# ALBERT

## 贡献：

1.减少了模型的参数

2.使用更少的内存

3.提升了模型的效果

1.对Vocabulary Embedding的改进（Factorized Embedding Parameterization)

2.Cross-layer parameter sharing

3.Inter-sentence coherence loss

## 介绍

ALBERT 架构的骨干网络与 BERT 是相似的，即使用 Transformer 编码器和 GELU 非线性激活函数。

GELUs正是在激活中引入了随机正则的思想，是一种对神经元输入的概率描述，GELUs对于输入乘以一个0,1组成的mask，而该mask的生成则是依概率随机的依赖于输入。假设输入为X, mask为m，则m服从一个伯努利分布(Φ(x),Φ(x)=P(X<=x),X服从标准正太分布)，这样选择是因为神经元的输入趋向于正太分布。这样设定使得当输入x减小的时候，输入会有一个更高的概率被dropout掉，这样的激活变换就会随机依赖于输入了。

**数学表达式如下：**

*GELU*(*x*)=*xP*(*X*<=*x*)=*x*Φ(*x*)q

**对于假设为标准正太分布的GELU,近似计算的数学公式如下：**

*GELU*(*x*)=0.5*x*(1+*tanh*[​(*x*+0.044715*x*3)])

**其代码为：**

1 | def gelu(input\_tensor):

2 | cdf = 0.5 \* (1.0 + tf.erf(input\_tensor / tf.sqrt(2.0)))

3 | return input\_tesnsor\*cdf

**论文地址：**

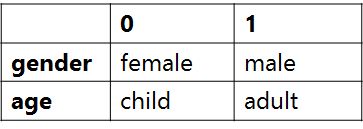
<https://openreview.net/pdf?id=H1eA7AEtvS>

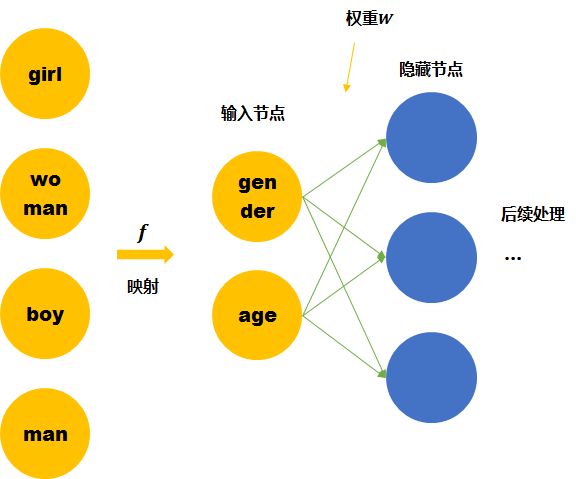
**Albert（中文）GitHub地址：**[https://github.com/brightmart/albert\_zh#%E6%A8%A1%E5%9E%8B%E4%B8%8B%E8%BD%BD-download-pre-trained-models-of-chinese](https://github.com/brightmart/albert_zh" \l "%E6%A8%A1%E5%9E%8B%E4%B8%8B%E8%BD%BD-download-pre-trained-models-of-chinese)

### 1. Factorized embedding parameterization

在BERT中，词表的embedding size与transformer层的hidden size都相同，会导致模型参数很大。因此对于vocabulary embedding，作者先把one-hot映射到一个低维空间之后再映射到hidden layer。

如：

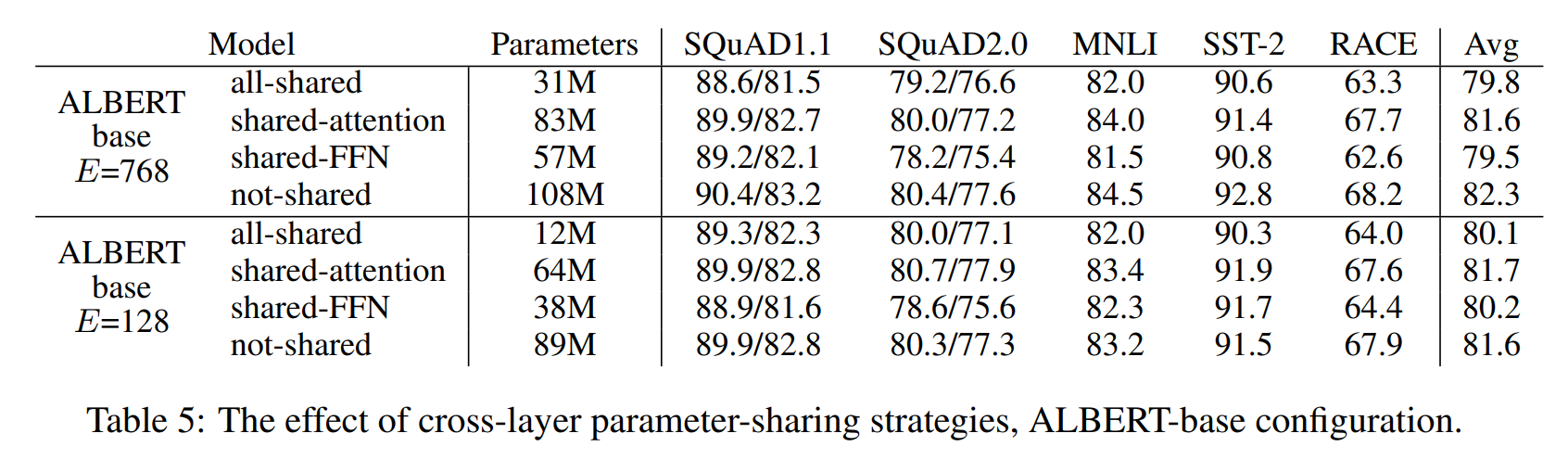
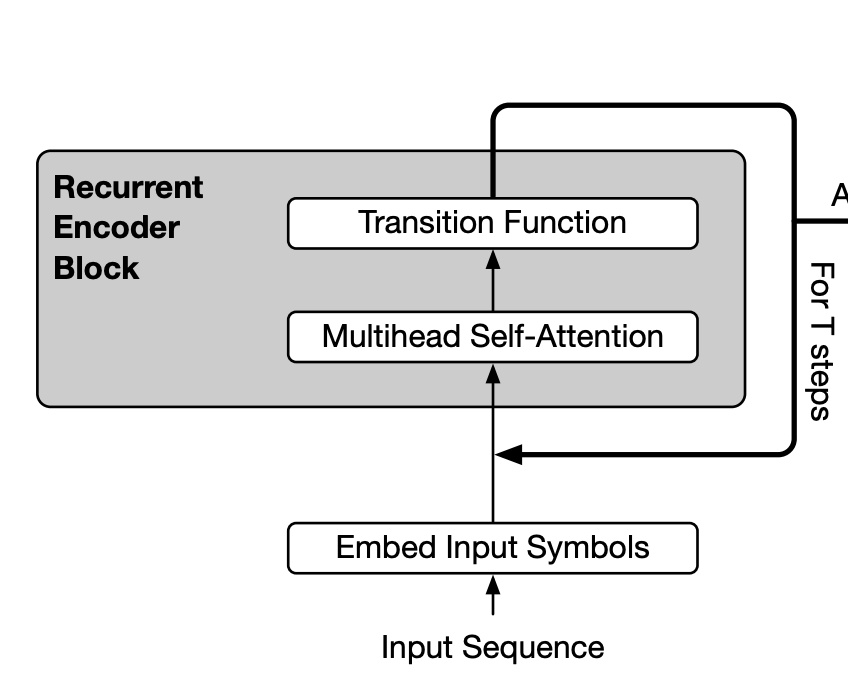


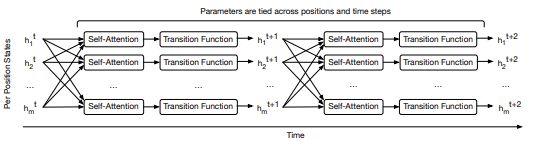


### 2. Cross-layer parameter sharing

共享参数主要分三种——只共享attention相关参数，只共享FFN相关参数，共享所有参数。

全部参数共享虽然ALBERT的平均性能有所降低，但是参数量减少了许多，当于把多个完全相同的层摞起来。



 比如一个序列( a, b, c, d )，先经过embedding表示成 ，在经过一层attention + transition表示成。如果是RNN，那就要先计算 ，再计算和 ，而transformer的self-attention可以同时计算 ，再计算下一级t+1的。

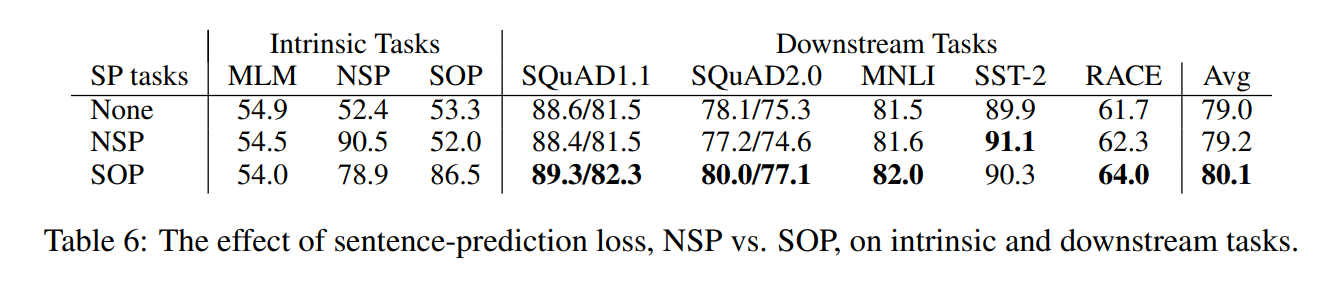
### 3.Inter-sentence coherence loss

a sentence-order prediction (SOP) loss：句子顺序预测，正样本=正常顺序的2个相邻句子，负样本=调换顺序的2个相邻句子

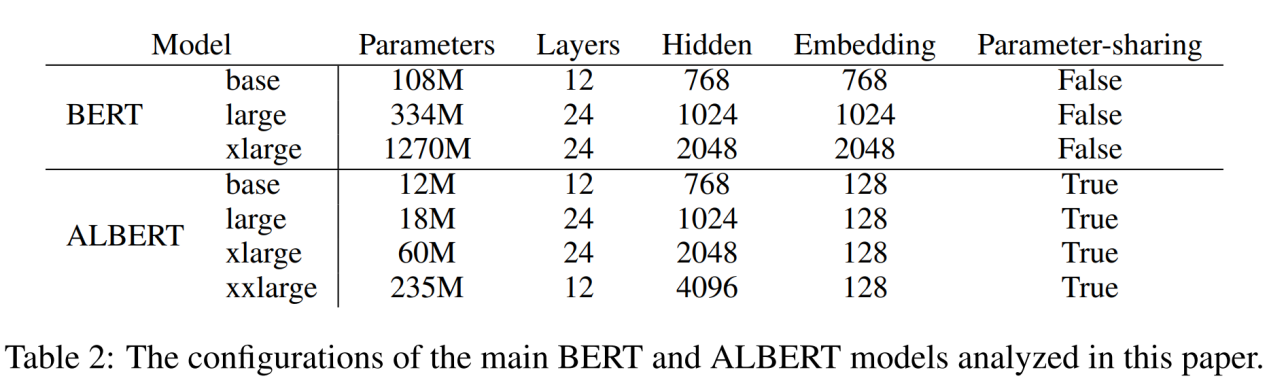
BERT的NSP loss：预测两个片段在原文本中是否连续出现的二分类损失。目标是为了提高如NLI等下游任务的性能

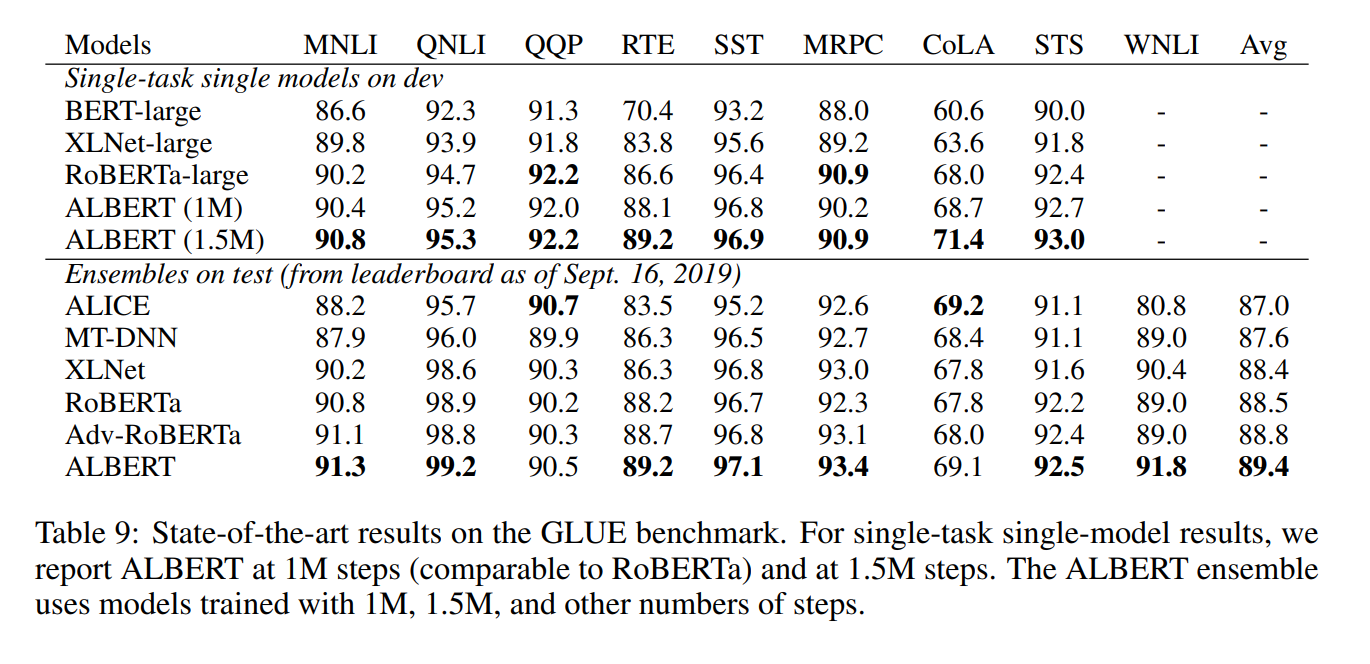
区别：SOP的正例选取方式与BERT一致（来自同一文档的两个连续段），而负例选取不同于BERT中的sample，同样是来自同一文档的两个连续段，但交换两段的顺序，从而避免了主题预测，只关注建模句子之间的连贯性。

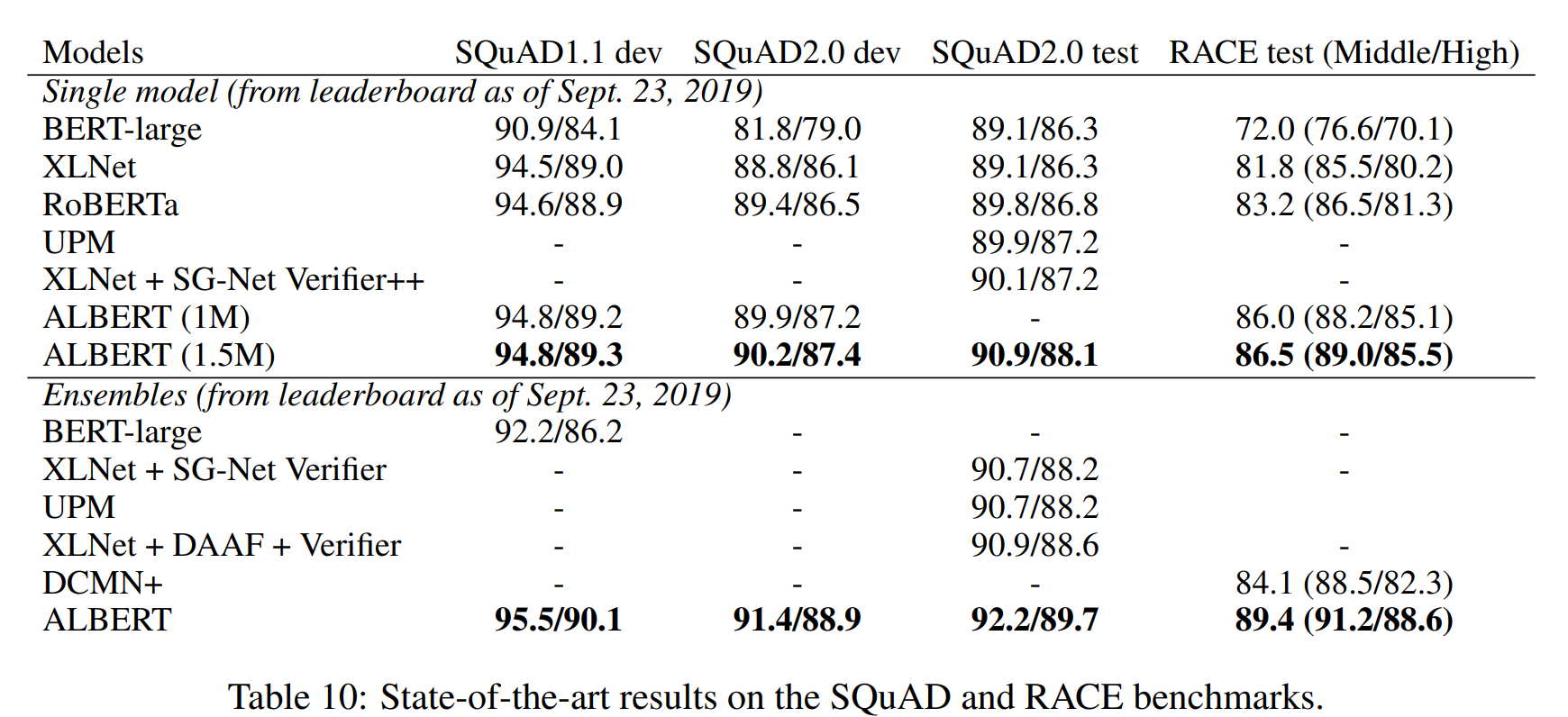
但最近的研究都表示 NSP 的作用不可靠，都选择了不使用NSP。作者推测，NSP效果不佳的原因是其难度较小。将主题预测和连贯性预测结合在了一起，但主题预测比连贯性预测简单得多，并且它与LM损失学到的内容是有重合的。



## BERT和ALBERT模型的结构对比

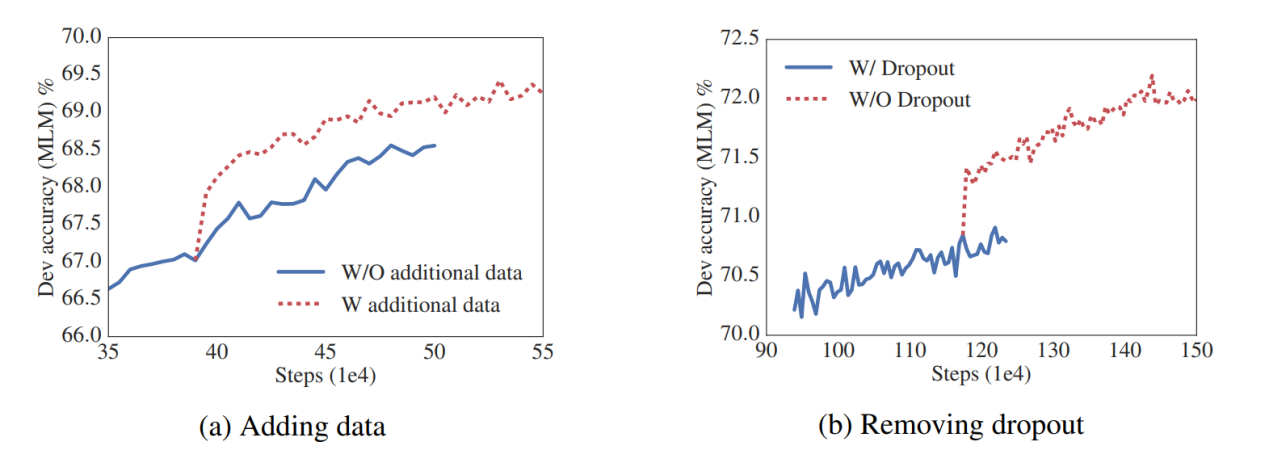


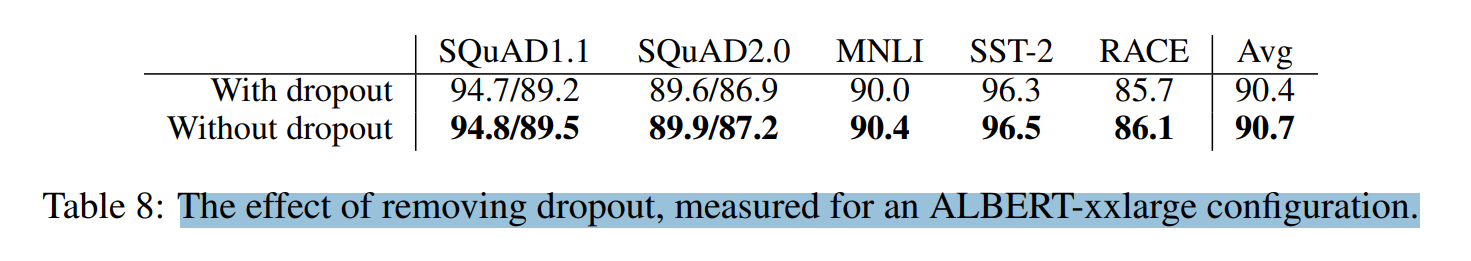




## ALBERT中一些小的trick

### 去掉了dropout

dropout是为了降低过拟合而增加的机制，但由于ALBERT的最大模型在训练1M步后仍然没有过拟合，所以对于Bert删除dropout，进一步提高模型能力。



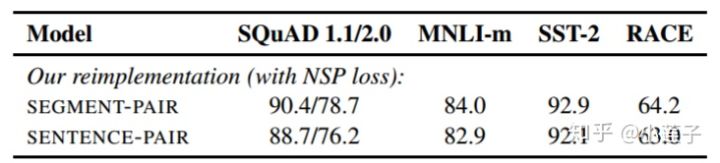
### Masked-ngram-LM

BERT的MLM目标是随机MASK15%的词来预测，ALBERT预测的是N-gram片段，包含更多的语义信息，每个片段长度n（最大为3），根据概率公式计算得到。比如1-gram、2-gram、3-gram的的概率分别为6/11、3/11、2/11.越长概率越小：

### Segments-Pair

BERT 为了加速训练，前 90% 的 steps 使用了 128 个 token 的短句，最后 10% 才使用 512 个词的长句来训练 position embedding：

ALBERT貌似90%的情况下使用512的segment，其输入序列要比 BERT 长接近一半。



在 RoBERTa 的文章中，作者对比了不同序列长度下 NSP 目标的效果，分别是使用自然句作为样本的 sentences-pair，和 BERT 系的 segments-pair。从数据上看，更长的数据提供更多的上下文信息，可能显著提升模型的能力。