CS 565 ASSIGNMENT 1

PRIYANSHU SINGH 170101049

Link to Google Colab

Instructions: Go to Runtime. Select run all cells to reproduce the results. It may take 30-40 mins to run completely.

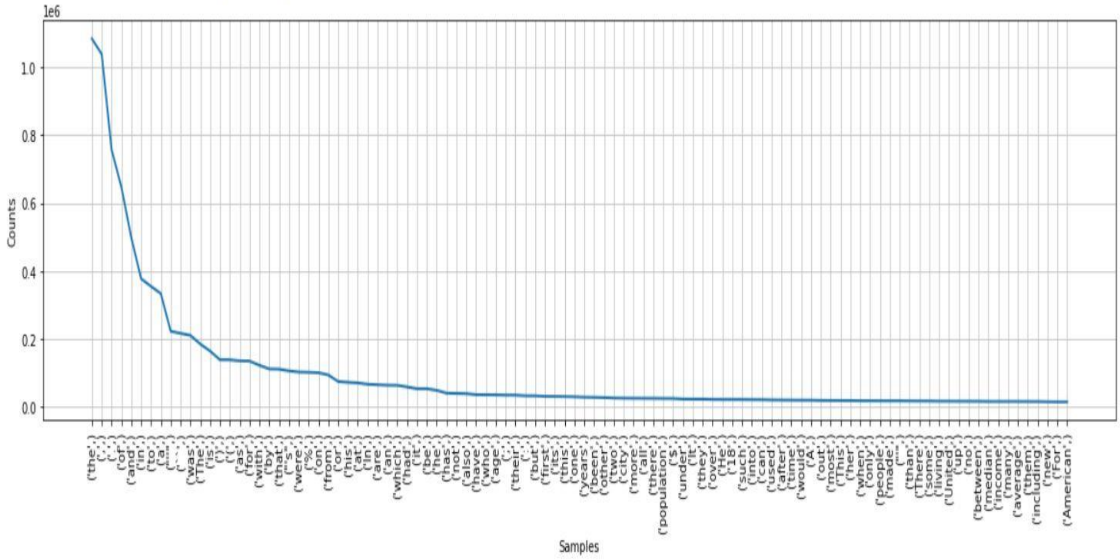
TASK 1.3.1 - Analysis using existing NLP tools

1. Sentence segmentation and Word Tokenization
   - English
     - NLTK
       - Sentences: - 7,61,582
       - Word Tokens: - 1,96,02,236
     - Spacy (only 1/4th corpus used)
       - Sentences: - 2,12,133
       - Word Tokens: - 52,29,511
   - Hindi
     - Indic-NLP
       - Sentences: - 3,48,593
       - Word Tokens: - 86,40,033
     - Spacy (only 1/4th corpus used)
       - Sentences: - 88,613
       - Word Tokens: - 21,81,076

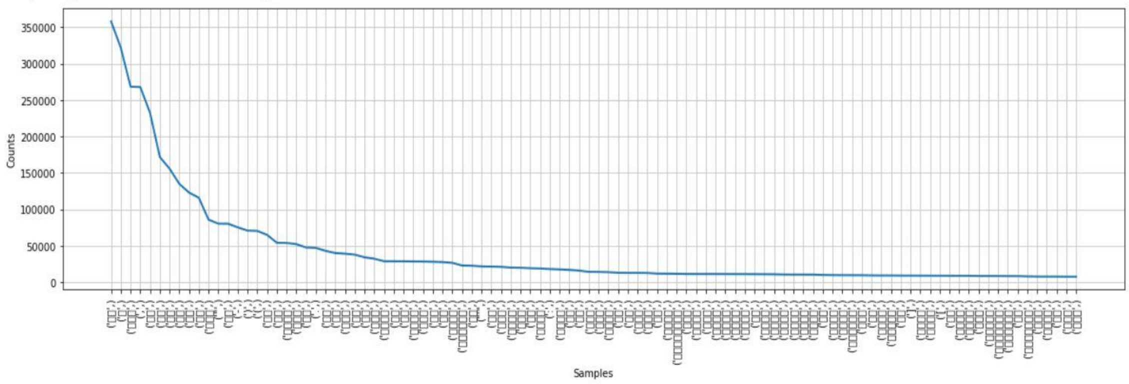
   As can be seen from the data, the number of sentences and word tokens found are different. This is because of the different heuristics used in different libraries. One heuristic is regarding **'.'**, different libraries may implement the use of '.' In titles like Mr. or Dr. differently, some may break the sentence at these points and some may not. One more example is, `` is regarded as a separate token in NLTK but not in Spacy. There are many such subtle differences in libraries across the NLP domain.

2. Total Unigrams (English Corpus): - 3,95,757
   Total Unigrams (Hindi Corpus): - 3,22,350
   Below is the plotted frequency distribution of 100 most frequent tokens

Frequency distribution of English unigrams

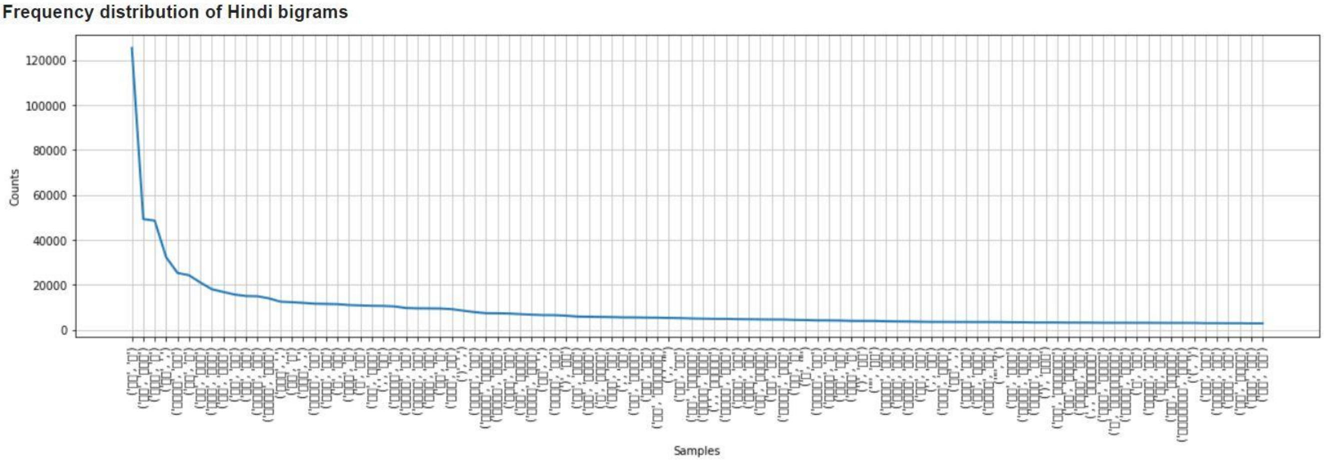


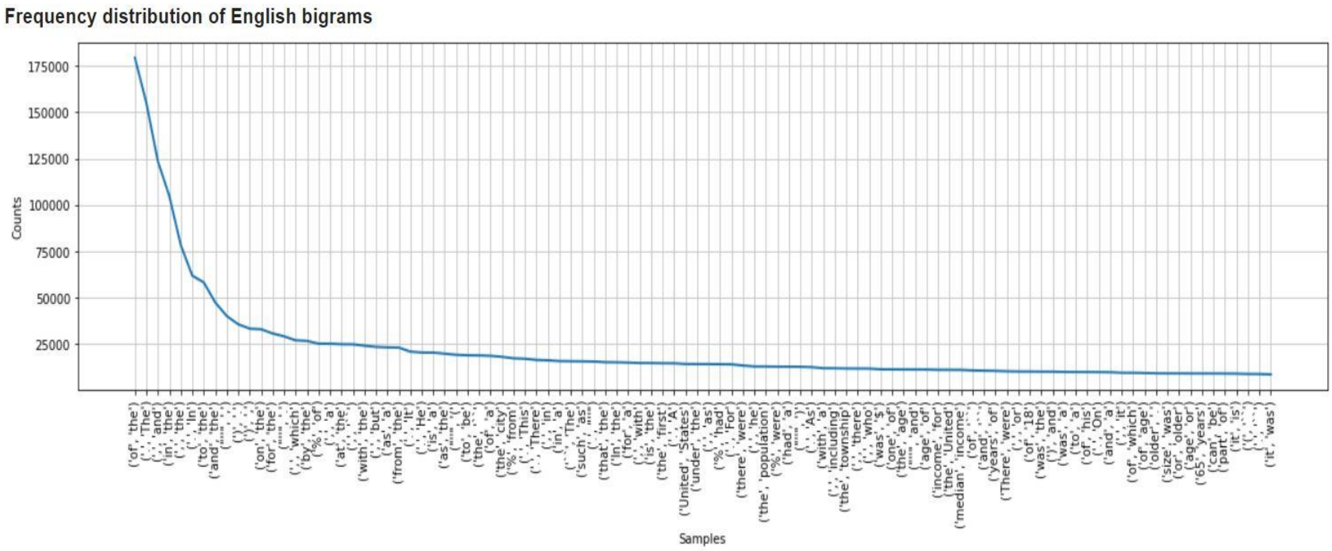
Frequency distribution of Hindi unigrams



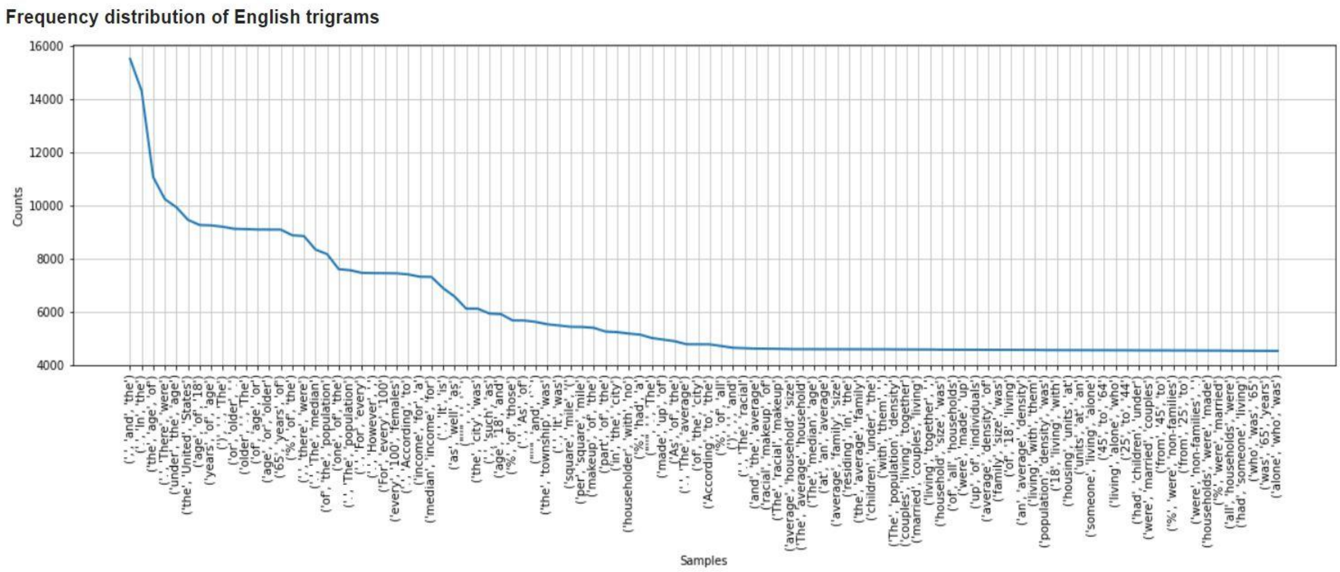
3. Total Bigrams (English Corpus): - 40,95,732
   Total Bigrams (Hindi Corpus): - 23,92,979
   Below is the plotted frequency distribution of 100 most frequent tokens

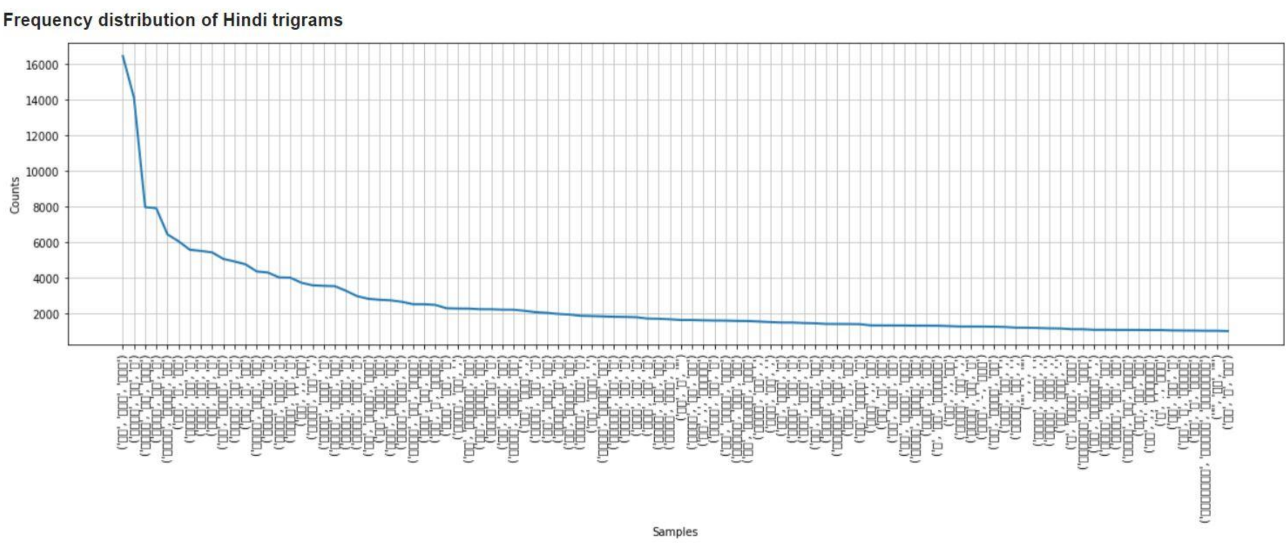
Frequency distribution of Hindi bigrams





4. Total Trigrams (English Corpus): - 1,08,27,502
   Total Trigrams (Hindi Corpus): - 55,08,737
   Below is the plotted frequency distribution of 100 most frequent tokens



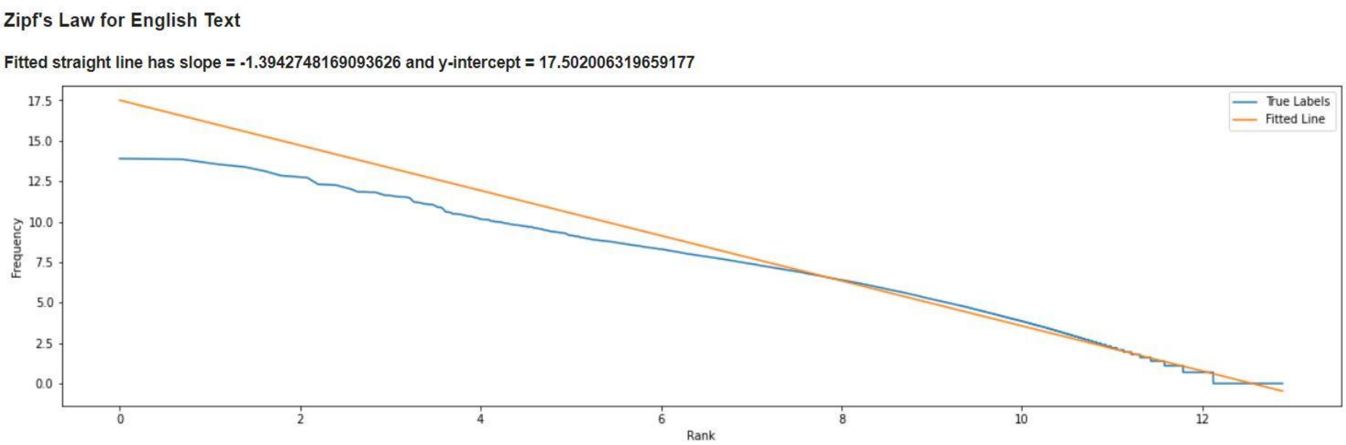


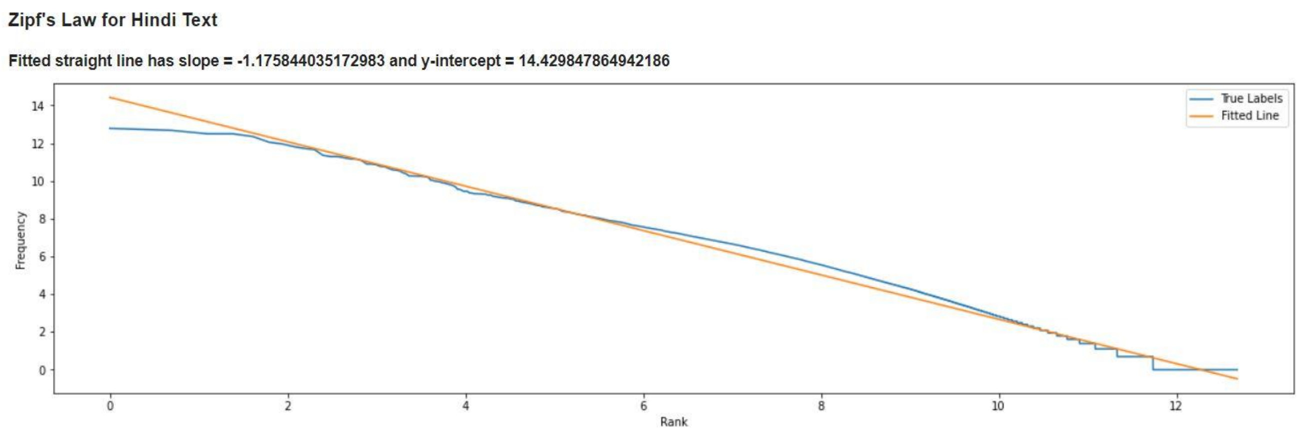
Some of the most frequent tokens in English Text are `['in', 'and', 'of', '.', ',', 'the']` which contains function words and punctuations as expected. Some of the least frequent tokens in English Text are `['100g', '13.5g', '27g', 'units—in', 'corpuscle', 'radiochemist', 'transuranium', 'alpha-particle']` which mostly contains numbers and very rare words.

Similarly, for Hindi Text, most frequent tokens are `['एक', '"', 'हैं', 'को', 'का', 'से', 'और', 'की', 'है', ',', 'में', '।', 'के']` which are again mostly function words and punctuations. Some of the least frequent tokens are `['“कमीशन', 'अक्रेदिसन', 'एजुकेशन”', 'सीएएचएमई', 'एचओएम']` which are again rare words.

<u>Zipf's Law</u>

Note: Both x and y axis are on $\log_e$ scale





As can be seen from the graphs, there is some deviation from the ideal Zipf's law as the slope of the fitting line is not exactly one. All the given graphs and tokens of least and highest frequency are present in the Colab Notebook for a better look.

TASK 1.3.2 – Few Basic Questions

Before Stemming

Type of n-gram	Corpus	% text to cover	Number
Unigram	English	90	33,731
Unigram	Hindi	90	30,969
Bigram	English	80	78,356
Bigram	Hindi	80	57,677
Trigram	English	70	4,36,876
Trigram	Hindi	70	3,27,002

Stemmer Used for English Text: - Porter Stemmer

Stemmer Used for Hindi Text: - Referred to this GitHub repo

After Stemming:

Type of n-gram	Corpus	% text to cover	Number
Unigram	English	90	14,063
Unigram	Hindi	90	19,298
Bigram	English	80	48,570
Bigram	Hindi	80	36,785
Trigram	English	70	4,24,256
Trigram	Hindi	70	2,58,215

Reduction in number of tokens required due to stemming:

Type of n-gram	Corpus	% text to cover	% Reduction
Unigram	English	90	58.30
Unigram	Hindi	90	37.68
Bigram	English	80	38.01
Bigram	Hindi	80	36.22
Trigram	English	70	2.88
Trigram	Hindi	70	21.03

As expected the number of n-grams required to cover a certain portion of corpora decreased because stemming reduces different forms of word with same root to the root itself, thus decreasing the number of words of same root type.

TASK 1.3.3

1. The heuristics applied in the task are (referring to lecture slides):
   For English

   Word Segmentation

   1. Applied putative boundaries at '.', '!' and '?'.
   2. Checked if a known abbreviation exists before a dot. This list is made separately and can be extended conveniently. If a known abbreviation exits before the dot, we do not break our sentence at that dot
   3. Check if a lowercase letter follows a '?' or a '!'. If yes, we do not break sentence on that '?' or '!'
   4. Check if an ending quote (") follows a '.', '?' or '!'. If yes, we extend the sentence boundary to the quote.

   Sentence Segmentation

   1. Split the sentences on the basis of white spaces.
   2. If a dot is preceded by a known abbreviation, it is kept as a single word
   3. Punctuation marks after a word is broken into 2 tokens, with punctuation as a separate token

   No of English sentences before applying heuristics = 7,61,582

   No of English sentences after applying heuristics = 7,98,448

   No of English words before applying heuristics = 1,96,02,236

   No of English words after applying heuristics = 1,86,10,493

   For Hindi

   Word Segmentation

   1. Applied putative boundaries at '|', '!' and '?'.
   2. Check if an ending quote (") follows a '|', '?' or '!'. If yes, we extend the sentence boundary to the quote.

   Sentence Segmentation

1. Split the sentences on the basis of white spaces.
2. Punctuation marks after a word is broken into 2 tokens, with punctuation as a separate token

No of Hindi sentences before applying heuristics = 3,48,593
No of Hindi sentences after applying heuristics = 3,50,201
No of Hindi words before applying heuristics = 86,40,033
No of Hindi words after applying heuristics = 81,86,382

Before Stemming (Heuristics case)

Type of n-gram	Corpus	% text to cover	Number
Unigram	English	90	57,189
Unigram	Hindi	90	44,071
Bigram	English	80	97,139
Bigram	Hindi	80	61,116
Trigram	English	70	5,25,589
Trigram	Hindi	70	3,24,451

Stemmer Used for English Text: - Porter Stemmer

Stemmer Used for Hindi Text: - Referred to this GitHub repo

After Stemming (Heuristics case):

Type of n-gram	Corpus	% text to cover	Number
Unigram	English	90	26,106
Unigram	Hindi	90	28,280
Bigram	English	80	57,856
Bigram	Hindi	80	39,040
Trigram	English	70	4,14,291
Trigram	Hindi	70	2,54,619

Reduction in number of tokens required due to stemming:

Type of n-gram	Corpus	% text to cover	% Reduction
Unigram	English	90	54.35
Unigram	Hindi	90	35.83
Bigram	English	80	40.43
Bigram	Hindi	80	36.12
Trigram	English	70	21.17
Trigram	Hindi	70	30.86

After applying heuristics, again after stemming there is reduction in number of tokens required which is expected.

2. In applying the **likelihood ratio test** to collocation discovery, we examine the following two alternative explanations for the occurrence frequency of a bigram w^1w^2:

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

   Hypothesis 1 is a formalization of independence (the occurrence of w^2 is independent of the previous occurrence of w^1), Hypothesis 2 is a formalization of dependence which is good evidence for an actual collocation.

   The log of likelihood ratio λ is defined as follows:

   $$\log \lambda = \frac{\log H_1}{\log H_2}$$

   Assuming binomial distribution, we find the values of $\frac{\log H_1}{\log H_2}$. We return the value of -2log λ. Higher this value, higher is the probability of w^1w^2 to be a collocation.

   Some of the higher value bigrams in English are:

w^1	w^2	-2 log λ
"	()	1,69,882.04
"	and	1,43,930.19
s	``	67,244.20
as	an	32,585.64
19^{th}	century	22,074.77

As you can see, some of the bigrams with highest value of -2log λ are the ones with punctuations marks, which is expected. There is need of grammatical filters to remove such bigrams. But apart from that, we can also see that we have bigrams like '*19^{th} century*' or '*as an*', which is a probable collocation as suggested by the maximum likelihood test.

Similar observations hold for Hindi Text. Some of its highest valued bigrams are:

w^1	w^2	-2 log λ
नदी	के	8174.15
एक	बहुत	7043.51
2005	में	6628.61
आदि	के	6110.02

TASK 1.3.4 – Morphological Parsing

I used Polyglot library for finding morphemes from both English and Hindi tokens. For finding POS(part-of-speech), I used *pos_tag* from NLTK library. The complete breakup of method can be found in the Colab Notebook, here I present few examples of Morphological parsing.

Token	Morphemes	Part-of-speech
many	['man', 'y']	JJ (adjective (large))
corpuscle	['corp', 'us', 'cle']	NN (noun, singular)
उन्होंने	['उन्हों', 'ने']	NNP(proper noun, singular)
अनुक्रमणियों	['अनु', 'क्रमण', 'ियों']	NNP(proper noun, singular)

As can be seen from the table, tokens are broken into morphemes (smallest meaningful unit in a language), and for the full analysis, we add POS tags.

Polyglot model: Polyglot offers trained morfessor models to generate morphemes from words. Morfessor is a family of probabilistic machine learning methods for finding the morphological segmentation from raw text data. In the morphological segmentation task, the goal is to segment words into morphemes, the smallest meaning-carrying units. Models of the Morfessor family are generative probabilistic models that predict compounds and their analyses (segmentations) given the model parameters. For prior probability, Morfessor Baseline defines a distribution over the lexicon of the model. The prior assigns higher probability to lexicons that store fewer and shorter constructions. The lexicon prior consists of two parts, a product over the form probabilities and a product over the usage probabilities. The former includes the probability of a sequence of atoms and the latter the maximum likelihood estimates of the constructions.

TASK 1.3.5 – SUB-WORD TOKENIZATION

One of the ways to do sub-word tokenization is via an algorithm called *Byte-Pair-Encoding(BPE)*.

I used 1000 words from corpus having length more than or equal to 3(to remove punctuations) and set the number of merges to 1000.

50 most frequent English tokens are:

```
[('the#', 1083394), ('and#', 498234), ('was#', 210600), ('The#', 185104), ('for#', 134955), ('with#', 122534), ('that#', 111073), ('were#', 102616), ('from#', 94024), ('his#', 72322), ('are#', 65124), ('which#', 63495), ('had#', 58807), ('has#', 40502), ('not#', 39793), ('also#', 39082), ('have#', 35996), ('their#', 34599), ('but#', 32675), ('first#', 31067), ('its#', 30949), ('this#', 30477), ('one#', 29760), ('years#', 28550), ('been#', 28122), ('other#', 27357), ('two#', 26105), ('more#', 25355), ('all#', 25250), ('there#', 25076), ('under#', 23031), ('they#', 22733), ('over#', 22060), ('such#', 21763), ('into#', 21396), ('can#', 21276), ('used#', 20526), ('after#', 20244), ('time#', 19953), ('most#', 18745), ('This#', 18717), ('her#', 18508), ('when#', 18030), ('only#', 17972), ('made#', 17780), ('than#', 17462), ('There#', 17269), ('some#', 17249), ('between#', 16452), ('average#', 15979)]
```

50 least frequent English tokens are:

```
[('mu', 11), ('ir#', 11), ('ch#', 10), ('w', 8), (',', 8), ('lu', 8), ('en', 8), ('characteri', 8), ('bari', 7), ('cts#', 7), ('od', 6), ('tassi', 6), ('4', 6), (':', 5), ('an#', 4), ('dy#', 4), ('M', 4), ('sub', 4), ('lead', 4), ('energ', 4), ('enty', 4), ('thi', 4), ('residu', 3), ('ron', 3), ('ter', 2), ('Ger', 2), ('la', 2), ('dron#', 2), ("'", 2), ('ce#', 2), ('is#', 2), ('boron', 2), ('cor', 1), ('us', 1), ('le#', 1), ('transurani', 1), ('range', 1), ('ranged#', 1), ('ton', 1), ('non', 1), ('recoverable#', 1), ('releasing#', 1), ('roun', 1), ('ununo', 1), ('adi', 1), ('lan', 1), ('thanum', 1), ('talu', 1), ('ev', 1), ('dal', 1)]
```

50 most frequent Hindi tokens are:

```
[('में#', 268249), ('हैं#', 85787), ('किया#', 53857), ('लिए#', 52226), ('गया#', 39123), ('रूप#', 32207), ('जाता#', 28738), ('करने#', 28440), ('साथ#', 28354), ('नहीं#', 26563), ('द्वारा#', 22877), ('तथा#', 21081), ('बाद#', 20249), ('अपने#', 19993), ('दिया#', 18776), ('होता#', 17543), ('कुछ#', 14228), ('हुआ#', 14137), ('करते#', 13756), ('हुए#', 12776), ('अधिक#', 11699), ('उन्होंने#', 11542), ('नाम#', 11306), ('सकता#', 11263), ('एवं#', 11259), ('इसके#', 11248), ('उनके#', 11209), ('करता#', 11150), ('होती#', 11111), ('भारत#', 11083), ('कारण#', 10939), ('होने#', 10867), ('अन्य#', 10415), ('वाले#', 10397), ('अपनी#', 10358), ('पहले#', 9712), ('सबसे#', 9618), ('प्रकार#', 9554), ('कहा#', 9536), ('किसी#', 9210), ('लेकिन#', 9199), ('जैसे#', 8991), ('होते#', 8922), ('जाती#', 8783), ('हुई#', 8536), ('शामिल#', 8520), ('क्षेत्र#', 8491), ('उन्हें#', 8382), ('प्राप्त#', 7970), ('तरह#', 7709)]
```

50 least frequent Hindi tokens are:

```
[('देख', 4), ('सबी#', 3), ('समे', 3), ('बि', 3), ('मिल', 3), ('निक', 3), ('ि#', 3), ('ुरु', 3), ('ष्य#', 3), ('जित#', 2), ('ुरहा', 2), ('d', 2), ('त्व', 2), ('दे', 2), ('विष्', 2), ('दार्थ', 2), ('“', 1), ('अक्रे', 1), ('सन#', 1), ('”', 1), ('पाठ', 1), ('शालाओं#', 1), ('डिग', 1), ('सा#', 1), ('राज्व', 1), ('ज्', 1), ('जी#', 1), ('फ़', 1), ('रह', 1), ('u', 1), ('i', 1), ('n', 1), ('स्त्र', 1), ('देव', 1), ('ब#', 1), ('लेक#', 1), ('संगार', 1), ('ुरा#', 1), ('फ#', 1), ('नारा', 1), ('की#', 1), ('बंध', 1), ('धा#', 1), ('त्र', 1), ('देवा', 1), ('विष', 1), ('संधान#', 1), ('सर्व', 1), ('यार्थ#', 1), ('श्चित#', 1)]
```

<u>Comparison of morphological parsing and BPE for English Text</u> (full table in Colab)

Token	Morphemes	BPE
there	[‘there’]	[‘there’]
who	[‘wh’, ‘o’]	[‘w’, ‘h’, ‘o’]
on	[‘on’]	[‘on’]
transuranium	['trans', 'ur', 'an', 'ium']	['transurani', 'u', 'm']

Similarly, observations hold for Hindi text. (complete list in Colab)

Key Observations:

1. The Morphological parsing and BPE tokenization of words like ‘there’ and ‘on’ are same.
2. There would be some words that do not have a BPE tokenization because of the limited vocabulary chosen.
3. We can decrease the number of such words by taking a larger library and increasing the number of merges.