BERT

——深度双向的Transformer： 用于做Pre-training

GPT-左侧信息预测右侧信息

BERT-运用两侧信息

1 Introduction

two strategies for apply pre-trained language representations to down-stream tasks:

- feature-based
  - ELMo (RNN architecture) 模型学习到一定特征 将学到的特征与input放在一起作为输入： 双向RNN
- fine-tuning
  - GPT 运用到下游任务时 Params进行微调： 单向Transformer

但是在文章理解上 理论上应该是全文理解 而非GPT从左到右的一个一个预测下一个token

A+B：双向transformer

而BERT模型：

- 带掩码的训练 掩码后利用双向信息
- "next sentences" 判断抽出的两个句子是否相邻

2 Method

BERT初始化预训练参数 然后在下游任务上微调

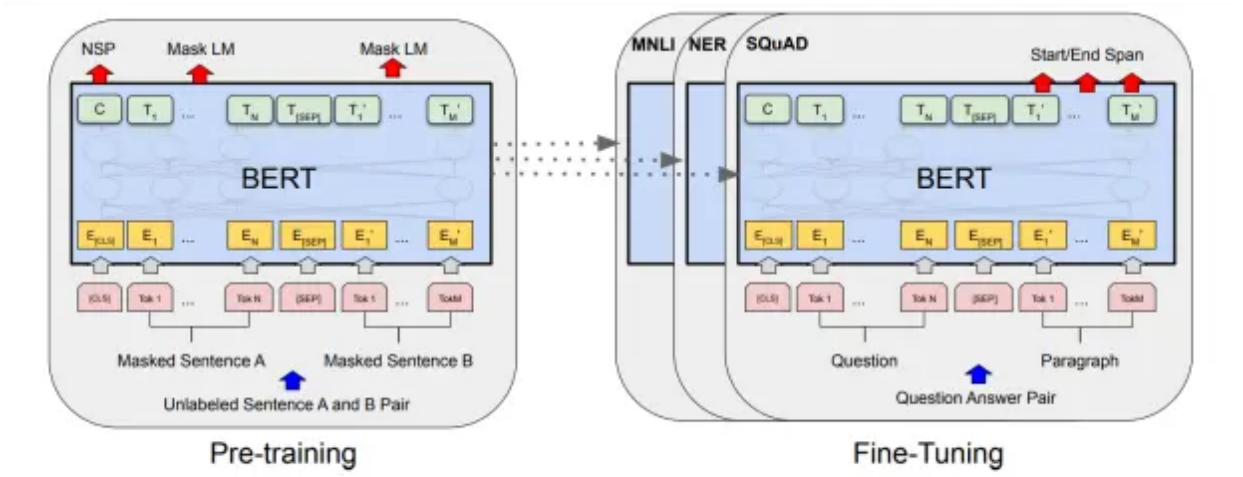

Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. `[CLS]` is a special symbol added in front of every input example, and `[SEP]` is a special separator token (e.g. separating questions/answers).

2.1 Parm：

L: transform blocks的个数

H: hidden size 隐藏层大小

A: 自注意力机制 multi-head 中 head 头的个数

Large 模型 层数 L 翻倍 12 -- 24；宽度 H 768 -- 1024

BERT 模型复杂度和层数 L 是 linear, 和宽度 H 是 平方关系

因为 深度 变成了 以前的两倍，在宽度上面也选择一个值，使得这个增加的平方大概是之前的两倍

H = 16，因为每个 head 的维度都固定在了64。因为你的宽度增加了，所以 head 数也增加了

2.2 Input and Output

transformer预训练的时候输入一个序列对 编码器和解码器格输入一个序列

BERT Encoder-only：输入时可以是a single sentence and a pair of sentences

当输入为两个句子时 需要将两个句子合并为一个sequence

1.**WordPiece**切词：（如果空格切词：词典太大）

1. **将词分解为更小的子词单元**（如词根、前缀、后缀），而不是保留完整词。
2. 通过统计方法自动学习一个**子词词表**，使高频词保持完整，低频词被拆分成更常见的子词。

例如，单词 `"unhappy"` 可能被拆分为 `["un", "##happy"]`，其中 `##` 表示该子词是后续部分。

终止条件：直到达到预设的词表大小（如32k）

2.INPUT

[CLS] 输出的是句子层面的信息 sequence representation

self-attention layer 会看输入的每个词和其它所有词的关系

区分 两个合在一起的句子 的方法：

- 每个句子后 + [SEP] 表示 seperate
- 学一个嵌入层 来表示 整个句子是第一句还是第二句

[CLS] [Token1] …… [Token n] [SEP] [Token1'] …… [Token m]

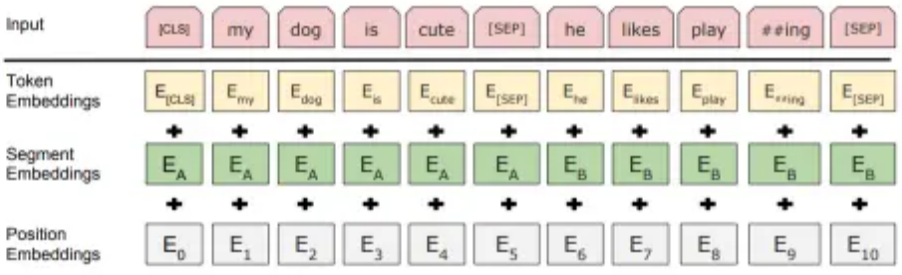

Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

Segement embeddings:这句话属于A还是B

注意 BERT的segment 和position 都是学习来的

2.3 Pre-train

预训练：15%概率随即替换成一个掩码[MASK]

其中80% 真的[MASK] 10%换成random token 还有10%啥都不干

微调时没有[MASK]

2.4 NSP Next Sentence Prediction

在问答和自然语言推理里都是句子对

我们需要：BERT 能学习到 sentence-level 信息

输入序列有 2 个句子 A 和 B，50% 正例，50%反例

50% B 在 A 之后，50% 是 a random sentence 随机采样的。

```
Input = [CLS] the man went to [MASK] store [SEP]
        he bought a gallon [MASK] milk [SEP]
Label = IsNext

Input = [CLS] the man [MASK] to the store [SEP]
        penguin [MASK] are flight ##less birds [SEP]
Label = NotNext
```

BERT 和一些基于encoder-decoder的架构为什么不一样？transformer 是encoder-decoder。

整个句子对被放在一起输入 BERT，self-attention 能够在两个句子之间相互看。BERT 更好，但代价是 不能像 transformer 做机器翻译。

在encoder-decoder的架构，编码器看不到解码器的东西。