Report of Deep Learning for Natural Langauge Processing

Yongquan Chen
cyqss@buaa.edu.cn

Abstract

This report examines the linguistic features of Chinese through Zipf's Law and information entropy using a deep learning approach in NLP. The first part analyzes the frequency distribution of words and characters in Jinyong's novels to validate Zipf's Law, revealing its applicability mainly among high-frequency terms. The second part calculates the information entropy of Chinese, using a tri-gram model for both characters and words. Preliminary findings suggest insights into the complexity of Chinese, despite challenges in direct comparison with English entropy. This study highlights the potential of applying deep learning to understand linguistic patterns in NLP.

Introduction

This report is divided into two main parts. The first part aims to verify Zipf's Law using a Chinese text corpus. Zipf's Law suggests that the frequency of any word is inversely proportional to its rank in the frequency table. This law is observed in various languages, and this report seeks to explore its applicability to Chinese. The second part of the report focuses on reading "Entropy Of English" and calculating the average information entropy for Chinese, considering both characters and words as units of analysis. Information entropy is a measure of unpredictability or information content, and this analysis aims to provide insights into the complexity and information density of the Chinese language.

Methodology

Part 1: Verify Zipf's Law in Chinese

To verify Zipf's Law, a corpus of Chinese text, Jinyong's books, was analyzed. The text was first preprocessed to remove any non-Chinese characters, such as punctuation and spaces, given that Chinese is written without the latter. Then words that have no practical meaning, such as stopwords, have also been removed. The corpus was then segmented into words by Jieba,which is a Chinese text segmentation tool, as Chinese does not use spaces between words. Each word's frequency was calculated, and the words were ranked based on their frequencies. I set up two modes to calculate the frequency of Chinese characters and the frequency of words.Two log-log plots of the rank versus frequency were both generated to visually inspect Zipf's Law's applicability.

Part 2: Calculate the Average Information Entropy of Chinese

To calculate the average information entropy of Chinese using a tri-gram model, both for characters and words, we'll follow a new approach. This analysis involves processing Jinyong’s corpus of Chinese text to ensure a broad and representative sample of the language. Given the unique characteristics of Chinese, including its logographic writing system and absence of spaces between words, Jieba library is employed for character and word-level analyses.

1.Tri-gram Generation

A tri-gram in this context refers to a sequence of three consecutive Chinese characters. The entire corpus is processed to extract every possible tri-gram, considering the overlap between sequences.In this section, words and characters are counted separately.

2.Frequency Calculation

Count the frequency of each unique tri-gram throughout the corpus. This step is crucial for understanding the distribution of tri-gram occurrences.

3.Probability Estimation

Calculate the probability of each tri-gram by dividing its frequency by the total number of tri-grams. This step transforms raw frequencies into a measure of how likely a tri-gram is to occur in the language.

4.Entropy Calculation

The entropy is calculated using the formula:.

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

Among them,H represents the entropy, $p(x)$ is the probability of occurrence of each tri-gram, joint probability $P(x,y,z)$can be approximated equal to the frequency of each ternary phrase in the corpus, conditional probability $P(x|y,z)$ can be approximated as the ratio of the frequency of each ternary phrase in the corpus to the frequency of the triplet that begins with the first two words of the ternary phrase.

Conclusions

In the first part, according to the generated graph, it can be seen that the first 1000 high-frequency words obey Zipf's law, but the low-frequency words are anomalies caused by insufficient data because of the small number of occurrences.

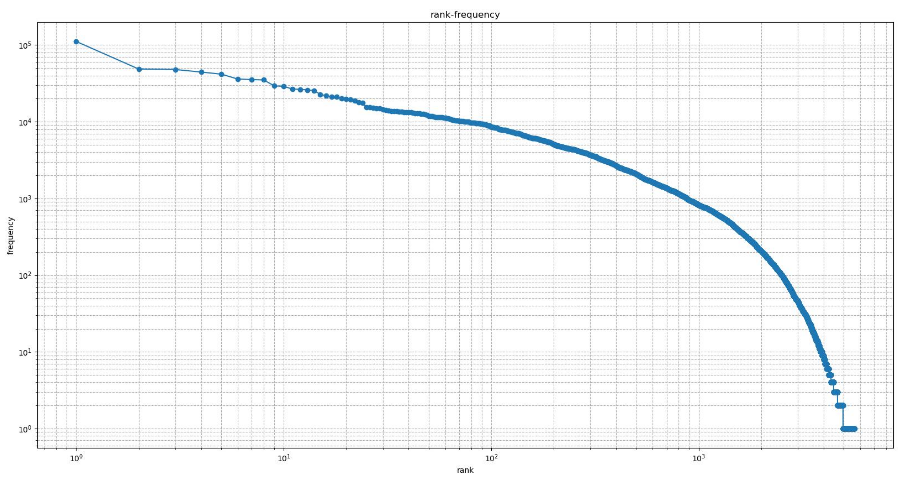

Figure 1： rank of Chinese characters vs. frequency

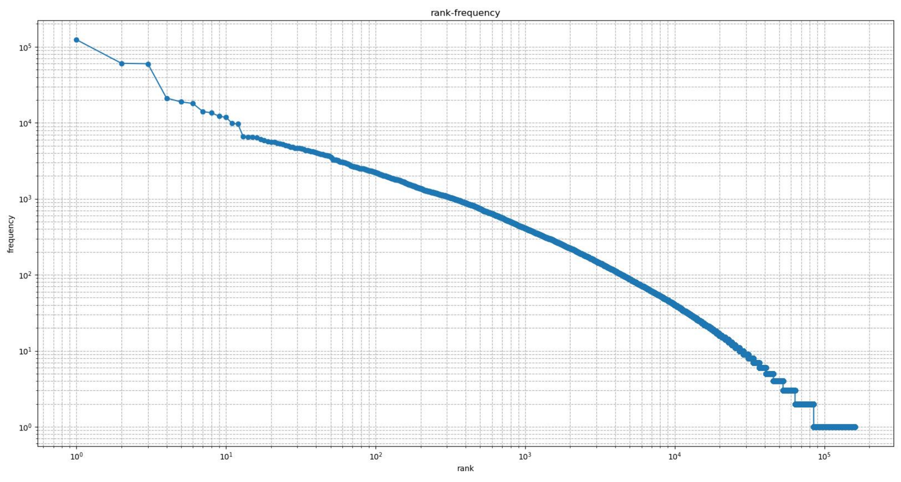

Figure 2： rank of Chinese words vs. frequency

In the second part, Comparing the char model and word model, I can’t see that Chinese’s entropy is better than English .Maybe I do something wrong.The application of a tri-gram model to calculate the average information entropy of Chinese, at both the character and word levels, provides valuable insights into the language's structure and complexity.

The calculated information entropy of words is smaller than that of characters. The reason is guessed that the more words that make up the phrase, the smaller its redundancy, the smaller the scope of use, and the smaller the uncertainty of appearing in the article.In addition, experiments have shown that clearing stop words and splitting sentences also leads to a reduction in information entropy. It is speculated that this method may cause each sentence to become shorter and reduce the scope of use.

Table 1: entropy after removing stopwords

	words	characters
三十三剑客图	-0.8172216645095713	0.25252881986560816
书剑恩仇录	-0.4178306086422391	1.4255154943406445
侠客行	-0.3454550251504398	1.345379007305063
倚天屠龙记	-0.19698791057129603	1.7888368203873148
天龙八部	-0.23238110066337625	1.8018325142293075
射雕英雄传	-0.3561275006955291	1.7644822611544384
白马啸西风	-0.40042838101299616	0.8344087257766142
碧血剑	-0.4865939369421253	1.3792879363272734
神雕侠侣	-0.17770615992326533	1.7819695421773583
笑傲江湖	-0.16907745521695305	1.8261699024235878
越女剑	-0.4754756283889604	0.5218881761869096
连城诀	-0.515333887074352	1.2533535805099887
雪山飞狐	-0.5402445581945754	0.9164238292618925
飞狐外传	-0.3641131402531958	1.4367828043039244
鸳鸯刀	-0.23312856007543678	0.49939374542736725
鹿鼎记	-0.23312856007543678	1.7889968050203204
all	0.196180209560775	2.732500350572619

Table 2: entropy after removing stopwords and splitting sentences

	words	characters
三十三剑客图	0.09083811441735534	0.6498437593496372
书剑恩仇录	0.4973640615095344	1.8612894868516765
侠客行	0.5124190074374463	1.819103092132845
倚天屠龙记	0.6440069251408704	2.2759386964620623
天龙八部	0.6632757305450651	2.351383271822965
射雕英雄传	0.5377051134355519	2.1958925023449427
白马啸西风	0.36992675520670965	1.2095027278567736
碧血剑	0.43068120821175604	1.7951478739007365
神雕侠侣	0.63628805397432	2.282397259448817
笑傲江湖	0.7954510508489345	2.3611940099884903
越女剑	0.24269578769874317	0.8369509109429615
连城诀	0.36850774505069295	1.6389494518915042
雪山飞狐	0.2905589630569319	1.3020994868686084
飞狐外传	0.4608462327550753	1.8649720362946534
鸳鸯刀	0.2428245940829211	0.8935978672381392
鹿鼎记	0.837987390083926	2.4093791096099
all	1.1750022340421034	3.4933403642525125

References

[1] Zenchang Qin and Lao Wang (2023)，How to learn deep learning? Journal of Paper Writing, Vol. 3: 23: pp. 1-12.
[2] hanmy1021(2019),NLP.Github. https://github.com/hanmy1021/NLP
[3] Mingyu Han(2019),中文信息熵的计算, CSDN. https://blog.csdn.net/qq_37098526/article/details/88633403