**OOV问题和PGN心得理解**

OOV

在NLP问题中，OOV问题是常见的，它出现的原因主要有两大方面

1. 新词每时每刻都在不停地被人们所创建出来，没有足够数据去训练的
2. 训练的数据有拼写的问题

解决方法

1. 扩大词汇表

这会造成词汇表不断虚造增大

1. 忽略
2. 使用特殊词去替代（如果大量的词被特殊词替代，则会缺失语义信息）
3. Individual Characters+hashing
4. 检查拼写
5. Subword+Hashing
6. BPE

BPE

步骤

1.准备⾜够⼤的训练语料

2.确定期望的subword词表⼤⼩

3.将单词拆分为字符序列并在末尾添加后缀“ </ w>”，统计单词频

率。 本阶段的subword的粒度是字符。 例如，“ low”的频率为

5，那么我们将其改写为“ l o w </ w>”：5

4.统计每⼀个连续字节对的出现频率，选择最⾼频者合并成新的

subword

5.重复第4步直到达到第2步设定的subword词表⼤⼩或下⼀个最⾼

频的字节对出现频率为1

WordPiece

1.准备⾜够⼤的训练语料

2.确定期望的subword词表⼤⼩

3.将单词拆分成字符序列

4.基于第3步数据训练语⾔模型

5.从所有可能的subword单元中选择加⼊语⾔模型后能最⼤程度地

增加训练数据概率的单元作为新的单元

6.重复第5步直到达到第2步设定的subword词表⼤⼩或概率增量低

于某⼀阈值

Unigram Language Model

1.准备⾜够⼤的训练语料

2.确定期望的subword词表⼤⼩

3.给定词序列优化下⼀个词出现的概率

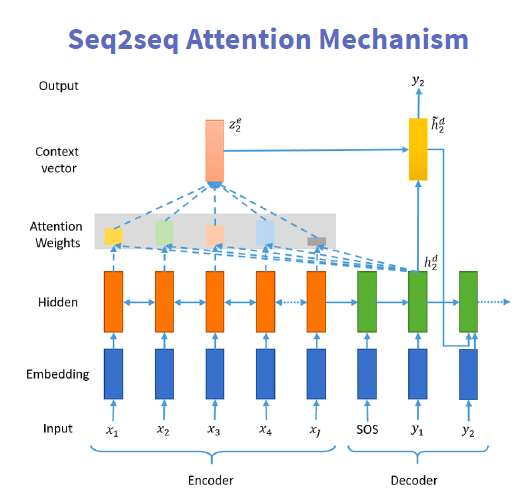
4.计算每个subword的损失

5.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

6.重复第3⾄第5步直到达到第2步设定的subword词表⼤⼩或第5步的结果不再变化

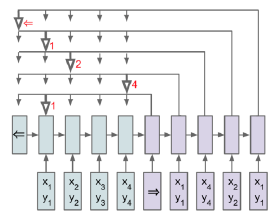
PGN

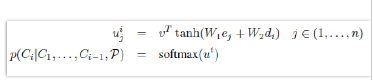
Seq2seq



上述Seq2Seq模型拥有了生成能力，它在解决摘要、翻译等问题时，主要是采用了生成的方法去解决问题。但是在遇到一些问题，我们也需要它能直接copy原文中一些重要信息展示出来，所以上述方法就没法解决。

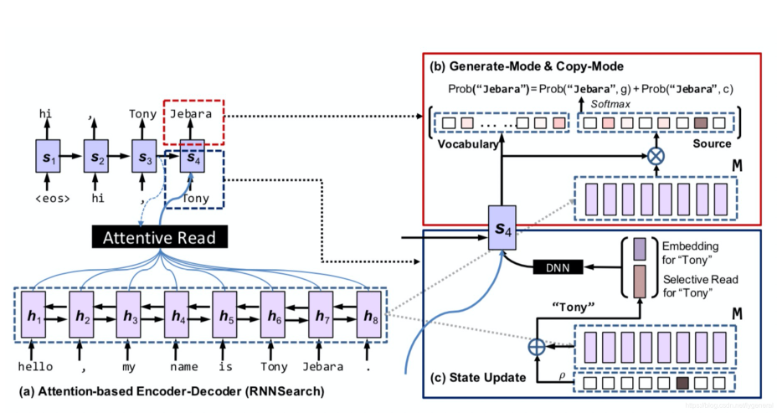
Pointer Network





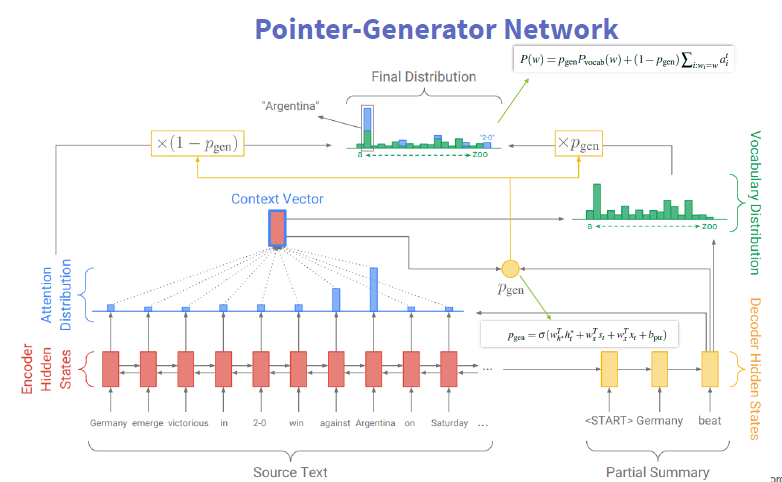
从上面公式可以看到，它是在计算attention时候，没有在softmax之后采用求和值，从而softmax之后的值可以作为一个指针替代原文中的一些重要信息。它拥有了一种copy的能力

CopyNet



CopyNet的模型可以分成左侧和右侧两个部分，左侧是一个对话模型，其实也是传统的翻译模型，由 encoder 和 decoder 两个模块构成。CopyNet有一个由概率建模的generate-mode和copy-mode。

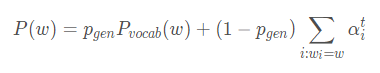
PGN



从上面可以看到，它左边部分采用了Seq2Seq模型，赋予模型生成能力，右侧采用了Pointer

Network，赋予了模型Copy能力，然后采用一个Pgen指针将他们连接起来。

1621751437(1)



这个模型也可以用来解决上述的OOV问题。