**一、主要思想**

1、用Standford constituency parser的方法从一个caption中分别提出skeleton的部分（名词）和skeleton中每一个noun word的attribute部分（形容每一个名词的sequential words）；

对skeleton和attribute分别学习一个LSTM；

2、Skel-LSTM是soft-attention的LSTM；

Attr-LSTM的输入是C:\Users\YanJack\AppData\Local\Temp\enhtmlclip\Image(13).png，

其中zT是Skel在T时刻的soft-attention，

训练时sT是Skel在T时刻的ground\_truth,测试时sT是Skel在T时刻的输出，

hT是Skel在T时刻的隐层；

3、Attr-LSTM中输入了Skel-LSTM在t时刻的attention（zt）

C:\Users\YanJack\AppData\Local\Temp\enhtmlclip\Image(14).png

C:\Users\YanJack\AppData\Local\Temp\enhtmlclip\Image(15).png

对此attention的计算方式进行改进，引入Pij把原来的pre-word alpha改进为post-word alpha；

C:\Users\YanJack\AppData\Local\Temp\enhtmlclip\Image(16).png

其中Pattend是Skel-LSTM在t-1时刻的输出，是vocabulary size维度的概率，表示下一个最可能的skel-word；

Pij是vij经过Skel-LSTM在t-1时刻的输出，是vocabulary size维度的概率，表示每一个vij最可能对应的下一个word；

只有当skel-word和vij对应的word一致时，相应的vij的alpha最大，

Z是归一化因子，使alpha对ij求和为1；

4、测试阶段引入r参数控制句子长度，与Beam Search结合使用；

C:\Users\YanJack\AppData\Local\Temp\enhtmlclip\Image(17).png

训练阶段照常，测试阶段，l表示句子长度，每生成一个词，在该词的对数概率上加一个r，共加l个r，

若r为正，则倾向于长句的最终‘句子概率'越大，反之，则不倾向于得到长句，

在Beam Search的候选句子中比较‘句子概率’，得到生成句子；

**二、idea**

1、post-word alpha能否用于Skel-LSTM呢？以及是否有理由这样做呢？

2、LSTM本身倾向于短句，引入normalization可以改善，本文用r控制句子长度的方法与normalization结合理论上似乎不work，那么这种方法没有考虑到LSTM本身倾向于短句的特性？