NATURAL LANGUAGE PROCESSING

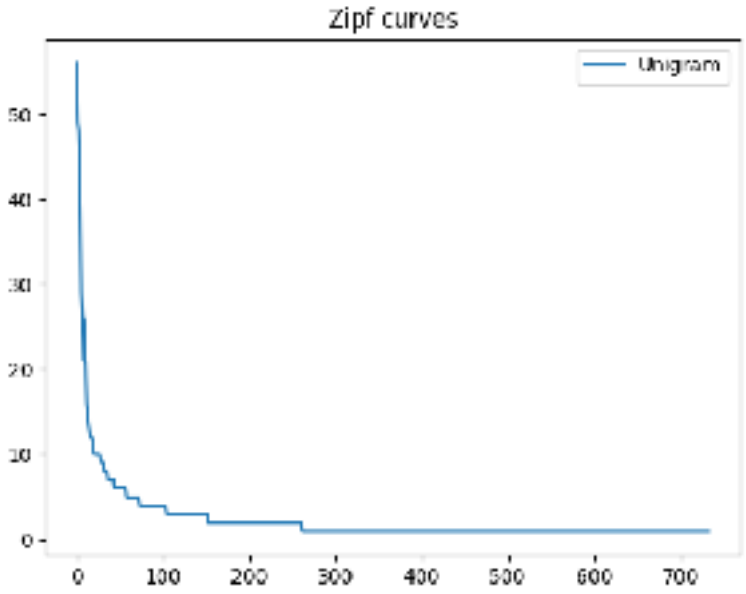
REPORT

LANGUAGE MODELS

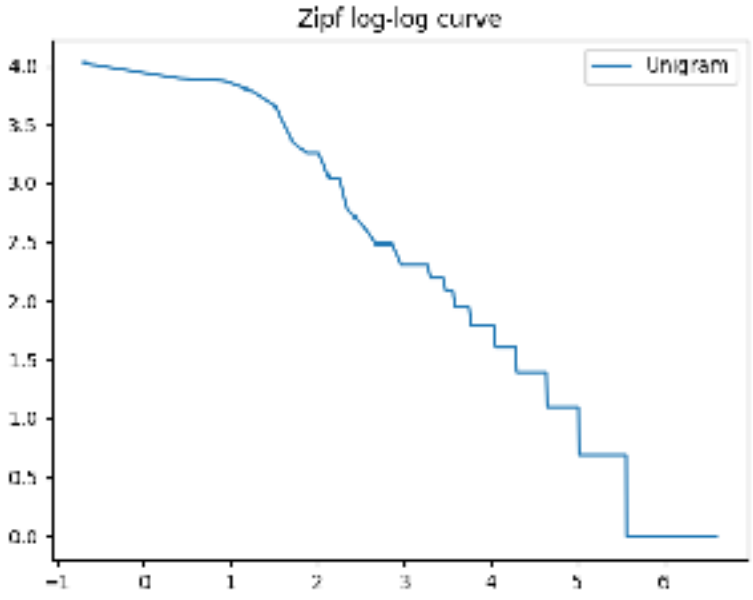
We implemented the unigram, bigram and trigram language models via the P(MLE) counts for the same. Below attached will be the graphs for the same

FOR UNIGRAMS:

Zip-f curve:



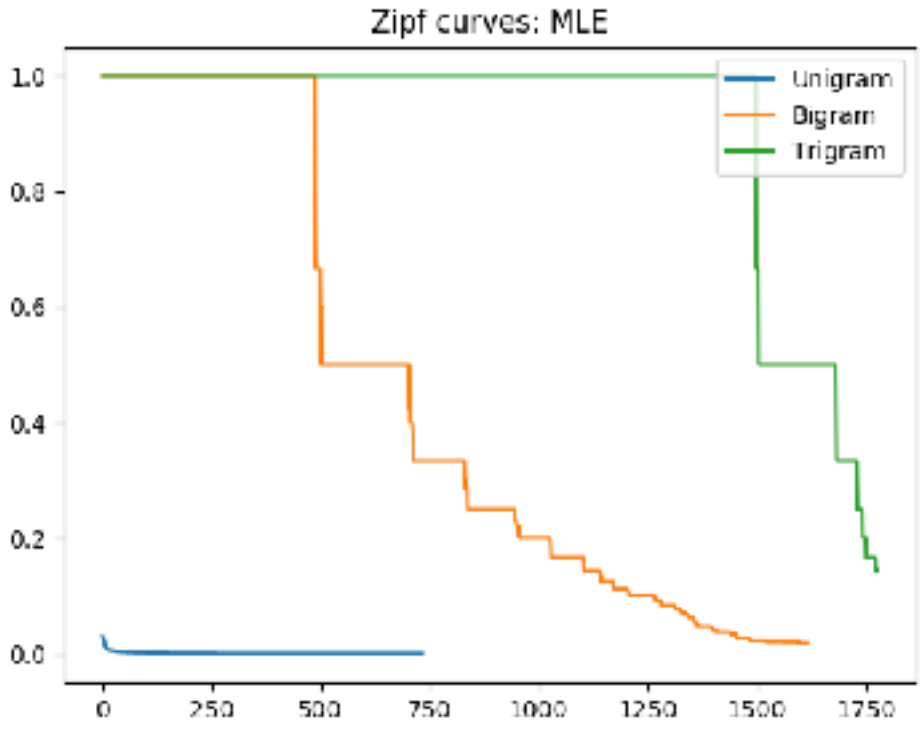
Log-Log Curve:



OBSERVATIONS:

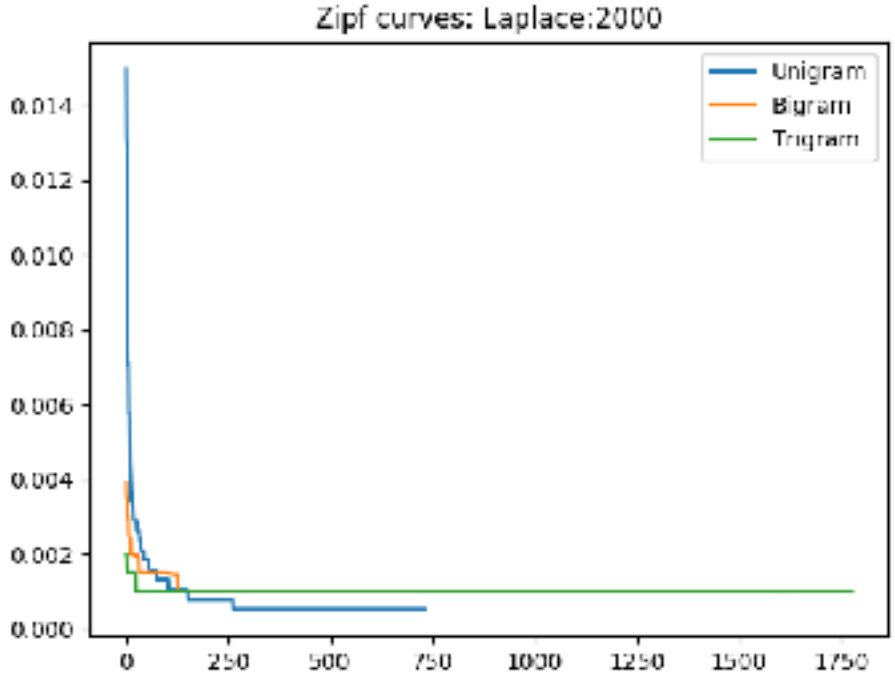
It is observed that the distribution of the unigrams, bigrams and trigrams follow the Zip-f's Law. (Power Law).
Upon plotting on the log-log scale, it is found that at certain places, we can get an indication of a straight line but in some places where the curve is flat, we couldn't find the straight line, rather a curved line in those spots.

MLE:

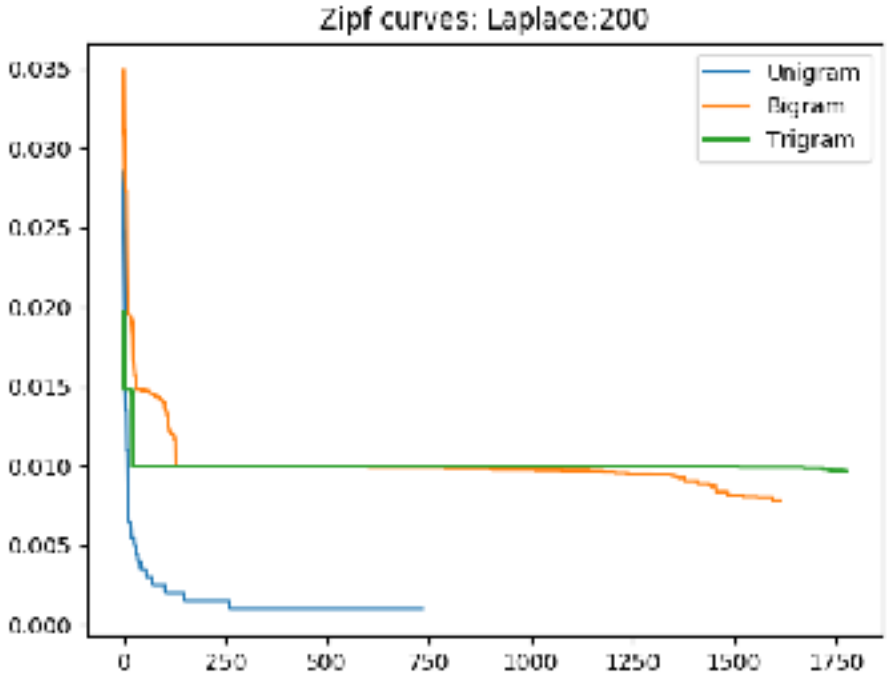


Laplace:

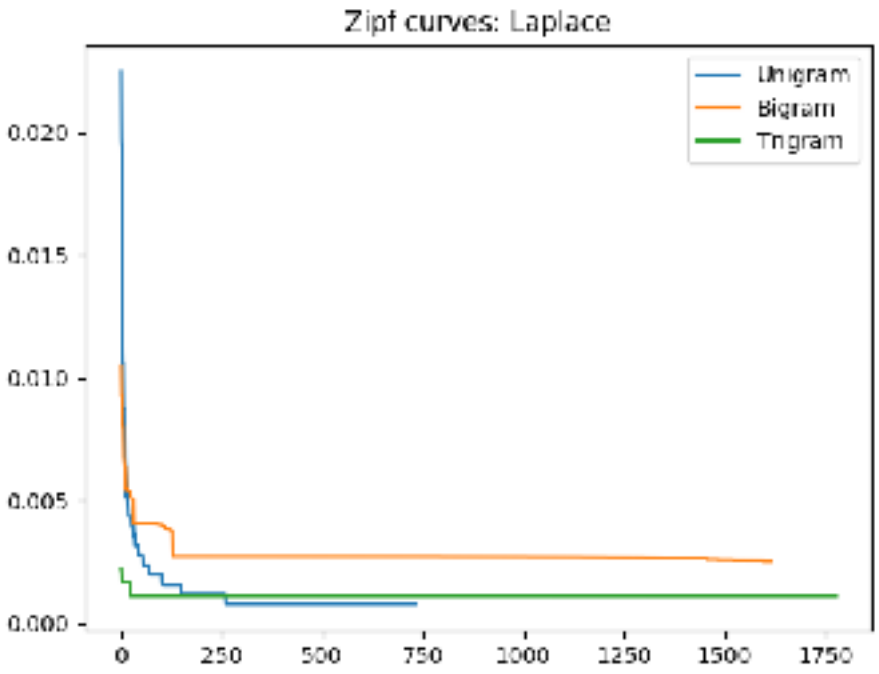
- **Vocabulary = 200**



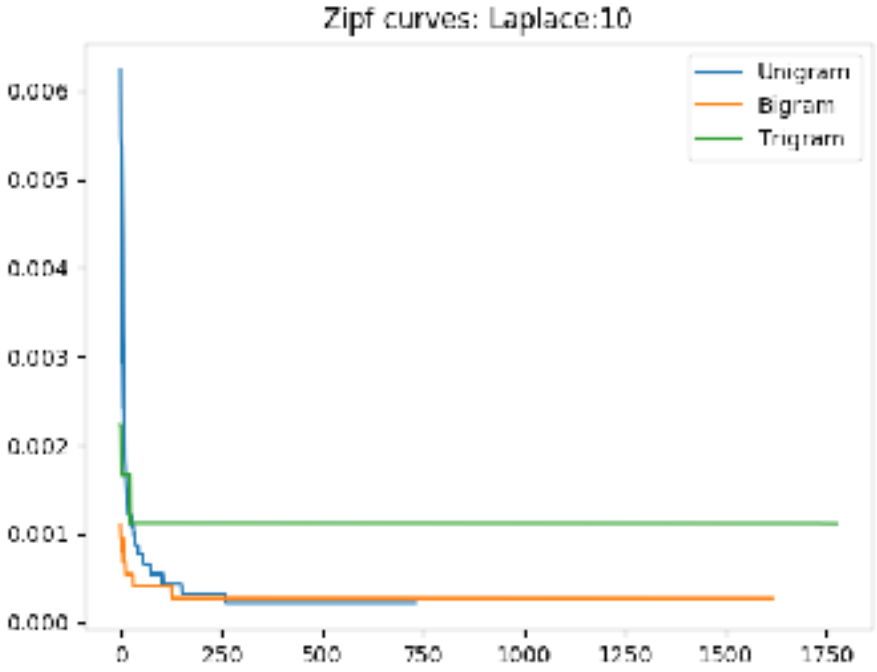
- **Vocabulary:2000**



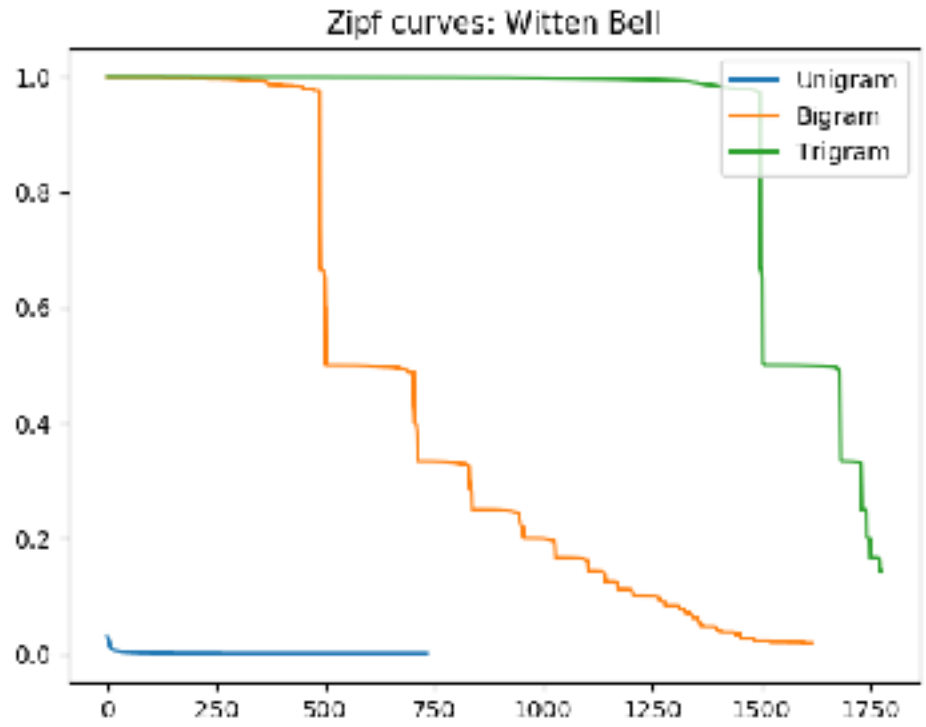
- **Vocabulary:Size of vocabulary**



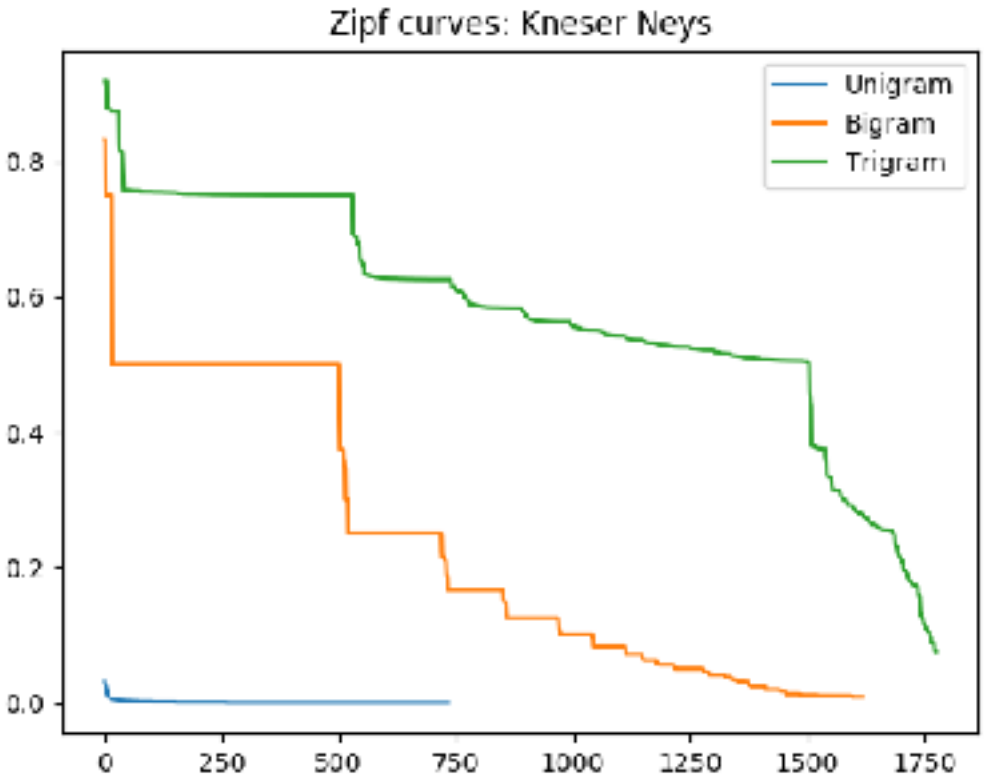
- **Vocabulary: 10 * size of vocabulary**



WB:



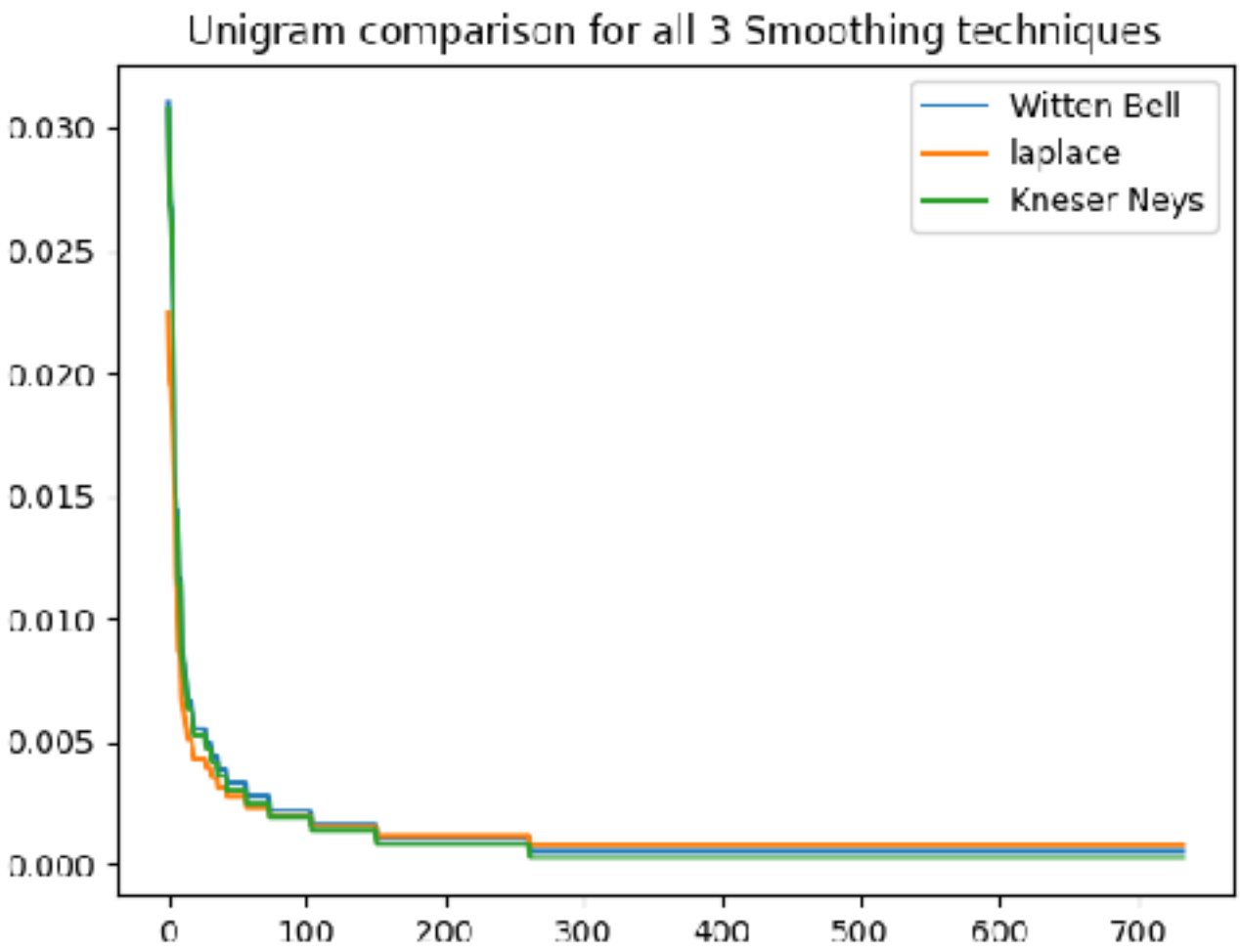
Kneser-Ney:



SMOOTHING METHODS

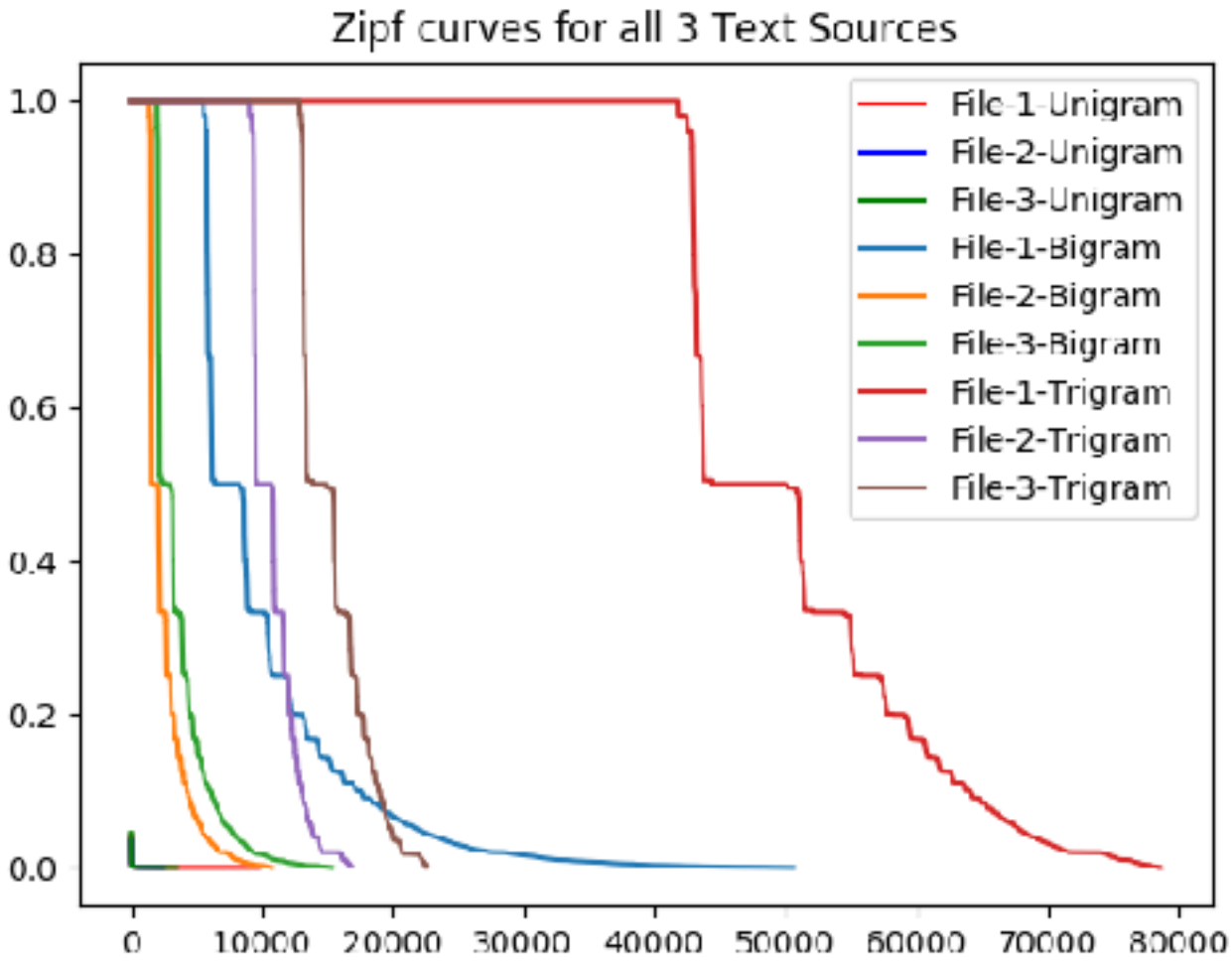
Employing a series of smoothing methods, it was found that most of the smoothed counts differed in the minute probabilities and Kneser-Ney provided a smoother curve. Although noting such differences in the graphs can give false positives unless one analyses them on generating texts, I below note down a comparison for the same.

Comparison of Unigram Smoothing of Laplace,KN and WB



Naive Bayes

ZIP-F's curves for all three sources



I formulated the Naive Bayes Supervised model for tokenisation by first annotating a small part of the corpus by classifying each of the characters of the text into either of

- Beginning of token[B]
- Inside Token[I]
- Outside Token[O]
- Ending of Token[E]
- Single Word Token[S]

Thus, we used this to determine tokens in tested data set using the naive Bayes algorithm

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(c \mid d)$$

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(d \mid c)P(c)}{P(d)}$$

$$= \underset{c \in C}{\operatorname{argmax}} P(d \mid c)P(c)$$

Tokenizer

For tokenization, a hack that was used was that all the non-alpha numeric characters excluding space were removed that converted the data into a easy to parse dataset