DOCUMENTATION

This module contains 5 files

1. parseTweets.py
This script uses the data and parse text from all the zip files by extracting it and storing it into database.txt

2. filterData.py
This script is used to remove all the strings of unwanted language and empty lines and store into final_data.txt

3. build_LM_Laplace.py
This scipt is used to build Language Model based on the Laplace smoothing distribution and the final entropy and perplexity values.

Entropy 2-gram	-	**0.12991264**
Perplexity 2-gram	-	**1.09422744**
Entropy 3-gram	-	**0.11156705**
Perplexity 3-gram	-	**1.08040113**

4. build_LM_GoodTuring.py
This scipt is used to build Language Model based on the Good Turing smoothing distribution and the final entropy and perplexity values.

Entropy 2-gram	-	**0.08265588**
Perplexity 2-gram	-	**1.05896571**
Entropy 3-gram	-	**0.09442491**
Perplexity 3-gram	-	**1.06763974**

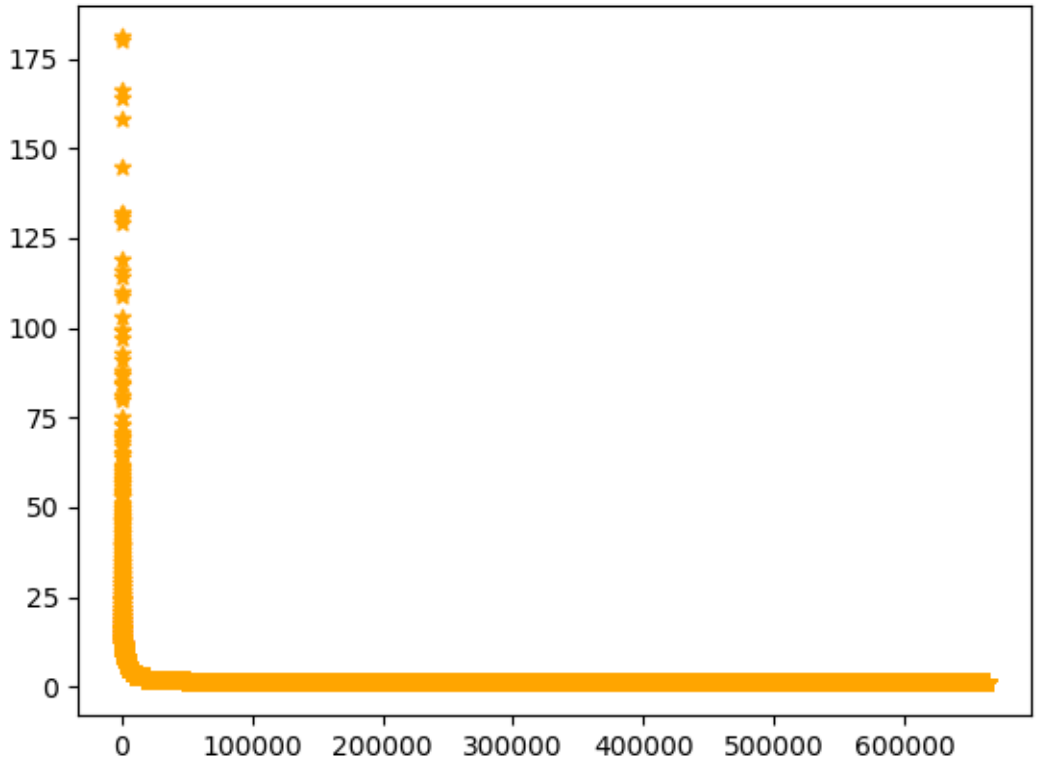
5. build_LM_KneserNey.py
This scipt is used to build Language Model based on the Good Turing smoothing distribution and the final entropy and perplexity values.

Entropy 3-gram	-	**0.05504654**
Perplexity 3-gram	-	**1.03889261**

Top 10 bigrams

('super', 'bowl')
('follow', 'back')
('happy', 'birthday')
('feel', 'like')
('youtube', 'video')
('last', 'night')
('bom', 'dia')
('mentionke', 'mentionke')
('justin', 'bieber')
('good', 'morning')

Frequency Plot Graph:



################ ---------- NOTE ---------- ################

- All the above scripts for finding perplexity values uses only 1 lakh tweets of the processed data.
  Command for making 1 lakh tweets:
  shuf final_data.txt | head -100000 > lakh.txt
- The top bigrams are made by removing stopwords and punctuations by using rank_graph.py .

##