RoBert

1. 静态Masking vs 动态Masking

原来Bert对每一个序列随机选择15%的Tokens替换成[MASK]，为了消除与下游任务的不匹配，还对这15%的Tokens进行

（1）80%的时间替换成[MASK]；

（2）10%的时间不变；

（3）10%的时间替换成其他词。

但整个训练过程，这15%的Tokens一旦被选择就不再改变，也就是说从一开始随机选择了这15%的Tokens，之后的N个epoch里都不再改变了。这就叫做静态Masking。

而RoBERTa一开始把预训练的数据复制10份，每一份都随机选择15%的Tokens进行Masking，也就是说，同样的一句话有10种不同的mask方式。然后每份数据都训练N/10个epoch。这就相当于在这N个epoch的训练中，每个序列的被mask的tokens是会变化的。这就叫做动态Masking。

1. 为什么说为了消除与下游任务的不匹配，才对15%的tokens进行概率mask？

（1）下游任务是什么？

在新的数据集上面继续训练bert

在新的任务上进行finetune（只动transformer层的参数，不动mlm层参数）

在上述两个任务上，所说的下游任务不匹配指的是：如果不mask的话意味着将要删除了这个单词。而删除了单词是无法做finetune的。

这个finetune指的是：如文本分类、ner等任务。

文本分类上会这样做：

transformer层+文本分类层，去除mlm层。更改的是transformer层参数和文本分类层参数，这就是finetune。

可见这个任务删除了词是不可以的。

1. 为什么要用80%的时间替换成mask，10%的时间不变，10%的时间替换成其他词？

首先为什么80%的时间替换mask？

正常就是要100%mask的，需要对mask的词进行预测

为什么10%的时间替换成其他词？

首先这20%的词都是不被mask的，意味着模型可以观察的到的，意味着这个可以作为信息来预测80%的mask的。加入了10%的错词，意味这使用错词拿来预测正确的词。一个可以解释的思路是：如果数据质量不佳的时候，模型预测依然准确。二是增强另外的不被mask的正确的词学到的特征更加强。

4. with NSP vs without NSP

Emmm。感觉只是对各种方法做了个实验，好就完事了。和bert的对比并不是在同样的数据集下的。。。这个没办法对比啊。不够严谨。

原本的Bert为了捕捉句子之间的关系，使用了NSP任务进行预训练，就是输入一对句子A和B，判断这两个句子是否是连续的。在训练的数据中，50%的B是A的下一个句子，50%的B是随机抽取的。

而RoBERTa去除了NSP，而是每次输入连续的多个句子，直到最大长度512（可以跨文章）。这种训练方式叫做（FULL - SENTENCES），而原来的Bert每次只输入两个句子。实验表明在MNLI这种推断句子关系的任务上RoBERTa也能有更好性能。