BERT

大模型的三种架构

大模型通常采用三种主要架构：Encoder-Only, Decoder-Only, 和 Encoder-Decoder。

1. Encoder-Only 架构

特点：Encoder-Only 架构仅包含编码器部分。它通常用于处理那些只需要理解输入数据而不需要生成新数据的任务。这种架构通过堆叠多层编码器（通常是自注意力层和前馈神经网络层）来处理和理解输入。

应用场景：**文本分类**：如情感分析、意图识别等。**实体识别**：从文本中识别和分类命名实体。**特征提取**：为下游任务提取有用的特征，比如在更复杂的模型中使用。

典型模型：

- BERT（Bidirectional Encoder Representations from Transformers）是最著名的 Encoder-Only 架构的例子，广泛用于各种文本理解任务。

BERT论文部分

- BERT是双向的model，GPT是单向的model
- GPT由左到右的预测，而BERT是利用左右的信息同时训练（双向的$Transformer$模型）。
- 预训练（Pre-training）是深度学习和自然语言处理（NLP）中的一种常见技术。它指的是在一个大规模的数据集上先训练一个模型，使其能够学习到广泛的基础知识和特征，然后再将这个模型应用到特定任务中，通过微调（fine-tuning）来进一步提升性能。

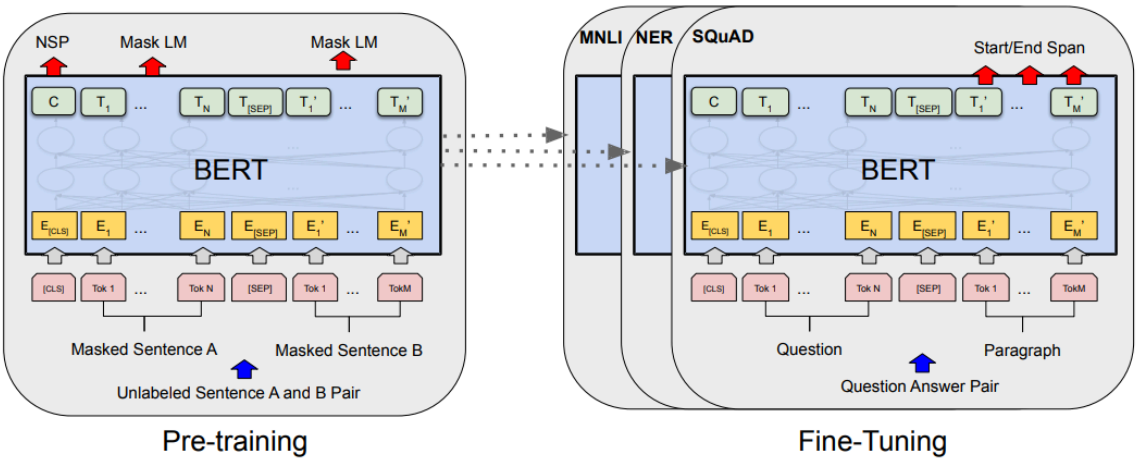
BERT的训练

预训练

- 预训练时使用的是：unlabeled sentences pair

微调

- 使用的是预训练的参数来针对不同的任务微调。



BERT的架构

In this work, we denote the number of layers (i.e., Transformer blocks) as L, the hidden size as H, and the number of self-attention heads as A.[3] We primarily report results on two model sizes: $\textbf{BERT}_{\textbf{BASE}}$ (L=12, H=768, A=12, Total Parameters=110M) and $\textbf{BERT}_{\textbf{LARGE}}$ (L=24, H=1024, A=16, Total Parameters=340M).

-

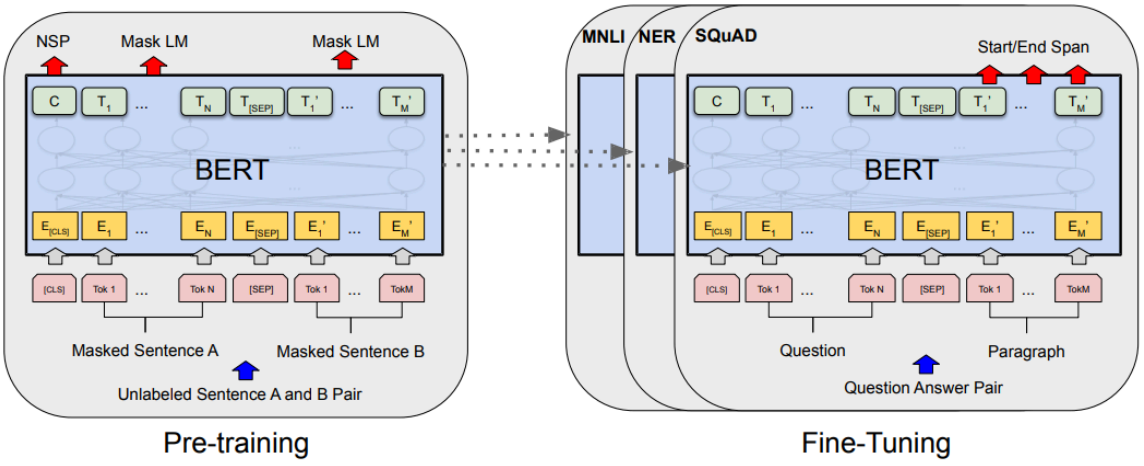
输入： $sequenece$**可以是一个句子或者是一个句子对**

We use WordPiece embeddings (Wu et al., 2016) with a 30,000 token vocabulary. The first token of every sequence is always a special classification token ([CLS]). The final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks. Sentence pairs are packed together into a single sequence. We differentiate the sentences in two ways. First, we separate them with a special token ([SEP]). Second, we add a learned embedding to every token indicating whether it belongs to sentence A or sentence B. As shown in Figure 1, we denote input embedding as E, the final hidden vector of the special [CLS] token as $C \in \mathbb{R}^H$, and the final hidden vector for the i^{th} input token as $T_i \in \mathbb{R}^H$.

- 利用词根词缀提取句子的公共部分可以减小嵌入矩阵的维度

```
penguin [MASK] are flight ##less birds [SEP]
```

- $flightless$在原文中为一个，分为两个词。##表示提示$flight$和$less$的联系
- [CLS]表示$classification$，放在每个句子的头部。利用$transformer$来将这个信息融入整个句子。
- [SEP]表示标记一个序列中的句子的分界线。



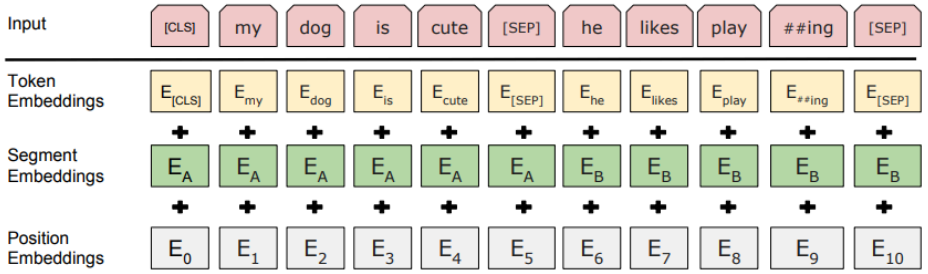


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

- 一个token的嵌入向量=词本身的嵌入向量E_{word}+此所在句子的信息$E_{segement}$+词的位置信息E_{index}

掩码的构造、

- $[mask]$表示用于替换掉需要猜的词

A.1 Illustration of the Pre-training Tasks

We provide examples of the pre-training tasks in the following.

Masked LM and the Masking Procedure Assuming the unlabeled sentence is `my dog is hairy`, and during the random masking procedure we chose the 4-th token (which corresponding to `hairy`), our masking procedure can be further illustrated by

- 80% of the time: Replace the word with the `[MASK]` token, e.g., `my dog is hairy` → `my dog is [MASK]`
- 10% of the time: Replace the word with a random word, e.g., `my dog is hairy` → `my dog is apple`
- 10% of the time: Keep the word unchanged, e.g., `my dog is hairy` → `my dog is hairy`. The purpose of this is to bias the representation towards the actual observed word.