深度学习与自然语言处理第一次作业

中文信息熵求解

惠仪
SY2202328@buaa.edu.cn

摘要

阅读《An Estimate of an Upper Bound for the Entropy of English》文献学习信息熵的理论计算，利用 17 个文本文档通过分词及单字符两种方式计算中文信息熵，对比并分析两种分词方式及不同的语言模型所得的信息熵。

1 理论方法

1.1 信息熵

对于某一事件，其发生的概率越小信息量越大。用信息量表示一个具体事件发生所带来的信息，当事件发生概率为 100%时，信息量为 0。信息熵表示结果出来之前对可能产生的信息量的期望，在随机事件中，某个事件发生的不确定度越大熵也就越大，要搞清楚时需要的信息量就越大。具体公式如下所示：

$$h(x) = -\log_2(p(x))$$

$$h(x) = -\sum p(x)\log_2(p(x))$$

其中，$p(x)$为随机事件 x 的概率。

假设有 $X = \{\dots X_{-2}, X_{-1}, X_0, X_1, X_2, \dots\}$是平稳随机过程，p 表示 X 的概率分布，$E_p$是 p 的数学期望，则 X 的熵可定义为：

$$H(X) = H(p) = -E_p\log(p(X_0|X_{-1}, X_{-2}\dots)$$

当概率分布 p 无法知道时，可以通过 p 的平稳随机过程 M 来计算熵。在合适的正则条件下，交叉熵为：

$$H(p, M) = \lim_{n\to\infty} -E_p\log M(X_0|X_{-1}, X_{-2}\dots X_{-n}) = \lim_{n\to\infty} -\frac{1}{n}E_p\log M(X_1X_2\dots X_n)$$

对任意模型 M，交叉熵 $H(p, M)$是熵 $H(p)$的上界，公式为：

$$H(p) \le H(p, M)$$

从文本压缩角度，对于任意编码方式，熵 $H(p)$是从 p 编码文本一段长字符串的每个符号的平均数的下界，表示为：

$$H(p) \le \lim_{n\to\infty} \frac{1}{n}E_p l(X_1X_2\dots X_n)$$

1.2 语言模型 n-gram model

语言模型对字符序列的概率分布进行建模。

- Unigram：当 N=1 时，模型被称为 unigram，即当前词的概率分布与给定的历史信息无关。
- bigram：当 N=2 时，模型被称为 bigram，即当前词的概率分布只与距离最近的词有关。bigram 模型类似于常见的一阶马尔可夫链，公式如下所示：

$$p(w_1|w_{i-1}) = \frac{w_i}{w_{i-1}}$$

- trigram：当 N=3 时， 模型被称为 trigram, 即当前词的概率分布与距离最近的两个词有关。公式如下：

$$p(w_i|w_{i-1,\ i-2}) = \frac{p(w_{i-2}w_{i-1}w_i)}{p(w_{i-2}w_{i-1})}$$

其中，$p(w_i|w_{i-1,\ i-2})$表示在已知前面两个字符或词语的情况下，第 i 个字符w_i出现的概率，$p(w_{i-2}w_{i-1}w_i)$是前面两个字符和第 i 个字符同时出现的概率，$p(w_{i-2}w_{i-1})$表示前面两个字符同时出现的概率。将具体的计数带入后，得到概率公式为：

$$p(w_i|w_{i-1,\ i-2}) = \frac{count(w_{i-2}w_{i-1}w_i)}{count(w_{i-2}w_{i-1})}$$

2 实验及分析

2.1 数据预处理

首先读取所有的文本文件，根据 cn_stopwords.txt 文件夹，去除整个文本库中符号及及无意义的中文，得到语料库。

```
#读stopwords
f = open('cn_stopwords.txt','r',encoding='utf-8')
stopwords = f.read().splitlines()
xx = '本书来自www.cr173.com免费txt小说下载站'
xx2 = '更多更新免费电子书请关注www.cr173.com'
# 去除符号及中文无意义stopwords
def filter(text):
    a = re.sub(xx, '', text)
    b = re.sub(xx2, '', a)
    pattern = '|'.join(stopwords)
    c = re.sub(pattern, '', b)
    d = re.sub(r'\*', '', c)
    e = re.sub(r'\.', '',d)
    f = re.sub(r'\n', '', e)
    g = re.sub(r'\u3000', '', f)
    return g
```

在分词时，分别以字符或词语为单位，构建两种方式。根据语料库得到单个的字符，根据 jieba 库的精确模式得到不重复的分词。

```
#处理文本，不同的分词方式
#以单个字为单位处理
word = [word for word in alltext]
#jieba分词，以词语为单位处理
word = jieba.lcut(alltext)
```

2.2 信息熵计算

以 Trigram 模型为例进行解释，首先计算词组的数量。

```
#数量
count_tri = {}
for i in range(len(word) - 2):
    count_tri[word[i], word[i + 1],word[i+2]] = count_tri.get((word[i], word[i + 1],word[i+2]), 0) + 1
```

之后结合条件概率计算信息熵。

```
    tr_entropy = 0
    word_num = sum(count_tri.values())
    for i in count_tri.keys():
        prob = count_tri[i]/word_num
        con_prob = count_tri[i]/count_bi[(i[0]),(i[1])]
        tr_entropy -= prob*math.log(con_prob,2)
    print("三元模型的中文信息熵为：{:.2f}".format(tr_entropy))
    return count_tri,tr_entropy
```

2.3 实验结果

1. 根据模型计算整个语料库的信息熵如表 1 所示。

表 1 语料库的信息熵

	分词	单字
一元	13.88	9.96
二元	6.18	7.02
三元	1.00	3.49

2.根据模型计算单个文本的信息熵。

通过 jieba 分词构建词语为单位的信息熵如表 2 所示。

表 2 词语为单位的信息熵

	一元	二元	三元
1	4.58	0.00	0.00
2	12.42	1.66	0.08
3	12.79	3.91	0.43
4	12.37	3.72	0.45
5	13.02	4.41	0.56
6	13.20	4.52	0.56
7	13.14	4.32	0.46

8	11.19	2.70	0.27
9	12.94	3.72	0.38
10	12.47	4.69	0.72
11	12.63	4.56	0.72
12	10.25	1.72	0.23
13	12.27	3.33	0.30
14	12.14	2.76	0.23
15	12.72	3.77	0.38
16	10.98	2.10	0.19
17	12.89	4.69	0.66

构建单字符为单位的信息熵如表 3 所示。

表 3 字符为单位的信息熵

	一元	二元	三元
1	5.53	0.39	0.03
2	10.01	4.26	0.66
3	9.76	5.58	1.86
4	9.44	5.36	1.81
5	9.71	5.97	2.27
6	9.79	6.10	2.34
7	9.76	5.95	2.19
8	9.23	4.07	1.21
9	9.76	5.66	1.79
10	9.56	6.04	2.37
11	9.52	5.83	2.35
12	8.78	3.10	0.85
13	9.52	5.07	1.63
14	9.50	4.79	1.30
15	9.63	5.55	1.86
16	9.21	3.65	0.90
17	9.67	6.00	2.40

2.3 实验分析

1. 从整个语料库的结果可以看出，不论是分词还是以字为单位的方式，一元、二元、三元模型的信息熵递减，表明以一个单位来预测下一个单位的信息不确定度最大，当前序提供一定的信息时更容易预测。

2. 在整个语料库进行比较时，以词语为单位的一元模型信息熵要高于单字，分析原因是词语的搭配方式及应用文章中的语义环境更丰富，因此在以一个词语预测时比单个字的不确定度要大。当前面的词语越多时，语义的信息越多，不确定度减小，因此二元、三元模型的信息熵要低于单个字。

3. 在单个小说文本的对比数据中，第一个文本是小说的题目汇总，因此数据与其余区别较大。对于不同的文本，同样的模式下信息熵的计算结果差距不大，且呈现出相同的规律。单个文本与整个语料库的信息熵结果对比时，单个文本的信息熵小于整个语料库，但结果的规律一致。

3 参考文献

[1] 机器学习入门：重要的概念---信息熵（Shannon’s Entropy Model） - 知乎 (zhihu.com)

[2] https://www.zhihu.com/question/35383385/answer/2284821767