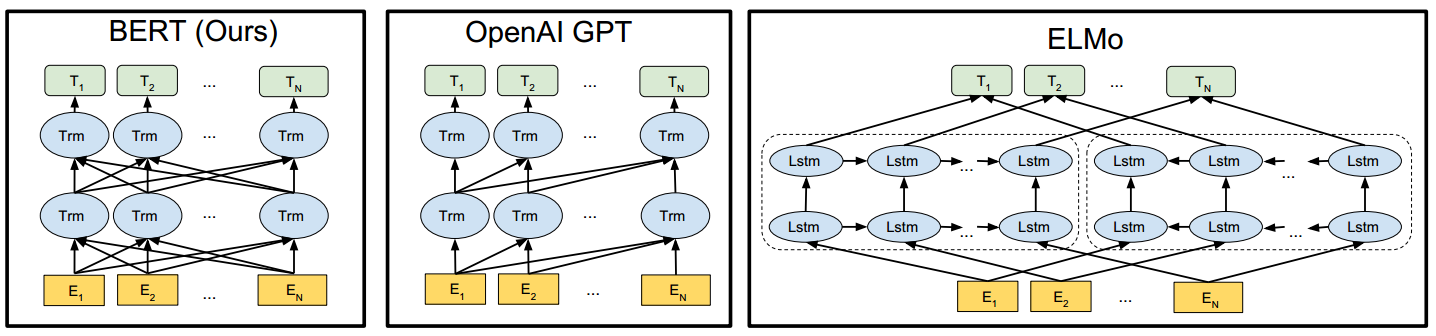
BERT: Pre-training of Deep Bidirectional Transformers forLanguage Understanding

**1.BERT模型**

BERT的全称是Bidirectional Encoder Representation from Transformers，即双向Transformer的Encoder，因为decoder是不能获要预测的信息的。模型的主要创新点都在pre-train方法上，即用了Masked LM和Next Sentence Prediction两种方法分别捕捉词语和句子级别的representation。

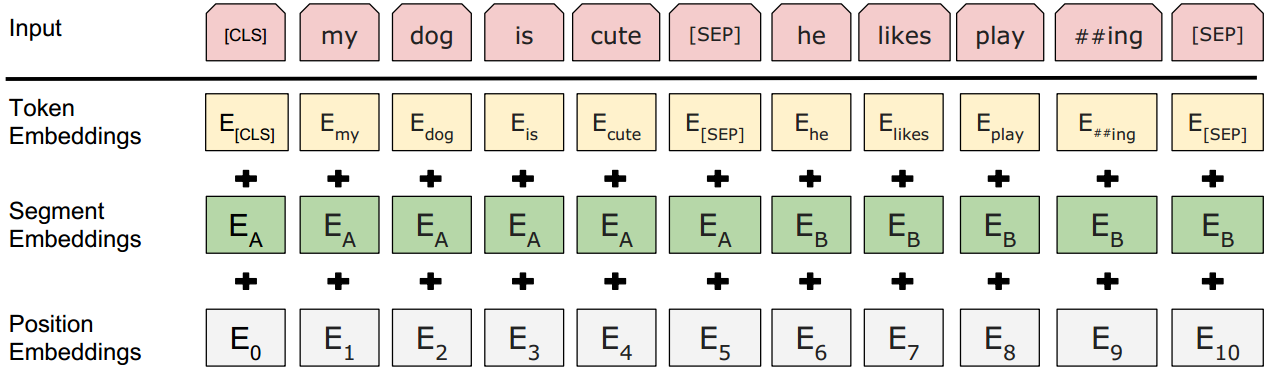
## 1.1 模型结构

由于模型的构成元素Transformer已经解析过，就不多说了，BERT模型的结构如下图最左：

对比OpenAI GPT(Generative pre-trained transformer)，BERT是双向的Transformer block连接；就像单向rnn和双向rnn的区别，直觉上来讲效果会好一些。

对比ELMo，虽然都是“双向”，但目标函数其实是不同的。ELMo是分别以 和  作为目标函数，独立训练处两个representation然后拼接，而BERT则是以  作为目标函数训练LM。

## 1.2 Embedding

这里的Embedding由三种Embedding求和而成：

其中：

* Token Embeddings是词向量，第一个单词是CLS标志，可以用于之后的分类任务
* Segment Embeddings用来区别两种句子，因为预训练不光做LM还要做以两个句子为输入的分类任务
* Position Embeddings和之前文章中的Transformer不一样，不是三角函数而是学习出来的

## 1.3 Pre-training Task 1#: Masked LM

第一步预训练的目标就是做语言模型，从上文模型结构中看到了这个模型的不同，即bidirectional。**关于为什么要如此的bidirectional**，作者在[reddit](http://link.zhihu.com/?target=http%3A//www.reddit.com/r/MachineLearning/comments/9nfqxz/r_bert_pretraining_of_deep_bidirectional/)上做了解释，意思就是如果使用预训练模型处理其他任务，那人们想要的肯定不止某个词左边的信息，而是左右两边的信息。而考虑到这点的模型ELMo只是将left-to-right和right-to-left分别训练拼接起来。直觉上来讲我们其实想要一个deeply bidirectional的模型，但是普通的LM又无法做到，因为在训练时可能会“穿越”（**关于这点我不是很认同，之后会发文章讲一下如何做bidirectional LM**）。所以作者用了一个加mask的trick。

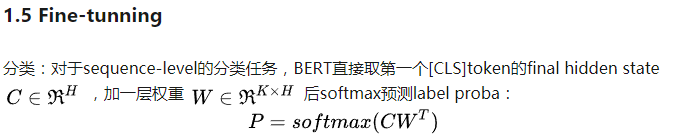
在训练过程中作者随机mask 15%的token，而不是把像cbow一样把每个词都预测一遍。**最终的损失函数只计算被mask掉那个token。**

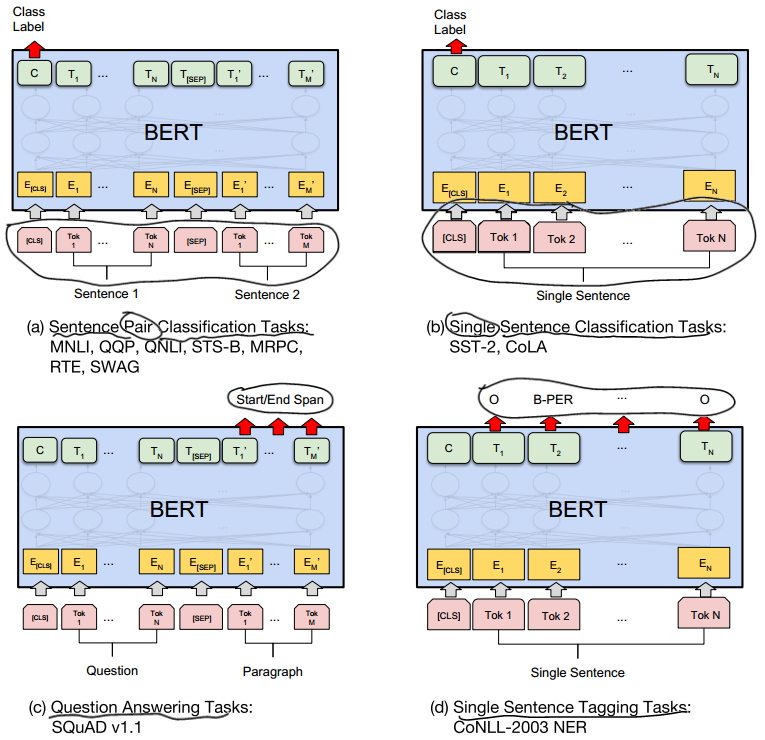
Mask如何做也是有技巧的，如果一直用标记[MASK]代替（在实际预测时是碰不到这个标记的）会影响模型，所以随机mask的时候10%的单词会被替代成其他单词，10%的单词不替换，剩下80%才被替换为[MASK]。具体为什么这么分配，作者没有说。。。要注意的是Masked LM预训练阶段模型是不知道真正被mask的是哪个词，所以模型每个词都要关注。

## 1.4 Pre-training Task 2#: Next Sentence Prediction

因为涉及到QA和NLI之类的任务，增加了第二个预训练任务，目的是让模型理解两个句子之间的联系。训练的输入是句子A和B，B有一半的几率是A的下一句，输入这两个句子，模型预测B是不是A的下一句。预训练的时候可以达到97-98%的准确度。

**注意：作者特意说了语料的选取很关键，要选用document-level的而不是sentence-level的，这样可以具备抽象连续长序列特征的能力。**



其他预测任务需要进行一些调整，如图：

可以调整的参数和取值范围有：

* Batch size: 16, 32
* Learning rate (Adam): 5e-5, 3e-5, 2e-5
* Number of epochs: 3, 4

因为大部分参数都和预训练时一样，精调会快一些，所以作者推荐多试一些参数。

## 2. 优缺点

## 2.1 优点

BERT是截至2018年10月的最新state of the art模型，通过预训练和精调横扫了11项NLP任务，这首先就是最大的优点了。而且它还用的是Transformer，也就是相对rnn更加高效、能捕捉更长距离的依赖。对比起之前的预训练模型，它捕捉到的是真正意义上的bidirectional context信息。

## 2.2 缺点

作者在文中主要提到的就是MLM预训练时的mask问题：

1. [MASK]标记在实际预测中不会出现，训练时用过多[MASK]影响模型表现
2. 每个batch只有15%的token被预测，所以BERT收敛得比left-to-right模型要慢（它们会预测每个token）

BERT的“里程碑”意义在于：证明了一个非常深的模型可以显著提高NLP任务的准确率，而这个模型可以从无标记数据集中预训练得到。

既然NLP的很多任务都存在数据少的问题，那么要从无标注数据中挖潜就变得非常必要。在NLP中，一个最直接的有效利用无标注数据的任务就是语言模型，因此很多任务都使用了语言模型作为预训练任务。但是这些模型依然比较“浅”，比如上一个大杀器，AllenNLP的[ELMO](https://link.zhihu.com/?target=https%3A//arxiv.org/abs/1802.05365" \t "_blank)也就是三层的BiLSTM。

那么有没有可以胜任NLP任务的深层模型？有，就是transformer。这两年，transformer已经在机器翻译任务上取得了很大的成功，并且可以做的非常深。自然地，我们可以用transformer在语言模型上做预训练。因为transformer是encoder-decoder结构，语言模型就只需要decoder部分就够了。OpenAI的[GPT](https://link.zhihu.com/?target=https%3A//s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf)就是这样。但decoder部分其实并不好。因为我们需要的是一个完整句子的encoder，而decoder的部分见到的都是不完整的句子。所以就有了BERT，利用transformer的encoder来进行预训练。但这个就比较“反直觉”，一般人想不到了。