实验报告

杨伊鸣 1200012723

构建了一个bigram的模型。尝试使用不同的平滑方式对比结果。

对train和test文件进行了预处理。

简单预处理：添加[s]和[e]（句子起始和结尾）

复杂预处理：在简单预处理的基础上，将所有数字归为一类

平滑方式尝试两种：

Add-del平滑：

P=(c+del)/(N+del\*v)

del分别取0.5和0.05。V由一元词种类数的平方估算。

Good-turin平滑：

r\*=(r+1)N(r+1)/Nr

给出现0次的n-gram分配N1/N的空间，而N0由一元词种类数的平方估算。

当r增大到Nr=0时，对更大的r不再进行平滑

最后进行归一化

结果如下：

简单预处理 正确率

del=0.5 0.9368

del=0.05 0.9432

g-t 0.9401

复杂预处理 正确率

Del=0.5 0.9373

Del=0.05 0.9439

g-t 0.943

结论：

1. 可见对数字的预处理是有效果的
2. Del较小时效果更好（比0。05更小时逐渐变差）

接下来尝试回退算法

首先对del=0.05 复杂与处理尝试加入简单的回退

Pb（W2|W1）= P(W2|W1) (c(W1W2)>0)

A\*P(W2)

结果如下

A=0.01 0.9398

A=0.001 0.9461

A=0.0001 0.9435

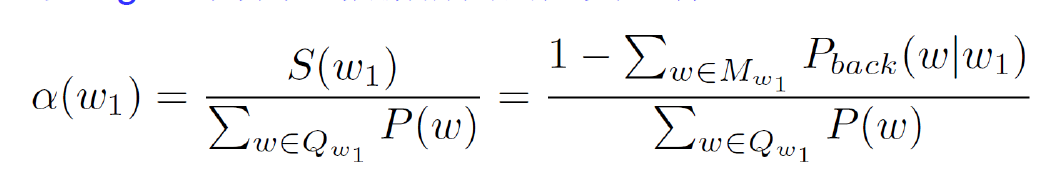
因为我们没有进行归一化的调整，因此我猜测当恰当的A使得sigma(Pb)=1时，效果最好

最后尝试在good-turing平滑的基础上katz回退：

对频数>=10的ngram，不进行平滑

对r=1-9，进行good-turing平滑

对于r=0，进行回退



但是这一步找不到在合理时间内能运行完成的实现方法，因此想以乘一个系数A做替代尝试但是效果很差，应该是方向有问题，有待继续研究。