



## 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

## 一、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

## 二、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

## 三、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

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

## 四、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：  $S = W_1, W_2, \dots, W_K$  ， 则它的概率可以表示为：

$$P(S) = P(W_1, W_2, \dots, W_K) = p(W_1)P(W_2 | W_1) \dots P(W_K | W_1, W_2, \dots, W_{K-1})$$

将上面的链式规则计算  $P(W)$  可以写为：  $P(w_1 w_2 \dots w_n) = \prod_i P(w_i | w_1 w_2 \dots w_{i-1})$

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  $w_i$  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：

$$P(w_i | w_1 w_2 \dots w_{i-1}) \approx P(w_i | w_{i-k} \dots w_{i-1})$$

这样计算的  $P(W)$  简化如下：

$$P(w_1 w_2 \dots w_n) \approx \prod_i P(w_i | w_{i-k} \dots w_{i-1})$$

当  $k=0$  时，这个时候对应的模型叫做一元模型，即  $w_i$  与它前面的 0 个词相关，即  $w_i$  不与任何词相关，每个词语都是相互独立的， $P(W)$  计算如下：

$$P(w_1 w_2 \dots w_n) = \prod_i P(w_i)$$

当  $k=1$  时，对应的模型叫做二元模型，此时  $w_i$  与它前面一个词相关， $P(W)$  计算如下：

$$\begin{aligned} P(S) &= P(w_1, w_2, w_3, \dots, w_n) \\ &= P(W_1)P(W_2 | W_1)P(W_3 | W_1, W_2), \dots P(W_n | W_1, W_2, \dots, W_{n-1}) \\ &\approx P(W_1)P(W_2 | W_1)P(W_3 | W_2, W_1), \dots P(W_n | W_{n-1}, W_{n-2}) \end{aligned}$$

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。

## 五、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：

```
完成的噪声数据替换点： 21276696
替换的噪声符号：
[
] - i o g e n + d . s c b a r j [ u ~ t h -
l - < > p 3 k V @ O 1 9 = ① ② ③ H C A T y
f ④ ⑤ G S ⑥ ⑧
● 4 m ⑦ | 铤 砒 : P " K M 6 ⑨ × 5 ※ 脰 菩 2
8 0 B w R L 倅 郃 峯 x z W 舛 颯 邠 韶 瑰 洩 勣 [ ]
N F v =
E D Y J 岫 涟 Γ τ 螭 綢 餽 駢 桃 鉄 (6) 荼 頰 (18) 桡 10.
滢 汙 ば / ー O a イ 頓 Ø x ソ (9)
f o 纒 π C 炯 鸞 璫 < ) 7 δ 績 5. 琛 ū ァ 惠
Y ° 鑊 Q \ é I (16) ∩ 鏢 駟 セ u ぶ 鄭 鈴 郊
崑 ヒ 鉉 ぐ 璩 ゲ 闕 冏 T っ ぶ 鉢 诨 去
轮 Ω 嶽 契 銑 6. い 哪
ジ Ψ (5) 齧 U 佻 (11) σ (4) て 钟 T X Z A 億
| (二) 珣 P 瞰 > T チ 沅 a シ L ㄣ
1 6 鑽 T 垠 卣 γ ド ≡ 駉 ζ ち 姆 Σ ξ 賠 ∞
| n 塆 T 崧 璫 鎬 フ
婁 阡 ρ #
```

1-Gram 结果：

```
总词数： 1816392 不同词的个数： 272530
听 12254
想 6957
笑 6501
武功 6463
韦宝 6453
师父 5552
心想 5489
走 5334
瞧 4385
杀 4071
entropy_1gram: 14.359870431336757
```

2-Grams 结果：

词库总词数: 1816391 不同词的个数: 1454073  
出现频率前10的2-gram词语:  
忽听 1198  
郭芙 484  
微微笑 474  
韦宝笑 414  
恒山派 395  
件事 373  
杨逍 370  
龙女 360  
派掌门 355  
南海鳄神 294  
entropy\_2gram: 5.735864421922388

3-Grams 结果:

出现频率前10的3-gram词语:  
五岳剑派 244  
【新语丝电子 145  
新语丝电子文库 145  
电子文库】 145  
怦怦乱跳 145  
文库】金庸 117  
五岳派掌门 100  
嗤嗤声响 70  
恒山派掌门 70  
心怦怦乱 65  
entropy\_3gram: 0.6397745795836112

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。