****

深度学习与自然语言处理

|  |  |
| --- | --- |
| 学院名称 | 自动化科学与电气工程学院 |
| 专业名称 | 自动化 |
| 学生姓名 | 张一凡 |
| 指导教师 | 秦曾昌 |

2023年 3月



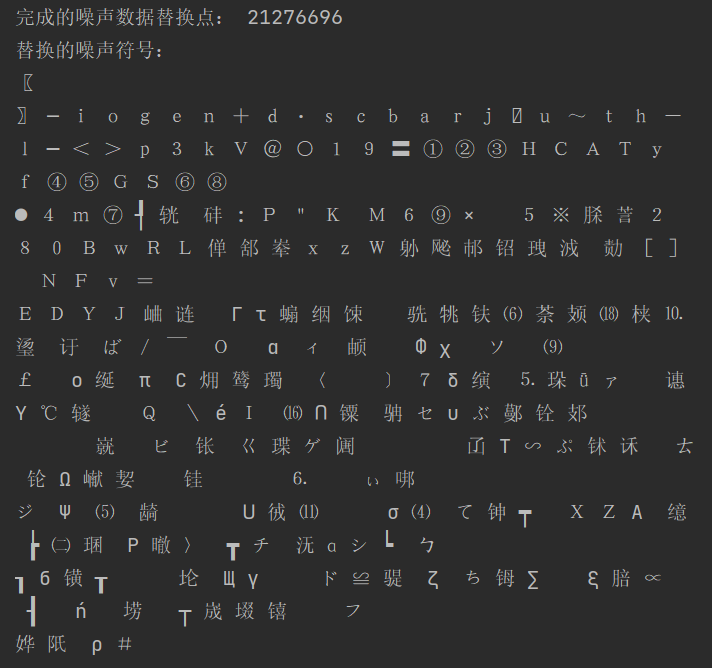
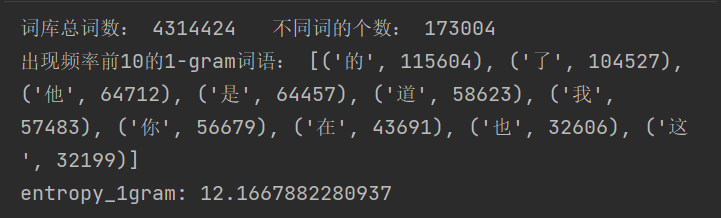
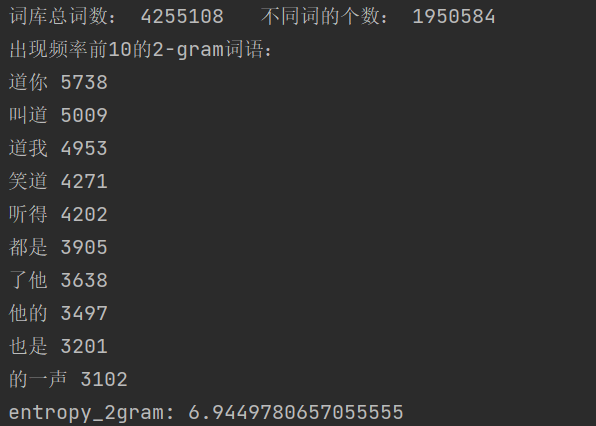
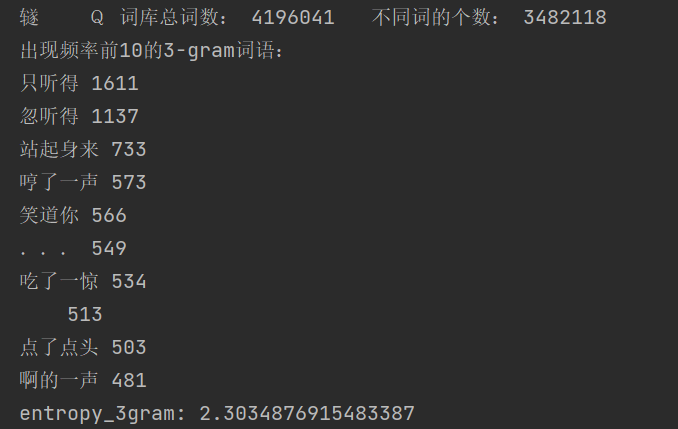


深度学习与自然语言处理第一次大作业-中文语料库平均信息熵计算

1. 实验内容  
   参考Entropy\_of\_English\_PeterBrown这篇文章来计算中文语料库的平均信息熵
2. 实验数据  
   实验数据集包括金庸的16部武侠小说，文件格式为.txt

实验步骤  
首先对于实验数据进行预处理，因为实验要求我们计算的是中文字符的平均信息熵，因此我们应该将除中文字符之外的字符（如英文字符、标点符号、噪声等）进行剔除。这里我们使用到的是正则表达式来进行提取，将\u4E00-\u9FA5之间的字符进行提取则为我们需要的中文字符，同时将英文字符（[a-z A-Z 0-9]）以及标点符号剔除。将处理后的文本每一行读取出来成为一个自然段进行存储，最后存入一个.txt文件便于数据的读入。需要注意的是，对于噪声数据，如一些出现频次极低的字符，我们可判定为噪声字符，这里我们直接替换成空字符，并在结果中将我们替换的噪声数据进行列举。  
其次，是我们通过阅读信息熵这篇文献所获得的如何计算信息熵，定义公式为



1. 实验算法设计  
   我们采用上述信息熵的计算公式，通过N-Gram这一语言模型对文本序列进行划分。该语言模型就是用来计算一个句子的概率，判断这句话是否被认定为正常句子的概率，给定一个词语序列： ，则它的概率可以表示为：   
   将上面的链式规则计算P(W)可以写为：   
   但是事实上，如果用以上模型来计算条件概率会出现参数空间过大，计算资源严重不足以及数据稀疏严重无法提供有效的信息熵计算等问题。因而我们采用N-Gram语言模型，引入马尔科夫假设，随意一个词语出现的概率只与它前面的有限的一个或者几个词语有关，上面的计算方式是通过马尔科夫假设得出的，马尔科夫假设是指假设第wi个词语只与它前面的k个词语相关，这样我们就得到前面的条件概率计算简化如下：   
   这样计算的P(W)简化如下：  
      
   当k=0时，这个时候对应的模型叫做一元模型，即wi与它前面的0个词相关，即wi不与任何词相关，每个词语都是相互独立的，P(W)计算如下：  
      
   当k=1时，对应的模型叫做二元模型，此时wi与它前面一个词相关，P(W)计算如下：   
   以此类推，可以让k=2，叫做trigrams，4-grams，5-grams，当k=N-1，模型成为n元模型，即N-grams。
2. 实验结果及分析  
   本实验分别采用k=1,2,3时的N-Gram模型进行对比分析，实验结果如下：  
   噪声数据替换：  
     
   1-Gram结果：  
     
   2-Grams结果：  
     
   3-Grams结果：  
     
   通过结果分析，当k值取得越大，即考虑前后词语关联的程度越大时，不同词出现个数增加，因而信息熵越小，因为分词之后得到文本的词组分布变得越简单，使得固定的词数量越多，固定的词能减少由字或者短词打乱文章的机会，使得文章变得更加有序，减少了由字组成词和组成句子的不确定性，因而降低了文本所含信息的信息量，也就是信息熵的大小。