成 绩 _____

北京航空航天大學
BEIHANG UNIVERSITY

深度学习与自然语言处理大作业

计算中文信息熵

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	ZY2103812
学生姓名	朱远哲
指导教师	秦曾昌

2022 年 4 月

目　　录

1 问题描述

阅读文章《An Estimate of an Upper Bound for the Entropy of English》，参考文章英文信息熵算法来计算中文（分别以词和字为单位）的平均信息熵。

2 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。香农在发表的论文“通信的数学理论”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，并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：$U_1 \ldots U_i \ldots U_n$，对应概率为：$P_1 \ldots P_i \ldots P_n$，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

$$H(U) = E[-\log P_i] = -\sum_{i=1}^{n} P_i \log P_i$$

式中对数一般取2为底，单位为比特。

3 中文信息熵的计算

3.1 N 元模型

一元模型的信息熵计算公式为

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

其中 $P(x)$ 可近似等于每个词在语料库中出现的频率。

二元模型的信息熵计算公式为

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

其中联合概率 $P(x, y)$ 可近似等于每个二元词组在语料库中出现的频率，条件概率 $P(x \mid y)$ 可近似等于在该二元词组的第一个字（词出现的情况下，该二元词组在语料库出现的频率。

三元模型的信息熵计算公式为

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

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 $P(x \mid y, z)$ 可近似等于在该三元词组的前两个字出现的情况下，该三元词组在语料库出现的频率。

3.2 中文分词

以词为单位进行信息熵计算需要将句子拆开分成一个一个的词语，需要用到中文分词工具，在本次大作业中采用python中的jieba分词系统。

jieba分词支持三种分词模式：

1. 精确模式，试图将句子最精确地切开，适合文本分析；
2. 全模式，把句子中所有的可以成词的词语都扫描出来, 速度非常快，但是不能解决歧义；
3. 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。

在本次作业中采用精确模式进行分词。

4 实验过程

实验采用数据集为提供的16本小说文本，分别以字和词为单位进行信息熵的计算。由于文本中含有较多非中文字符，需要先对文本进行预处理去除特殊字符，但需保留逗号、句号、感叹号、问号作为句子的分割，以字或词为单位形成语料库。以字为单位的流程中，按一个字、两个字、三个字分别形成词频库；以词为单位的流程中，需要先对文本利用jieba系统进行分词，按一个词、两个词、三个词分别形成词频库。最后分别利用一元模型、二元模型、三元模型处理词频库进行计算得到中文平均信息熵。

5 运行结果

5.1 按字分类

小说名称	字数	信息熵（一元模型）	信息熵（二元模型）	信息熵（三元模型）
鹿鼎记	1024168	9.2946	5.8818	2.8857
天龙八部	1021364	9.4063	6.0282	2.8872
神雕侠侣	827762	9.3511	5.9513	2.777
笑傲江湖	824759	9.2081	5.8017	2.8107
倚天屠龙记	818424	9.3952	5.9339	2.7528
射雕英雄传	772682	9.4512	5.9647	2.7087
书剑恩仇录	435834	9.4719	5.6714	2.3402
碧血剑	416441	9.458	5.7691	2.3021
飞狐外传	375487	9.3128	5.6681	2.361
侠客行	309819	9.1537	5.51	2.3124
连城诀	194506	9.1729	5.3112	2.0964
雪山飞狐	117209	9.172	5.0524	1.7108
白马啸西风	59333	8.8732	4.4608	1.5613
三十三剑客图	53388	9.6725	4.7446	0.927
鸳鸯刀	30484	8.9814	4.1326	1.0945
越女剑	13695	8.8313	3.56	0.8601

5.2 按词分类

小说名称	词数	平均词长	信息熵（一元模型）	信息熵（二元模型）	信息熵（三元模型）
鹿鼎记	605026	1.693	11.4449	5.6777	1.5334
天龙八部	603958	1.691	11.732	5.6069	1.4084
神雕侠侣	495045	1.672	11.5815	5.4698	1.3767
笑傲江湖	482447	1.710	11.3974	5.5478	1.4586
倚天屠龙记	475216	1.722	11.7563	5.465	1.2629
射雕英雄传	456896	1.691	11.8033	5.4032	1.2166
书剑恩仇录	253477	1.719	11.7174	4.9567	0.9522
碧血剑	242858	1.715	11.7454	4.9421	0.9053
飞狐外传	221298	1.697	11.5247	4.9401	0.988
侠客行	183238	1.691	11.1874	4.8998	1.0507
连城诀	117327	1.658	11.0363	4.6251	0.8626
雪山飞狐	71087	1.649	10.9135	4.1156	0.7632
白马啸西风	37059	1.601	10.0695	3.937	0.7806
三十三剑客图	31359	1.702	11.6737	2.8819	0.1627
鸳鸯刀	18361	1.660	10.2461	3.1302	0.4991
越女剑	8069	1.697	10.0791	2.4626	0.2331

5.3 结果讨论与分析

从实验结果中可以看出，对于以字和词为单位的分析中，一元模型、二元模型、三元模型计算得到的平均信息熵呈递减趋势。整体上，字数越多，信息熵越大。

6 个人总结与体会

本次大作业实现了利用一元模型、二元模型、三元模型对于中文平均信息熵的计算，也是第一次接触自然语言处理方面的内容，学习到了利用信息熵表征文本的信息量，同时也使用了jieba分析这个强大的分词工具，可以将一个句子按照词性等分成词语，并利用条件概率等概率论知识进行信息熵的计算。由于python基础较为薄弱，在本次实验中也参考了一些现有资料并进行学习，对多种代码进行认真阅读学习后，进行了整合加入了自己的想法和调整，部分自己重新写的部分在调试中出现了许多问题，也进行了很多调整，但有些地方还不是不够简洁，也体会到了很多简洁代码的精妙之处，希望在之后的学习过程中，尽量保证结果准确性的情况下，学习更多向量化等精简代码的方法和技巧，积累更多编程经验。