# **Report of Deep Learning for Natural Language Processing**

曹思远

[3203998114@qq.com](mailto:3203998114@qq.com)

## **Abstract**

本文档共分为两个部分，第一部分是通过提供的中文语料库来验证Zipf's Law.；第二部分是通过阅读Entropy Of English，分别计算中文(以词和字为单位)的平均信息熵。对于第一部分作业，是利用 Python及其jieba库对作业提供的中文语料库进行精准分词，后通过图像显示来验证Zipf's Law.定律；对于第二部分作业，同样是利用Pathon及其jieba库对作业提供的中文语料库进行精准分词，后依次通过1-gram、2-gram与3-gram统计语言模型针对上述中文语料库分别以汉字和词语两个方面计算中文的信息熵，其中针对2-gram与3-gram计算的是条件信息熵。

# **Introduction**

**Zipf's Law**

Zipf's Law（齐夫定律）是由美国语言学家George Zipf提出的经验定律，描述了自然语言中单词使用频率与它们在频率排序表中的排名之间的关系。Zipf's Law有以下特点：

1.在一个给定的自然语言语料库中，排名第二的单词的出现频率大约是排名第一的单词的频率的一半，排名第三的单词的频率大约是排名第一的单词的频率的三分之一，以此类推。换句话说，单词的使用频率与其在频率排序表中的排名成反比关系。

2.这个定律适用于各种自然语言，包括英语、中文等。

3.Zipf's Law的应用不仅局限于自然语言，还可以在其他领域如城市人口规模、公司规模等方面找到类似的规律。

4.尽管Zipf's Law是一个经验定律，但它对于自然语言处理、信息检索等领域具有重要意义，可以帮助我们理解和分析文本数据中单词的分布规律。

总的来说，Zipf's Law提供了一种简洁而有力的描述自然语言中单词使用频率的规律，对于文本数据的分析和处理具有重要的指导意义。

**信息熵**

信息熵是信息论中的一个重要概念，用于衡量随机变量的不确定性或信息量。信息熵由克劳德·香农（Claude Shannon）在他的信息论中首次提出。

信息熵的定义如下：

对于一个离散随机变量X，其取值集合为{ , , ..., }，概率分布为P(X)，信息熵H(X)定义为：

其中，) 是随机变量X取值为的概率。

**N-gram 语言模型**

根据大数定理，当统计量足够大的时候，词、二元词组、三元词组出现的概率大致等于其出现的频率。

则有一元模型的信息熵计算公式为：

其中P(x)可近似于每个词在语料库中的出现频率

二元模型的条件信息熵计算公式为：

其中联合概率P(x,y)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P(x|y)可近似等于每个二元词组在语料库中出现的频率与以该二元词组的第一个词为词首的二元词组的频数的比值。

三元模型的条件信息熵计算公式为：

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|y,z)可近似等于每个三元词组在语料库中出现的频率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

# **Methodology**

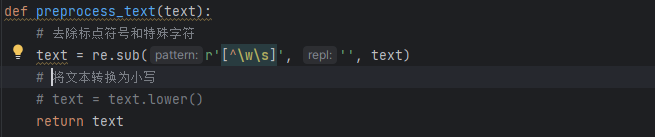
## **M1:通过中文语料库来验证Zipf’s Law**

1.文本的加载

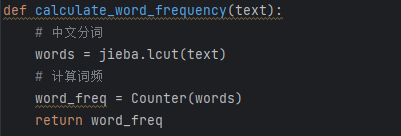


2.中文语料的预处理

数据库中的语料是txt格式的，其中包含得有部分代码和无用的符号，首先需要对数据进行一下预处理，其中包括删除隐藏符号（如换行符、分页符等）



3.利用jieba精准分词并计算词的频率



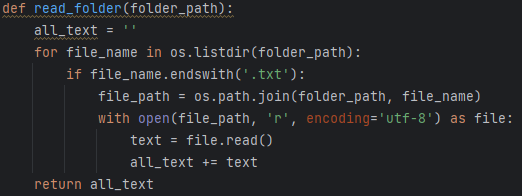
4.结果可视化

通过 matplotlib 将词频-排名可视化，均使用log化，方便验证其乘积为常数。仅取排名前5000的显示

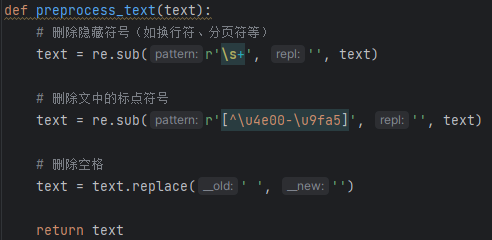


## **M2:基于 N-gram 模型的中文信息熵计算**

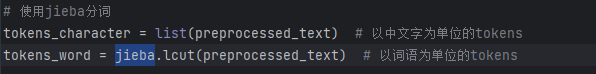
1. 读取文件夹中的所有txt文件并整合在一起



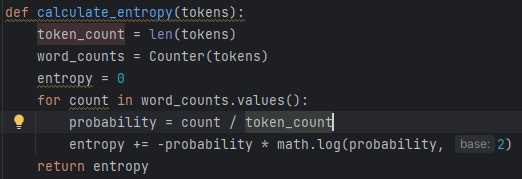
2.数据库中的语料是txt格式的，其中包含得有部分代码和无用的符号，首先需要对数据进行一下预处理，其中包括删除隐藏符号（如换行符、分页符等）；删除文中的标点符号；删除空格。

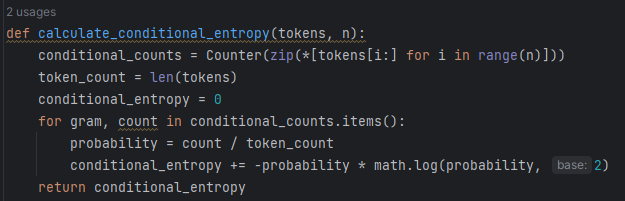


3.利用jieba精准分词，生成字的库和词库



4.1-gram计算信息熵；2-gram、3-gram计算条件信息熵



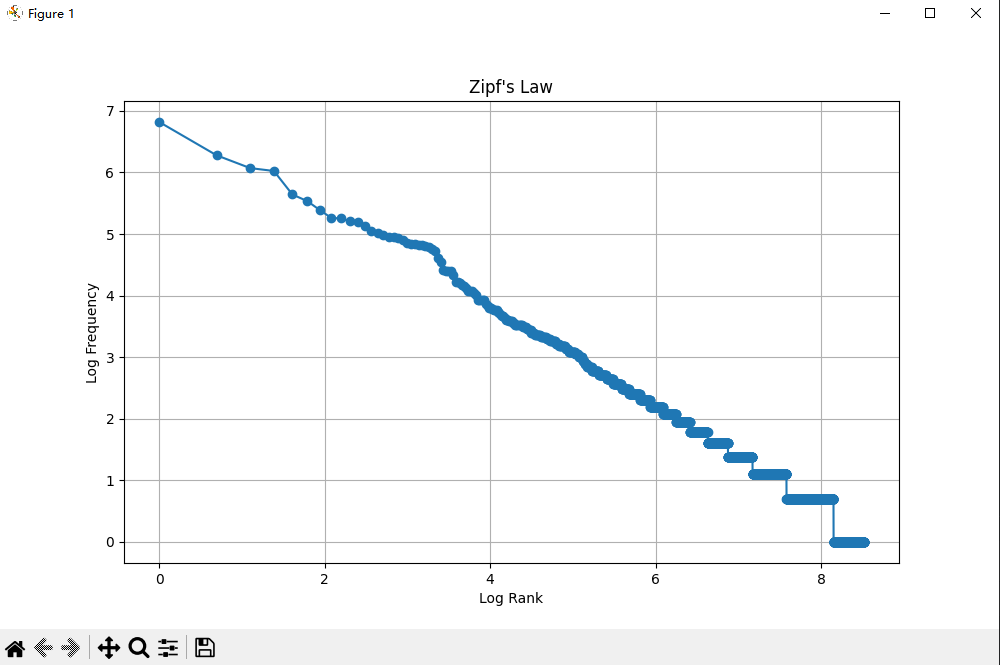


5.数据显示

输出以字和词为单元的排名频率前10的；并且输出相应的信息熵。

# **Experimental Studies**

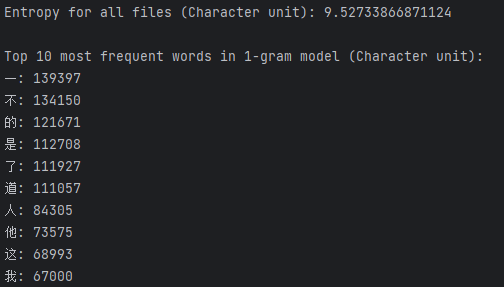
## **M1:通过中文语料库来验证Zipf’s Law**



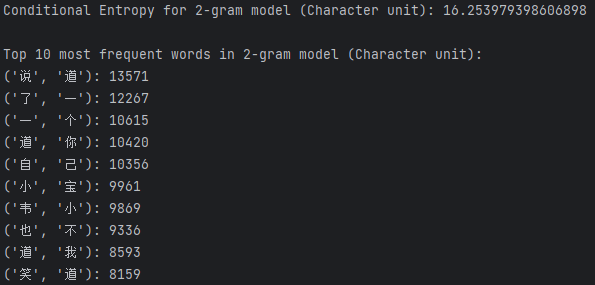
可以从词频-排名的对数坐标下，看到近似成一条直线，两者是成反比的

## **M2:基于 N-gram 模型的中文信息熵计算**

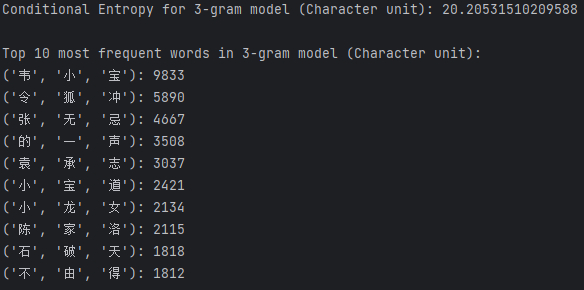
字的1-gram信息熵**：**



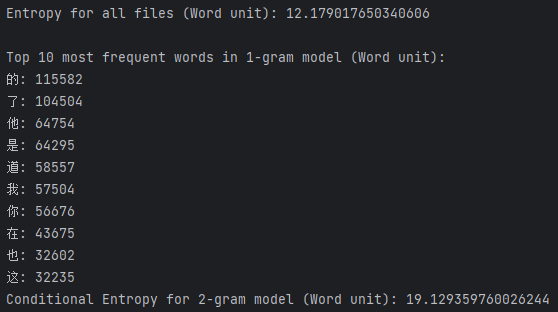
字的2-gram条件信息熵：



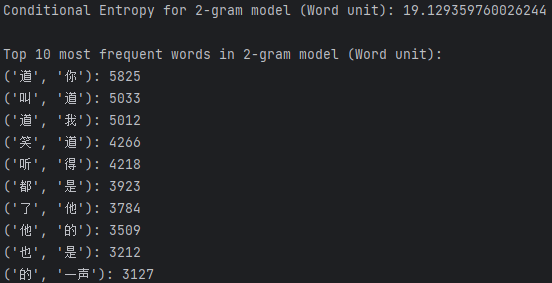
字的3-gram条件信息熵：



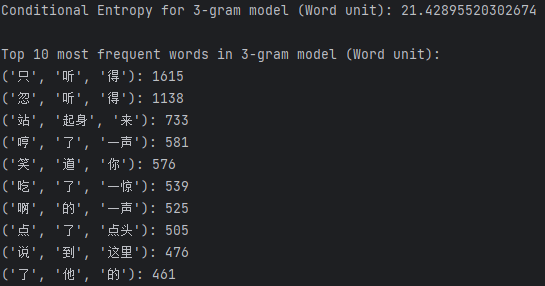
词的1-gram信息熵：



词的2-gram条件信息熵：



词的3-gram条件信息熵：



对比1-Gram、2-Gram、3-Gram三种语言模型得到的结果可以看出，在考虑条件信息熵后，2-Gram、3-Gram的熵值在增大，可能是因为存在一些两个词或三个词的固定搭配，这样在算相应的条件概率时值可能会增大。

# **Conclusion**

综上所述，本作业利用 Python 及其提 jieba 库对提供的中文语料库进行了分词与词频统计，验证了中文下的 Zipf’s Law定理；其次基于1-gram、2-gram和3-gram统计语言模型在上述语料库中以字与词两个方面计算了中文的信息熵，并分析两个实验结果。

# **Referances**

1.An Estimate of an Upper Bound for the Entropy of English

2.[深度学习与自然语言处理第一次作业——中文平均信息熵的计算\_中文信源的独立熵要算标点符号吗-CSDN博客](https://blog.csdn.net/weixin_50891266/article/details/115723958?ops_request_misc=%7B%22request%5Fid%22%3A%22164863247216780255276249%22%2C%22scm%22%3A%2220140713.130102334.pc%5Fall.%22%7D&request_id=164863247216780255276249&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2allfirst_rank_ecpm_v1~rank_v31_ecpm-1-115723958.142v5pc_search_result_cache,143v6register&utm_term=Entropy+of+English&spm=1018.2226.3001.4187 %E2%80%94%E2%80%94%E2%80%94%E2%80%94%E2%80%94%E2%80%94%E2%80%94%E2%80%94%E2%80%94%E2%80%94%E2%80%94%E2%80%94%E2%80%94%E2%80%94%E2%80%94%E2%80%94 %E7%89%88%E6%9D%83%E5%A3%B0%E6%98%8E%EF%BC%9A%E6%9C%AC%E6%96%87%E4%B8%BA%E5%8D%9A%E4%B8%BB%E5%8E%9F%E5%88%9B%E6%96%87%E7%AB%A0%EF%BC%8C%E9%81%B5%E5%BE%AA CC 4.0 BY-SA %E7%89%88%E6%9D%83%E5%8D%8F%E8%AE%AE%EF%BC%8C%E8%BD%AC%E8%BD%BD%E8%AF%B7%E9%99%84%E4%B8%8A%E5%8E%9F%E6%96%87%E5%87%BA%E5%A4%84%E9%93%BE%E6%8E%A5%E5%92%8C%E6%9C%AC%E5%A3%B0%E6%98%8E%E3%80%82 %E5%8E%9F%E6%96%87%E9%93%BE%E6%8E%A5%EF%BC%9Ahttps://blog.csdn.net/weixin_44966965/article/details/124007507)

3.[深度学习与自然语言处理作业第一次作业——中文平均信息熵计算\_自然语言 平均每字符信息熵-CSDN博客](https://blog.csdn.net/weixin_44966965/article/details/124007507)