# n元语言模型计算中文信息熵报告

ZY2203325-申海

[18231180@buaa.edu.cn](mailto:18231180@buaa.edu.cn)

## 摘要

本研究采用N-gram模型计算中文信息熵，并使用Python实现了该方法。我们首先使用os库读取指定路径下的所有txt文件，同时对.url文件做了特别处理，合并成一个文本文件。接着，使用jieba库对文本进行分词，并使用停用词表进行预处理。最后，使用2-gram模型计算文本的信息熵，并进行实验验证。实验结果表明，该方法可以有效地计算中文文本的信息熵。

## 介绍

如今，数据呈爆炸式增长，如何有效地处理、分析和提取信息成为重要的研究方向。其中，文本信息处理与分析是一项重要的任务。在中文文本处理中，N-gram模型是一种基于概率的模型，可以用于语言模型、文本分类、情感分析等任务中。

N-gram模型是一种基于统计的自然语言处理技术，它能够对一个句子或一个文本进行概率建模，用于描述该文本中每个词汇出现的概率以及相邻词汇出现的概率。N-gram模型将文本中相邻的N个词汇组合成一个序列，并统计该序列在文本中出现的频次和概率，从而可以计算文本的信息熵。在实际应用中，通常采用2-gram或3-gram模型进行文本处理。

本文主要介绍使用N-gram模型计算中文信息熵的方法。首先，我们使用Python中的jieba库进行中文分词，并结合停用词表进行文本预处理。接着，我们使用2-gram模型计算文本中每个词汇出现的概率和相邻词汇出现的概率，并通过公式计算文本的信息熵。最后，我们通过实验验证了该方法的有效性，结果表明该方法可以有效地计算中文文本的信息熵。

通过本文的介绍，读者可以了解到N-gram模型在中文文本处理中的应用，并了解如何使用Python实现计算中文信息熵的方法。该方法在文本处理、文本分类、情感分析等任务中具有广泛的应用价值。

## 方法

N-gram 模型是一种基于概率的语言模型，用于预测一个文本序列中下一个词出现的概率。它基于条件概率公式：

其中， 表示文本序列中的词， 表示该序列的概率， 表示前 个词组成的序列的概率。

N-gram 模型假设当前词出现的概率只与前 个词有关，即：

其中， 表示前 个词组成的序列。

对于其中可近似等于每个二元词组在语料库中出现的频率，条件概率可近似等于每个n元词组在语料库中出现的频率与以该n元词组的前n-1个词为词首的二元词组的频数的比值。

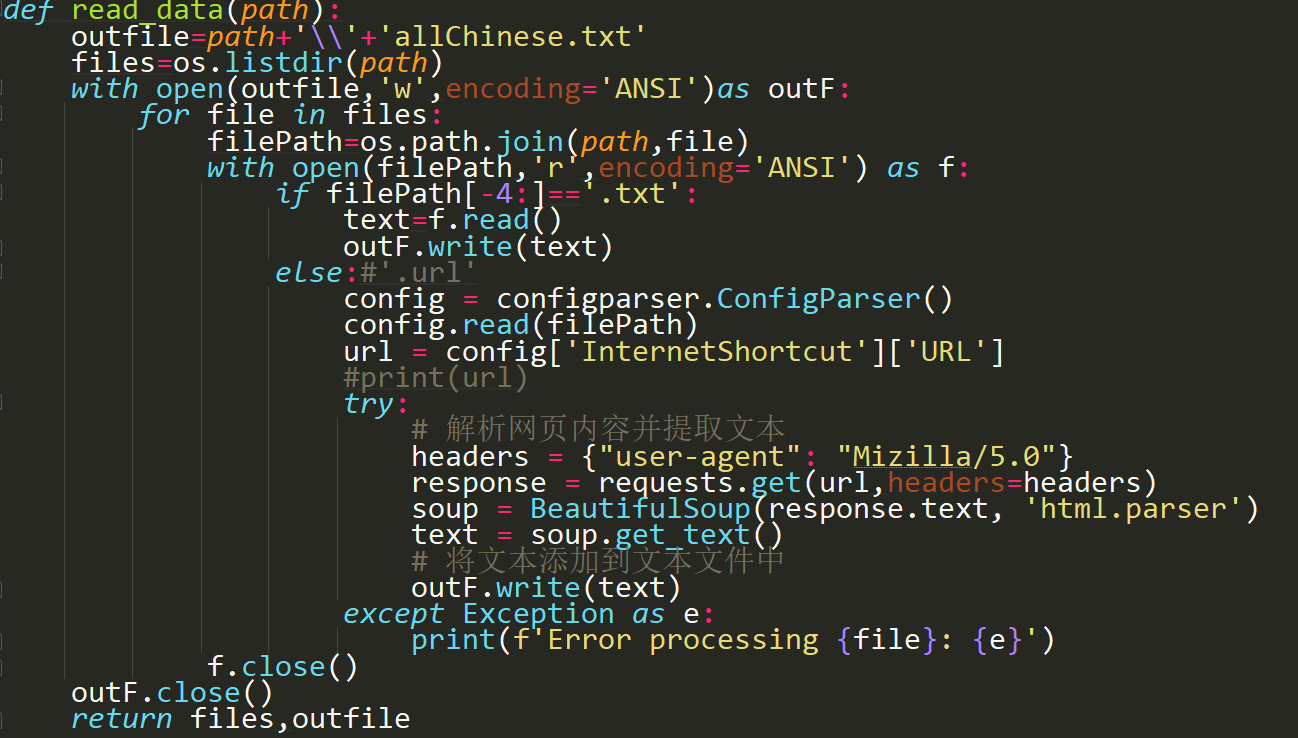
N-gram 模型通过统计语料库中不同词组合出现的次数来计算概率。例如，2-gram 模型就是统计相邻两个词组合出现的次数，并除以该词的前一个词出现的次数。3-gram 模型类似，统计相邻三个词组合出现的次数，并除以前两个词组成的序列出现的次数。

## 实验研究

使用给定得语料库和停词表，将语料库读取整合成一个txt文件。然后使用python中的jieba库进行中文分词，使用停词表去除停词。最后n-gram语言模型进行中文信息熵的计算。具体过程如下：

1. 读取语料库和停词表

使用os库给定语料库所在文件夹，使用os.path.listdir获取所有文件名，然后读取并整合成一个txt文件，对于后缀名为.url的文件，使用configparser和requests库进行处理。读取函数如下图。



1. 中文分词

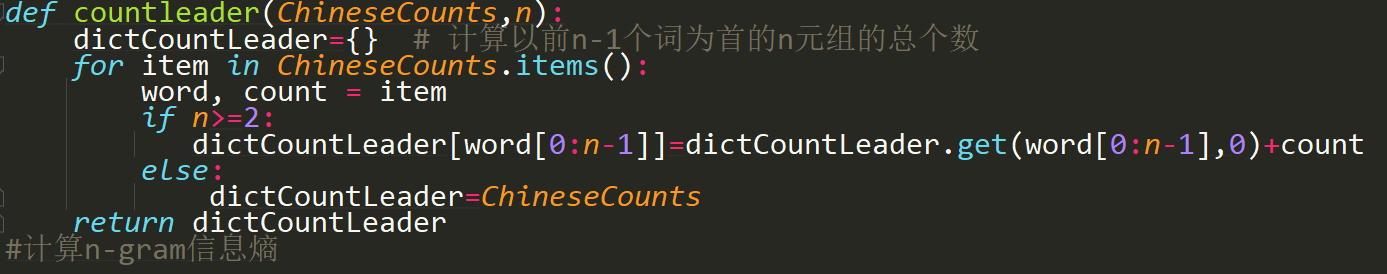
使用jieba.lcut进行中文分词并判断是否处于停词表中，如果处于停词表中其舍去。

1. 计算中文信息熵

使用一个字典保存n个词序列和出现次数，字典的key为n个词组成的n元组，字典的value为出现的次数。同时使用一个字典保存该字典中key中前n-1个词为首组成的n词序列和其出现的总次数，最后使用

进行信息熵计算。

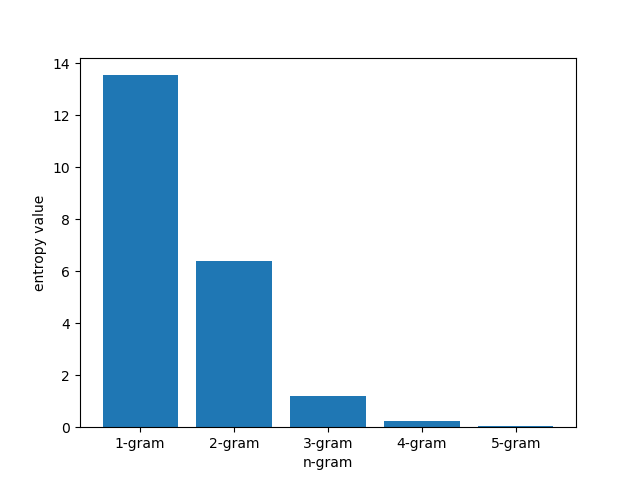
保存以n-1词为首的n个词序列的子函数和获取信息熵的子函数如下图。





## 结论

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1-gram | 2-gram | 3-gram |
| 中文信息熵值 | 13.54 | 6.38 | 1.18 |
| 运行用时 | 32.5s | 40.5 | 55.9s |



通过实验结果可以发现，2-gram模型的计算信息熵小于1-gram,这与实际情况是符合的。随着n的增加，信息量将减少。使用N-gram模型计算中文信息熵的方法可以有效地处理中文文本，并且该方法的计算结果与实际情况相符。因此，该方法具有很好的实用价值，可以应用于文本处理、文本分类、情感分析等任务中。

## 参考书

Cover, T. M., & King, R. G. (1992). An estimate of an upper bound for the entropy of English. Information Theory, IEEE Transactions on, 38(2), 587-591.