CS-565 INTELLIGENT SYSTEMS AND INTERFACES

ASSIGNMENT - 1

-MAYANK WADHWANI (170101038)

Link for Google Colab Notebook (contains all the codes): https://bit.ly/34aX2Qq
Link for Google Drive that contains all the results obtained (all files): https://bit.ly/3n8GtNQ

TASK 1.3.1

For both the languages, different approaches in different tools were explores. For example, for English, NLTK supports many different types of tokenizers, following is an image from geeksforgeeks to show different types of tokenizers supported. This will be discussed briefly later.

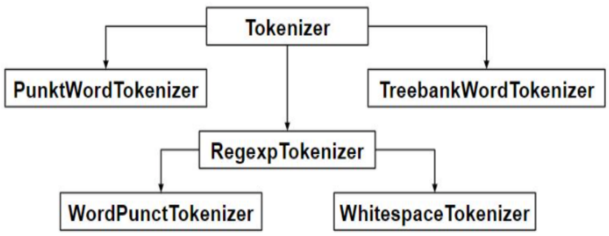


Figure 1 Different types of tokenizers supported in NLTK

For the task, the tools used are:

For English,

- NLTK
- spaCy

For Hindi,

- indicNLP
- Stanford NLP

As mentioned above, we observed different methods for sentence segmentation and word tokenization the corpora, for NLTK, we saw two different methods, WordPunctTokenizer and TreebankTokenizer. They differ in implementations as if there is a clitic in the text, like **let's**, then WordPunctTokenizer would break it into three tokens namely let, ' and s whereas Treebank Tokenizer breaks it into only 2 tokens ie. let and 's. For spaCy, we used the default tokenizer where we passed our sentence and got a list of tokens as an output.

A notable distinction which was observed when outputs of spaCy and NLTK were compared was that double inverted quotes in NLTK is written as two single quotes. Below is a image to display the same: (Here one token is written in one line)

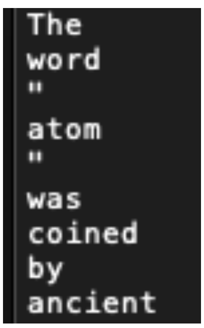


Figure 3 Double Quotes in spaCy

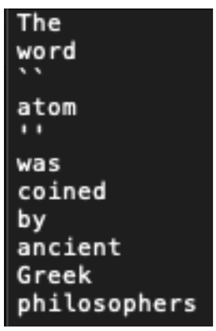


Figure 2 Double qoutes in NLTK TreeBank Tokenizer

Now as far as different methods of a tool are concerned, as mentioned above, there were some changes that were observed between the 2 tokenizers. For instance, if there is a ratio involved in the text, for example, 3:4, then The WorkPunct Tokenizer breaks this into three separate tokens whereas the TreeBankTokenizer keeps it as one token itself. Following is an image to support the point:

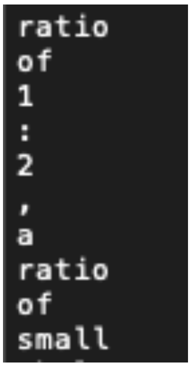


Figure 4 WordPunct Tokenizer NLTK

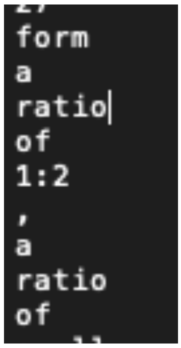


Figure 5 TreeBank Tokenizer NLTK

Similar things were observed in the tools for Hindi as well (indicNLP and Stanford NLP) We now present some results that were obtained for all the tools for ngrams, specifically, for each tool, we present the following:

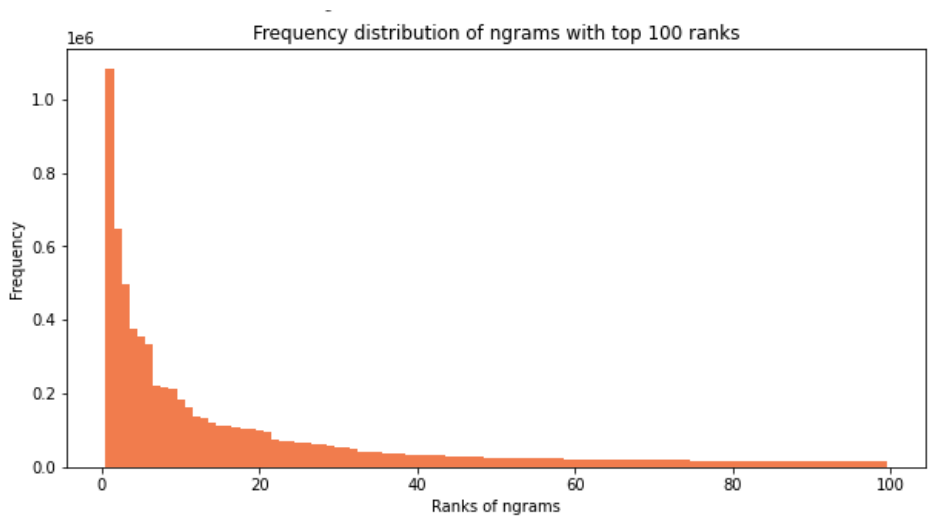
- A figure of a few ngrams having the highest frequencies along with the frequency of the ngram also.
- A plot of frequency vs rank. We first find the frequencies of all ngrams. If n=1, we will find all frequencies of unigrams, then we sort them in descending order and assign ranks to each ngram. We then display the plot of frequency vs rank for the top 100 ranks of the corpus.
- Word frequency vs frequency of frequencies. We present the frequency of frequencies for each word frequency, ie. if the word frequency is taken to be 1, then how many words are there which occur 1 time. Similary, if the word frequency is 100, then how many words are there which occur 100 times. Needless to say, the words with word frequency 100 will have lower rank than a word with word frequency 1.
- The most frequent ngram and the least frequent ngram

ENGLISH

NLTK

UNIGRAM

Word	Frequency
the	1083392
of	646178
and	498233
in	377175
to	354649
a	333421
''	222661
``	216424
was	210600
The	185104
is	164328
as	135557
for	134955
with	122534
by	111722
that	111073
's	105968
were	102616
%	101860
on	100558
from	94024
or	74671
his	72322
at	70592
In	66696
are	65124
an	63776

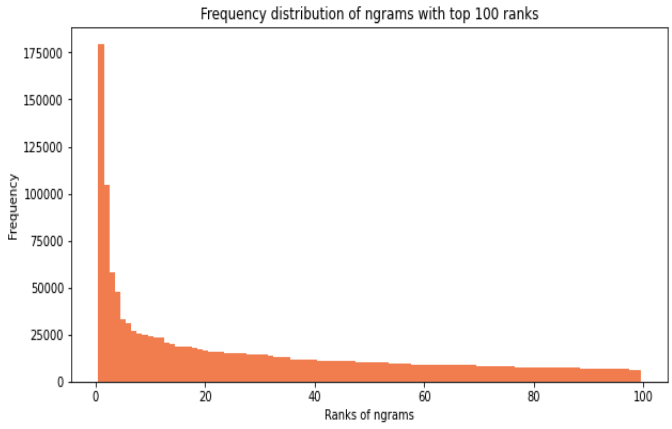


Word Frequency	Frequency of frequency
1	212320
2	52139
3	24564
4	15038
5	10194
6	7532
7	6047
8	4876
9	3945
10	3335
11-50	34842
51-100	8054
>100	12864

```
Most frequent word: the with frequency of 1083392
Least frequent word: '++ with frequency of 1
```

BIGRAM

Word	Frequency
of the	179482
in the	104824
to the	58259
and the	47472
on the	33031
for the	30720
by the	26707
% of	25277
at the	24911
with the	24124
as a	23184
from the	23096
is a	20419
as the	19809
to be	18912
the ``	18859
of a	18662
the city	18081
% from	17365
'' and	16693
in a	15810
`` The	15732
such as	15656
that the	15265
In the	15161
for a	14982
is the	14873

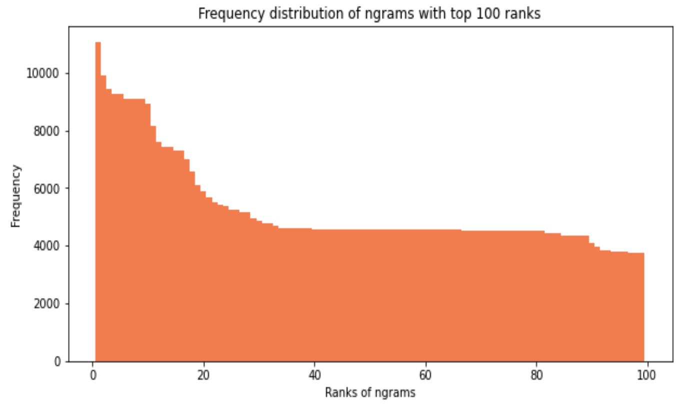


Most frequent word: of the with frequency of 179482
Least frequent word: ! # with frequency of 1

Word Frequency	Frequency of frequency
1	3409865
2	529750
3	209409
4	112977
5	70679
6	49134
7	35815
8	27606
9	21676
10	17947
11-50	134408
51-100	16696
>100	13857

TRIGRAM

Word	Frequency
the age of	11064
under the age	9928
the United States	9454
age of 18	9262
years of age	9247
or older The	9093
of age or	9092
age or older	9090
65 years of	9088
% of the	8909
of the population	8166
one of the	7600
For every 100	7447
every 100 females	7442
income for a	7315
median income for	7308
'' and ``	7000
as well as	6585
the city was	6118
age 18 and	5908
% of those	5670
the township was	5525
per square mile	5425
makeup of the	5391



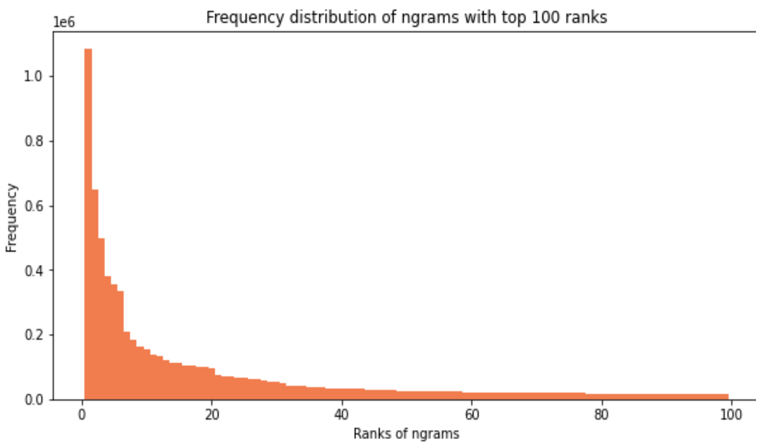
Most frequent word: the age of with frequency of 11064
Least frequent word: ! ! ... with frequency of 1

Word Frequency	Frequency of frequency
1	9621843
2	745341
3	229978
4	108810
5	62720
6	40925
7	28476
8	21204
9	16083
10	12573
11-50	82757
51-100	7308
>100	4442

spaCy

UNIGRAM

Word	Frequency
the	1084514
of	646899
and	499200
in	379038
to	356072
a	333780
was	210706
The	185385
is	164424
-	152875
as	135717
for	135246
with	122617
by	112168
that	111213
's	105982
were	102675
%	101769
on	101415
from	94084
or	74791
his	72411
at	70784

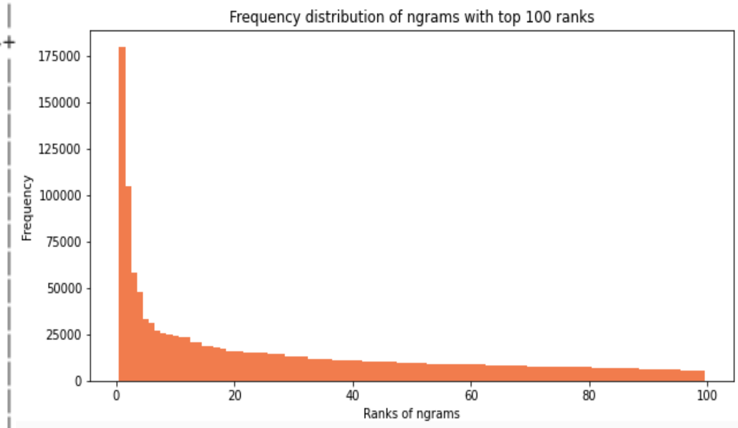


Word Frequency	Frequency of frequency
1	163807
2	44758
3	21885
4	13695
5	9364
6	6970
7	5652
8	4583
9	3758
10	3137
11-50	33841
51-100	8057
>100	13090

Most frequent word: the with frequency of 1084514
Least frequent word: "(1997 with frequency of 1

BIGRAM

Word	Frequency
of the	179647
in the	104892
to the	58357
and the	47617
on the	33071
for the	30761
by the	26741
% of	25274
at the	24935
with the	24155
as a	23613
from the	23127
is a	20490
as the	20424
to be	18933
of a	18729
the city	18120
% from	17365
in a	15844

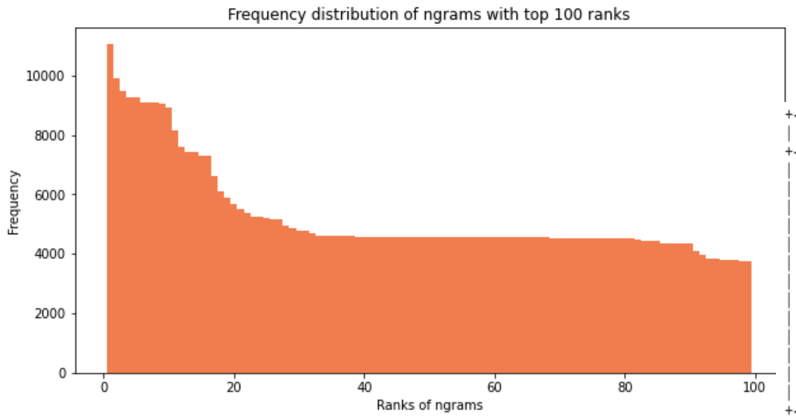


Word Frequency	Frequency of frequency
1	3411462
2	534263
3	210290
4	113275
5	71032
6	48954
7	35620
8	27656
9	21745
10	17746
11-50	134962
51-100	16882
>100	13959

Most frequent word: of the with frequency of 179647
Least frequent word: ! # with frequency of 1

TRIGRAM

Word	Frequency
the age of	11066
under the age	9928
the United States	9475
age of 18	9262
years of age	9248
or older The	9093
of age or	9092
age or older	9090
65 years of	9066
% of the	8906
of the population	8167
one of the	7621
For every 100	7447
every 100 females	7434
income for a	7315
median income for	7308
as well as	6605
the city was	6119
age 18 and	5902
% of those	5670
the township was	5525
makeup of the	5391
part of the	5264
in the city	5236



Word Frequency	Frequency of frequency
1	9653467
2	738758
3	226516
4	106974
5	61679
6	40041
7	27926
8	20776
9	15681
10	12351
11-50	81641
51-100	7209
>100	4329

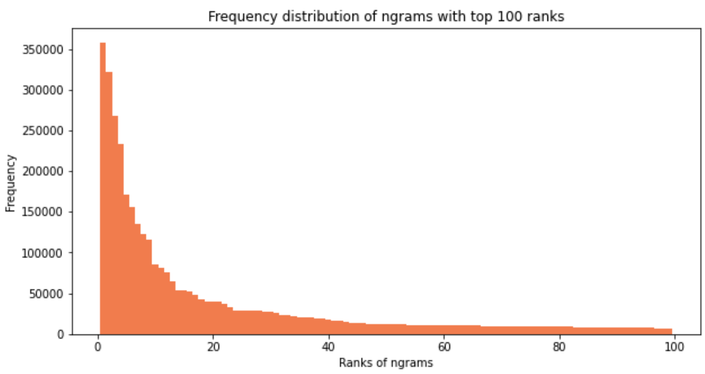
Most frequent word: the age of with frequency of 11066
Least frequent word: ! ... with frequency of 1

HINDI

INDICNLP

UNIGRAM

Word	Frequency
के	357772
।	321495
में	268252
है	232894
की	171759
और	155952
से	134909
का	122937
को	115884
हैं	86054
एक	81729
–	75111
पर	65040
ने	54114
किया	53886
लिए	52266
भी	47616
था	43078
कि	40042
यह	39796
गया	39158
इस	36672
रूप	32214
जो	28768

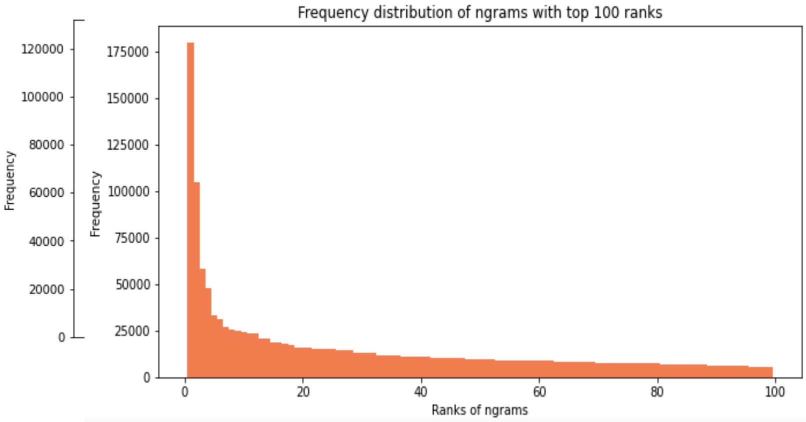


Word Frequency	Frequency of frequency
1	168451
2	37696
3	16651
4	9711
5	6707
6	4764
7	3748
8	2981
9	2444
10	2028
11-50	19461
51-100	4094
>100	6279

Most frequent word: के with frequency of 357772
Least frequent word: 0,0,0 with frequency of 1

BIGRAM

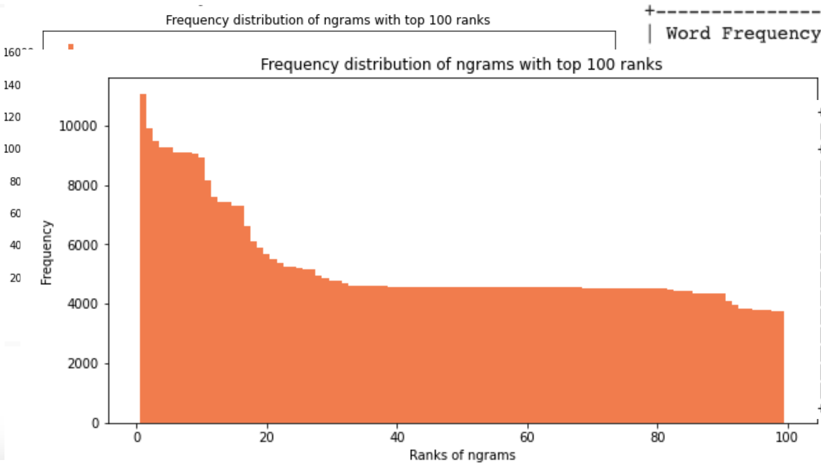
Word	Frequency
है ।	125716
के लिए	49338
हैं ।	48675
जाता है	25322
था ।	24339
के साथ	21025
रूप में	18063
के रूप	16826
है और	16603
है कि	15800
होता है	14949
किया गया	13970
। इस	13456
। यह	12585
थे ।	12288
करने के	11607
रूप से	11497
गया ।	11009
के बाद	10654
किया ।	10386
सकता है	9696
करता है	9545



Word Frequency	Frequency of frequency
1	3411462
2	534263
3	210290
4	113275
5	71032
6	48954
7	35620
8	27656
9	21745
10	17746
11-50	134962
51-100	16882
>100	13959

```
Most frequent word: of the with frequency of 179482
Least frequent word: ! # with frequency of 1
```

Word	Frequency
के रूप में	16517
जाता है ।	14178
करने के लिए	8006
होता है ।	7951
है । यह	6976
किया जाता है	6476
है । इस	5674
सकता है ।	5613
गया था ।	5548
गया है ।	5458
जा सकता है	5099
करता है ।	4955
किया गया था	4383
होती है ।	4325
करते हैं ।	4049
जाती है ।	4033
होते हैं ।	3578
किया गया है	3565
किया गया ।	3289
जाते हैं ।	2993
के लिए एक	2960



Word Frequency	Frequency of frequency
	4569559

Word Frequency	Frequency of frequency
1	9653467
2	738758
3	226516
4	106974
5	61679
6	40041
7	27926
8	20776
9	15681
10	12351
11-50	81641
51-100	7209
>100	4329

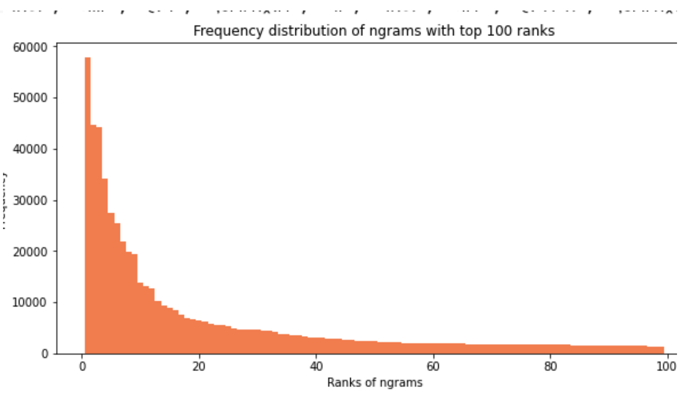
TRIGRAM

```
Most frequent word: the age of with frequency of 11066
Least frequent word: ! ... with frequency of 1
```

STANFORD NLP

UNIGRAM

Word	Frequency
के	57848
।	44626
में	44125
है	34244
की	27519
और	25401
से	21850
का	19811
को	19412
हैं	13856
है ।	13241
एक	12798
पर	10286
ने	9318
किया	8907
लिए	8553
भी	7624
था	6820
कि	6737
गया	6357
यह	6295
इस	5875
हैं ।	5573
के लिए	5483
रूप	5353
जो	4847
ही	4727
करने	4605
कर	4576

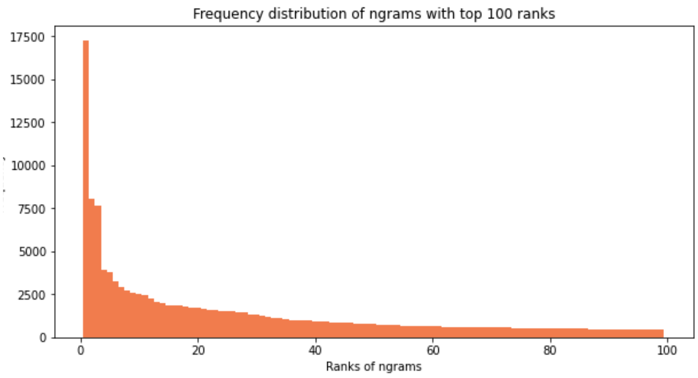


Word Frequency	Frequency of frequency
1	5537742
2	170170
3	41969
4	19876
5	11767
6	7444
7	5622
8	3892
9	2992
10	2319
11-50	16675
51-100	2116
>100	1901

```
Most frequent word: के with frequency of 57848
Least frequent word: ! ! आपके य अक्षर द्वारा with frequency of 1
```

BIGRAM

Word	Frequency
है ।	17264
के लिए	8054
हैं ।	7676
जाता है	3918
था ।	3785
के साथ	3241
रूप में	2950
के रूप	2739
है कि	2595
है और	2550
होता है	2439
किया गया	2255
। इस	2031
रूप से	1967
करने के	1877
थे ।	1847
के रूप में	1842
गया ।	1788
। यह	1752
के बाद	1697
किया ।	1634
जाता है ।	1583
सकता है	1578
गया है	1527
करता है	1504
है जो	1497

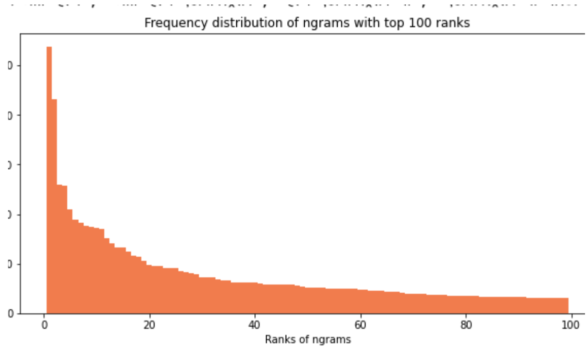


Most frequent word: है । with frequency of 17264
Least frequent word: ! ! आपके य अक्षर द्वारा जाने with frequency of 1

Word Frequency	Frequency of frequency
1	6541027
2	162613
3	38940
4	18334
5	10715
6	6860
7	4895
8	3511
9	2591
10	2094
11-50	13131
51-100	1144
>100	774

TRIGRAM

Word	Frequency
के रूप में	2690
जाता है ।	2161
करने के लिए	1301
होता है ।	1285
किया जाता है	1043
है । यह	946
सकता है ।	912
जा सकता है	885
है । इस	866
गया था ।	858
गया है ।	851
करता है ।	757
होती है ।	702
किया गया था	658
करते हैं ।	658
जाती है ।	621
किया गया है	581



Most frequent word: के रूप में with frequency of 2690
Least frequent word: ! ! !align=" with frequency of 1

Word Frequency	Frequency of frequency
1	7266210
2	121076
3	22146
4	9218
5	5241
6	3164
7	2375
8	1443
9	957
10	839
11-50	4311
51-100	282
>100	184

ZIPF'S LAW

A plot for log(rank) vs log(frequency) was plotted for both the corpora. It was found that Zipf's Law holds for intermediate values but for extreme ends starts to show some variations. Following is the curves for Zipf's law:

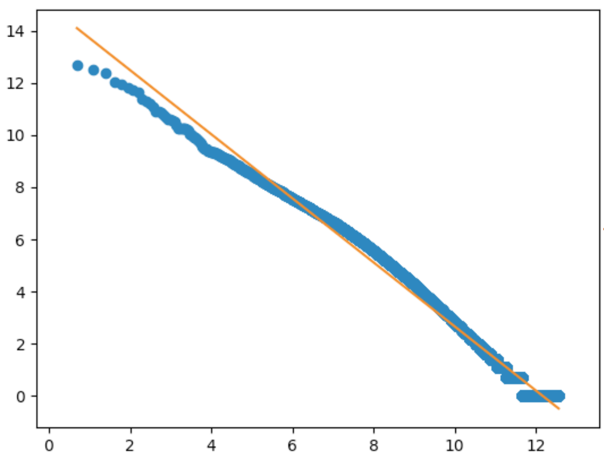


Figure 7 Zipf's law for Hindi Corpus

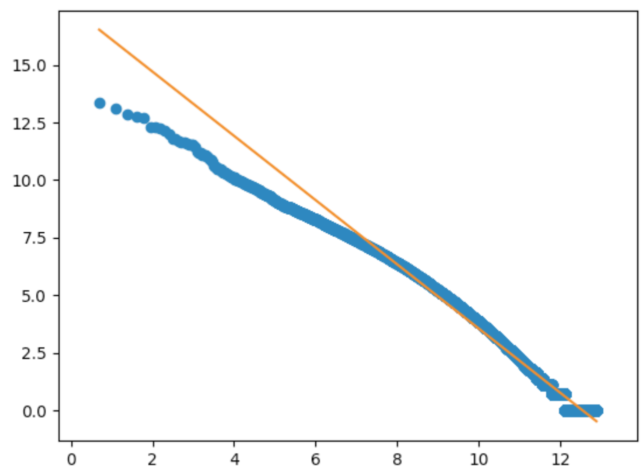


Figure 6 Zipf's Law for English Corpus

Here on x-axis we have plotted **log(rank)** and on y axis we have plotted **log(frequency).** Since the points can be approximated to a straight line with slope -1, we get log(rank) = -log(frequency) + c which implies log(rank) = -log(const*frequency) or

Rank = const/frequency. So we have verified that the rank is inversely proportional to the frequency.

Similar trends were observed for bigrams and trigrams also. The reader may see the results for bigrams and trigrams for both corpora from the following link: https://bit.ly/30oFJua.

Moreover, It was observed that the most frequent words is the same in different tools, whereas least frequent word changes. Maybe because of the implementation details. Like if we have a word let's in the corpora. And just a word s, then in one tokenizer, let's will break in let, ' , and s. So frequency of s will become 2 (as one s was already present in the corpora and one s came from let's) whereas in the other tokenizer, it will break in let and 's, so here the frequency of s and 's will remain 1.

In this way, the least frequent word can change depending on the tokenizer.

TASK 1.3.2

ENGLISH

The below images show the count of unigrams, bigrams, trigrams before stemming to cover 90%, 80%, 70% respectively of the total corpora.

```
Number of unigrams to cover 90% of corpora: 16647
Number of bigram to cover 80% of corpora: 1197120
Number of trigram to cover 70% of corpora: 5739159
```

HINDI

The below images show the count of unigrams, bigrams, trigrams before stemming to cover 90%, 80%, 70% respectively of the total corpora.

```
Number of unigrams to cover 90% of corpora: 13136
Number of bigram to cover 80% of corpora: 742340
Number of trigram to cover 70% of corpora: 2823591
```

AFTER STEMMING IS APPLIED:

NOTE: WE HAVE USED THE FOLLOWING LIBRARIES TO PERFORM STEMMING OF THE CORPORA:

For English, we used PorterStemmer which is a part of NLTK.

And for Hindi, we used HindiStemmer which is a part of SnowBallStemmer

ENGLISH

The below images show the count of unigrams, bigrams, trigrams before stemming to cover 90%, 80%, 70% respectively of the total corpora.

```
Number of unigrams to cover 90% of corpora: 6858
Number of bigram to cover 80% of corpora: 718226
Number of trigram to cover 70% of corpora: 5099952
```

HINDI

The below images show the count of unigrams, bigrams, trigrams before stemming to cover 90%, 80%, 70% respectively of the total corpora.

```
Number of unigrams to cover 90% of corpora: 13073
Number of bigram to cover 80% of corpora: 739366
Number of trigram to cover 70% of corpora: 2822695
```

Some notable results seen after and before stemming. As expected the number of ngrams required to cover a given percentage of the corpora got **drastically reduced**. This is due to the fact that the frequency of the ngrams increases after we perform stemming. For example if we had words perform, performs, and performed In our corpora. Then before stemming each would have counted as a separate token but after stemming all gets reduced to one token ie perform. So count of distinct words will increase leading to less number of most frequent wods required to cover the corpus.

We also notice as seen from the figures on the next page, that the frequency of frequencies for the different word frequencies will **change drastically**. The images 8 and 9 are from the English corpora. We notice that the number of words having frequency 1 is 145041 with stemming and 171968 with stemming.

This all arises due to the same fact. Earlier, if we had only three words perform, performed and performs, the number of words which appear once would have

Word	Frequency
क	825140
।	321495
म	282168
है	232956
और	155987
स	138583
ज	105168
कर	100504
हैं	86055
थ	84449
एक	81988
-	75111
न	65879
पर	65449
ल	60589
ह	60313
किय	56729
हु	35854
अप	35324

Figure 8 Hindi unigrams before stemming

Word	Frequency
क	825140
।	321495
म	282168
है	232956
और	155987
स	138583
ज	105168
कर	100504
हैं	86055
थ	84449
एक	81988
-	75111
न	65879
पर	65449
ल	60589
ह	60313
किय	56729
होत	37717
ह	35854

Figure 9 Hindi unigrams after stemming

been three but after applying stemming, it would become 0 as now perform occurs thrice. Here are the images to support the point.

Word Frequency	Frequency of frequency
1	171968
2	40578
3	18355
4	11147
5	7385
6	5327
7	4212
8	3256
9	2732
10	2312
11-50	23241
51-100	5317
>100	9361

Figure 11 Before Stemming

Word Frequency	Frequency of frequency
1	145041
2	31911
3	13928
4	8119
5	5464
6	3895
7	2980
8	2438
9	1967
10	1668
11-50	15374
51-100	3184
>100	5342

Figure 10 After stemming

TASK 1.3.3

A short note on heuristic used:

We first talk in brief about the English sentence segmentation and word tokenization used.

We first made a list of known abbreviations, like dr., mr., prof., etc., jr. and so on. The idea is to not break these abbreviations into two tokens. So dr. will remain one token. But if the period is used as a full stop, then we would be break into a separate token. So the text "I live in Delhi." will break into 5 tokens.Moreover, we also check if the end delimiters which are '.' ':' '?' '!' are in single quotes or not. So for instance if the corpora is->

"Where is Dr. Mehta going today?", the mother asked.

So this will be taken as one line whole. So if end delimiters like full stop, exclamation mark, etc appear inside quotes, we do not break that line.

Now as far as word tokenizer is concerned, we are given a line as an input and we are supposed to break it into tokens. So a simple heuristic as suggested in class is to break the tokens by space. So if we have the corpora "I live in Delhi", we would simply break this by spaces to get a total of 4 words. We improve this by adding conditions for delimiters like ',' or '.' or even '?' so if we encounter such a delimiter, we break it. We also cover the case of clitics, ie. if the word let's appears in the corpus, then we break this into only two tokens, one let and the other 's.

So these were some details about the custom sentence segmentation and word tokenization that was performed.

A similar approach was also followed for Hindi language also. A main difference here is that the end delimiter is changed from period (.) to pipe symbol (|). Rest major intricate details remains the same.

So below are the required counts of ngrams to cover the given percentage of the corpus for our custom model:

ENGLISH

The below images show the count of unigrams, bigrams, trigrams before stemming to cover 90%, 80%, 70% respectively of the total corpora.

```
Number of unigrams to cover 90% of corpora: 14074
Number of bigram to cover 80% of corpora: 1112569
Number of trigram to cover 70% of corpora: 5535610
```

HINDI

The below images show the count of unigrams, bigrams, trigrams before stemming to cover 90%, 80%, 70% respectively of the total corpora.

```
Number of unigrams to cover 90% of corpora: 16310
Number of bigram to cover 80% of corpora: 949644
Number of trigram to cover 70% of corpora: 3092818
```

These numbers are quite similar to what we got from the models of libraries such as NLTK and indicNLP.

For english, the number of ngrams required was observed to be less than the NLTK one. This may by due to the fact that we had converted the letters in lowercase(as suggested by slides) to get tokens. So now perform and Perform both will lead to the same token. The main demerit of this is that we will loose the context of the line if we lowercase the letter. But since, Part of speech tagging was not required in the question, we can afford that.

Even for Hindi, the numbers are not drastically different that suggests that the custom heuristic approach turned out to be pretty decent.

Likelihood Test

Link of Reference: https://stanford.io/36kilwe

Likelihood ratio test is used to find and predict if a given bigram is a collocation or not. It is simply a number that tells how much a bigram can be a collocation as compared to another bigram. So if the value is lets say 30 and 40. Then the bigram having the value 40 is more probable to be a collocation than the one having the value 30.

The main theory behind the ratio is based on a ratio of two hypothesis for a bigram **x y** :

- unigrams x and y are independent. ie. y will have no effect on whether or not x will follow it
- unigrams x and y are dependend ie. y will depend on whether x will follow it

Formally we can say for a bigram w^2 w^1 :

- Hypothesis 1. $P(w^2|w^1) = p = P(w^2|\neg w^1)$
- Hypothesis 2. $P(w^2|w^1) = p_1 \neq p_2 = P(w^2|\neg w^1)$

We find log(x) for this test where log(x) = log(L(H1)) / log (L (H2)) where H1 and H2 stands for the above hypotheses.

Where L is the binomial function x^k (1-x) ^ (n-k).

When we expand H1 and H2 and solve further the final equation that we get is as follow:

$$\text{Log(x)=} \quad \log L(c_{12}, c_1, p) + \log L(c_2 - c_{12}, N - c_1, p) - \log L(c_{12}, c_1, p_1) - \log L(c_2 - c_{12}, N - c_1, p_2)$$

Likelihood ratio is then defined as -2log(x).

We have taken our threshold value to be 50, so all bigrams which have the value of -2log(x) greater than 50 will be considered as a collocation. Below images show some collocations with the ratio value. And the total number of collocations along with the ratio of this number to the total bigrams in the corpus.

```
प्रसाद बिस्मिल 167.01392548625014
स्पोर्ट्स इलस्ट्रेटेड 166.50060745712577
डेमलर एजी 166.37948730155563
Rajasthan 331518 165.6034480448771
ज़ीनत अमान 165.03316500800128
संबंधी स्थिरांक 164.72877490207918
एक्सचेंज स्टॉक-एक्सचेंज 163.89479611530254
सेंट लुइस 163.75718400921022
फॉर वेंडेट्टा 163.05696162785088
नहीं मानते। 162.84003569143312
डीडीसी DDC 162.6612993298811
गतिजनक न्यूरॉन 162.6104815978748
ISO 9001: 162.37489996970686
द किलर्स 162.25499952961295
बामर लॉरी 162.02861775894993
एयर फोर्स 162.01819662727235
माइक पोर्टनॉय 161.70552817005733
& nbsp 161.28460797170465
खाप पंचायतों 161.25137660758878
लाइफटाइम अचीवमेंट 160.61281165435435
एडी ब्रॉक 160.33501532054254
च्यांग काई 160.10400155511832
मादक द्रव्यों 160.07209112747233
डिग्री फारेनहाइट 159.95052741873434
- रोहिलखण्ड 159.34452430848734
लीला भंसाली 159.34433753575132
चाइना ईस्टर्न 159.1310425118678
राष्ट्र महासभा 158.78122718508575
निर्माण करवाया। 158.64361958387883
```

Figure 12 Likelihood Ratio Test For Hindi

```
clark gable 143.10640444308387
kwisatz haderach 142.78195913699048
judi dench 142.78195913649677
la paz 142.55366244815593
lady godiva 142.51031243117444
market capitalization 142.437813856753
paw paw 142.30682649999915
golden horde 142.23088243834377
aromatic hydrocarbons 142.10545707389008
on sundays 142.0916640037358
is isomorphic 142.04869945830706
a mockingbird 142.01681556113823
san marino 141.8852610766474
wire wrap 141.81695935114715
de palma 141.8144216380975
del biaggio 141.77026886955548
honorary doctorates 141.76875390556532
britney spears 141.61432043815262
ventral tegmental 141.4839313552781
diego rivera 141.36994690280125
sun microsystems 141.32444081437157
ehud barak 141.32256845976383
genital mutilation 141.31671754022875
slab rollback 141.17052081995354
cerebral palsy 141.1657809270747
henry kissinger 141.16132668568065
river waveney 141.15309772880255
yoshitaka amano 140.92390186547357
```

Figure 13 Likelihood Ratio Test For English

```
Enter input file : hindi_words_custom.txt
NUMBER OF BIGRAMS: 7601232
NUMBER OF COLLOCATIONS: 2882
RATIO: 0.00037914906425695204
```

```
Enter input file : english_words_custom_heuristic_approach.txt
NUMBER OF BIGRAMS: 17103434
NUMBER OF COLLOCATIONS: 5554
RATIO: 0.00032473010975456743
```

The exact results can be seen from the following links:

Collocations for English corpus: https://bit.ly/3kWGEty

Collocations for Hindi corpus: https://bit.ly/2SbYDQo

TASK 1.3.4

We have used the morphological analyser "polyglot" for this task.

We first present a brief on the model of the analyser.

Link of Reference: https://bit.ly/2HCKfOY

The analyser uses the so called Morfessor Baseline method which used raw text as the training data in order to learn a model ie. a vocabulary of morphemes. A morph vocabulary also called as lexicon of morphs is created so that it is possible to form any word in the data by the concatenation of some morphemes. Each word in the data is then written as a sequence of morph pointers which points to the entries in the lexicon. The model makes use of the Minimum Description Length (MDL) principle. The segmentation procedure resembles text segmentation as no simplifying assumptions about number of morphemes per word are made.

Using the polyglot morphological analyser, we tried to get the morphemes for 5 random words from 100 most frequent and 100 least frequent words from our vocabulary. Here are the findings:

For 100 most frequent words:

```
We first attempt Morphological analysis of 5 random words from 100 most frequent words:
Random Chosen Word: below with a frequency of: 5991
['be', 'low']
Random Chosen Word: population with a frequency of: 24858
['popul', 'ation']
Random Chosen Word: species with a frequency of: 3918
['spec', 'ies']
Random Chosen Word: where with a frequency of: 13949
['where']
Random Chosen Word: Road with a frequency of: 1951
['Road']
```

So words like population are broken into 2 morphemes: 'popul' and 'ation'. Similarly we find morphemes for the less frequents words as well.

For 100 least frequent words:

```
Random Chosen Word: Dictyoptera with a frequency of: 1
['D', 'ic', 'ty', 'opt', 'er', 'a']
Random Chosen Word: Hackbridge with a frequency of: 1
['Hack', 'bridge']
Random Chosen Word: Derrymore with a frequency of: 1
['Der', 'ry', 'more']
Random Chosen Word: Desembarco with a frequency of: 1
['De', 's', 'emb', 'ar', 'co']
Random Chosen Word: Lawrason with a frequency of: 1
['Law', 'ra', 'son']
```

TASK 1.3.5

For this task, we chose the dictionary size to be 1,00,000 words and number of iterations to be 15,000.

Even though this produces fairly decents output, one can expect to get better results if these values are increased.

To illustrate this, we present the sub-tokens of an input word on running the algorithm for different number of iterations and different dictionary size. When the above values are used and the word population is given as an input, we get the following output (the file **english_bpe.txt** contains the vocabulary and frequency iterating 15000 times):

```
Enter input file : english_bpe.txt
Enter word for morphological analysis : population
population~
```

In a similar fashion, when we sub-tokenize input keeping the dictionary size to be 10,000 and number of iterations to be 10,000, we get the following output (the file **english_alt_bpe.txt** contains the vocabulary and frequency iterating 15000 times):

```
Enter input file : english_alt_bpe.txt
Enter word for morphological analysis : population
po p ul ation~
```

So we see the effect of iterations and dictionary size of input. We will get better and more meaningful sub-tokens on increasing the values. The sub-token population has a defined meaning but the sub-tokens po, ul, ation do not have a meaning as such.

We now present the 50 most frequent and least frequent sub-tokens for both the corpora along with the frequencies.

Figure 17 50 Most frequent sub-tokens for HINDI

```
Enter input file : hindi_wo
50 most frequent sub-tokens
के~ with frequency: 357961
।~ with frequency: 321495
में~ with frequency: 268398
है~ with frequency: 232950
की~ with frequency: 172617
और~ with frequency: 155952
से~ with frequency: 135131
का~ with frequency: 123668
को~ with frequency: 116053
हैं~ with frequency: 86054
एक~ with frequency: 81729
—~ with frequency: 75111
पर~ with frequency: 65311
ने~ with frequency: 56671
किया~ with frequency: 54020
लिए~ with frequency: 52266
भी~ with frequency: 47616
था~ with frequency: 43141
कि~ with frequency: 40160
यह~ with frequency: 39796
गया~ with frequency: 39158
इस~ with frequency: 36794
रूप~ with frequency: 32277
कर~ with frequency: 29655
ही~ with frequency: 29318
जो~ with frequency: 28828
जाता~ with frequency: 28744
करने~ with frequency: 28432
साथ~ with frequency: 28382
हो~ with frequency: 27698
नहीं~ with frequency: 26580
या~ with frequency: 23730
द्वारा~ with frequency: 22939
थे~ with frequency: 21446
तथा~ with frequency: 21108
अपने~ with frequency: 20317
बाद~ with frequency: 20311
तक~ with frequency: 19628
दिया~ with frequency: 18799
होता~ with frequency: 17555
थी~ with frequency: 16918
वह~ with frequency: 16230
कुछ~ with frequency: 14794
हुआ~ with frequency: 14150
करते~ with frequency: 13759
वे~ with frequency: 13469
तो~ with frequency: 12805
हुए~ with frequency: 12786
समय~ with frequency: 12760
जा~ with frequency: 12098
```

Figure 16 50 Least frequent sub-tokens for HINDI

```
पक् with frequency: 55
पहुं with frequency: 55
पोलियन~ with frequency: 55
फ़रा with frequency: 55
बलडन~ with frequency: 55
मिन with frequency: 55
मेह with frequency: 55
मॉड with frequency: 55
वं with frequency: 55
बन with frequency: 54
मही with frequency: 54
माउ with frequency: 54
र्च~ with frequency: 54
वरि with frequency: 54
शुल् with frequency: 54
संख्या with frequency: 54
सप्ता with frequency: 54
सर्वि with frequency: 54
स्थू with frequency: 54
अश् with frequency: 55
आग with frequency: 55
कॉन with frequency: 55
क्ट~ with frequency: 55
गण with frequency: 55
ङ with frequency: 55
चों~ with frequency: 55
छा with frequency: 55
झु with frequency: 55
द्धि~ with frequency: 55
नज with frequency: 55
नर् with frequency: 55
समो with frequency: 53
सार्वभौ with frequency: 53
सिल् with frequency: 53
सूक्ष्म with frequency: 53
सेना with frequency: 53
ऊ~ with frequency: 54
कड़ों~ with frequency: 54
किन with frequency: 54
चतु with frequency: 54
चार्य~ with frequency: 54
जनों~ with frequency: 54
जोखि with frequency: 54
डिवी with frequency: 54
त्~ with frequency: 54
दिखा with frequency: 54
नाग with frequency: 54
निषे with frequency: 54
प्रभा with frequency: 54
बन with frequency: 54
मही with frequency: 54
```

Figure 15 50 Most frequent sub-tokens for ENGLISH

```
50 most frequent sub-tokens:
the~ with frequency: 1084015
of~ with frequency: 646937
and~ with frequency: 500600
in~ with frequency: 387202
to~ with frequency: 357901
a~ with frequency: 356203
''~ with frequency: 222661
``~ with frequency: 216424
was~ with frequency: 210637
The~ with frequency: 185313
is~ with frequency: 170623
as~ with frequency: 143866
for~ with frequency: 134991
with~ with frequency: 122578
by~ with frequency: 114391
on~ with frequency: 112808
that~ with frequency: 111109
's~ with frequency: 105968
were~ with frequency: 102628
%~ with frequency: 101860
from~ with frequency: 94043
or~ with frequency: 81274
an~ with frequency: 75113
at~ with frequency: 74676
his~ with frequency: 72391
are~ with frequency: 67189
In~ with frequency: 66747
which~ with frequency: 63517
had~ with frequency: 59076
it~ with frequency: 57173
be~ with frequency: 55517
he~ with frequency: 49487
age~ with frequency: 41230
has~ with frequency: 40547
not~ with frequency: 39924
also~ with frequency: 39088
have~ with frequency: 36018
who~ with frequency: 35636
;~ with frequency: 34681
their~ with frequency: 34599
its~ with frequency: 33401
but~ with frequency: 32765
one~ with frequency: 32654
first~ with frequency: 31130
this~ with frequency: 30508
years~ with frequency: 28591
been~ with frequency: 28126
other~ with frequency: 27386
all~ with frequency: 26920
two~ with frequency: 26215
```

Figure 14 50 Most least sub-tokens for ENGLISH

```
Poi with frequency: 4
Sansk with frequency: 4
Transpor with frequency: 4
^ with frequency: 4
acco~ with frequency: 4
astern~ with frequency: 4
chools~ with frequency: 4
consi with frequency: 4
contro with frequency: 4
icip with frequency: 4
incor with frequency: 4
inte with frequency: 4
joint with frequency: 4
ketball~ with frequency: 4
oppon with frequency: 4
overn with frequency: 4
provinc with frequency: 4
responsib with frequency: 4
soldi with frequency: 4
teach with frequency: 4
territ with frequency: 4
psych with frequency: 4
surg with frequency: 4
adop with frequency: 4
amin~ with frequency: 4
arity~ with frequency: 4
atistics~ with frequency: 4
ced with frequency: 4
isms~ with frequency: 4
r.~ with frequency: 4
recogni with frequency: 4
sey~ with frequency: 4
spec with frequency: 4
struc with frequency: 4
tured~ with frequency: 4
ually~ with frequency: 4
woo with frequency: 4
Soc with frequency: 4
fic with frequency: 4
pra with frequency: 4
stly~ with frequency: 4
ü with frequency: 4
0 with frequency: 4
Chanc with frequency: 4
0b with frequency: 4
alized~ with frequency:4
aul~ with frequency: 4
circul with frequency: 4
disa with frequency: 4
rol with frequency: 4
separ with frequency: 4
tus~ with frequency: 4
univers with frequency: 4
Or with frequency: 4
RA with frequency: 4
absol with frequency: 4
```

We now present the results of running the algorithm on unknown words from both the corpora.

As we can see, there are some words which were sub-tokenized into one complete word, like व्यक्त. Also words like disinterested were broken appropriately to get dis as one of the sub-tokens.

Figure 19 BPE on unknown words HINDI

```
Enter input file : hindi_bpe.txt
Enter word (press e to exit): सुधारना
सुधार ना~
Enter word (press e to exit): इंतज़ार
इं त ज़ा र~
Enter word (press e to exit): खुबसूरत
खु ब सूरत~
Enter word (press e to exit): अचानक
अ चा नक~
```

Figure 18 BPE on unknown words ENGLISH

```
Enter input file : english_bpe.txt
Enter word (press e to exit): ironic
ir onic~
Enter word (press e to exit): unabashed
un ab ashed~
Enter word (press e to exit): nonplussed
non pl us sed~
Enter word (press e to exit): disinterested
dis in te re sted~
```

```
Enter word (press e to exit): व्यक्त
व्यक्त~
Enter word (press e to exit): सुझाव
सुझाव~
Enter word (press e to exit): गिराना
गि रा ना~
Enter word (press e to exit): शिकायत
शिकायत~
Enter word (press e to exit): लापरवाह
ला पर वाह~
Enter word (press e to exit): संतुष्ट
संतुष्ट~
Enter word (press e to exit): इस्तेमाल
इ स् ते माल~
Enter word (press e to exit): e
```

```
Enter word (press e to exit): maneuver
maneu ver~
Enter word (press e to exit): contemplate
cont em plate~
Enter word (press e to exit): lieutenant
li e ut en ant~
Enter word (press e to exit): colonel
colon el~
Enter word (press e to exit): enormity
enorm ity~
Enter word (press e to exit): super
su per~
Enter word (press e to exit): scramble
sc ram ble~
Enter word (press e to exit): e
```

We now try to contrast this algorithm with Morphological analysis as done in task 1.3.4

Here are the results:

```
Enter input file : english_bpe.txt
Enter word (press e to exit): below
below~
Enter word (press e to exit): population
population~
Enter word (press e to exit): species
species~
Enter word (press e to exit): where
where~
Enter word (press e to exit): road
road~
Enter word (press e to exit): Dictyopetra
Dic ty op e tra~
Enter word (press e to exit): Hackbridge
H ack bridge~
Enter word (press e to exit): Derrymore
D err y more~
Enter word (press e to exit): Desembarco
Des embar co~
Enter word (press e to exit): Lawrason
Law ra son~
Enter word (press e to exit): e
```

In the figure, the top 5 words represents most frequent and bottom 5 are the less frequent words as can be seen in Task 1.3.4

Now we can notice the difference between morphological analysis and BPE algorithm. Morphological analysis tries to break the word into morphemes, which cannot be broken down further. So for example, if we look at the word hackbridge, we notice that the analyser has broken it into hack and bridge, where hack is the verb and cannot be broken down more. But BPE algorithm divided it into 3 sub-tokens H ack and bridge.

So we notice how these 2 methods differ in the way they divide the word, BPE algorithm is an algorithm that divides the tokens into sub-tokens based on the frequency factor whereas the morphological analyser tries to break it into meaningful morphemes. Morphological analyser divided below into be and low which are meaningful morphemes, whereas the BPE algorithm took it as a full word because the frequencies of consequent sub-tokens were fairly high.

Link for Google Colab Notebook (contains all the codes): https://bit.ly/34aX2Qq
Link for Google Drive that contains all the results obtained (all files): https://bit.ly/3n8GtNQ

END OF ASSIGNMENT