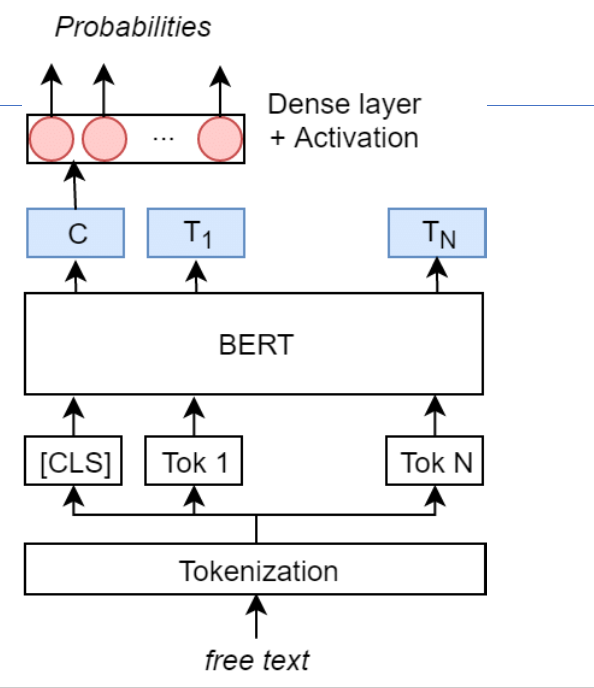
关于NLU的理解

NLU的目的在于生成intend和sloty

intend是一个文本分类任务，更准确的来说是一个多标签文本分类任务。它与传统的不同点在于一方面它是多标签的，传统的是多类的。多类的在于向量采用的是一个softmax的激活函数，loss是交叉熵的。

多标签是将向量采用的sigmoid的激活函数，loss则是二值熵的函数。



Intend的获取则是将文本或者多轮对话的文本，先获取到token，然后通过Bert预训练模型，获得它的向量，其中cls处生成的是全局向量，然后通过全连接层，一个sigmoid的激活函数，loss采用的是二值熵，就可以处理多标签模型了。

Bert

它是一个预训练模型，是一个双向的transformer模型。它有两个预训练任务。它采用了wordpieces代替了token，一方面它可以减少词汇表的大小，同时增加了对于每个单词数据的可用性，也可以防止OOV。

词级别的mask任务

它对输入采用15%的mask，任务则是预测mask。为了抗噪声检测,而其中的80%则是用mask替代，10%随机替代，10%的token则是不变。

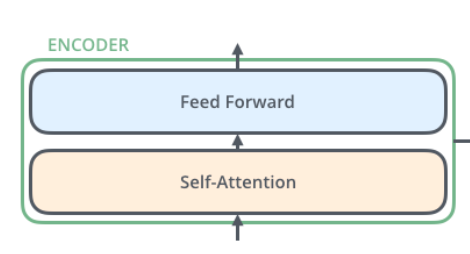
句子级别的NSP任务

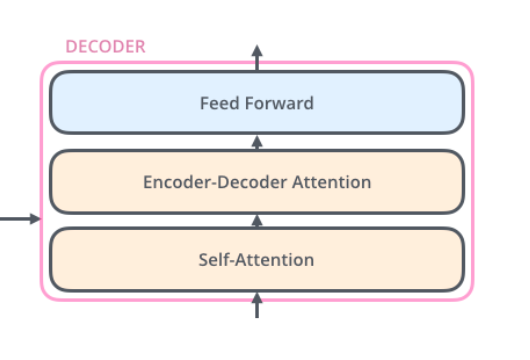
50%的两个句子是正常顺序的，50%的则是随机替代下一句。它的任务则是预测下一句是否是正常下一句。

Transformer

它是由多个Encoders和Decoders组成

Encoder





它的核心在于self-Attention

Self-Attention它的理解在于句子中每个词都有重要性，Attention则是每个词都有自己权重。它通过WQ,WK,WV矩阵获得Q,K,V。通过计算QK的相似性，然后经过标准化，获得权重，然后与V相乘。

Multi-Head-Attention,多头的理解在于一个句中词通过不同的方式可以有不同的向量表示，也就代表着多头。其余的与Self-Attention相同。

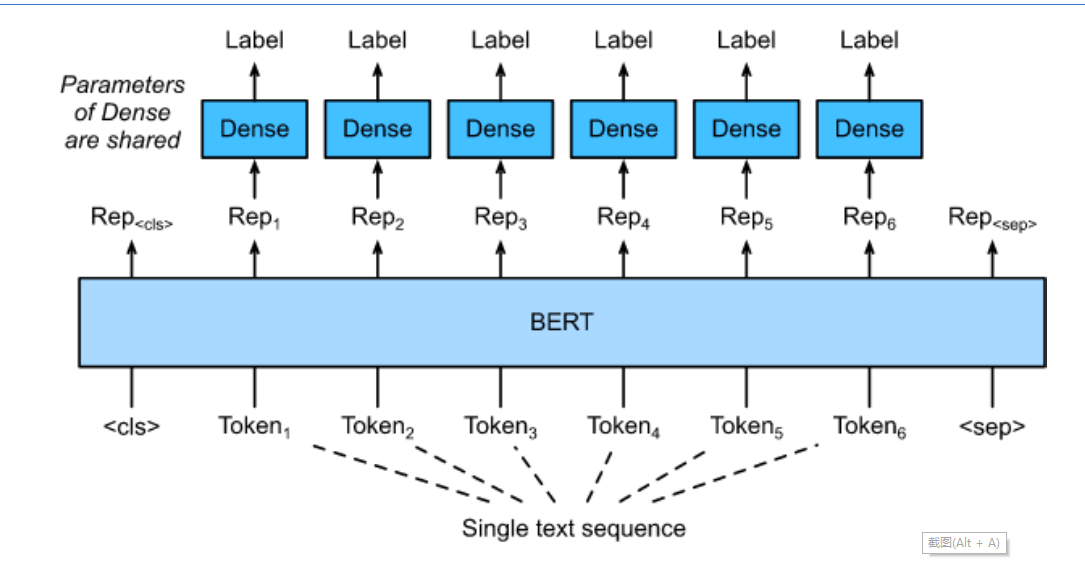
通过Self\_Attention获得向量，然后经过一个正规化，然后喂养给一个全连接层。

encoder与decoder不同在于encoder的输出作为K传递给decoder，用来计算encoder与decoder的attention。

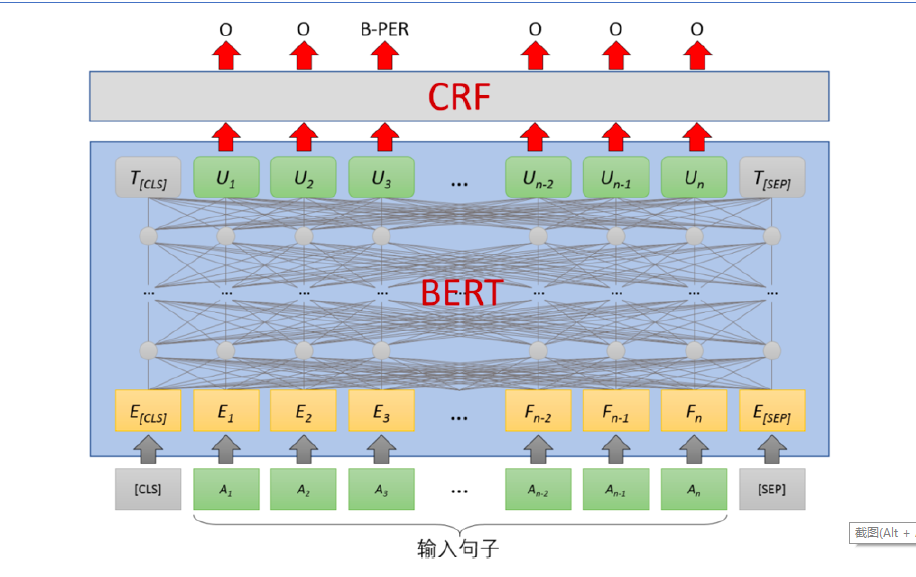
由于句子中不同位置的注意力计算起来是相同的，但是这显然不合理，所以里面加入了Positional Encoding。

Slot

它的任务是一个词性识别任务。



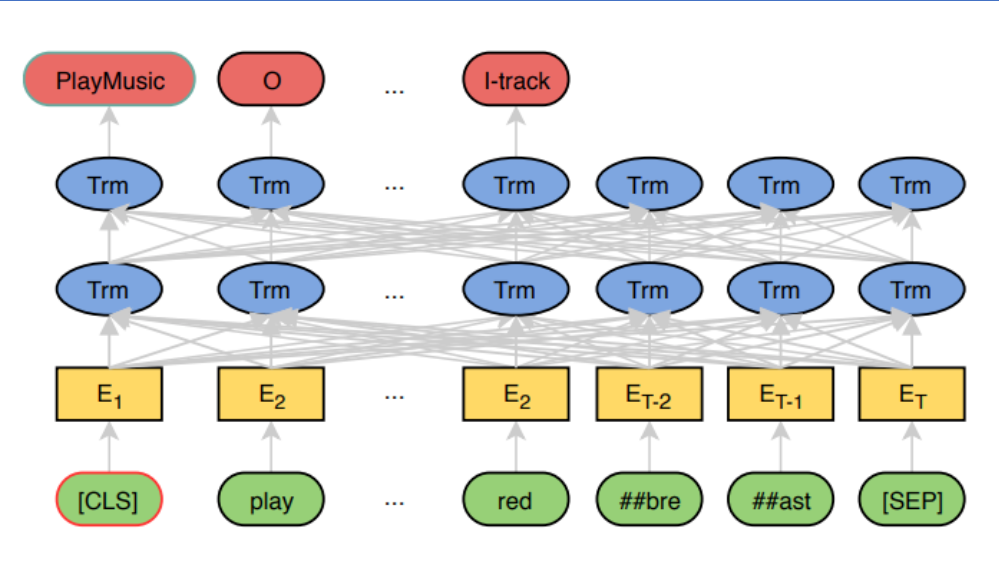
它也可以用过Bert获得向量，然后每个token通过一个全连接层,来预测label。



也可以加入一个CRF

Joint Intent-Slot

它的结构



它是同时分别将CLS和token采用sigmoid，其中CLS预测intend,token来预测词性识别。我们需要将这两个任务的loss加起来作为全局的loss来做反向传播处理。

关于作业,自身的代码能力还是需要继续学习，尤其是数据loader这部分。