预训练语言模型

Word Embedding → BERT

技术演化图

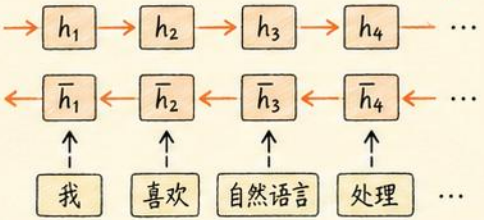
1. Word Embedding

将词映射为低维稠密向量

word	vector
我	[0.21, -0.35, ...]
喜欢	[-0.18, 0.42, ...]
自然语言处理	[0.33, 0.11, ...]
...	...

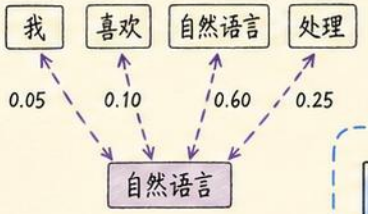
2. ELMo

基于双向LSTM的上下文表示



3. Attention

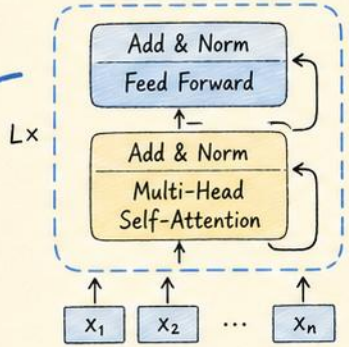
让模型关注更重要的词



加权聚合上下文信息

4. Transformer

多头自注意力 + 前馈网络

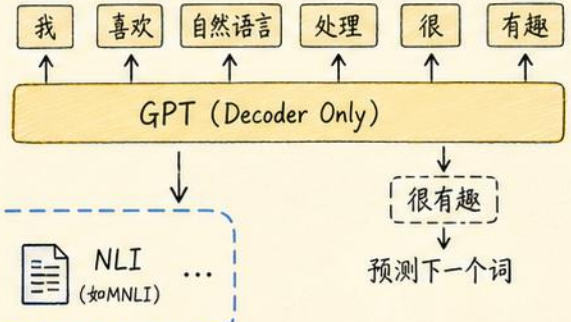


BERT

Bidirectional Encoder Representations from Transformers

5. GPT

基于Transformer的自回归语言模型



Fine-tune

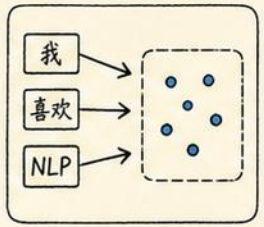
BERT = 双向理解 + 深度上下文 + 大规模预训练

→ 汇聚多种关键思想，成为NLP的里程碑模型

走向 BERT 的时间线

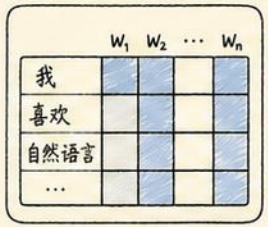
从词向量到预训练：关键技术的演进与汇合

2013 Word2Vec



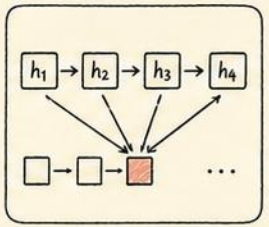
用上下文预测词
获得词向量表示

2014 GloVe



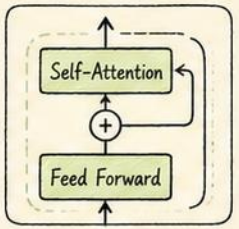
基于全局共现统计
学习词向量

2015 LSTM/Attention



建模序列依赖，
注意力聚合关键信息

2017 Transformer

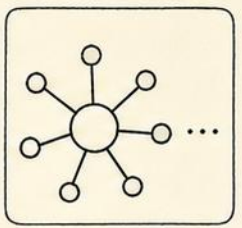


纯注意力架构，
并行高效建模长依赖

2018 GPT / ELMo / BERT

- **GPT** 左到右生成式预训练
- **ELMo** 双向语言模型表征
- **BERT** 双向编码器预训练

2020 GPT-3



大规模模型 + 大数据，
通用能力进一步涌现

预训练范式汇合

不同思路，共同推动
通用语言表示能力跃升

- 表示学习（词向量）
- 序列建模（RNN/注意力）
- 注意力架构（Transformer）
- 预训练范式（通用表示）

预训练 = 参数迁移

大数据预训练

Model A

海量无标注数据

参数初始化

Frozen

冻结预训练参数，只训练上层

：迁移的预训练参数（可复用）

：随机初始化的新参数

Fine-tuning

微调预训练参数，整体训练

小数据任务

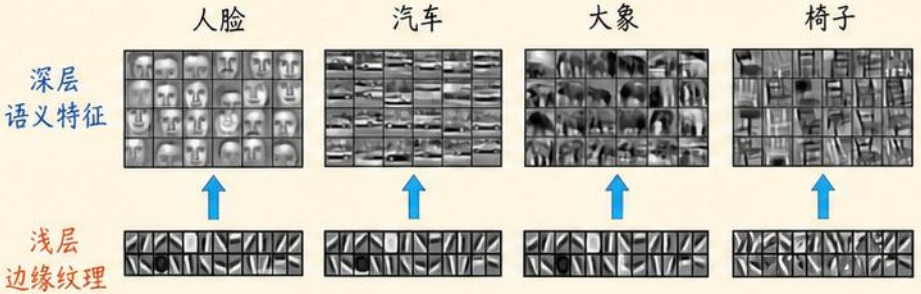
Model B

少量标注数据

目标任务

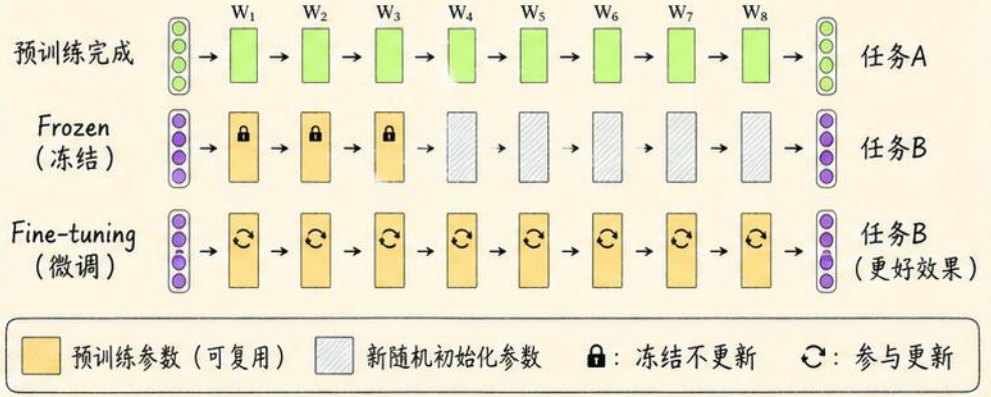
（分类 / 检测 / 分割 …）

证据1：预训练学习通用特征（以自然图像为例）



从浅层边缘纹理 → 深层语义特征，逐层学习通用表示

证据2：冻结或微调的训练流程



语言模型：预测下一个词

语言模型的概率基础

$$P(w_i \mid w_1 \dots w_{i-1})$$

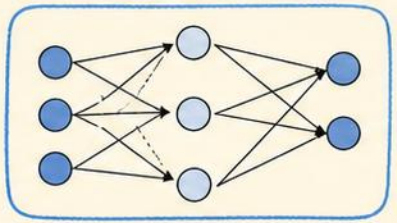
P(next | context)

统计计数

w_1	…	w_{i-1}	count
我	喜欢	自然语言	25
我	喜欢	机器学习	18
我	喜欢	深度学习	12
…	…	…	…

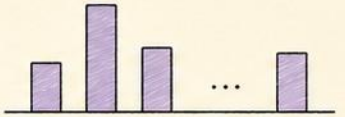
稀疏、维度高

神经估计



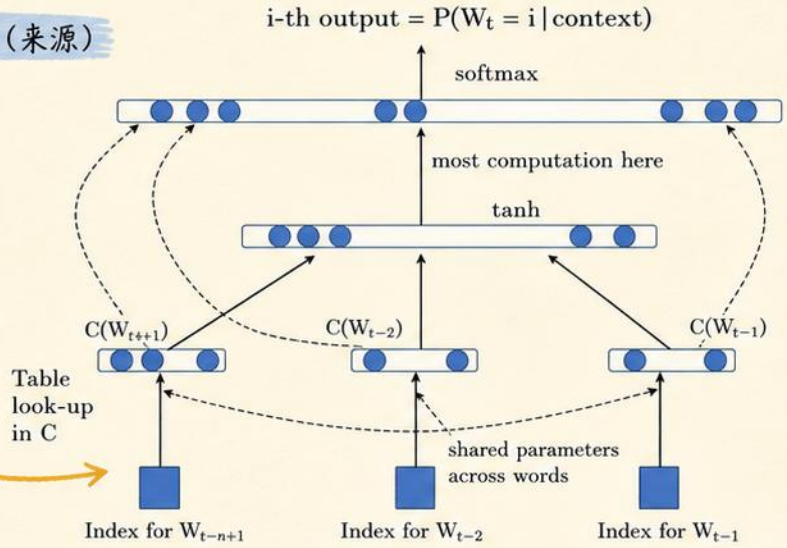
学习语义与上下文

Softmax



每个词的概率分布
（和为 1）

神经语言模型（NNLM）架构（来源）



1 输入：词的索引
通过共享的查表矩阵 C 得到词向量（Embedding）

2 输出：词表上的概率分布
Softmax 将隐藏表示映射为每个词的概率

W_t：待预测的当前词
context：前面的上下文词
C：共享的词向量矩阵

第一代 NLP 预训练

学习一个共享的 Embedding 矩阵，作为下游任务的初始化参数。 但向量是静态的，不懂上下文。

1. One-hot（稀疏表示）

词（如: flies）

time	fruit	flies	like	a	an	arrow	banana
1	0	0	0	0	0	0	0

One-hot（稀疏，高维）

2. Embedding 矩阵（查表）

Embedding 矩阵 $E \in \mathbb{R}^{V \times d}$

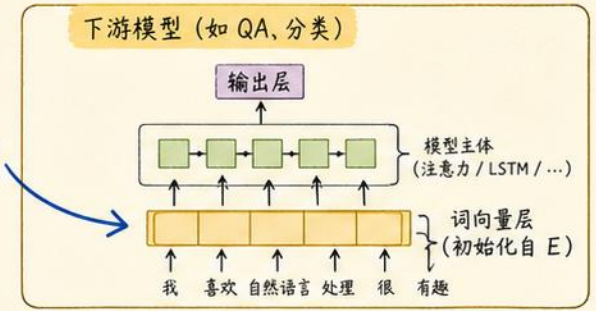
词表					
time	0.21	-0.35	0.12	…	0.08
fruit	-0.18	0.42	-0.07	…	0.31
flies	0.33	0.11	-0.24	…	0.16
like	-0.05	0.27	0.09	…	-0.13
a	…	…	…	…	…
an					
arrow					
banana					

查表

0.33	0.11	-0.24	…	0.16

查表得到稠密向量（静态表示）

3. 下游初始化（静态词向量）

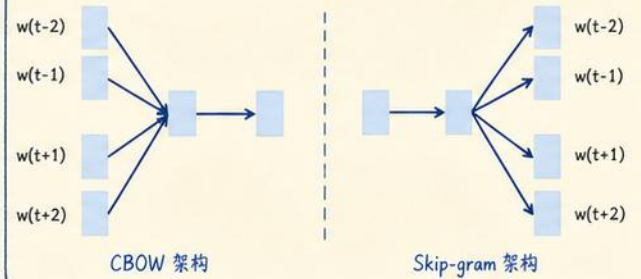


静态词向量：同一词一成不变

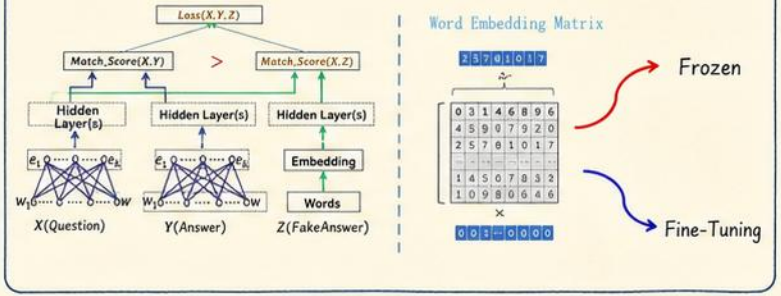
参考图：One-hot 表示

	time	fruit	flies	like	a	an	arrow	banana
1 time	0	0	0	0	0	0	0	0
1 fruit	0	1	0	0	0	0	0	0
1 flies	0	0	0	0	0	0	0	0
1 like	0	0	0	1	0	0	0	0
1 a	0	0	0	1	1	0	0	0
1 an	0	0	0	0	0	0	1	0
1 arrow	0	0	0	0	0	0	1	0
1 banana	0	0	0	0	0	0	1	1

参考图：Word2Vec（查表思想）



参考图：下游任务中的 Embedding 复用



Word Embedding 是第一代 NLP 预训练：通过在大规模语料上学习 Embedding 矩阵，作为通用的初始化参数迁移到下游任务。 但它是静态的，无法根据上下文动态变化。

词义随上下文变化

context matters

一词多义

...very useful to protect banks or slopes from being washed away by river...

...the location because it was high, about 100 feet above the bank of river...

...The bank has plan to branch throughout the country...

...They throttled the watchman and robbed the bank...

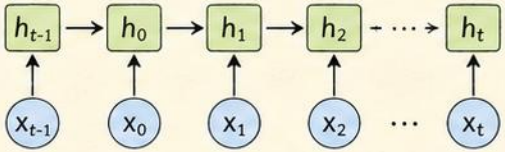
静态Word Embedding（同一向量）

bank

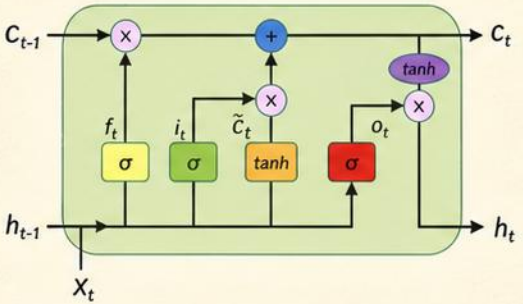
0.25	0.57	-0.13	...	0.17

无法区分不同含义 ☹

RNN / LSTM

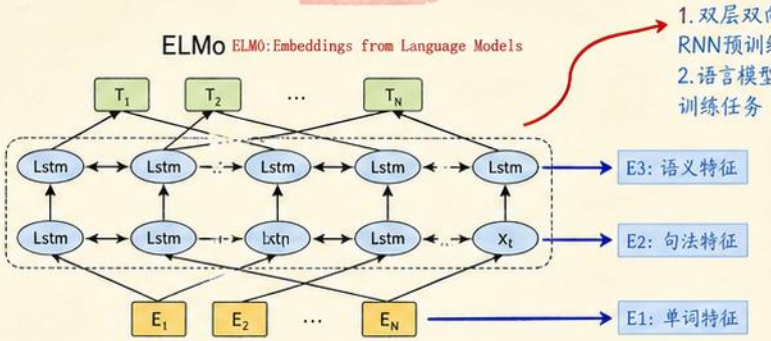


LSTM



顺序建模，捕获上下文信息

ELMo



结合多层双向上下文，得到动态表示

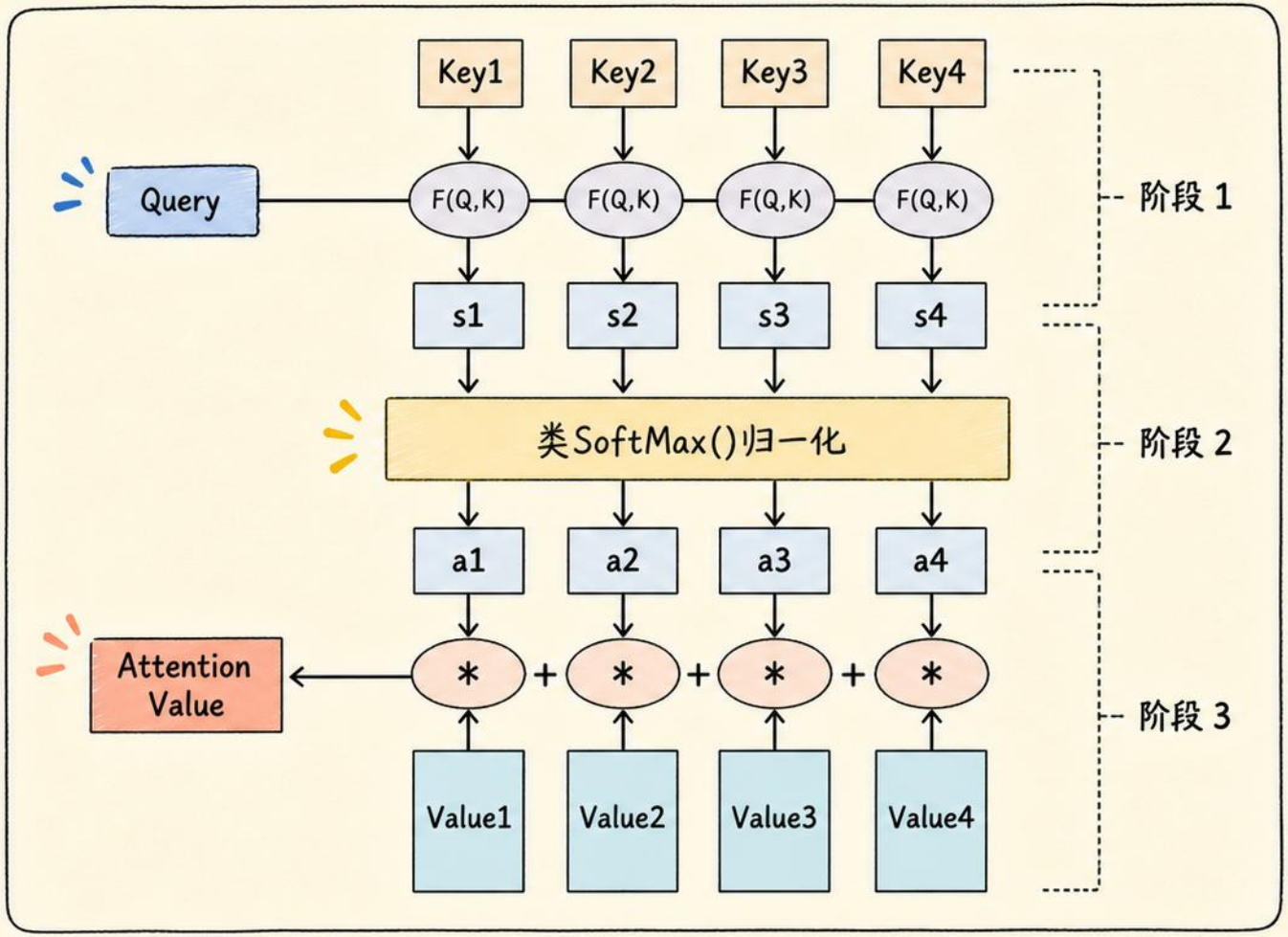
... washed away by river ...	0.82	-0.31	0.47	...	0.21
... above the bank of river ...	-0.63	0.71	-0.22	...	0.08
... bank has plan to branch ...	0.11	0.46	0.93	...	-0.35
... robbed the bank ...	-0.27	-0.89	0.31	...	0.66

同一词在不同上下文 → 不同表示 ✓

静态向量（忽略上下文） → RNN / LSTM（建模顺序，感知上下文） → ELMo（双向多层，动态表示）

词义由上下文决定！

Attention = 相关性加权



1 Q-K 打分

Query 与每个 Key 计算相关性分数
得到 s1, s2, s3, s4 ...

2 Softmax

对分数做 Softmax 归一化，
得到注意力权重 a1, a2, a3, a4 ...

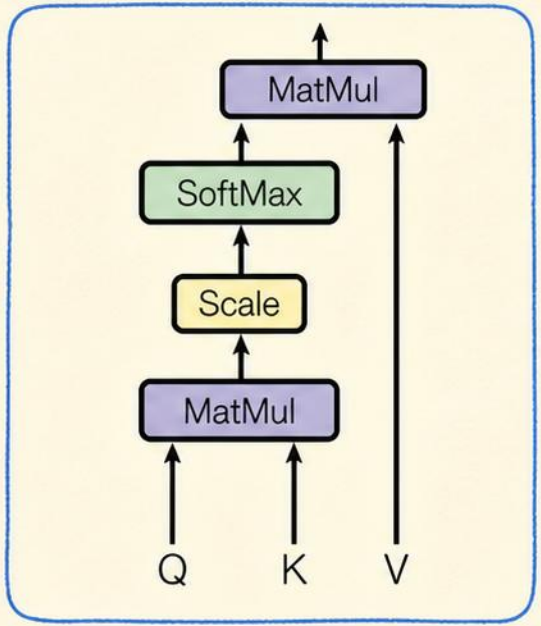
3 加权求和

用权重乘以对应的 Value，
再把结果加和，得到输出

$$\text{softmax}(QK^T)V$$

Attention 家族

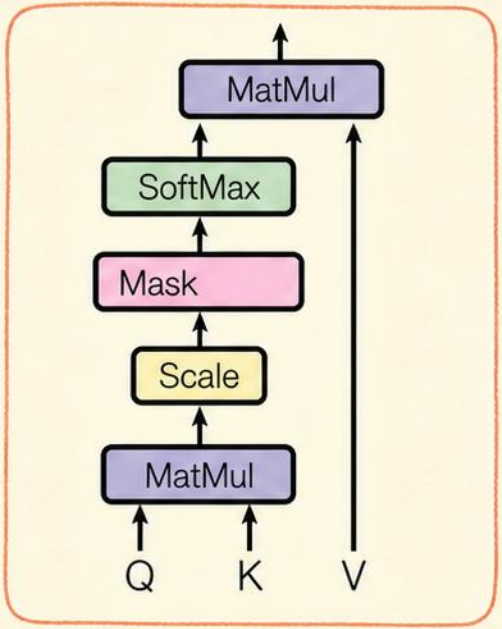
Self-Attention



全局互看

任意位置可关注序列中所有位置，
捕捉全局依赖关系。

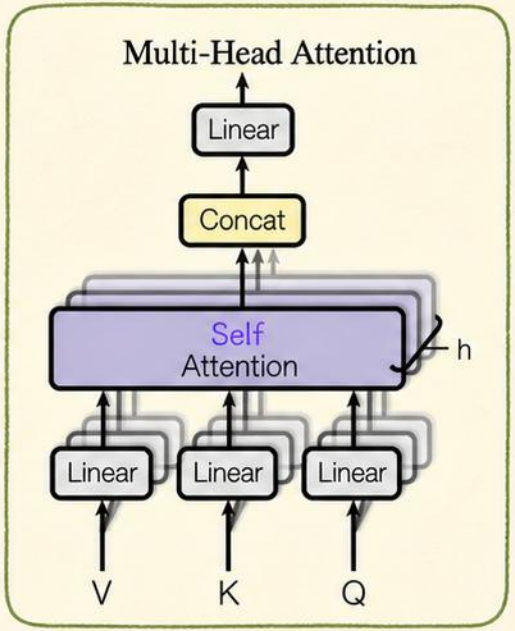
Masked



遮住未来

通过因果 Mask 防止看到未来信息，
保障自回归生成的正确性。

Multi-head



多头并行

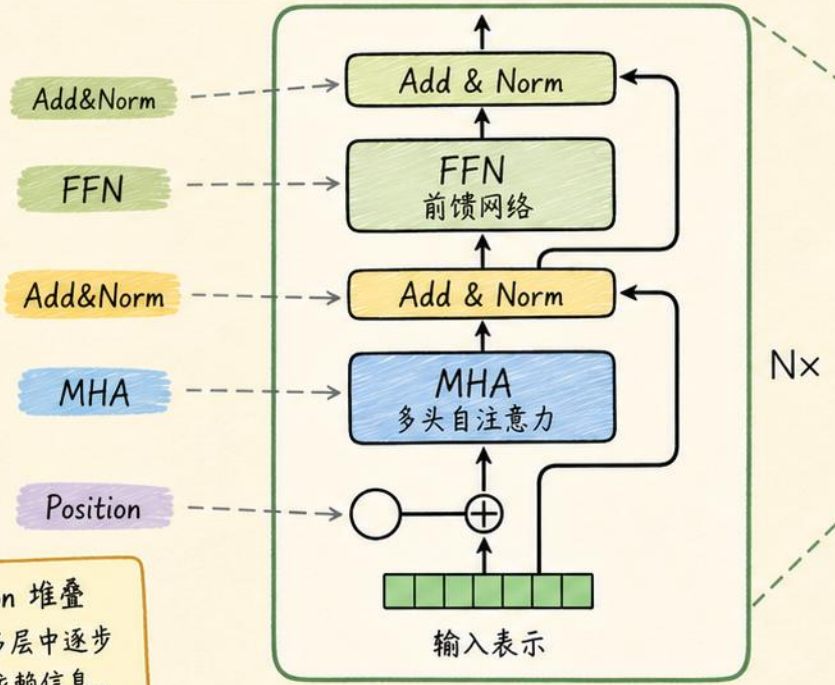
多个注意力头并行学习不同子空间，
增强表示能力与建模灵活性。

Transformer 总装

核心思想：

用注意力（信息交互）+ 前馈网络(特征变换)
在残差连接与归一化下反复堆叠，构成强大的序列建模器。

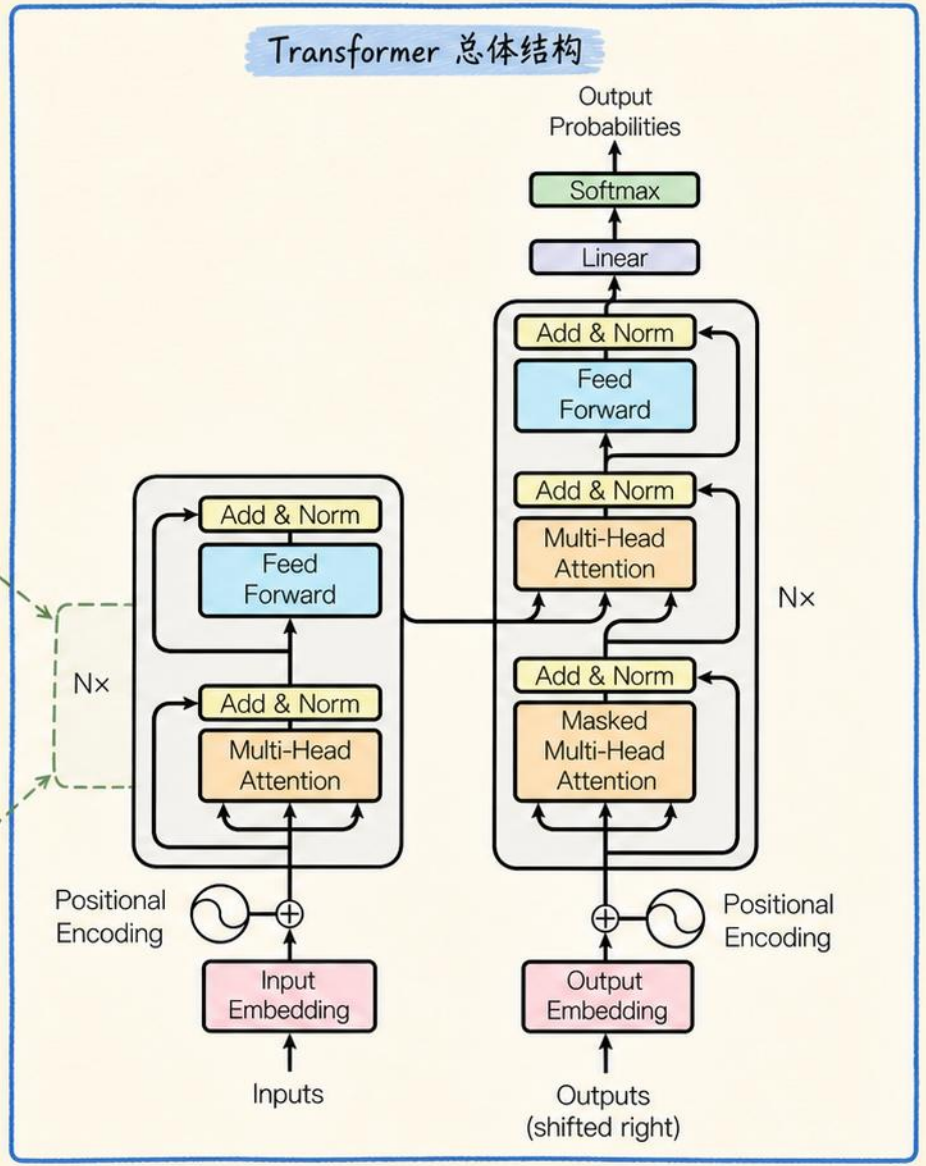
Encoder 单层结构（放大看）



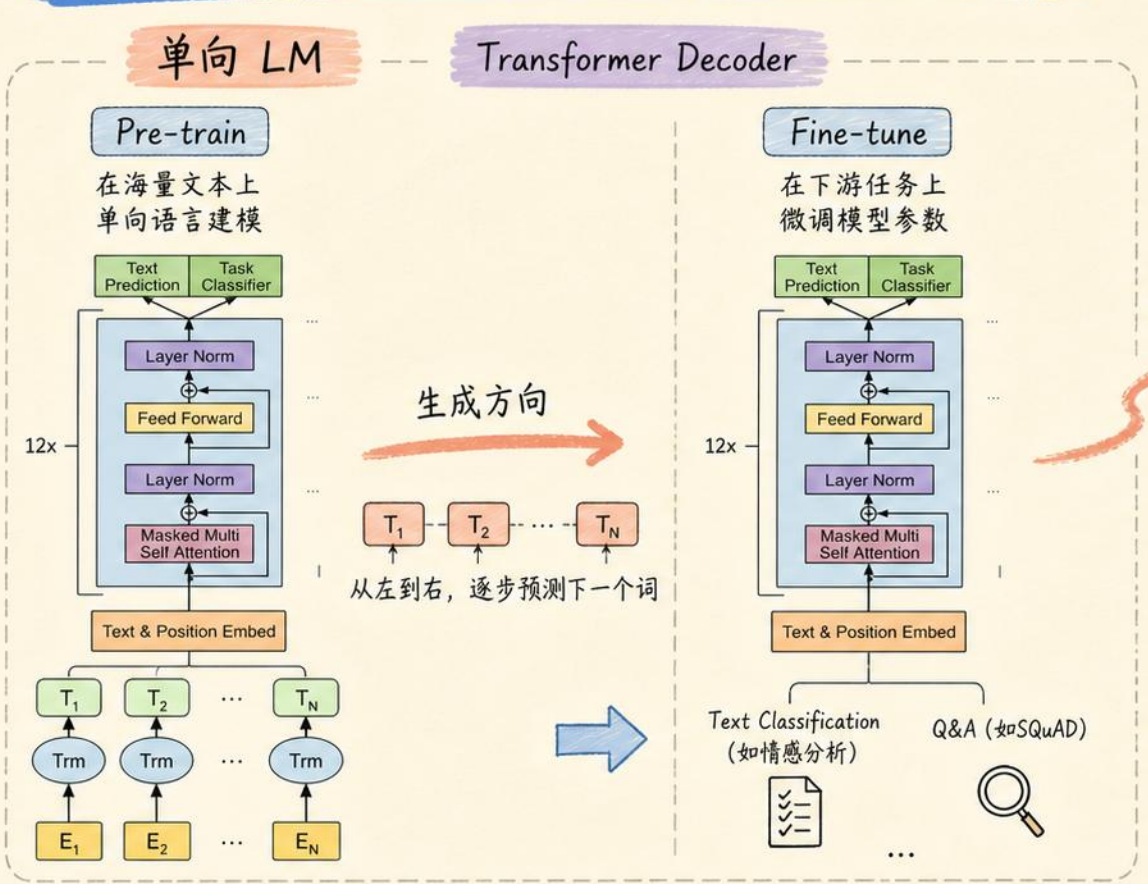
Attention 堆叠

让模型在多层中逐步聚合长程依赖信息。

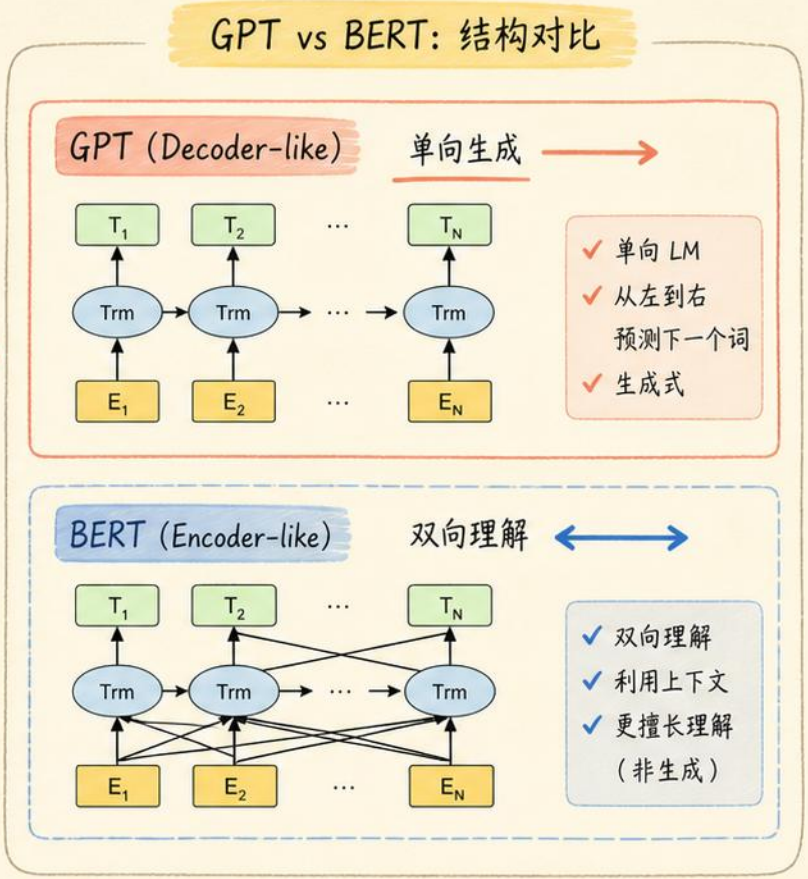
Transformer 总体结构



GPT：单向 LM 路线

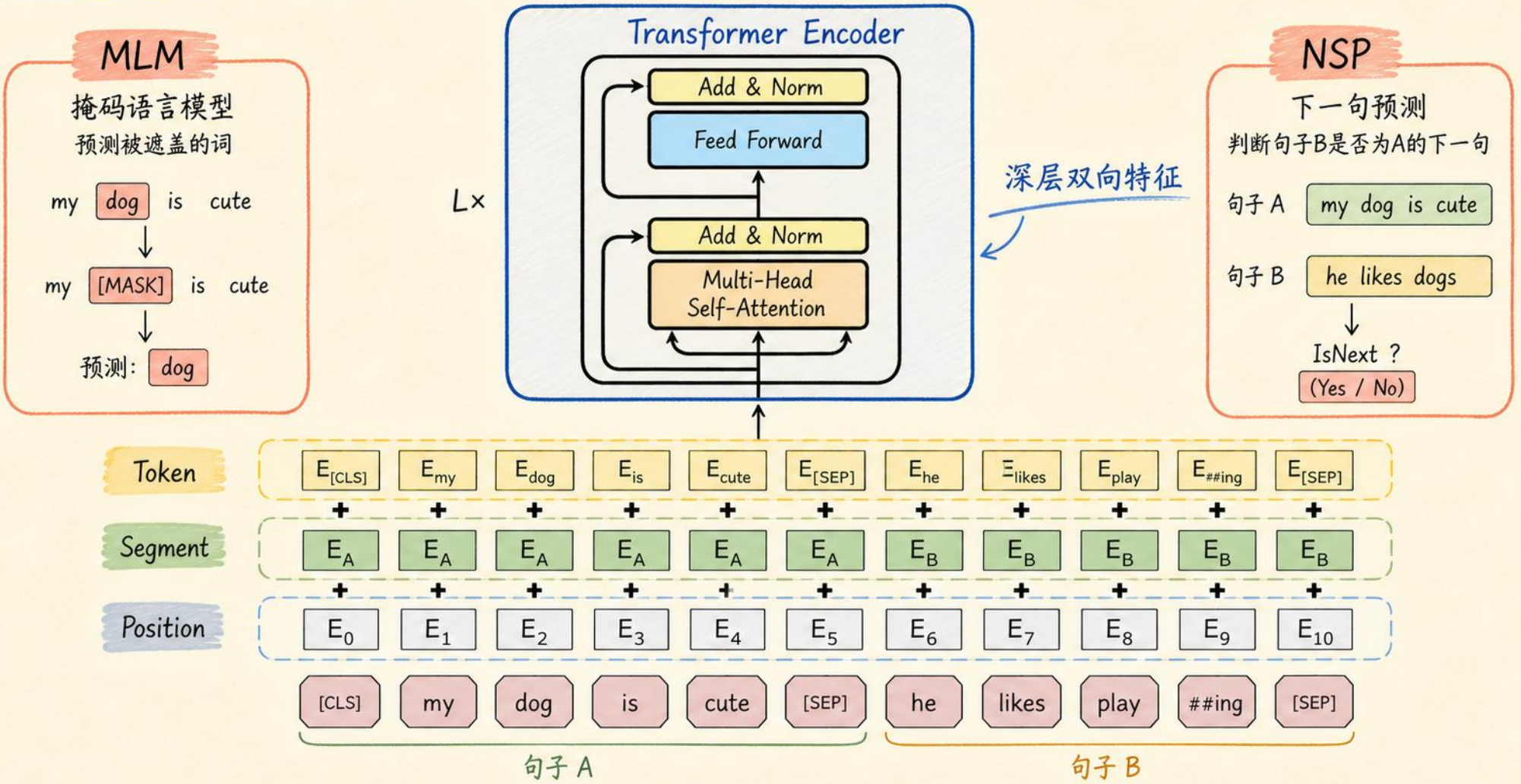


对比



GPT 路线：先在大规模文本上做单向语言建模（Pre-train），再在具体任务上微调（Fine-tune），以生成式方式预测下一个词。

BERT：双向语义理解


MLM
掩码语言模型
预测被遮盖的词
my dog is cute
my [MASK] is cute
预测：dog
Transformer Encoder
Add & Norm
Feed Forward
Add & Norm
Multi-Head Self-Attention
L×
深层双向特征
NSP
下一句预测
判断句子B是否为A的下一句
句子 A
my dog is cute
句子 B
he likes dogs
IsNext ?
(Yes / No)
Token
Segment
Position
E[CLS] Emy Edog Eis Ecute E[SEP] Ehe Elikes Eplay E##ing E[SEP]
EA EA EA EA EA EA EB EB EB EB EB
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 E10
[CLS] my dog is cute [SEP] he likes play ##ing [SEP]
句子 A
句子 B

BERT 作为通用底座

集大成者

统一的预训练模型，
适配多种下游任务

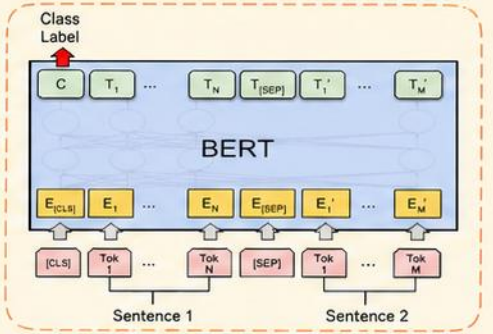
通用预训练底座

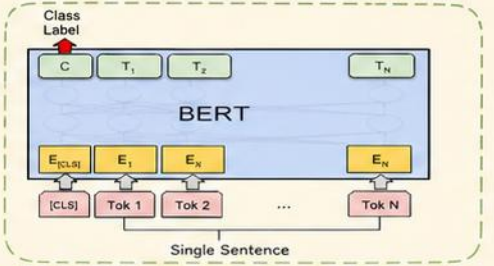
双向理解 + 深度上下文
大规模无监督预训练

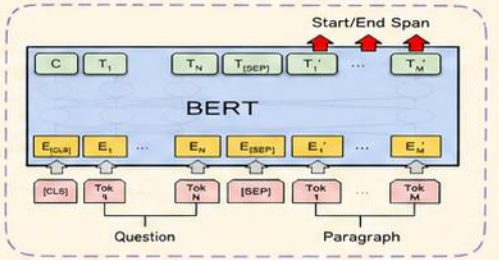
BERT

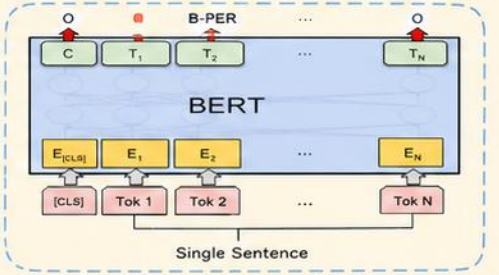
Bidirectional Encoder Representations from Transformers

- **句对分类** → 判断句对关系
- **单句分类** → 理解整句语义
- **问答抽取** → 定位答案片段
- **序列标注** → 逐词标注标签

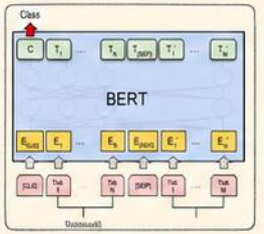


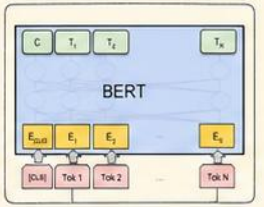


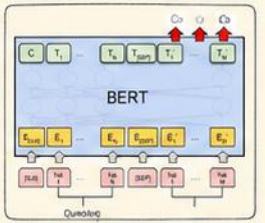


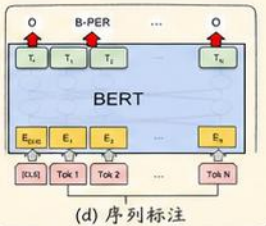


来源图示（参考）


(a) 句对分类


(b) 单句分类


(c) 问答抽取


(d) 序列标注

1. 统一输入
相同的词/位置表示与编码器
适用于多种任务格式

2. 简单任务头
在预训练表示上添加轻量结构
快速适配不同下游任务

3. 两阶段训练
先大规模无监督预训练
再小规模有监督微调