COMPUTATIONAL LINGUISTICS – I

MINI PROJECT – 1

NAME: ADITYA RAGHUVANSHI
ROLL NUMBER: 2021114009

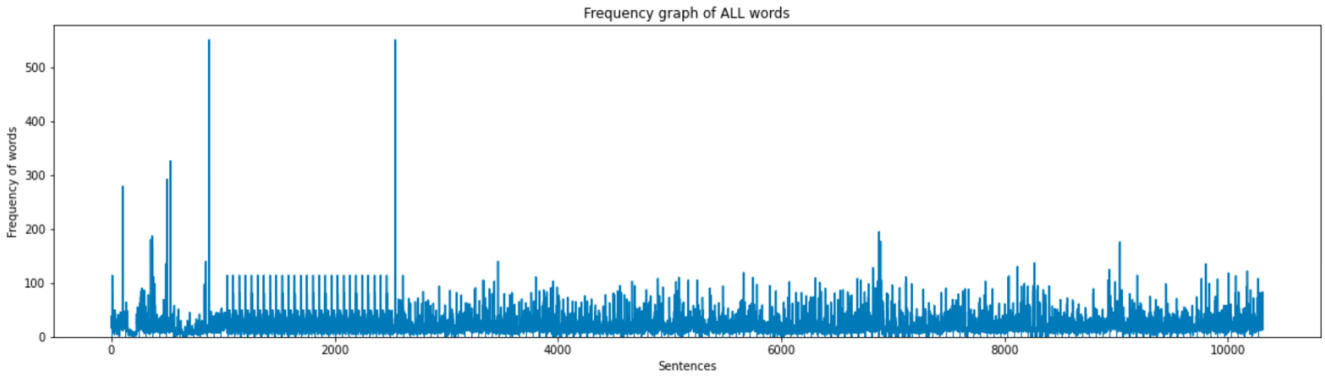
ENGLISH

A) frequency graphs and their analysis :

1) After sentence tokenization (using NLTK), I passed that data to a function which gave me the word count of each sentence and then this is the graph of comparison of words per sentence in corpus.

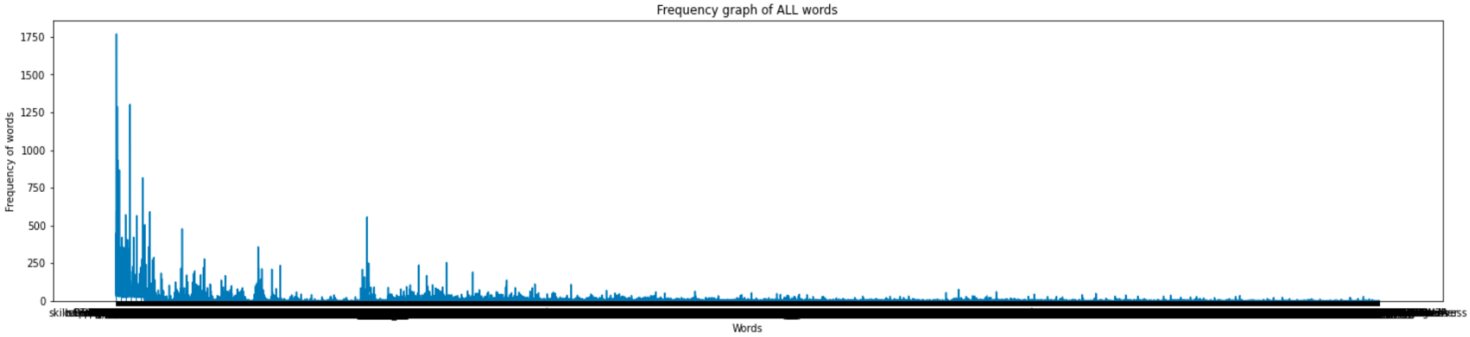
   Over here we can see that some sentences are having more that 100 words which are technically not possible, so my analysis from the tokenized data is that while tokenizing the data tokenizer applied a rule to break tokenize where a word is ended by a full stop attached to the word(excluding exceptions like dr. etc) and in corpus there were sentences like

   I am Aditya.my age is 18. → the problem here is that this tokenizer will count this a single sentence whereas the correct count is 2.

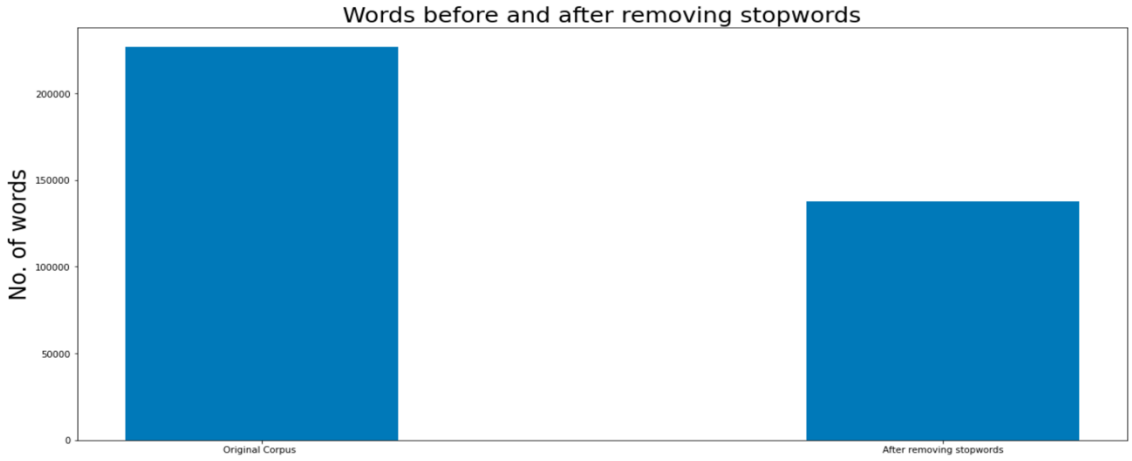


2) This graph shows the no. of times each word appeared in the corpus.

for example, this is the part of the output I used to make the graph after analyzing over 2.2 lakh words.



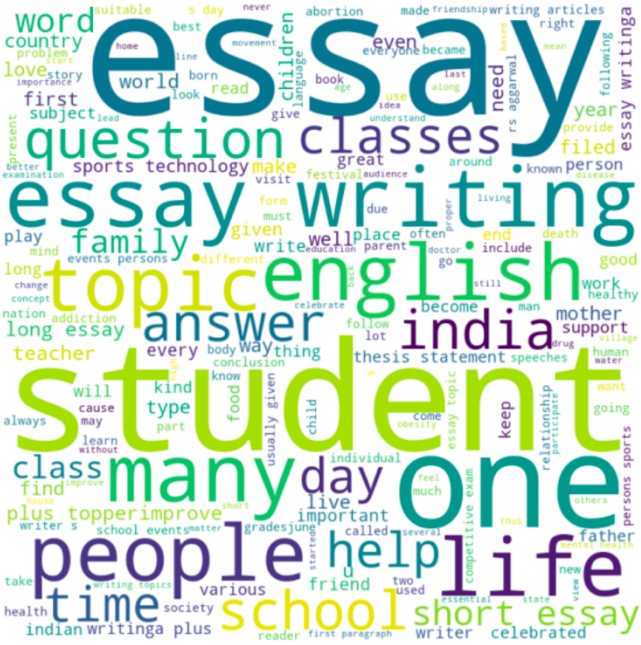
3) This graph shows the comparison between word count in the corpus before and after the removal of stopwords. This graph can clearly show that stopwords consists of approx. more than 33% of the corpus word count.



B. OVERVIEW AND ALGORITHM USED TO GENERATE WORD CLOUD.

This is the portion of my sorted word output from my corpus. Here we can see that after approximately 50 words the rate of decrease of frequency is almost 0 , i.e. the frequency is almost same of two consecutive words and so in the wordcloud the the sizes of words would then start becoming almost same.

```
[('Essay', 1769), ('The', 1302), ('essay', 1287), ('students', 929), ('also', 867), ('I', 815), ('English', 595), ('He', 590), ('many', 571),
('one', 565), ('people', 556), ('essays', 523), ('It', 505), ('life', 478), ('1', 449), ('10', 447), ('school', 429), ('India', 422), ('us', 422),
('2', 406), ('In', 406), ('classes', 402), ('writing', 381), ('events', 380), ('Writing', 374), ('3', 372), ('Class', 358), ('day', 358), ('topic',
355), ('like', 355), ('first', 353), ('long', 346), ('words', 343), ('8', 340), ('4', 339), ('A', 336), ('Plus', 328), ('9', 328), ('7', 328),
('This', 312), ('short', 303), ('5', 298), ('time', 294), ('given', 288), ('would', 278), ('family', 278), ('TopperImprove', 275), ('help', 273),
('writer', 264), ('class', 263), ('find', 262), ('Under', 258), ('person', 254), ('6', 251), ('every', 251), ('sports', 251), ('make', 249),
('main', 244), ('food', 243), ('good', 241), ('They', 238), ('Day', 235), ('helps', 230), ('different', 222), ('What', 222), ('Essays', 220),
('children', 220), ('articles', 219), ('even', 217), ('get', 216), ('2020', 216), ('technology', 216), ('We', 213), ('My', 212), ('world', 212),
('topics', 211), ('love', 210), ('country', 209), ('way', 204), ('need', 202), ('best', 201), ('body', 198), ('well', 198), ('must', 197),
('various', 195), ('persons', 192), ('health', 191), ('There', 188), ('important', 185), ('competitive', 184), ('type', 183), ('write', 180),
('WritingA', 178), ('year', 178), ('work', 177), ('great', 173), ('Indian', 171), ('Solutions', 168), ('healthy', 168), ('usually', 167), ('She',
159), ('part', 152), ('right', 150), ('mother', 150), ('How', 148), ('Aggarwal', 148), ('teachers', 147), ('take', 147), ('things', 147), ('lines',
145), ('ICSE', 144), ('types', 143), ('always', 143), ('friends', 142), ('Short', 138), ('two', 137), ('celebrated', 137), ('basic', 134), ('place',
132), ('play', 130), ('around', 130), ('end', 129), ('years', 127), ('speeches', 126), ('points', 126), ('Topics', 125), ('If', 125), ('thesis',
125), ('made', 125), ('read', 124), ('150', 124), ('know', 123), ('500', 123), ('form', 122), ('keep', 122), ('As', 121), ('become', 121), ('For',
119), ('To', 119), ('paragraph', 119), ('better', 119), ('lives', 119), ('Long', 118), ('writers', 117), ('But', 117), ('provided', 116), ('etc',
116), ('needs', 113), ('Statement', 113), ('physical', 113), ('education', 112), ('teacher', 112), ('death', 112), ('subject', 111), ('used', 111),
```

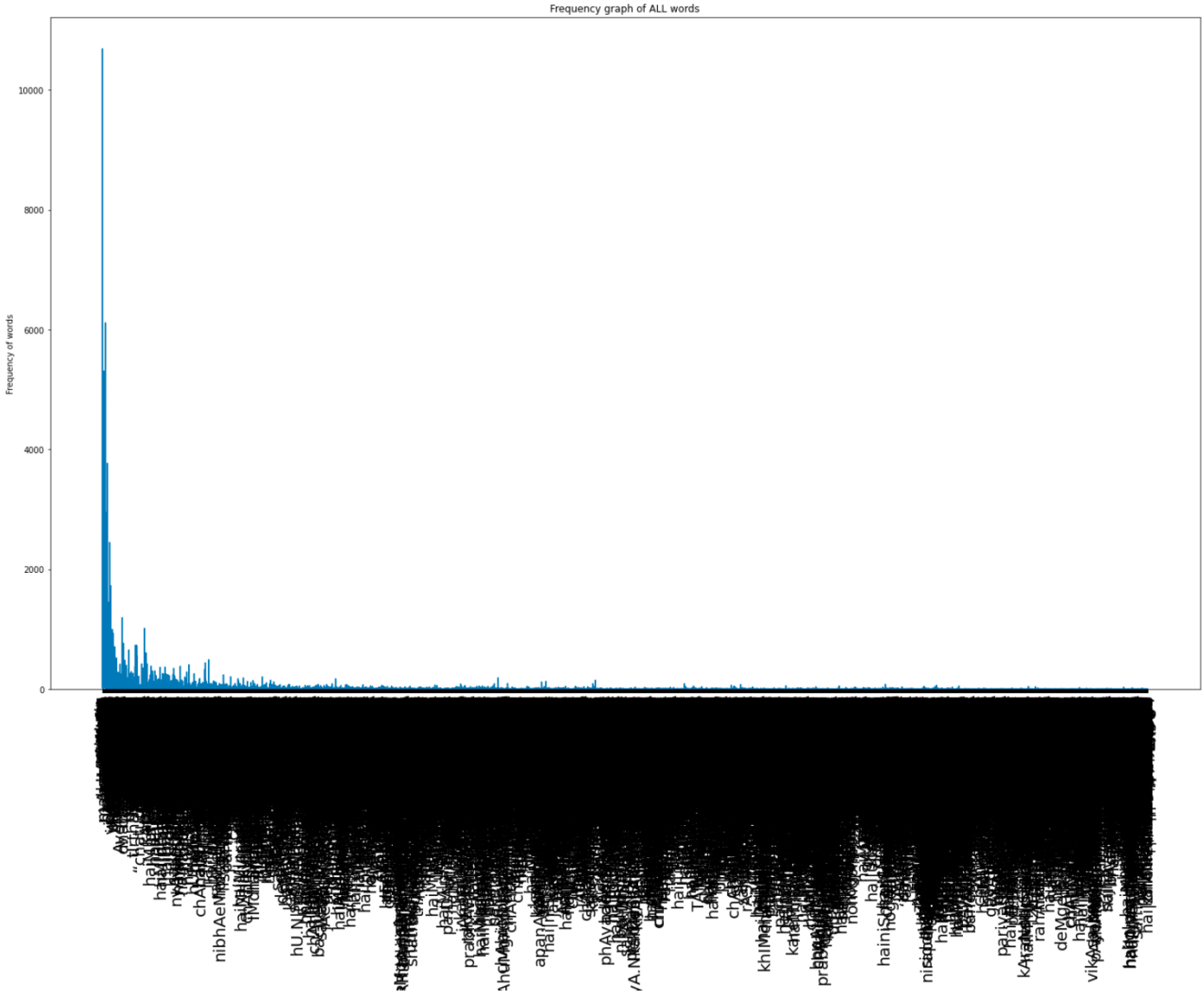


overview of algo :

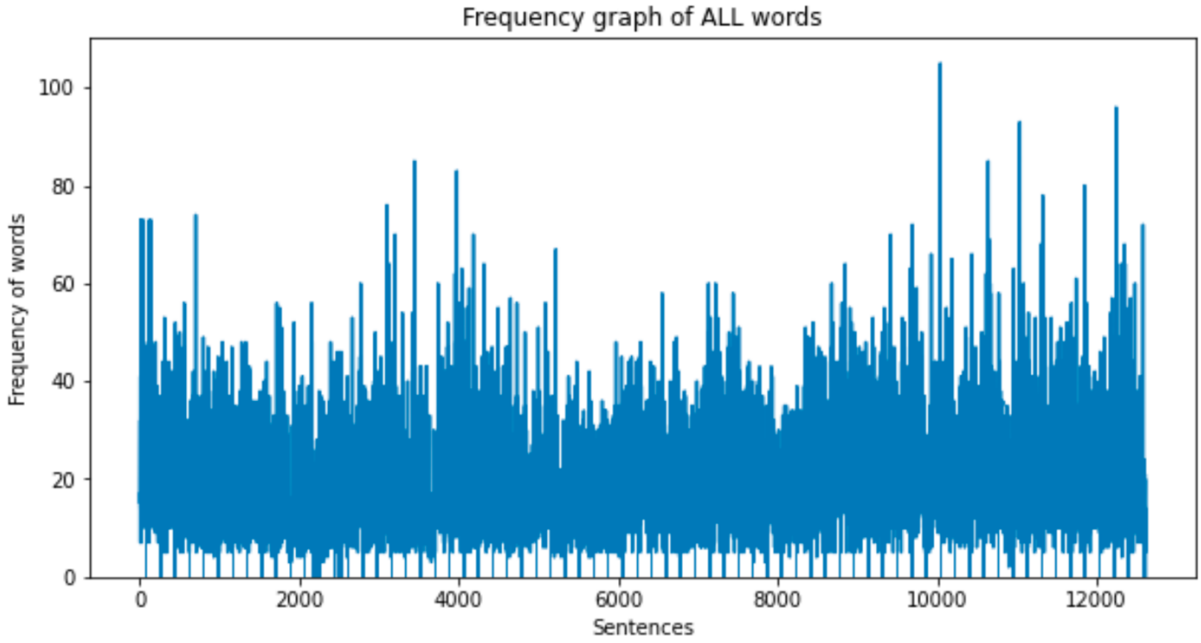
now to make this wordcloud I used the wordcloud , matplotlib and pandas modules . I gave the corpus as an input to the wordcloud function which then calculated the importance of words and respectively decided the fonts

HINDI

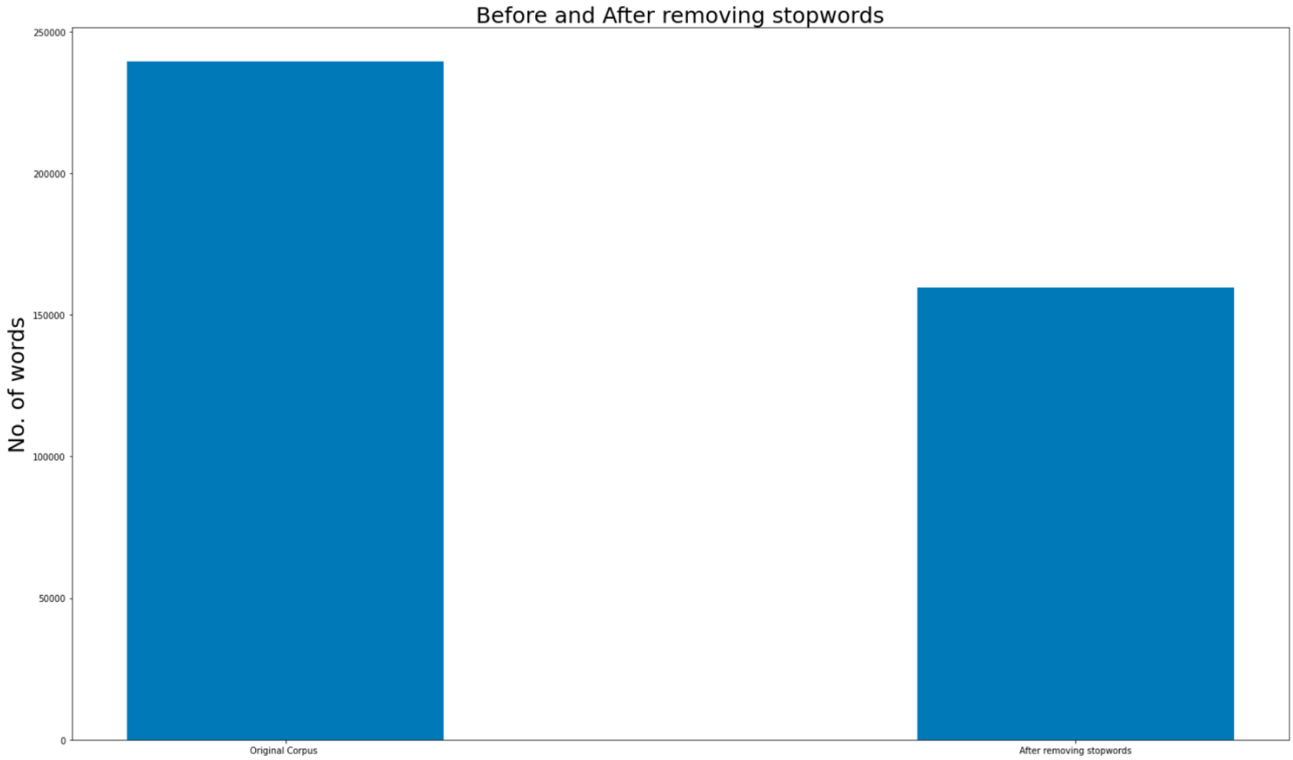
A) frequency graphs and their analysis :



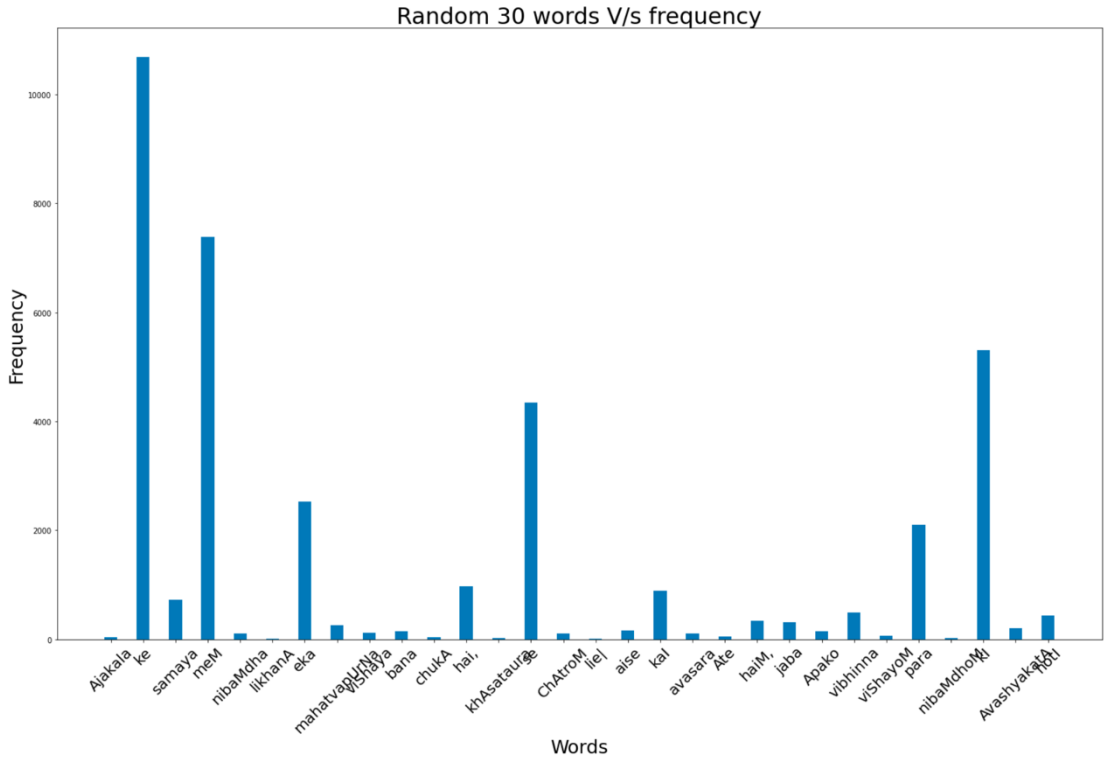
This graph shows the no. of times each word appeared in the corpus. And here we can see that the word `'के': 10687` has been used the maximum number of times.



This graph shows the distribution of words over 12000+ sentences, here in this graph we can conclude that the avg length of sentences is around 40 words. And the exceptional 5-10 cases in 12000 cases where the length of sentences is going above 60 might be due to data mistakes in source websites.



In this graph, we are comparing the word count of corpus before and after the removal of stopwords. Here also after seeing the graph we can say that about one third of the corpus is made of stopwords.



This graph shows 30 randomly picked words over the corpus and their frequency. Here we have converted pure Hindi script to romanised form using the indic-transliteration module.

In my wordcloud I have again preffered to include approximately 50 words due to the reason that after a point the count of words become almost same.

Overview of algo :

now to make this wordcloud I used the wordcloud , matplotlib and pandas modules . I gave the corpus as an input to the wordcloud function which then calculated the importance of words and respectively decided the fonts. Here to execute the output in Hindi script I have also uploaded a devnagri front script in my zip file so and added its path to font_path in my code.