深度学习与自然语言处理
第一次大作业

中文平均信息熵的计算

院（系）名称	自动化科学与电气工程学院
学 生 学 号	ZY2103803
学 生 姓 名	李鑫磊

2022 年 **04** 月

一、问题描述

根据下载的中文书籍，计算其中文信息熵。

二、具体算法实现

2.1 中文语料预处理

由于一元模型不需要考虑上下文关系，所以其读取语料的方式与二元模型和三元模型不一样，直接将文件夹中的 txt 文件合并写入一个文件中，再通过 jieba 进行分词，得到所需要的 txt 格式语料库。

二元模型和三元模型需要考虑上下文关系，不能直接去掉所有标点符号得到无分隔的语料。通过中文停词表清理语料，生成由停词分行的 txt 格式语料库。

2.2 词频统计

一元模型只需要统计每个词在语料库中出现的频数，得到词频表。二元模型也需要统计每次词在语料库中出现的频数，得到词频表，作为计算条件概率 $P(w_i \mid w_{i-1})$ 时的分母，并且需要统计每个二元词组在语料库中出现的频数，得到二元模型词频表。三元模型需要统计每个二元词组在语料库中出现的频数，得到二元模型词频表，作为计算条件概率 $P(w_i \mid w_{i-2}, w_{i-1})$ 时的分母，并且需要统计每个三元词组在语料库中出现的频数，得到三元模型词频表。

2.3 计算信息熵

一元模型的信息熵计算公式为

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

其中 $P(x)$ 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 $P(x, y)$ 可近似等于每个词在语料库中出现的概率，条件概率 $P(x \mid y)$ 可近似等于每个二元词组在语料库中出现的频数与该二元词组的第一个词为词首的二元词组的频数的比值。

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。

三、运行结果

将 16 本书导入到 python 中，运行的结果如表 1 所示：

表 1 各书平均信息熵

书名	1-gram	2-gram	3-gram
三十三剑客	11.654	2.9647	0.2879
书剑恩仇录	11.7007	5.0327	1.0507
侠客行	11.1696	4.9674	1.1364
倚天屠龙记	11.7391	5.5257	1.3419
天龙八部	11.7099	5.6954	1.4917
射雕英雄传	11.8224	5.485	1.272
白马啸西风	10.2789	3.9877	0.7365
碧血剑	11.7309	5.0051	1.0017
神雕侠侣	11.6801	5.4905	1.339
笑傲江湖	11.3827	5.6221	1.5396
越女剑	10.1031	2.5278	0.3243
连城诀	11.0263	4.7036	0.962
雪山飞狐	11.0988	4.1173	0.6991
飞狐外传	11.513	5.0008	1.0627

鸳鸯刀	10.4994	3.0937	0.4279
鹿鼎记	11.4413	5.7694	1.6213

从表中我们可以看出，各书 N-gram 模型的信息熵差距不大，而随着 N 取值变大，文本的信息熵则越小，这是因为 N 取值越大，通过分词后得到的文本中词组的分布就越简单，N 越大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的不确定性，也即减少了文本的信息熵，符合实际认知。

四、个人总结和体会

通过本次大作业的算法设计，对于自然语言处理有了初步的认知，同时对于信息熵的概念及意义有了更深入的了解，掌握了一个文本的信息熵计算方法。