单位代码_____
学　　号 ZY2203811
分 类 号_____

北京航空航天大學
BEIHANG UNIVERSITY

计算金庸小说数据集的中文平均信息熵

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　许铁

2023 年 03 月

许铁 自动化科学与电气工程学院 ZY2203811

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。

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。

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

由于文件夹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

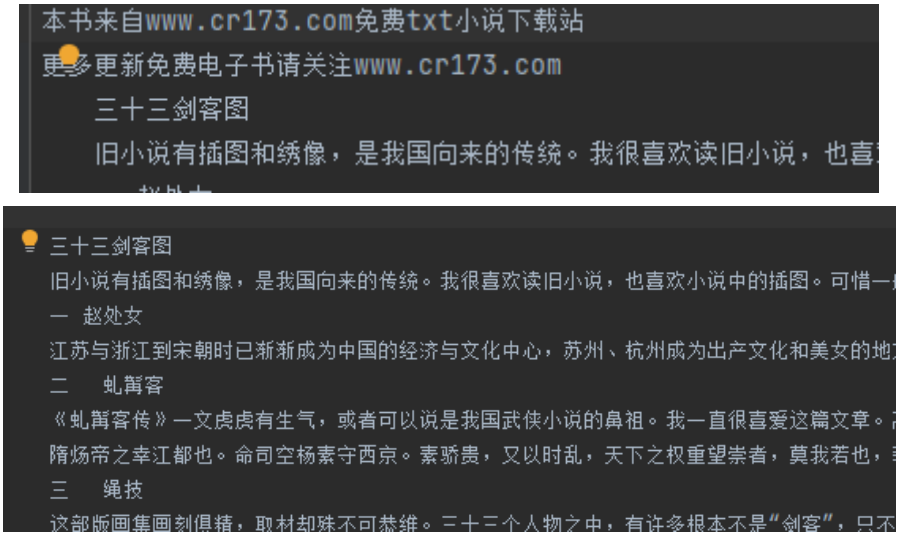

图 1 网页信息去除

根据提供的停用词表，去除包含信息量不大的停用词：

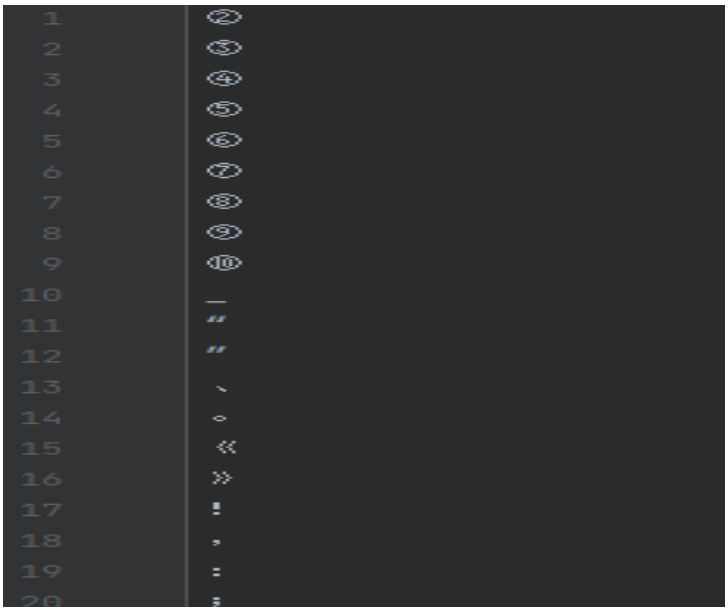

图 2 停用词表

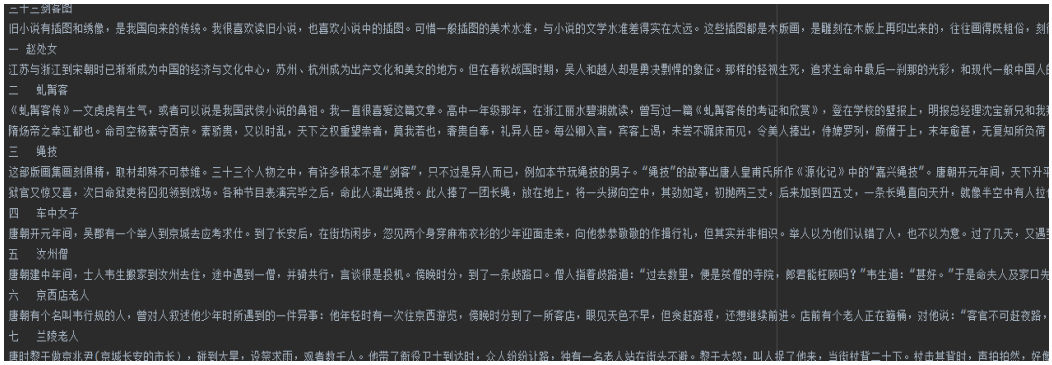

图 3 去除停用词前

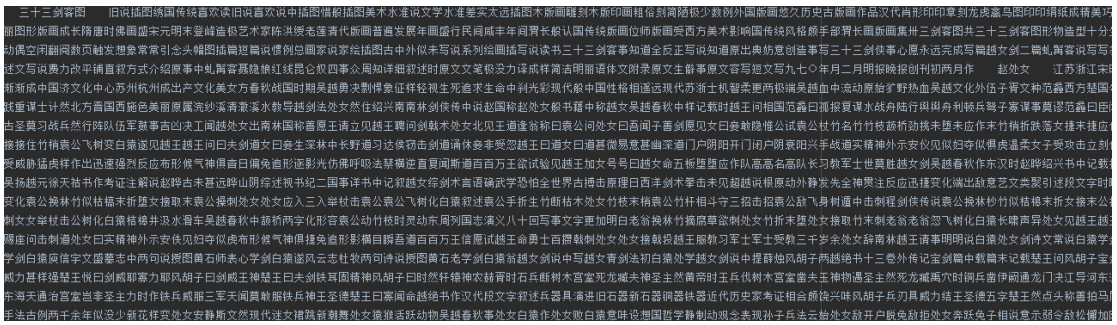

图 4 去除停用词后

3.2 分词

对数据预处理得到的文档进行分词。按词分词，使用 jieba.cut()进行分词，将分词后的词当成一个 token。按字分词，直接就将中文语料库中的每一个字当成一个 token。分词处理后的 token 都存入 split_words=[]中。 词频统计： unigram、bigram、trigram 模型分别要统计每个词、每个二元词组、每个三元词组出现的次数，得到词频表：words_tf={}，bigram_tf={}，trigram_tf={}

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，直接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
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，
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

采用 cn_stopwords 去除停词的实验结果如下所示：

表 1 通过 jieba 分词结果

	按照jieba分词	按字分词
语料库字数	4601062	4601642
分词字数	2545006	4601642
平均字长	1.80811	1

表 3 信息熵计算结果

	按照jieba分词	按字分词
一元模型长度	2545006	4601642
二元模型长度	2544990	4601626
三元模型长度	2544974	4601610
一元模型中文平均信息熵	13.52596 比特/词	9.86707 比特/词
二元模型中文平均信息熵	6.12781 比特/词	6.94823 比特/词
三元模型中文平均信息熵	1.28372 比特/词	3.50856 比特/词

采用 cn_punctuation 去除停词的实验结果如下所示：

表 4 通过 jieba 分词结果

	按照jieba分词	按字分词
语料库字数	7393695	7393695
分词字数	4402456	7393695
平均字长	1.67945	1

表 5 信息熵计算结果

	按照jieba分词	按字分词
一元模型长度	4402456	7393695
二元模型长度	4402440	7393679
三元模型长度	4402424	7393663
一元模型中文平均信息熵	12.0263 比特/词	9.49986 比特/词
二元模型中文平均信息熵	6.8896 比特/词	6.68025 比特/词
三元模型中文平均信息熵	2.40386 比特/词	3.94326 比特/词

5 总结

由信息熵计算结果可得，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，由该模型得到的信息量减少。并且从平均信息熵可以看出，一元模型和二、三元模型的平均信息熵差了一个或多个数量级，一元模型明显更适用于提取该中文语料库的信息。

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。

去除停词相比于直接去除标点符号，在一元模型、二元模型中区别不大，但是在三元模型中可以提升中文平均信息熵，说明去除信息量较少的停词（例如“一些”、“大概”等）可以提升信息密度，由前面的结论，n 的增加使得固定词组出现数量变多，当去除了大量无意义的量词组合等停词，平均信息熵随之上升。

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/weixin_42663984/article/details/115718241

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