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。

2. 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。

3. 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。