



学 期 2021-2022 (2)

北京航空航天大学
BEIHANG UNIVERSITY

深度学习与自然语言处理第一次作业

中文信息熵的计算

院（系）名称	自动化科学与电气工程学院
专业名称	电子信息
学生姓名	蔡尽云
学号	ZY2103501
指导老师	秦曾昌

2022 年 4 月

1 实验原理

1.1 信息熵

香农从热力学当中借鉴提出信息熵的概念，解决了对信息量化度量的问题。其定义为：

$$H(x) = - \sum_{x \in \mathcal{X}} P(x) \log P(x)$$

信息熵的三个性质：

- 1、单调性，发生概率越高的事件，其携带的信息量越低；
- 2、非负性，信息熵可以看作作为一种广度量，非负性是一种合理的必然；
- 3、累加性，即多随机事件同时发生存在的总不确定性的量度是可以表示为各事件不确定性的量度的和，这也是广度量的一种体现。

1.2 语言模型

对于自然语言相关的问题，比如机器翻译，最重要的问题就是文本的序列有时候不是符合我们人类的使用习惯，语言模型就是用于评估文本序列符合人类语言使用习惯程度的模型。

当前的语言模型是以统计学为基础的统计语言模型，统计语言模型是基于预先人为收集的大规模语料数据，以真实的人类语言为标准，预测文本序列在语料库中可能出现的概率，并以此概率去判断文本是否“合法”，是否能被人所理解。

假定 S 表示某个有意义的句子，由一连串特定顺序排列的词 $\omega_1, \omega_2, \omega_3, \dots, \omega_n$ 组成这里 n 是句子长度。现在我们想知道 S 在文本中出现的可能性，即：

$$p(s) = p(\omega_1, \omega_2, \omega_3, \omega_4 \cdots \omega_n)$$

利用条件概率公式：

$$p(\omega_1, \omega_2, \omega_3, \omega_4 \cdots \omega_n) = p(\omega_1)p(\omega_2|\omega_1) \cdots p(\omega_n|\omega_1, \omega_2, \cdots \omega_{n-1})$$

当计算 $p(\omega_1)$ ，仅存在一个参数；计算 $p(\omega_2|\omega_1)$ ，存在两个参数，以此类推，难算，所以马尔可夫提出一种假设：假设 ω_i 出现的概率只与前面 $N-1$ 个词相关，当 $N=2$ 时，就是二元模型， $N=3$ 就是三元模型。

1.3 jieba 分词系统

1、jieba.cut：给定中文字符串，分解后返回一个迭代器，需要用 for 循环访问。


2、jieba.cut_for_search：该方法和 cut 一样，分解后返回一个迭代器，需要用 for 循环访问。不过它是搜索引擎模式，在精确模式的基础上，对长词再次切分，提高召回率，适合用

于搜索引擎分词。

3、jieba.lcut: 和 jieba.cut 使用方法一样, 不过返回的是列表。

2 实验设计

1、语料库来自 16 部金庸小说。对数据集进行预处理, 删除所有的隐藏符号, 标点符号, 以及非中文字符, 生成合并所有小说的待处理文本 preprocess.txt。

 白马啸西风.txt	2011/10/15 0:05	文本文档	146 KB
 碧血剑.txt	2011/10/15 0:06	文本文档	963 KB
 飞狐外传.txt	2011/10/14 23:56	文本文档	869 KB
 连城诀.txt	2011/10/14 23:56	文本文档	463 KB
 鹿鼎记.txt	2011/10/14 23:58	文本文档	2,446 KB
 三十三剑客图.txt	2011/10/14 23:58	文本文档	124 KB
 射雕英雄传.txt	2011/10/15 0:00	文本文档	1,824 KB
 神雕侠侣.txt	2011/10/15 0:00	文本文档	1,899 KB
 书剑恩仇录.txt	2011/10/15 0:01	文本文档	1,010 KB
 天龙八部.txt	2011/10/15 0:01	文本文档	2,412 KB
 侠客行.txt	2011/10/15 0:02	文本文档	733 KB
 笑傲江湖.txt	2011/10/15 0:03	文本文档	1,931 KB
 雪山飞狐.txt	2011/10/15 0:03	文本文档	267 KB
 倚天屠龙记.txt	2011/10/15 0:04	文本文档	1,904 KB
 鸳鸯刀.txt	2011/10/15 0:04	文本文档	75 KB
 越女剑.txt	2011/10/15 0:04	文本文档	35 KB

2、分别以中文字和词为单位, 依次使用一元模型、二元模型和三元模型计算中文信息熵, 同时计算语料库的字/词总数及运行时间:

- 1) 一元模型 Unigram: $P(w_i)$ 等于 word 出现的次数除以 word 总数。
- 2) 二元模型 Bigram: $P(w_i|w_{i-1}^{i-1})$ 等于 w_iw_{i-1} 总数除以 w_i 总数。
- 3) 三元模型 Trigram: $P(w_i|w_{i-1}^{i-1})$ 等于 $w_iw_{i-1}w_{i-2}$ 总数除以 w_i 总数。

3 结果分析与总结

3.1 实验结果

	基于词			基于字		
	Unigram	Bigram	Trigram	Unigram	Bigram	Trigram
语料库总词数	4314303	4255009	4195966	7299572	7240278	7180984
运行时间 (秒)	106.09265	112.98539	121.43911	2.03176	9.251	16.77267
信息熵 (比特/词)	12.16669	6.94505	2.30352	9.53834	6.71575	3.93812

```
语料库总词数: 4314303  
基于中文词的一元模型信息熵为: 12.16669 比特/词  
运行时间: 106.09265 秒  
语料库总字数: 7299572  
基于中文字的一元模型信息熵为: 9.53834 比特/词  
运行时间: 2.03176 秒  
语料库总词数: 4255009  
基于中文词的二元模型信息熵为: 6.94505 比特/词  
运行时间: 112.98539 秒  
语料库总字数: 7240278  
基于中文字的三元模型信息熵为: 6.71575 比特/词  
运行时间: 9.251 秒  
语料库总词数: 4195966  
基于中文词的三元模型信息熵为: 2.30352 比特/词  
运行时间: 121.43911 秒  
语料库总字数: 7180984  
基于中文字的三元模型信息熵为: 3.93812 比特/词  
运行时间: 16.77267 秒
```

3.2 代码链接

见 <https://github.com/ErrricCai/DL-NLP/tree/main/HW1>

代码参考: https://blog.csdn.net/qq_40412713/article/details/115742092

https://blog.csdn.net/weixin_50891266/article/details/115723958?spm=1001.2014.3001.5502