大模型应用实战课 Week 1-2

今日课程内容

- 目录：
    - GPT 系列模型架构详解：生成式大模型为什么这么强?
        - Transformer架构详解
            - Attention机制的原理及作用
            - GPT 的模型基础：Decoder如何生成内容
        - GPT系列模型原理详解
            - GPT 1与大模型训练范式
            - GPT 2与zero—shot
            - GPT 3与in-context learning
            - GPT 4与思维链涌现
    - GPT-5技术发展前瞻

- 什么是 GPT?
- GPT 是如何做到将文字组成一句话并且输出的?
- GPT 为什么能够按照我们问题输出相关的且正确的答案?
- ……

1、GPT 系列模型架构详解：生成式大模型为什么这么强?

什么是 GPT?

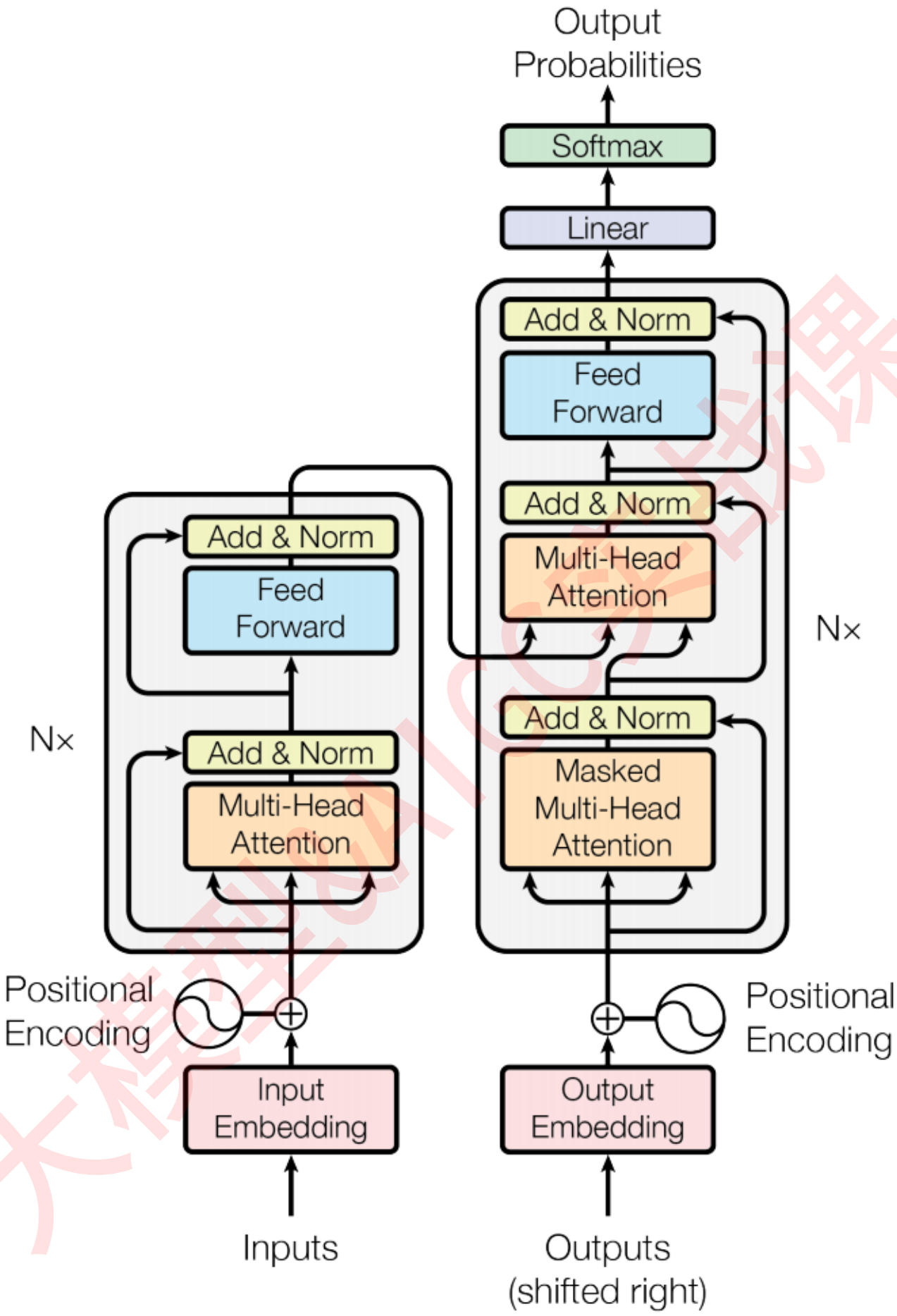
GPT 是Generative Pre-trained Transformer（生成式预训练 transformer）的缩写，GPT由OpenAI开发，并已推出多个版本。

GPT模型基于Transformer架构，该架构是一种用于处理序列数据的神经网络模型。Transformer由编码器（Encoder）和解码器（Decoder）组成，其中编码器用于学习输入序列的表示，解码器用于生成输出序列。GPT主要采用了transformer的解码器部分，用于构建语言模型。

GPT模型通过在大规模文本数据上进行无监督预训练来学习语言的统计特征和语义表示。它使用了自回归的方式，即基于前面已经生成的词来预测下一个词。通过这种方式，GPT模型可以学习到词之间的语义和语法关系，以及句子和文本的整体上下文信息。

Transformer

文献："Attention is All You Need"（Vaswani, et al., 2017）
https://arxiv.org/abs/1706.03762

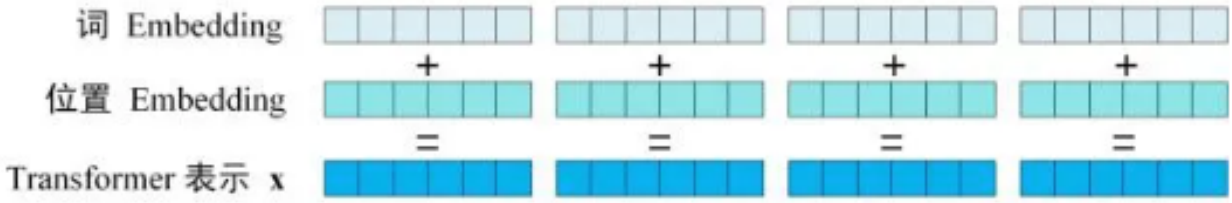


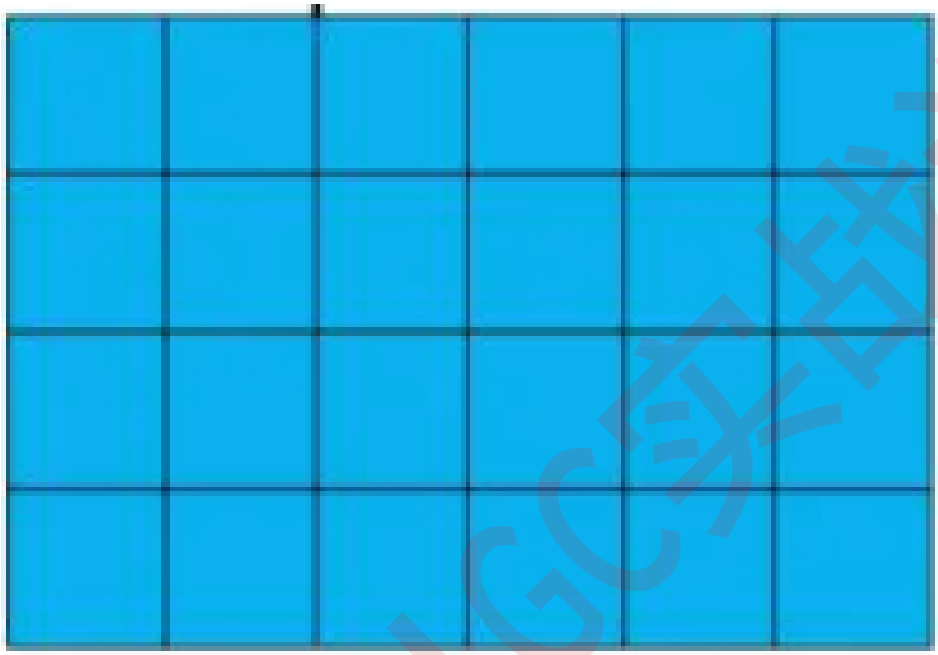
两个重点：

1、transformer 模型的核心机制：注意力机制

2、Decoder 的生成式输出到底是什么意思，怎么做的

(1) 单词Embeding以及位置Embeding



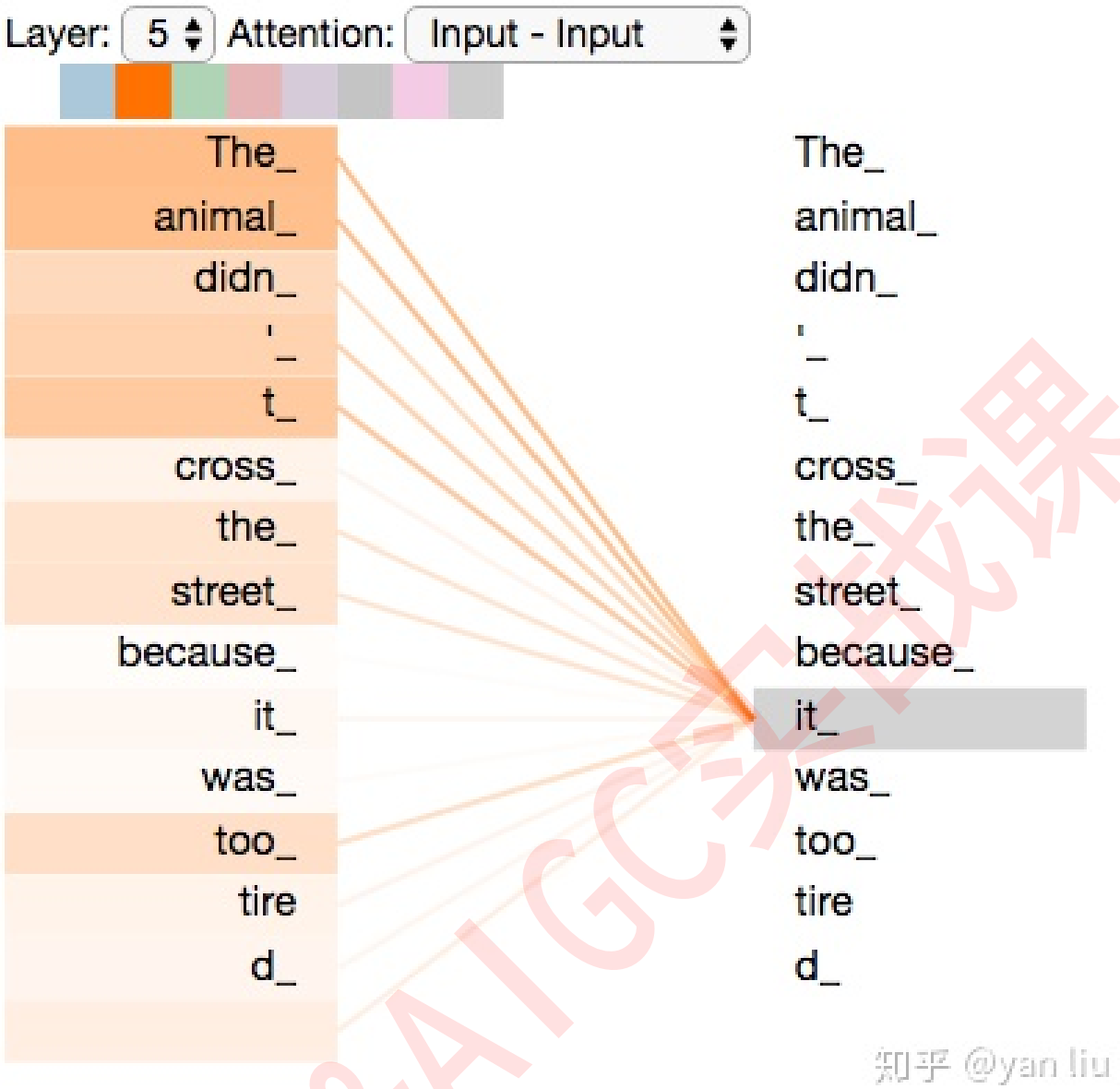


单词Embedding的方法有很多，例如Word2Vec、GLove等预训练算法得到，或者 transformer 种也可以训练的到。
位置Embedding是指句子种的每个词位置的信息，由于Transformer使用的是全局信息，没法捕捉到每个单词的顺序信息，而句子种每个词的顺序信息对对NLP各种任务来说又是非常重要的，所以需要使用Embedding 来保存单词在句子中的位置信息。

（2）注意力机制

注意力机制，到底“注意”了什么?

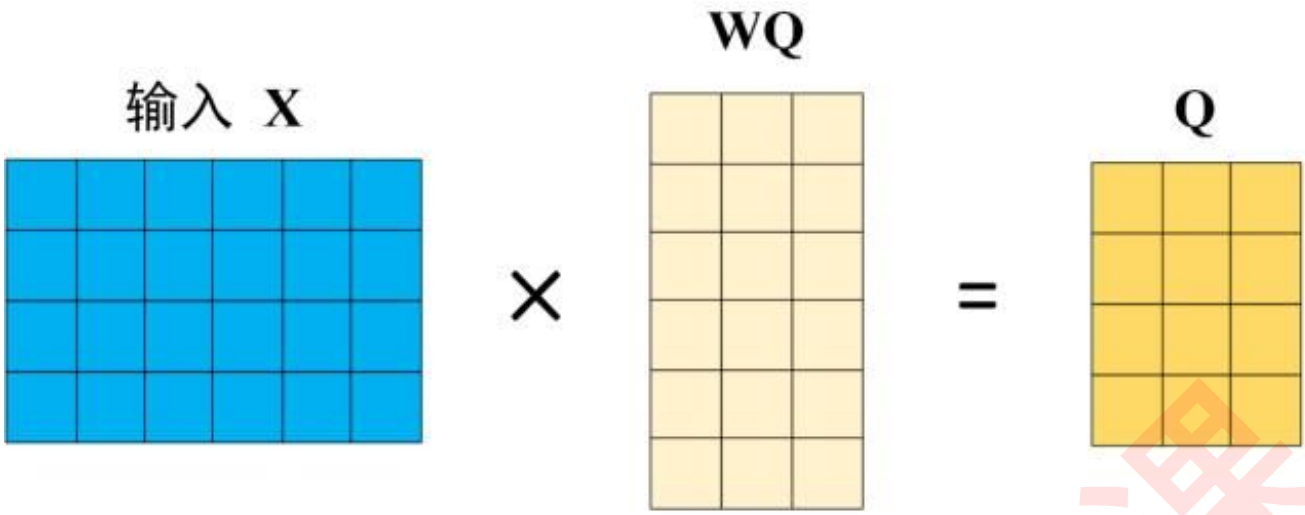
上下文信息(重构词向量)

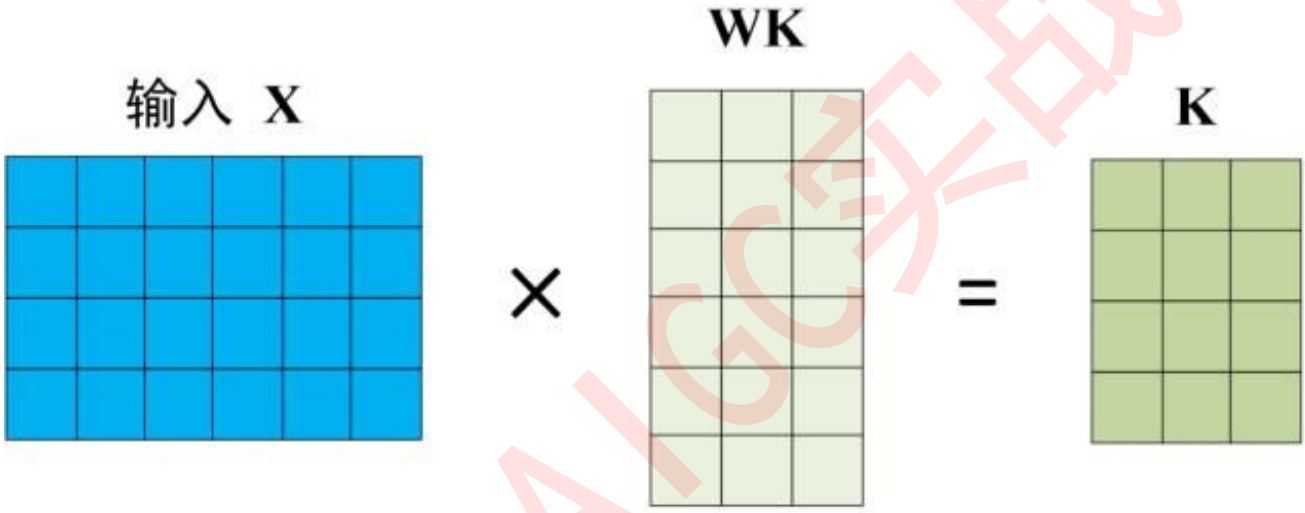


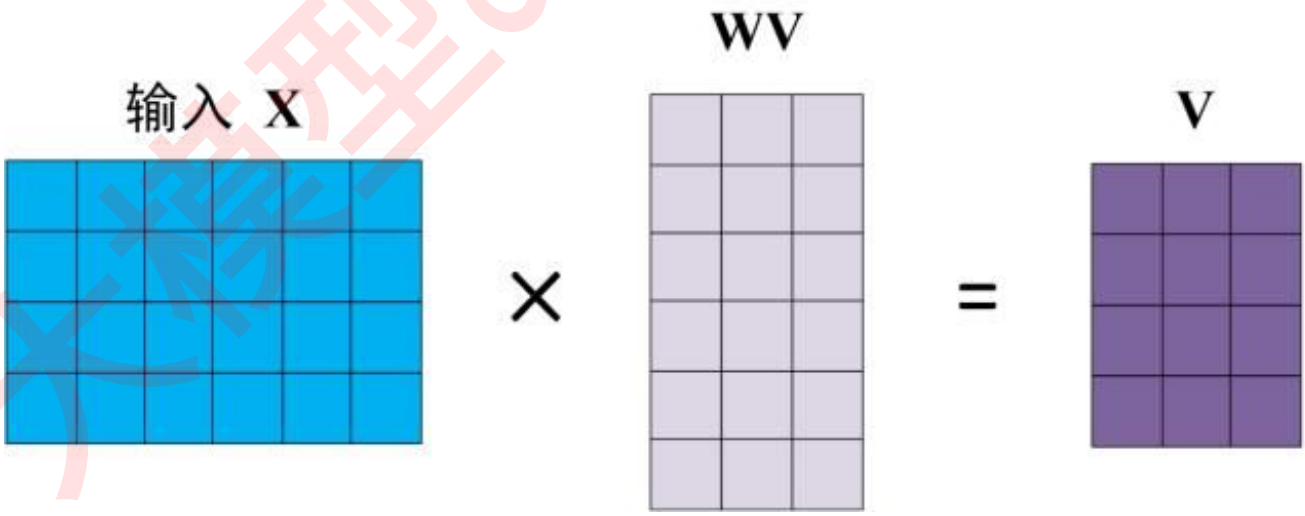
例：The animal didn't cross the street because it was too narrow

自注意力机制（self-attention）

注意力机制的三个重要元素：Q（query：某个单词向其他的词发出询问）；K（Key：某个单词回答其他词的提问）；V（value:某个单词的实际值）

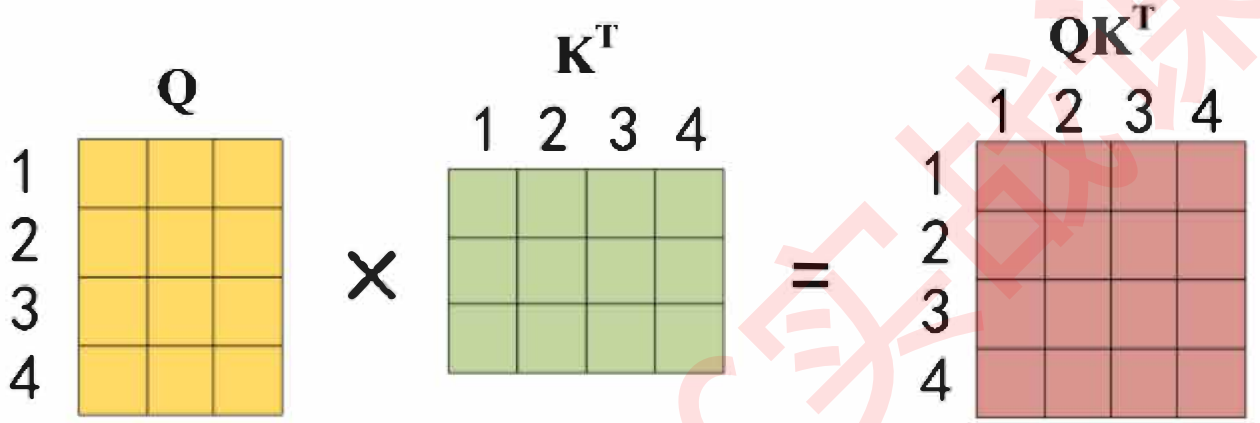


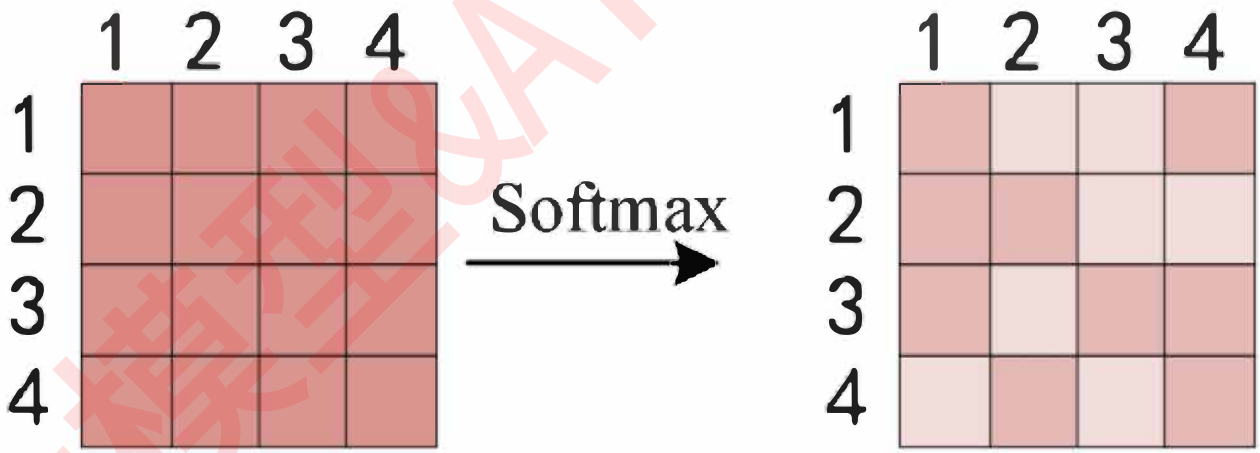


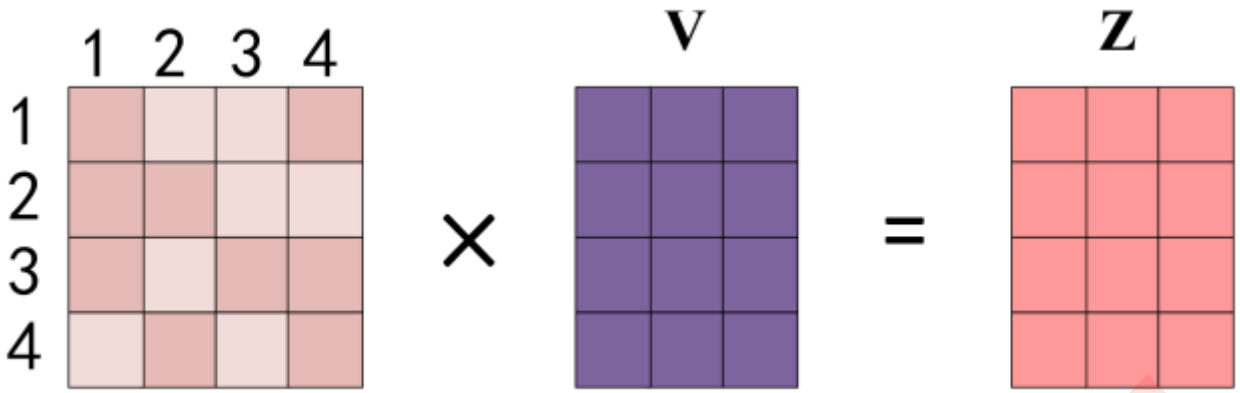


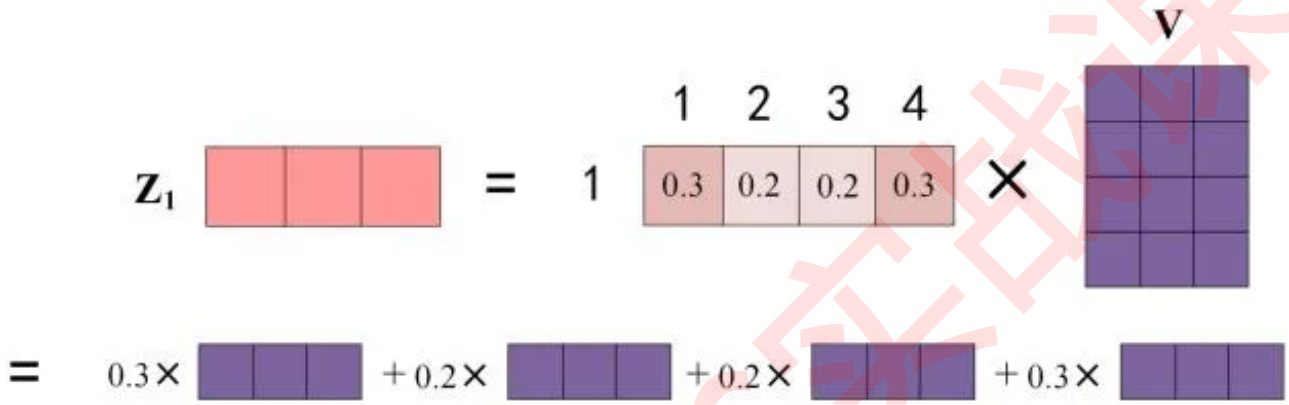
$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

d_k是Q,K矩阵的列数，即向量维度

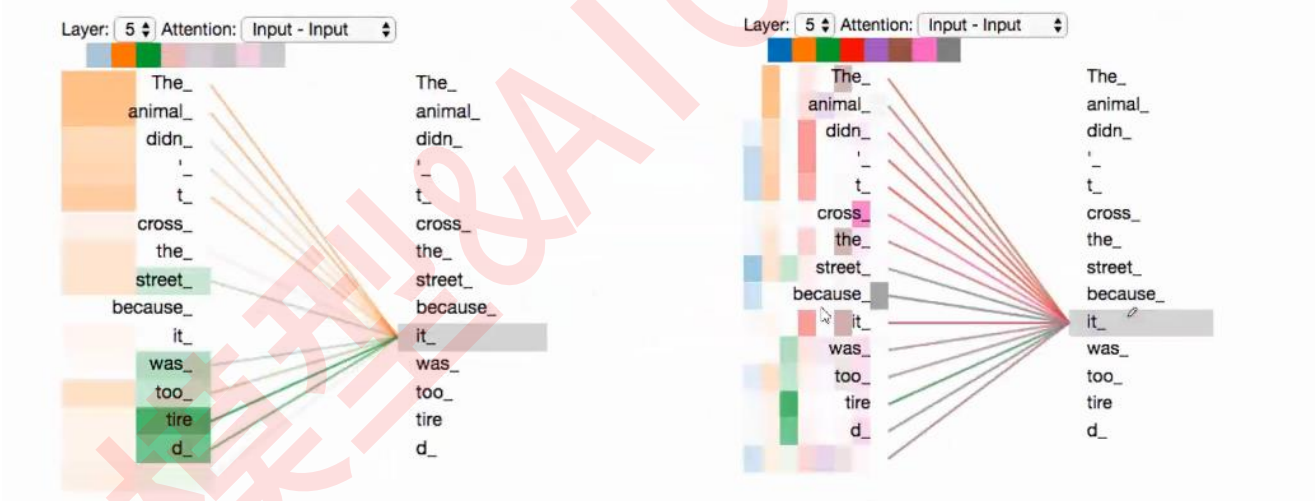




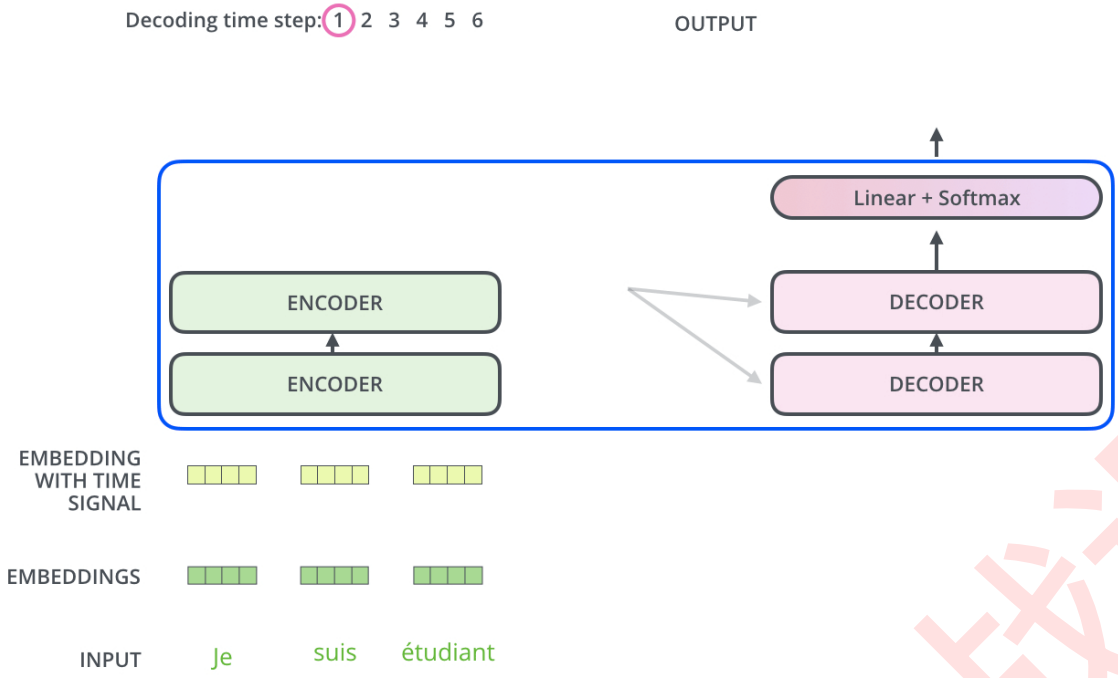




多头注意力机制



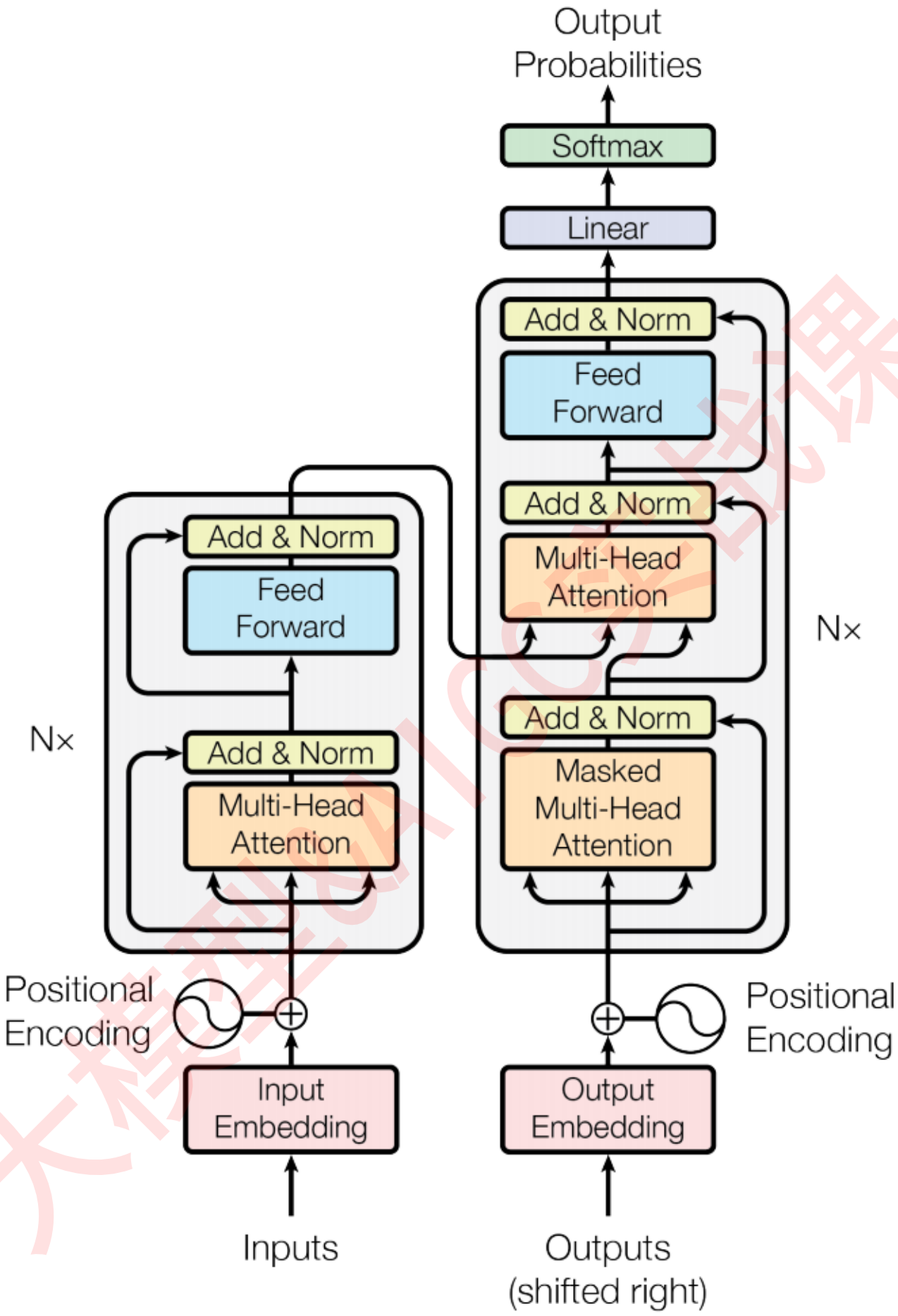
As we encode the word "it", one attention head is focusing most on "the animal", while another is focusing on "tired" -- in a sense, the model's representation of the word "it" bakes in some of the representation of both "animal" and "tired".

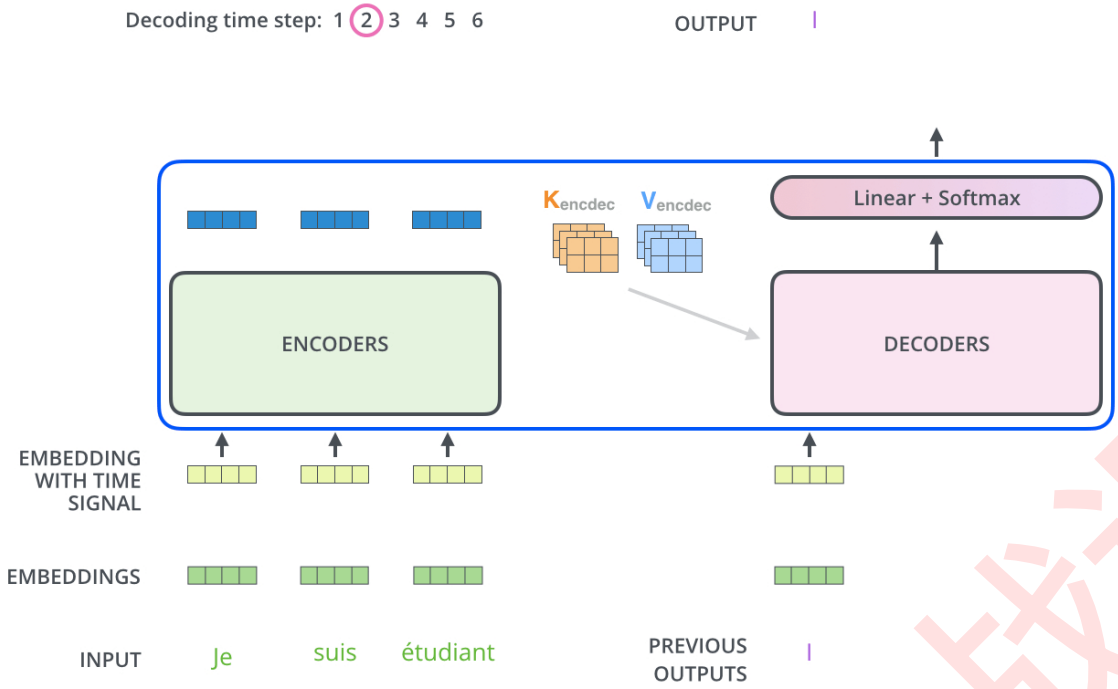


两个重点：

1、transformer 模型的核心机制：自注意力机制

2、Decoder 的生成式输出到底是什么意思，怎么做的

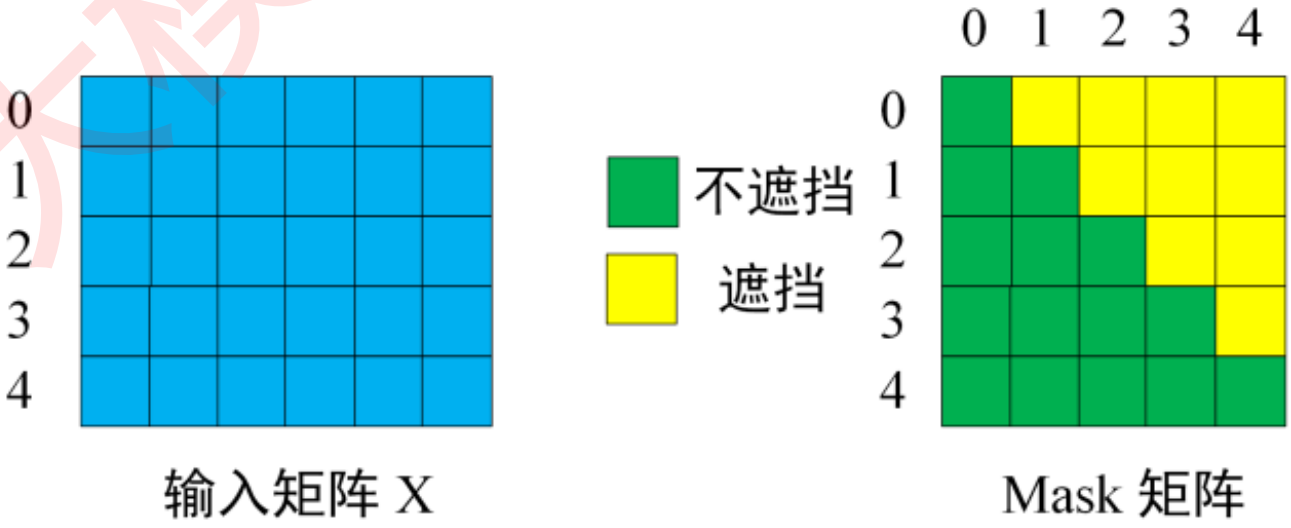

Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)



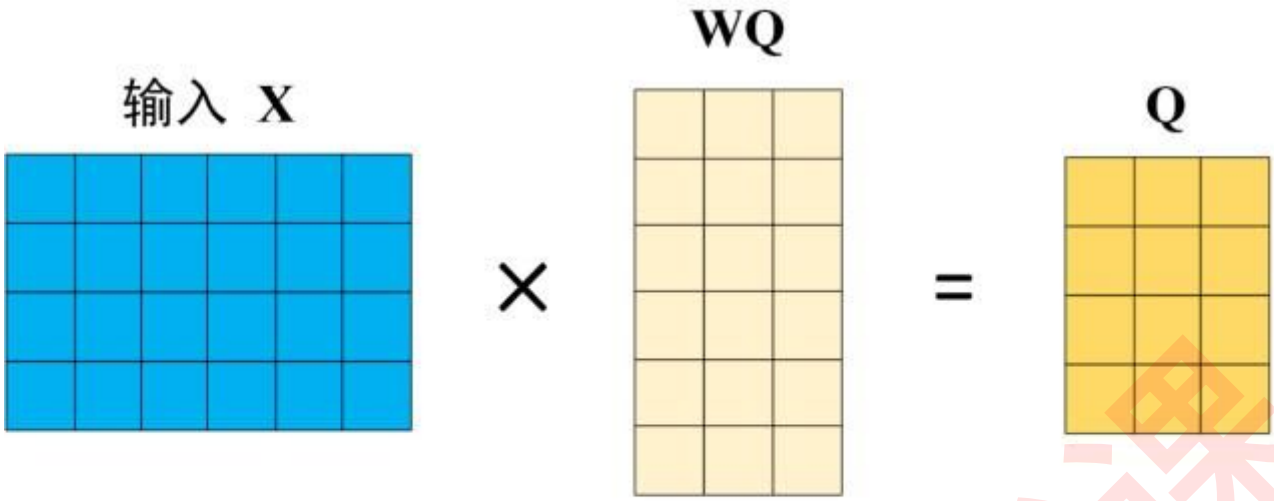
- Decoder的初始输入：训练集的标签Y，并且需要整体右移（Shifted Right）一位
  举例说明：`我是一个学生 → I am a student`
  0-“I”
  1-“am”
  2-“a”
  3-"student"
  操作：整体右移一位（Shifted Right）
  0-</s>【起始符】目的是为了预测下一个Token
  1-“I”
  2-“am”
  3-“a”
  4-"student"
- Shifted Right的原因：T-1时刻需要预测T时刻的输出，所以Decoder的输入需要整体后移一位

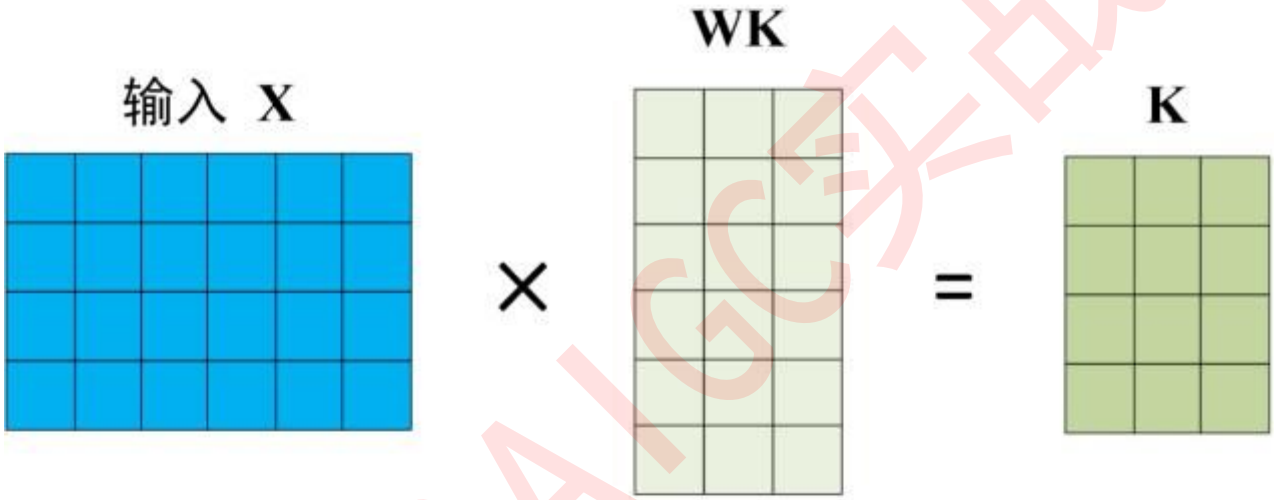
步骤一：Masked Attention

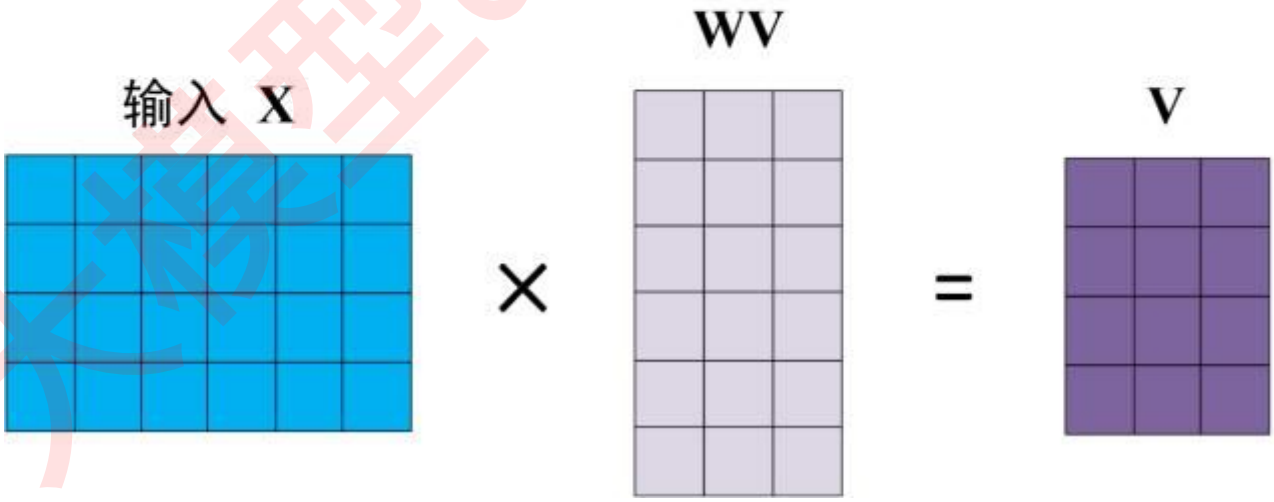
（1） 准备好输入的数据矩阵（此处的数据矩阵是由标签+起始符编码后形成的数据矩阵）和掩码矩阵

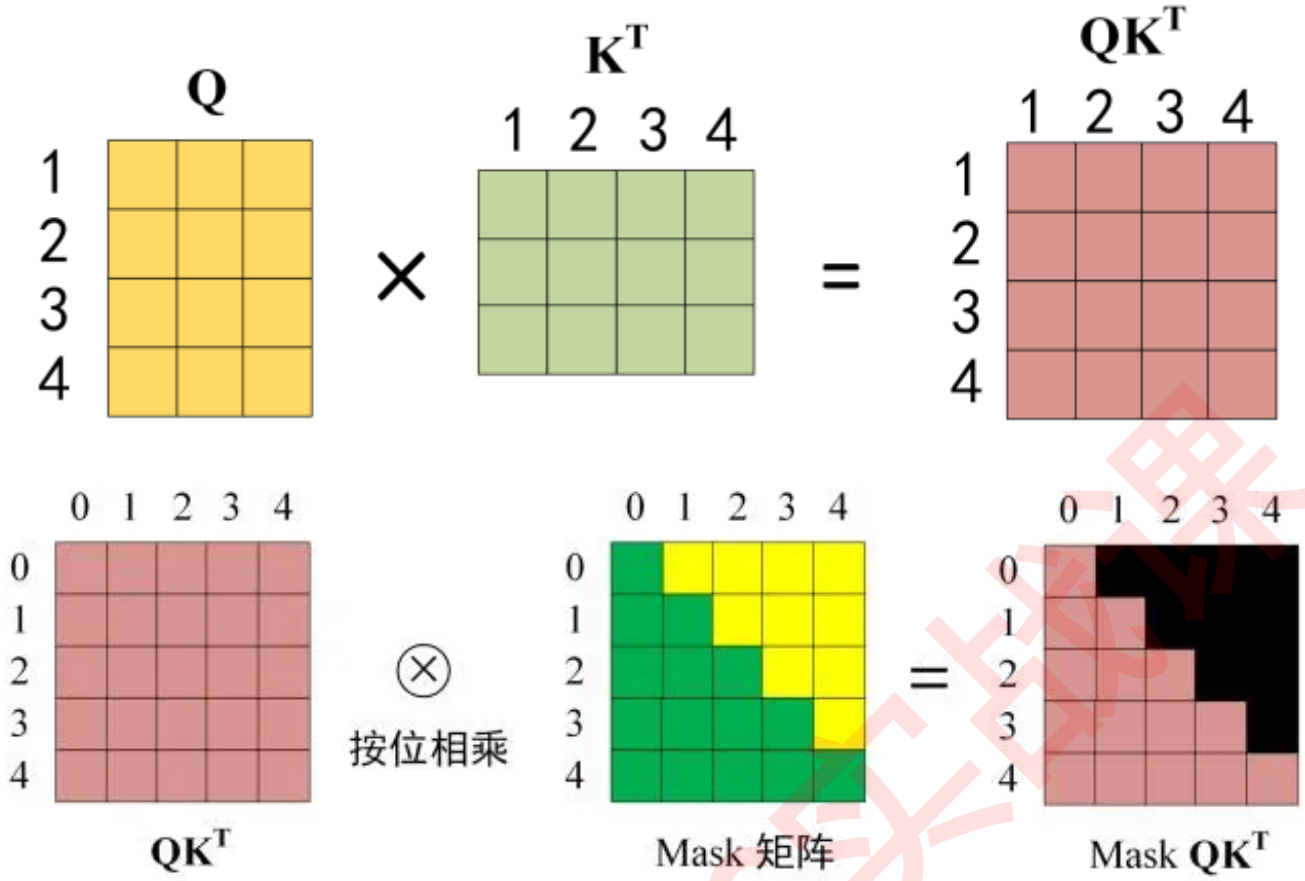


（2）计算自注意力机制需要的三元素：Q、K、V

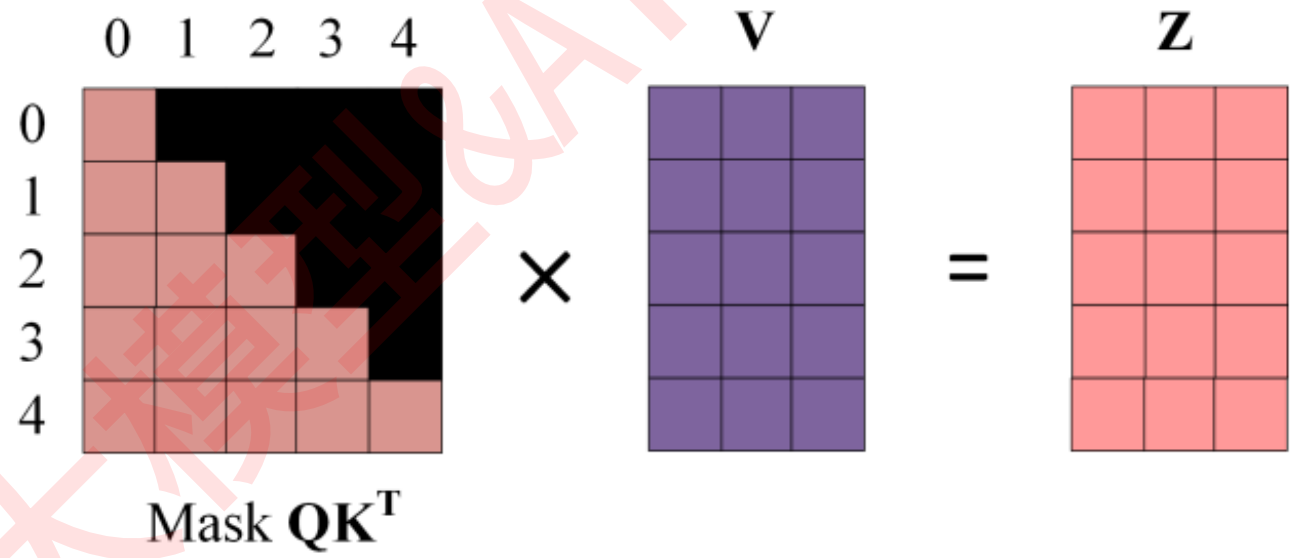






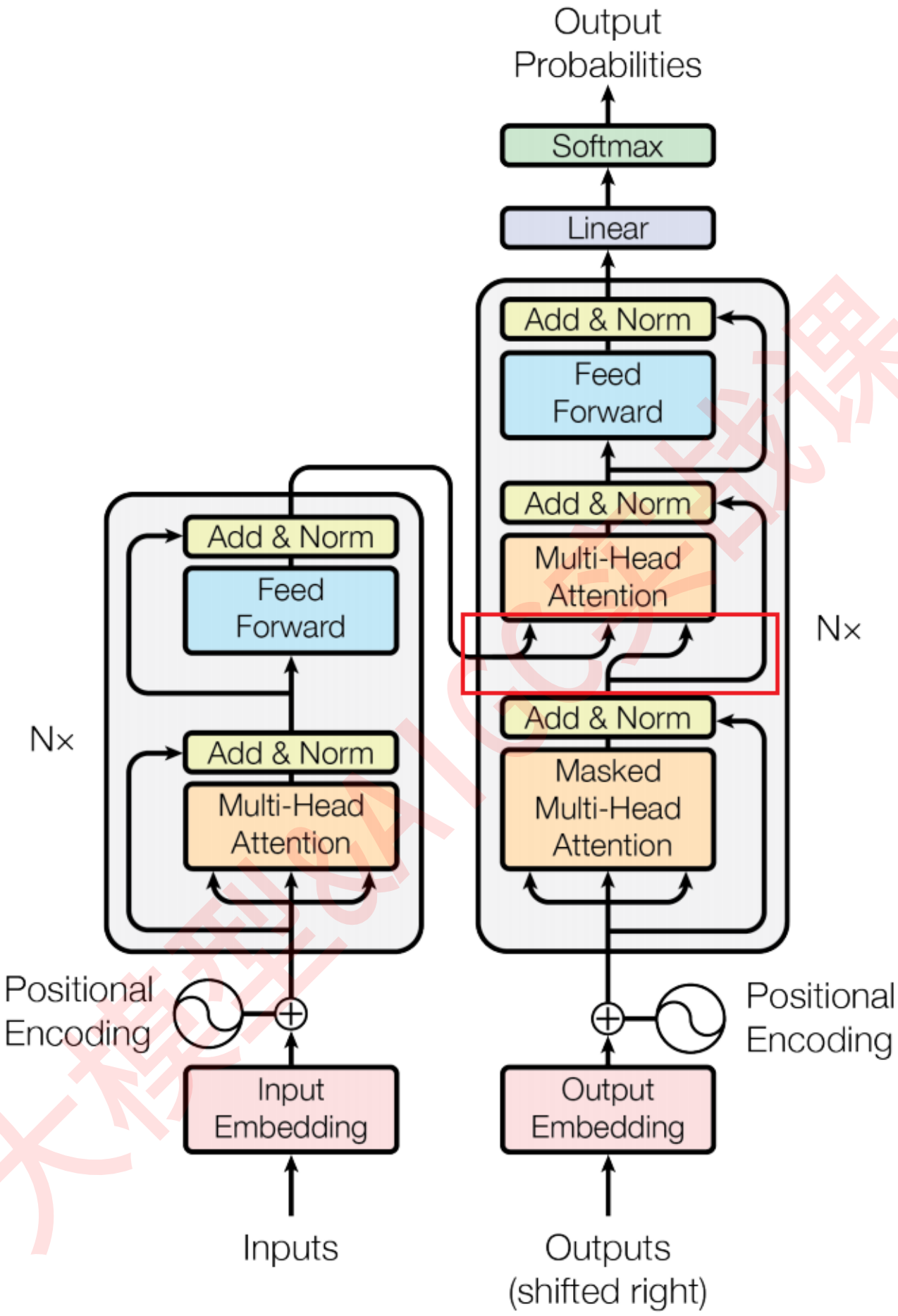


得到 **Mask** QK^T 之后在 **Mask** QK^T 上进行 Softmax，每一行的和都为 1。但是单词 0 在单词 1, 2, 3, 4 上的 attention score 都为 0



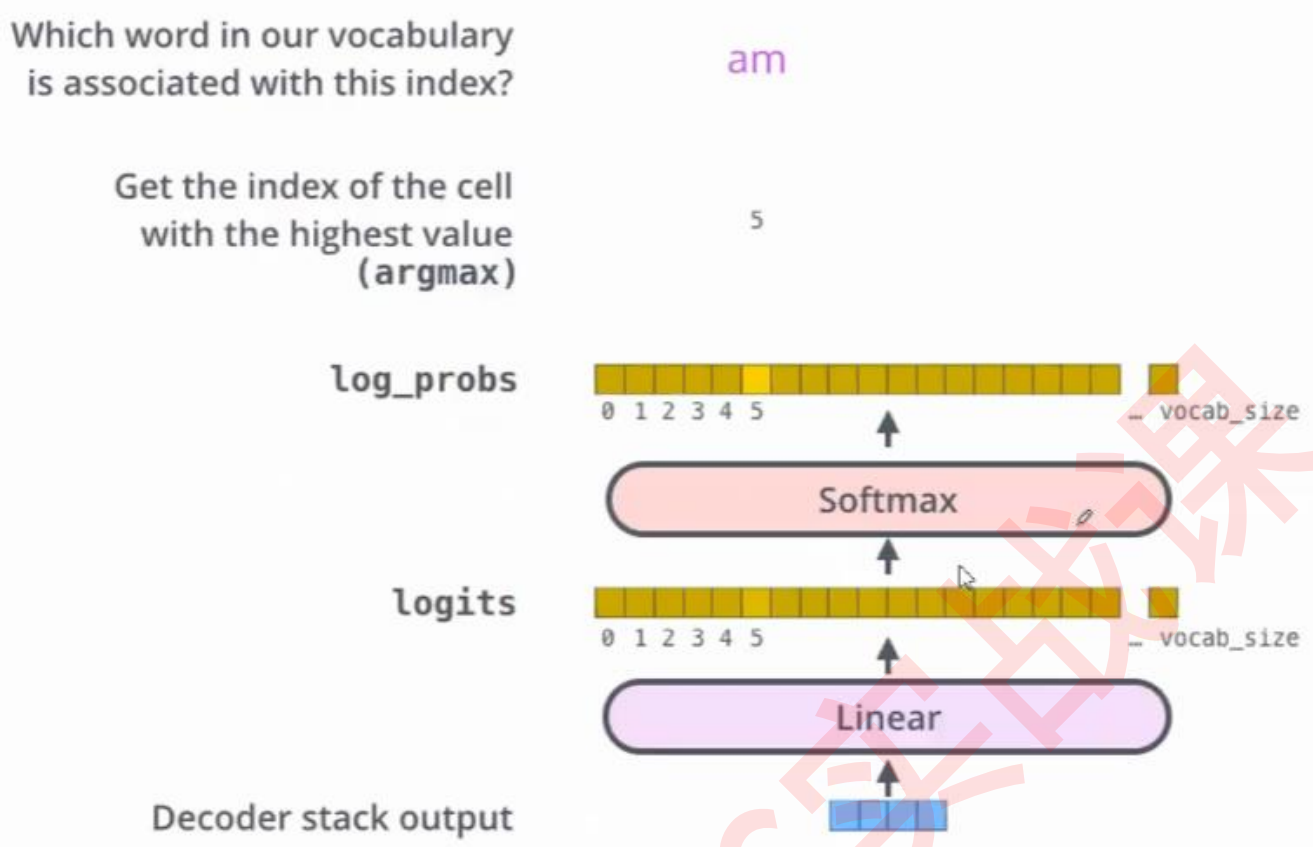
步骤 2：第二个Attention

第二个 Attention 的计算基本方式变化不大，但不再是单纯的自注意力机制，而是 Cross Attention(下图红色方框部分)



最终输出：Linear+softmax

Linear 的作用：将Decoder部分的输出映射为与词典相同维度的向量
softmax 的作用：将每一个词向量的概率进行输出，最终要输出的词就是概率最大对应的词。



GPT系列模型

GPT-1

模型	数据	框架	模型参数	发布日期
GPT-1	BooksCorpus；7000本未出版的书籍，约5GB	Transformer decoder	层数：12；维度：768；参数：1.17亿	2018/6

当时面临的问题：

- 常用的深度学习方法是通过大量的已标注数据对模型进行监督训练，这种方式使得模型的灵活性有限（有些领域中缺乏相关的有标签数据）
  已标注数据：指在机器学习和数据挖掘任务中，对于每个数据样本都标注了一个或多个正确的输出值或类别的数据集。自然语言领域中例如：命名实体识别、翻译
- 从无标签数据中获取信息非常困难，原因有两个：1）不确定哪种优化目标对于学习有用的文本表示最有效（意思是没有一个通用的、对所有自然语言任务都有效的目标函数）；2）没有关于如何将这些表示最有效地迁移到目标任务的共识

gpt1模型的训练方法：1）利用未标记的数据进行无监督预训练，主要为的是让语言模型学习到初始参数，这种方式下模型可以学习到一种根据给定的上文，继续续写下文的能力（为什么？）；2）使用相应的有标签数据进行微调，使模型能够满足不同任务的要求，这个过程需要的有标签数据就会少很多

续写下文的能力举例：

给定的文字：我是一个学生

模型续写：可能 1）当我遇到困难和挑战时，不要放弃和灰心，相信自己的能力和潜力，持之以恒地学习和努力，一定能够实现自己的梦想和目标。

可能 2）我不仅仅要关注学业成绩还要积极参与社会实践和公益活动，还要增强自身综合素质和社会责任感。

GPT-1模型的训练分为两步：

(1) 无监督的预训练

第一阶段是在大型文本语料库上学习大容量语言信息。

首先给定一个无监督语言序列$U = \{\mu_1 \ldots . \mu_n\}$

使用标准语言模型**目标函数**：

标准语言模型：用于预测自然语言序列中词汇概率分布；目标是给定上文，然后对下一个词进行概率预测。

$$L_1(U) = \sum_i logP(\mu_i|\mu_{i-k}, \ldots . . \mu_{i-1}; \Theta), i = 1, 2, \ldots . n$$

k 表示序列窗口（k 个词）

Θ 表示所使用的模型（多层transformer Decoder）

$$h_0 = UW_e + W_p$$
$$h_l = transformerbloc(h_{l-1} \forall i \in [1, n])$$
$$P(\mu) = softmax(h_n W_e^T)$$

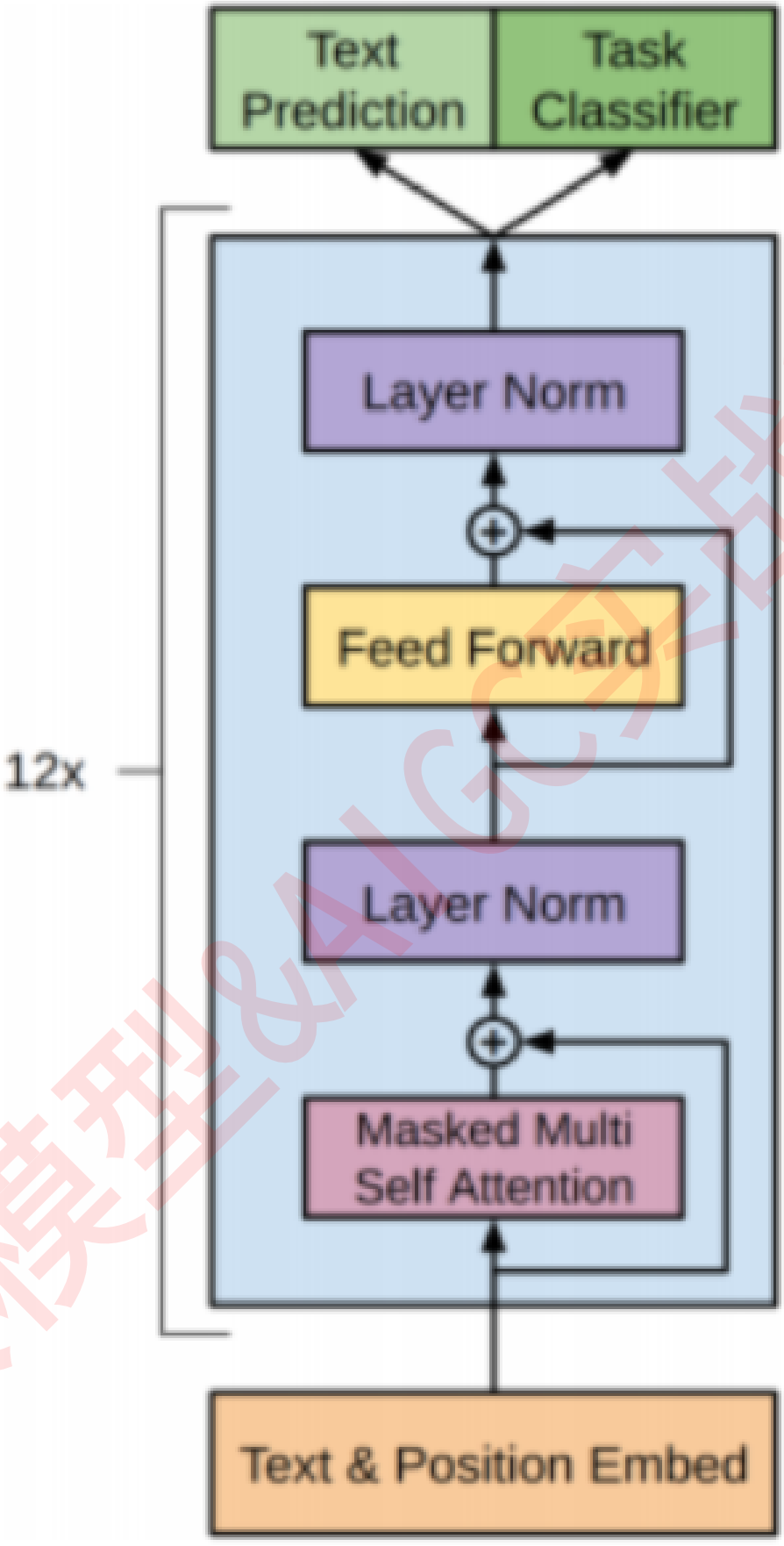
其中$U = (\mu_{-k}, \ldots . \mu_{-1})$ 是输入词向量token

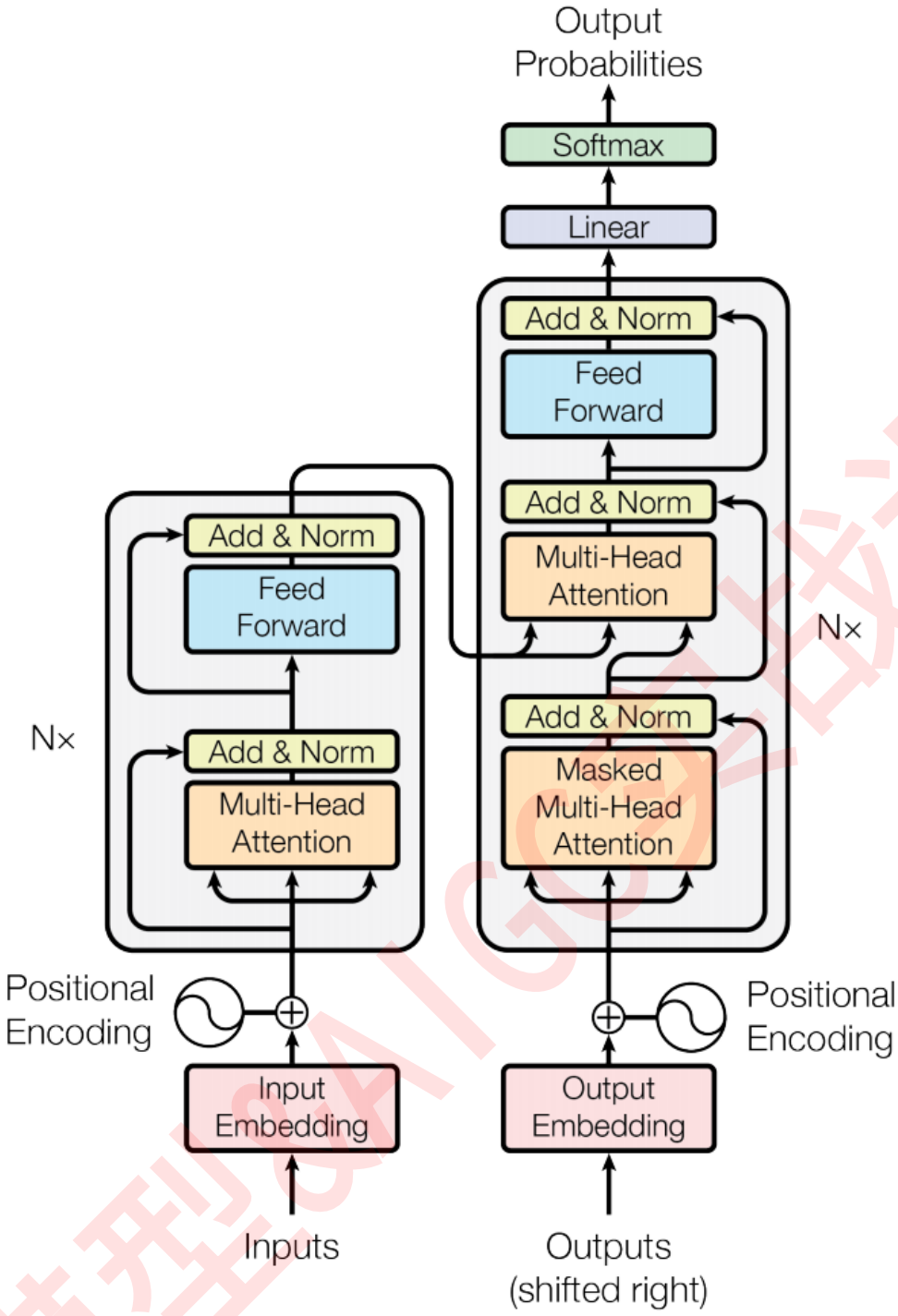
n 是模型的层数；

W_e 是embediding矩阵；W_p 是位置矩阵

GPT 具体结构：

GPT只使用了 Transformer 的 Decoder 部分，并且每个子层只有一个 Masked Multi Self-Attention（768 维向量和 12 个 Attention Head）和一个 Feed Forward，共叠加使用了 12 层的 Decoder。


Text Prediction
Task Classifier
Layer Norm
Feed Forward
Layer Norm
Masked Multi Self Attention
12x
Text & Position Embed



(2) 有监督的模型微调

在经过第一阶段的无监督，训练后，进入微调阶段，利用有标签数据使模型适应特定的任务

具体做法：

假定有标签的数据C：

$x^1, x^2, \ldots\ldots x^m$ 为输入需要列，y 为标签

将数据输入已经预训练好的模型中预测y 的概率：

$$P(y|x^1, x^2, \ldots\ldots x^m) = softmax(h_l^m W_y)$$

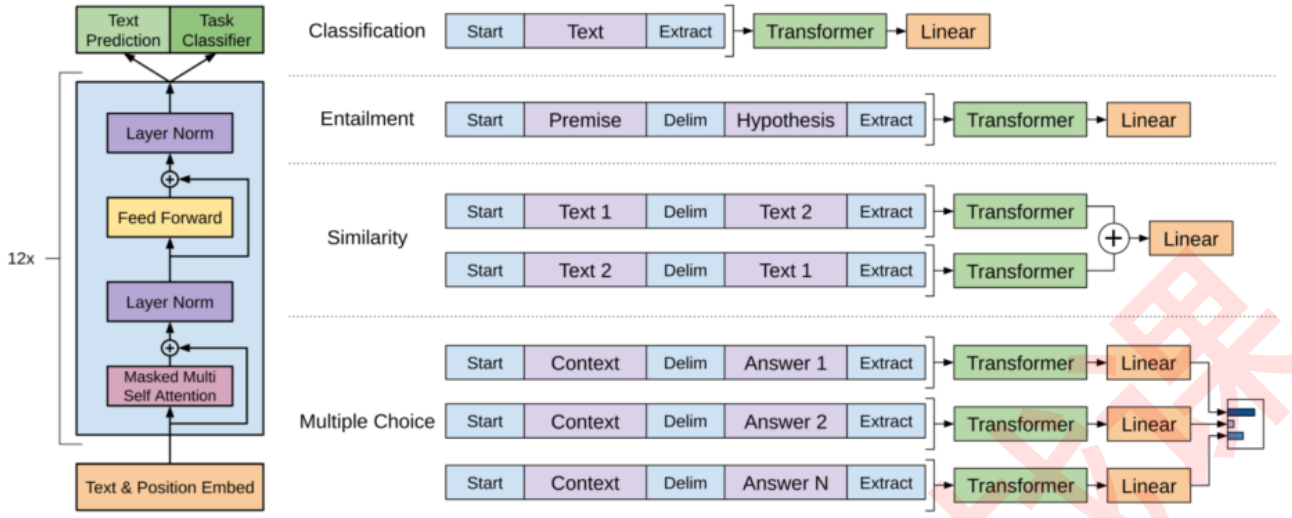
最大化目标函数：

$$L_2(C) = \sum_{(x,y)} logP(y|x^1, x^2, \ldots\ldots x^m)$$

同时还加入了一个**辅助目标**，也即同时将无监督阶段和有监督阶段的目标函数结合起来：

$$L_3(C) = L_2(C) + \lambda * L_1(C)$$

微调阶段需要调的参数就只有W_y 以及分隔符标记



（左）预训练结构；（右）不同下游任务的有监督微调

（3）gpt1的效果

gpt1在12个数据集上进行效果评估，在其中9个数据集上都达到了新的SOTA效果

Table 2: Experimental results on natural language inference tasks, comparing our model with current state-of-the-art methods. 5x indicates an ensemble of 5 models. All datasets use accuracy as the evaluation metric.

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

Table 3: Results on question answering and commonsense reasoning, comparing our model with current state-of-the-art methods.. 9x means an ensemble of 9 models.

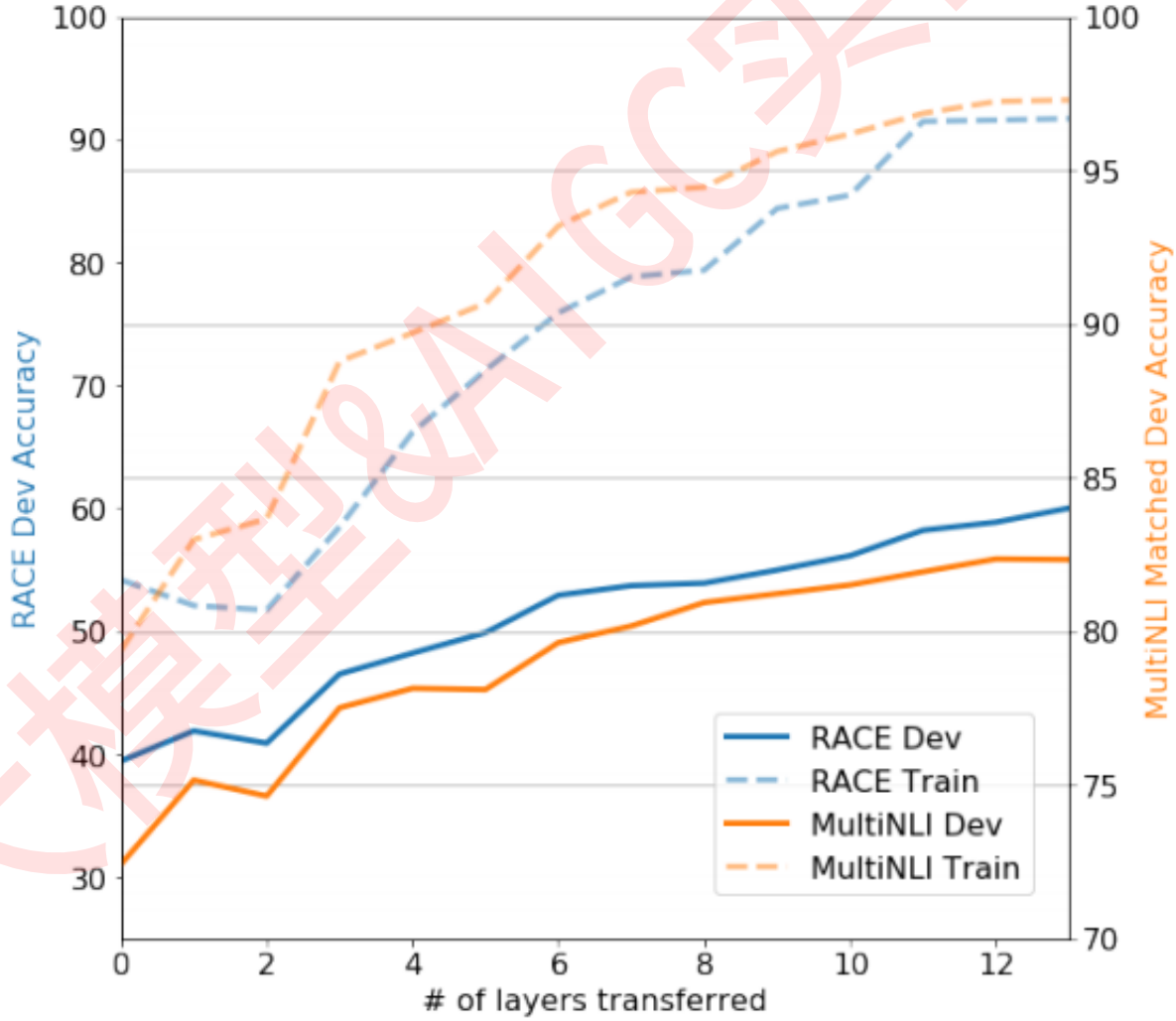
Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	77.6	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	60.2	50.3	53.3
Finetuned Transformer LM (ours)	**86.5**	**62.9**	**57.4**	**59.0**

Table 4: Semantic similarity and classification results, comparing our model with current state-of-the-art methods. All task evaluations in this table were done using the GLUE benchmark. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

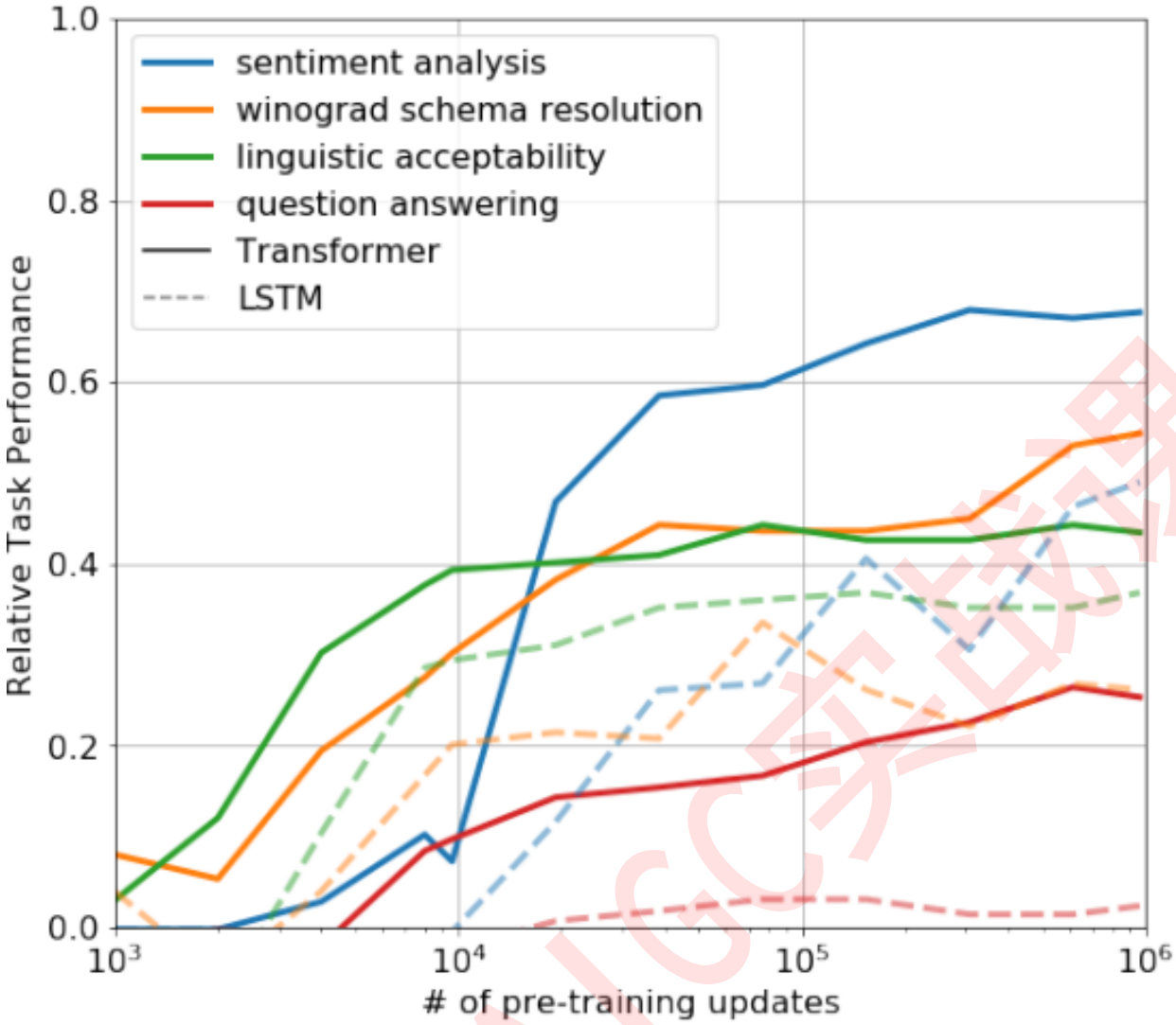
Method	Classification		Semantic Similarity			GLUE
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	68.9
Finetuned Transformer LM (ours)	**45.4**	91.3	82.3	**82.0**	**70.3**	**72.8**

GPT-1值的关注点

（1）随着模型增大，模型精度和泛化能力还有提升空间



（2）zero-shot



Bert

模型	框架	模型参数	发布日期
BERT_base	Transformer encoder	层数：12；维度：768；多头数：12；参数：1.1亿	2018/10
BERT_large	Transformer encoder	层数：24；维度：1024；多头数：16；参数：3.4亿	

BERT_large使用数据：BooksCorpus（800M words）；
English Wikipedia（2500M words）
约为gpt1的4倍

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
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
$BERT_{BASE}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$BERT_{LARGE}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

gpt2

模型	数据	框架	模型参数	发布日期
GPT-1	BooksCorpus：7000本未出版的书籍，约5GB	Transformer decoder	层数：12；维度：768；参数：1.17亿	2018/6
GPT-2	WebText:清洗过的Reddit数据，约 40 GB	同 GPT-1,进行了少量改动	层数：48；维度：1600；参数：15亿	2019/2

gpt2的改进方向：

更大的数据集，更大的模型

模型结构变化：

（1）后置层归一化（ post-norm ）改为前置层归一化（ pre-norm ）；

（2）在模型最后一个自注意力层之后，额外增加一个层归一化;

（3）调整参数的初始化方式，按残差层个数进行缩放，缩放比例为 1：$\sqrt{n}$；

（4）输入序列的最大长度从 512 扩充到 1024;

zero-shot

当时研究的现状：针对某一个任务，需要该任务专门的数据集，训练出一个专门的模型来应对该任务，主要原因是当前系统缺乏泛化性.

一个解决的办法—多任务学习：多任务学习通过训练一个模型来同时执行多个相关任务。相比于传统的单一任务学习，多存在的问题：需要一个模型处理多种任务的数据集；可能需要设置多个损失函数；在NLP领域应用并不广泛

GPT-1所使用的无监督预训练+有监督的微调也还是没能避免这个问题。所以GPT-2在训练之初就提出了zero-shot的概念：当训练完成一个模型后，无需再对模型的参数或者架构进行修改，就能执行各项任务并获得好的成绩。

如何做到zero-shot?

zero-shot想要达到的一个效果是：在构建下游任务的时候，不再使用gpt1时那样的开始符、分割符以及抽取符的形式告诉模型要执行任务了，而是通过自然语言的方式来指定任务，比如要执行一个翻译任务，就可以给模型这样输入：将英语翻译为中文，英语内容。通过这样的方式来提示（promot）模型应该做什么。

Zero-shot

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

为了达到这样的效果，OpenAI的理念就是数据足够多足够好，模型足够大足够强，就可以去掉fine-tune这一步骤，得到一个通用的模型。数据集方面，不在采用以前采用在单一文本领域训练模型的方式，而是尽可能构建更大和更多样化的数据。gpt2训练用的数据集：gpt2中采用的方法是从社交平台Reddit上抓取这个数据集中被给了至少3个Karma的内容。通过这样的方式，获得了总共40GB的数据。因为**一段普通的文字里，可能已经蕴含了“任务描述”、“任务提示”和“答案”这些关键信息**。比如，我想做英法文翻译这件事，那么我从网上爬取的资料里可能有这样的内容：

”I’m not the cleverest man in the world, but like they say in French: Je ne suis pas un imbecile [I’m not a fool] .

””I hate the word ‘perfume,”’ Burr says. ‘It’s somewhat better in French: ‘parfum .

如果我把这样的文本喂给GPT，它是不是就能在学习文字接龙的过程里，领悟到英法互译这一点？如果我的数据集又多又棒，那GPT自主揣摩的能力是不是就能更强？在这个方式下，训练出了1.5B的GPT2，效果基本与Bert差不多。从实用性的角度上，GPT2并没有带来突破，但是，zero-shot的训练方式，却有效证明了NLP领域训练出一个完全通用模型的可行性，这一刻开始，GPT模型AGI的可能性初见萌芽，因为整个训练流程看起来就像是模型自主学习知识。

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

Table 2. Architecture hyperparameters for the 4 model sizes.

Language Models are Unsupervised Multitask Learners

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

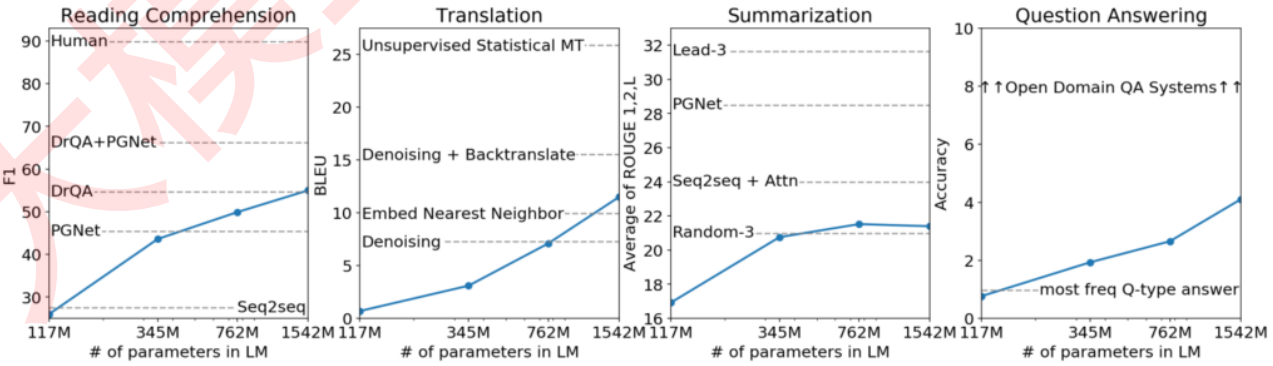


Figure 1. Zero-shot task performance of WebText LMs as a function of model size on many NLP tasks. Reading Comprehension results are on CoQA (Reddy et al., 2018), translation on WMT-14 Fr-En (Artetxe et al., 2017), summarization on CNN and Daily Mail (See et al., 2017), and Question Answering on Natural Questions (Kwiatkowski et al., 2019). Section 3 contains detailed descriptions of each result.

gpt2虽然提出了一些创新，但是其论文中展现出来的效果确是不尽如人意，并没有表现出什么令人惊艳的效果。

gpt3

模型	数据	框架	模型参数	发布日期
GPT-1	BooksCorpus：7000本未出版的书籍，约5GB	Transformer decoder	层数：12；维度：768；参数：1.17亿	2018/6
GPT-2	WebText:清洗过的Reddit数据，约40GB	同 GPT-1,进行了少量改动	层数：48；维度：1600；参数：15亿	2019/2
GPT-3	Common Books1,Books2 and Wikipedia,Crawl,WebText2, 约 45 TB	同 GPT-2，做了部分改动	层数：96；维度：12888；参数：1750亿	2020/6

模型结构变动：引入了 sparse transformer

gpt3的特点：
模型很大，有1750亿个参数
gpt3在执行所有任务时，都不会再更新参数或者微调，所有的任务提示都过文本和模型进行交互完成
gpt3在各种NLP任务数据集上都表现出了最好的成绩（翻译、问答、完形填空等等）
能生成人类都难以区分的新闻文章

三个问题：

- 1、任然是数据问题，每个不同的任务都需要大量的已标记的数据，限制了语言模型的适用性。
- 2、微调后的模型泛化性不一定就能更好
- 3、人类在学习的时候仅需要少量的示例，而不是需要大量的已标注数据。

那么，GPT-3 都做了些什么呢?
为了使模型能够解决上述的问题，论文中重点方法如下：

在上述情景提示中，通常分为zero-shot、one-shot以及few-shot

The three settings we explore for 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        ←
```

给模型输入情景的描述，模型会依据用户提出的情景要求来生成答案，这部分被成为in-context learning

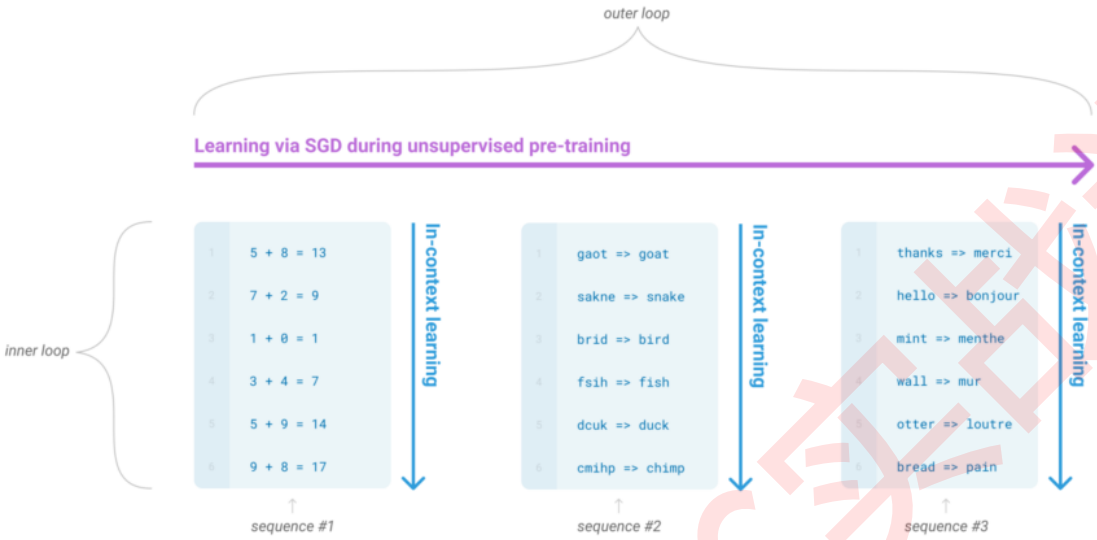


Figure 1.1: Language model meta-learning. During unsupervised pre-training, a language model develops a broad set of skills and pattern recognition abilities. It then uses these abilities at inference time to rapidly adapt to or recognize the desired task. We use the term "in-context learning" to describe the inner loop of this process, which occurs within the forward-pass upon each sequence. The sequences in this diagram are not intended to be representative of the data a model would see during pre-training, but are intended to show that there are sometimes repeated sub-tasks embedded within a single sequence.

那么上述提出的解决方案最终效果如何呢？可以从下面这副图中看出：

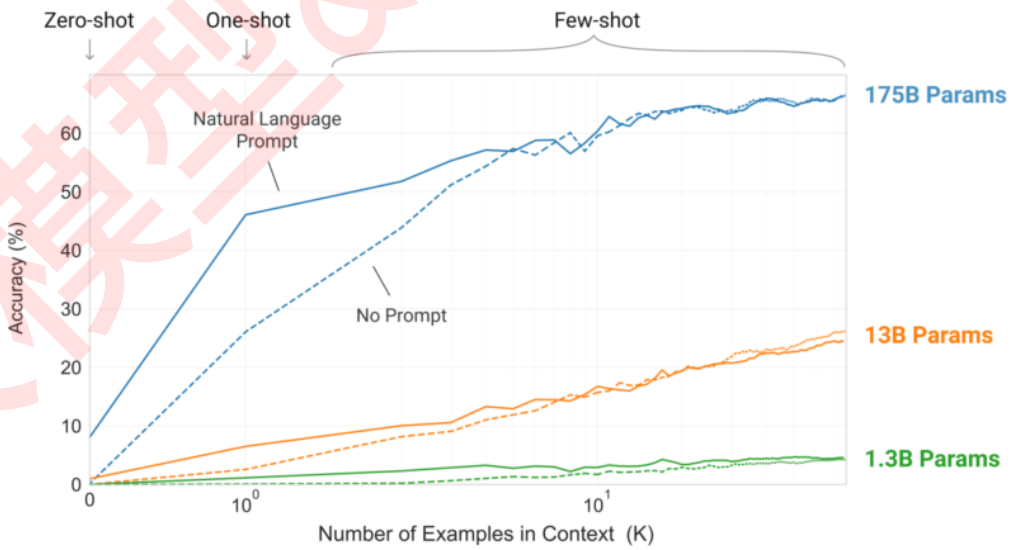


Figure 1.2: Larger models make increasingly efficient use of in-context information. We show in-context learning performance on a simple task requiring the model to remove random symbols from a word, both with and without a natural language task description (see Sec. 3.9.2). The steeper "in-context learning curves" for large models demonstrate improved ability to learn a task from contextual information. We see qualitatively similar behavior across a wide range of tasks.

总的来看，在NLP任务中，gpt3通在zero-shot、one-shot以及few-shot中都取得了非常亮眼的成绩。

chatgpt

不断的将模型扩大规模，虽然使其在各种任务展现出了出色的表现，但是在本质上并不能使模型很好的遵循用户的意图。例如大语言模型可能会生成不真实的、不符合人类道德标准和社会规范的（有毒的）以及对用户没有用的答案，总的来说就是没法和人类进行沟通。为什么会出现输出的答案不是人类想要的答案这种现象呢？一种解释是模型训练时目标函数设定的问题，大语言模型在训练时是以让模型生成文本下一个词为目标的，与我们希望根据指示来生成高质量的优秀答案的目标是有差别的。

那么，如何将模型与人类链接起来呢？chatgpt背后采用了这样一种技术：生物反馈强化学习（reinforcement learning from human feedback，RLHF）。具体做法是首先通过提交在OpenAI API上问题标注数据并写出答案对模型进行微调，然后又收集了模型对一个问题的不同答案的排序数据集，并人工对其答案的好坏进行排序，然后用强化学习模型对其进行学习。

ChatGPT的训练

ChatGPT的训练过程分为以下三个阶段：

第一阶段：训练监督策略模型

GPT 3本身很难理解人类不同类型指令中蕴含的不同意图，也很难判断生成内容是否是高质量的结果。为了让 GPT 3初步具备理解指令的意图，首先会在数据集中随机抽取问题，由人类标注人员，给出高质量答案，然后用这些人工标注好的数据来微调 GPT-3模型（获得SFT模型, Supervised Fine-Tuning）。

此时的SFT模型在遵循指令/对话方面已经优于 GPT-3，但不一定符合人类偏好。

第二阶段：训练奖励模型（Reward Mode，RM）

这个阶段的主要是通过人工标注训练数据（约33K个数据），来训练回报模型。在数据集中随机抽取问题，使用第一阶段生成的模型，对于每个问题，生成多个不同的回答。人类标注者对这些结果综合考虑给出排名顺序。这一过程类似于教练或老师辅导。

接下来，使用这个排序结果数据来训练奖励模型。对多个排序结果，两两组合，形成多个训练数据对。RM模型接受一个输入，给出评价回答质量的分数。这样，对于一对训练数据，调节参数使得高质量回答的打分比低质量的打分要高。

第三阶段：采用PPO（Proximal Policy Optimization，近端策略优化）强化学习来优化策略。

PPO的核心思路在于将Policy Gradient中On-policy的训练过程转化为Off-policy，即将在线学习转化为离线学习，这个转化过程被称之为Importance Sampling。这一阶段利用第二阶段训练好的奖励模型，靠奖励打分来更新预训练模型参数。在数据集中随机抽取问题，使用PPO模型生成回答，并用上一阶段训练好的RM模型给出质量分数。把回报分数依次传递，由此产生策略梯度，通过强化学习的方式以更新PPO模型参数。

如果我们不断重复第二和第三阶段，通过迭代，会训练出更高质量的ChatGPT模型。

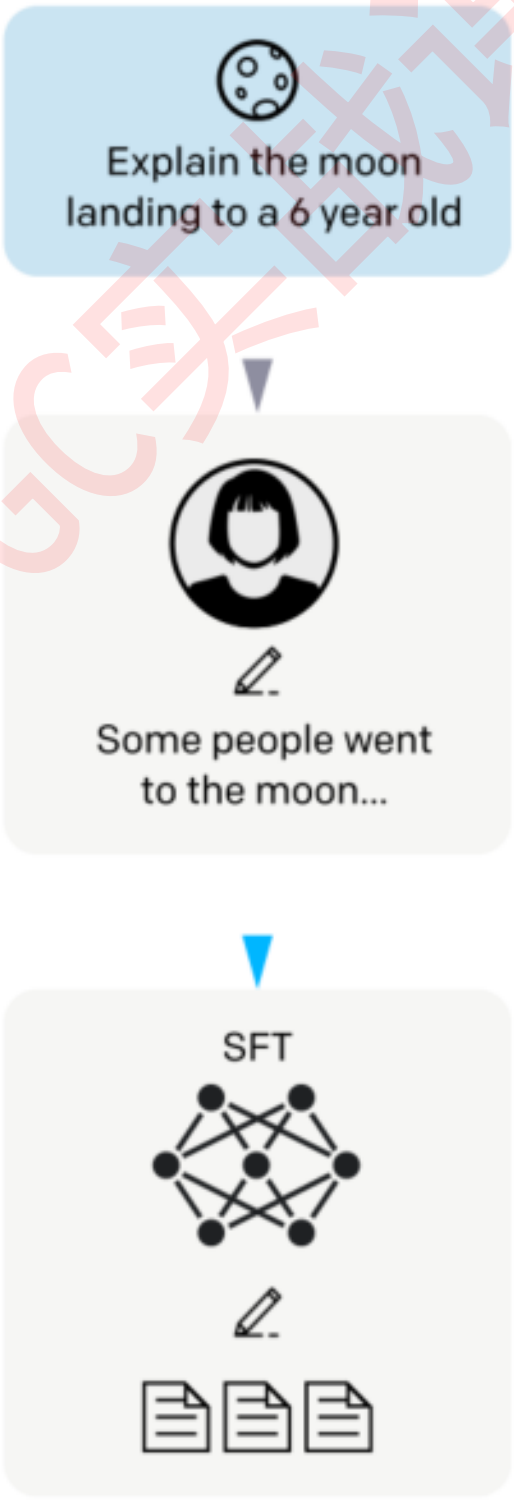
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

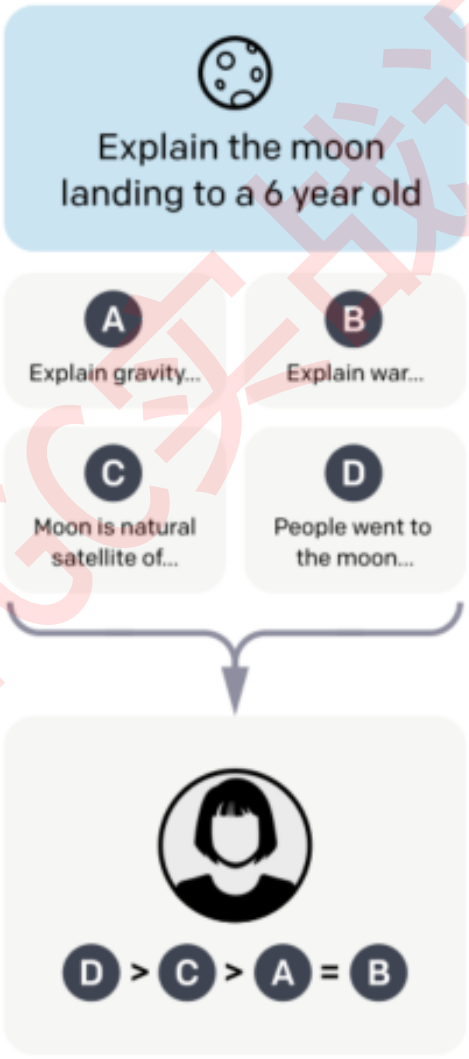
This data is used to fine-tune GPT-3 with supervised learning.



Step 2

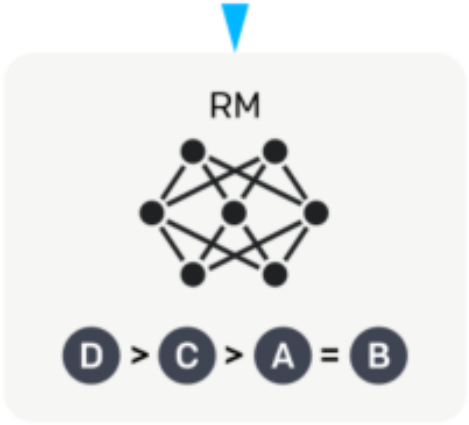
Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

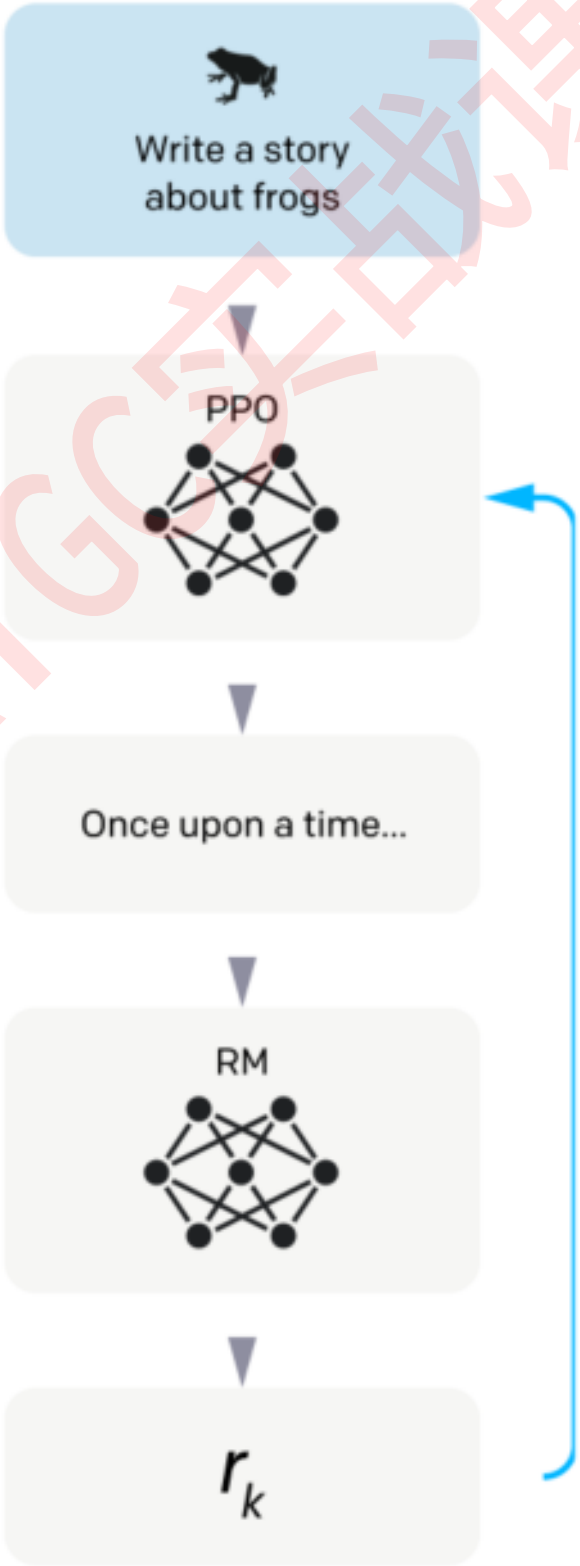
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy



using PPO.

ChatGPT 还具有以下特征：

1）可以主动承认自身错误。若用户指出其错误，模型会听取意见并优化答案。

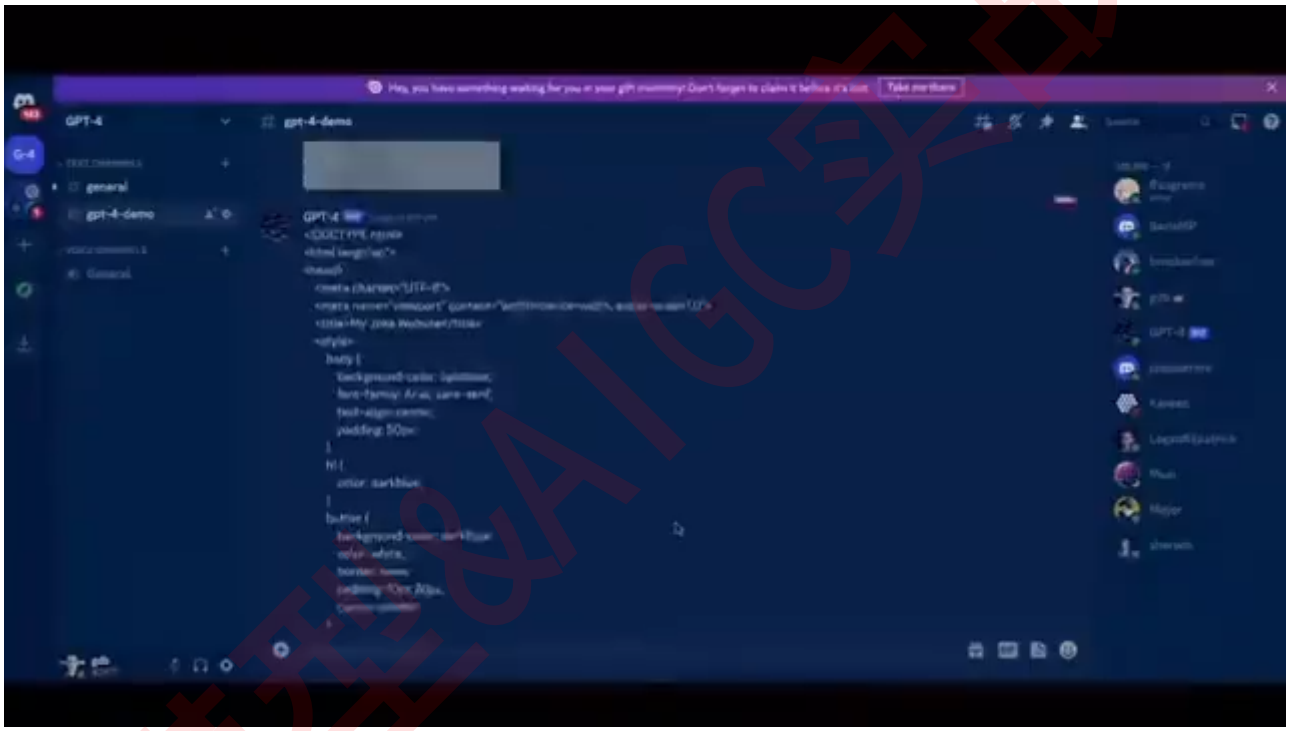
2）ChatGPT 可以质疑不正确的问题。例如被询问 “哥伦布 2015 年来到美国的情景” 的问题时，机器人会说明哥伦布不属于这一时代并调整输出结果。

3）ChatGPT 可以承认自身的无知，承认对专业技术的不了解。

4）支持连续多轮对话

gpt4

GPT-4 是一个大型多模态模型（接受图像和文本输入，发出文本输出），虽然在许多现实世界场景中的能力不如人类，但在各种专业和学术基准上表现出人类水平的表现。



- 多模态模型
  人类或其他高等生物的认知能力通常与从多种模式中学习有关。例如，苹果这一概念包括从视觉和语言获得的多重语义。包括苹果的颜色、形状、纹理以及吃苹果的声音，苹果在词典或其他网络媒体的相应定义等等。我们大多数人在学习认字的时候，也是先看到苹果的卡片图像，然后再记住对应的文字。

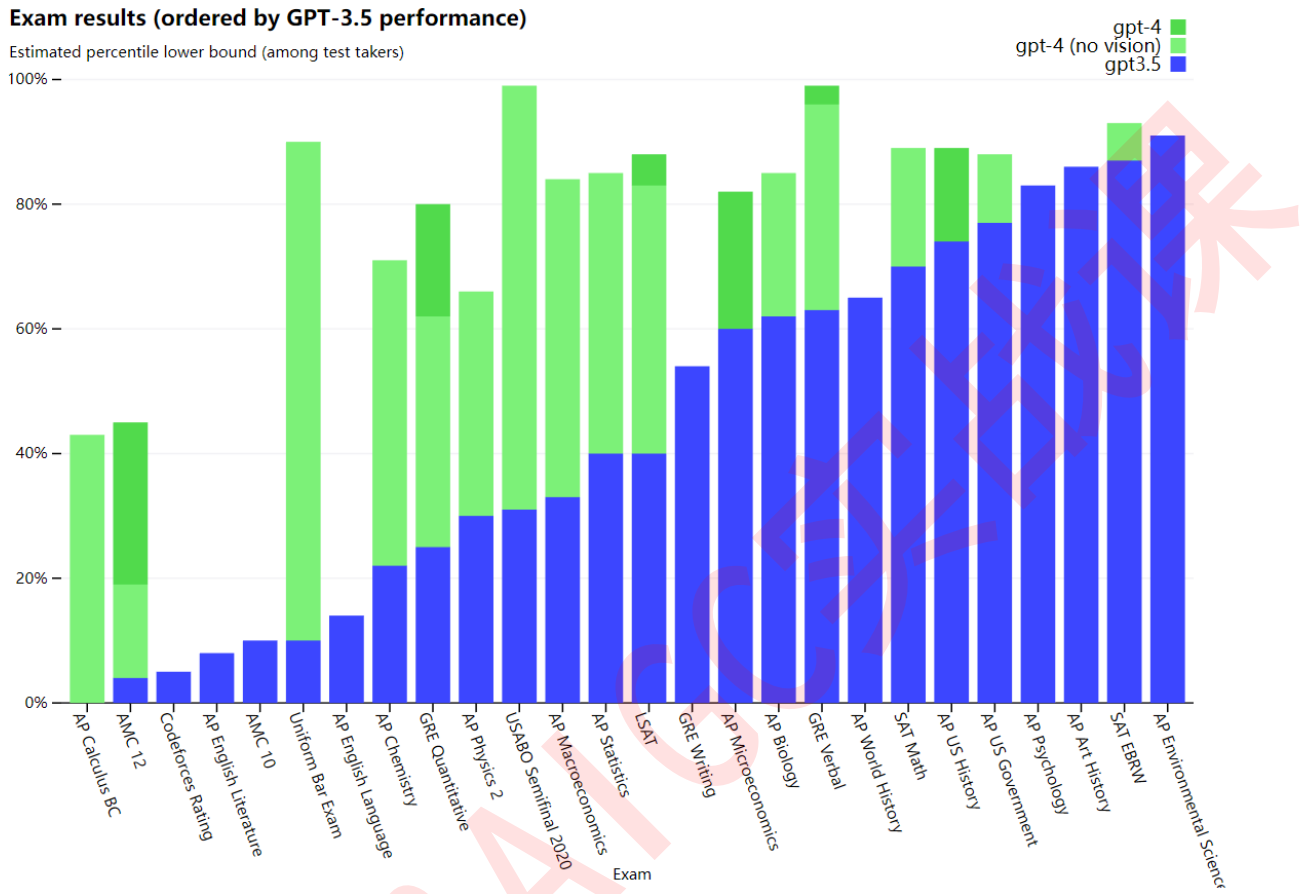
gpt4之前，gpt系列模型都是语言模型，是通过上文对下一个可能的词进行预测，所有输入（例如苹果）只是**单纯的语义符号和概率。**

GPT-4等模型新出现的多模态输入的能力对语言模型至关重要，使得“苹果”等单纯的符号语义**扩展为更多的内涵。**

- 第一，多模态感知使语言模型能够获得文本描述之外的常识性知识。
- 第二，多模态感知与语义理解的结合为新型任务提供了可能性，例如机器人交互技术和多媒体文档处理等等，仅列出的这两项应用就市场巨大。
- 第三，通过多模态感知统一了接口。图形界面其实是最自然和高效的人机自然交互方式。多模态大语言模型可通过图形方式直接进行信息交互，提升交互效率和模式融合。

- 专业和学术水平
  在随意的谈话中，GPT-3.5 和 GPT-4 之间的区别可能很微妙。当任务的复杂性达到足够的阈值时，差异就会出现——GPT-4 比 GPT-3.5 更可靠、更有创意，并且能够处理更细微的指令。

各种基准测试测试，包括最初为人类设计的模拟考试中，通过使用最新的公开测试，gpt4在大部分考试中都得到了相当高的分数。



大语言模型为什么能有这样优异的性能，它的优势在什么地方呢？
涌现能力和思维链，这两者是大语言模型不断接近人类的**关键特征。**

涌现能力（Emergent Abilities）是指模型具有从原始训练数据中自动学习并发现新的、更高层次的特征和模式的能力。就中文释义而言，涌现能力也指大语言模型涌现出来的新能力。

涌现能力的另一个重要表现是多模态模型的**泛化能力**。在没有专门训练过的情况，GPT-4也可以泛化到新的、未知的多模态数据样本上。这种泛化能力主要取决于模型的结构和训练过程，以及数据的数量和多样性。如果模型具有足够的复杂性和泛化能力，就可以从原始数据中发现新的、未知的特征和模式。

多模态大语言模型（Multi-modal Large Language Model，MLLM）可实现更好的常识推理性能，跨模态迁移更有利于知识获取，产生更多新的能力，加速了能力的涌现。这些独立模态或跨模态新特征、能力或模式通常不是通过**目的明确**的编程或训练获得的，而是模型在大量多模态数据中**自然而然**的学习到的。

思维链（Chain of Thought）可视为大语言模型涌现出来的核心能力之一。思维链是ChatGPT和GPT-4能让大众感觉到语言模型“像人”的**关键特性。**

通过多模态思维链技术，GPT-4将一个多步骤的问题（例如图表推理）**分解为可以单独解决的中间步骤。**在解决多步骤推理问题时，模型生成的思维链会模仿人类思维过程。

虽然GPT-4这些模型并非具备真正的意识或思考能力，但用类似于人的推理方式的思维链来提示语言模型，极大的提高了GPT-4在推理任务上的表现，打破了精调（Fine-tune）的平坦曲线。具备了多模态思维

链能力的GPT-4模型具有一定逻辑分析能力，**已经不是**传统意义上的词汇概率逼近模型。

GPT-5技术发展前瞻

通过上述对GPT系列模型的介绍当中我们可以看到，GPT3用到的数据已经达到45TB，包含数十亿甚至数千亿的文本，GPT4虽然没有具体的说明所用的数据的容量大小，但是推测下来应该不会比GPT3的数据容量小，那么未来对GPT5的训练所用数据，是继续无限制的扩大吗？其实在数据已经如此大的规模的情况下，更应该考虑的是如何更高效率的使用数据，从前面ChatGPT的讲解中可以看到，用某种方式让模型更人的交互，可以在很大程度上降低数据规模的成本。那么在GPT5的一个发展方向很有可能是在当前数据规模下考虑怎么更加高效的使用数据。

对此有这样一种猜想，认为GPT-5大概率会采用model-based深度强化学习进行自我提升，因为GPT本身是一个Policy，也是一个World Model，效仿AlphaGo和AlphaStar，GPT5通过这项技术也可以无限提升自己的能力。

World model（世界模型）是由DeepMind的研究团队提出的一个概念，用于描述强化学习中的一种方法。它是指在强化学习中使用一个包含三个主要组件的模型，用于学习和模拟环境的动态过程，从而辅助智能体进行规划和决策。

World model的三个主要组件包括：

1. Vision model（视觉模型）：这是一个用于学习和处理环境的视觉输入的深度神经网络。它可以接收环境的图像或像素数据，并将其转换为有意义的状态表示，以供后续的学习和决策使用。
2. Memory model（记忆模型）：用于学习和模拟环境的动态转移。它将前一时刻的状态和行动作为输入，并输出下一时刻的状态，从而模拟环境中状态的演变。
3. Controller（控制器）：这是一个强化学习智能体，它利用视觉模型和记忆模型来构建对环境进行规划和决策的能力。它可以通过与环境交互来收集经验数据，并利用这些数据进行训练和优化。

通过将这三个组件结合起来，World model能够学习环境的动态模型，并使用该模型进行规划和决策。它可以在内部进行模拟和预测，从而减少对真实环境的依赖，并提供高效的学习和决策能力。这种方法在某些情况下可以提高强化学习的效率和性能，特别是在数据样本稀缺或环境复杂的情况下。

Model-based Deep Reinforcement Learning（基于模型的深度强化学习）是一种结合了模型学习和深度强化学习的方法。它通过构建环境模型来估计环境状态的转移和奖励函数，然后利用这个模型进行规划和决策。Model-based Deep Reinforcement Learning的优势在于可以更高效地利用数据，减少对真实环境的依赖，并提供更好的规划和决策能力。

除了数据之外，GPT-4也还存在一些问题，例如输出的答案不是完全可靠的，GPT-4的答案可能会有偏差；GPT-4无法联网，信息无法更新等，都是需要解决的问题，那么未来的GPT-5也可能在这些方面的问题上做出更大的突破。