Report of Deep Learning for Natural Langauge Processing：

Verify Zipf’s Law and Compute Average Information Entropy

Yuqi Shi
1650278787@qq.com

Abstract

This is a Report of Deep Learning for Natural Langauge Processing about verifying Zipf's Law and computing Average Information Entropy.This report including two parts.Firstly,verify Zipf’s Law by a Chinese corpus which includes 16 novels of Jin Yong.Secondly,the Average Information Entropy on characters and words are calculated by applying 1-gram,2-gram and 3-gram model respectively.

Introduction

Zipf's Law, proposed by American scholar G.K. Zipf in the 1940s, is a word frequency distribution law. It can be formulated as follows: If the frequency of each word appearing in a longer article is counted, and the words are arranged in descending order according to the high-frequency words appearing first and the low-frequency words appearing later, and natural numbers are assigned to these words as rank serial numbers, with the rank of the highest frequency word being 1, the rank of the next highest frequency word being 2, ..., and the rank of the lowest frequency word being D. If f represents frequency and r represents rank serial number, then

$$f * r = C$$

(where C is a constant). This equation is referred to as Zipf's Law.

Information entropy, originally proposed by Claude Shannon (1916-2001) in 1948, drawing inspiration from the concept of "entropy" in thermodynamics. It aims to quantify the uncertainty of information. The greater the entropy value, the greater the uncertainty of the information. Its mathematical formula can be expressed as:

$$H(x) = \sum_{x \in X} P(x) \log\left(\frac{1}{P(x)}\right) = -\sum_{x \in X} P(x) \log(P(x))$$

Random variables associated with joint distributions$(X,Y)\sim P(X,Y)$.In the case of two variables being mutually independent, their joint entropy is:

$$\begin{aligned} H(X|Y) &= -\sum_{y\in Y} P(y)\log(P(x|y)) \\ &= -\sum_{y\in Y} P(y)\sum_{x\in X} P(x)\log(P(x|y)) \\ &= -\sum_{y\in Y}\sum_{x\in X} P(x)P(y)\log(P(x|y)) \\ &= -\sum_{y\in Y}\sum_{x\in X} P(x,y)\log(P(x|y)) \end{aligned}$$

The joint entropy can be used for the calculation of the 2-gram and 3-gram models in subsequent sections.

Methodology

The research will be introduced in 4 parts :Chinese corpus data processing,Zips'f Law verification, the average information entropy calculation and results visualization.In addition,I applied 3 models to calculate the average information entropy.

Preprocessing of the dataset involves the following steps:

a. Removing all hidden symbols.

b. Deleting all non-Chinese characters.

b. Removing all punctuation marks without considering contextual relationships.

For this preprocessing, we will use the Jieba package for Chinese word segmentation. Jieba is a Chinese word segmentation package in Python, and we will use the precise mode for segmentation in this experiment.

M1: 1-gram Model

The formula for calculating the entropy of a 1-gram model is:

$$H(x) = -\sum_{x\in X} P(x)logP(x)$$

Where P(x) can be approximated by the occurrence frequency of each word in the corpus.

M2: 2-gram Model

The formula for calculating the entropy of a 2-gram model is:

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

Where the joint probability P(x,y)can be approximated by the frequency of occurrence of each bigram in the corpus, and the conditional probability P(x|y) can be approximated by the ratio of the frequency of each bigram in the corpus to the frequency of bigrams where the first word of the bigram is the same as y .

M3: 3-gram Model

The formula for calculating the entropy of a 3-gram model is:

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

Where the joint probability P(x,y,z) can be approximated by the frequency of occurrence of each trigram in the corpus, and the conditional probability P(x|y,z) can be approximated by the ratio of the frequency of each trigram in the corpus to the frequency of trigrams where the first two words of the trigram are the same as y and z.

Experimental Studies

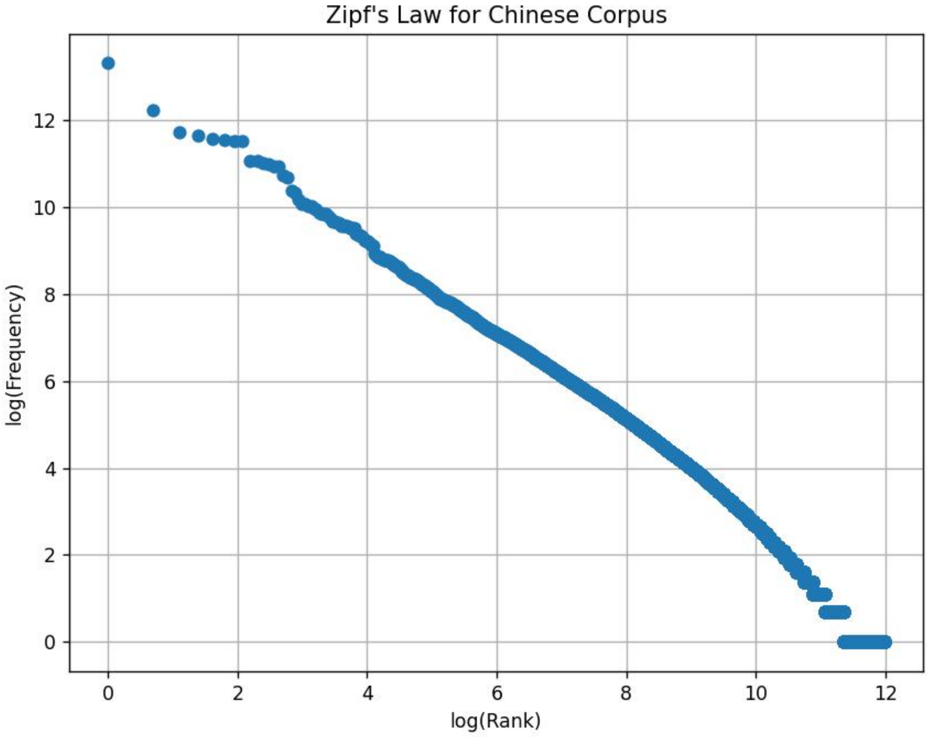

Figure 1： Zipf’s Law Vertification curve

Comparing the results obtained from the 1-gram, 2-gram, and 3-gram language models, it can be observed that as the value of N increases, indicating a longer consideration of contextual relationships, the number of different words also increases. This is because with the increase in length, the number of possible combinations of characters forming words increases, resulting in a greater variety of words.

Furthermore, from the comparison of the three models, it can be seen that as the value of N increases, the entropy of the text decreases. This is because when N is larger, the distribution of word combinations in the segmented text becomes simpler. Analysis reveals that as N increases, the number of fixed words that can be formed decreases. With fewer fixed words, there is less chance for individual characters or short words to disrupt the text. Consequently, the text becomes more ordered, reducing the uncertainty of character-word and word-sentence composition, thereby lowering the entropy of the text.

Table 1:Average information entropy

New Table	1-gram	2-gram	3-gram
Average information entropy (per character)	9.53837072218095	8.129679915435238	8.60508379478606
Average information entropy (per word)	12.166792988733498	6.94497412025617	2.3034871425566523

Conclusions

A Chinese corpus which includes 16 novels of Jin Yong is used as the data to verify the Zipf’s Law.Meanwhile, the Average Information Entropy on characters and words are calculated by applying 1-gram,2-gram and 3-gram model respectively.

References

[1] Brown, P. F., Della Pietra, V, .., Mercer, R. L., Della Pietra, S A., & Lai, J.C. (1992). An Estimate of an Upper Bound for the Entropy of English. Computational Linguistics, 18(1), 31-40