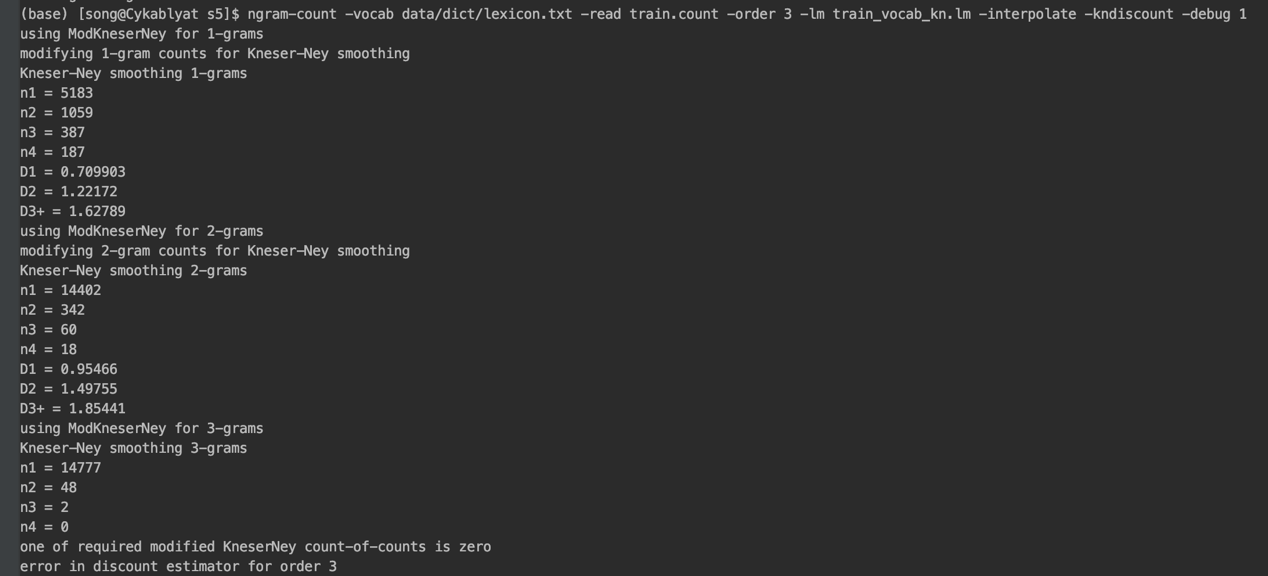
作业1：SRILM实验

基于thchd30中文语料，尝试采用modified Knser-Ney算法训练相应的语言模型并计算困惑度。

1. 语料数据清洗

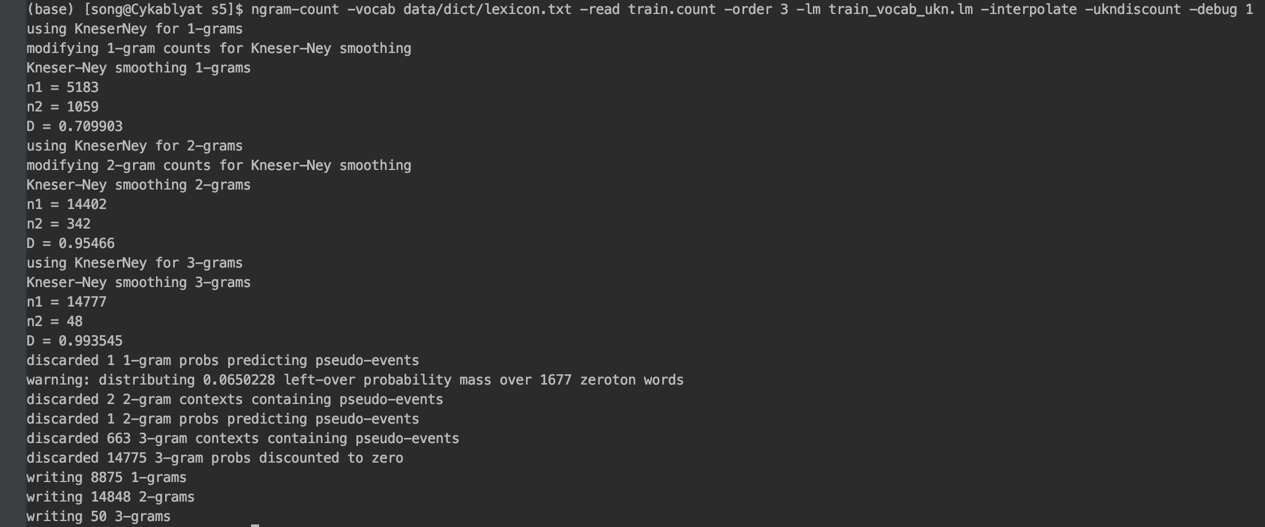
在thchs30的语料文本中有很多重复的句子，经过检测发现，每条句子都至少重复 7 次，这样会导致训练时统计ngram出现的频次低于7次的词条数目不存在，从而无法应用平滑算法。故对语料进行分词后还需进行去重才能进行训练。经过统计：train.txt 有10000条句子，去重后有750条，存放在train\_clean.txt中；test.txt 有2495条句子，去重后有250条，存放在train\_clean.txt中。

1. 语言模型训练
   1. 输入ngram-count˙指令直接训练如下图所示：

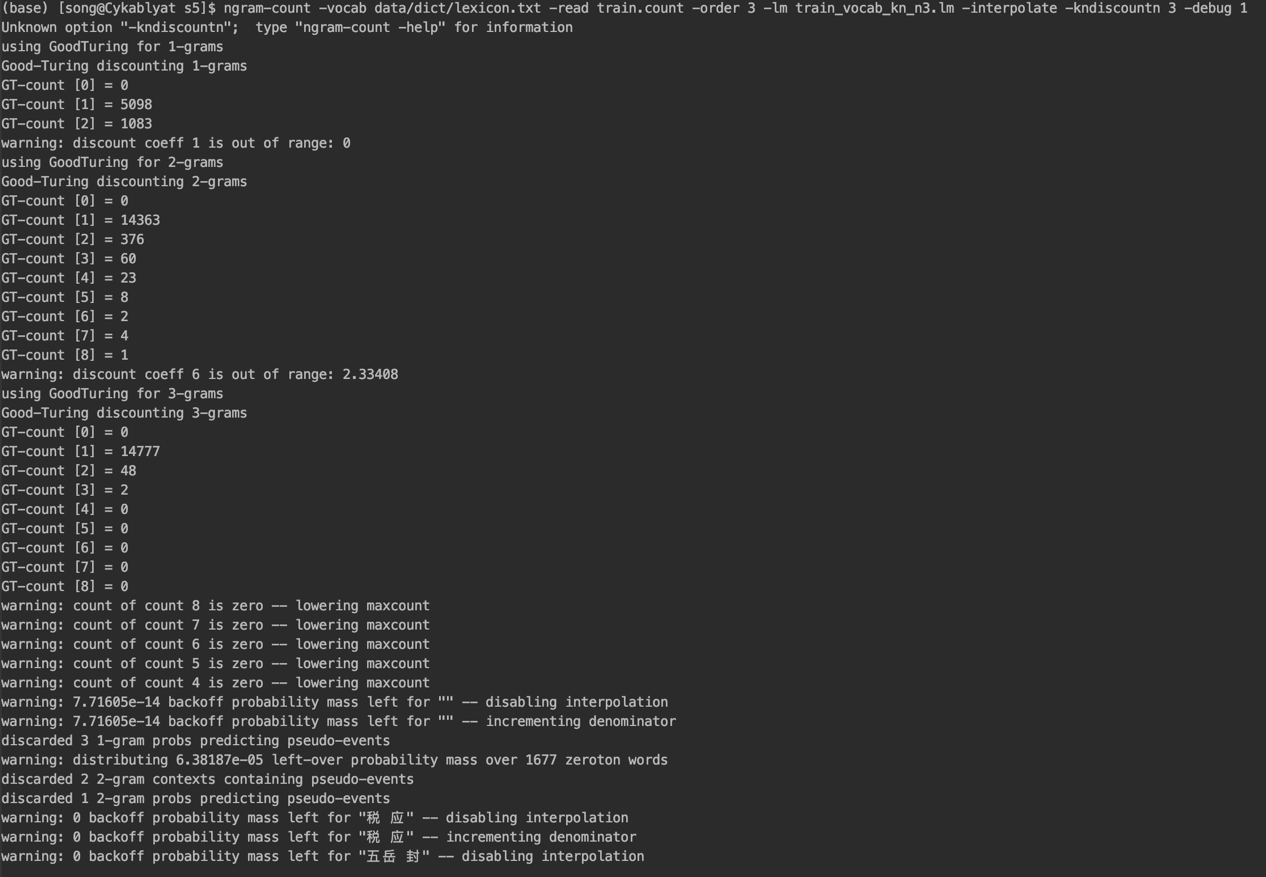


报错原因：3grams的参数n4=0，表明频次高于4次的3gram词条不存在。这主要是训练语料太少造成的。

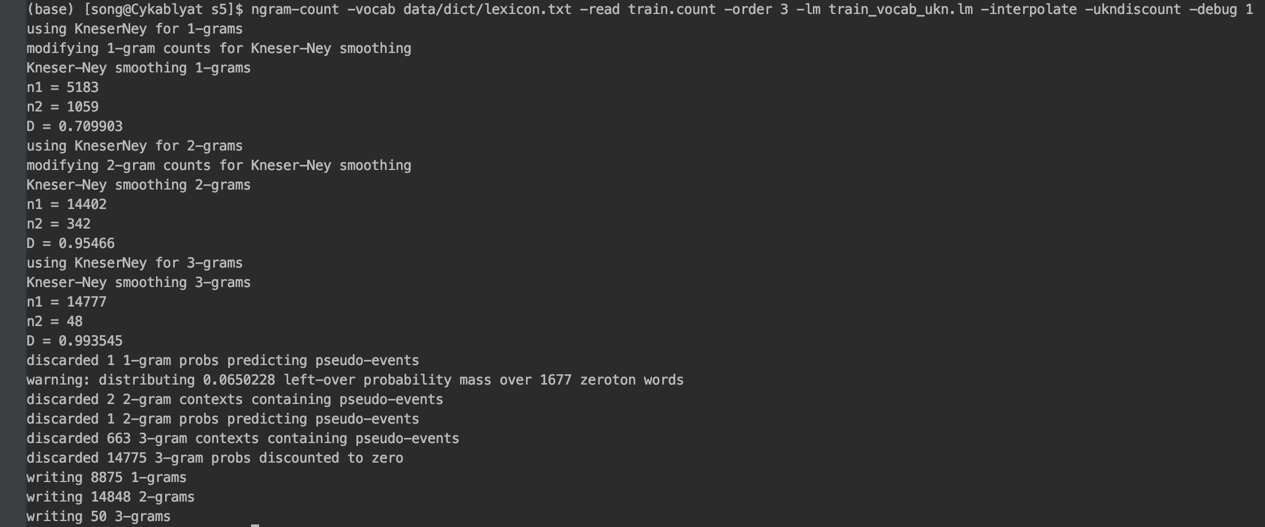
在不改变语料有解决办法：1）改用原始版本的Knser-Ney算法；2. 通过 “-kndiscountn 3” 指令，除去算法计算过程中对n4的依赖；3）参看业界常用做法，改用witten bell算法；实验日志依次展示如下：



1）原始Knser-Ney算法

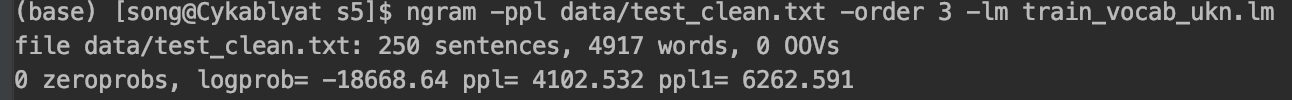


2）添加“-kndiscountn 3” 指令

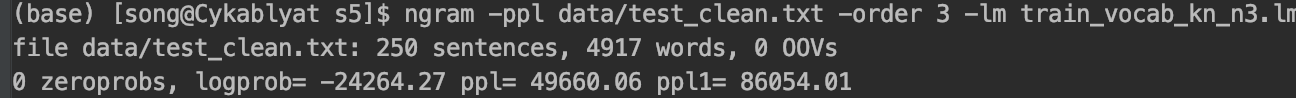


3）witten bell算法

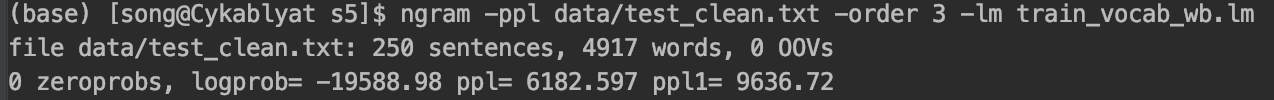
3. 根据上述过程中生成的语言模型，在测试集test\_clean.txt计算各自的困惑度：



1）原始Knser-Ney算法困惑度（ppl, ppl1）结果



2）添加“-kndiscountn 3” 指令困惑度（ppl, ppl1）结果



3）witten bell算法困惑度（ppl, ppl1）结果

结果表明，通过原始Knser-Ney算法训练得到的模型在测试集上表现最好。当然由于训练集和测试集样本较小，困惑度的比较并无多大的实际意义，只是通过一个小的实验用于展示SRILM工具包的使用方法。