Report of Deep Learning for Natural Langauge Processing

Jiayi Zhang
zy2303814@buaa.edu.cn

Abstract

This report investigates Zipf's law and calculates the entropy of Chinese using a corpus of 16 Chinese novels. The experiment verifies Zipf's law by plotting frequency-rank figures in both original and logarithmic coordinates. Additionally, it calculates the entropy of Chinese text based on word and character levels using Entropy calculation formula and *N-Gram language model*.

Introduction

Zipf's law [1] is an empirical law. In many texts in human languages, word frequencies approximately follow a Zipf distribution with exponent s close to 1: that is, the most common word occurs about n times the nth most common one.The best known instance of Zipf's law applies to the frequency table of words in a text or corpus of natural language:

$$word\ frequency \propto \frac{1}{word\ rank} \tag{1.1}$$

The actual rank-frequency plot of a natural language text deviates in some extent from the ideal Zipf distribution, especially at the two ends of the range. The deviations may depend on the language, on the topic of the text, on the author, on whether the text was translated from another language, and on the spelling rules used. Some deviation is inevitable because of sampling error.

At the low-frequency end, where the rank approaches N, the plot takes a staircase shape, because each word can occur only an integer number of times. This is also reflected in **Figure 1** of the later experiment.

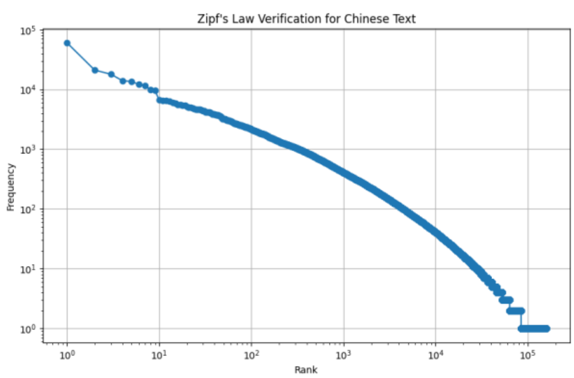


Figure 1: Word frequency statistics raw data

In 1992,Peter F. Brown[2] proposed a method for calculating English entropy. Suppose $X=\{...X_{-2},X_{-1},X_0,X_1,X_2...\}$ is a stationary stochastic process over a finite alphabet. Let P denote the probability distribution of X and let E_p denote expectations with respect to P. **The entropy of** X is defined by

$$H(X)\equiv H(P)\equiv -E_P\log P(X_0|X_{-1},X_{-2},...) \tag{1.2}$$

When P is not known, an upper bound to $H(P)$ can still be obtained from an approximation to P. Suppose that the stationary stochastic process M is a model for P. **The cross-entropy of** P **as measured by** M is defined by

$$H(P,M)=\lim_{n\to\infty}-\frac{1}{n}E_p\log M(X_1X_2...X_n) \tag{1.3}$$

The relationship between entropy $H(P)$ and cross-entropy $H(P,M)$ is as follows

$$H(P)\le H(P,M) \tag{1.4}$$

Then they proposed a language model *The Token Trigram Model* which captures the structure of English only through token trigram frequencies. They also take into account the spelling of English words and are case sensitive.

The token trigram model is a second-order Markov model that generates a token string $t_1t_2...t_n$ by generating each token t_i, in turn, given the two previous tokens t_{i-1} and t_{i-2}. Thus the probability of a string is

$$M_{token}(t_1t_2...t_n)=M_{token}(t_1t_2)\prod_{i=3}^{n}M_{token}(t_i|t_{i-2}t_{i-1}) \tag{1.5}$$

By combining the formula (1.5) and (1.3), the cross-entropy of as measured by is can be obtained

$$H(P,M)=\lim_{n\to\infty}-\frac{1}{n}E_p\log(M(t_1t_2)\prod_{i=3}^{n}M(t_i|t_{i-2}t_{i-1})) \tag{1.6}$$

wher $l_M(X_1X_2...X_n)$ is the number of bits in the encoding of the strin $X_1X_2...X_n$.

Methodology

There are models of my research. The first part verifies Zipf's Law through Chinese corpus. The second part calculates the average entropy of Chinese information.

M1: Verify Zipf's Law through Chinese corpus

Based on this principle, the experimental scheme for verifying Zipf's Law with Chinese corpus is designed as follows:

Step1: Access to Chinese corpus.

Step2: Process the text: including word segmentation, removal of stop words, get a vocabulary list.

Step3: Calculate word frequency: the frequency of each word in the processed text can be counted to:obtain a word frequency list.

Step4: Sort: Sort the word frequency list from highest to lowest.

Step5: Draw a chart: Using the sorted word frequency list, draw a chart with word rank as the horizontal axis and word frequency as the vertical axis. If Zipf's Law holds, you can get a pattern that approximates a straight line.

Step6: Fit the curve: Fit the data to see if it fits the mathematical model of Zipf's Law.

Step7: Analyze the results.

M2: Calculate the average information entropy of Chinese

In information theory, [3] the entropy of a random variable is the average level of "information", "surprise", or "uncertainty" inherent to the variable's possible outcomes. The entropy is

$$H(X) := -\sum_{x\in\chi} p(x)\log p(x) \tag{1.7}$$

Where $p(x_i)$ is the probability that the ith word or word appears in the text.

Unlike English words, since each Chinese character has its own meaning, the smallest unit that makes up a complete sentence is a word, and a sentence is usually a sequence of words and words with a complete meaning.

Suppose a sequence of words

$$S = W_1, W_2, ...W_K \tag{1.8}$$

The probability of occurrence of this sentence can be expressed as

$$P(S) = P(W_1, W_2, ...W_K) = P(W_K | W_1, W_2, ...W_{K-1}) \tag{1.9}$$

But in fact, according to the *Markov hypothesis*, the probability of occurrence of a random word is only related to a limited number of words or words that precede it. By using the *N-Gram language model*, we can simplify the calculation of the probability of sentence occurrence by introducing the *Markov hypothesis*.

Therefore, the probability formula for the occurrence of a fixed length sequence can be obtained as follows

$$P(w_1 w_2 ... w_n) \approx \prod_i P(w_i | w_{i-k} ... w_i) \tag{1.10}$$

When n takes different values, there are different models n-grams.

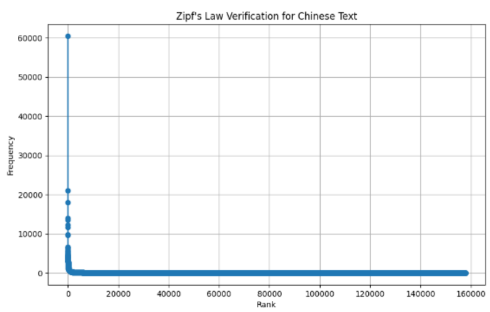
Experimental Studies

In this experiment, 16 novels in the ***jyxstxtqj_downcc.com*** document are read and divided into words, and the contents provided by ***cn_punctuation.txt*** and ***cn_stopwords.txt*** in the ***DLNLP2023-main*** document are used. By removing the stops and punctuation marks in the vocabulary, and removing punctuation marks such as *'=', '\n', '\n3000'*, and *Spaces* that are not in the range of processing, the word frequency in Chinese novels is obtained.

Table1 Top 20 high-frequency words

Rank	Word	Frequency	Rank	Word	Frequency
1	道	60460	11	武功	6499
2	说	21031	12	想	6464
3	便	17962	13	没	6377
4	中	13987	14	心中	6076
5	说道	13559	15	笑	5882
6	听	12172	16	师父	5581
7	见	11729	17	瞧	5576
8	韦小宝	9833	18	不知	5451
9	一个	9677	19	知道	5373
10	一声	6662	20	走	5166

After word frequency is counted, word frequency ranking of all words is obtained by sorting word frequency from high to bottom. The **top 20** words are shown in **Table 1**.



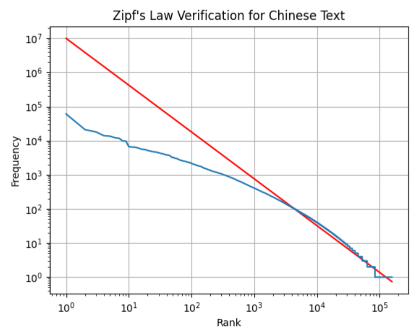


Figure 2: Original Word Frequency curve   Figure 3: Log Word Frequency curve

Then, linear fitting was performed on the data to obtain the linear change rule of word frequency. Word frequency curve and linear fitting curve were drawn with rank as horizontal axis and word frequency as vertical axis, which was drawn on the same picture, as shown in **Figure 2 and Figure 3**. It can be seen that there is an inverse relationship between word frequency and ranking, which proves that Zipf's Law is established.

In order to solve question 2.The segmentation methods for Chinese are based on words and based on characters. The text is preprocessed by modifying the code of question 1, according to different requirements of word segmentation, word segmentation is performed and word frequency is counted.

1.Divide the text into words:

After processing with 1-gram model, the results of the top 20 words with frequency are shown in **Table 2**, and the calculated average information entropy of **1 elements** is **12.16449983083691**.

After processing with 2-gram model, the results of the top 20 words with frequency are shown in **Table 3**, and the calculated average information entropy of **2 elements** is **6.946307379425751.**

After processing with 3-gram model, the results of the top 20 words with frequency are shown in **Table 4**, and the calculated average information entropy of **3 elements** is **2.30409619270501**.

Table2 Top 20 high-frequency words by using 1-gram model

Rank	Word	Frequency	Rank	Word	Frequency
1	的	115616	11	那	26875
2	了	104556	12	又	23831
3	他	64718	13	她	22599
4	是	64466	14	不	22088
5	道	58625	15	得	22016
6	我	57483	16	说	20862
7	你	56681	17	去	18702
8	在	43698	18	便	18040
9	也	32608	19	有	17432
10	这	32207	20	将	15694
Average information entropy			12.16449983083691		

Table3 Top 20 high-frequency words by using 2-gram model

Rank	Word	Frequency	Rank	Word	Frequency
1	道你	5738	11	只听	2970
2	叫道	5009	12	又是	2709
3	道我	4953	13	了我	2560
4	笑道	4271	14	你的	2461
5	听得	4203	15	韦小宝道	2360
6	都是	3906	16	我的	2303
7	了他	3638	17	道是	2239
8	他的	3497	18	见他	2181
9	也是	3201	19	那是	2129
10	的一声	3102	20	了你	2098
Average information entropy			6.94630737942575		

Table4 Top 20 high-frequency words by using 3-gram model

Rank	Word	Frequency	Rank	Word	Frequency
1	只听得	1611	11	跟你说	441
2	忽听得	1138	12	道是啊	431
3	站起身来	733	13	笑道我	396
4	哼了一声	573	14	叹了口气	375
5	笑道你	566	15	道是是	374
6	吃了一惊	535	16	韦小宝笑道	355
7	点了点头	503	17	的一声响	350
8	啊的一声	481	18	过了一会	348
9	说到这里	477	19	便在此时	346
10	了他的	454	20	但听得	341
Average information entropy			2.304096192705019		

2.Divide the text into characters:

After processing with 1-gram model, the results of the top 20 words with frequency are shown in **Table 5**, and the calculated average information entropy of **1 elements** is **9.536612497753614.**

After processing with 2-gram model, the results of the top 20 words with frequency are shown in **Table 6**, and the calculated average information entropy of **2 elements** is **6.716221966189054**.

After processing with 3-gram model, the results of the top 20 words with frequency are shown in **Table 7**, and the calculated average information entropy of **3 elements** is **3.9388582389958655.**

Table5 Top 20 high-frequency words by using 1-gram model

Rank	Word	Frequency	Rank	Word	Frequency
1	一	139423	11	来	64161
2	不	134170	12	你	61633
3	的	121683	13	大	59729
4	是	112725	14	在	52364
5	了	111944	15	上	50751
6	道	111066	16	中	48547
7	人	84314	17	得	48084
8	他	73581	18	之	48068
9	这	69005	19	说	47853
10	我	67001	20	下	45273
Average information entropy			9.536612497753614		

Table6 Top 20 high-frequency words by using 2-gram model

Rank	Word	Frequency	Rank	Word	Frequency
1	说道	13528	11	不是	8031
2	了一	12180	12	什么	7891
3	一个	10572	13	一声	7553
4	自己	10319	14	不知	7271
5	道你	10262	15	咱们	6829
6	小宝	9942	16	的一	6779
7	韦小	9856	17	令狐	6707
8	也不	9306	18	这一	6654
9	道我	8473	19	武功	6524
10	笑道	8140	20	心中	6409
Average information entropy			6.716221966189054		

Table7 Top 20 high-frequency words by using 3-gram model

Rank	Word	Frequency	Rank	Word	Frequency
1	韦小宝	9803	11	了出来	1685
2	令狐冲	5889	12	只听得	1673
3	张无忌	4645	13	在地下	1457
4	的一声	3478	14	欧阳锋	1411
5	袁承志	3037	15	低声道	1408
6	小宝道	2417	16	在这里	1407
7	陈家洛	2116	17	了起来	1380

8	小龙女	2081	18	起身来	1293
9	石破天	1818	19	有什么	1266
10	不由得	1803	20	洪七公	1236
Average information entropy			3.9388582389958655		

Conclusions

By studying a corpus containing 16 Chinese novels, this paper verifies the Zipf law of Chinese, and calculates the information entropy of words and words in the text. It is proved that there is an inverse relationship between the frequency and ranking of words in Chinese corpus, which accords with Zipf's law. When calculating Chinese information entropy, the larger the word length, the smaller the Chinese information entropy.

Because of the different segmentation methods of Chinese characters, the statistical average information entropy of Chinese text is different. It can be found that for longer elements, the information entropy of character is lower, but the information entropy of 1 element is higher than that of word. This is because the character segmentation method can distinguish the text more carefully, but it also easily leads to the inaccuracy of the meaning of a single element. At the same time, we can find that the results of 2-garm and 3-gram models divided into characters are lower, probably because the two individual characters may not have practical meaning, but the number of combinations is more, so the average information entropy is lower.

References

[1] Wikipedia. Zipf's law[EB/OL].https://en.wikipedia.org/wiki/Zipf%27s_law.
[2] Peter F. Brown, Vincent J. Della Pietra, Robert L. Mercer, Stephen A. Della Pietra, and Jennifer C. Lai. 1992. An estimate of an upper bound for the entropy of English. Comput. Linguist. 18, 1 (March 1992), 31–40.
[3] Wikipedia. Entropy [EB/OL].https://en.wikipedia.org/wiki/Entropy_(information_theory).