N-gram 实验报告

一：实现方法

本次试验中，为了方便进行数据的读取，编写 dataset 类，其接收四个参数，分别为 mode，word，type 和 data_dir。Mode 可以是 sentence 或者 line，分别表示以句号结束作为一句话和以一行结束作为一句话；word 可以是 True 或者 False，分别表示字分割和词分割；type 可以是 True 或者 Flase，分别表示考虑词性与不考虑词性。

Adding-one 和 good-turing 的实现方法参照课件所描述。

二：测试结果

测试结果如下：

首先是按句号分割的结果：

测试类别	验证集-uni	验证集-bi	测试集-uni	测试集-bi
Adding-one 考虑词性	2182.66	10752.65	2202.23	10328.61
Adding-one 不考虑词性	1846.66	8913.88	1889.27	8614.26
Adding-one 字分割	661.59	241.22	675.04	237.99
Good-turing 考虑词性	884.88	26.99	894.75	27.74
Good-turing 不考虑词性	771.52	28.39	792.35	29.02
Good-turing 字分割	564.49	37.26	576.16	37.19

接着是按一行结束作为分割的结果：

测试类别	验证集-uni	验证集-bi	测试集-uni	测试集-bi
Adding-one 考虑词性	3092.91	14779.31	2996.97	13829.64
Adding-one 不考虑词性	2624.97	12387.96	2586.59	11690.99

Adding-one 字分割	784.79	284.48	801.49	283.51
Good-turing 考虑词性	1019.05	27.99	1012.46	29.75
Good-turing 不考虑词性	997.23	29.51	992.25	30.89
Good-turing 字分割	673.25	40.97	686.87	40.89

可以发现以下几点：

1：uni-gram 很差，无论是否考虑词性，是否进行字分割效果都很差，这是因为其原理过于简单，没有考虑到词的上下文关系。

2：good-turing 下的 bi-gram 可以达到非常好的效果，这是因为首先 bi-gram 相比 uni-gram 在原理上有了加深，且 good-turing 的平滑方法使得频繁出现的搭配的概率并不会因为未出现词对而下降态度，故最终效果很好。

3：按句号分句要比按行分句效果要好，这其实很直白，按照句号分句是最正确，合理的方式。

4：字分割对于 adding-one 的 uni-gram 和 bi-gram 以及 good-turing 的 uni-gram 都起到了正面作用，而对于 good-turing 的 bi-gram 起到了负面作用。

三：对比分析

对比分析中，我们选用一个句子进行分析：

这/r 是/v 山东省/ns 第一/m 家/q 大型/b 仓储式/b 、/w 会员制/n 物流/vn 和/c 配送/v 中心/n 。/w

测试时，选取按照句号分割，词分割，考虑词性。

首先是 adding-one 的结果：

首先是 uni-gram：

这	是	山东省	第一	家	大型	仓储式	会员制	物流	和
0.0027	0.0083	2e-5	7e-4	4e-4	1e-4	2e-6	1e-6	1e-6	0.009
配送	中心	。							
9.6e-6	2e-4	0.031							

接下来是 bi-gram：（表示以这个词结束的词对的概率）

这	是	山东省	第一	家	大型	仓储式	会员制	物流	和
0.008	0.0078	3.2e-5	3.6e-5	2e-4	5.4e-5	1.8e-5	1.4e-5	1.8e-5	1.8e-5
配送	中心	。							
1.6e-5	5.5e-5	1e-4							

接着是 good-turing 的结果：

首先是 uni-gram：

这	是	山东省	第一	家	大型	仓储式	会员制	物流	和
0.0052	0.0157	4e-5	0.00135	0.00089	0.0003	1e-6	4e-6	4e-6	0.017
配送	中心	。							
1.4e-5	0.0004	0.059							

接下来是 bi-gram：（表示以这个词结束的词对的概率）

这	是	山东省	第一	家	大型	仓储式	会员制	物流	和
0.056	0.2667	0.0001	0.25	0.0272	0.006	0.025	4e-4	1e-6	1e-6
配送	中心	。							
0.0006	0.667	0.0802							

可以看出，对于 unigram 来讲，采用 adding-one 或者 good-turing 方法的效果都很差，good-turing 的效果略优于 adding-one，这是因为 good-turing 对于训练集中未出现的词的概率有着更科学的计算方式。

对于 bigram 来讲，采用 good-turing 的效果要远好于 adding-one。分析可知，对于 bi-gram 来讲，如果简单通过 adding-one 来进行 smoothing，会导致原本出现概率很高的词对概率大幅度降低。如上文例子中，[山东省，第一]在 good-turing 中出现的概率为 0.25，但在 adding-one 中却只有 3.6e-5，这就验证了上文对于两方法优缺点的分析。此外，还有[配送，中心]，这个在现实生活中是非常高频的短语，可以看出在 good-turing 中其概率为 0.667，而在 adding-one 中仅仅只有 5.5e-5，再次验证了分析。