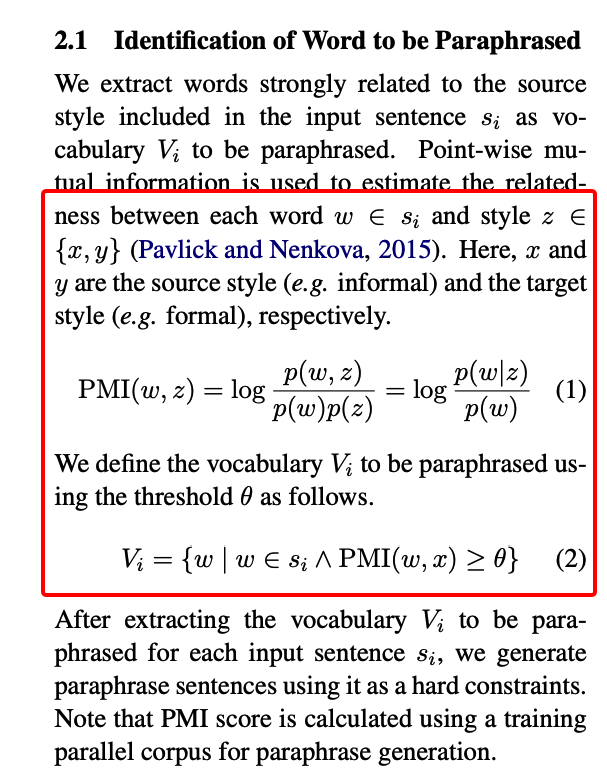
论文《Negative lexically constrained decoding for paraphrase generation》

论文来源： 2019 ACL

论文主要内容：

作者提出了一种生成paraphrase的方法，首先识别source sentence中哪些词应该被paraphrase，然后通过使用一种negative lexically constrained decode方法来强制decode时不包括这些词。作者在文本简化和风格转换任务【口语化英语转换为书面语】上验证了它们的实验效果。

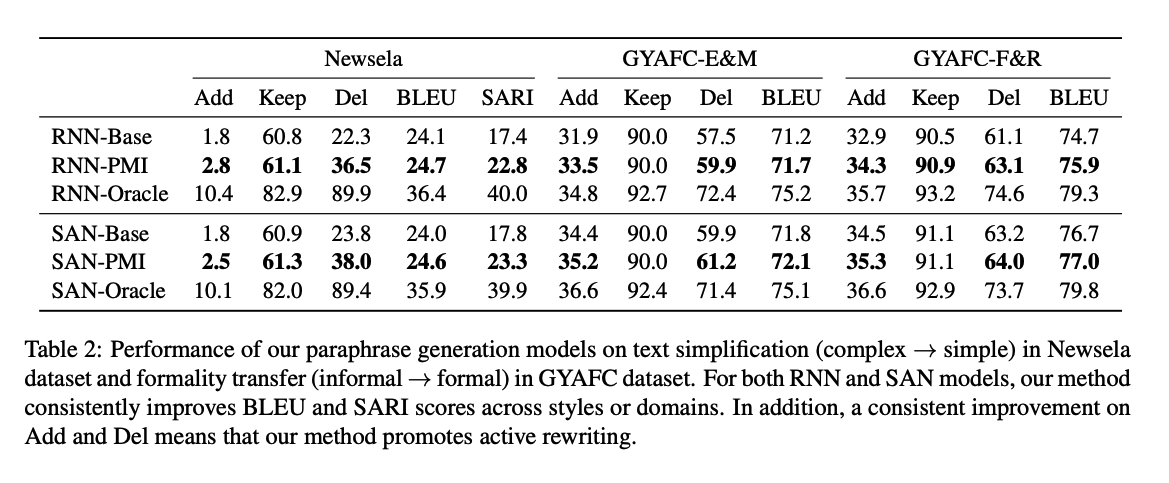
首先识别句子中哪些词要被paraphrase，作者主要采用了Pavlick2015论文里的方法，从第二个公式可以推断出作者主要是计算PMI(w,x)的分值大小，当分值满足阈值的时候，就把他加入vocabulary 中。【具体计算PMI(w,x)的方法就是，先计算p(w|x)，也就是在formal的文档中，w出现的频率，再计算p(w)，也就是w在所有文档中出现的频率】



然后是negative lexically constrained decode,来限制生成的paraphrase中没有上一步找到的词【vocabulary中的词】，这里作者采用的是Post and Vilar 2018里的方法，采用了dynamic beam allocation。

实验部分，对于lexical constrains，识别词时，作者对阈值取值{0,0.1,0.2…0.7}进行了实验。然后对于paraphrase generate model，对于encode和decode作者都使用了同一个LSTM，然后还有一个self-attention模块，self-attention模块使用了一个6层的transformer模块。

实验结果如下：



实验中的add, keep, del具体操作可能是formal transformer的一些常规操作，后续往前查看具体操作。

这篇论文是个短论文，讲的不是很详细，但是给我提供了生成paraphrase的新思路。

Pavlick2015论文里对于词的刻画，基于两个维度，一个是formality和casual，一个是complex和simple。

为了分析一个词或短语是formality，还是casual，作者使用Europarl corpus作为formal text，然后使用Switchboard corpus of informal telephone conversations 作为casual text。

为了分析一个词或短语是complex还是simple，作者采用wikipedia作为complex corpus，采用simplified wikipedia作为simple corpus。

然后可以计算每个词/短语的分值，比如要计算一个词的formality的分值，作者将formal text作为reference corpus(REF)，把formal text + casual text作为All corpus，然后可以按照下列式子来计算短语w的分值Formality(w)：

