单位代码_____
学　　号 ZY2303803
分 类 号_____

北京航空航天大學
BEIHANG UNIVERSITY

深度学习与自然语言处理（NLP）第一次课后作业

院（系）名称　自动化科学与电气工程学院

专 业 名 称　自动化

学 生 姓 名　董晨辉

2024 年 04 月

深度学习与自然语言处理（NLP）第一次课后作业

董晨辉
1127666815@qq.com

Abstract

通过中文语料库金庸小说大全的.txt 文件验证了 Zipf's Law(奇夫定律)，并在此基础上分别以词和字为单位计算了十余部小说全部文本的平均信息熵。

Introduction

(一) 奇夫定律

齐夫定律（Zipf's law）由哈佛大学的语言学家乔治·金斯利·齐夫于 1949 年提出，是一种描述自然语言中单词出现频率的统计规律。

根据齐夫定律，在给定的语料库中，出现频率最高的单词出现的频率大约是出现频率第二位的单词的两倍，而出现频率第二位的单词则是出现频率第四位的单词的两倍。这种关系可以扩展到整个词汇表，其中每个单词的出现频率与其在频率表中的排名成反比。

(二) 熵与信息熵

熵，泛指某些物质系统状态的一种量度，某些物质系统状态可能出现的程度。亦被社会科学用以借喻人类社会某些状态的程度。熵的概念是由德国物理学家克劳修斯于 1865 年所提出。最初是用来描述“能量退化”的物质状态参数之一，在热力学中有广泛的应用。它在控制论、概率论、数论、天体物理、生命科学等领域都有重要应用，在不同的学科中也有引申出的更为具体的定义，按照数理思维从本质上说，这些具体的引申定义都是相互统一的，熵在这些领域都是十分重要的参量。

信息熵这一概念由克劳德·香农于 1948 年提出。香农是美国著名的数学家、信息论创始人，他提出的“信息熵”的概念，为信息论和数字通信奠定了基础。信息熵是用于衡量不确定性的指标，也就是离散随机事件出现的概率，简单地说“情况越混乱，信息熵就越大，反之则越小”。 数学家香农给出了信息熵的计算公式，如下所示：

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

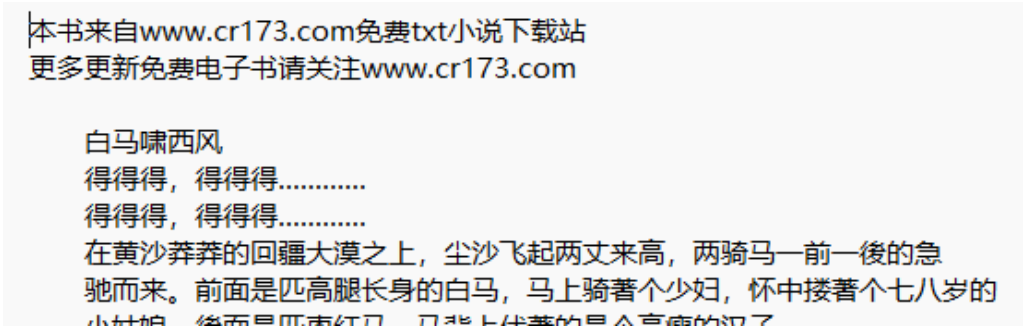
信息熵的三个性质为：单调性，即发生概率越高的事件，其携带的信息量越低。非负性，信息熵可以看作为一种广度量，非负性是一种合理的必然。累加性，即多随机事件同时发生存在的总不确定性的量度是可以表示为各事件不确定性的量度的和，这也是广度量的一种体现。香农从数学上严格证明了满足上述三个条件的随机变量不确定性度量函数具有唯一形式。

Methodology

在实验过程中，首先需要对.txt 文本进行预处理，并对文本进行分词。在分词的基础上进行词频统计以及信息熵的计算。

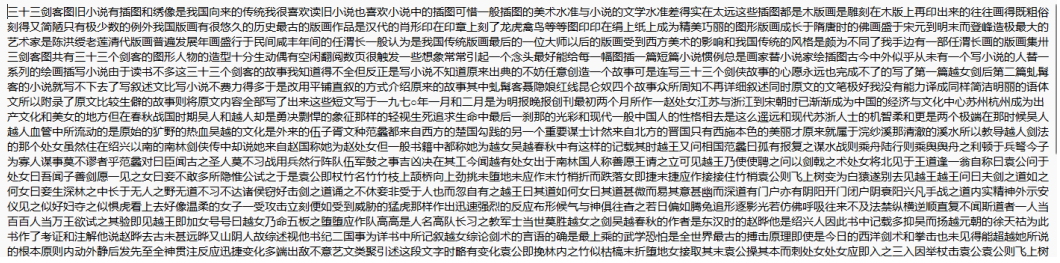
M1：文本预处理

由于下载的文件里存在各种标点符号以及网页信息，去除了文件里面无关信息：




```
for filename in filelist:
    filepath = datapath + '/' + filename
    with open(filepath, "r", encoding="gb18030") as file:
        filecontext = file.read()
        filecontext = filecontext.replace(
            _old: "本书来自www.cr173.com免费txt小说下载站\n更多更新免费电子书请关注www.cr173.com", _new: '')
        filecontext = filecontext.replace( _old: "本书来自www.cr173.com免费txt小说下载站", _new: '')
        corpus.append(filecontext)
        file.close()
```

文本处理之后：


三十三剑客图旧小说有插图和绣像是我国向来的传统我很喜欢读旧小说也喜欢小说中的插图可惜一般插图的美术水准与小说的文学水准差得实在太远这些插图都是木版画是雕刻在木版上再印出来的往往画得既粗俗刻得又简陋只有极少数的例外我国版画有很悠久的历史最古的版画作品是汉代的肖形印在印章上刻了龙虎禽鸟等等图印印在绢上纸上成为精美巧丽的图形版画成长于隋唐时的佛画盛于宋元到明末而登峰造极最大的艺术家是陈洪绶老莲清代版画普遍发展年画盛行于民间咸丰年间的任渭长一般认为是我国传统版画最后的一位大师以后的版画受到西方美术的影响和我国传统的风格是颇为不同了我手边有一部任渭长画的版画集卅三剑客图共有三十三个剑客的图形人物的造型十分生动偶有空闲翻阅数页很触发一些想象常常引起一个念头最好能给每一幅图插一篇短篇小说惯例总是画家替小说家绘插图古今中外似乎从未有一个写小说的人替一系列的绘画插写小说由于读书不多这三十三个剑客的故事我知道得不全但反正是写小说不知道原来出典的不妨任意创造一个故事可是连写三十三个剑侠故事的心愿永远也完成不了的写了第一篇越女剑后第二篇虬髯客的小说就写不下去了写叙述文比写小说不费力得多于是改用平铺直叙的方式介绍原来的故事其中虬髯客聂隐娘红线昆仑奴四个故事众所周知不再详细叙述同时原文的文笔极好我没有能力译成同样简洁明丽的语体文所以附录了原文比较生僻的故事则将原文内容全部写了出来这些短文写于一九七○年一月和二月是为明报晚报创刊最初两个月所作一赵处女江苏与浙江到宋朝时已渐渐成为中国的经济与文化中心苏州杭州成为出产文化和美女的地方但在春秋战国时期吴人和越人却是勇决剽悍的象征那样的轻视生死追求生命中最后一刹那的光彩和现代一般中国人的性格相去是这么遥远和现代苏浙人士的机智柔和更是两个极端在那时候吴人越人血管中所流动的是原始的犷野的热血吴越的文化是外来的伍子胥文种范蠡都来自西方的楚国勾践的另一个重要谋士计然来自北方的晋国只有西施本色的美丽才原来就属于浣纱溪那清澈的溪水所以教导越人剑法的那个处女虽然住在绍兴以南的南林剑侠传中却说她来自赵国称她为赵处女但一般书籍中都称她为越女吴越春秋中有这样的记载其时越王又问相国范蠡曰孤有报复之谋水战则乘舟陆行则乘輿輿舟之利顿于兵弩令子为寡人谋事莫不谬者乎范蠡对曰臣闻古之圣人莫不习战用兵然行阵队伍军鼓之事吉凶决在其工今闻越有处女出于南林国人称善應王请之立可见越王乃使使聘之问以剑戟之术处女将北见于王道逢一翁自称曰袁公问于处女曰吾闻子善剑愿一见之女曰妾不敢多所隐惟公试之于是袁公即杖竹名竹竹枝上颉桥向上劲挑未墮地未应作未竹梢折而跌落女即捷未捷应作接接住竹梢袁公则飞上树变为白猿遂别去见越王问曰夫剑之道如之何女曰妾生深林之中长于无人之野无道不习不达诸侯窃好击剑之道谨之不休妾非受于人也而忽自有之越王曰其道如何女曰其道甚微而易其意甚幽而深道有门户亦有阴阳开门闭户阴衰阳兴凡手战之道内实精神外示安仪见之似好妇守之似惧虎看上去好像温柔的女子一受攻击立刻便如受到威胁的猛虎那样作出迅速强烈的反应布形候气与神俱往吾之吾日偏如腾兔追形逐影光若仿佛呼吸往来不及法禁纵横逆顺直复不闻斯道者一人当百百人当万王欲试之其验即见越王即加女号号曰越女乃命五板之墮墮应作队高高是人名高队长习之教军士当世莫胜越女之剑吴越春秋的作者是东汉时的赵晔他是绍兴人因此书中记载多抑吴而扬越元朝的徐天祜为此书作了考证和注解他说赵晔去古未甚远晔又山阴人故综述视他书纪二国事为详书中所记叙越女综论剑术的言语的确是最上乘的武学恐怕是全世界最古的搏击原理即使是今日的西洋剑术和拳击也未见得能超越她所说的根本原则内动外静后发先至全神贯注反应迅捷变化多端出敌不意艺文类聚引述这段文字时略有变化袁公即挽林内之竹似枯槁未折墮地女接取其未袁公操其本而刺处女处女应即入之三入因举杖击袁公袁公则飞上树

M2：中文分词

中文分词，通俗来说，就是将一句(段)话按一定的规则(算法)拆分成词语、成语、单个文字。中文分词是很多应用技术的前置技术，如搜索引擎、机器翻译、词性标注、相似度分析等，都是先对文本信息分词处理，再用分词结果来搜索、翻译、对比等。在 Python 中，最好用的中文分词库是 jieba。用“结巴”给一个中文分词库命名，非常生动形象，同时还带有一种程序员式的幽默感。

jieba 分词支持四种分词模式，精确模式是最常用的分词模式，分词结果不存在冗余数据：

jieba.cut（text, cut_all=False）

如图所示使用 jieba.cut()函数逐行进行分词并统计分词个数：

```
split_words = []
words_num = 0
for line in corpus:
    for x in jieba.cut(line):
        split_words.append(x)
        words_num += 1
```

M3：去除停用词

在进行分词后应根据停用词表去除无关的停用词：

```
# 去除停用词
stopword_file = open('cn_stopwords.txt', "r", encoding='utf-8')
# stopword_file = open('cn_punctuation.txt', "r", encoding='utf-8')
stopwordlist = stopword_file.read().split('\n')  # 分解字符串
stopword_file.close()
for stopword in stopwordlist:
    del word_fc[stopword]
```

M4：词频统计

在 Python 中，NLTK（Natural Language Toolkit）库是一个功能强大、广泛使用的自然语言处理库。词频统计使用 NLTK 库中的 FreqDisk 函数，能够统计数组 zho 给单词出现的频率。

如图所示，利用 FreqDisk 函数将 split_works[ ]列表中的分词进行词频统计，并按照词频数从大到小排列：

```
word_fc = FreqDist(split_words)
word_fc_sort = sorted(word_fc.items(), key=lambda x: x[1], reverse=True)
```

并按照公式计算信息熵：

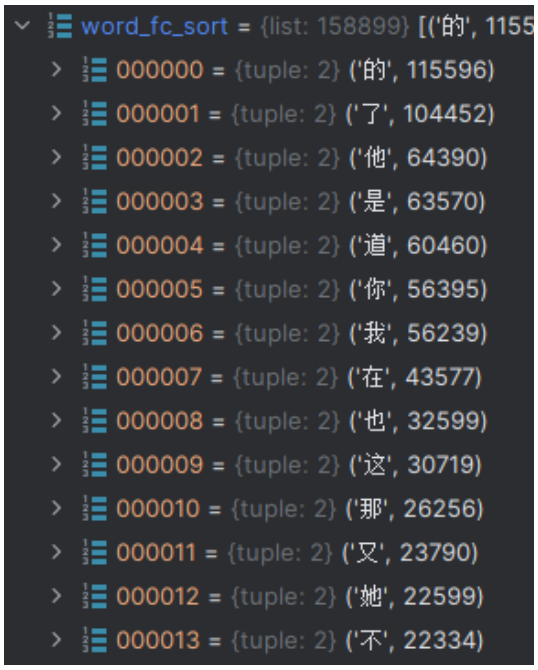
```
entropy = []
entropy = [-(uni_word[1] / words_num) * math.log( x: uni_word[1] / words_num,  base: 2) for uni_word in word_fc_sort]
print("基于jieba分割的中文平均信息熵为:", round(sum(entropy), 5), "比特/词")
```

Experimental Studies

采用 cn_punctuation 去除停词的实验结果如下所示：

运行脚本后词频统计 word_fc_sort[ ]如下图所示：

Jieba 分词



按字分词

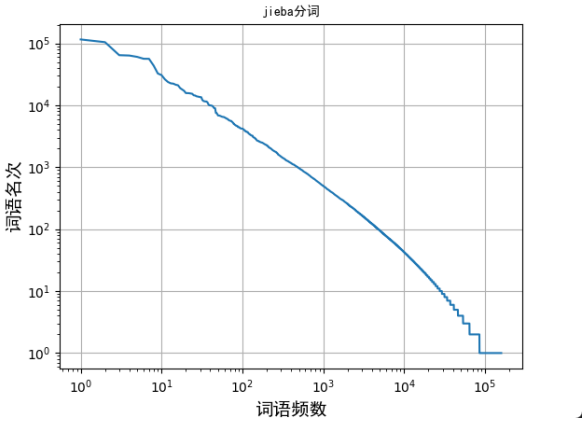
表 1 分词结果

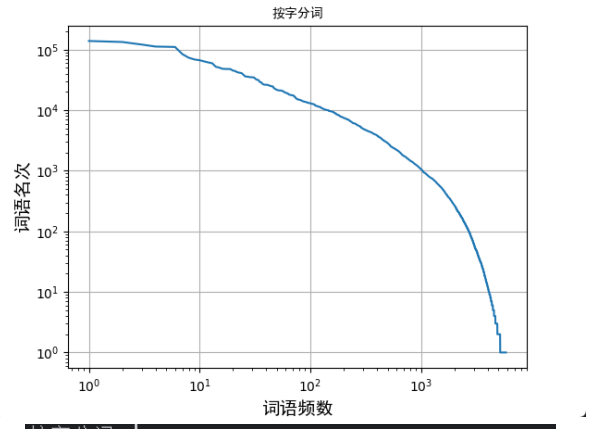
	按照 jieba 分词	按字分词
语料库字数	7273194	7269812
分词字数	4289223	7269812
平均字长	1.69569	1

表 2 信息熵

	按照 jieba 分词	按字分词
中文平均信息熵	12.15446 比特	9.53684 比特

绘制 log-log 图像:






```
jieba分词：
语料库字数：7273194
分词个数：4289223
平均词长：1.69569
基于jieba分割的中文平均信息熵为：12.15446 比特/词
```

```
按字分词：
语料库字数：7269812
分词个数：7269812
平均词长：1.0
基于按字分割的中文平均信息熵为：9.53684 比特/词
```

图 1 只去除标点后的实验结果

采用 cn_stopwords 去除停词的实验结果如下所示：

运行脚本后词频统计 word_fc_sort[ ]如下图所示：

Jieba 分词

按字分词

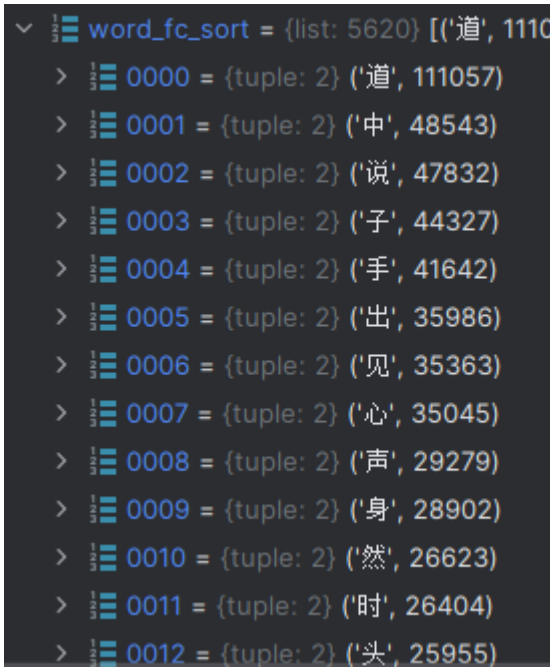


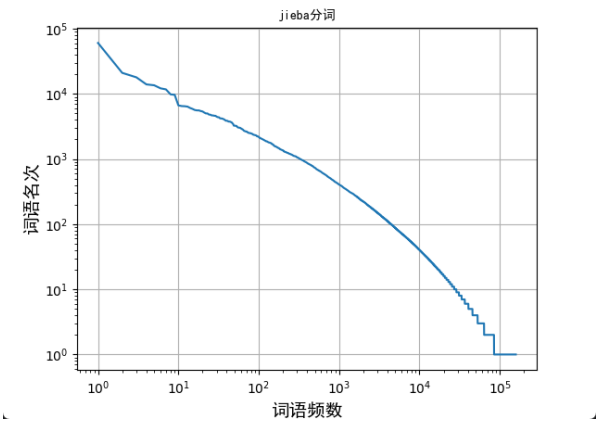
表 3 去除停用词分词结果

	按照 jieba 分词	按字分词
语料库字数	5660926	4495115
分词字数	2872492	4495115
平均字长	1.85417	1

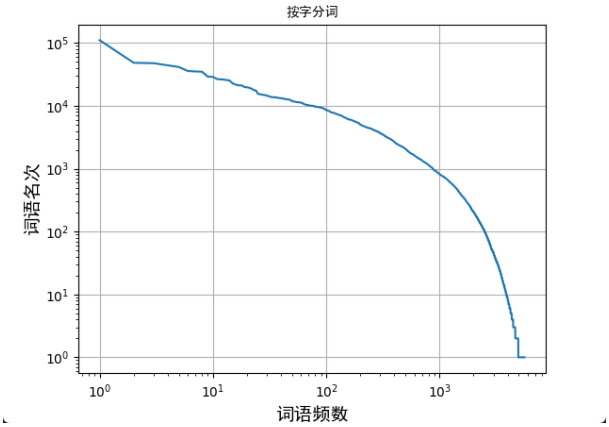
表 4 信息熵

	按照 jieba 分词	按字分词
中文平均信息熵	13.58133 比特	9.95158 比特

绘制 log-log 图像：




```
jieba分词：
语料库字数：5660962
分词个数：2872492
平均词长：1.97075
基于jieba分割的中文平均信息熵为：13.58133 比特/词
```




```
按字分词：
语料库字数：4495115
分词个数：4495115
平均词长：1.0
基于按字分割的中文平均信息熵为：9.95158 比特/词
```

图 2 去除所有停用词后的实验结果

Conclusions

本文通过分词和词频统计，初步验证了奇夫定律，即在给定的语料库中，出现频率最高的单词出现的频率大约是出现频率第二位的单词的两倍，而出现频率第二位的单词则是出现频率第四位的单词的两倍，反映到 log-log 图中则是一条斜率约为-1 的直线。这种关系可以扩展到整个词汇表，其中每个单词的出现频率与其在频率表中的排名成反比。

而通过对比去除停用词或标点可以看出对于一元模型，说明去除信息量较少的停词（例如“一些”、“大概”等）对于信息熵的影响较小。同时对比去除停用词前后可观察到去除停用词更加符合奇夫定律的规则。

通过对比使用 jieba 库进行分词和直接按字分词，表明了按照语意和词义进行分词所得到的平均信息熵要高于只按字分词的平均信息熵，并且 jieba 分词能够更好的验证奇夫定律。

References

[1] Peter F. Brown, Vincent J. Della Pietra, Robert L. Mercer, Stephen A. Della Pietra, and Jennifer C. Lai. 1992. An estimate of an upper bound for the entropy of English. Comput. Linguist. 18, 1 (March 1992), 31–40.

[2] https://blog.csdn.net/weixin_42663984/article/details/115718241

[3] https://blog.csdn.net/GWH_98/article/details/117001985