## Text Summarization-OOV & Word Repetition

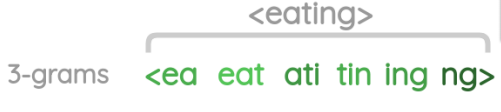
1. **Out of Vocabulary (OOV)**

OOV是超出词表的一些生僻词，对于这些不在词表中的词需要进行一些处理。对于英文，除了增大词表外，还有种主流思想，将单词拆分成子部分，即subword。

1. **Subword**

有以下主流方法：

* 1. Sub-word generation



根据n-gram将单词循环拆分成子词，由于这样会生成很多子词，因此加上一个哈希函数，将n-gram子词映射到一个1到B之间的整数。

* 1. Byte Pair Encoding(BPE)

基于词频的方法，确定期望的subword词表⼤⼩，将所有单词拆分为字符序列并在末尾添加后缀“\w”，每次循环寻找出现频率最大的⼀个连续字节对，将其合并为一个新的subword，直至达到设定的subword词表⼤⼩或下⼀个最⾼ 频的字节对出现频率为1。

* 1. WordPiece

wordPiece和BPE类似，确定期望的subword词表⼤⼩，将所有单词拆分为字符序列并在末尾添加后缀“\w”，但是WordPiece需要在训练集上训练一个语言模型，每次挑选能最大化减少loss的subword，直至达到设定的subword词表⼤⼩或概率增量低于某⼀阈值。

* 1. Unigram Language Model

该模型需根据给定词序列优化下⼀个subword出现的概率，计算每个subword的损失，基于损失对subword排序并保留前X%。为了避免OOV，建议保留字符级的单元。

1. **Pointer-Generator Network**

上面将的是解决OOV的常用方法，这一节看看从模型层面如何解决OOV。

* 1. Pointer Network(Ptr-Net)

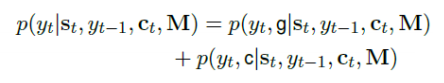
Ptr-Net为了解决OOV问题，将生成任务变成了抽取任务，抽取原文种的片段作为输出结果。



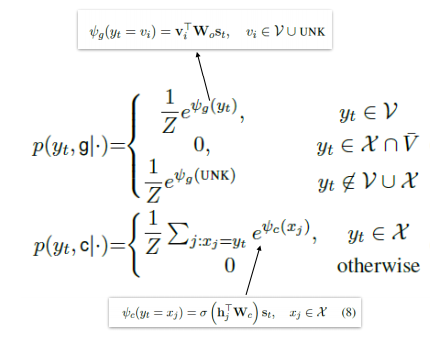
在进行attention时，直接对attention分布做softmax找出原文中合适的词。

* 1. CopyNet

CopyNet将vocabulary分布和attention分布进行相加得到最终的生成分布。



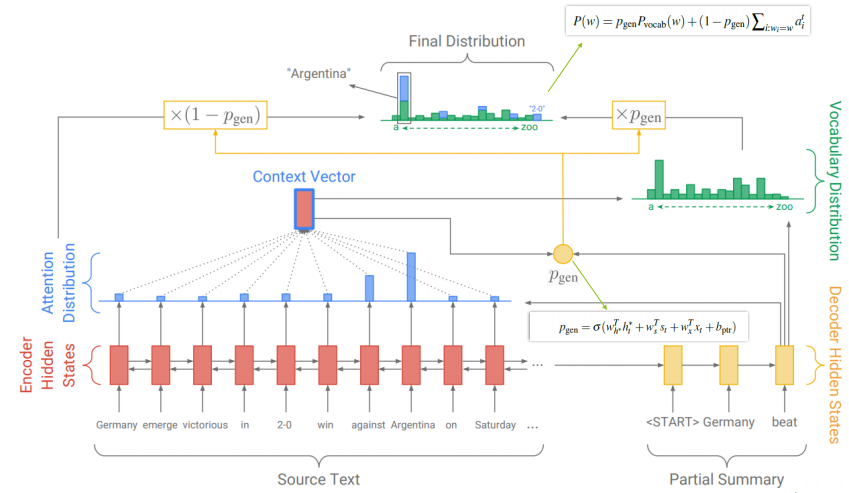
其中，和 需要根据是属于词表、原文等情况分别带入公式计算。



* 1. Pointer-Generator Network

Pointer-Generator Network同样是将vocabulary分布和attention分布进行相加得到最终的生成分布，但是这两个分布的权重的通过模型学习出来的。



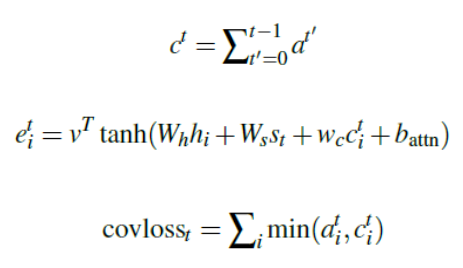


权重是通过 计算得出的。



1. **Word Repetition**

Pointer-Generator Network采用了Coverage机制来解决重复生成的问题。与之前的方法类似，都是在损失函数中，将重复关注的地方加大惩罚。



**具体做法是，在时刻计算时刻的**attention attribution之和，在计算loss的时候选择之前未被关注的值。



注意，这里在计算attention score 的公式相比上式（原始式）多了这项。