1.任务定义

探索不同的N-gram语言模型，使用Word Perplexity(困惑度)评估其不同的性能

2.输入输出

- 输入:train.txt,train.txt
- 输出:1-gram/2-gram/3-gram模型的困惑度

3.方法描述

3.1 N-gram模型

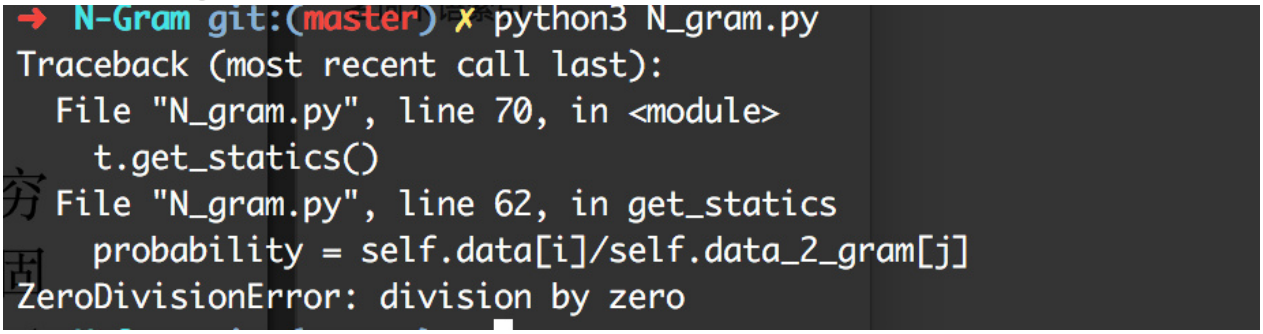
依据马尔科夫假设而建立的模型

3.2 Laplace平滑

又叫Add 1 平滑法，保证每个n-gram模型在训练中至少出现1次

4.结果分析（性能评价）

直接使用Big Gram模型，由于存在较多的零值参数，导致模型建立失败。



使用了Add 1的平滑策略，成功取得了结果，但是结果却违背了越高阶，困惑度越小，模型越好的定理。

- 1元模型得到的困惑度为4040
- 2元模型得到的困惑度为46543
- 3元模型得到的困惑度为122143

模型代码应该是存在问题的，但是没能找到问题所在。

5.源码运行环境

unix环境,python3，运行`python3 N_gram.py`

使用第三方库:

- zhon:用于过滤中文符号
- sklearn:用于划分训练集与测试集