单位代码
学　　号 ZY2103116
分 类 号

北京航空航天大學
BEIHANG UNIVERSITY

基于金庸小说数据集的中文平均信息熵计算

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　杨卓昆

2022 年 04 月

基于金庸小说数据集的中文平均信息熵计算

深度学习与自然语言处理（NLP）第一次课后作业

杨卓昆 自动化科学与电气工程学院 ZY2103116
yangzhuokun1998@buaa.edu.cn

摘要

如何对信息进行量化度量是计算机对于信息进行高效处理的前提，一直以来研究者都在不断尝试寻找最为合理解决的方法，这引起了许多人的思考。衡量信息量即为衡量信息价值，信息带来的效益。我们可以认为，信息量的度量就等于不确定性的多少。本文参考 Peter Brown 文章来计算中文的平均信息熵。选取数据集为 16 本金庸武侠小说，搭建三种 n-gram 模型，计算得出不同分词下的中文信息熵。本文通过对比三种 n-gram 语言模型（1-gram、2-gram、3-gram）得到的结果，分析得出 n 的取值越大，即在估计时考虑的词数越多，则上下文之间的联系越多，不同词组合出现的种类个数也会越多，则文本的信息熵则越小的结论。

同时，文章在此基础上研究了繁体字对于信息熵的影响。在使用繁体字替换之后，词库总词数和不同词的个数都有所减少。这可能是由于 jieba 分词模块对于简体字的识别率更高，从而减少了个别字单独划分的现象。同时，可以看到除了 2-gram 以外，替换繁体字的信息熵都有所减少。

目录

1 内容介绍

自然语言是指自然地随文化演变发展而成的语言。汉语、英语、日语都是自然语言的例子。自然语言是人类交流和思维的主要工具，而自然语言处理是人工智能中最为困难的问题之一。自然语言是一种上下文相关的信息表达和传递的方式，让计算机处理自然语言，一个基本的问题就是为自然语言这种上下文相关的特性建立数学模型，即统计语言模型。如何对信息进行量化度量是计算机对于信息进行高效处理的前提，一直以来研究者都在不断尝试寻找最为合理解决的方法，这引起了许多人的思考。衡量信息量即为衡量信息价值，信息带来的效益。我们可以考虑以下场景：有两位考生试图在考场上通过非正当的信息交流以传递一道考试题的答案，于是他们约定了手指分别指向上下的手势以传递对或错的信息，假如这是一道判断题，则只需要做一次动作表达对或错，而如果是一道四项选择题，他们则需要做两次手势，分别传达是否为 A 或 B、是否为 A 或 C，通过传递这两个信息以精准定位到正确的答案上。如果判断题和选择题的分值都是两分，则前一种情况下一次手势为两分的信息价值，后一种情况下由于两次手势才得到两分，所以每次手势为一分的信息价值。在这种情况下，指向上下的两种手势在计算机中可以通过比特数表示，一个比特（bit）即为 0 或者 1，可以看到信息量的比特数和所有可能情况的对数函数有关。1948 年，香农提出了“信息熵”的概念，解决了对信息的量化度量问题。一条信息的信息量大小和它的不确定性有直接的关系。比如说，我们要搞清楚一件非常非常不确定的事，或是我们一无所知的事情，就需要了解大量的信息。相反，如果我们对某件事已经有了较多的了解，我们不需要太多的信息就能把它搞清楚。所以，从这个角度，我们可以认为，信息量的度量就等于不确定性的多少。本文通过参考 Peter Brown[1]的文章计算了中文的平均信息熵，并对结果进行了分析。

1.1 实验要求

本次实验需要参考 Peter Brown 文章来计算中文的平均信息熵。该文章通过构造一个三元词模型，计算该模型与一个英文文本样本之间的交叉熵，得到了印刷英文字符熵的估计上限为 1.75 比特。文章计算了包括 596 万个字符的布朗语料库的交叉熵，是由该团队从 5.83 亿个训练文本中构建的单词三元组合语言模型测量的。自从香农 1951 年的论文以来，人们对英语的熵有过很多的估计，但是该文章的方法在这方面与以前的工作不同，不光使用了更大的英语文本样本，还使用语言模型来近似字符串的概率，并且预测了所有可打印的 ASCII 字符。所以，本文希望在对中文的信息熵进行估计时，也能够尽可能做到以上三点。

1.2 数据集介绍

本次数据集为 16 本金庸武侠小说，武侠小说辞藻华艳而又通俗易懂，贴近生活而又包罗万象，文白夹杂而又雅俗共赏，非常适合作为本次实验的数据集。在所有的武侠小说作家中，金庸的文笔首屈一指，语言流畅、凝练、准确、画面感强。本次数据包含“飞雪连天射白鹿，笑书神侠倚碧鸳”十四篇长篇小说以及《越女剑》和《三十三剑客图》，基本上涵盖了金庸的所有武侠作品。

数据库地址： https://share.weiyun.com/5zGPyJX

2 实验原理

2.1 熵和信息熵

2.1.1 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。但那时熵仅仅是一个可以通过热量改变来测定的物理量，其本质仍没有很好的解释，直到统计物理、信息论等一系列科学理论发展，熵的本质才逐渐被解释清楚，即，熵的本质是一个系统“内在的混乱程度”。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。

2.1.2 信息熵

衡量信息量即为衡量信息价值，信息带来的效益。我们可以考虑以下场景：有两位考生试图在考场上通过非正当的信息交流以传递一道考试题的答案，于是他们约定了手指分别指向上下的手势以传递对或错的信息，假如这是一道判断题，则只需要做一次动作表达对或错，而如果是一道四项选择题，他们则需要做两次手势，分别传达是否为 A 或 B、是否为 A 或 C，通过传递这两个信息以精准定位到正确的答案上。如果判断题和选择题的分值都是两分，则前一种情况下一次手势为两分的信息价值，后一种情况下由于两次手势才得到两分，所以每次手势为一分的信息价值。

在这种情况下，指向上下的两种手势在计算机中可以通过比特数表示，一个比特（bit）即为 0 或者 1，可以看到信息量的比特数和所有可能情况的对数函数有关。1948 年，香农提出了“信息熵”的概念，解决了对信息的量化度量问题。一条信息的信息量大小和它的不确定性有直接的关系。比如说，我们要搞清楚一件非常非常不确定的事，或是我们一无所知的事情，就需要了解大量的信息。

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。不确定性函数f是概率P的减函数；两个独立符号所产生的不确定性应等于各自不确定性之和，即：

$$f(P1,P2) = f(P1) + f(P2)$$

这称为可加性。同时满足这两个条件的函数f是对数函数，即:

$$f(P1) = log\frac{1}{p} = -logp$$

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：$U1 \dots Ui \dots Un$，对应概率为：$P1 \dots Pi \dots Pn$，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$-logPi$的统计平均值E，可称为信息熵。即

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

信息论之父克劳德·香农给出的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。

2.2 N-Gram 语言模型

N-Gram 是一种基于统计语言模型的算法。它的基本思想是将文本里面的内容按照字节进行大小为 N 的滑动窗口操作，形成了长度是 N 的字节片段序列。每一个字节片段称为 gram，对所有 gram 的出现频度进行统计，并且按照事先设定好的阈值进行过滤，形成关键 gram 列表，也就是这个文本的向量特征空间，列表中的每一种 gram 就是一个特征向量维度。

2.2.1 语言模型

语言模型可以对一段文本的概率进行估计，对信息检索，机器翻译，语音识别等任务有着重要的作用。语言模型分为统计语言模型和神经网络语言模型。统计语言模型，即要判断一段文字是不是一句自然语言，可以通过确定这段文字的概率分布来表示其存在的可能性。 语言模型中的词是有顺序的，给定m个词看这句话是不是一句合理的自然语言，关键是看这些词的排列顺序是不是正确的。所以统计语言模型的基本思想是计算条件概率。

对于自然语言相关的问题，比如机器翻译，最重要的问题就是文本的序列有时候不是符合我们人类的使用习惯，语言模型就是用于评估文本序列符合人类语言使用习惯程度的模型。当前的语言模型是以统计学为基础的统计语言模型，统计语言模型是基于预先人为收集的大规模语料数据，以真实的人类语言为标准，预测文本序列在语料库中可能出现的概率，并以此概率去判断文本是否“合法”，

是否能被人所理解。

2.2.2 N-Gram 语言模型

N-Gram 是一种基于统计语言模型的算法。它的基本思想是将文本里面的内容按照字节进行大小为 N 的滑动窗口操作，形成了长度是 N 的字节片段序列。

每一个字节片段称为 gram，对所有 gram 的出现频度进行统计，并且按照事先设定好的阈值进行过滤，形成关键 gram 列表，也就是这个文本的向量特征空间，列表中的每一种 gram 就是一个特征向量维度。

该模型基于这样一种假设，第 N 个词的出现只与前面 N-1 个词相关，而与其它任何词都不相关，整句的概率就是各个词出现概率的乘积。这些概率可以通过直接从语料中统计 N 个词同时出现的次数得到。常用的是二元的 Bi-Gram 和三元的 Tri-Gram。

如果我们有一个由 m 个词组成的序列（或者说一个句子），我们希望算得概率$P(W1,W2,\dots Wm)$，根据链式规则，可得：

$$P(w_1,w_2,\dots w_m)=P(w_1)\times P(w_2|w_1)\times P(w_3|w_1,w_2)\dots\times P(w_m|w_1,\dots,w_{m-1})$$

这个概率显然并不好算，不妨利用马尔科夫链的假设，即当前这个词仅仅跟前面几个有限的词相关，因此也就不必追溯到最开始的那个词，这样便可以大幅缩减上述算式的长度。即：

$$P(w_1,w_2,\dots w_m)=P(w_i|w_{i-n+1},\dots,w_{i-1})$$

下面给出一元模型，二元模型，三元模型的定义：

当 n=1, 一个一元模型（unigram model)即为 ：

$$P(w_1,w_2,\cdots,w_m)=\prod_{i=1}^{m}P(w_i)$$

当 n=2, 一个二元模型（bigram model)即为 ：

$$P(w_1,w_2,\cdots,w_m)=\prod_{i=1}^{m}P(w_i|w_{i-1})$$

当 n=3, 一个三元模型（trigram model)即为

$$P(w_1,w_2,\cdots,w_m)=\prod_{i=1}^{m}P(w_i|w_{i-2}w_{i-1})$$

3 实验过程

3.1 数据预处理

由于数据库里存在各种标点符号以及网页信息，所以首先需要对数据进行预处理操作。具体的操作步骤如下所示，首先按照换行符将文件分割为多个段落，由于前两行和最后两行的信息是网页页眉，所以之后删除前两行以及最后两行。之后删除英文标点符号以及中文标点符号，这两个标点分别由 string.punctuation 以及 zhon.hanzi.punctuation 提供。删除的标点如下所示：

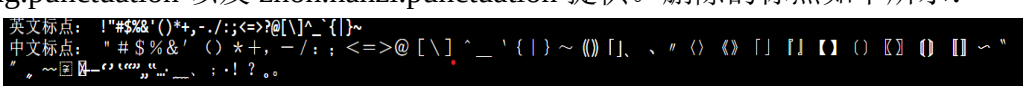

图 1 删除标点

最终得到的结果示例如下：

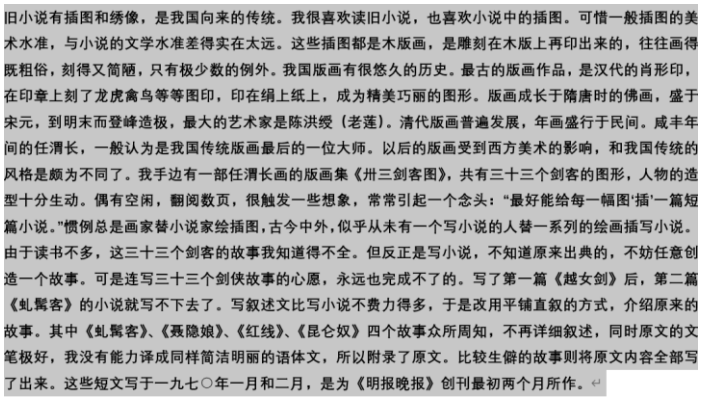
旧小说有插图和绣像，是我国向来的传统。我很喜欢读旧小说，也喜欢小说中的插图。可惜一般插图的美术水准，与小说的文学水准差得实在太远。这些插图都是木版画，是雕刻在木版上再印出来的，往往画得既粗俗，刻得又简陋，只有极少数的例外。我国版画有很悠久的历史。最古的版画作品，是汉代的肖形印，在印章上刻了龙虎禽鸟等等图印，印在绢上纸上，成为精美巧丽的图形。版画成长于隋唐时的佛画，盛于宋元，到明末而登峰造极，最大的艺术家是陈洪绶（老莲）。清代版画普遍发展，年画盛行于民间。咸丰年间的任渭长，一般认为是我国传统版画最后的一位大师。以后的版画受到西方美术的影响，和我国传统的风格是颇为不同了。我手边有一部任渭长画的版画集《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。这些短文写于一九七〇年一月和二月，是为《明报晚报》创刊最初两个月所作。↵

图 2 预处理前

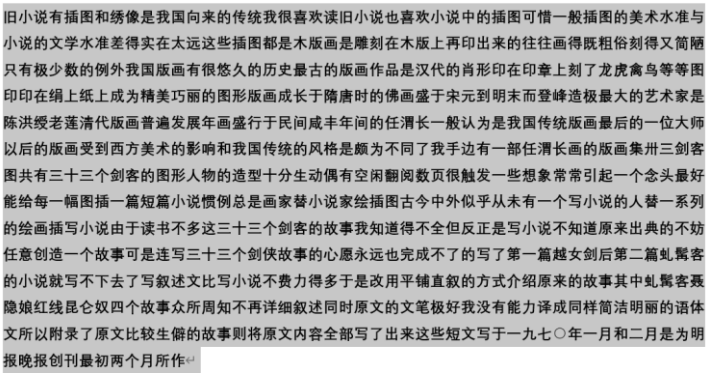
旧小说有插图和绣像是我国向来的传统我很喜欢读旧小说也喜欢小说中的插图可惜一般插图的美术水准与小说的文学水准差得实在太远这些插图都是木版画是雕刻在木版上再印出来的往往画得既粗俗刻得又简陋只有极少数的例外我国版画有很悠久的历史最古的版画作品是汉代的肖形印在印章上刻了龙虎禽鸟等等图印印在绢上纸上成为精美巧丽的图形版画成长于隋唐时的佛画盛于宋元到明末而登峰造极最大的艺术家是陈洪绶老莲清代版画普遍发展年画盛行于民间咸丰年间的任渭长一般认为是我国传统版画最后的一位大师以后的版画受到西方美术的影响和我国传统的风格是颇为不同了我手边有一部任渭长画的版画集卅三剑客图共有三十三个剑客的图形人物的造型十分生动偶有空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一九七〇年一月和二月是为明报晚报创刊最初两个月所作↵

图 3 预处理后

3.2 分词

本文选择"结巴（jieba）"中文分词模块，该模块可以支持三种分词模式：精确模式，试图将句子最精确地切开，适合文本分析；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；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。同时，由于金庸小说中包括部分繁体字，该模块可以支持繁体分词、支持自定义词典。

本文使用 jieba.cut()进行分词，例如对以下一句话：

"武林至尊宝刀屠龙号令天下莫敢不从倚天不出谁与争锋"

进行断句后得到：

['武林', '至尊', '宝刀', '屠龙', '号令', '天下', '莫敢', '不', '从', '倚天', '不出', '谁', '与', '争锋']

可以看出 jieba 可以对中文句子很好地进行分词操作。

3.3 计算信息熵

3.3.1 1-gram 模型

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，直接采用去掉文本中的所有标点，留下中文字符和 utf-8 编码

下字节数为 1 的标点符号，去掉这些符号后再进行繁简转换和分词，得到所需要的 txt 格式语料库。

依旧以上述例子进行说明，首先统计各个分词，直接使用上述分词后的结果：

['武林', '至尊', '宝刀', '屠龙', '号令', '天下', '莫敢', '不', '从', '倚天', '不出', '谁', '与', '争锋']

并通过下式对信息熵进行计算：

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

3.3.2 2-gram 模型

对于 2-gram 模型，需要对每一个词和其前一个词进行统计，这里通过“_”对其进行连接，对于示例语句得到的结果如下所示：

['武林_至尊', '至尊_宝刀', '宝刀_屠龙', '屠龙_号令', '号令_天下', '天下_莫敢', '莫敢_不', '不_从', '从_倚天', '倚天_不出', '不出_谁', '谁_与', '与_争锋']

并通过下式对信息熵进行计算：

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

3.3.3 3-gram 模型

对于 3-gram 模型，需要对每一个词和其前两个词进行统计，这里通过“_”对其进行连接，对于示例语句得到的结果如下所示：

['武林至尊_宝刀', '至尊宝刀_屠龙', '宝刀屠龙_号令', '屠龙号令_天下', '号令天下_莫敢', '天下莫敢_不', '莫敢不_从', '不从_倚天', '从倚天_不出', '倚天不出_谁', '不出谁_与', '谁与_争锋']

并通过下式对信息熵进行计算：

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

4 实验结果与分析

4.1 实验结果

通过 jieba 分词后得到结果如下所示:

表 1 通过 jieba 分词 1-gram 结果

词库总词数	4295637
不同词的个数	173229
出现频率前 10 的 1-gram 词语	
'的'	115593
'了'	104521
'他'	64714
'是'	64465
'道'	58618
'我'	57491
'你'	56679
'在'	43693
'也'	32605
'这'	32219
1-gram 信息熵	12.173700064022842

表 2 通过 jieba 分词 2-gram 结果

词库总词数	4236380
不同词的个数	1949059
出现频率前 10 的 2-gram 词语	
'道_你'	5776
'叫_道'	5009
'道_我'	4976
'笑_道'	4266
'听_得'	4202
'都_是'	3905
'了_他'	3664
'他_的'	3493
'也_是'	3201
'的_一声'	3102
2-gram 信息熵	6.946172001493654

表 3 通过 jieba 分词 3-gram 结果

词库总词数	4177410
不同词的个数	3472676
出现频率前 10 的 3-gram 词语	
'只听_得'	1611
'忽听_得'	1137
'站起身_来'	731
'哼了_一声'	578
'笑道_你'	570
'吃了_一惊'	534
'啊的_一声'	520
'点了_点头'	503
'说到_这里'	475
3-gram 信息熵	2.2980186245849845

直接将每一个字作为一个 token 后得到的结果如下所示:

表 4 直接分词 1-gram 结果

词库总词数	7283004
不同词的个数	5817
出现频率前 10 的 1-gram 词语	
'一'	139390
'不'	134140
'的'	121659
'是'	112702
'了'	111911
'道'	111054
'人'	84295
'他'	73562
'这'	68988
'我'	66993
1-gram 信息熵	9.535962449248872

表 5 直接分词 2-gram 结果

词库总词数	7223747
不同词的个数	729213
出现频率前 10 的 2-gram 词语	
'说_道'	13525
'了_一'	12187
'一_个'	10570
'道_你'	10320
'自_己'	10319
'小_宝'	9942
'韦_小'	9856
'也_不'	9303
'道_我'	8511
'笑_道'	8140
2-gram 信息熵	6.713136540255647

表 6 直接分词 3-gram 结果

词库总词数	7164490
不同词的个数	3165503
出现频率前 10 的 3-gram 词语	
'韦小_宝'	9803
'令狐_冲'	5890
'张无_忌'	4645
'的一_声'	3478
'袁承_志'	3037
'小宝_道'	2417
'陈家_洛'	2115
'小龙_女'	2081
'石破_天'	1818
'不由_得'	1803
3-gram 信息熵	3.939805705351134

4.2 分析

本文通过对比三种 n-gram 语言模型（1-gram、2-gram、3-gram）得到的结果，分析得出 n 的取值越大，即在估计时考虑的词数越多，则上下文之间的联系越多，不同词组合出现的种类个数也会越多，则文本的信息熵则越小。之所以出现 n 元模型计算该中文语料库的信息熵随着 n 的增大、总/平均信息熵减少的现象，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。并且从平均信息熵可以看出，1-gram 和 2-gram、3-gram 模型的平均信息熵差距较大。从 1-gram 和 2-gram 模型来看，分词得到的总信息熵大于按字分词得到的总信息熵，3-gram 的总信息熵按照 jieba 分词小于按字分词，从总信息熵来看较难比较分词和按字分词。但从平均信息熵来看，可以看到分词得到的平均信息熵比按字分词得到的平均信息熵小，说明按照字直接进行分词可以从该语料库中获取更多的信息。

4.3 探索

由于发现金庸小说中包含有部分繁体字，这里为了探索繁体字对与最终结果的影响，进行了以下实验。首先，根据维基百科中给出的繁体简体对照表，对数据集中的繁体字进行了替换。通过对比替换前后不同分词方式下的信息熵变化得到不同的结论。替换方式如下：

```
from langconv import *
def Traditional2Simplified(sentence):
  sentence = Converter('zh-hans').convert(sentence)
  return sentence
```

图 3 替换繁体字

最终结果如图所示：（使用 jeiba 进行分词）

表 7 替换繁体字前后结果

语言模型	参数	繁体替换前	繁体替换后
1-gram	词库总词数	4295637	4291200
	不同词的个数	173229	172576
	信息熵	12.173700064022842	12.171605088094893
2-gram	词库总词数	4236380	4231943
	不同词的个数	1949059	1946706
	信息熵	6.946172001493654	6.9474289219528975
3-gram	词库总词数	4177410	4172975
	不同词的个数	3472676	3469281
	信息熵	2.2980186245849845	2.297536417535415

可以看到，在使用繁体字替换之后，词库总词数和不同词的个数都有所减少。这可能是由于 jieba 分词模块对于简体字的识别率更高，从而减少了个别字单独划分的现象。同时，可以看到除了 2-gram 以外，替换繁体字的信息熵都有所减少。

5 总结

自然语言是一种上下文相关的信息表达和传递的方式，让计算机处理自然语言，一个基本的问题就是为自然语言这种上下文相关的特性建立数学模型，即统计语言模型。如何对信息进行量化度量是计算机对于信息进行高效处理的前提，一直以来研究者都在不断尝试寻找最为合理解决的方法，这引起了许多人的思考。衡量信息量即为衡量信息价值，信息带来的效益。一条信息的信息量大小和它的不确定性有直接的关系。比如说，我们要搞清楚一件非常非常不确定的事，或是

我们一无所知的事情，就需要了解大量的信息。相反，如果我们对某件事已经有了较多的了解，我们不需要太多的信息就能把它搞清楚。所以，从这个角度，我们可以认为，信息量的度量就等于不确定性的多少。

本文通过对比三种 n-gram 语言模型（1-gram、2-gram、3-gram）得到的结果，分析得出 n 的取值越大，即在估计时考虑的词数越多，则上下文之间的联系越多，不同词组合出现的种类个数也会越多，则文本的信息熵则越小。之所以出现 n 元模型计算该中文语料库的信息熵随着 n 的增大、总/平均信息熵减少的现象，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。并且从平均信息熵可以看出，1-gram 和 2-gram、3-gram 模型的平均信息熵差距较大。从 1-gram 和 2-gram 模型来看，分词得到的总信息熵大于按字分词得到的总信息熵，3-gram 的总信息熵按照 jieba 分词小于按字分词，从总信息熵来看较难比较分词和按字分词。但从平均信息熵来看，可以看到分词得到的平均信息熵比按字分词得到的平均信息熵小，说明按照字直接进行分词可以从该语料库中获取更多的信息。

6 参考文献

[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/qq_37098526/article/details/88633403

[3] https://blog.csdn.net/weixin_44966965/article/details/124007507