成 绩

北京航空航天大學
BEIHANG UNIVERSITY

自然语言处理第一次作业

中文信息熵计算

院（系）名称	自动化科学与电气工程学院
专业名称	电子信息
学生学号	ZY2103207
学生姓名	杨斌发
指导教师	秦曾昌

2022 年 4 月

中文信息熵计算

一、实验目的

参考 Brown, Peter L 等人的论文《An Estimate of an Upper Bound for the Entropy of English.》，使用金庸的 16 本小说作为中文语料来源，分别以词和字为单位，计算中文的平均信息熵。

二、实验原理

2.1 信息熵

信息熵，是 1948 年 C.E.Shannon（香农）从热力学中借用过来提出的概念，解决了对信息的量化度量问题。C. E. Shannon 在 1948 年发表的论文“通信的数学理论（ A Mathematical Theory of Communication ）”中指出，任何信息都存在冗余，冗余大小与信息中每个符号（数字、字母或单词）的出现概率或者说不确定性有关。Shannon 借鉴了热力学的概念，把信息中排除了冗余后的平均信息量称为“信息熵”，并给出了计算信息熵的数学表达式。

通常，一个信源发送出什么符号是不确定的，衡量它可以根据其出现的概率来度量。概率大，出现机会多，不确定性小；反之不确定性就大。在信源中，考虑的不是某一单个符号发生的不确定性，而是要考虑这个信源所有可能发生情况的平均不确定性。若信源符号有n种取值：$U_1 \dots U_i \dots U_n$，对应概率为：$P_1 \dots P_i \dots P_n$，且各种符号的出现彼此独立。这时，信源的平均不确定性应当为单个符号不确定性$-logP_i$的统计平均值（E），可称为信息熵，即

$$H(U) = E[-logp_i] = -\sum_{i=1}^{n} \log_2 p_i$$

式中对数一般取 2 为底，单位为比特。

不同的语言平均每个字符所含有的信息量也是不同的，中文可以说是世界上最简洁的语言，如果将一本英文书翻译成中文，如果字体大致相同，中译本会比原书要薄很多。从中文和英文字符的平均熵的计算结果也可知一二，利用单词一级的语言模型，对大规模语料进行统计的结果为 1.75 比特/字符，对于中文，从字频出发得到的粗略结果为 9.6 比特/汉字。

2.2 统计语言模型

假定S表示某一个有意义的句子，由一连串特定顺序排列的词$w_1, w_2, \ldots w_n$组成，n为句子的长度。现在想知道S在文本中出现的可能性，即$P(S)$。此时需要有个模型来估算，不妨把$P(S)$展开表示为$P(S) = P(w_1, w_2, \ldots, w_n)$。利用条件概率的公式，$S$这个序列出现的概率等于每一个词出现的条件概率相乘，于是$P(w_1, w_2, \ldots, w_n)$可展开为：

$$P(w_1, w_2, \ldots w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\ldots P(w_n|w_1, w_2, \ldots, w_{n-1})$$

其中$P(w_1)$表示第一个词w_1出现的概率；$P(w_2|w_1)$是在已知第一个词的前提下，第二个词出现的概率；以此类推。

显然，当句子长度过长时，$P(w_n|w_1, w_2, \ldots, w_{n-1})$的可能性太多，无法估算，俄国数学家马尔可夫假设任意一个词w_i出现的概率只同它前面的词w_{i-1}有关，这种假设成为马尔可夫假设，S的概率变为

$$P(w_1, w_2, \ldots w_n) = P(w_1)P(w_2|w_1)P(w_3|w_2)\ldots P(w_n|w_{n-1})$$

其对应的统计语言模型就是二元模型。也可以假设一个词由前面$N-1$个词决定，即N元模型。当$N = 1$时，每个词出现的概率与其他词无关，为一元模型，对应S的概率变为

$$P(S) = P(w_1)P(w_2)P(w_3)\ldots P(w_i)\ldots P(w_n)$$

当 N=3 时，每个词出现的概率与其前两个词相关，为三元模型，对应S的概率变为

$$P(w_1, w_2, \ldots w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\ldots P(w_n|w_{n-2}, w_{n-1})$$

根据使用语言模型的不同，信息熵的计算方法也会有所不同，将在之后的实验过程中详细阐述。

三、实验过程

本次试验中使用的语料数据为网上下载的 16 本金庸小说 txt 文本文档，实验中的程序代码使用 Java 语言编写，具体代码见 https://github.com/Xiaoming4249/entropy-compute。

3.1 语料预处理

（1）干扰信息过滤。数据源所提供的 txt 文本文件中存在许多不必要的空格、缩进、换行以及英文字符等，这些都与之后的处理和计算无关，因此需要进行过滤处理。此步处理中采用 Java 中的 *InputStreamReader* 类依次读入文件中的字符，并进行判断和删除。对于文本中的中文标点符号，参考论文中作者对英文符号进行保留的做法，本人也对中文的标点符号进行了保留，将每个标点符

号看作汉字处理，因为其与汉语文本所表达的信息是有关系的

（2）拆分汉字和词语。拆成汉字即读取每一个字符，存储为字符列表；拆成词语则使用 jieba 工具来完成，存储为字符串列表。

3.2 信息熵计算

（1）一元模型的信息熵计算公式为

$$H(x) = -\sum_{x \in X} P(x) \log_2 P(x)$$

其中$P(x)$可近似为每个字或词在语料库中出现的频率。

（2）二元模型的信息熵计算公式为

$$H(X|Y) = -\sum_{x \in X, y \in Y} P(x,y) \log_2 P(x|y)$$

其中联合概率$P(x,y)$可近似为每个二元字组或词组在语料库中出现的频率，条件概率$P(x|y)$ 可近似等于每个二元词组在语料库中出现的频数与以该二元词组的第一个词为词首的二元词组的频数的比值。

（3）三元模型的信息熵计算公式为

$$H(X|Y,Z) = -\sum_{x \in X, y \in Y, z \in Z} P(x,y,z) \log_2 P(x|y,z)$$

其中联合概率$P(x,y,z)$可近似等于每个三元词组在语料库中出现的频率，条件概率$P(x|y,z)$可近似等于每个三元词组在语料库中出现的频数与以该三元词组的前两个词为词首的三元词组的频数的比值。

四、实验结果

划分方式		汉字	词语
总个数		8565392	5348536
平均词长		1.00	1.60
不同字/词个数		5885	161035
信息熵（bits/字（词））	一元模型	9.1014592383	11.0518595348
	二元模型	6.1878995009	6.5732454591
	三元模型	3.8918060582	3.0603282904

计算结果如上表所示。

参考

[1] 百度百科：信息熵 https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF%E7%86%B5/7302318?fr=aladdin

[2] 博客《中文信息熵的计算》https://blog.csdn.net/qq_37098526/article/details/88633403

[3] Brown P F， Pietra S D A， Pietra V D J , et al. An Estimate of an Upper Bound for the[J]. Computational Lingus, 1992, 18(1):31-40.

[4] Jieba Java 库 https://github.com/bluemapleman/jieba-analysis.