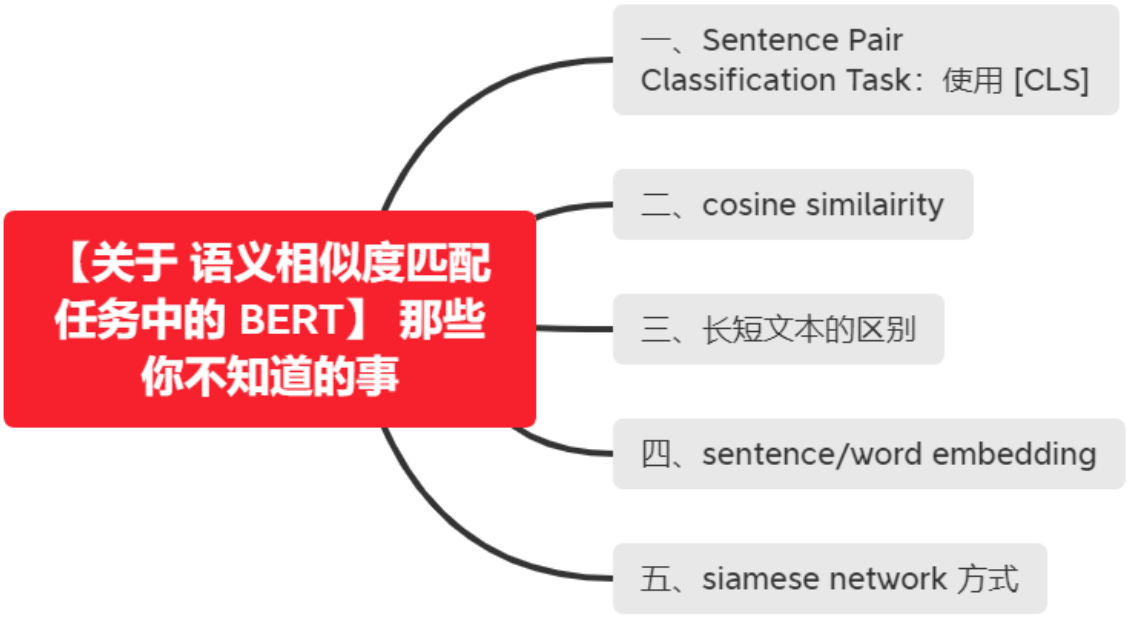
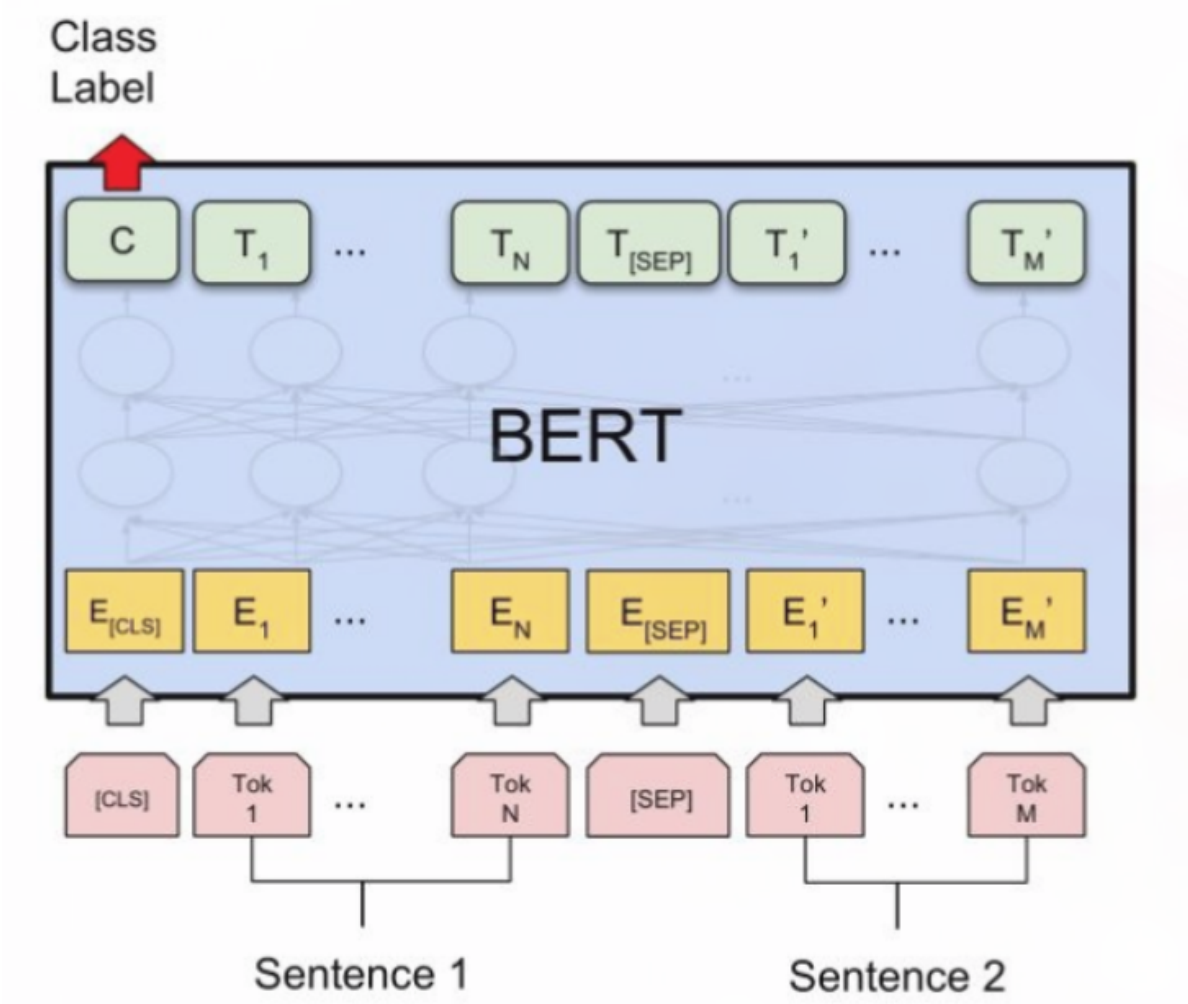
【关于 语义相似度匹配任务中的 BERT】 那些你不知道的事



一、Sentence Pair Classification Task：使用 [CLS]



- 方法：

    - 1. 将输入送入BERT前，在首部加入[CLS]，在两个句子之间加入[SEP]作为分隔；
    - 2. 取到BERT的输出（句子对的embedding），取[CLS]即可完成多分类任务/相似度计算任务；
    - 3. 取到的[CLS]对应的embedding为 c；
        - 3.1. 多分类任务，需进行：P = softmax(cW')；
        - 3.2. 相似度计算，需进行：P = sigmoid(cW')；
    - 4. 计算各自所需的loss；
- 解析：c 可一定程度表示整个句子的语义
- 举例
    - 原文中有提到“ The final hidden state (i.e., output of Transformer) corresponding to this token is used as the aggregate sequence representation for classification tasks.”这句话中的“this token”就是CLS位。

二、cosine similairity

- 方法：
    - 1. 利用 bert 生成两个句子的句向量；
    - 2. 在不finetune的情况下，计算 cosine similairty 绝对值；
- 问题：不合理的
- 原因：bert pretrain计算的cosine similairty都是很大的；
    - 如果直接以cosine similariy>0.5之类的阈值来判断相似不相似那肯定效果很差；
    - 如果用做排序，也就是cosine(a,b)>cosine(a,c)->b相较于c和a更相似，是可以用的；
- 评价指标：auc，而不是accuracy；

三、长短文本的区别

- 短文本（新闻标题）语义相似度任务：用先进的word embedding（英文fasttext/glove，中文tencent embedding）mean pooling后的效果就已经不错；
- 长文本（文章）：用simhash这种纯词频统计的完全没语言模型的简单方法也可以；

四、sentence/word embedding

bert pretrain模型直接拿来用作 sentence embedding效果甚至不如word embedding，cls的emebdding效果最差（也就是pooled output）。把所有普通token embedding做pooling勉强能用（这个也是开源项目bert-as-service的默认做法），但也不会比word embedding更好。

五、siamese network 方式

- 思路：除了直接使用bert的句对匹配之外，还可以只用bert来对每个句子求embedding，再通过向Siamese Network这样的经典模式去求相似度；
- 用siamese的方式训练bert，上层通过cosine做判别，能够让bert学习到一种适用于cosine作为最终相似度判别的sentence embedding，效果优于word embedding，但因为缺少sentence pair之间的特征交互，比原始bert sentence pair fine tune还是要差些。

参考

1. 用BERT做语义相似度匹配任务：计算相似度的方式