ReportofDeepLearningforNaturalLangaugeProcessing

作业1中文语料库的Zipf'sLaw与Entropy

李祎柔
lyr478144791@163.com

Abstract

本次作业旨在利用若干小说组成的中文语料库验证ZipfsLaw，并分别计算以词和字为单位的熵。结果表明，去除停用词后的中文的确满足ZipfsLaw所描述的规律;中文以“词”和“字”为符号的信息熵分别大致为11bit和9bit。

PART1Zipf'slaw

Introduction

齐夫定律（Zipf'slaw）是由哈佛大学的语言学家乔治·金斯利·齐夫于1949年发表的实验定律。它可以表述为：在自然语言的语料库里，一个单词出现的频率与它在频率表里的排名成反比。所以，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2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2倍。这个定律被作为任何与幂定律概率分布有关的事物的参考。

Methodology

Zipf定律（Zipf'sLaw）是指在自然语言中，单词出现的频率与其频率排名之间的关系呈现一种特定的分布。Zipf定律表明，排名第n的词的频率与排名第一的词的频率成反比关系，即频率大约与排名的倒数成正比。

Result

所绘制的词频与排序的关系曲线如图1所示。可以看到，在对数尺度下，词频与排序确实大致呈现出一个斜率为负的一次函数。验证了ZipfsLaw的正确性。

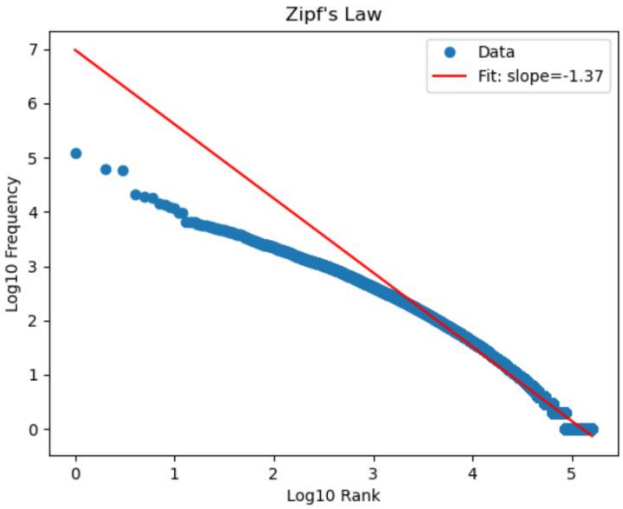

图1中文语料库的Zipf'sLaw曲线

PART2Entory

Introduction

信息熵是信息论中一个非常重要的概念，它可以用来衡量随机变量的不确定度或信息量。熵的概念最初是由克劳德香农在1948年提出的，它被广泛应用于通信、密码学、数据压缩、信号处理等领域。

在信息论中，熵的定义是一个离散随机变量的平均信息量。它可以被看作是描述随机事件的不确定性的量度。

对于一个离散型随机变量X，其信息熵的定义为

$$H(X)=-\sum_i^n P(x_i)\log P(x_i)$$

Methodology

假定S表示某个有意义的句子，由一连串特定顺序排列的字或者词$\omega_1,\omega_2,\omega_3,...\omega_n$组成这里$n$是组成句子的字或词的数量。现在我们想知道$s$在文本中出现的可能性，即：

$$P(s)=P(\omega_1,\omega_2,\omega_3,...\omega_n)$$

利用条件概率公式可以得到：

$$P(\omega_1,\omega_2,\omega_3,...\omega_n)=P(\omega_1)P(\omega_2\mid\omega_1)...P(\omega_n\mid\omega_1,\omega_2,\omega_3,...\omega_{n-1})$$

下面以词为例（字与其原理相同），进行理论分析。其中$P(\omega_1)$表示第一个词ω_1出现的概率；$P(\omega_2|\omega_1)$是在已知第一个词的前提下，第二个词出现的概率，以此类推$P(\omega_n|\omega_1,\omega_2,\omega_3,\ldots\omega_n)$是在已知第1,2,3...n-1个词的前提下，第$n$个词出现的概率。当计算$P(\omega_1)$时，仅存在一个参数；计算$P(\omega_2|\omega_1)$，存在两个参数，以此类推，在句子又多又长的情况下，难易计算。

所以马尔可夫提出一种假设：假设N元模型的每个词出现的概率只与前面N-1个词相关，当N=2时，就是二元模型，N=3就是三元模型。N元模型，即当前这个词ωi依赖于前面N-1个词，上述N=1为与前面单词都没有关系，N=2表示与前面一个单词有关；N=3表示与前面两个词有关。选取不同模型最终结果也会不同。

当N=1时，一元模型的数学表达如下所示：

$$P(s)=P(\omega_1)P(\omega_2)P(\omega_3)...P(\omega_i)...P(\omega_n)$$

而当N=2时，二元模型的数学表达如下所示：

$$P(s)=P(\omega_1)P(\omega_2\mid\omega_1)P(\omega_3\mid\omega_2)...P(\omega_i\mid\omega_{i-1})...P(\omega_n\mid\omega_{n-1})$$

而当N=3时，三元模型的数学表达如下所示：

$$P(s)=P(\omega_1)P(\omega_2\mid\omega_1)P(\omega_3\mid\omega_2,\omega_1)...P(\omega_i\mid\omega_{i-1},\omega_{i-2})...P(\omega_n\mid\omega_{n-1},\omega_{n-2})$$

如果统计量足够，字、词、二元词组或三元词组出现的概率大致等于其在语料库中出现的频率。

下面以词为例（字与其原理相同），进行理论分析。

一元模型的信息熵计算公式如下

$$H(X)=-\sum P(x)\log P(x)$$

其中$P(x)$可近似等于每个词组在语料库中出现的频率。

二元模型的信息熵计算公式如下

$$H(X\mid Y)=-\sum P(x,y)\log P(x\mid y)$$

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数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。

三元模型的信息熵计算公式如下

$$H(X|Y,Z)=-\sum P(x,y,z)\log P(x|y,z)$$

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数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。

Result

表1 计算所得的信息熵

字/词	数量	信息熵（bits）
字	5650166	9.85
一元词	2819193	13.77197
二元词	2819176	6.41183
三元词	2819159	1.10362

通过对比字和一元词的信息熵可以发现，一元词的大于字的，基于字模型的信息熵要低于基于一元词模型的信息熵，主要有以下几个原因：

字的种类比词的种类要少。汉字的种类有几千种，而常用的字不到一千个。相比之下，汉语的词汇量非常庞大，达到了数十万或者更多，所以基于一元词模型的信息熵会更大。字在不同的上下文中具有更加确定的含义。相对于词而言，字可以更加准确地表示某个概念。因为在不同的上下文中，同一个词可能具有不同的含义，而同一个字在不同的上下文中一般来说只有一个含义。

基于字模型更容易识别新词。如果在使用基于词模型的语言模型时遇到了新词，模型就会无法处理。但是基于字模型的语言模型在遇到新词时可以通过组合已有的字来表示新词，因此更加灵活和准确。

而二元模型的信息熵要低于一元模型，是因为在二元模型中，每个字符的出现概率与其前一个字符有关。因此，二元模型在语言建模中能够更好地捕捉到字符之间的关系，使得相邻字符之间的信息冗余量减少，从而降低了信息熵。而在一元模型中，每个字符的出现概率只与该字符本身有关，不能捕捉到字符之间的关系，因此相邻字符之间的信息冗余量较大，导致信息熵较高。三元模型信息熵要低于二元模型的原因与之同理。