Natural Language Processing

Empirical Laws: Zipf, Mandelbrot, Heaps

Ramaseshan Ramachandran

Empirical Laws

ZIPF'S LAW

This empirical law models the frequency distribution of words across various languages. Zipf's law [1] states that, in a given corpus, the frequency of any word is inversely proportional to its rank in the term-frequency table.

$$f(r) \propto \frac{1}{r^{\alpha}} \tag{1}$$

where $\alpha \approx 1$, r is the *frequency rank* of a word and $f(r)$ is the frequency in the corpus.

Distribution of terms/words

This empirical law models the frequency distribution of words in languages. This distribution is observed across several languages when a large corpus is used.

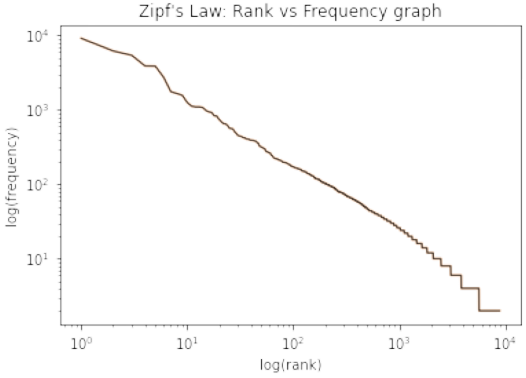
ZIPF'S LAW - FREQUENCY VS RANK

Output generated from Indian budget speeches using the bash script [8] with minimal preprocessing

Word	Frequency	Rank
the	59042	1
of	46110	2
to	35873	3
and	24661	4
in	22916	5
for	14426	6
a	14366	7
is	11155	8
be	9760	9
I	9070	10
...	...	...

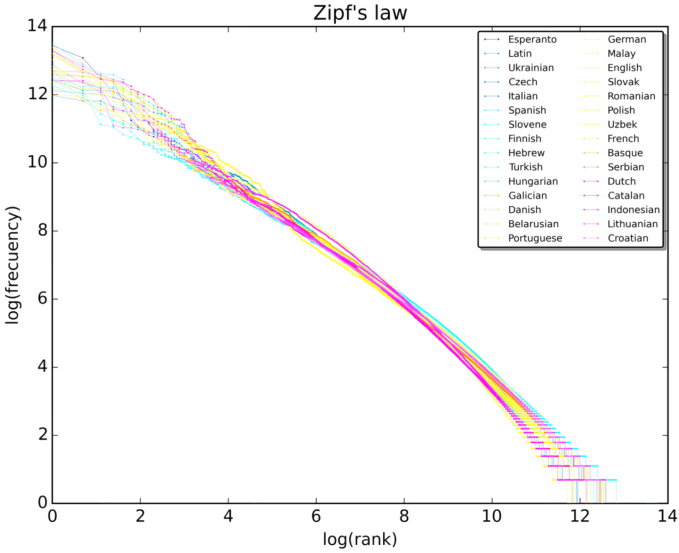
Word	Frequency	Rank
agricultural	479	233
important	478	234
policy	477	235
reduced	476	236
balance	476	237
...	...	...
(SFB)	1	92363
(SFAC),1	1	92364
(SFAC)	1	92365
(SEZs).,1	1	92366
...	...	...

ZIPF'S LAW I



$$f \propto \frac{1}{r^{\alpha}}$$

A plot of the rank versus frequency in a log-log scale.



$$f \propto \frac{1}{r^{\alpha}}$$

A plot of the rank versus frequency for the first 10 million words in 30 Wikipedia (dumps from October 2015) in a log-log scale.

Source: Zipf's law -Wikipedia

Web Traffic Patterns

- Google.com dominates with billions of daily visits
- The top 20% of websites account for roughly 80% of total traffic
- Follows the Pareto principle

Caching Strategies

- CDNs and single-page apps exploit Zipf's law to optimize content delivery
- Small number of web pages receive the majority of requests
- Caching popular pages to improve response times

Search Algorithms

- Most searches focus on a limited set of popular topics

Text Analytics

- Helps in identifying stopwords
- Helps in identifying the important words to focus on
- Reduces the sample space
- Motivates frequency-based subsampling and vocabulary truncation in word-embedding and language models (e.g., word2vec subsampling, BPE vocabularies)

BASH SCRIPT FOR COMPUTING RANK AND FREQUENCY

```
# 10. Concatenate all files in TXT directory
# 11. Remove form feed character (^L)
# 12. Split into words (replace spaces with newlines)
# 13. Sort in reverse
# 14. Count unique words
# 15. Sort by frequency, descending
# 16. Add word, frequency rank
# 17. output as word_freq_rank.csv

cat xx.txt | \
tr -d '\f' | \
tr -s ' ' '\n' | \
sort -r | \
uniq -c | \
sort -nr | \
awk '{print $2 "," $1 "," NR }' \
> word_freq_rank.csv
```

MANDELBROT APPROXIMATION

Mandelbrot derived a more general law that fits the observed frequency distribution more closely by adding an offset to the rank

$$f(r) \propto \frac{1}{(r+\beta)^{\alpha}} \tag{2}$$

The offset β mainly improves the fit for the most frequent words (small r), where plain Zipf overestimates the frequency.

HERDAN OR HEAPS' LAW

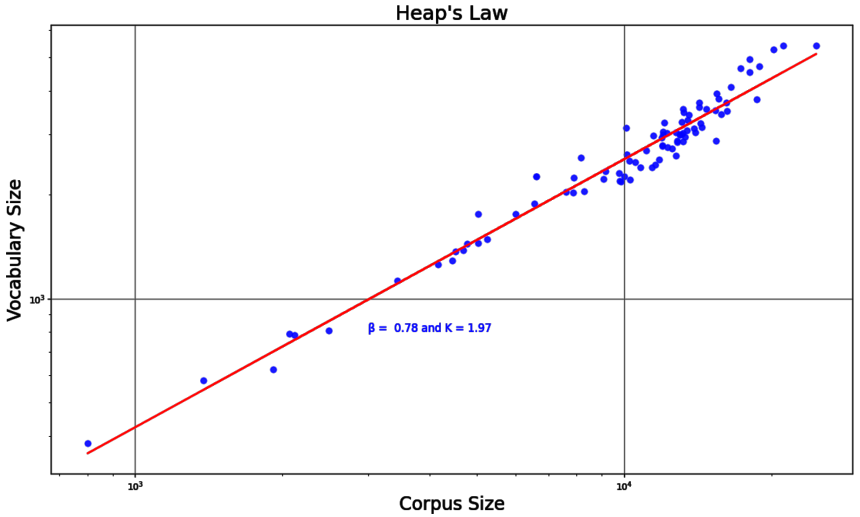
This empirical law [2] describes how vocabulary size grows as a sublinear function of corpus size.
It estimates the number of unique terms M in a corpus containing T tokens:

$$\begin{aligned} M &\propto T^{\beta} \\ &= kT^{\beta} \end{aligned} \tag{3}$$

where typically $10 \leq k \leq 100$ and $0.4 \leq \beta \leq 0.6$ for English text

- As a corpus grows, new words keep appearing, but at a diminishing rate
- Uses: vocabulary-size estimation, index-memory planning, and system scaling
- For LLMs: motivates fixed-size *subword* vocabularies (BPE/SentencePiece), which sidestep unbounded vocabulary growth

HEAPS' LAW


Heap's Law
Vocabulary Size
Corpus Size
β = 0.78 and K = 1.97
10³
10³
10⁴

PREPROCESSING USING EMPIRICAL LAWS

- Raw text typically exhibits the highest k values (around 44–50)
- Stop-word removal reduces the token count sharply, but the vocabulary only slightly
- Stemming and lemmatization can shrink the vocabulary by 50–60%
- Aggressive preprocessing guided by Zipf's law (dropping rare, low-rank words) can reduce the vocabulary by 70–85%

IMPLICATIONS FOR NLP

Predictive Power

Helps in predicting language characteristics.

Corpus Analysis

Essential for understanding large text datasets.

Model Development

Guides vocabulary sizing and smoothing choices.

REFERENCES I

[1] George Kingsley Zipf. "The psychology of language". In: *Encyclopedia of psychology*. Philosophical Library, 1946, pp. 332–341.

[2] Kim Chol-jun. *Proper Interpretation of Heaps' and Zipf's Laws*. 2025. arXiv: 2305.15413 [physics.soc-ph]. URL: https://arxiv.org/abs/2305.15413.