the total number of apples in the image is 5 + 4 + 6 = 15 apples.

Prompt:
You are an expert in counting things in the image. Let's count the number of apples in the image below row by row to be sure we have the right answer.

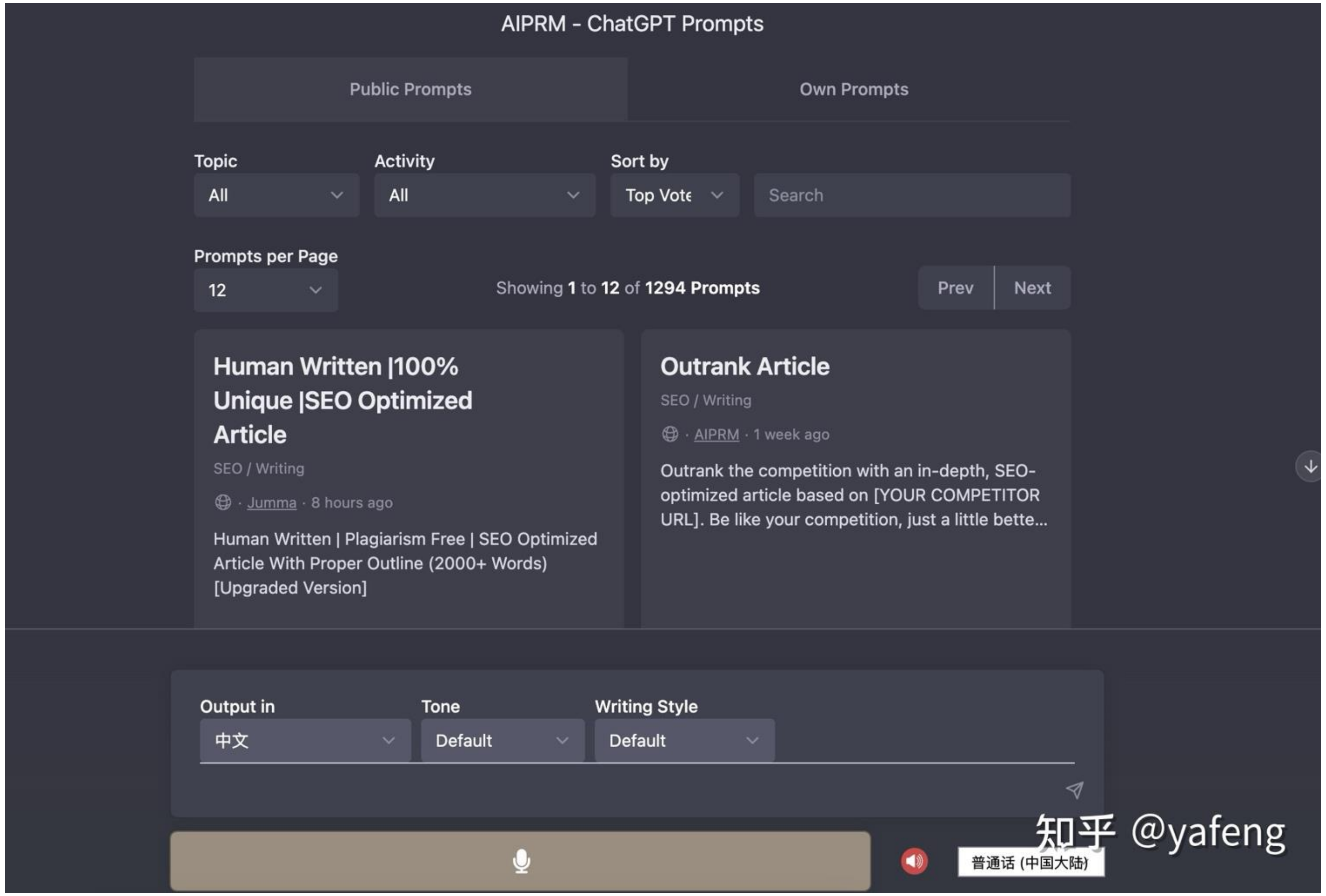
Note: Same query image as shown above, resized to save space

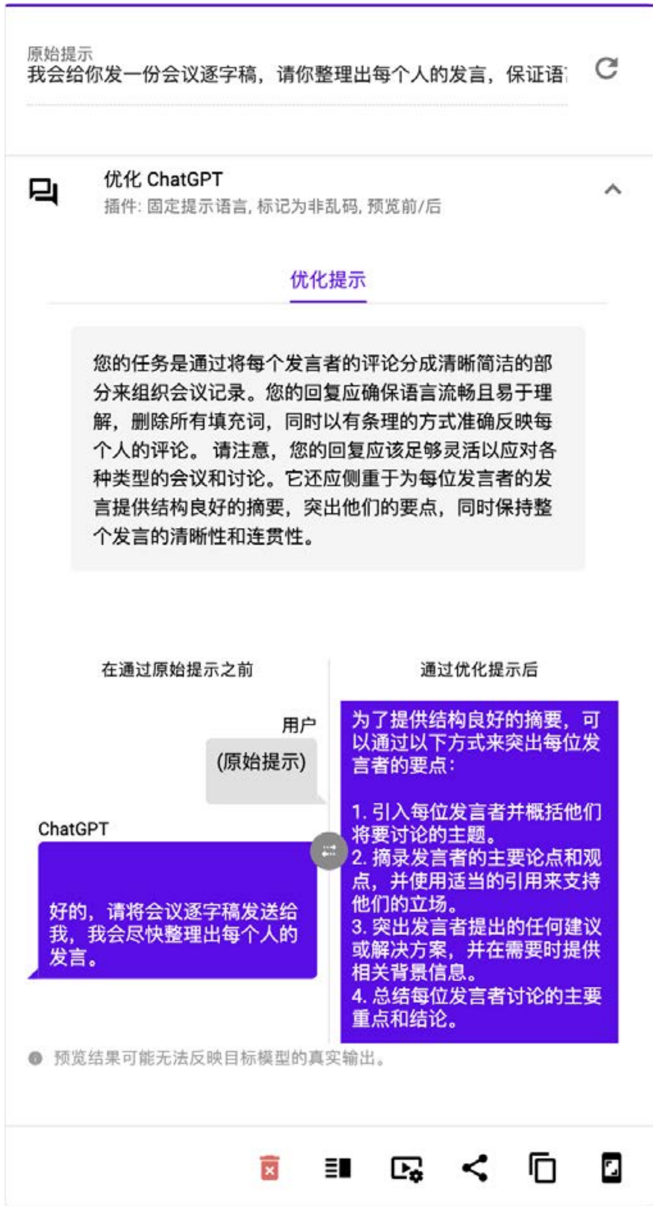
GPT-4V:
Row 1: 4 apples
Row 2: 4 apples
Row 3: 3 apples
Total number of apples in the image: 11 apples

指令提示（Instruction Prompting ）

如何学好指令提示（Instruction Prompting）？

利用一些辅助工具/商业化产品：AIPRM



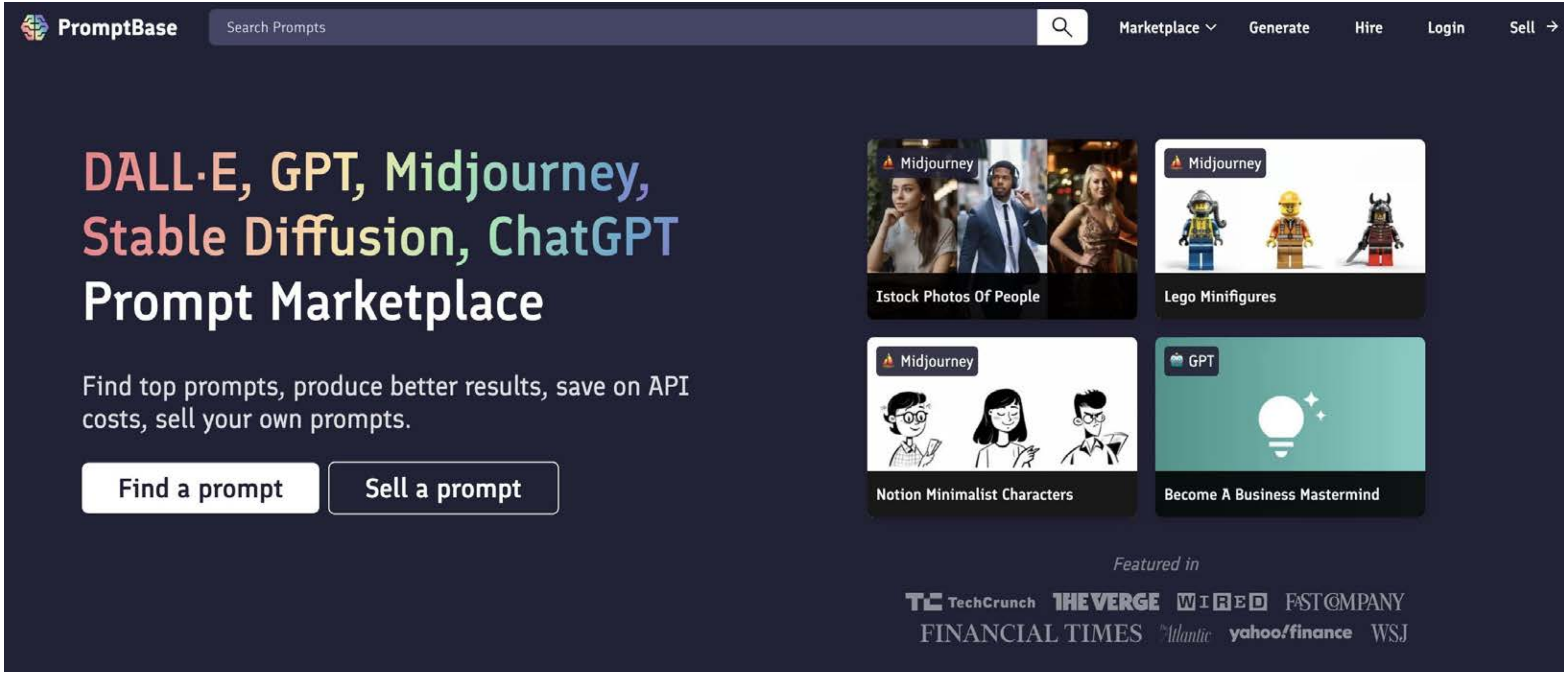


https://zhuanlan.zhihu.com/p/611365958

指令提示（Instruction Prompting ）

如何学好指令提示（Instruction Prompting）？

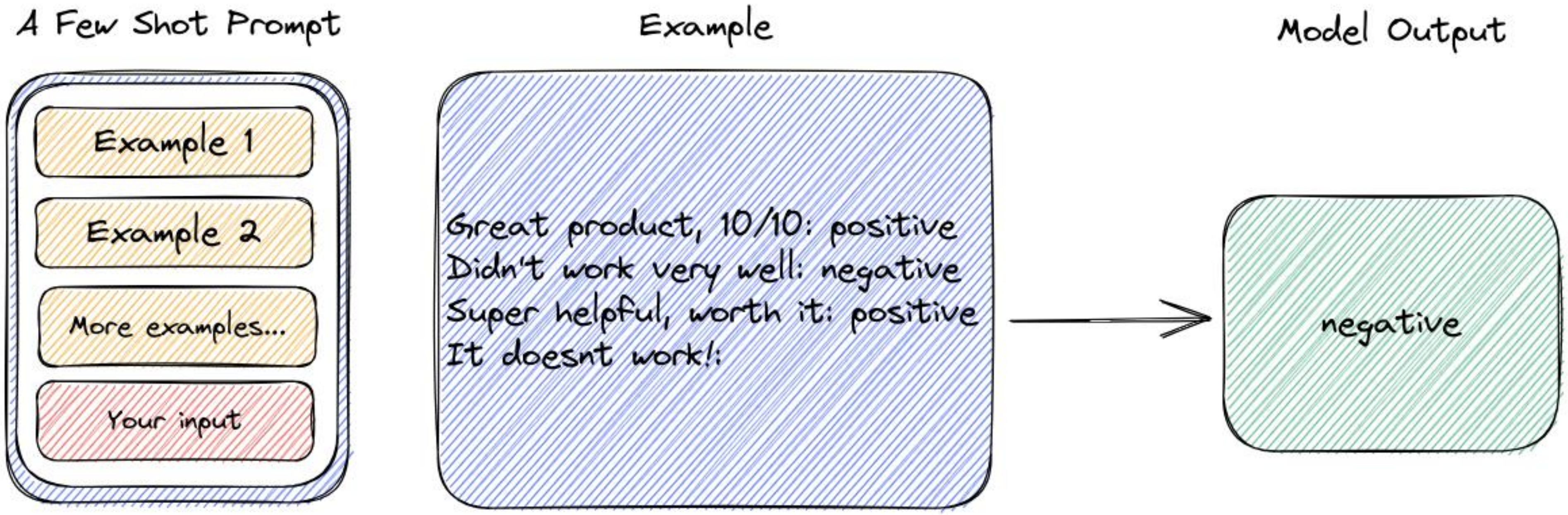
利用一些辅助工具/商业化产品



https://promptbase.com/

少样本提示（Few-Shot Prompting）

少样本提示（Few-shot Prompting）：也可以叫做上下文学习（In-Context Learning, ICL），即给模型几个示例作为范本，然后让LLM解决新问题。



少样本提示（Few-Shot Prompting）

示例:

```
Text: (lawrence bounces) all over the stage, dancing, running, sweating, mopping his face and generally
Sentiment: positive

Text: despite all evidence to the contrary, this clunker has somehow managed to pose as an actual featur
Sentiment: negative

Text: for the first time in years, de niro digs deep emotionally, perhaps because he's been stirred by t
Sentiment: positive

Text: i'll bet the video game is a lot more fun than the film.
Sentiment:
```

少样本提示（Few-Shot Prompting）

GPT-4V示例:

Sec. 3.4 In-context Few-shot Learning

Zero-shot

Prompt:
What is the read of the speed meter?

GPT-4V:
The read of the speed meter is approximately 22 miles per hour (mph).

Zero-shot (think step-by-step)

Prompt:
What is the read of the speed meter? Let's think step-by-step.

Note: Same query image as shown above, resized to save space

GPT-4V:
There are two scales on this speedometer - one in miles per hour (mph) and one in kilometers per hour (km/h). The needle is pointing between the 20 and 40 on the mph scale, and between the 40 and 60 on the km/h scale. Since the numbers on the scales increase in increments of 20, we can estimate that the needle is pointing at approximately 30 mph and 50 km/h.
Therefore, the read of the speed meter is approximately 30 mph (or 50 km/h).

Sec. 3.4 In-context Few-shot Learning

In-context Few-shot (2-shot)

Prompt:
What is the read of the speed meter?

The yellow pointer is roughly at the middle between 80 and 100 mph. The middle between 80 and 100 is 90. The speed is around 91 mph, as the pointer is just passing 90 mph.

The yellow needle is roughly at the middle between 20 and 40 mph. The middle between 20 and 40 is 30. The speed is around 29 mph, as the needle is close to 30 mph but not quite 30 yet.

GPT-4V:
The yellow pointer is at the long tick between 0 and 20 mph. The long tick at the middle between 0 and 20 is 10. The speed is around 9 mph, as the pointer is close to 10 mph but not quite 10 yet.

零样本和少样本提示的区别

少样本提示（Zero-Shot Prompting）：没有向模型展示任何完整的示例

Prompt Copy

```
Add 2+2:
```

单样本提示（One-Shot Prompting）：只向模型展示了一个完整的示例

Prompt Copy

```
Add 3+3: 6
Add 2+2:
```

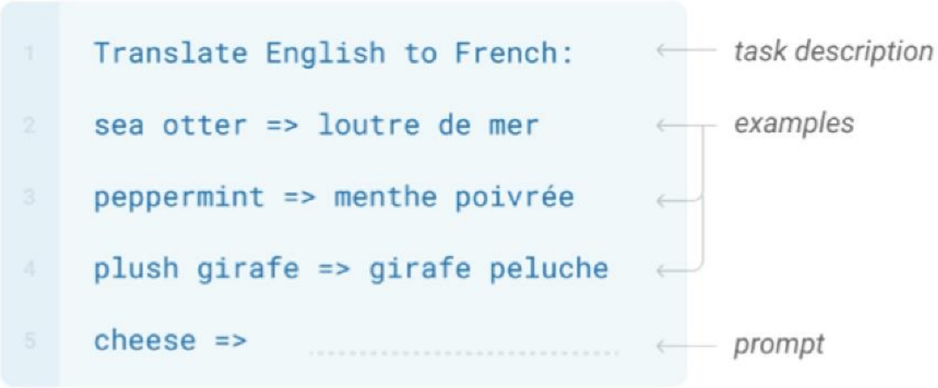
少样本提示（Few-shot Prompting）：向模型展示了至少两个完整的示例

Prompt Copy

```
Add 3+3: 6
Add 5+5: 10
Add 2+2:
```

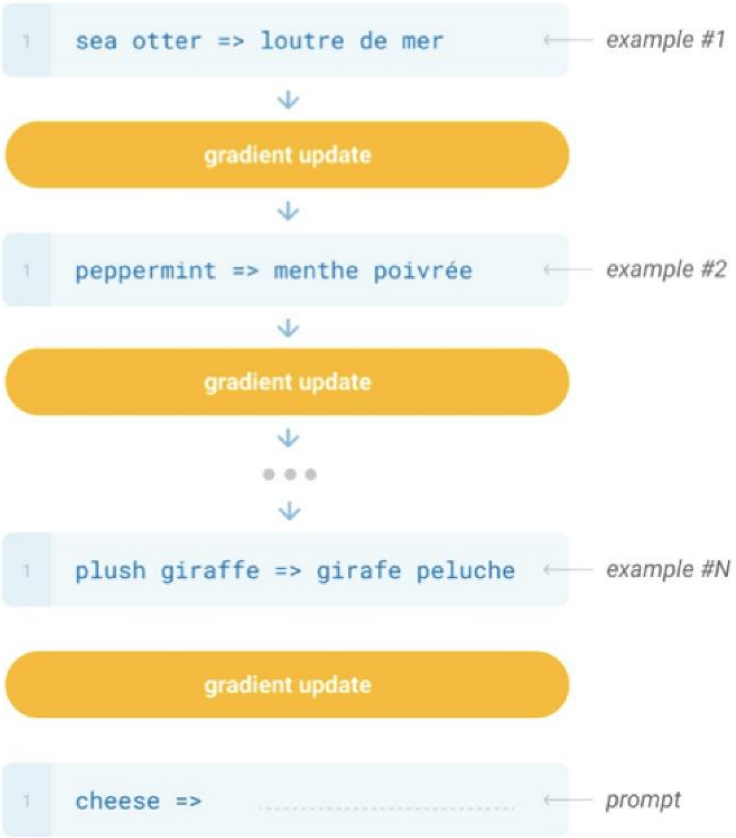
少样本提示和微调的区别

少样本提示（Few-shot Prompting）： 除了任务的描述，同时给模型几个完整的例子，不涉及模型梯度的更新



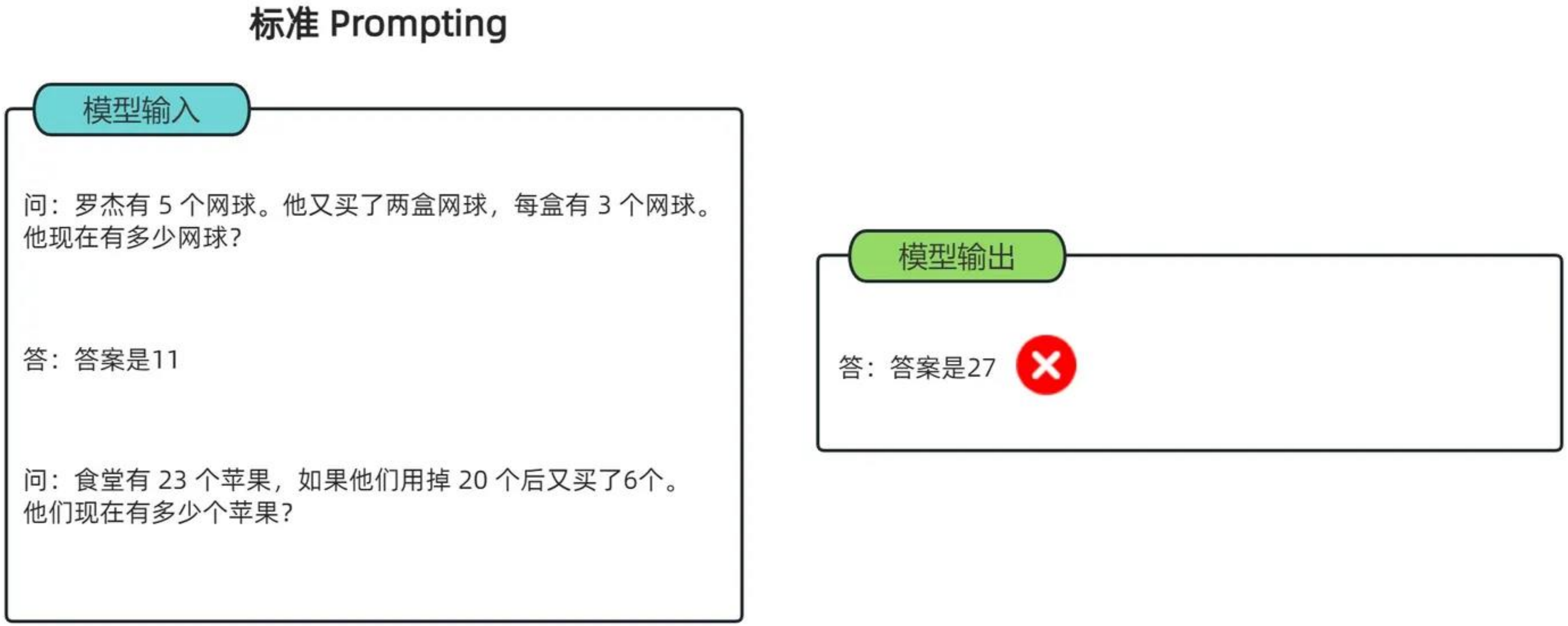
一些研究认为：上下文学习可以看作一种隐式的Fine-tuning

微调（Fine-tuning）： 通过大量的语料数据反复更新梯度来实现模型的训练



思维链（Chain of Thought)

如何让大语言模型回答/解决更复杂的问题?



https://zhuanlan.zhihu.com/p/629087587

思维链（Chain of Thought)

如何让大语言模型回答/解决更复杂的问题?

思维链（Chain of Thought，CoT）！

Chain of thought prompting elicits reasoning in large language models

零样本思维链（Zero-shot Chain of Thought）：

- LLM本身是具备推理能力，需要通过合适的提示语来让LLM释放潜力。

Zero-shot-CoT

模型输入

Q：一个杂耍演员可以玩杂耍 16 个球。一半的球是高尔夫球，其中一半的高尔夫球是蓝色的。蓝色高尔夫球有多少个？

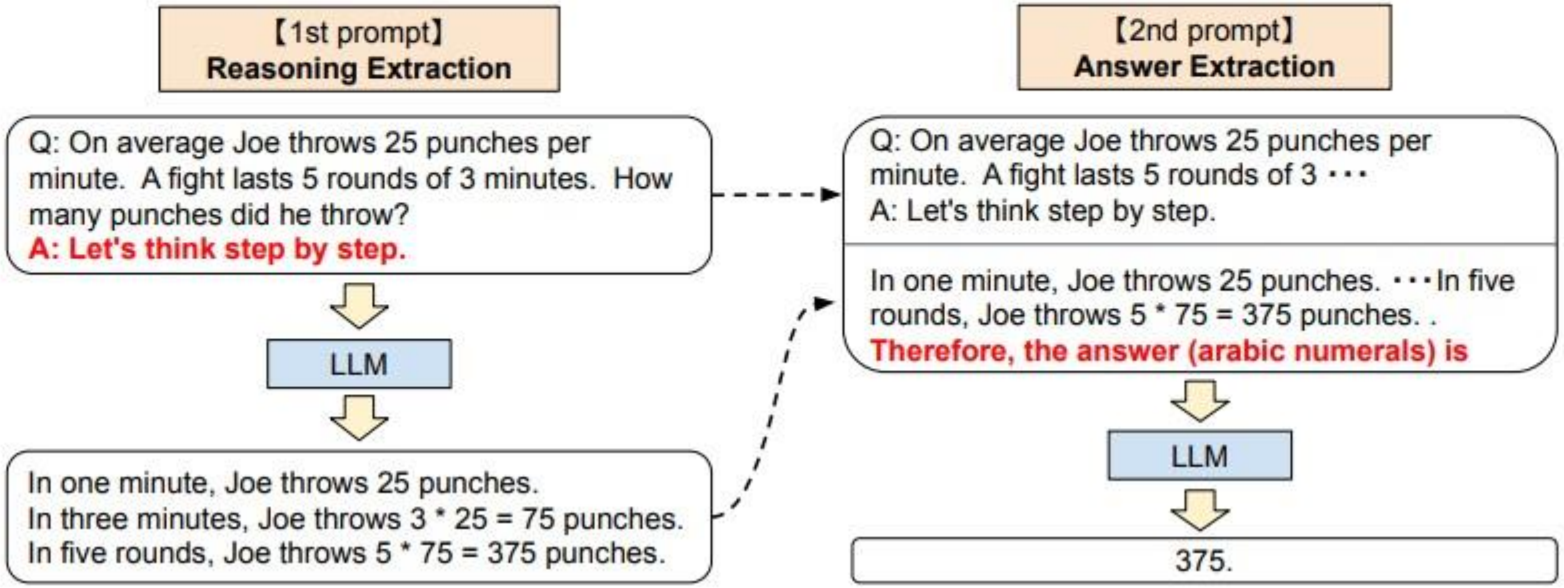
A：让我们一步步思考（Let’s think step by step）。

模型输出

答：一共有16个球。一半的球是高尔夫球，这意味着有 8 个高尔夫球。一半的高尔夫球是蓝色的，这意味着有 4 个蓝色高尔夫球。✅

零样本思维链（Zero-shot Chain of Thought）：

- 加“Let's think step by step”这句提示语，让LLM会输出具体的推理过程
- 再将生成的推理步骤和最终问题拼在一起，重新提问



少样本思维链（Few-shot Chain of Thought）：

- 为了教会LLM模型学会推理，可以给出一些人工写好的推理示例。

USER

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?

A: We start with 15 trees. Later we have 21 trees. The difference must be the number of trees they planted. So, they must have planted 21 - 15 = 6 trees. The answer is 6.

Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?

A: There are 3 cars in the parking lot already. 2 more arrive. Now there are 3 + 2 = 5 cars. The answer is 5.

Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?

A: Leah had 32 chocolates and Leah's sister had 42. That means there were originally 32 + 42 = 74 chocolates. 35 have been eaten. So in total they still have 74 - 35 = 39 chocolates. The answer is 39.

Q: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny?
A:

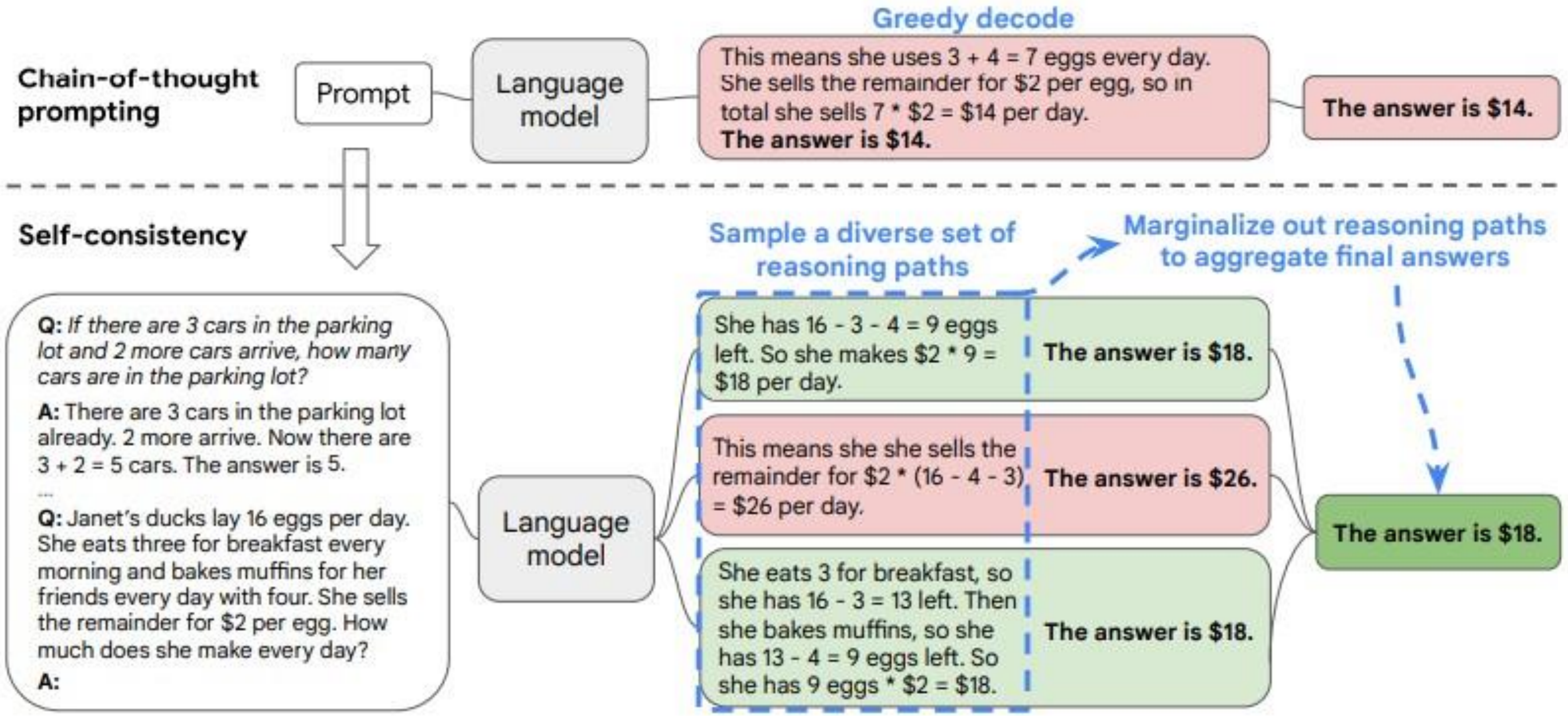
ASSISTANT

originally had 20 lollipops and now has 12. That means he gave away 20 - 12 = 8 lollipops. The answer is 8.

自洽性（Self-Consistency）

自洽性（Self-Consistency）：

- 自洽性是对 CoT 的一个补充，它多次生成推理链条，然后取多数答案作为最终答案

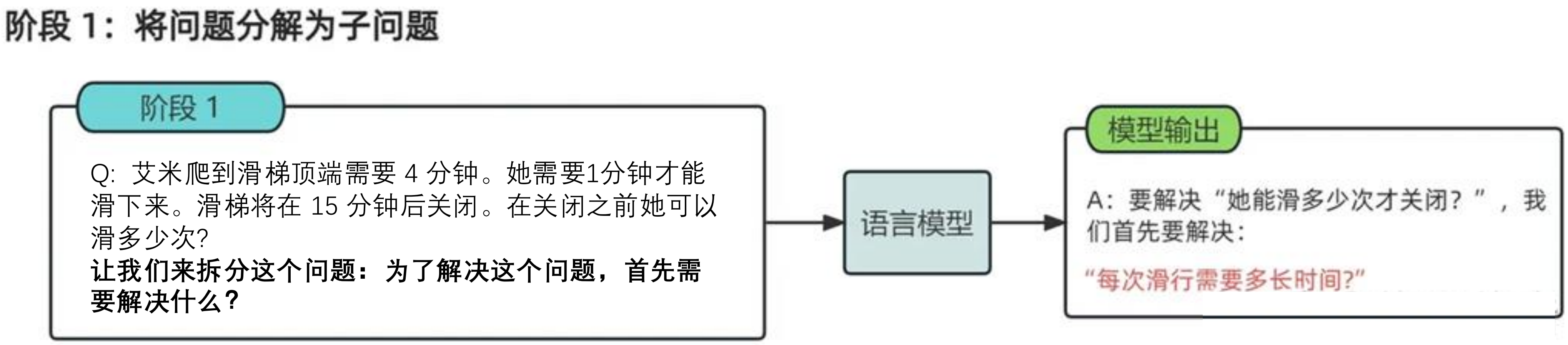


Self-Consistency Improves Chain of Thought Reasoning in Language Models

最少到最多提示（Least-to-most prompting）

最少到最多提示（Least-to-most prompting, LtM）：

- 最少到最多提示过程将思维链提示过程进一步发展，首先将问题分解为子问题，然后逐个解决。它是受到儿童教育策略的启发而发展出的一种技术。

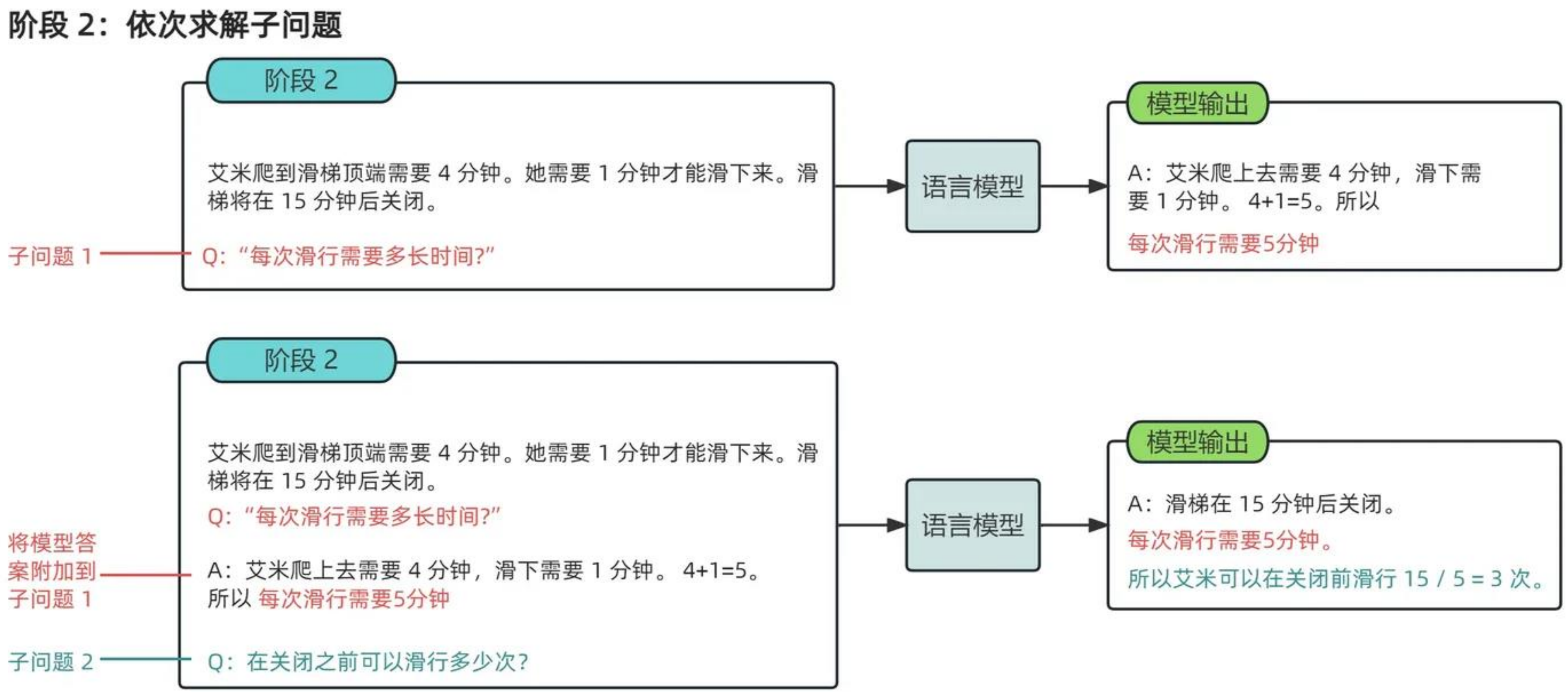


该阶段中，使用提示“让我们来拆分这个问题”，以此生成解决问题的步骤（子问题）

Least-to-Most Prompting Enables Complex Reasoning in Large Language Models

最少到最多提示（Least-to-most prompting）

最少到最多提示（Least-to-most prompting, LtM）：



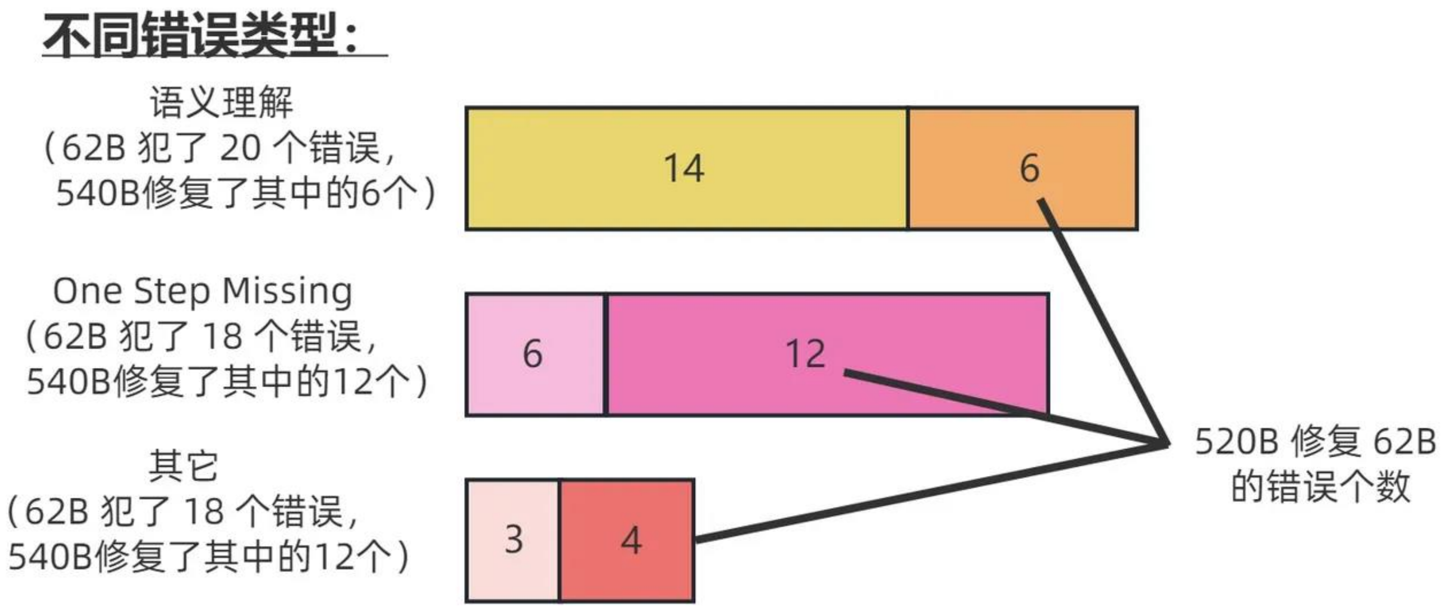
该阶段中，使用第一阶段产生的子问题及其答案，来询问原问题的答案

Least-to-Most Prompting Enables Complex Reasoning in Large Language Models

思维链的局限性

思维链必须在模型规模足够大时才能涌现

- Google PaLM 模型在扩展到 540B 参数时，与思维链提示结合，才表现出了先进的性能
- 策略问题需要大量的世界知识，而小型模型没有足够的参数来记忆这些世界知识，所以也不太可能产生正确的推理步骤



思维链的局限性

思维链的应用领域有限

- 思维链只是在一些有限的领域，比如数学问题，五个常识推理基准（CommonsenseQA，StrategyQA，Date Understanding 和 Sports Understanding 以及 SayCan）上显现出作用，在其他类型的任务（如机器翻译）上性能提升效果还有待评估。

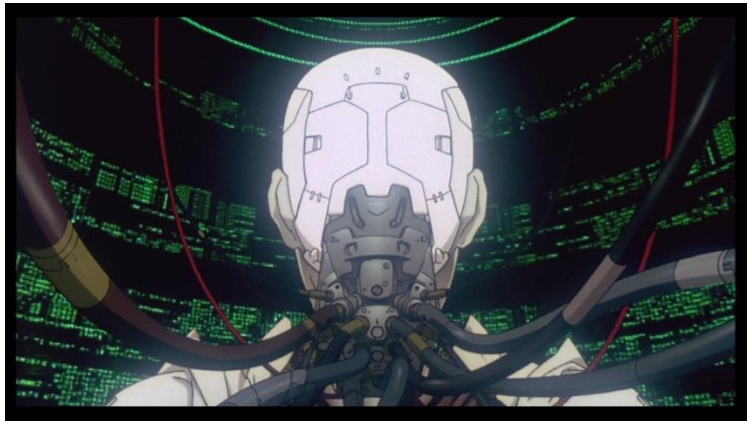
大语言模型依然不能解决小学水平的数学问题

- 没有思维链，数学推理几乎一定不行。但有了思维链，大语言模型也可能出现错误推理，尤其是非常简单的计算错误：

$$2 + 2 = 5$$

深入发掘思维链的潜力

- LLM提供了与人类动态交互、理解人类指令的能力
- 进一步，利用思维链技术，我们能够有效拆分复杂问题，提升LLM解答的可靠性和可解释性
- 如果要充分发掘思维链潜力，可以把LLM当做一个多功能“中枢”，通过它调用各种工具和资源
- 当思维链技术与各种工具API集成时，它带来的解释能力、推理能力能够延伸到更广泛的应用中，例如数据分析、数学推理、图像理解等等
- 这为解决更加复杂和多维的问题提供了新的途径



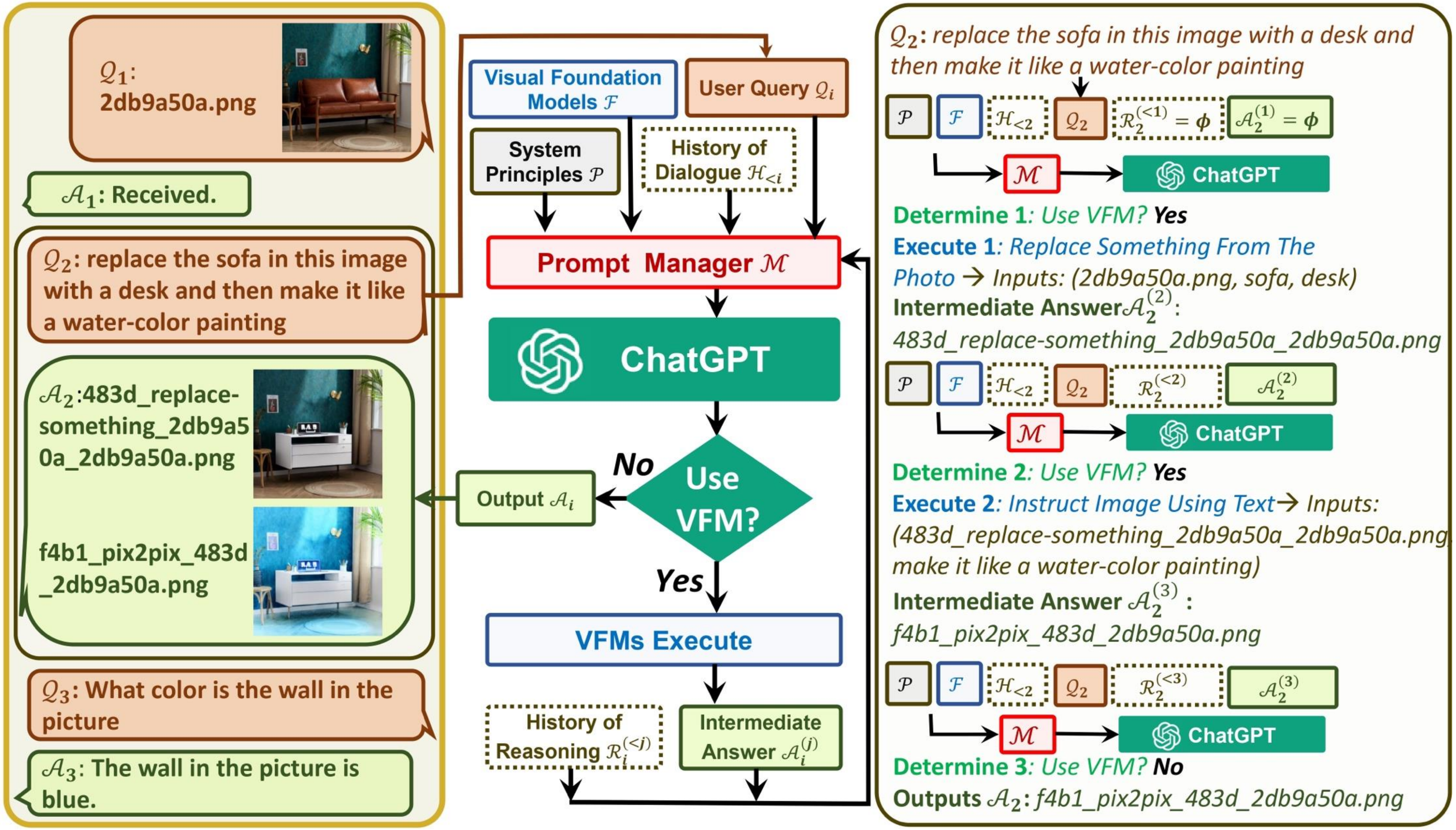
思维链 APIs：LangChain

LangChain 是一个用于开发由语言模型支持的应用程序的框架。

- 具有上下文感知能力：将语言模型连接到上下文源
- 推理能力：依靠语言模型进行推理（CoT）

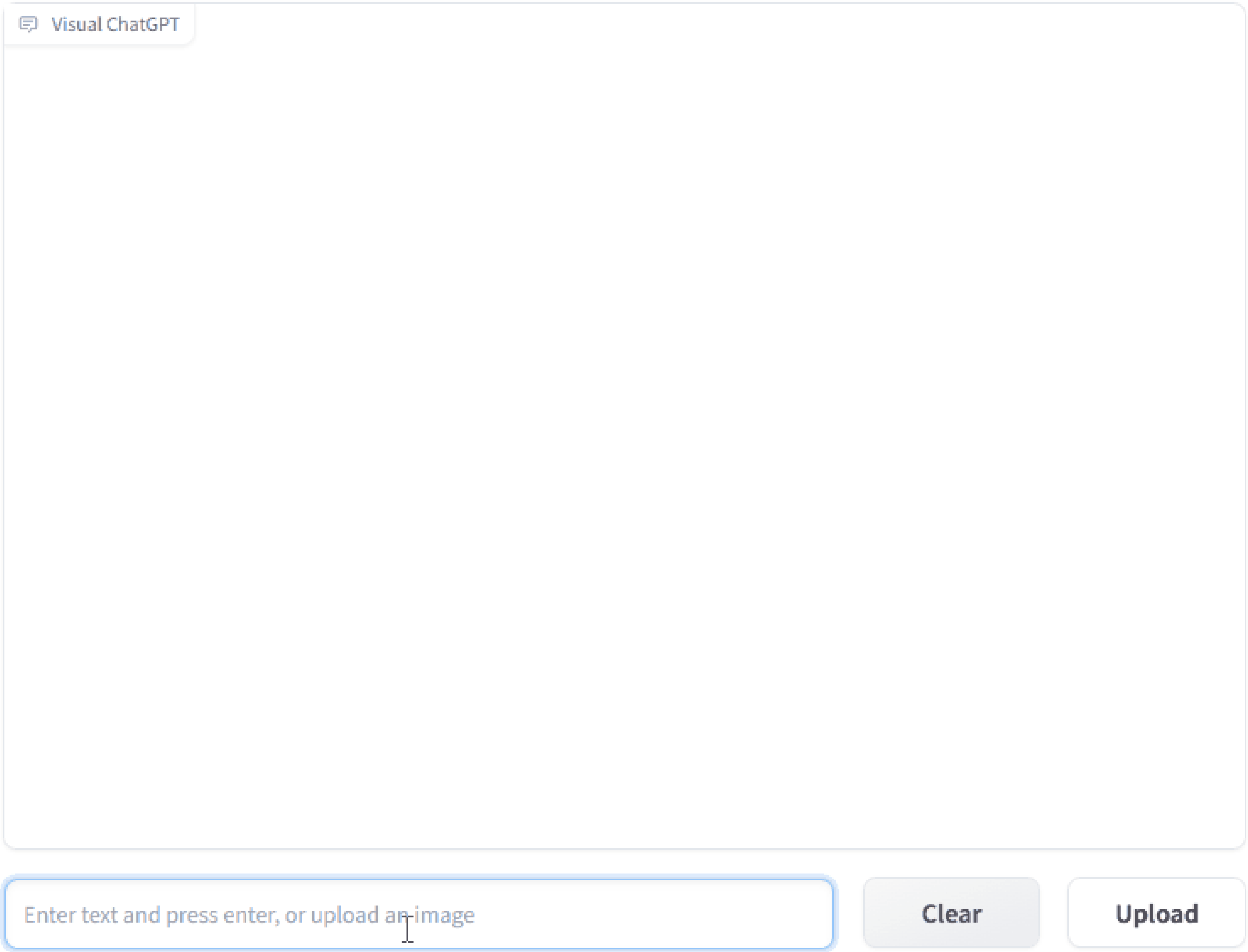
https://python.langchain.com/docs/get_started/introduction

LangChain的应用案例： Visual ChatGPT



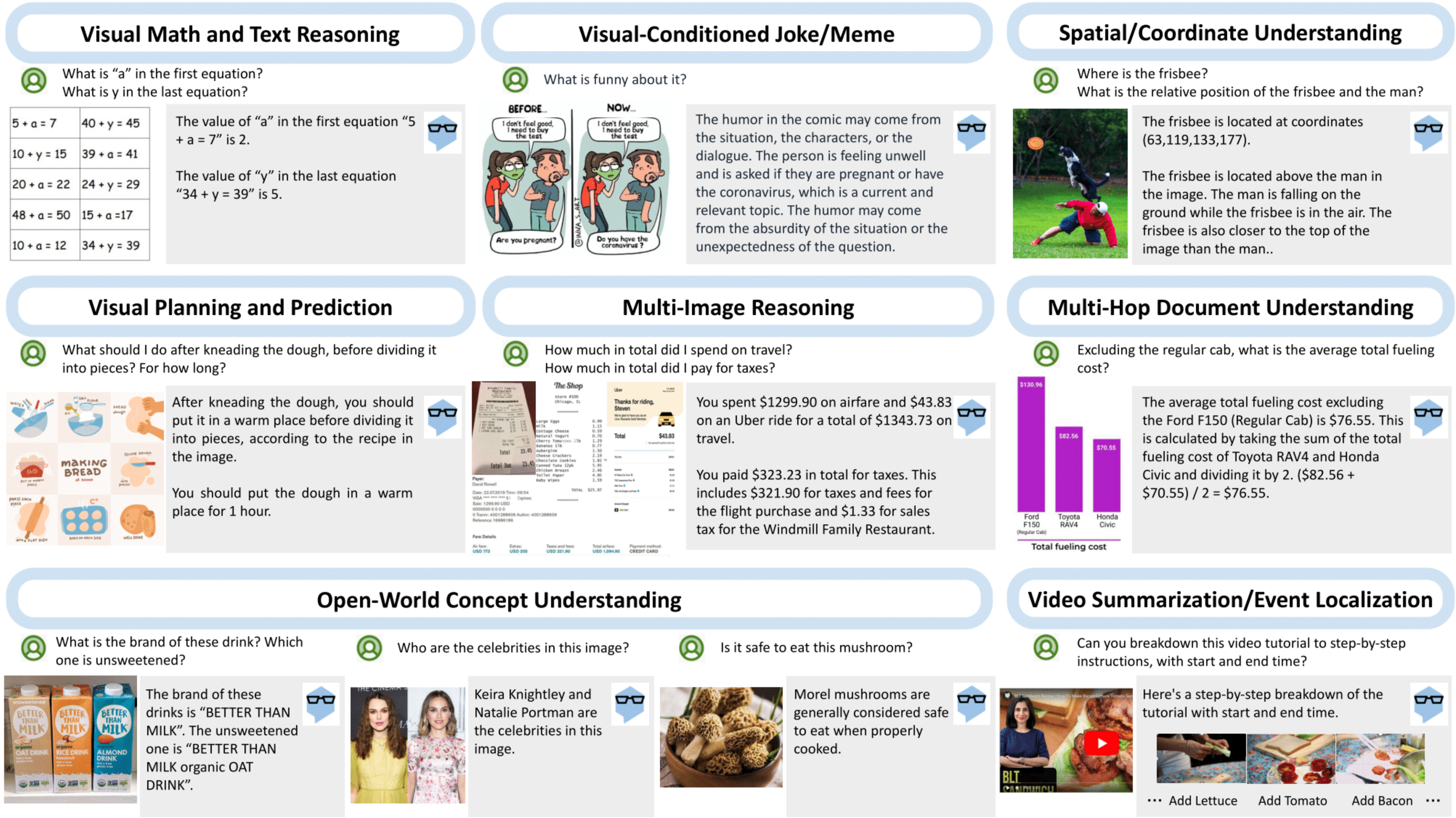
Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models

LangChain的应用案例： Visual ChatGPT



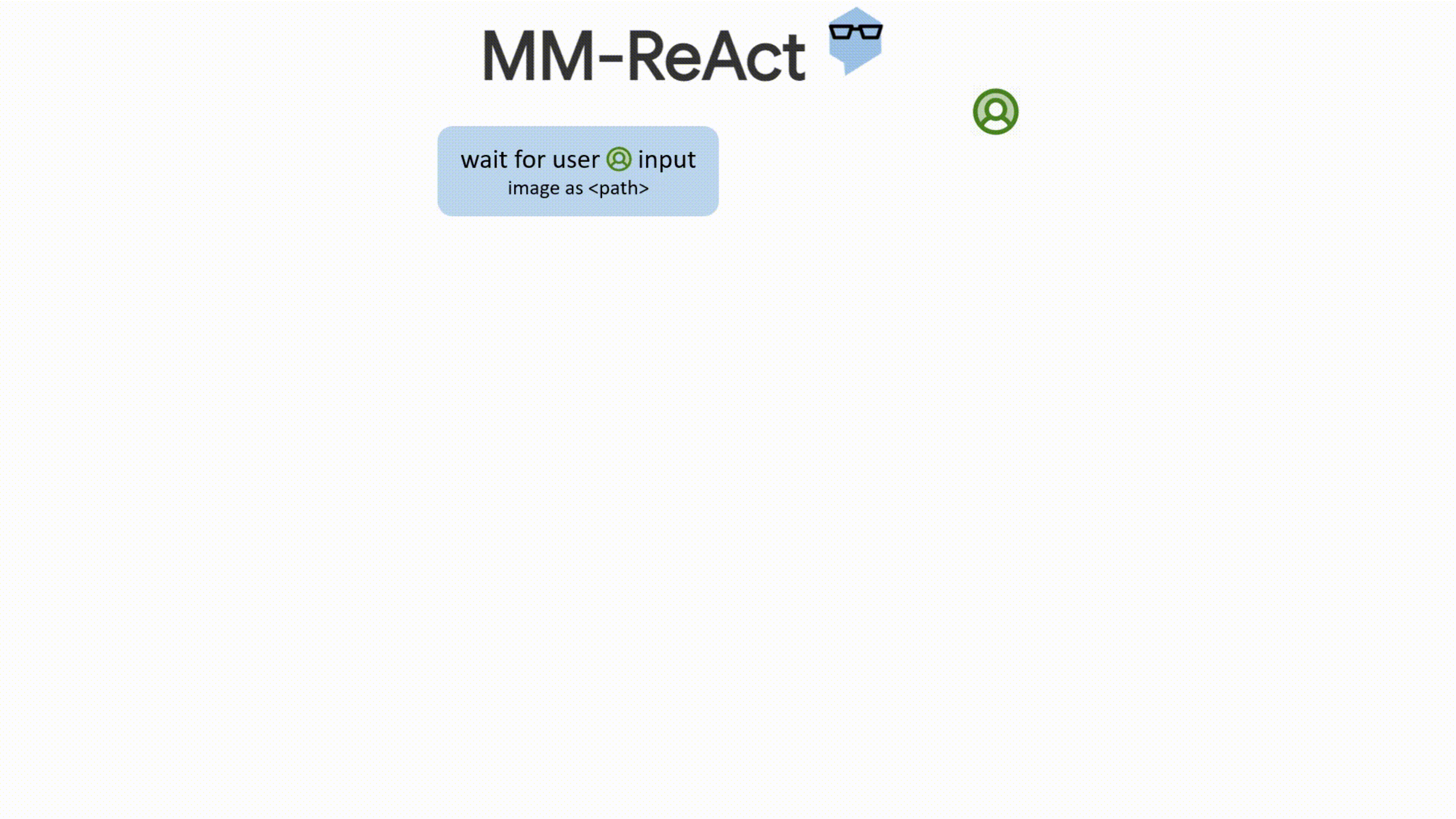
https://huggingface.co/spaces/microsoft/visual_chatgpt

LangChain的应用案例： MM-REACT



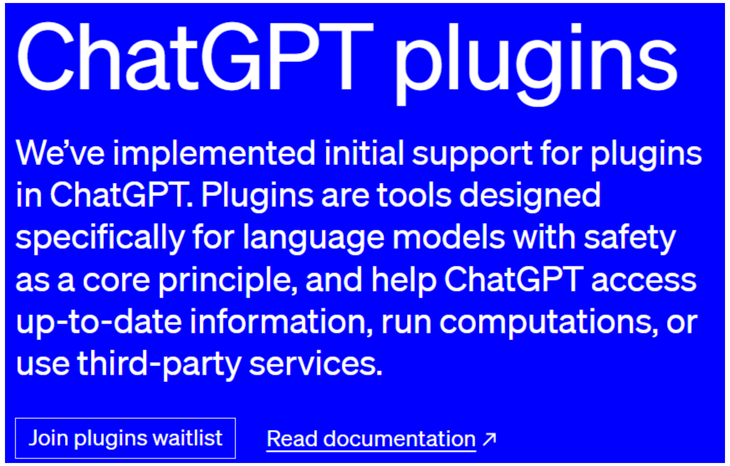
MM-REACT: Prompting ChatGPT for Multimodal Reasoning and Action

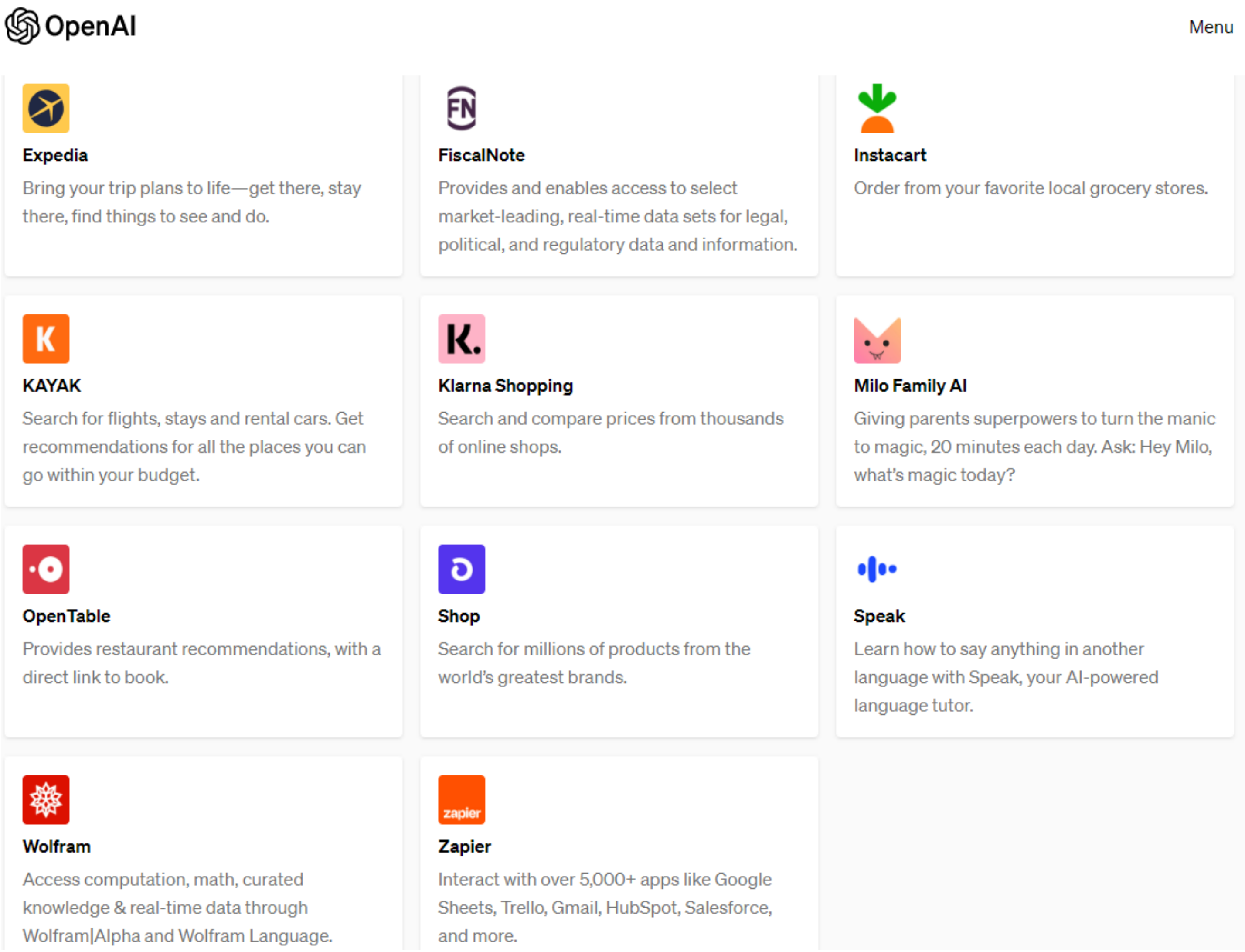
LangChain的应用案例： MM-REACT



https://huggingface.co/spaces/microsoft-cognitive-service/mm-react

ChatGPT 插件

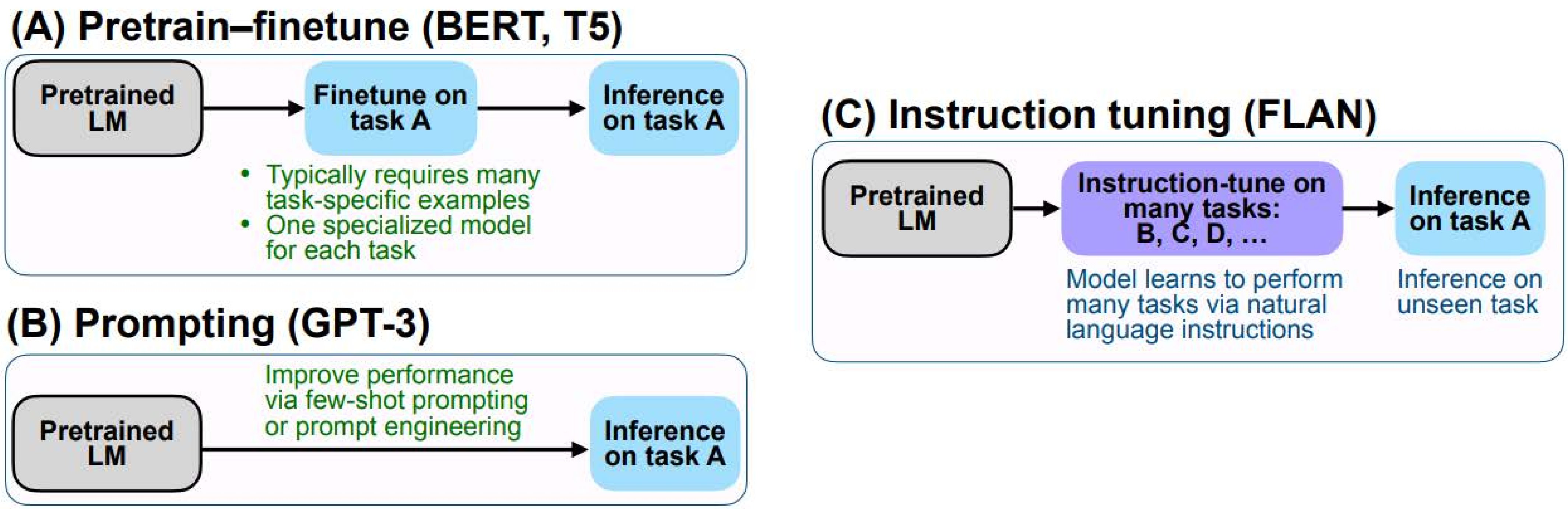




https://openai.com/blog/chatgpt-plugins

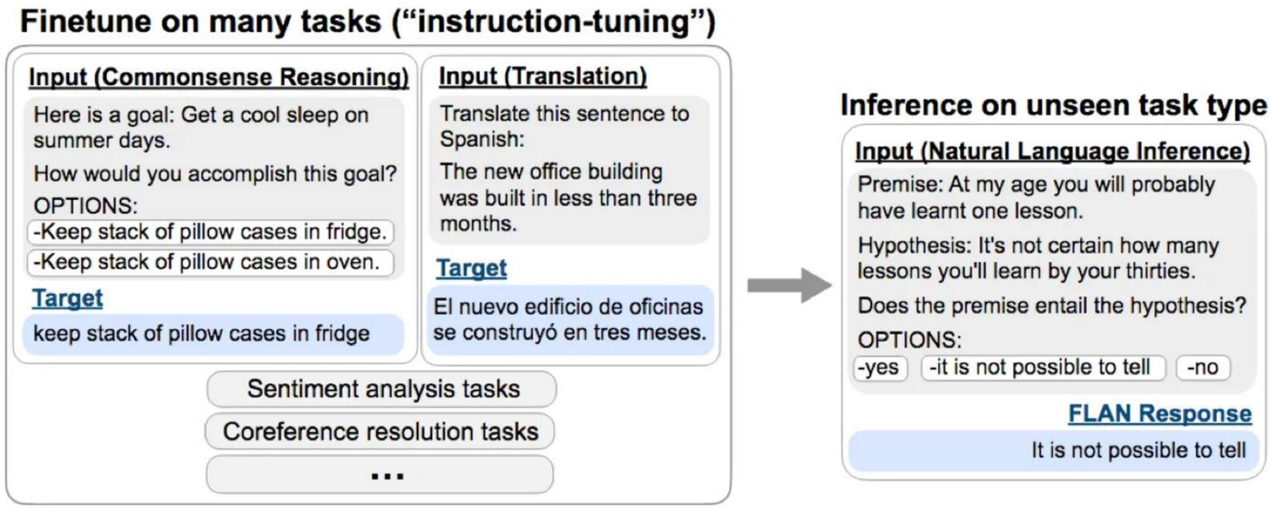
大语言模型的使用方式

- **提示（Prompting）**：无需训练，通过少量提示提升性能
  - 零样本提示（Zero-shot Prompting）/ 指令提示（Instruction Prompting）
  - 少样本提示（ Few-shot Prompting）
- **指令微调（Instruction tuning）**：通过语言指令让模型学习较多任务，从而完成未见过的任务
- **微调（Fine-tuning）**：需要尽可能多的任务相关的数据；每一个任务需要单独训练一个模型



指令微调（Instruction tuning）

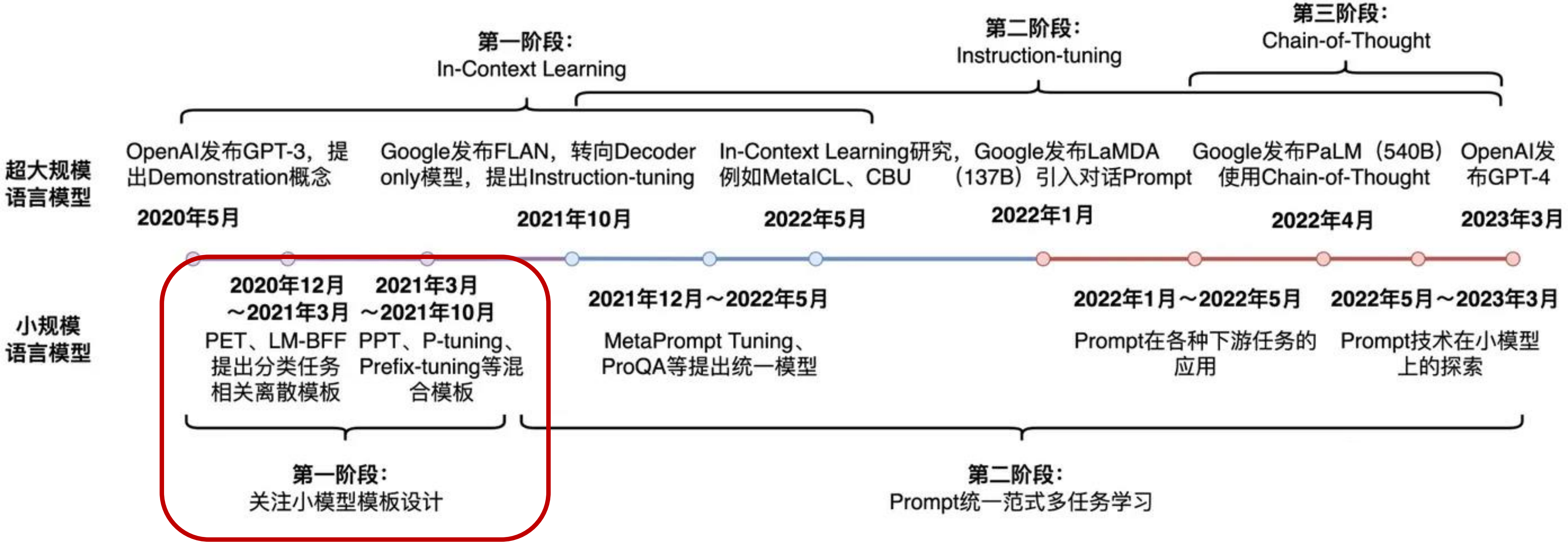
- 假设有很多NLP任务，研究人员针对每个NLP任务设计多种提示模板作为指令，并使用这些指令以及相应的数据对模型进行微调。
- 训练完成后，给大语言模型一个它从未见过的全新任务的指令，即让大语言模型解决零样本(zero-shot)任务。零样本上任务上的表现体现了模型的泛化能力。



问题：微调整个大模型成本太高，不可能训练所有模型参数

大模型轻量化微调技术

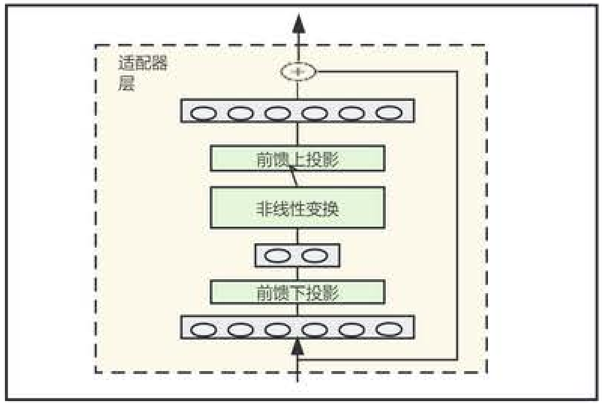
- 为了解决训练成本问题，科学家提出了Parameter-Efficient Fine-Tuning (PEFT) 技术。
- PEFT 技术旨在尽可能减小微调参数的数量和计算复杂度的前提下提高预训练模型在新任务上的性能



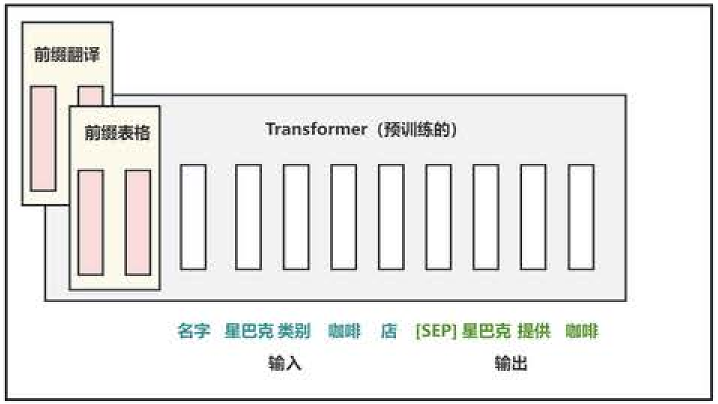
大模型轻量化微调技术

PEFT 技术中的常用方法

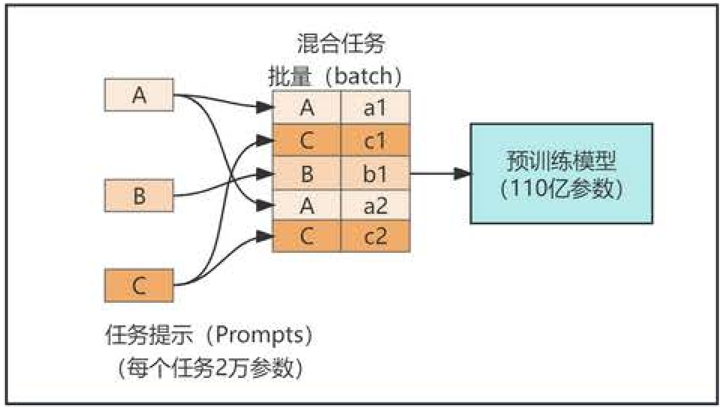
Adapter（谷歌2019）



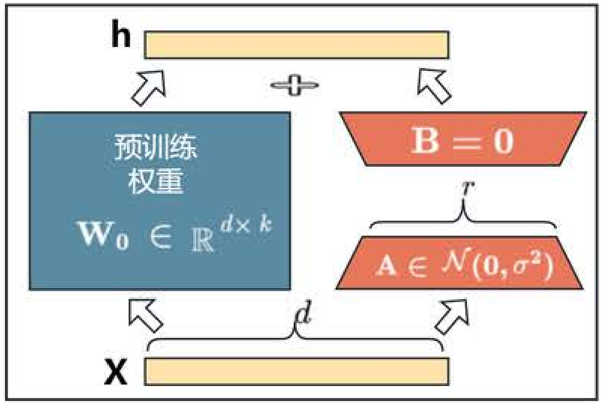
Prefix Tuning（斯坦福2021）



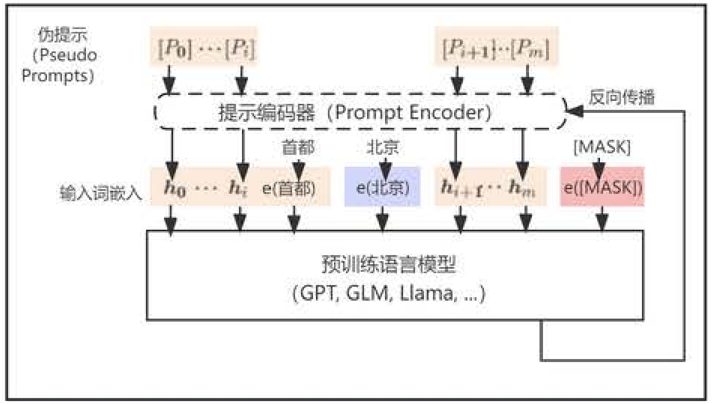
Prompt Tuning（谷歌2021）



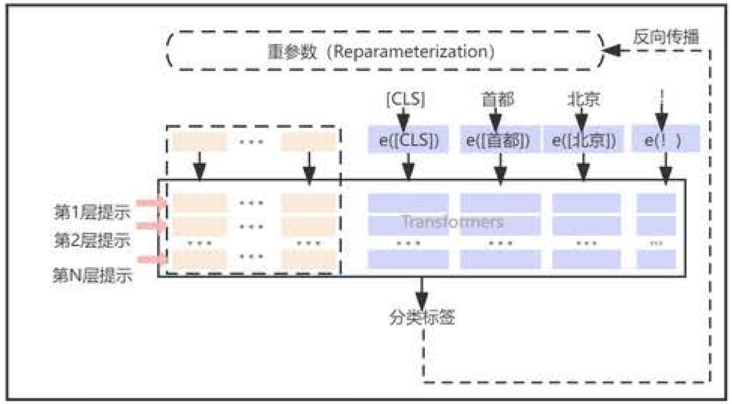
LORA（2021）



P-Tuning（清华2022）



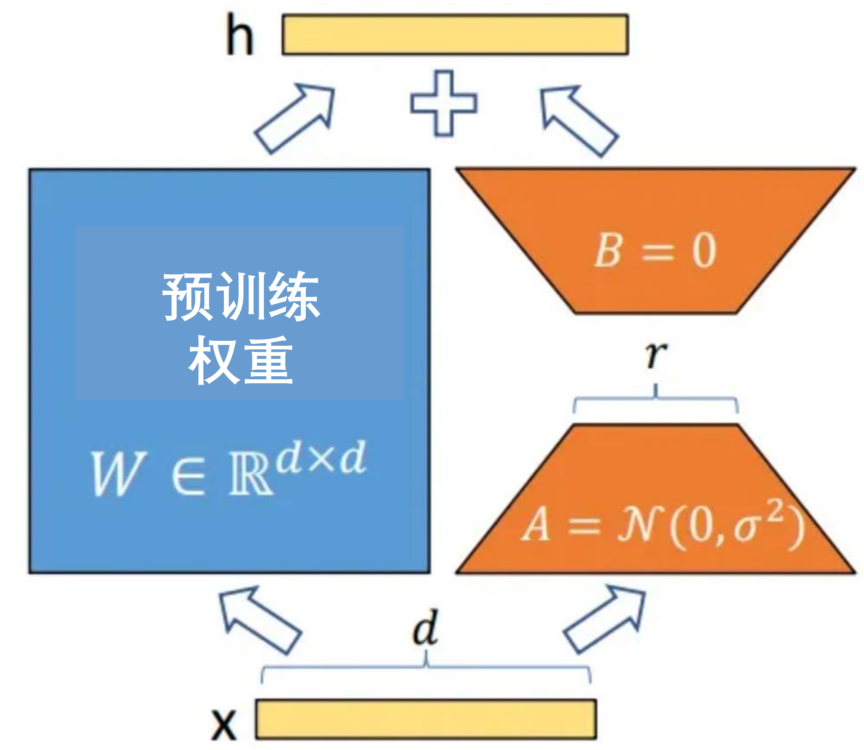
P-Tuning v2（清华2022）



大模型轻量化微调技术，LoRA

低秩自适应（Low-Rank Adaptation）：

- 核心思想：在原始预训练语言模型（PLM）的某些层中，插入低秩矩阵来微调模型。
- 具体步骤：
  - 选择需要修改的权重矩阵，例如某层注意力的 W_Q, W_K, W_V矩阵
  - 将选择的权重矩阵替换为两个低秩矩阵的乘积A, B
  - 微调时只训练降维矩阵A与升维矩阵B
- 只需微调少量参数！

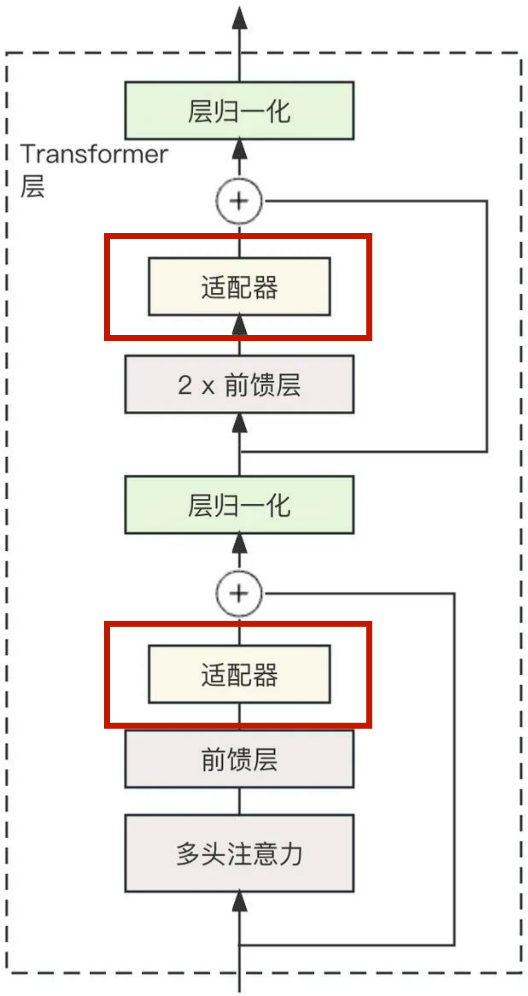


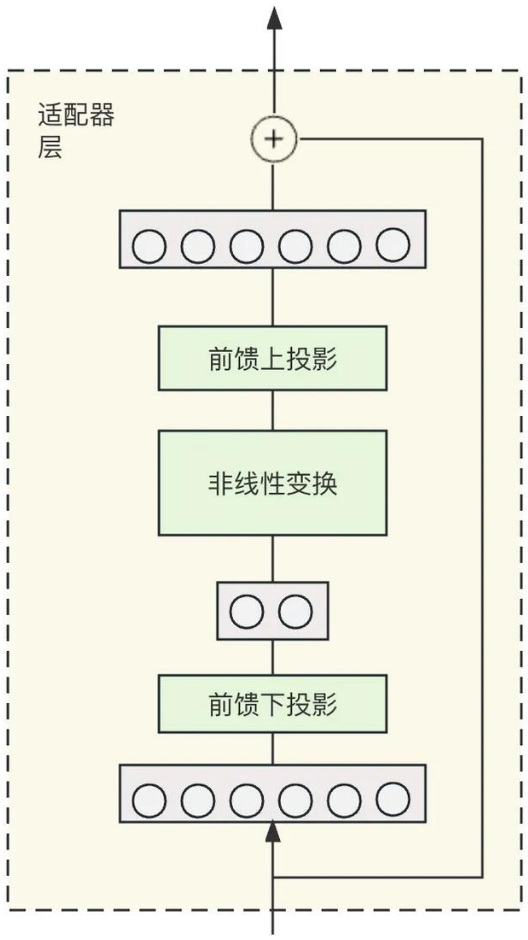
LoRA: Low-Rank Adaptation of Large Language Models

大模型轻量化微调技术， Adapter

适配器（Adapter）：

- 在预训练模型每一层(或某些层)中添加Adapter模块
- 微调时冻结预训练模型主体，由Adapter模块学习特定下游任务的知识



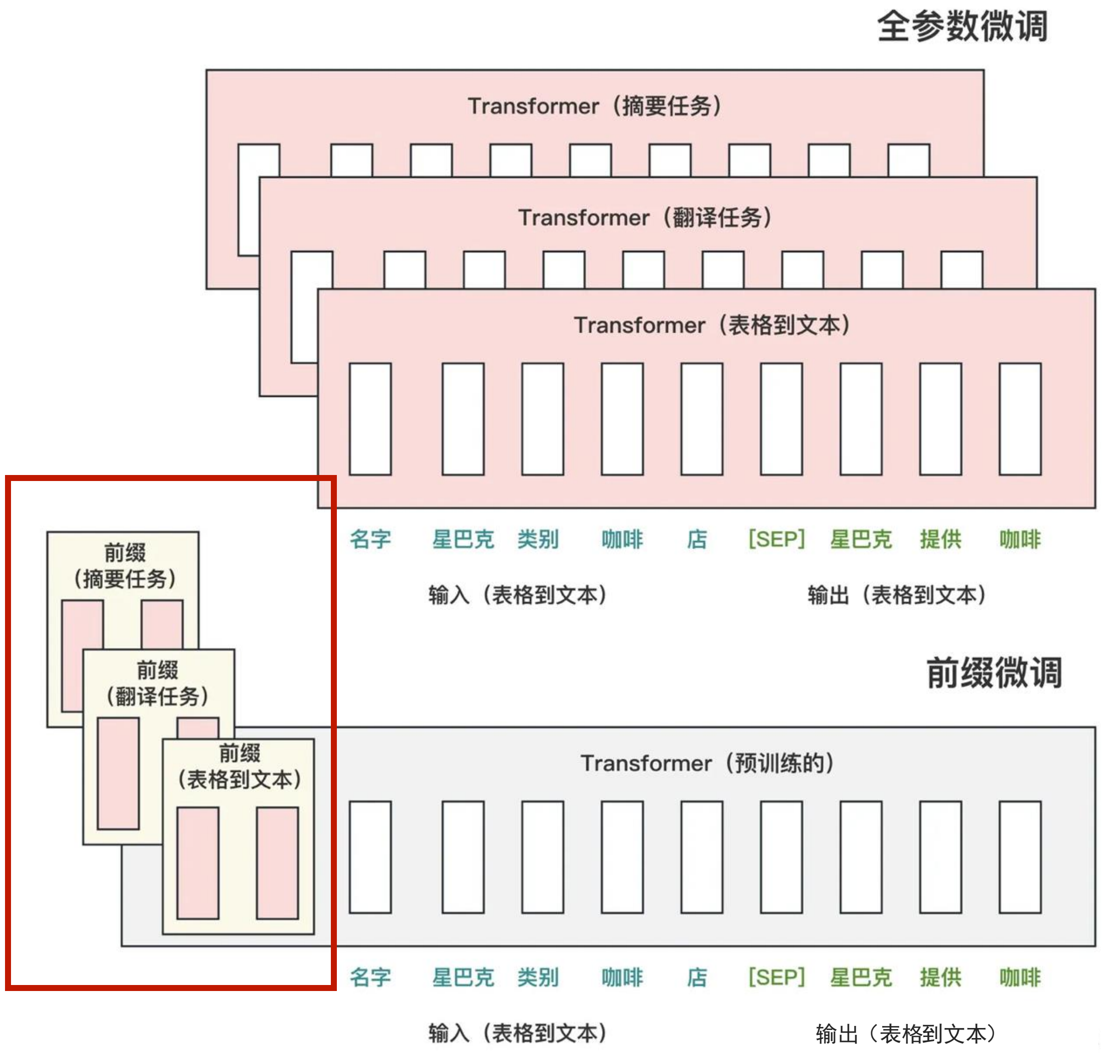


Parameter-Efficient Transfer Learning for NLP
MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer

大模型轻量化微调技术， Prefix-tuning

前缀微调（Prefix-tunning）：

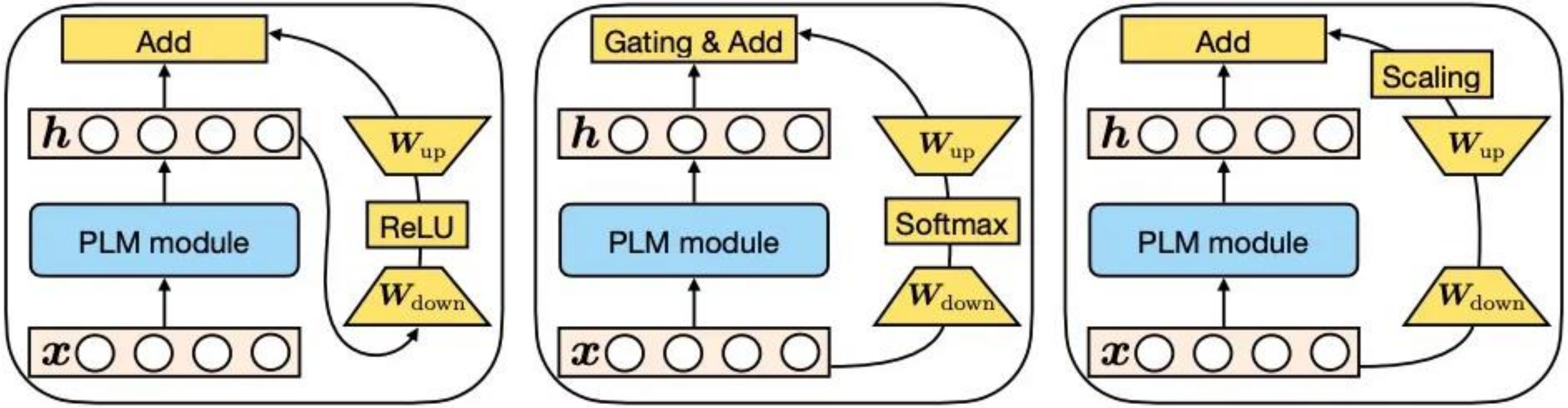
- 前缀微调将一个连续的特定于任务的向量序列添加到输入
- 与提示不同的是，前缀完全由自由参数组成
- 对每个额外任务，只产生非常小的开销



Prefix-Tuning: Optimizing Continuous Prompts for Generation

三种微调技术的对比

- Adapter，Prefix-Tuning，LORA 都定义为预训练模型中添加可调整的特定的隐层状态，只是设计的参数维度、修改函数的计算和位置不同。



适配器　　　　前缀微调　　　　低秩自适应

大语言模型微调开源项目

- **LLaMA**： Meta 的全新大型语言模型
- **Stanford Alpaca**：一个指令调优的 LLaMA 模型
- **Vicuna**：基于 LLaMA 的微调大语言模型
- **ProtTrans**： 是国内最大的蛋白质预训练模型
- **华驼 (HuaTuo)** ： 基于中文医学知识的 LLaMA 微调模型。
- **LaWGPT**：基于中文法律知识的大语言模型
- **CodeCapybara**：代码生成大语言模型。
- ……

谢谢！

Jifeng Dai
Department of Electronic Engineering
daijifeng@tsinghua.edu.cn