Modern Byte Pair Tokenizers are Zipfian

Jack Hanke Daniel Plotkin
Nicole Birova David Demeter
Northwestern University

Abstract

A majority of large language models ingest word fragments produced by a data compression algorithm known as byte pair encoding. This algorithm groups high-frequency letter pairings in natural language into individual units called *tokens*. A natural question is whether the tokens produced by this pairing process deviate significantly from the source language's frequency distribution. Zipf famously showed that many natural language's frequency distribution follows a power law, commonly known as Zipf's Law. We examine two modern tokenizer's adherence to Zipf's law at the token level. We demonstrate that these tokenizers are Zipfian on two corpuses, and speculate as to why this is.

1 Introduction

George Zipf in his hallmark *The Psycho-Biology of Language* (Zipf, 1935) introduced the remarkable trend that the word frequencies in many human languages exhibit the same power law distribution. Specifically, Zipf said the number of times a word appears in a corpus is proportional to the word's frequency rank. Mathematically, Zipf's laws states

$$\text{word frequency} \propto \frac{1}{\text{word rank}}. \tag{1}$$

A more descriptive model of word frequency that is more commonly referenced in linguistics literature is the Zipf-Mandelbrot distribution

$$\text{word frequency} \propto \frac{1}{(\text{word rank} + b)^a}. \tag{2}$$

where a, b are fitted parameters. We say a distribution that follows the trend in Equation 2 with $a \approx 1$ is *Zipf distributed*, or simply *Zipfian*.

The accuracy of the trend in Equation 2 has been examined in 10^8 English words in (Ferrer-i Cancho and Solé, 2000), over 50 languages in (Yu et al., 2018), and written-versus-spoken corpuses in (Lin et al., 2015). Each of these studies demonstrate that Zipf's law is generally exhibitted for common and somewhat-uncommon words, but rare words (words with high rank) appear less frequently than predicted.[1] This deviation is shown to be statistically significant, and appears as two trendlines in the log-log plot. In English corpuses, this transition is found around the 10^5-*th* ranked word. The authors further explore the linguistic relevance and universality of these multiple regimes of words.

Nearly a century after Zipf's discovery, large language models (LLMs) generate text comparable to human communication. However, unlike humans, LLMs digest text using a fixed vocabulary of word fragments called *tokens*. The mapping between natural language and tokens is most commonly computed using the *byte pair encoding (BPE) algorithm* (Gage, 1994) over some large training corpus. Byte pair encoding identifies the most frequent pair of letters, and creates a new token in the "token vocabulary" to replace that pair. Iterating this procedure until some fixed vocabulary size is reached creates the vocabulary of tokens. Finally, any sequence that does not map to one of the derived tokens is labelled with the “unknown” token.

When considering these ideas in tandem, a natural question arises: given a corpus that appears Zipfian, how does the tokenization process affect this trend?

2 Methods

To explore this question, we examine two modern tokenizers. We choose the tokenizer for the RoBERTa language model (Liu et al., 2019), as the training data is publicly known. This allows us to conduct frequency analyses on corpuses that are on and off-distribution for the tokenizer. We also

[1] Some of these studies also find that extremely common words appear slighly more commonly than the Zipfian prediction.

choose to compare our results with the tokenizer for the GPT-4 language model (OpenAI et al., 2024) as a representative of industry-scale byte pair encoding.

We choose two corpuses based on RoBERTa's training data. We choose the 4.4GB bookscorpus dataset (Zhu et al., 2015), which is within RoBERTa's training data, and the 5.6GB MiniPile dataset (Kaddour, 2023), which is not. Note that it is likely that the GPT-4 tokenizer was trained (in-part) on both of these corpuses.

We compute the word and token frequency for both tokenizers on each corpus, where we define a word as anything separated by a space. We then compute the data's Kolmogorov-Smirnov goodness-of-fit statistic for the fitted Zipf distribution, and compare to that for a fitted exponential and log-normal distribution.

We do not remove any words or tokens from consideration unless otherwise stated, as to a language model all non-padding tokens are treated the same in training.

3 Results

We find that token vocabularies are remarkably Zipfian, excluding extremely rare tokens. This holds true for both tokenizers and both corpuses studied, including the out-of-distribution text. In the log-log plots in Figure 1, we find a clear linear trend over a large number of words and tokens. We only see deviations from the Zipfian trend for the final few tokens in the RoBERTa tokenizer for both corpuses and the GPT-4 tokenizer for the MiniPile corpus. However, the GPT-4 tokenizer trend on the bookscorpus seems to deviate more dramatically, similar to the multiple regime studies mentioned in Section **??**.

We also report the lowest rank words and tokens for each tokenizer and corpus to identify trends across models and corpuses. We report these words and tokens both including and excluding control tokens and punctuation, which are summarized in Table 1 and Table 2 respectively.

Finally, we report the Kolmogorov-Smirnov test statistics for fitted Zipf-Mandelbrot, Exponential, and log-normal distributions in Table **??**.

4 Analysis

The BPE algorithm builds a finite vocabulary by directly referencing the frequency of subwords and maps everything not captured to a "catch-all" un-

Rank	**Word**		**RoBERTa**		**GPT-4**	
1	.	the	<s>	␣	“	,
2	,	of	\n	\n	i	␣the
3	the	and	<\s>	.	he	.

Table 1: Summary of most common words & tokens for each corpus, where the left subcolumn is bookscorpus and the right subcolumn MiniPile, including control tokens and punctuation

Rank	**Word**		**RoBERTa**		**GPT-4**	
1	the	the	the	the	i	the
2	to	of	to	of	he	of
3	i	and	and	and	she	and

Table 2: Summary of most common words & tokens for each corpus, where the left subcolumn is bookscorpus and the right subcolumn MiniPile, *not* including control tokens and punctuation. All of these words are considered *function words* in linguistics literature.

Distribution	**Words**		**RoBERTa**		**GPT-4**	
Zipf						
Exponential	0.87	0.81	0.58	0.48	0.72	0.56
log-normal	0.38	0.45	0.03	0.05	0.40	0.28

Table 3: Kolmogorov-Smirnov test statistics for distribution fit, where the left subcolumn is bookscorpus and the right subcolumn MiniPile

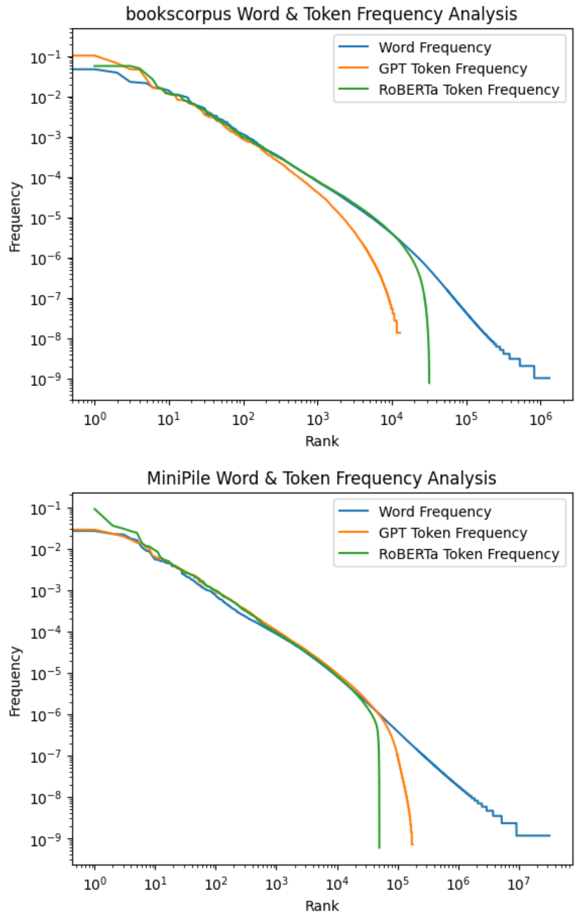

Figure 1: Log-log plots of word & token rank vs word & token frequency

known token. Despite this, these generated vocabularies are still Zipfian up to the rarest few tokens. Moreover, the tokenizers this algorithm produces creates Zipfian distributed tokens outside of the corpus it is trained on. When considering all possible tokens, the most common tokens tend to be punctuation, newline characters, and control tokens. Excluding punctuation and special tokens, we recover the known most common English words.

5 Conclusions & Future Work

Noticeable theoretical work has been done in service of explaining Zipf's law for language (Li, 1992). Most significant was Belevitch's *On the statistical laws of linguistic distributions*(Belevitch, 1959), in which the author shows the first order approximation of the rank ordering of many statistical distributions are all Zipfian. This indicates that Zipf's law may be due to the rank ordering of words more than the underlying formation of language, which is what Zipf was originally interested in. Can Belevitch's proof be shown to be invariant under the byte pair encoding transformation?

References

V. Belevitch. 1959. On the statistical laws of linguistic distributions. *Annales de la Société Scientifique de Bruxelles*, 73:310–326.

Ramon Ferrer-i Cancho and Ricard Solé. 2000. Two regimes in the frequency of words and the origins of complex lexicons: Zipf's law revisited. *Santa Fe Institute, Working Papers*.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2):23–38.

Jean Kaddour. 2023. The minipile challenge for data-efficient language models. *Preprint*, arXiv:2304.08442.

W. Li. 1992. Random texts exhibit zipf's-law-like word frequency distribution. *IEEE Transactions on Information Theory*, 38(6):1842–1845.

Ruokuang Lin, Qianli D. Y. Ma, and Chunhua Bian. 2015. Scaling laws in human speech, decreasing emergence of new words and a generalized model. *Preprint*, arXiv:1412.4846.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Shuiyuan Yu, Chunshan Xu, and Haitao Liu. 2018. Zipf's law in 50 languages: its structural pattern, linguistic interpretation, and cognitive motivation. *Preprint*, arXiv:1807.01855.

Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *Preprint*, arXiv:1506.06724.

G. K. Zipf. 1935. *The Psycho-Biology of Language*. Houghton Mifflin, Boston.