云南大学软件学院
实 验 报 告

姓名：王子陈 学号：20171050008 专业：电子科学与技术 日期：2019/11/20 成绩：_____
任课教师：谢仲文

数据挖掘技术 实验五

一、实验目的

1. 探索开放性实验的方法。
2. 从熵的角度分析英文文章比中文文章通常要长的原因。

二、实验内容

1. 在一个简单的、虚拟的数据集（训练集）上分析：英文文章比中文文章通常要长的原因。
   该数据集如下：
   中文：

   江南春

   千里莺啼绿映红，水村山郭酒旗风。南朝四百八十寺，多少楼台烟雨中。
   英文：

   Spring on the Southern Rivershore

   Orioles sing for miles amid red blooms and green trees,
   By hills and rills wine shop streamers wave in the breeze.
   Four hundred eighty splendid temples still remain,
   Of Southern Dynasties in the mist and rain.

   释义：rivershore，河岸边；oriole 指一种鸟，金莺；amid，在……之间；blooms，花；rill，小溪；wine，酒；streamer，飘带；breeze，微风；splendid，锦绣、辉煌；dynasty，朝代；mist，薄雾。
2. 解释你的解决方案及其思路。
3. 尝试建立你自己的分析模型。

三、实验要求

1. 完成实验内容，源码作为实验报告附件一起打为一个压缩包提供。该压缩包要包含实验报告、代码文件。
2. 关键部分要求有注释，注释量不低于 20%
3. 要求独立完成，不得抄袭代码。

四、关键实验步骤（请粘贴关键步骤、代码、实验结果）

1. 中文以汉字为单位，英文以单词为单位，计算一个汉字和一个单词的信息熵

假设古诗和英译文的信息量相等，古诗用了31个汉字,英译文用了41个单词。以每个汉字或单词在材料中出现的频率作为它们可能在某个位置上出现的概率P_n，那么在这个位置上可能出现$\frac{1}{P}$种情况，需要$\sum \log_2 \frac{1}{P_n}$个二进制位表示这个符号，然后可以得到平均每个符号所占用的二进制位：$\frac{1}{n}\sum \log_2 \frac{1}{P_n} = \sum P_n \log_2 \frac{1}{P_n}$，也是每个符号的信息熵。

计算古诗中**每个汉字的信息熵：ent_chars=4.88968018**（统计时去掉了标点符号）

```
#构建字典统计汉字个数
chinese_characters={}
count_char=len(chinese_txt)
for character in chinese_txt:    #chinese_txt 是一个字符串
    if character not in chinese_characters.keys():
        chinese_characters[character]=0
    chinese_characters[character]+=1
#计算每个汉字的熵
info_chars={} #每个汉字的自信息量
ent_chars=0
count_chars=len(chinese_txt)
for key in chinese_characters:
    if key not in info_chars.keys():
        pc=chinese_characters[key]/count_chars
        info_chars[key]=-log(pc,2)
        ent_chars+=pc*info_chars[key]
```

计算英译文中**每个单词的信息熵：ent_words=5.02804529**

```
#构建字典统计字母个数
nglish_words={}
space_idx=0   #英文以空格分隔单词
count_words=0
for word_idx in range(len(english_txt)): #english_txt 是一个字符串,包含所有句子
    if english_txt[word_idx]==" "or word_idx==(len(english_txt)-1):#遇到空格或最后个
        count_words+=1
        if english_txt[space_idx:word_idx].strip().lower() not in english_words.keys(): #去
            掉空格并转成小写
            english_words[english_txt[space_idx:word_idx].strip().lower()]=0
        english_words[english_txt[space_idx:word_idx].strip().lower()]+=1
        space_idx=word_idx
#计算每个单词的熵
info_words={} #每个单词的自信息量
```

```
ent_words=0
for key in english_words:
    if key not in info_words.keys():
        pw=english_words[key]/count_words
        info_words[key]=-log(pw,2)
        ent_words+=pw*info_words[key]
```

可以发现一个英语单词的信息熵要大于一个汉字，熵代表的是不确定性的度量，不确定性越大，所包含的信息量越多，表示该字符在一段文字中就越重要（如果删掉就会对意思的理解产生很大影响）。从直观上看，古诗讲究言辞精炼，自然每个汉字的信息量很大，然而英语中有很多虚词，介词等并没有包含很多信息，但是在这段英译文中大部分单词只出现了一次，概率很小导致信息熵变大了，所以如果在更长的数据集上作比较的话，大量的虚词介词应该会降低信息熵。

这一点可以在以字母作为基本单位计算信息熵时体现出来，另外要比较长度的话，英文是按照字母在计算机中编码存储的，应该考虑每个字母所含的信息熵：

ent_letters=3.9819637

```
#统计英文字母
count_letters=0
letters_dict={}
for letter in english_txt.replace(" ","").replace(",","").replace(".","").lower():
    count_letters+=1
    if letter not in letters_dict.keys():
        letters_dict[letter]=0
    letters_dict[letter]+=1
#计算每个字母的熵
info_letters={} #存每个字母的自信息量
ent_letters=0
for key in letters_dict:
    if key not in info_letters.keys():
        pl=letters_dict[key]/count_letters
        info_letters[key]=-log(pl,2)
        ent_letters+=pl*info_letters[key]
```

可以看到一个字母的信息熵要小于一个汉字，所以在表示相同的信息量时需要更多的字母，但是因为中英文编码方式不同，编码效率不同，在计算机中存储的长度不能一起比较。

为了避免出现符号较多，但不能从中获得更多信息的情况，比如所给的英译文，实验中也做了：把英文再翻译成现代汉语：

"江南春 辽阔的江南到处莺歌燕舞绿树红花相映，小山小河旁酒铺的彩带在微风中摇曳，四百八十座辉煌的南朝庙宇仍在雾霭雨中留存"（56 字）

计算得一个汉字信息熵为 3.565，还是小于英文单词的信息熵。原因也可能是字母的组合使得一个单词的信息量增加了，熵变大了。