NLP Assignment :1

Submitted By: Anupam Pandey
Roll No: 20162118

1. Tokenization:

By seeing the data observations was like:
-presence of URL’s
-garbage data like “ “, “&gt” etc..
-Important puncuations needs to be preserved
-remove extra spaces
-dates (Numeric data need to be preserved)

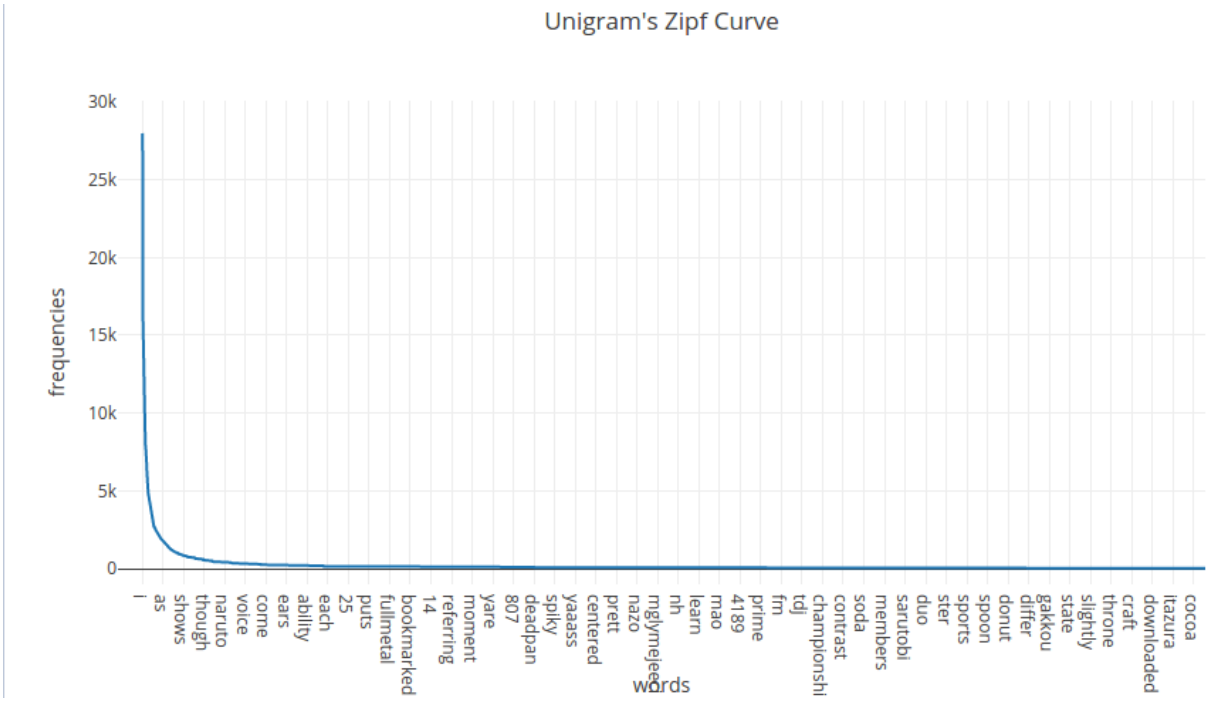
Ist step:
preserving the URL’s
2nd step :
remove garbage words
3rd step:
preserving the puncuations because for language modelling (i.e for Unigram , Bigram & Trigram) it is important to know that when a sentence is ending or getting started.
4th step: removing of extra spaces

5th step: numeric data needs to be preserved

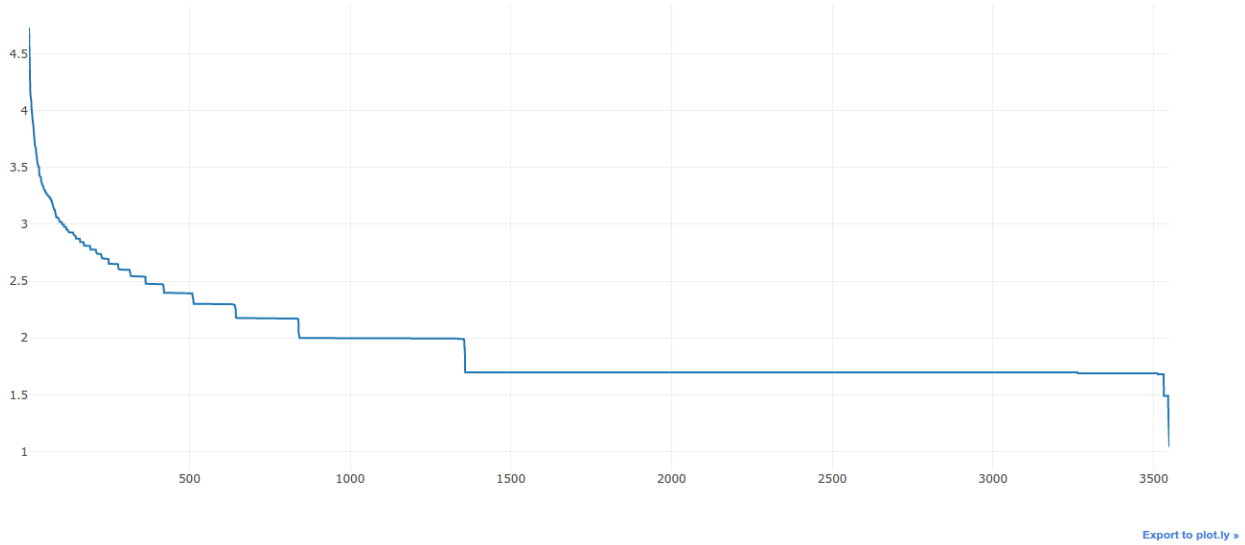
Language Modelling:

Unigram:splitted the text into tokens as mentioned above and also counting frequency for each and every token which is used later to find the probablity, it is also used in capturing and updating the count for each token (e.g in smoothing technique) . The frequency is also use to plot the log-log curve & ziph curve for Uni,Bi & Trigram.

Unigram ziph curve for Anime.txt:

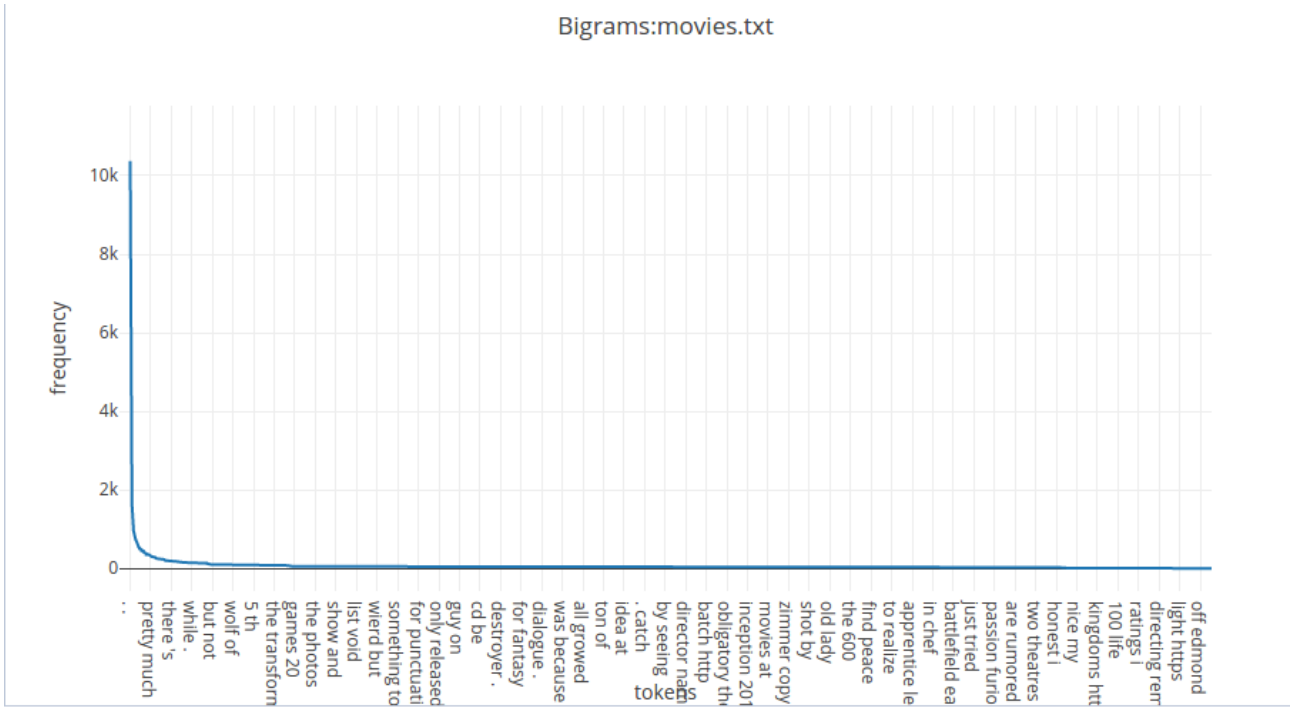


log-log Anime.txt curve:

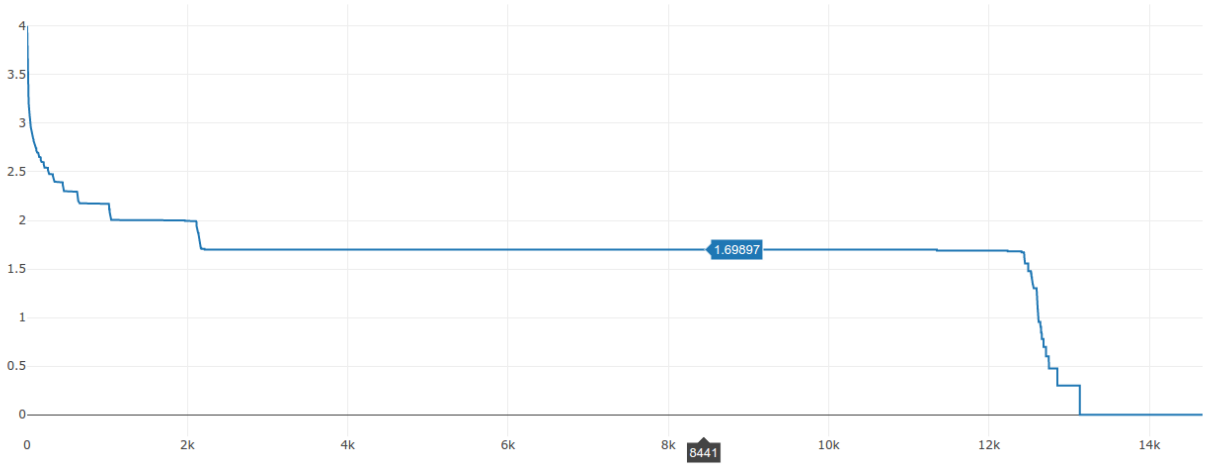


rank on X : axis & frequency on Y axis

Bigram ziph curve for movies.txt:

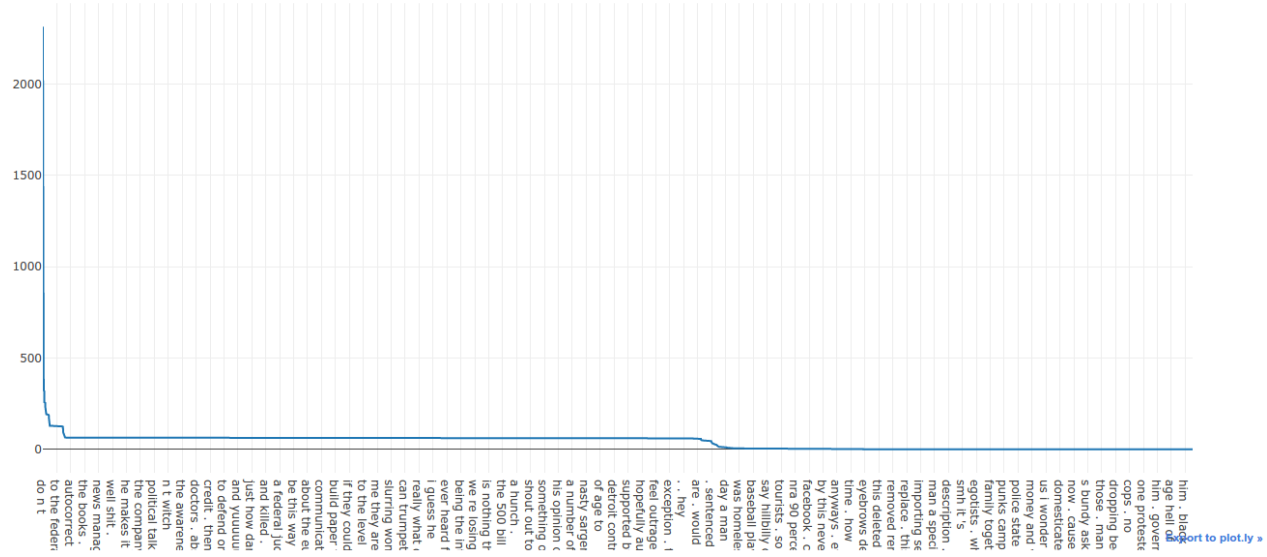


log-log curve for movies.txt:

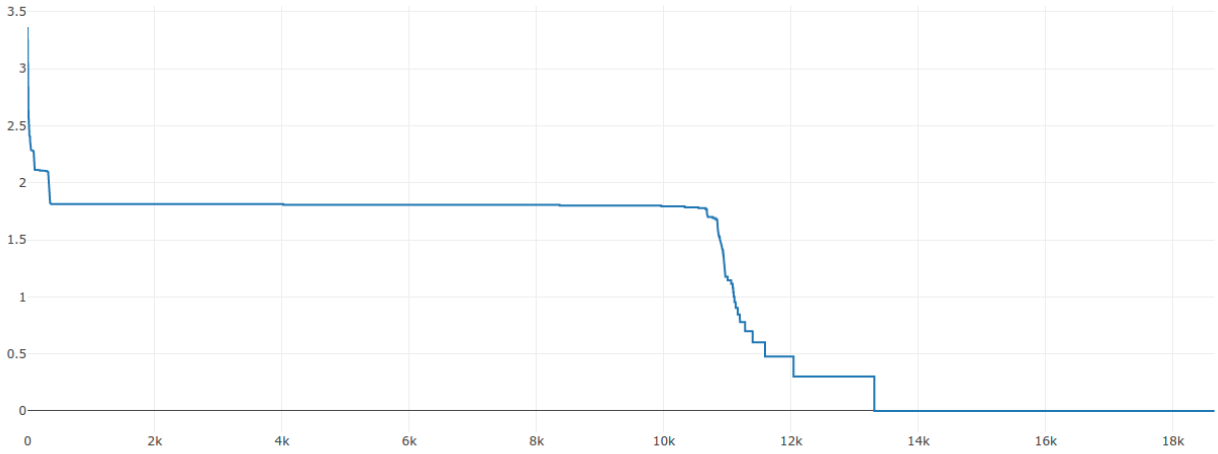


rank on X : axis & frequency on Y axis

Trigram ziph curve for news.txt:



log -log curve : trigram : news.txt:

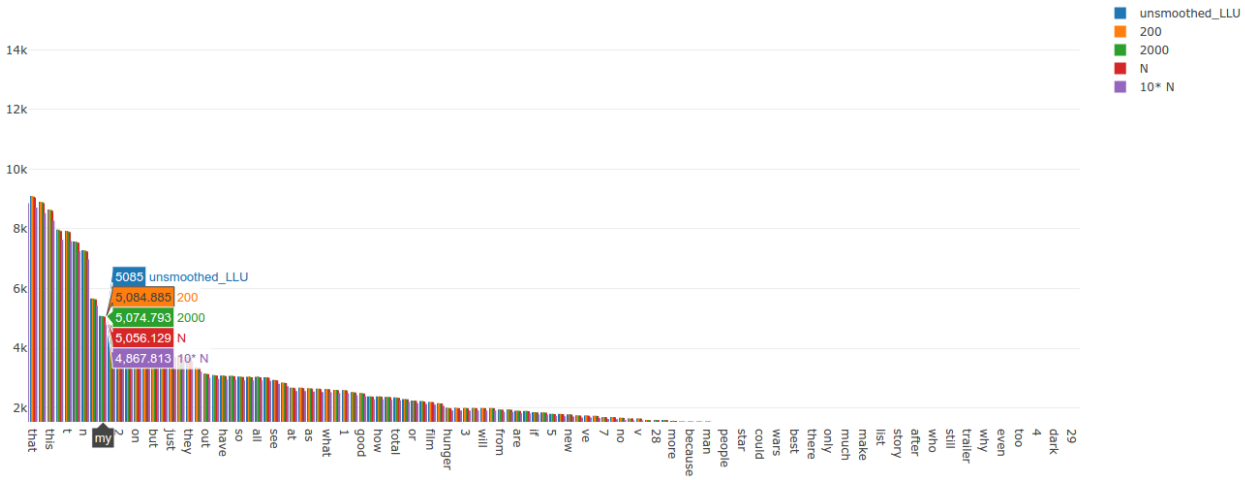


rank on X : axis & frequency on Y axis

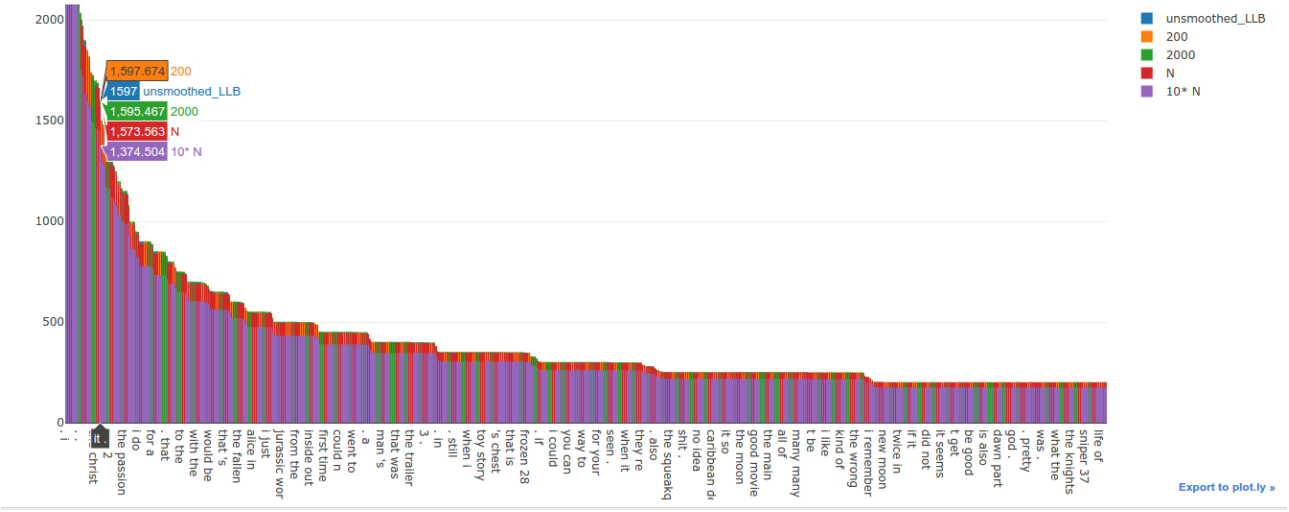
laplace / Add one smoothing :

it is done for different set of values as given in the question:
-unsmoothed
-smoothed_200
-smoothed_2000
-N*10
-N
N : size of Vocabulary

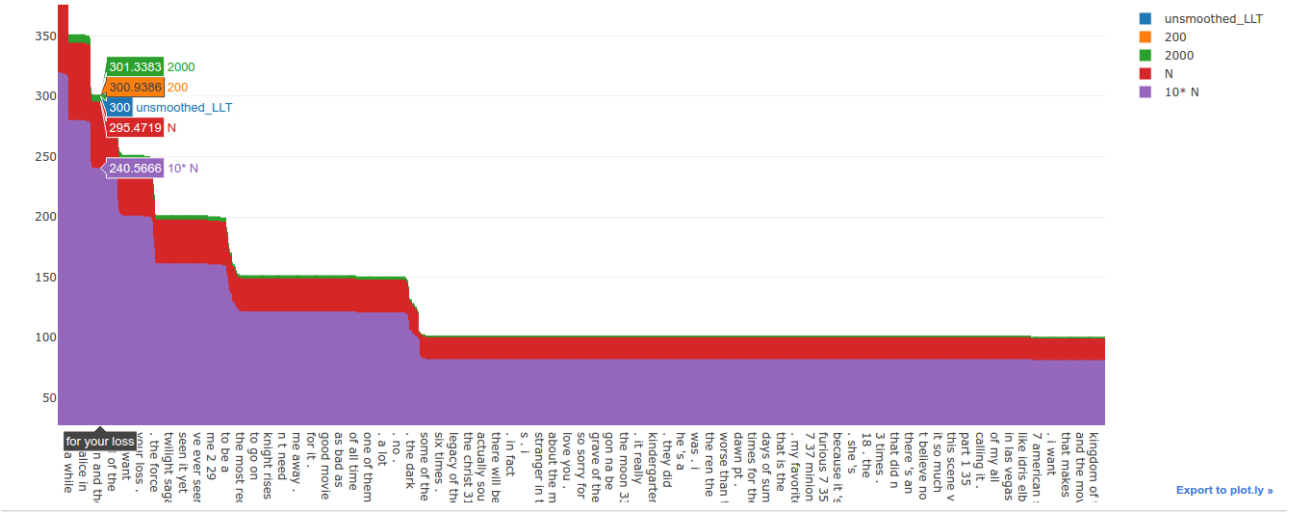
unigrams:



bigrams:



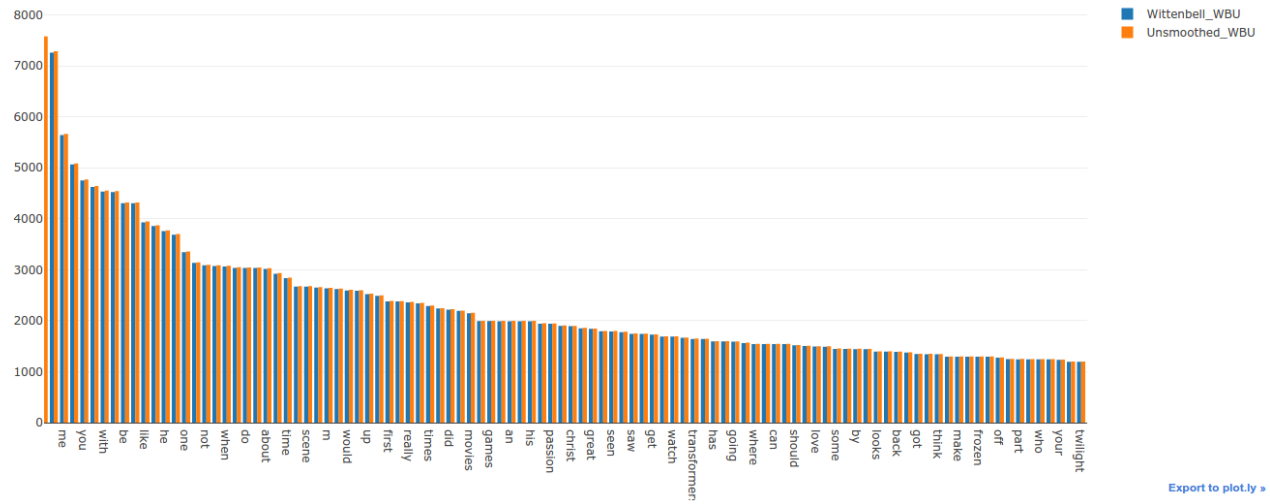
trigrams:



it was being noticed that smoothed 2000 & smooth 200 was almost the same.Un smoothed was on the top &for 10*N there is lot of shift in probability mass from words with count greater than 0 to words with count equal to zero

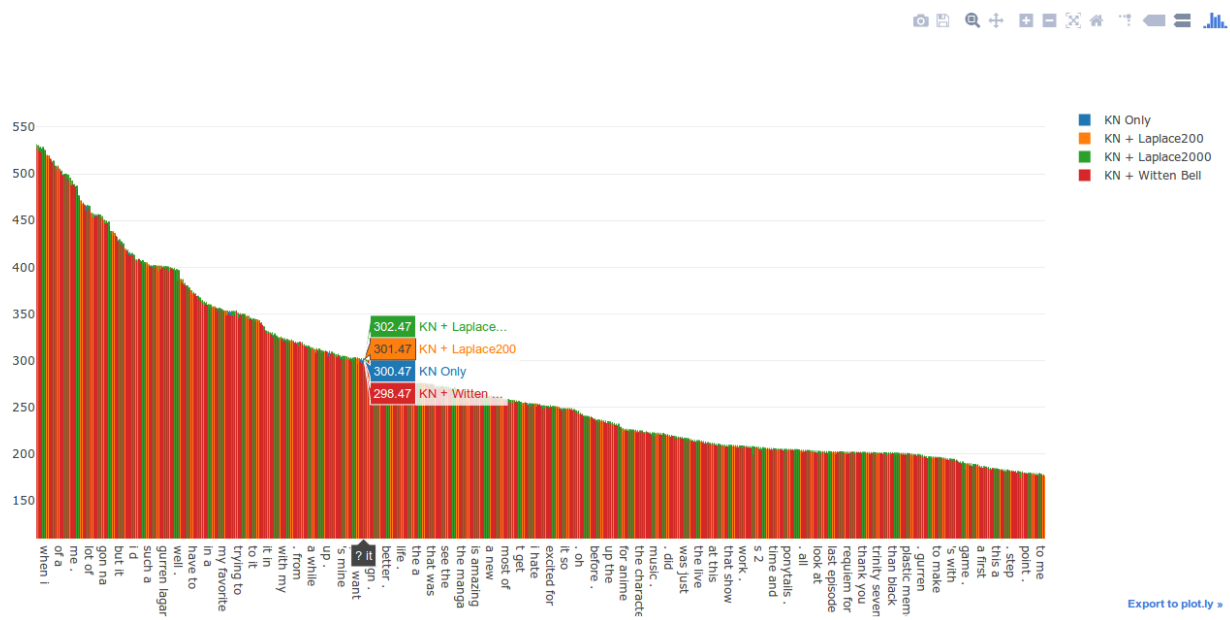
Witten-Bell backoff:
the idea is to get the probability of those tokens which are actually not present in the data set with those tokens which are being seen one time.since no test set is there so all the count will be greater that zero

unigrams:

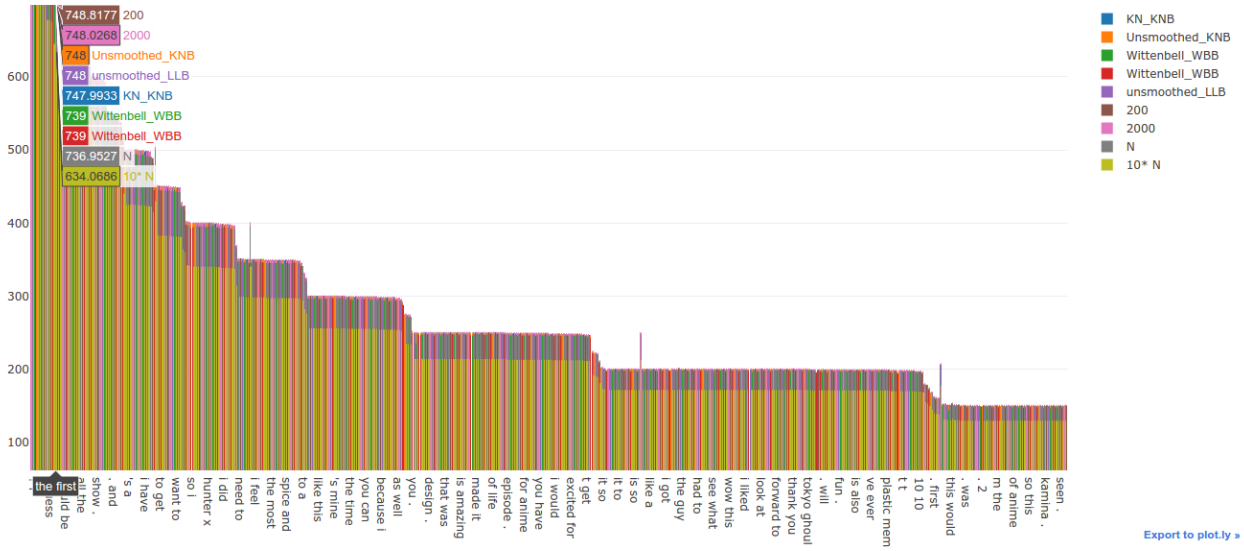


kneser-ney:

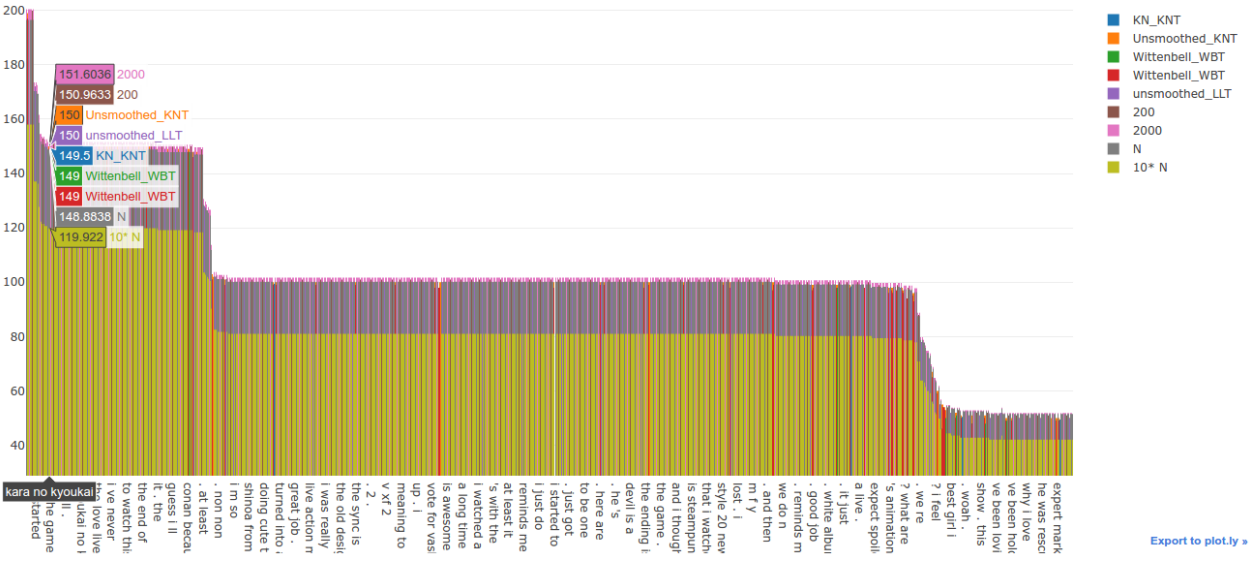
kneser ney performs really well as comapared to the Wittenbell & laplace, idea of kneser ney is the same as witten bell & as per the observation KN + WB in absolute discounting outperforms every other algorithm



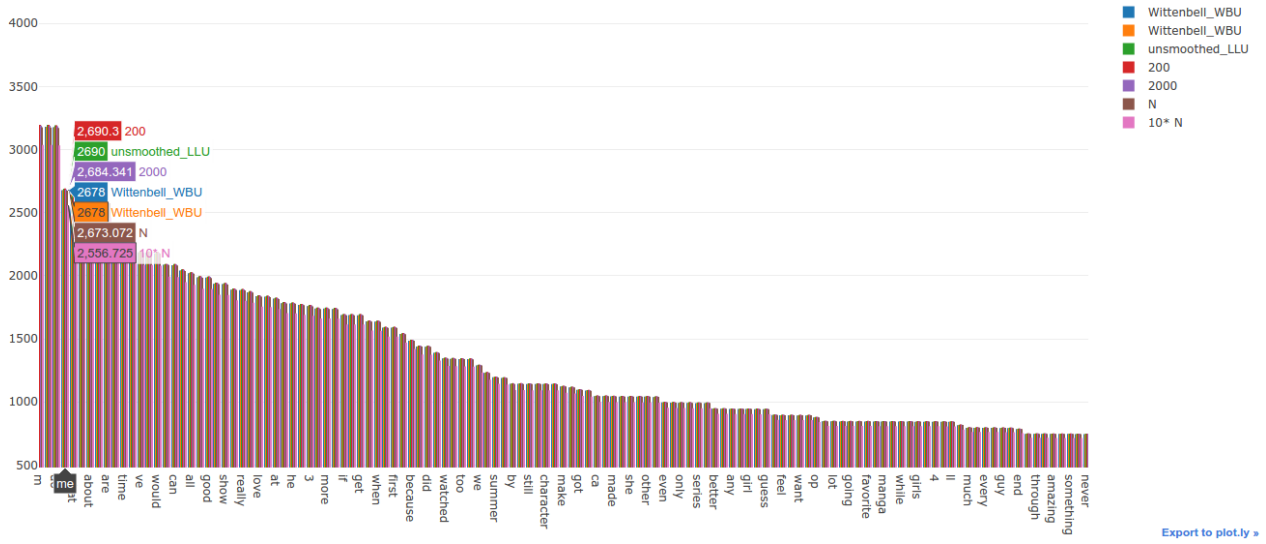
comparison:
All algo + bigram+anime.txt:



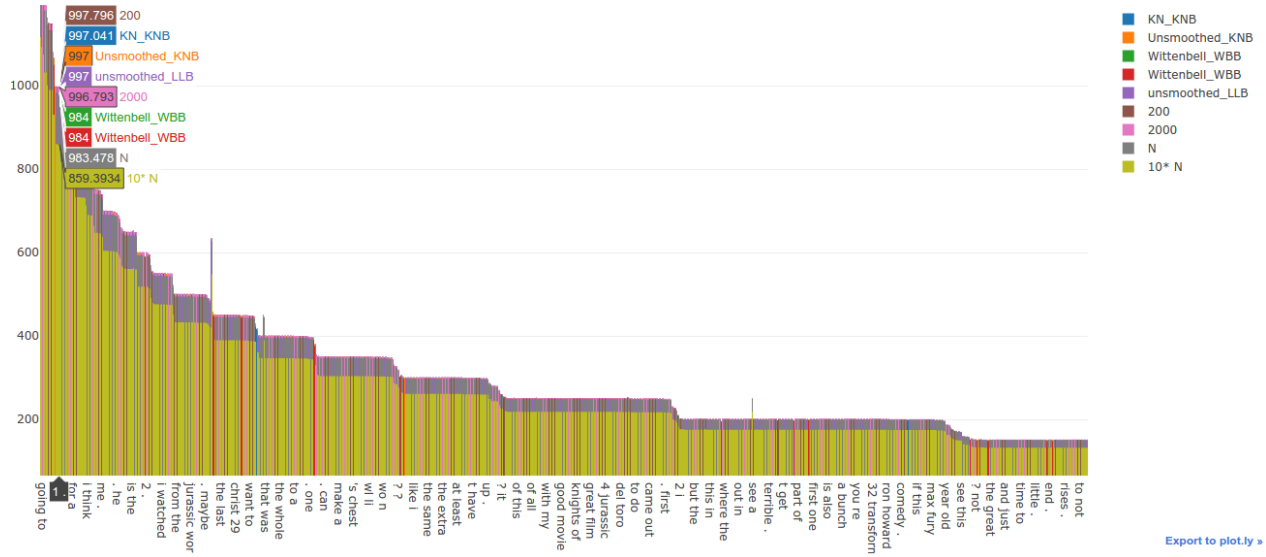
All algo + trigram+anime.txt:



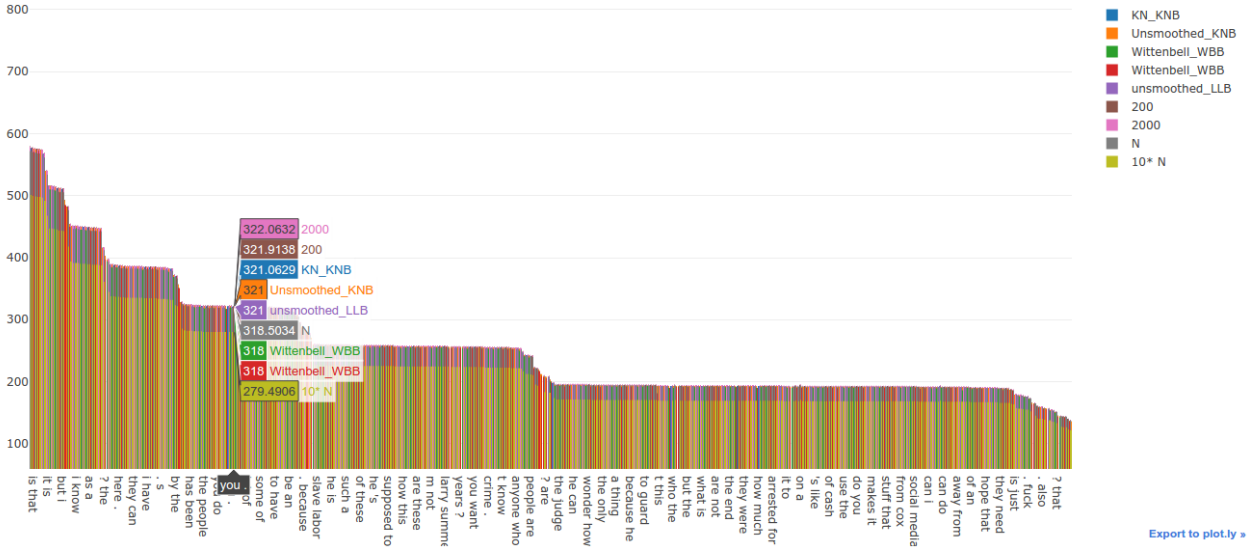
All algo + unigram+anime.txt:



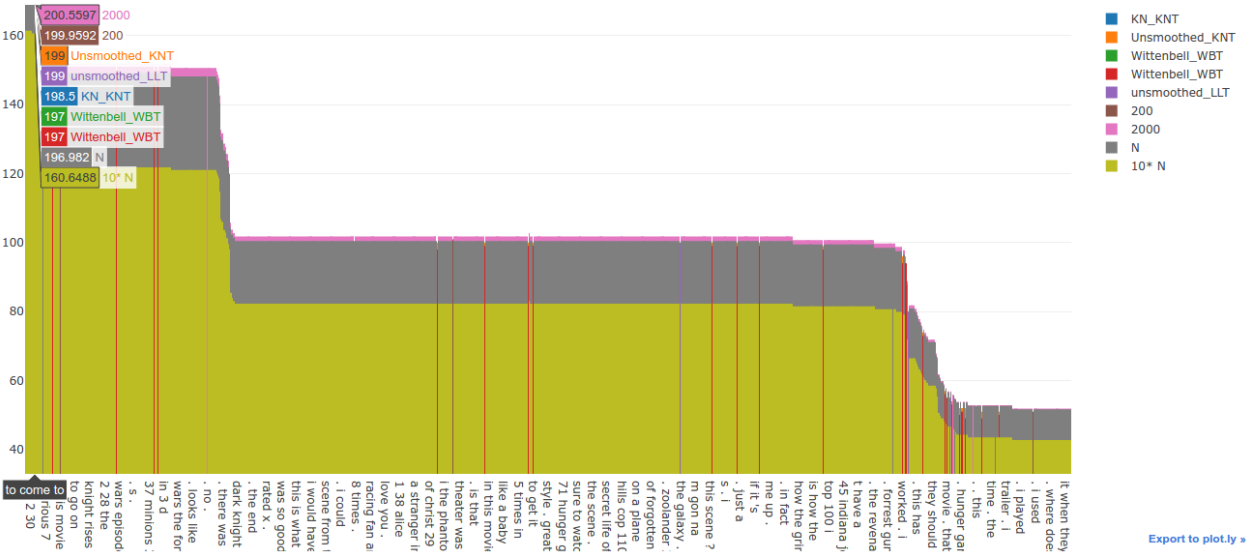
All_bigram_movies:



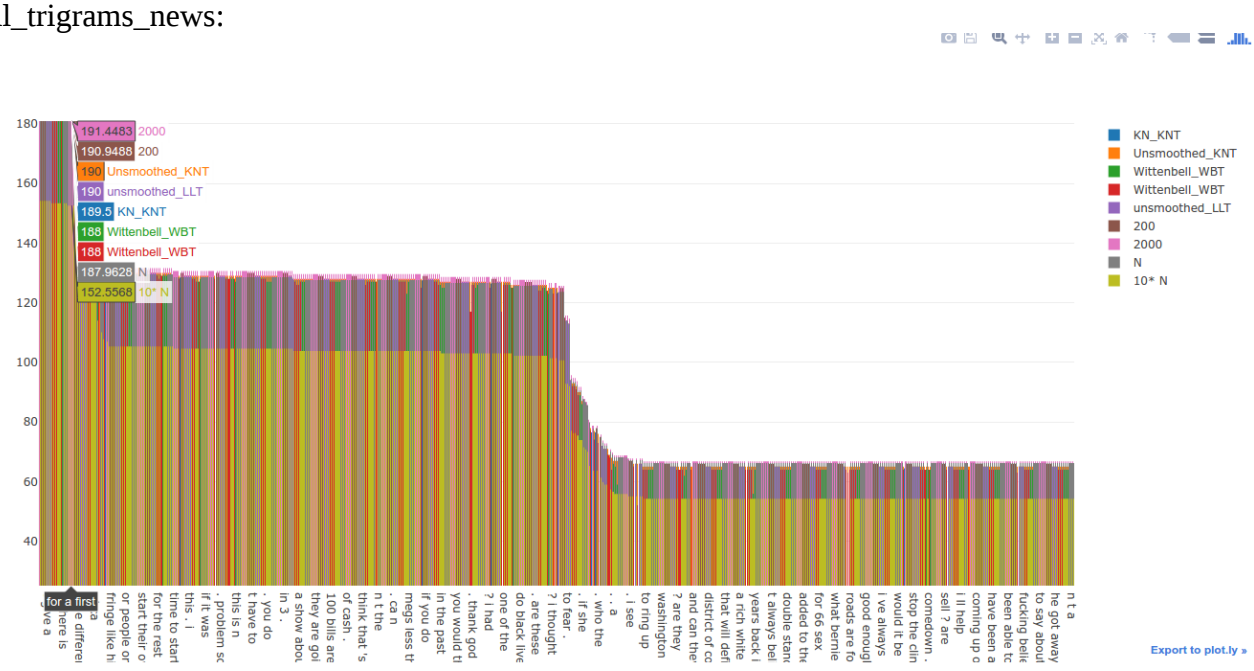
All_bigrams_news:



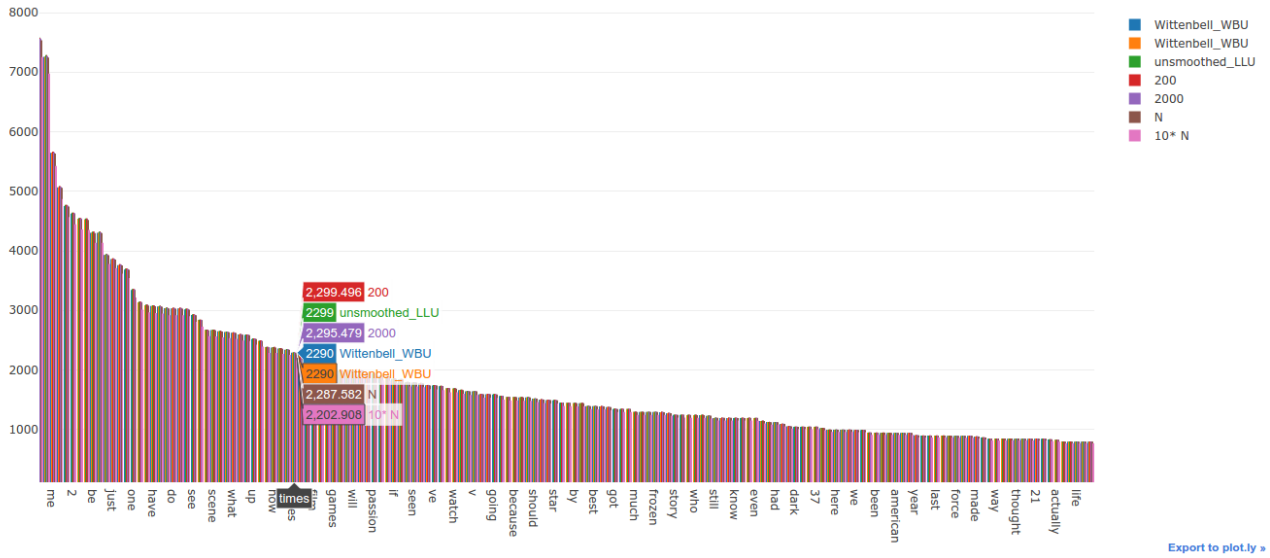
All_trigram_movies:



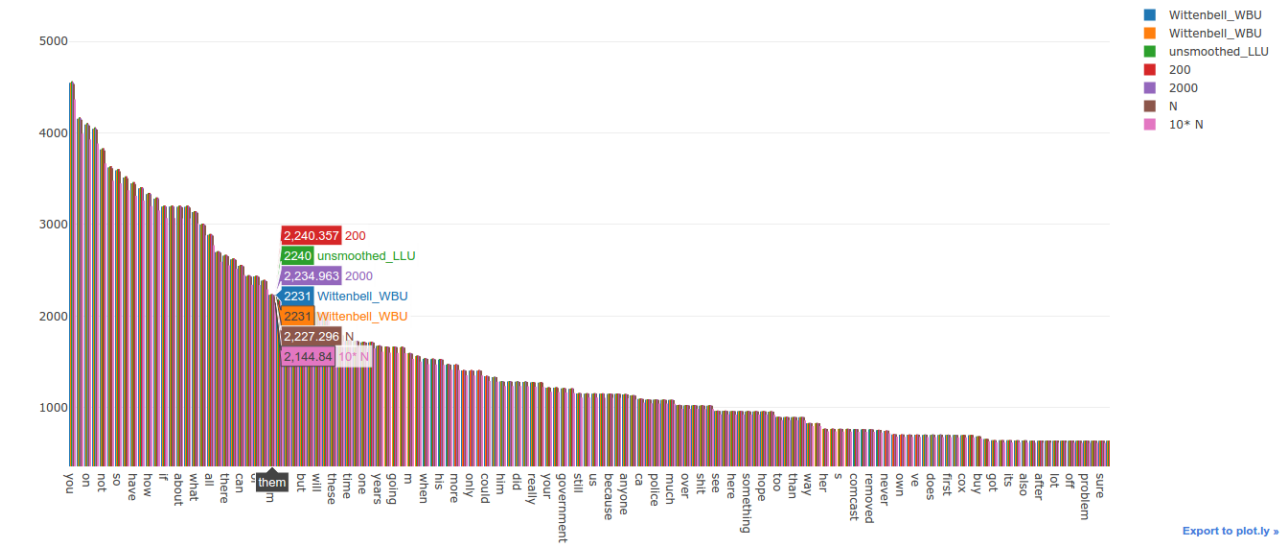
All_trigrams_news:



All_Unigrams_movies:

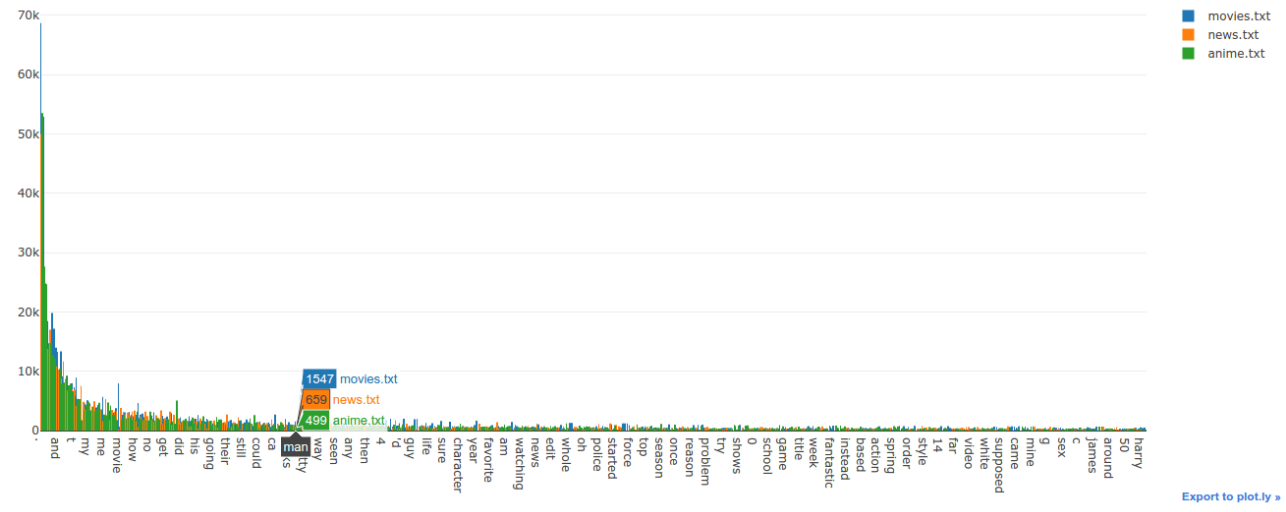


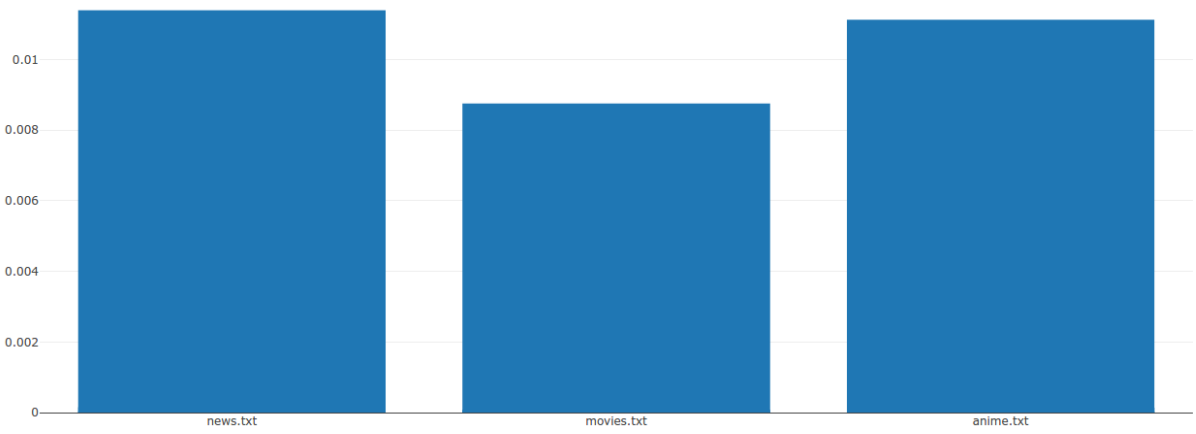
All_unigrams_news:



Naive bayes:

We plotted Zipf's curve using all the three datasets combined. We created a supervised learning mechanism where the classes are datasets. Here, when a new sentence is given, we would be able to predict from which dataset the sentence is coming from. For this we calculate Naive Bayes method to derive the probability of the sentence coming from each dataset using priori probability and class conditional probabilities..




0.01
0.008
0.006
0.004
0.002
0
news.txt
movies.txt
anime.txt