BERT

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. NAACL 2018 [PDF]

---

1 简介

近年来的工作显示，预训练可以有效地提高各项NLP任务的成绩。把预训练表征(representation)应用在下游任务上有两种策略：一种是基于特征，类似于ELMo这种杀鸡取卵式；另一种是微调(fine-tuning)，比如GPT，在下游任务上引入很少的task-specified parameters。二者都包含一个预训练模型，并且预训练目标都一样，区别在于面对下游任务时，ELMo只提供词表征，而模型需要根据下游任务确定；GPT则是保留预训练模型，然后在此基础上引入少量的网络层，微调时将调整模型所有参数。

Jacob Devlin等人认为当前预训练语言模型（如Open GPT）受限于单向建模，由此提出一个深度的双向语言模型BERT(bidirectional encoder representation from Transformer).

BERT的贡献有两点：

- 在GPT的基础上，提出双向语言建模，即Masked Language Modeling(MLM)；
- 提出NSP预训练任务，同时来学习句子对的表征；

> 🎨 感想
>
> BERT指名道姓GPT的单向建模会损害性能，但是依然没有阻止后来GPT2、GPT3的崛起，而NSP后来也成为BERT的诟病。如果不是BERT简单且高效，恐怕早已被人遗忘。由此可见，在深度学习领域，在没得到充分的数学证明之前，任何理论和观点是站不住脚的，只有成绩才能证明一切，并且即使成绩好也只能说明自己的方法有效，无法证明别人的方法不对。

2 模型架构

模型方面，BERT由12层Transformer Encoder层组成，参考 Transformer ；

输入和输出方面，BERT的输入有两种形式，一种是单句子，另一种是句子对。对于输入的任意句子，BERT首先通过BPE子词算法将句子转化成token sequence，详见博客《子词算法》。词典大小为21228，既包含英文也包含中文；

随后，BERT将根据token sequence生成token id、position id和token type id，并通过lookup的方式得到三种embedding。以单句子`I like playing soccer`和句子对`(I like playing soccer, 我喜欢踢足球)`两种输入为例，得到的id分别为：

Plain Text

```
token:          [CLS]  i  like  play  ##ing  soc  ##cer  [SEP]
token id:        101  151 8993  8942  8221  11405 10326   102
position id:      0    1   2     3     4      5     6      7
token type id:    0    0   0     0     0      0     0      0


token:          [CLS]  i  like  play  ##ing  soc  ##cer  [SEP]   我     喜
欢     踢     足     球 [SEP]
token id:        101  151 8993  8942  8221  11405 10326   102  2769  1599
3614  6677  6639  4413   102
position id:      0    1   2     3     4      5     6      7      8     9
10    11    12    13    14
token type id:    0    0   0     0     0      0     0      0      1     1
1     1     1     1     1
```

- 引入[CLS]和[SEP]两个特殊符号将两种输入形式统一起来：[CLS]总是位于句子首位，用于表达句子的整体语义，[SEP]为分隔符；
- 引入token type id来区分当前token所属的句子id；
- BERT的position embedding不再是固定的正弦余弦曲线，而是通过学习生成；

最终，将得到的token embedding、position embedding和token type embedding相加，即为BERT最终的输入。

3 预训练任务

BERT的预训练任务有Masked Language Modeling(MLM)和Next Sentence Prediction(NSP)两个。

3.1 MLM

BERT随机用[MASK]字符来覆盖token，然后通过上下文来预测被遮掩的字符的真实id，这样既可以避免"偷窥"还能起到预训练"语言模型"的作用。具体地，BERT随机选择15%的token来预测，一旦第i个token被选中，有三种处理结果：

- 80%的概率用[MASK]替换
- 10%的概率用随机token替换
- 10%的概率不替换

无论采取那种处理，都通过交叉熵损失来预测。

3.2 NSP

BERT在[CLS]字符处，预测输入的句子对是否相邻，这个预训练任务主要的目的是学习句子间的语义。实际操作时，将生成等量的正负句子对样例(分别对应标签IsNext和NotNext)数据以供训练。

3.3 预训练语料

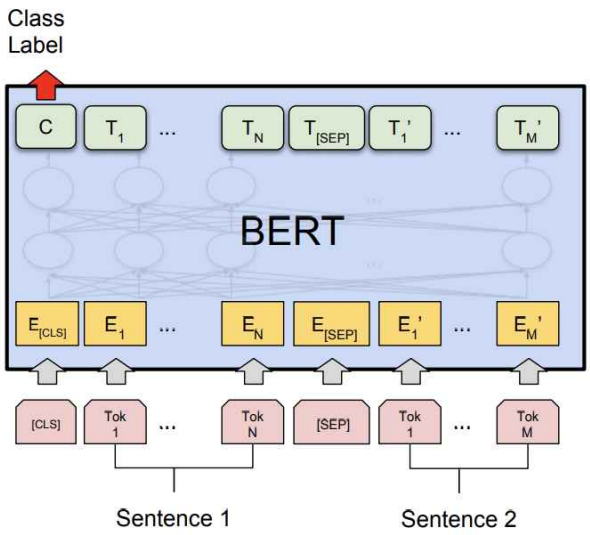
预训练的语料来自于BookCorpus和English Wikipedia（只抽取文本文章，忽略列表、表格和标题）。这里作者指出：为了抽取出连续的长句子，使用文档语料而不是打乱的句子语料很关键。

其他预训练超参数

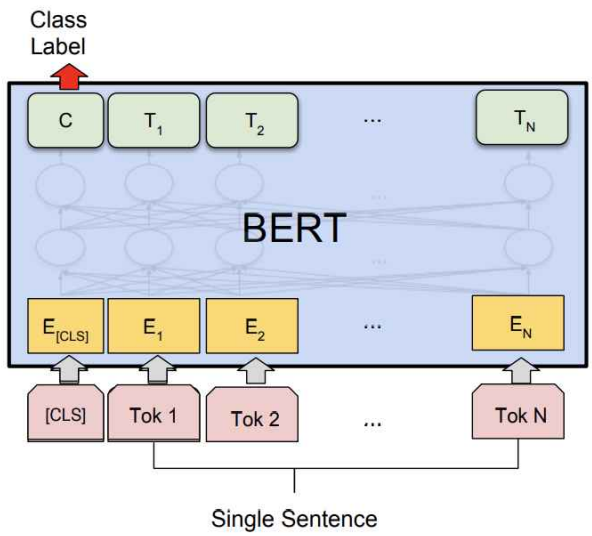
- batch_size: 256
- max_seq_len: 512（前90% steps使用128的序列长度，后10% steps使用512的序列长度）
- epochs: 40
- 优化器: Adam，$\beta_1 = 0.9$ ， $\beta_2 = 0.999$ ，L2 weight decay = 0.01
- 学习率: 1e-4，学习率线性衰减
- 激活函数: gelu (same as GPT)
- 正则: 所有层使用dropout=0.1
- 损失值: mean masked LM likelihood + mean next sentence prediction likelihood

4 微调任务

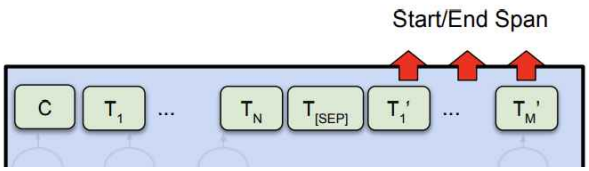
微调的方式和GPT一样，最大化减少task-specific参数。

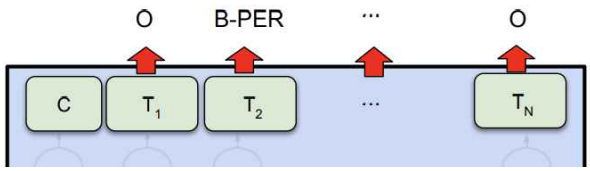


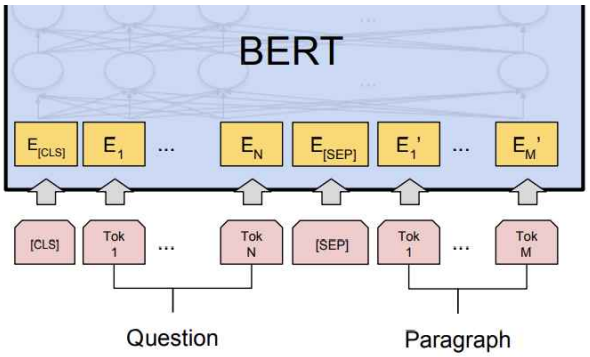
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



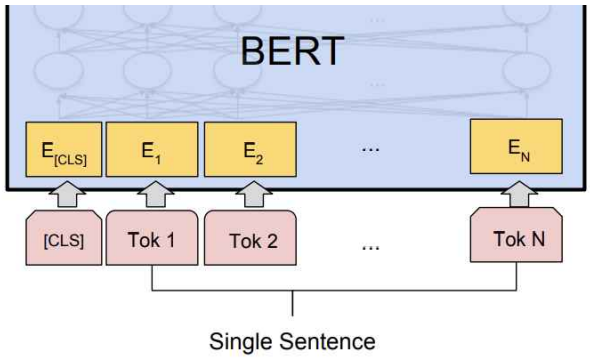
(b) Single Sentence Classification Tasks: SST-2, CoLA







(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

微调超参数和预训练一致，除了：

- batch_size: 16, 32
- learning_rate: 5e-5, 3e-5, 2e-5
- number of epochs: 2, 3, 4

5 实验结果

BERT刷新了11项NLP任务的成绩，其中8个glue任务、SQuAD 1/SQuAD 2和SWAG，实验结果对比如下：

1. GLUE（8个任务）

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0

BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

2. SQuAD（2个任务）

System	Dev		Test	
	EM	F1	EM	F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

Table 2: SQuAD 1.1 results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

System	Dev		Test	
	EM	F1	EM	F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-		71.4	74.4
Ours				
$\text{BERT}_{\text{LARGE}}$ (Single)	78.7	81.9	80.0	83.1

Table 3: SQuAD 2.0 results. We exclude entries that use BERT as one of their components.

3. SWAG

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
OpenAI GPT	-	78.0

$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

6 总结

BERT的核心：

- 输入mask机制
- 三层embedding输入
- 两个预训练任务：mlm、nsp
- 微调