Identification of Protein Language Units using Protein Oriented Tokenization Methods and Language Models

PhD Progress

Burak Suyunu - 16.05.2024

LifeLU

HORIZON

Call: ERC-2022-COG
(Call for Proposals for ERC Consolidator Grant)

Topic: ERC-2022-COG

Type of Action: HORIZON-ERC
(HORIZON ERC Grants)

Proposal number: 101089287

Proposal acronym: LifeLU

Arzucan Özgür, Gökçe Uludoğan, Enes Taylan

Introduction

- Proteins play a vital role in the maintenance and regulation of life.
- Sequences of amino acids (2D) —> determining structure and function of protein.
- Proteins as written language —> Language of Life
  - Characters —> Amino acids
  - Sentences —> Protein sequences
- No one can understand, no vocabulary, no words.

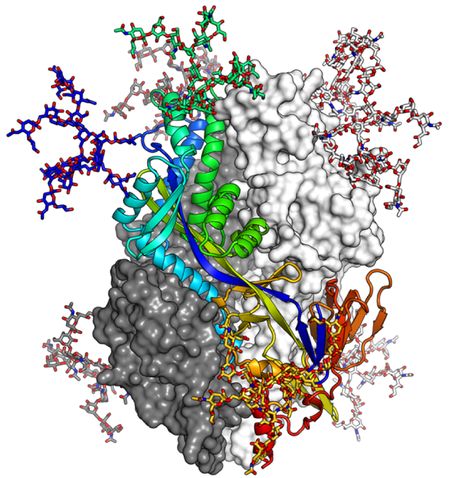


Image courtesy of Nat Commun 8, 1528 (2017)

Motivation

- Complexity and variability of protein sequences necessitate efficient representation and segmentation strategies.
- Using tokenization methods from NLP which are not necessarily the best for the protein sequences
- Tokenization methods are examined under downstream tasks but not from linguistic perspective.

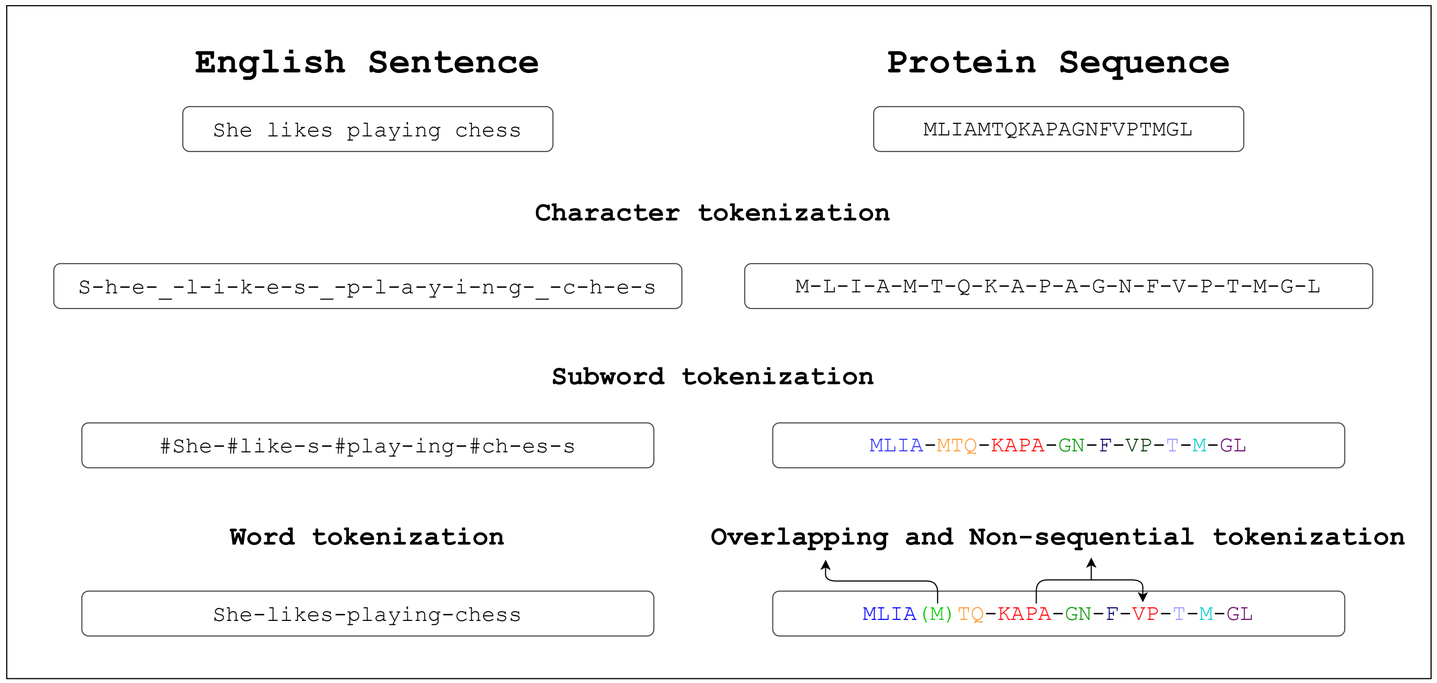
Objective

- Comprehensively evaluate BPE, WordPiece, and SentencePiece tokenization methods on protein sequences using the UniRef50 dataset.
- Our evaluation focuses on key statistical properties and explores how these tokenizers adhere to fundamental linguistic laws.
- Systematically comparing these subword tokenization methods across various vocabulary sizes (400, 800, 1600, 3200, and 6400), statistical metrics, and linguistic laws.
- Uncover insights into their strengths, weaknesses, and applicability to protein sequences.

Background

Tokenization

- Tokenization involves dividing a piece of text into smaller units, such as words, phrases, symbols, or other meaningful elements, called tokens.



Background

Tokenization (why)

- Still at the level of “molecular language processing” rather than “molecular language understanding”.
  - **Incorrect basic assumptions** about the fundamental units of meaning in the language of life - no clearly defined word boundaries.
  - **Single amino acids:** Only 20 common naturally occurring aa —> too limited.
  - **Windows of amino acids:** Allows overlapping but not for detecting word boundaries.
  - **Protein domains:** Not ideal granularity, do not cover the entire protein sequences.

Background

Subword Tokenization

- Subword tokenization methods are based on the idea that commonly used words should not be broken down into smaller subwords, while infrequent words should be divided into meaningful subparts.
- Subword tokenization enables the model to process new words by breaking them down into familiar subwords.
- The most well-known algorithms for subword tokenization are Byte Pair Encoding (BPE), WordPiece, Unigram, and SentencePiece.
    - **BPE:** Initial vocabulary of symbols, merge two most frequent pair of symbols until desired vocabulary size.
    - **WordPiece:** Similar to BPE, likelihood instead of frequency.
    - **Unigram:** Big initial vocabulary, discard tokens according to loss and unigram language model
    - **SentencePiece:** More of an implementation of BPE and Unigram. Can work on text without space.

Literature Review

- In bioinformatics, many studies evaluate tokenization methods within protein language models (pLMs) using various downstream tasks.
- While these approaches help identify high-performing methods, they fail to assess whether the chosen tokenizers are well-suited for encoding protein sequences in language models.
- Our study is one of the first to focus specifically on understanding how effective NLP tokenizers are for protein language models.

Literature Review

Bioinformatics

- **PETA: Evaluating the Impact of Protein Transfer Learning with Sub-word Tokenization on Downstream Applications (Tan et al., 2023)**
  - Tan et al. assess how protein language models perform across various downstream tasks with different vocabulary sizes and tokenization methods.
  - Comparing per-amino acid, BPE, and Unigram methods (with vocabulary sizes ranging from 50 to 3200), the study shows that vocabulary size significantly affects protein representation.
  - Larger vocabularies often worsen optimization in structure prediction datasets, with vocabulary sizes above 800 generally degrading performance.
- **Effect of Tokenization on Transformers for Biological Sequences (Dotan et al., 2024)**
  - Similarly, Dotan et al. evaluate the impact of different tokenization methods and vocabulary sizes on PLM performance across downstream tasks.
  - They compare per-amino acid, BPE, WordPiece, and Unigram methods (with vocabulary sizes ranging from 100 to 3200), demonstrating that advanced tokenization strategies can substantially reduce sequence lengths and improve performance.
  - They suggest that tasks involving proteins with similar domains may benefit more from task-specific tokenizers.

Literature Review

Bioinformatics

- **Protein language models are biased by unequal sequence sampling across the tree of life (Ding & Steinhardt, 2024)**
  - Ding and Steinhardt identify and quantify a species bias in pLM likelihoods due to uneven sequence sampling, revealing that certain species are systematically favored, resulting in higher likelihoods independent of the protein sequence itself.
  - By calculating an Elo rating for each species based on pre-trained pLM likelihoods, they show that this bias reduces thermostability and salt tolerance for protein designs starting from under-represented species.
- **Protein language models meet reduced amino acid alphabets (Ieremie et al., 2024)**
  - Ieremie et al. demonstrate that protein language models trained on reduced amino acid alphabets fail to capture critical evolutionary details and cannot distinguish between mutations within amino acid clusters, which distorts evolutionary information and degrades model performance.
- **Evaluating Protein Transfer Learning with TAPE (Rao et al., 2019)**
  - Rao et al. set a benchmark for future studies by introducing five downstream tasks for evaluating protein language models, despite not comparing different tokenizers and using outdated models.

Literature Review

NLP

- **Language Model Tokenizers Introduce Unfairness Between Languages (Petrov et al., 2023)**
  - Petrov et al. reveal that tokenization disparities across languages can lead to significant unfairness, advocating for multilingually fair subword tokenizers to ensure similar encoded lengths for equivalent content in different languages.
- **Tokenizer Choice For LLM Training: Negligible or Crucial? (Ali et al., 2024)**
  - Ali et al. highlight that tokenizer choice critically affects model performance and training costs, recommending balanced tokenizers across languages to maintain consistent performance metrics and avoid degradation in multilingual models.
- **Languages Through the Looking Glass of BPE Compression (Gutierrez-Vasques et al., 2023)**
  - Gutierrez-Vasques et al. use subword productivity and idiosyncrasy to show that BPE compression properties can vary with a language's morphological type, thus distinguishing different linguistic types through subword analysis

Literature Review

NLP

- **Incorporating Context into Subword Vocabularies (Yehezkel & Pinter, 2023)**
  - Yehezkel and Pinter develop the SAGE tokenizer, which, unlike traditional subword tokenizers, incorporates contextual information during vocabulary creation, resulting in more context-aware subwords that outperform others like BPE in various linguistic metrics.
- **Joint Optimization of Tokenization and Downstream Model (Hiraoka et al., 2021)**
  - Hiraoka et al. propose a method for jointly optimizing a tokenizer and a downstream model using loss values from the model, which enhances performance across various NLP tasks and can be applied as a post-processing step to refine existing models.

Experiment Setup

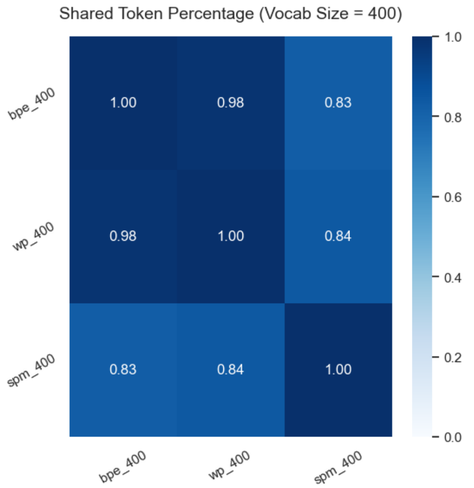
- **Dataset**: UniRef50 - 50 Million Protein Sequences
  - The UniRef50 dataset is a subset of UniRef (Universal Protein Resource) and contains representative sequences that are at least 50% identical to any other sequence in the cluster.
  - https://huggingface.co/datasets/agemagician/uniref50
  - Subword tokenizers were trained on randomly sampled 15 million sequences of the data's train split.
  - Experiments are applied to the combination of validation and test splits (11957 sequences).
  - We discarded 14 sequences from the test set that are longer than 3k in length.
- **Methods:** BPE (HuggingFace), WordPiece (HuggingFace), and SentencePiece-Unigram (Google)
- **Vocabulary Sizes:** 400, 800, 1600, 3200, and 6400

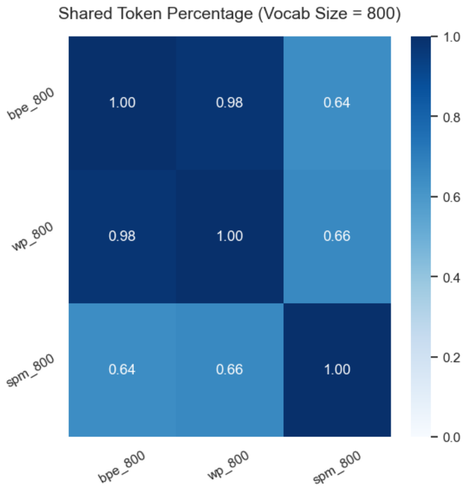
Evaluation Metrics

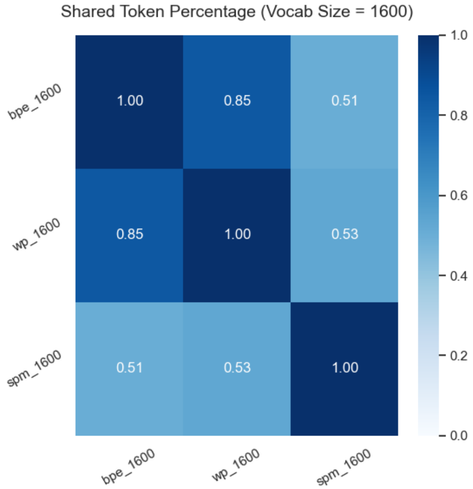
- **Shared token percentages:** The percentage of tokens shared between vocabularies of different tokenizers, offering insight into segmentation consistency.
- **Token length distribution:** Distribution of token lengths in the vocabulary, reflecting the granularity of the segmentation.
- **Fertility:** The average number of tokens required to represent a protein sequence, indicating encoding efficiency.
- **Contextual Exponence:** The diversity of neighboring tokens each token encounters, shedding light on semantic relationships.
- **Zipf's law:** The inverse proportionality between token frequency and rank, indicating linguistic regularity.
- **Brevity law:** The tendency of frequently used tokens to be shorter.
- **Heaps' law:** The growth of vocabulary size with dataset size, but at a decreasing rate.
- **Menzerath's law:** The inverse relationship between a protein sequence's length and its tokens' length.

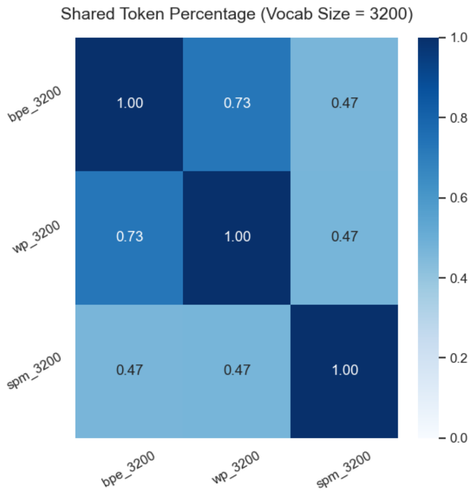
Experiments

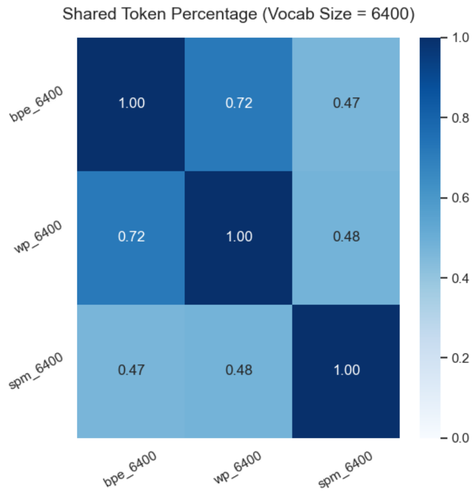
Shared Token Percentages





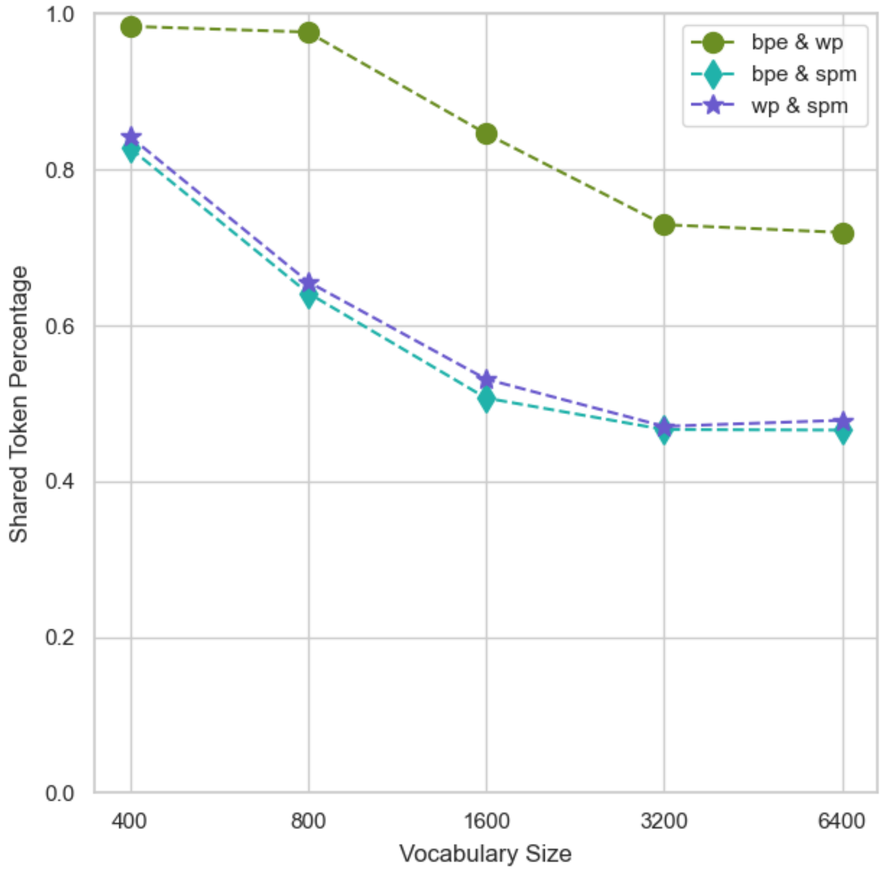






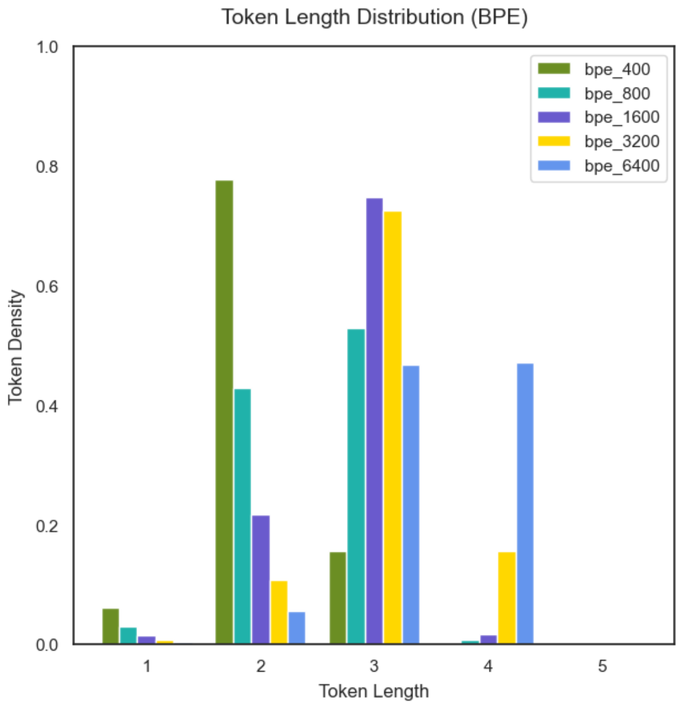
Experiments

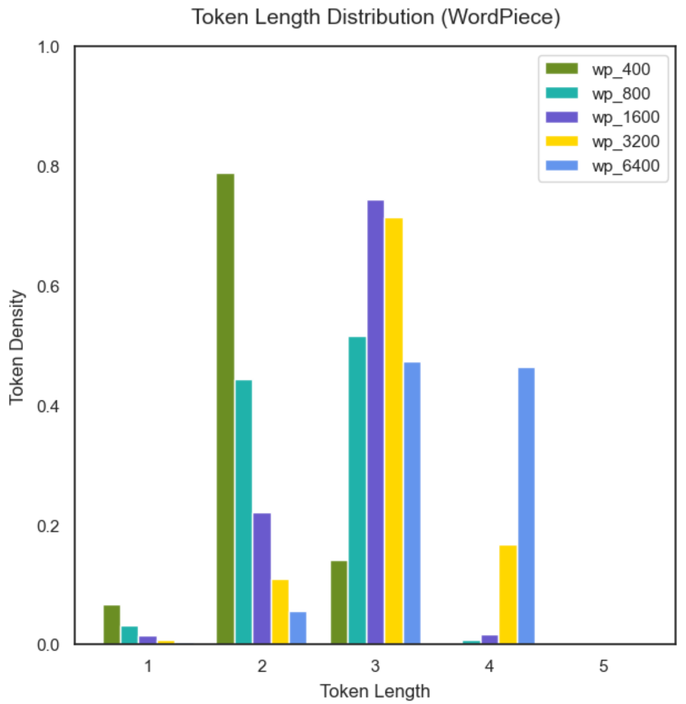
Shared Token Percentages

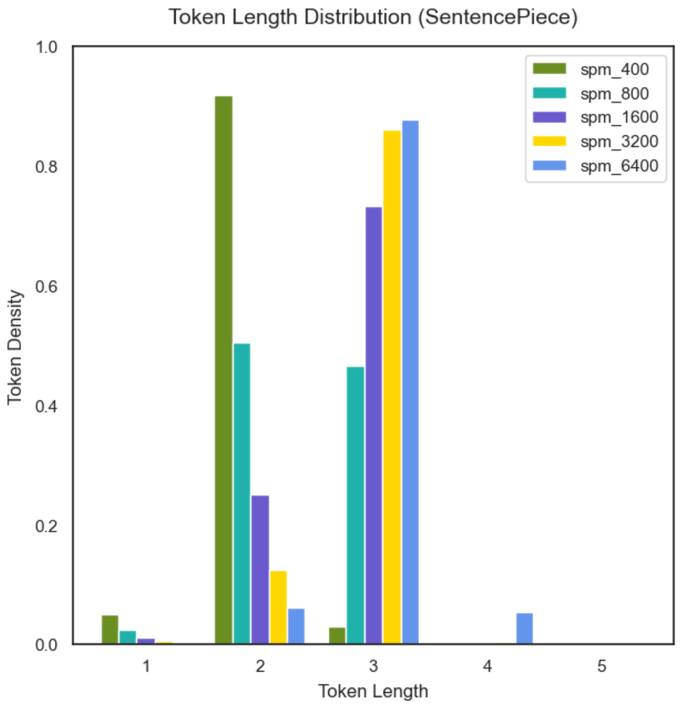


Experiments

Token Length Distribution

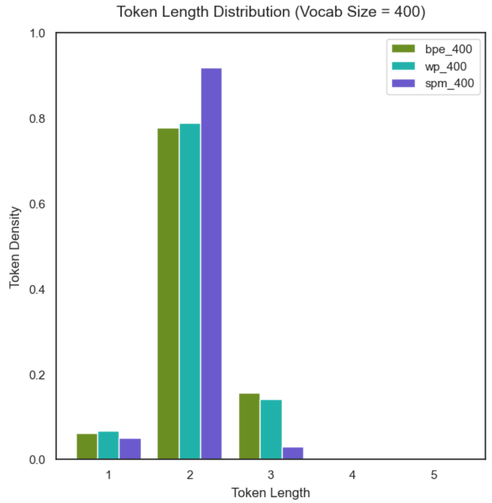


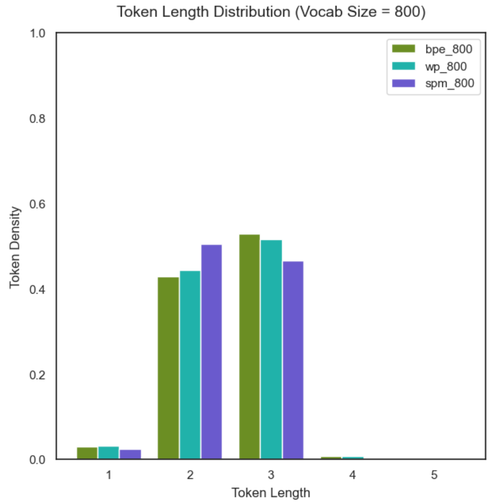


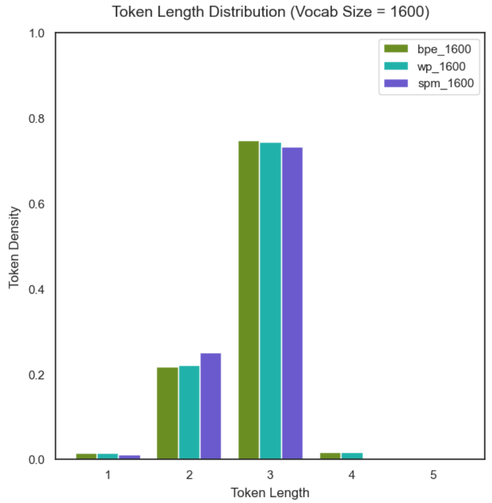


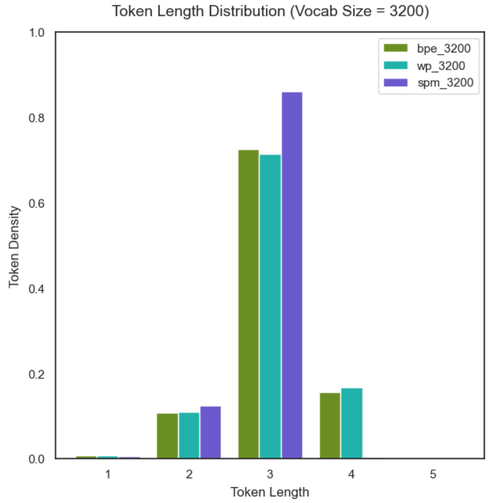
Experiments

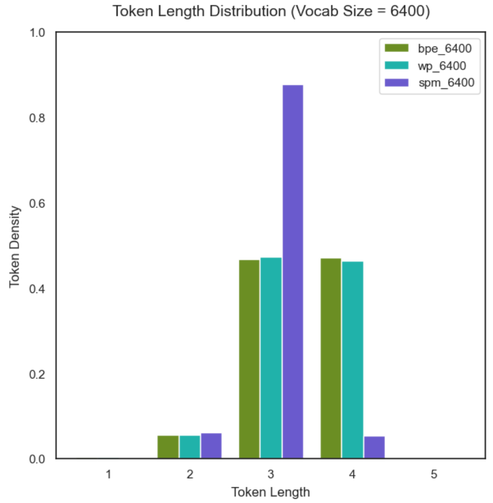
Token Length Distribution





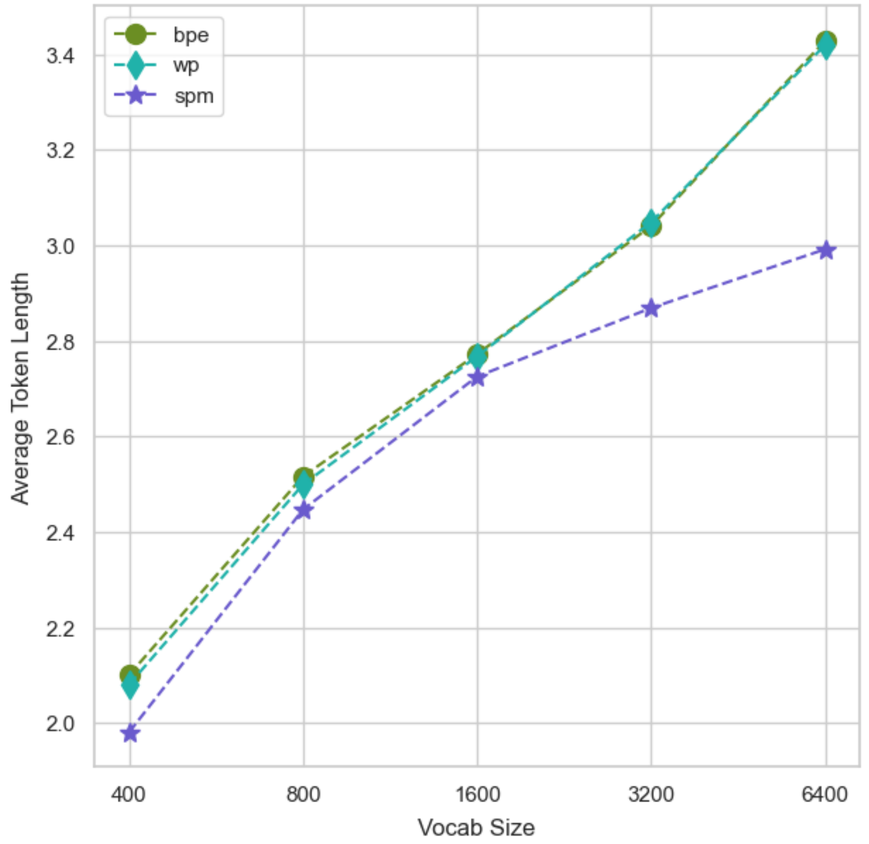






Experiments

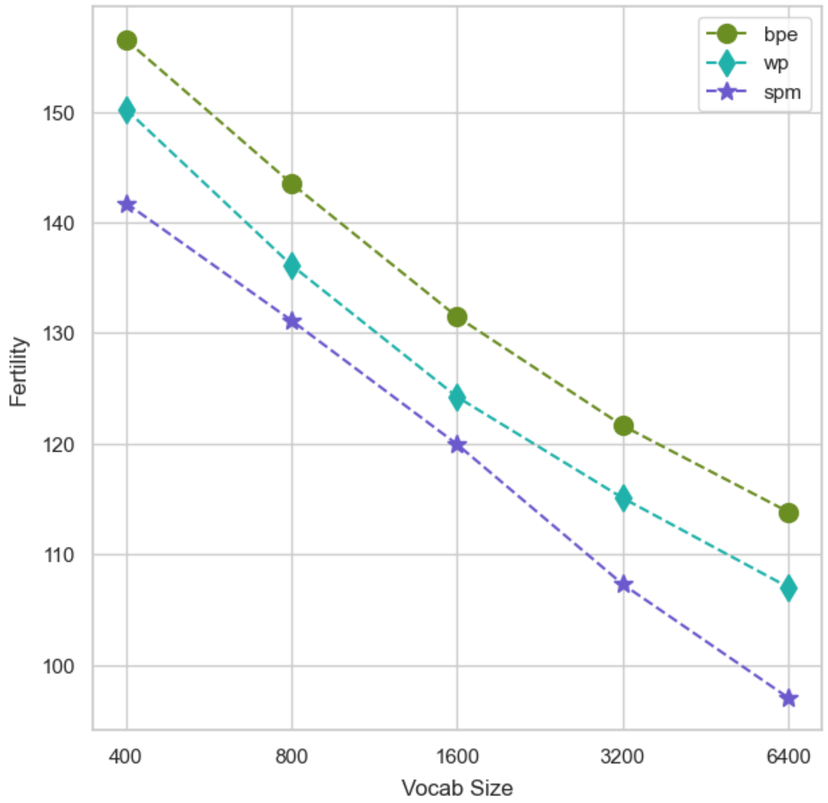
Token Length Distribution



Experiments

Fertility

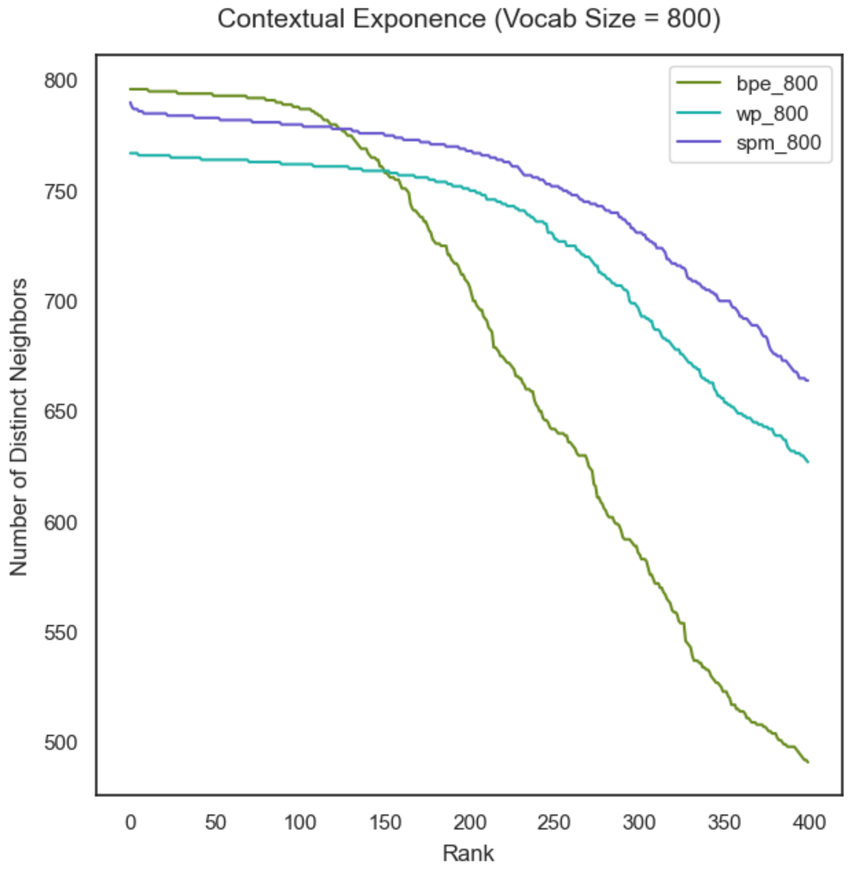
The average number of tokens required to represent a protein sequence, indicating encoding efficiency.



Experiments

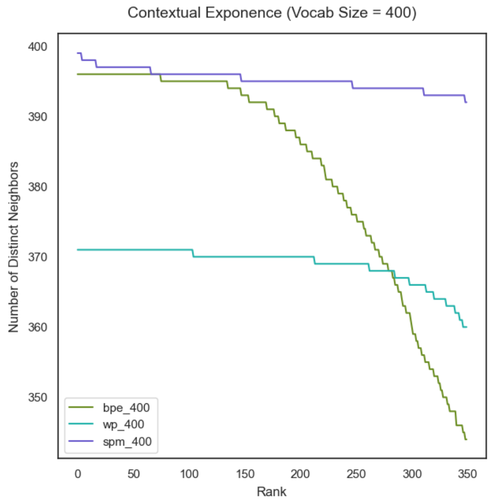
Contextual Exponence

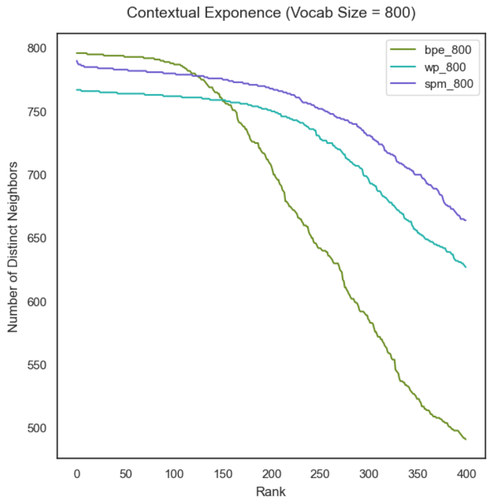
The diversity of neighboring tokens each token encounters in a 5-width window, shedding light on semantic relationships.

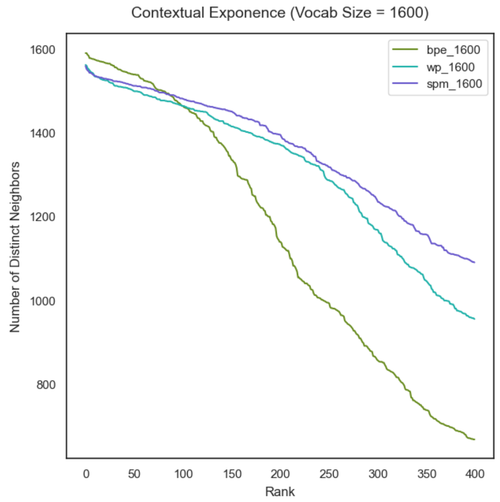


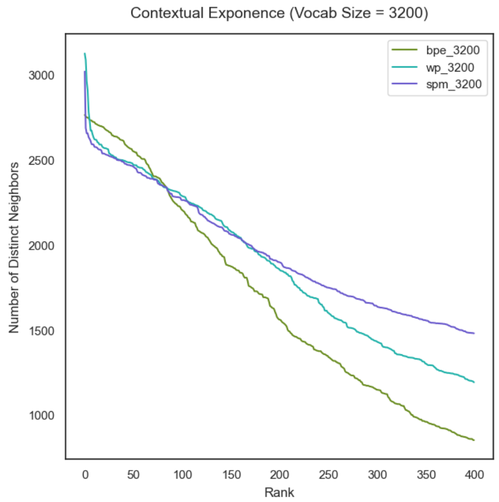
Experiments

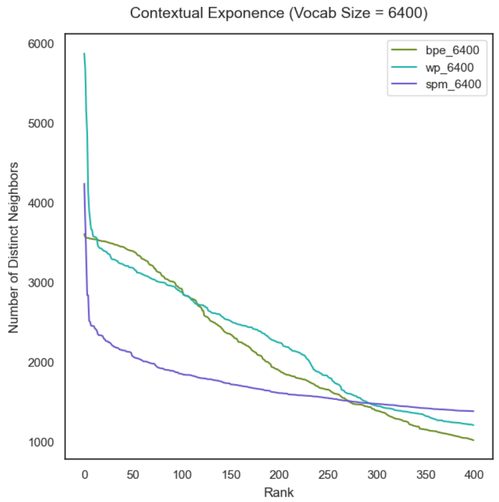
Contextual Exponence







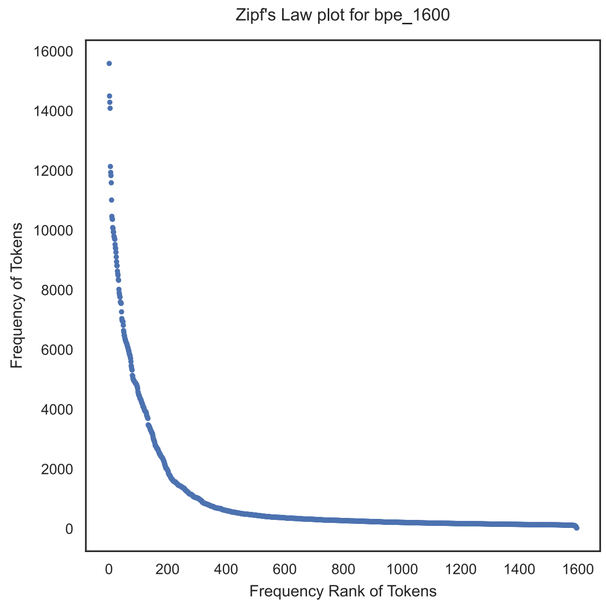


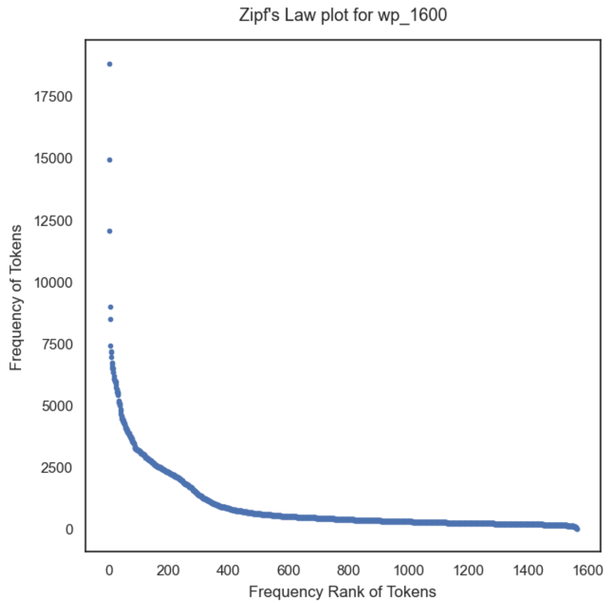


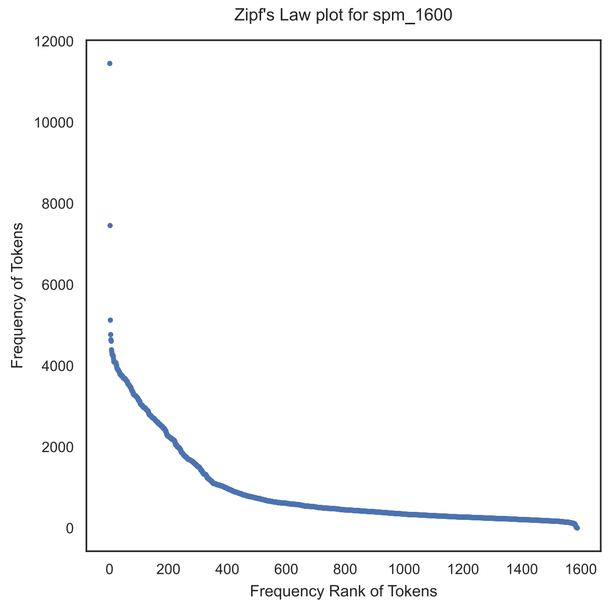
Experiments

Zipf’s Law

Zipf's Law states that the frequency of a particular element is inversely proportional to its rank. This means that a few elements occur very frequently, while the majority of elements occur infrequently.

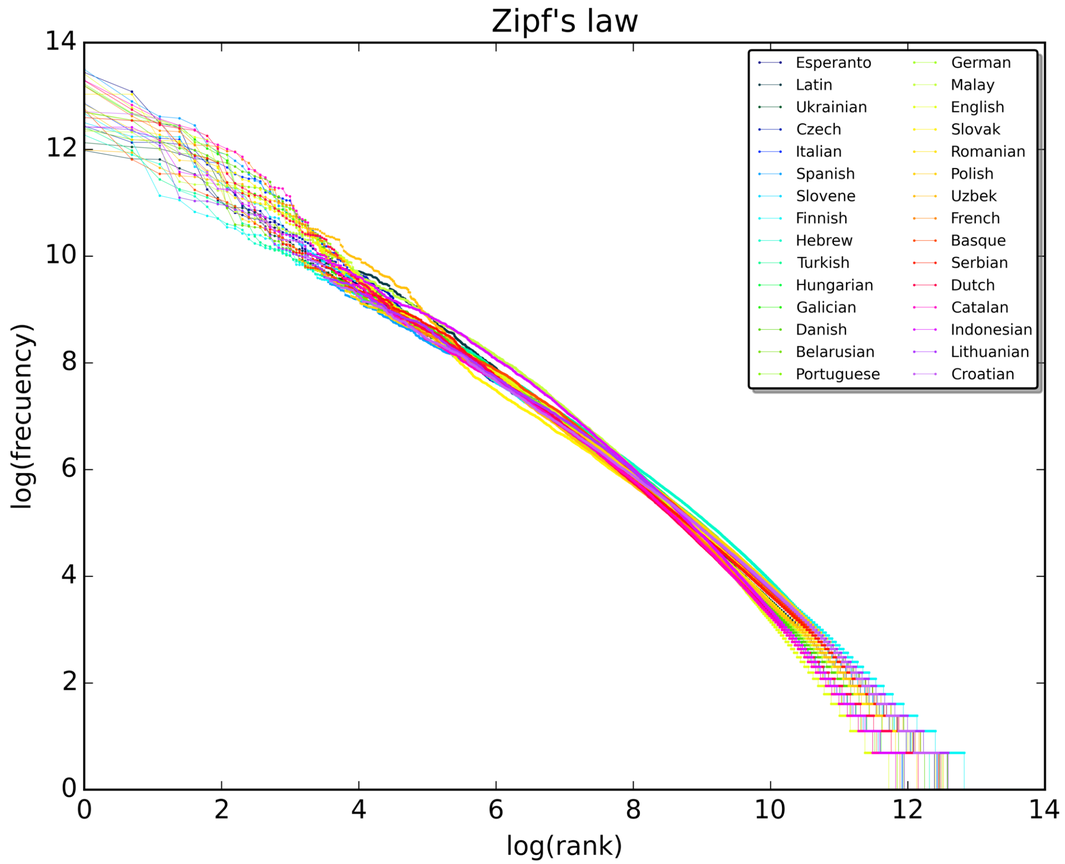






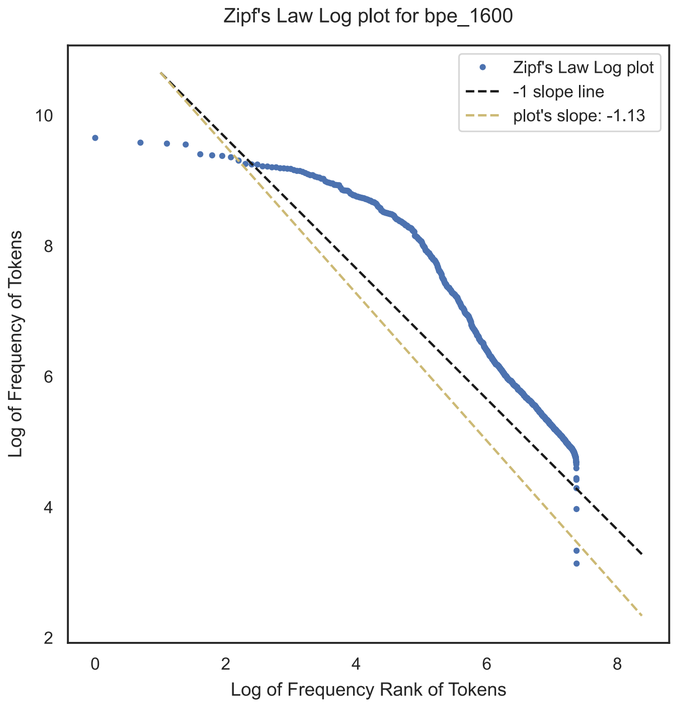
Experiments

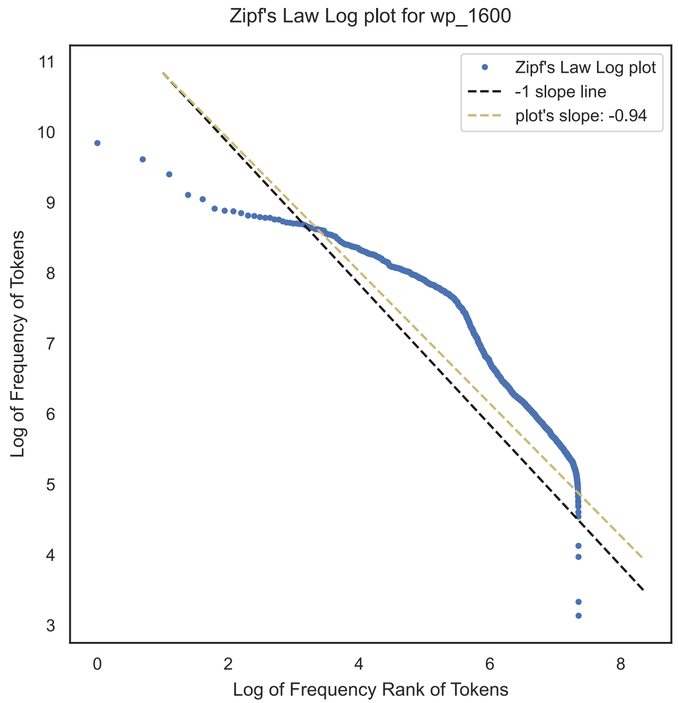
Zipf’s Law

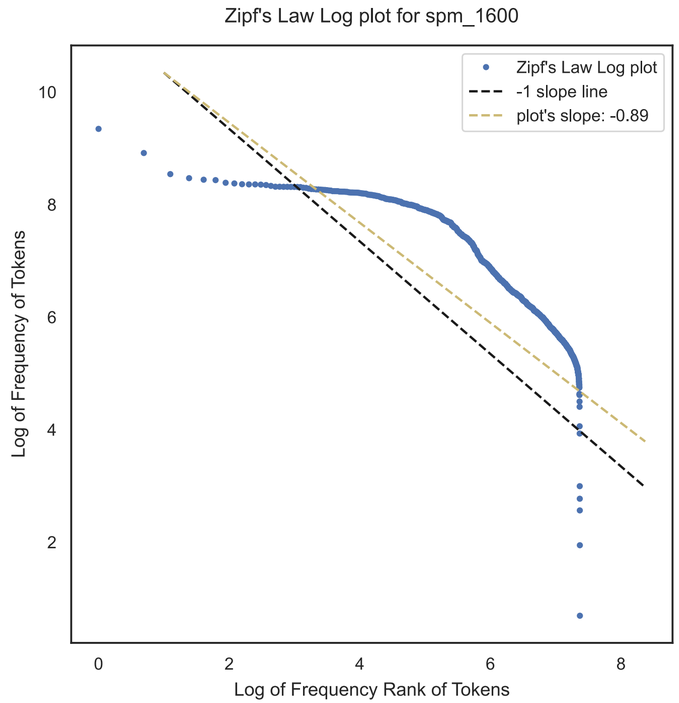


Experiments

Zipf’s Law

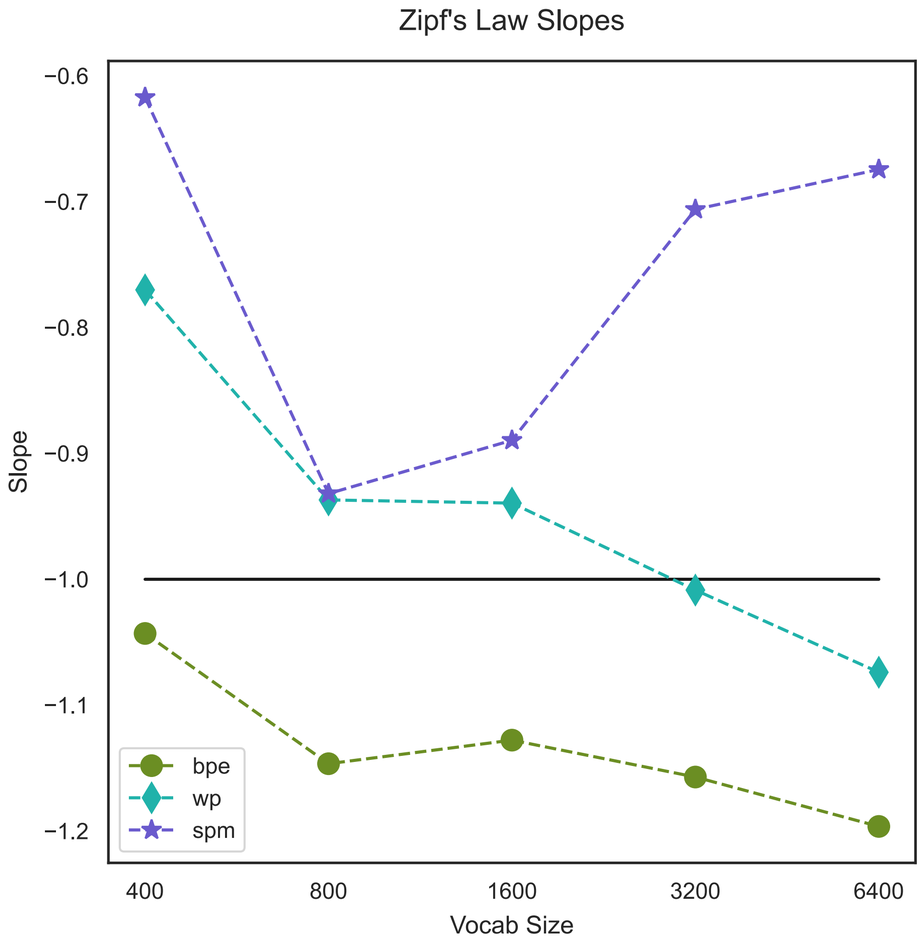






Experiments

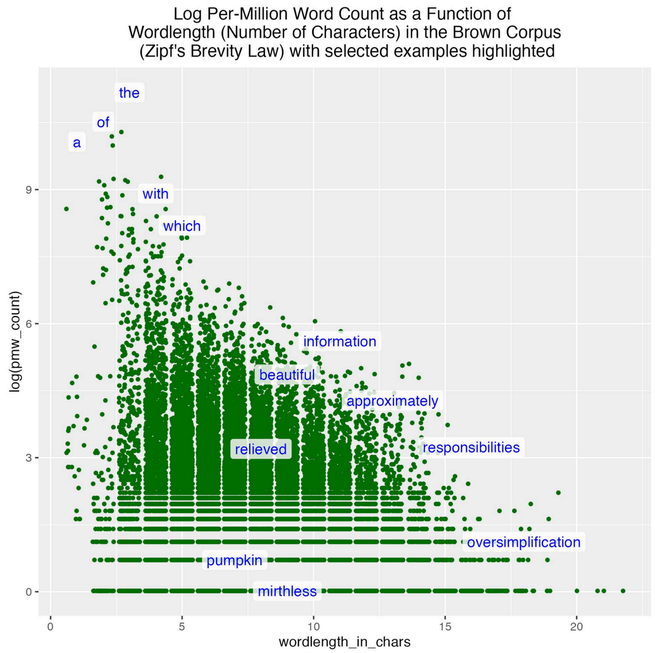
Zipf’s Law



Experiments

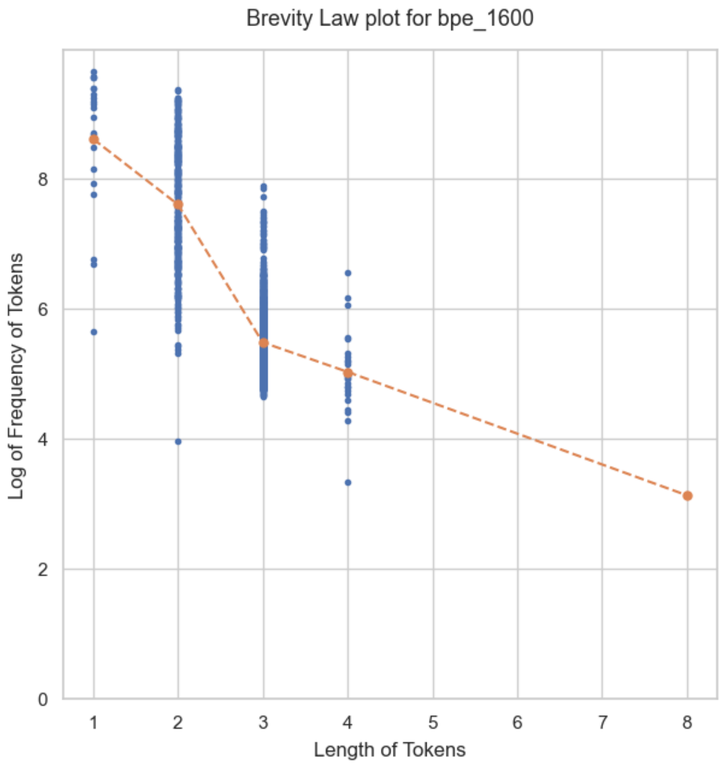
Brevity Law (Zipf's law of abbreviation)

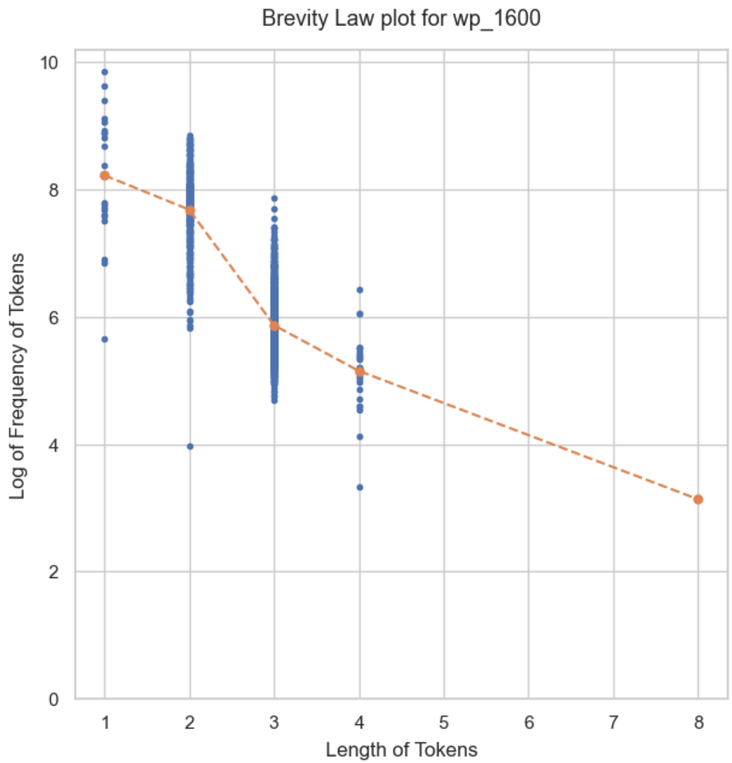
Brevity Law qualitatively states that the more frequently a word is used, the shorter that word tends to be, and vice versa; the less frequently a word is used, the longer it tends to be.

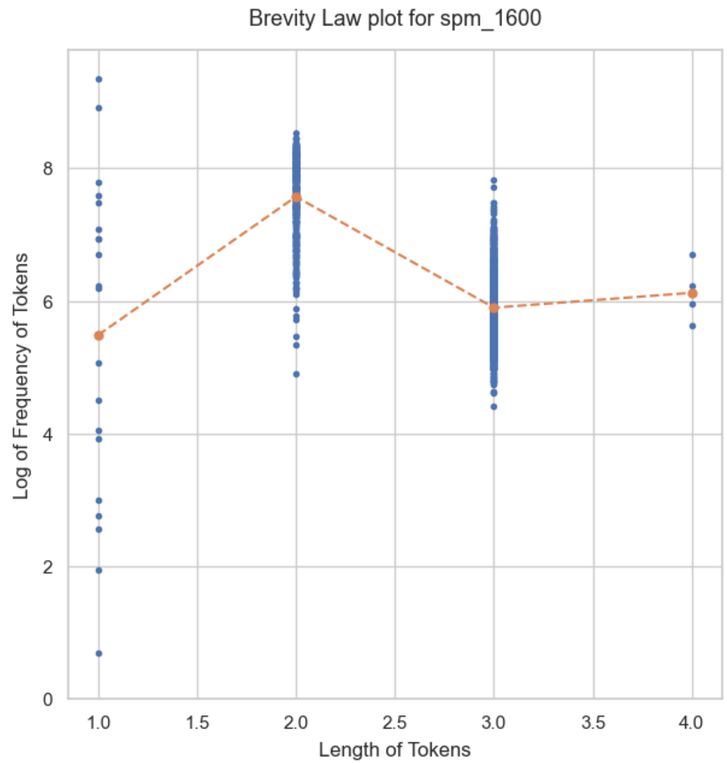


Experiments

Brevity Law (Zipf's law of abbreviation)

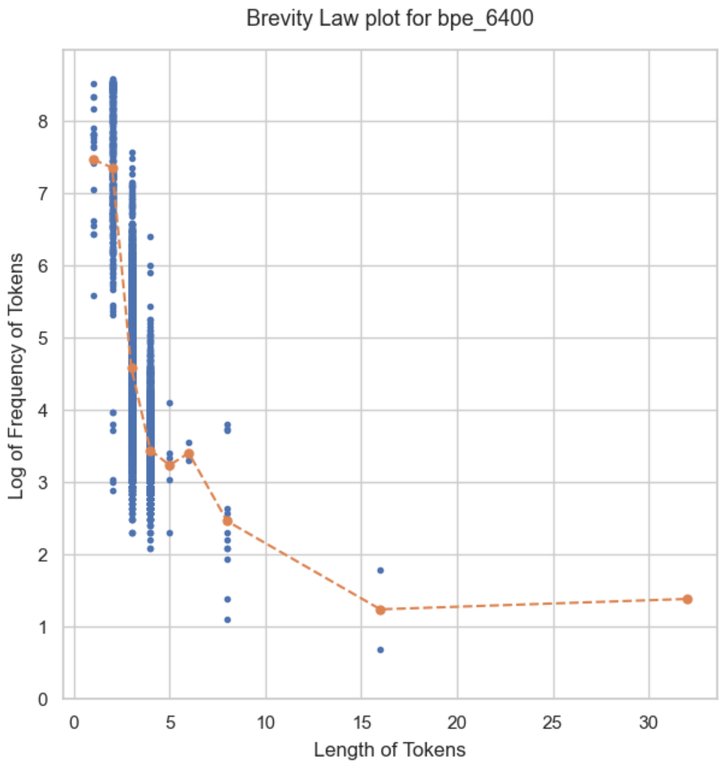


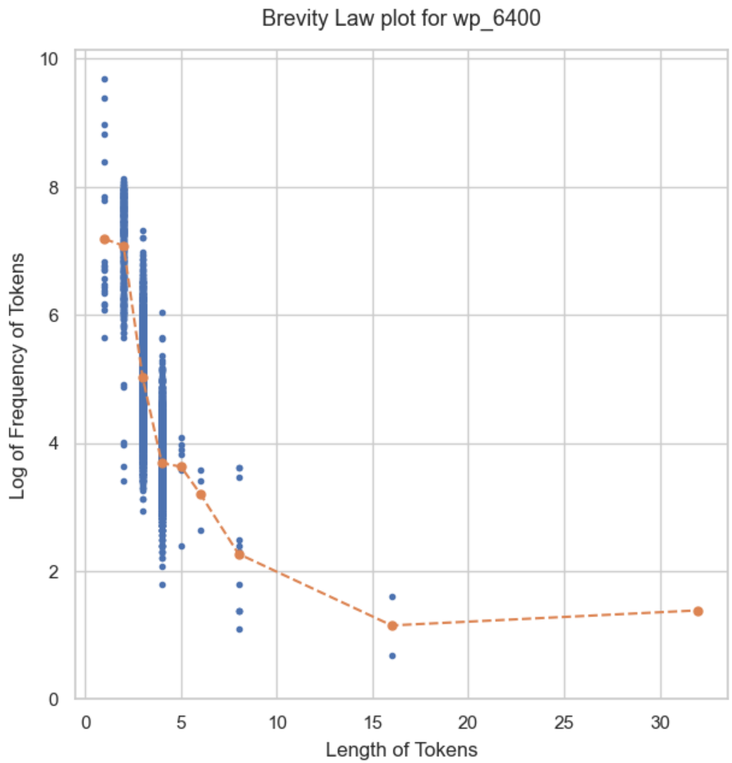


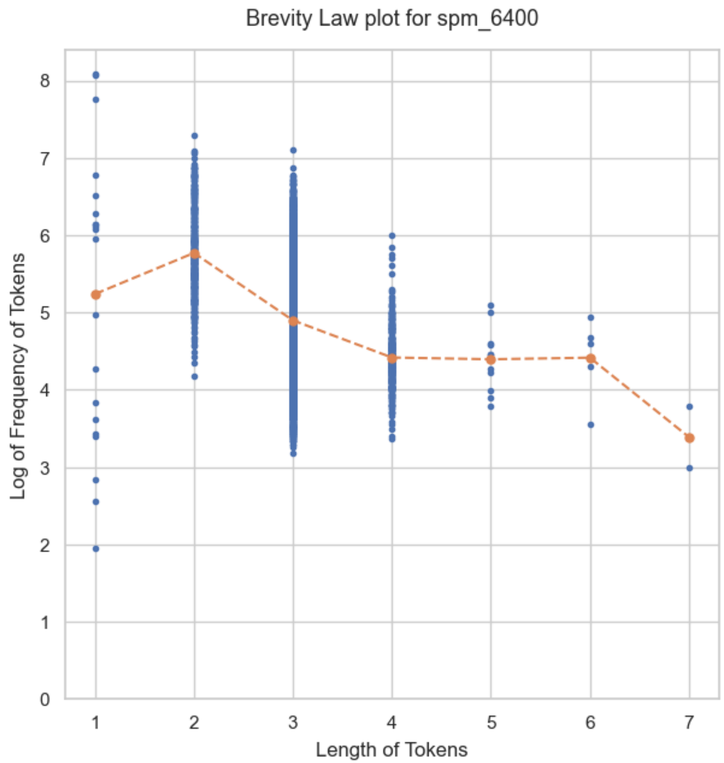


Experiments

Brevity Law (Zipf's law of abbreviation)

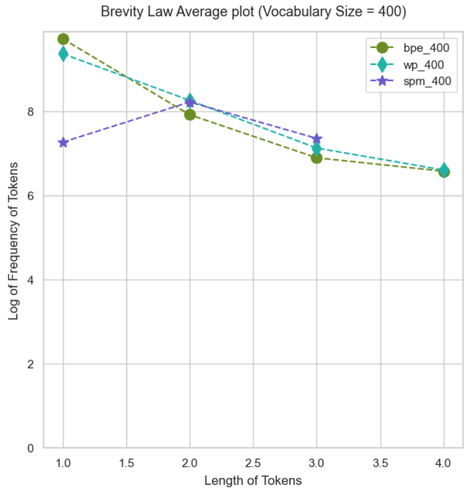


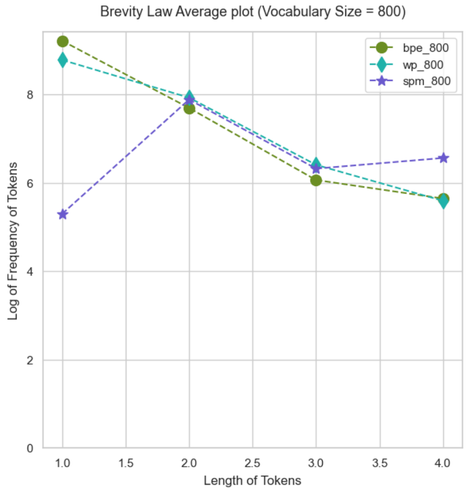


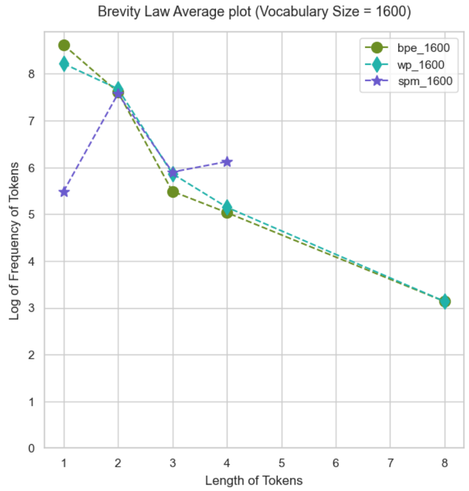


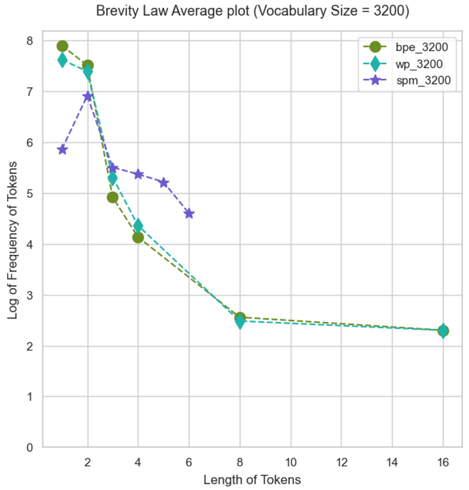
Experiments

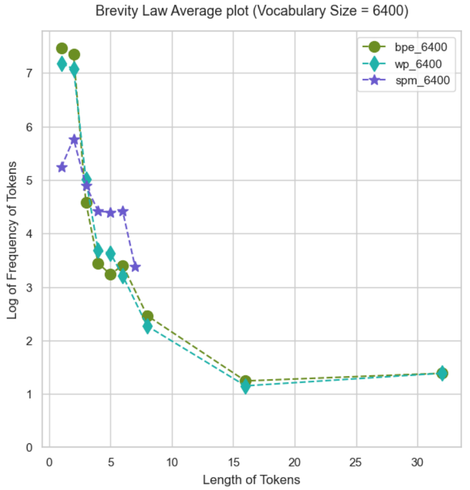
Brevity Law (Zipf's law of abbreviation)











Experiments

Heap’s Law (Herdan’s Law)

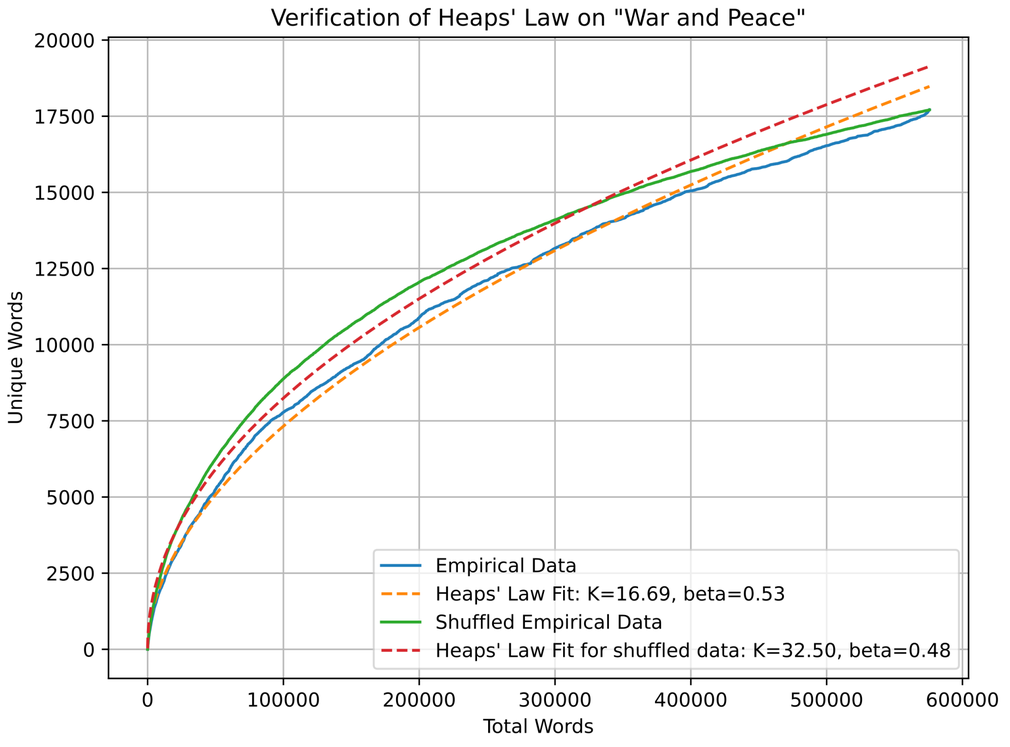
Heap's Law suggests that as the size of a document or dataset increases, the vocabulary size (the number of unique words) also increases, but at a decreasing rate.

The formula is expressed as:

$V(n) = K \cdot n^{\beta}$

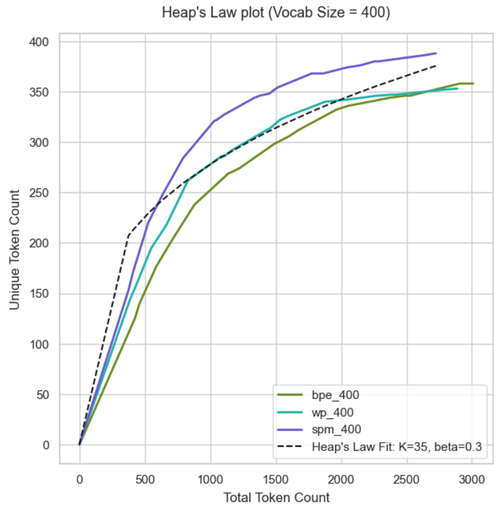
where:

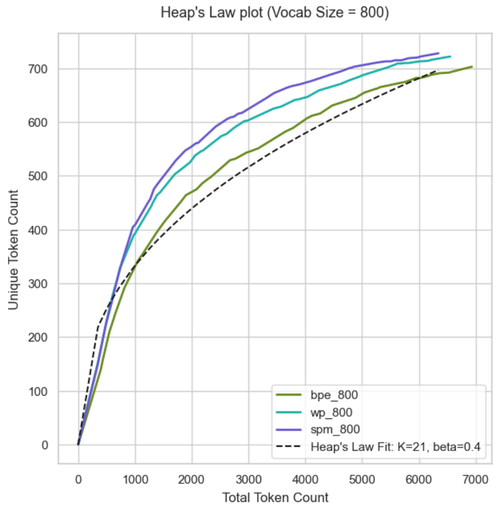
- V(n) is the estimated vocabulary size when the document or collection contains n words,
- K is a constant, typically in the range of 10 to 100.
- β is an exponent, typically in the range of 0.4 to 0.6.

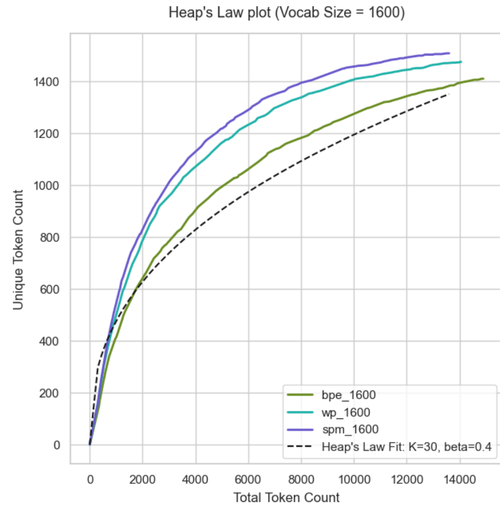


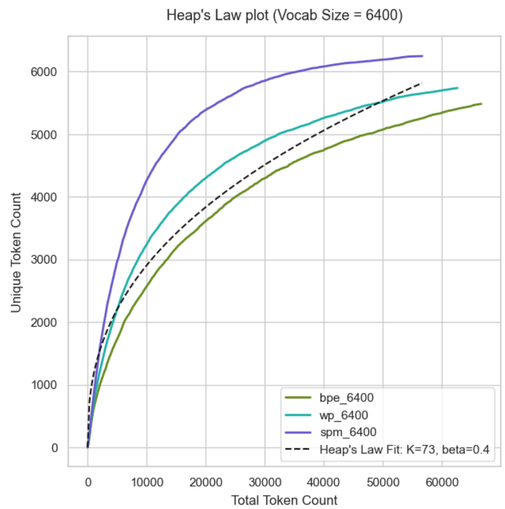
Experiments

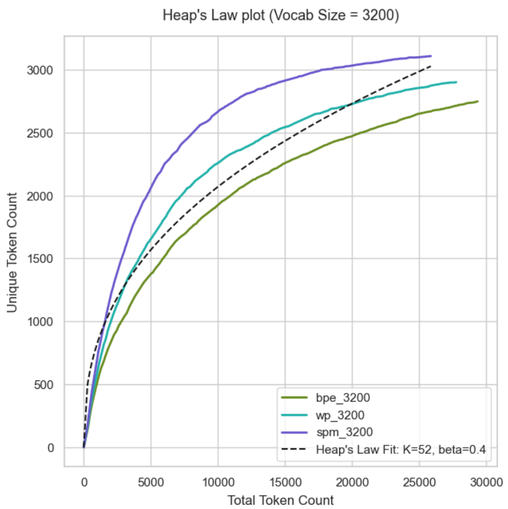
Heap’s Law (Herdan’s Law)







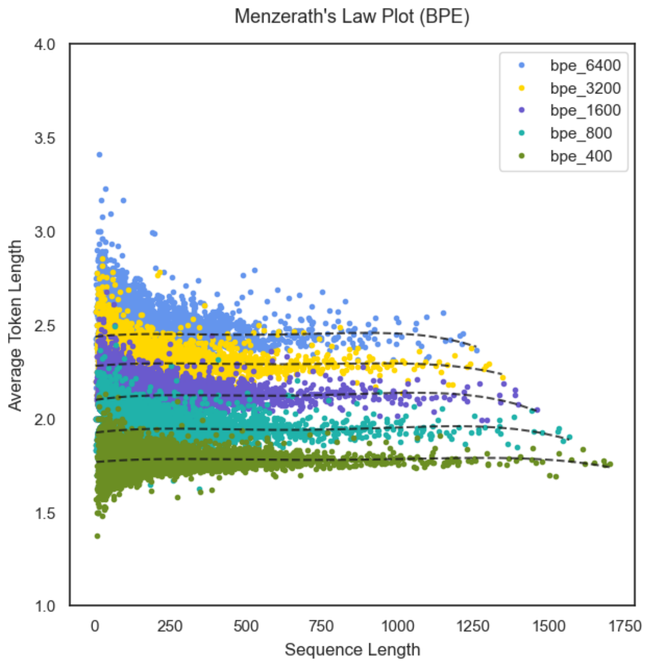


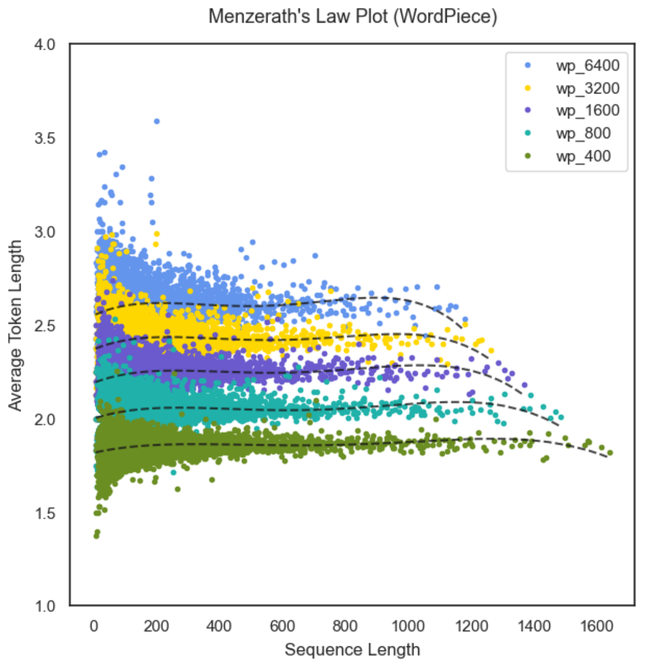


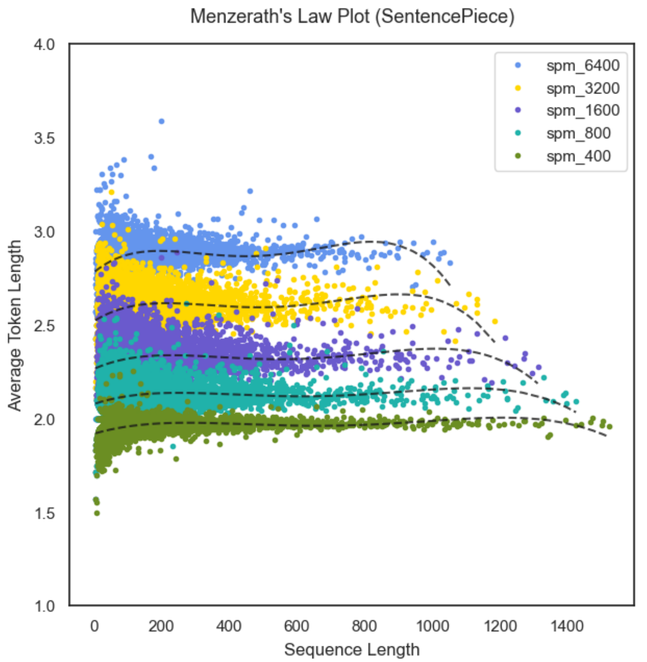
Experiments

Menzerath’s Law (Menzerath–Altmann law)

Menzerath’s Law is an empirical linguistic principle that describes the relationship between the size of linguistic constructs and the size of their constituents. The law is often expressed as follows: "The larger the whole, the smaller the constituents.” For instance, the longer a protein (in tokens), the shorter its tokens (in aminoacids).

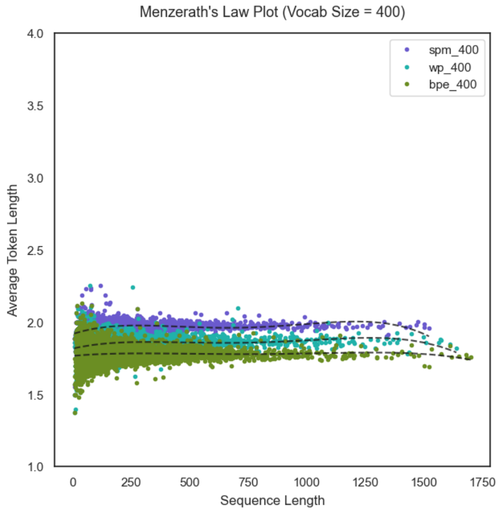


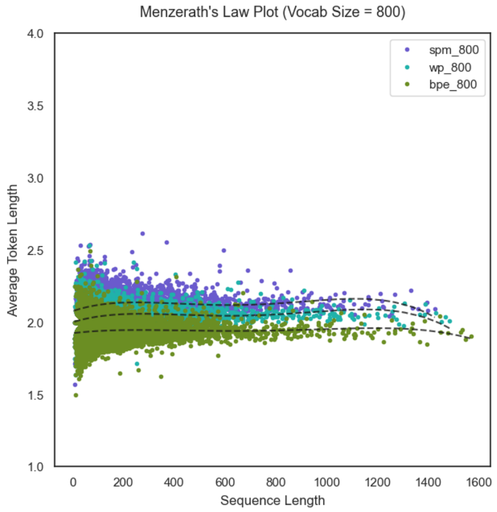


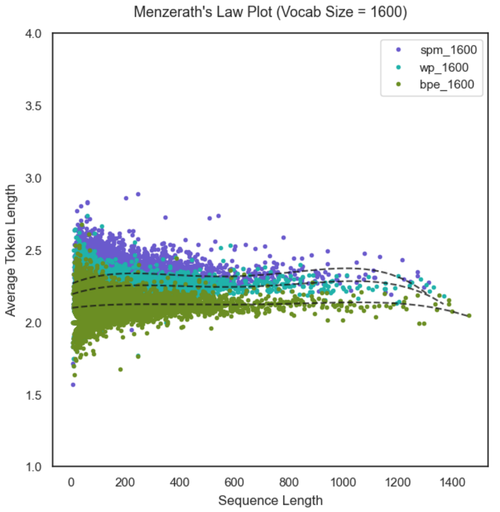


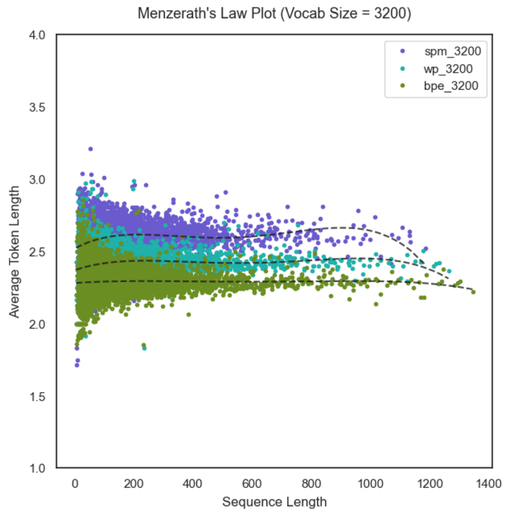
Experiments

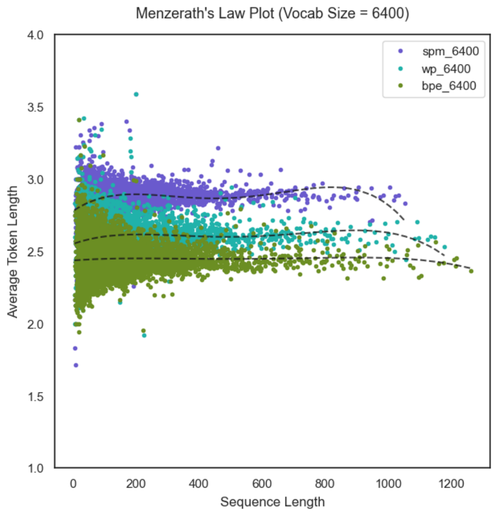
Menzerath's Law (Menzerath–Altmann law)











Conclusion

- Evaluated **Byte-Pair Encoding (BPE)**, **WordPiece**, and **SentencePiece** across varying **vocabulary sizes** (200, 400, 800, 1600, 3200, and 6400) in the context of protein sequences using the **UniRef50** dataset.
- Assessed each tokenizer through shared token percentages, token length distribution, fertility, and contextual exponence, in addition to examining their behavior under linguistic laws including Zipf's, Brevity, Heaps', and Menzerath's laws.
- Results illustrate distinct characteristics and efficiencies of **BPE**, **WP**, and **SPM** across various tokenization metrics, each showing strengths and weaknesses depending on the metric and vocabulary size.
- **All tokenizers** exhibit adherence to *Zipf's Law*, and *Heaps' Law*, while **BPE** and **WordPiece** show stronger compliance with *Brevity Law* compared to **SentencePiece** and **none** of them comply with *Menzerath's Law*.
- **BPE** tends to show *higher context usage*, **WP** balances between *encoding efficiency* and *token diversity*, and **SPM** exhibits *better efficiency in encoding* and *unique frequency distribution* characteristics.
- Overall, none of the tokenizers fully meet all the statistical metrics or adhere to the linguistic laws assessed. This indicates the need for a more effective tokenizer specifically tailored for protein sequences, especially if we continue to model proteins as analogous to linguistic systems.

Future Work

- Repeat experiments for larger vocabulary sizes.
- Include different metrics such as parity, productivity, and idiosyncracy.
- Compare these results to Evolutionary Subword Tokenization approaches.
- Treat species as different languages and work on specie-focused tokenizers.
  - Train specie-specific tokenizers and
    - apply tests that are available to multilingual approaches.
    - combine them for a multi-specie tokenizer.
    - compare the results with the generic approach.

Thank You For Listening