目录

[1 理论分析 2](#_Toc14604083)

[1.1 BERT主要特点 2](#_Toc14604084)

[1.2 预训练网络结构 2](#_Toc14604085)

[1.3 预训练方式 Masked LM 和 NSP 3](#_Toc14604086)

[1.3.1 数据处理 3](#_Toc14604087)

[1.3.2 输入Embedding的计算 4](#_Toc14604088)

[1.3.3 Masked LM模型训练方式 5](#_Toc14604089)

[1.3.4 NSP模型训练方式 5](#_Toc14604090)

[1.3.5 最终的训练数据格式 6](#_Toc14604091)

[1.4 应用 6](#_Toc14604092)

# 理论分析

## BERT主要特点

1. 使用Transformer作为特征提取器，代替LSTM。
   1. LSTM具有长期依赖问题，而Transformer学习的则是Source内部各个元素之间的关系，特征提取效果更佳。
   2. 与《Attention is all you need》不同的是，BERT中的Transformer只用了Encoder，而没用Decoder。
2. BERT使用上下文来计算上下文Embedding，而不是使用上文、下文单独计算再融合的方式。
3. 提出了Masked LM预训练语言模型
4. 统一的网络结构，只需要修改一下最终的输出层就可以将预训练网络应用于各种NLP问题中。

## 预训练网络结构

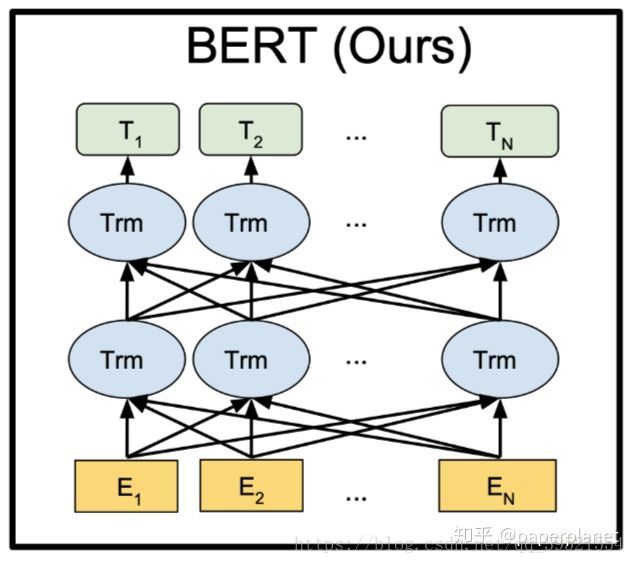


图 1.1BERT模型网络结构

1. BERT的预训练模型使用的是双向Transformer。双向指的是在预测序列中某个token i时，使用i的上下文进行计算：



1. 网络参数说明：
   1. L：Transformer内部encoder的层数
   2. H：hidden size，最终输出的中间编码的维数
   3. A：Transformer中Multi-HeadAttention 模块的head个数，类似于CNN中卷积核的个数，从不同的维度提取特征
   4. 为了与GPT模型作对比，本文设计了两种参数组合：
2. 图1.1中每一层虽然画出了很多Trm，但它们其实是一个Transformer，不过对于每个，都要循环计算一次产生一个对应的输出

## 预训练方式 Masked LM 和 NSP

### 数据处理

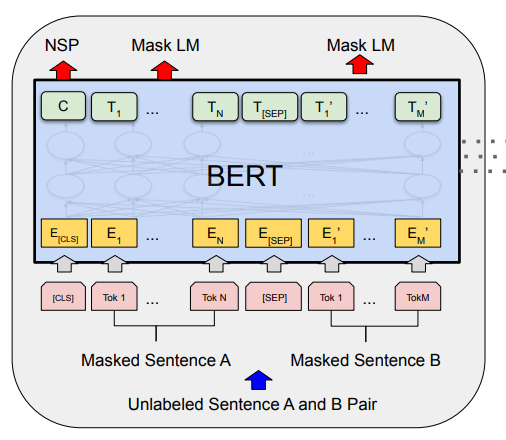


图 1.2数据的处理

1. 每个输入序列的第一个token都是特殊的标志字符[CLS]
2. 多个序列融合成一个序列，原序列之间使用标志字符 [SEP]分割
   1. 比如每组问答数据就是由两个序列构成：<Query,Answer>
3. 在输入BERT预训练模型前，要将序列中的所有token都转换成相应的Embedding表示

### 输入Embedding的计算

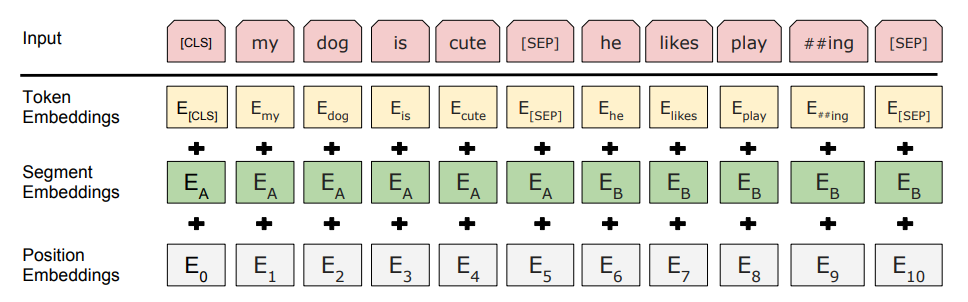


图 1.3输入数据的Embedding计算

输入BERT预训练网络的序列的Embedding是通过3个Embedding融合得到的：

1. Token Embedding：也就是词向量Word Embedding
2. Segment Embedding：用于标识序列中的元素属于哪个句子，句子A或者句子B。
3. Position Embedding：在《Attention is all you need》中，PE是通过三角函数计算出的，但在BERT中，是通过网络学习到的。

在实际的源码里，这三个Embedding都是先随机生成一个embedding词表，然后从其中提取相应词对应的embedding。这三个词表都是tf.Variable变量，在训练的过程中进行学习更新。

### Masked LM模型训练方式

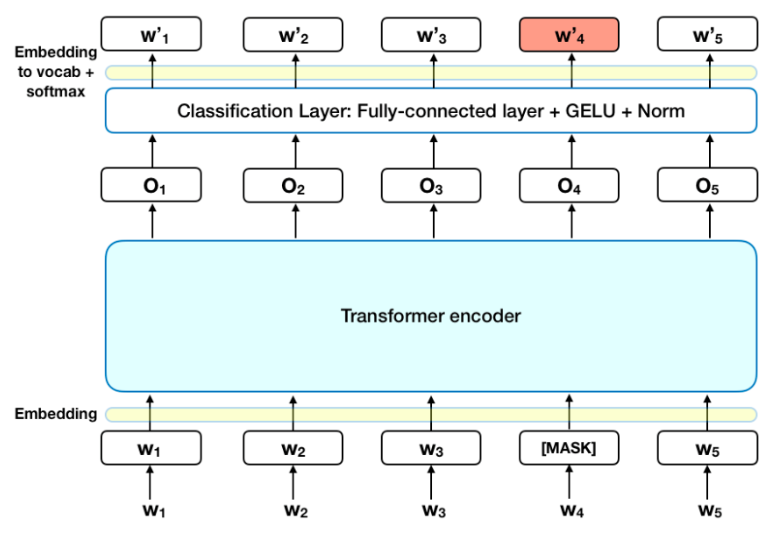


图 1.4 Masked LM模型训练方式

BERT的预训练使用的是无监督训练，对于训练集中的输入序列tokens：

1. 随机选择序列中15%的位置，对该位置对应的token进行Masked，如图1.4中的
2. Masked有三种方式：
   1. 80%的情况下使用标志字段[MASK]替换原来的token
   2. 10%的情况下使用一个随机的token替换原来的token
   3. 10%的情况下保持原有token不变
3. 对于每一个被Masked掉的token i，都会经过预训练网络计算出一个隐藏层输出(如图1.4中的)，将其传入softmax层，然后将其预测结果与替换掉的token i进行对比，计算损失函数，进而更新网络参数

### NSP模型训练方式

NSP(Next Sentence Prediction)，用于解决计算两个句子之间相关性的问题，比如QA。对于这种问题：

1. 我们使用数据集里任意的两个序列A和B组成一个输入序列
2. 经过计算之后，图1.2中最终输出的C就代表模型计算出的序列A和B之间的相关性
3. 将C与序列A、B之间真实的相关性label进行计算损失函数，进而更新网络参数

### 最终的训练数据格式

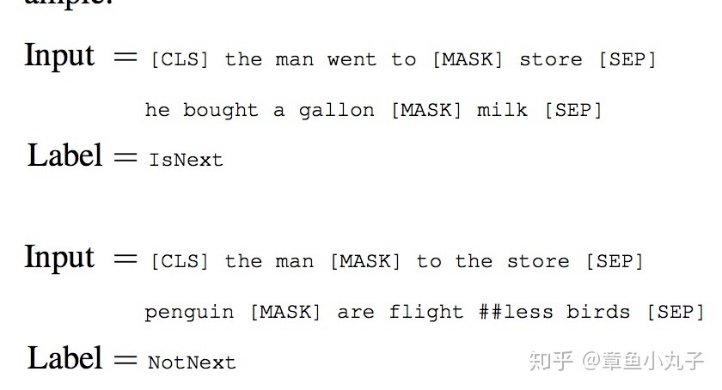


图 1.5训练数据样式

BERT在训练时会同时使用Masked LM 和 NSP两种方式进行训练，综合计算损失函数，以便适应不同的NLP问题，所以其训练数据中既有MASK，也有两个句子之间相关性的标签。

## 应用

在BERT的预训练网络结构之上在加上一个输出层就可以应用到各类问题中，如图1.6所示。

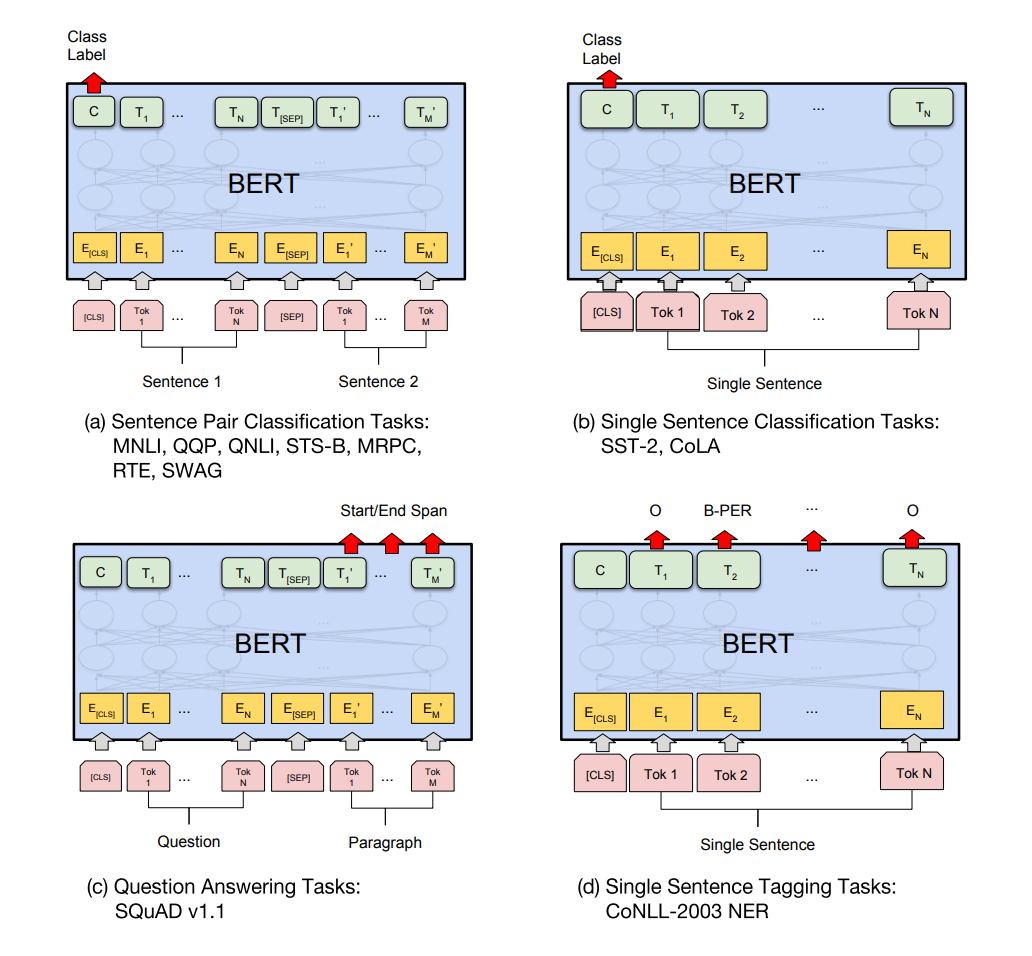


图 1.6 BERT的应用

1. 输入句子对，C是输出，用于以下几种问题：
   1. MNLI、RTE：判断第二个句子是否是第一个句子的限定句、矛盾句或中性句
   2. QQP、MRPC：判断两个句子在语义上是否等价
   3. QNLI：判断第二个句子是否是第一个问题句子的答案
   4. STS-B：判断两个句子在语义上的相似程度
   5. MRPC：
2. 输入单一句子，C是输出，用于以下几种问题：
   1. SST-2：句子情感分类
   2. CoLA：英文句子是否合乎语法
3. 输入一个问题和一个段落，输出答案，用于以下几种问题：
   1. SQuAD：根据问题，从段落中提取答案
4. 输入一个句子，输出各个词对应的标记，用于以下几种问题：
   1. NER：命名实体标记