CS565: Assignment-1 Report

Umang(170101074)

Link to colab notebook: [Assignment1(CS565)_170101074.ipynb]

1.3.1 Analysis using existing NLP tools

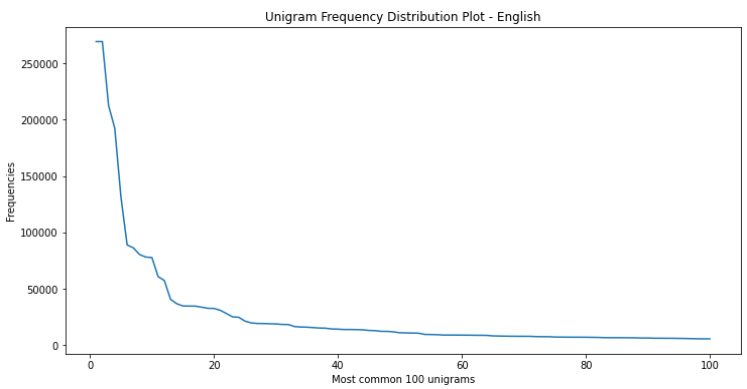
1. Sentence segmentation and word tokenization using different tools
   For English corpus, following tools were used:
   - NLTK
     Total number of sentences = 213196
     Total number of words = 5057940
   - spaCy
     Total number of sentences = 212133
     Total number of words = 5229511

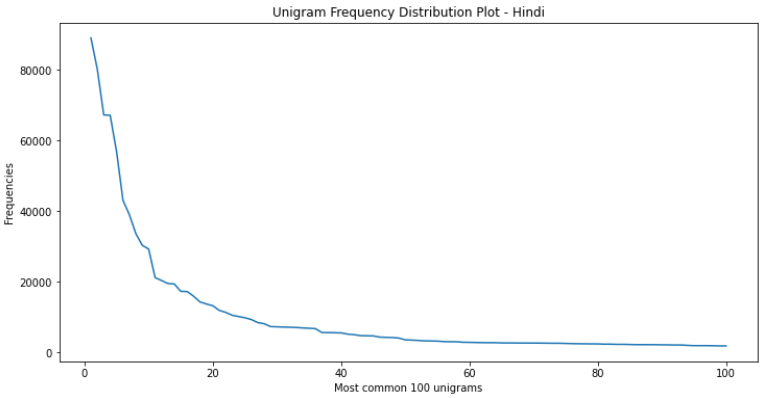
   For HIndi corpus, following tools were used:
   - IndicNLP
     Total number of sentences = 86882
     Total number of words = 2160191
   - StanfordNLP
     Total number of sentences = 91130
     Total number of words = 2095356

   Other explored tools include iNLTK, CoreNLP etc

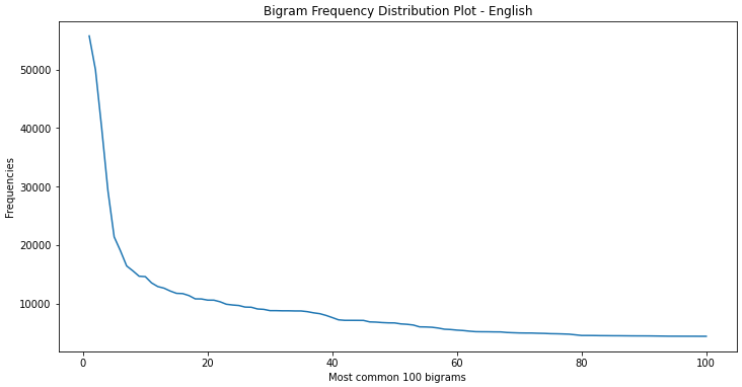
For the following questions, for generating and plotting n-grams in English, tools provided by NLTK library were used. However, there were no available tools for n-grams in hindi. So for Hindi, tokenized words(done using indicNLP) were used along with NLTK to generate n-grams.

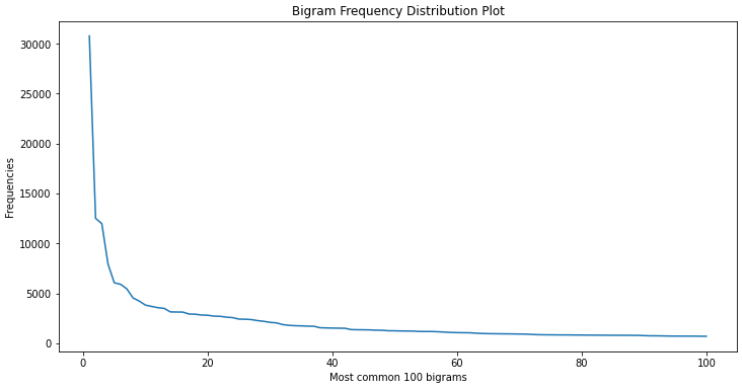
2. Uni-grams Frequency Distribution
   Plots for both the corpora are as follows:





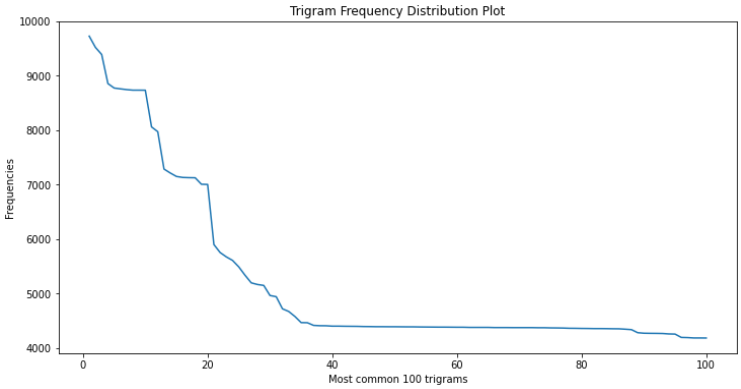
3. Bi-grams Frequency Distribution
   Plots for both the corpora are as follows:

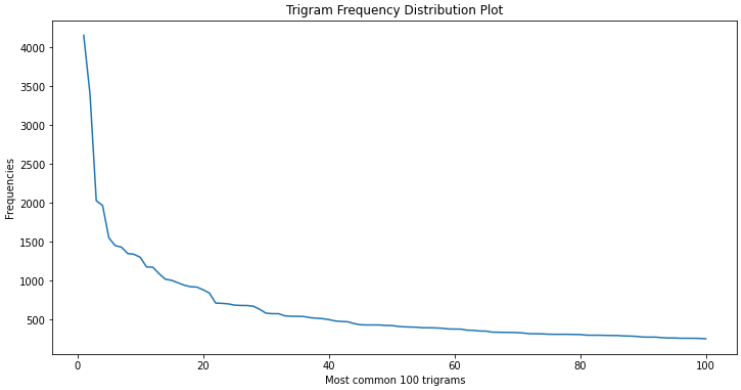




4. Tri-grams Frequency Distribution

Plots for both the corpora are as follows:





5. Questions:

- For English, there are many tools available. Some of them include: NLTK, spaCy, stanfordNLP etc. For Hindi, a lesser number of tools are available which include iNLTK, indicNLP and stanfordNLP.
- Difference in results using different tools can be attributed to different heuristics used by models i.e. how they handle punctuations and abbreviations etc. However the difference isn't very large.
- Most frequent words include articles(the,a), punctuations, conjunctions etc. Least frequent words include numerical values and proper nouns.

Zipf's Law:

According to Zipf's law, frequency of a word type is inversely proportional to its rank.

$$f \propto 1/r$$

Taking log of the equation makes it easier to fit the equation on frequency distribution.

$$\log f = \log c - \log r$$

Plotting frequency distribution of words(on log scale) and the fitting a linear equation gives the following results:

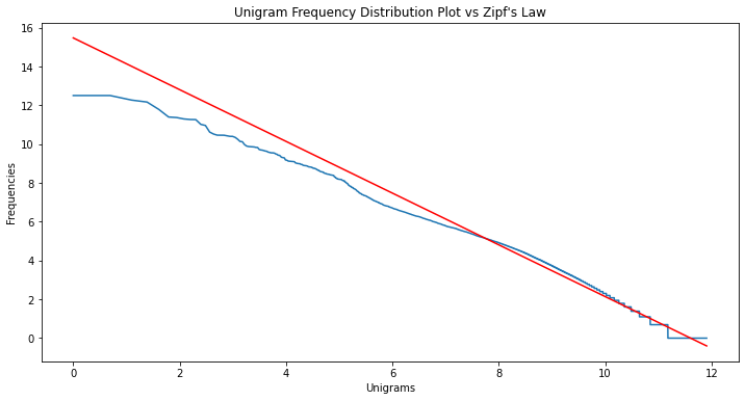
Slope for English corpus = `-1.3329679351853299`

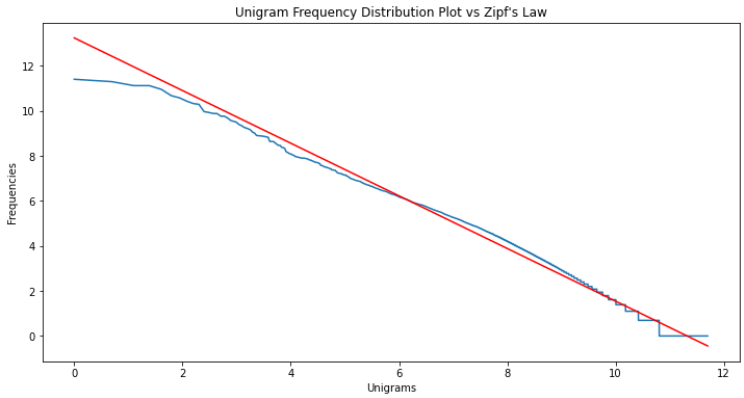
Intercept for English corpus = `15.471324627261797`

Slope for Hindi corpus = `-1.1687693027291786`

Intercept for Hindi corpus = `13.234523105816232`

As it can be seen, the curves for frequency distribution are quite close to a fitted straight line with slope quite close to -1.





1.3.2 Few basic questions

Coverage analysis before stemming:

1. English Corpus:
   Uni-grams required for 90% coverage of corpus = `8947`
   Bi-grams required for 80% coverage of corpus = `242884`
   Tri-grams required for 70% coverage of corpus = `992668`

2. Hindi Corpus:
   Uni-grams required for 90% coverage of corpus = `11658`
   Bi-grams required for 80% coverage of corpus = `366558`
   Tri-grams required for 70% coverage of corpus = `923215`

For stemming in English corpus, PorterStemmer from NLTK is used. For stemming in Hindi corpus, I referred to a github repo(Link) as there was no tool available for it.

Coverage analysis after stemming:

1. English Corpus:
   Uni-grams required for 90% coverage of corpus = `4270`
   Bi-grams required for 80% coverage of corpus = `160317`
   Tri-grams required for 70% coverage of corpus = `884495`

2. Hindi Corpus:
   Uni-grams required for 90% coverage of corpus = `7179`
   Bi-grams required for 80% coverage of corpus = `284614`
   Tri-grams required for 70% coverage of corpus = `866237`

Clearly there is a decrease in the number of n-grams required after stemming. This is because some groups of words have the same stem word. Thus their frequencies are accumulated, decreasing the number of stem words required for corpus coverage.

1.3.3 Writing some of your basic codes

Implementing heuristics for sentence segmentation and word tokenization:

For English corpus, first splitted the given copus by space. This gives the broad level segmentation. Then marked the putative boundary at '.' , '?' and '!'. Discarded a period boundary if it is preceded by a known abbreviation. Discarded a '?' or '!' boundary if it is followed by a lowercase letter.
For Hindi corpus, first splitted the given corpus by space. Then marked the sentence boundaries by '|' , '?' and '!' . For tokenization of words removed the punctuation marks from tokens.
Results are as follows:

1. English Corpus:

Number of sentences = 216829
Number of words = 4496840

Uni-grams required for 90% coverage of corpus before stemming = 6089
Bi-grams required for 80% coverage of corpus before stemming = 202020
Tri-grams required for 70% coverage of corpus before stemming = 1298015

Uni-grams required for 90% coverage of corpus after stemming = 2571
Bi-grams required for 80% coverage of corpus after stemming = 123094
Tri-grams required for 70% coverage of corpus after stemming = 740085

3. Hindi Corpus:

Number of sentences = 79004
Number of words = 1890275

Uni-grams required for 90% coverage of corpus before stemming = 20112
Bi-grams required for 80% coverage of corpus before stemming = 454286
Tri-grams required for 70% coverage of corpus before stemming = 1136333

Uni-grams required for 90% coverage of corpus after stemming = 2571
Bi-grams required for 80% coverage of corpus after stemming = 123094
Tri-grams required for 70% coverage of corpus after stemming = 740085

Likelihood ratio Test:

Implemented the likelihood ratio test as described in FSNLP Ch. 5
In likelihood ratio test we examine two hypothesis for a bigram w1w2, one for formalization of independence and the other for formalization of dependence:

$$Hypothesis\ 1.\ P(w2|w1) = p = P(w2|\neg w1)$$

$$Hypothesis\ 2.\ P(w2|w1) = p1 \neq p2 = P(w2|\neg w1)$$

Calculated the likelihood ratio as follow:

$$log\ \lambda = log(\ L(H1)/L(H2)\ )$$

$$log\ \lambda = log\ L(c12,\ c1,\ p) + log\ L(c2 - c12,\ N - c1,\ p) - log\ L(c12,\ c1,\ p1) - log\ L(c2 - c12, N - c1,\ p2)$$

This $log\,\lambda$ is calculated in function `get_likelihood_ratio`. More the value of $-2 * log\,\lambda$ more the chances of $w1w2$ being a collocation.
First few values of $-2 * log\,\lambda$ can be seen in the notebook.
For English corpus, the bigram `('.', 'The')` has the highest value of $-2 * log\,\lambda$ because every sentence ends with a period and the most probable first word for next sentence is 'The'. Similarly for Hindi corpus, `('है', '।')` has the highest value of $-2 * log\,\lambda$ because every sentence ends with a '|' and the most probable last word for the previous sentence is `'है'`.

1.3.4 Morphological parsing

Used polyglot library for morphological analysis as it provides morphological analysis for both hindi and english. For POS-tagging used NLTK pos-tagger. Results for five random most frequent and least frequent words are as follows:

For random 5 words from 100 most frequent

Word --> morphemes + POS Tag

```
city --> ['city'] + NN
median --> ['media', 'n'] + JJ
Township --> ['Town', 'ship'] + NNP
had --> ['had'] + VBD
first --> ['first'] + JJ
```

For random 5 words from 100 least frequent

Word --> morphemes + POS Tag

```
immunize --> ['imm', 'un', 'ize'] + VB
handbills --> ['hand', 'bill', 's'] + NNS
Long-term --> ['Long', '-', 'term'] + JJ
Pension --> ['Pen', 's', 'ion'] + NNP
Purrington --> ['Pur', 'ring', 'ton'] + NNP
```

For random 5 words from 100 most frequent

Word --> morphemes + POS Tag

```
जाता --> ['जा', 'ता'] + JJ
द्वारा --> ['द्वारा'] + NNP
जैसे --> ['जैस', 'े'] + NNP
कि --> ['कि'] + NNP
किसी --> ['किस', 'ी'] + NN
```

For random 5 words from 100 least frequent

Word --> morphemes + POS Tag

```
दरजी --> ['दर', 'जी'] + JJ
आधीनस्थ --> ['आध', 'ी', 'न', 'स्थ'] + NNP
एक्सटीरियर --> ['एक्स', 'टी', 'रियर'] + NNP
मूर्खतापूर्ण --> ['मूर्ख', 'ता', 'पूर्ण'] + NNP
मोर्गेसों --> ['मोर्', 'गे', 'स', 'ों'] + NN
```

Polyglot offers trained morfessor models to generate morphemes from words.

1.3.5 Sub-word tokenization

Implementation of Byte-Pair-Encoding (BPE) algorithm is done as described in the JurafskyMartin Book. *Vocabulary size is 3000 and number of merges are set to 4500.*

50 most common tokens learnt can be seen here:

- For English Corpus: Link
- For Hindi Corpus: Link

Tokenization for 10 unknown words:

Tokenization of unknown words

```
abstruse --> ['ab', 'str', 'use']
bilk --> ['b', 'il', 'k']
enervate --> ['ener', 'v', 'ate']
accomodate --> ['ac', 'com', 'od', 'ate']
multifarious --> ['multi', 'f', 'ar', 'i', 'ous']
cajole --> ['ca', 'j', 'ole']
reprobate --> ['re', 'prob', 'ate']
swarthy --> ['s', 'w', 'ar', 'th', 'y']
covet --> ['co', 've', 't']
bashful --> ['b', 'a', 'sh', 'ful']
```

Tokenization of unknown words

```
विभीषिका --> ['वि', 'भी', 'ष', 'ि', 'का']
किंकर्तव्यविमूढ़ --> ['कि', 'ं', 'कर्', 'त', 'व्य', 'वि', 'मू', 'ढ़']
व्योमबाला --> ['व्य', 'ो', 'म', 'बा', 'ला']
वैतरिणी --> ['वै', 'त', 'रि', 'णी']
निक्षेपक --> ['नि', 'क्षे', 'प', 'क']
स्वतांत्रयोत्तर --> ['स्व', 'ता', 'ं', 'त्र', 'यो', 'त्तर']
रासस्मीकरण --> ['रा', 'स', 'स्', 'मी', 'करण']
धृष्टता --> ['ध', 'ृ', 'ष्ट', 'ता']
अच्युतानंद --> ['अ', 'च', '्य', 'ु', 'ता', 'न', 'ंद']
प्रत्युत्पन्नमति --> ['प्रत्य', 'ु', 'त्', 'प', 'न्', 'न', 'म', 'ति']
```

Comparison of tokens with morphemes

For random 5 words from 100 most frequent

```
Word    -->    From Polyglot    and    From BPE

city    -->    ['city']    and    ['city']

median    -->    ['media', 'n']    and    ['medi', 'an']

Township    -->    ['Town', 'ship']    and    ['T', 'ow', 'n', 'ship']

had    -->    ['had']    and    ['had']

first    -->    ['first']    and    ['first']
```

For random 5 words from 100 least frequent

```
Word    -->    From Polyglot    and    From BPE

immunize    -->    ['imm', 'un', 'ize']    and    ['imm', 'uni', 'ze']

handbills    -->    ['hand', 'bill', 's']    and    ['hand', 'b', 'il', 'ls']

Long-term    -->    ['Long', '-', 'term']    and    ['Lon', 'g', '-', 'term']

Pension    -->    ['Pen', 's', 'ion']    and    ['P', 'en', 'sion']

Purrington    -->    ['Pur', 'ring', 'ton']    and    ['P', 'ur', 'r', 'ing', 'ton']
```

For random 5 words from 100 most frequent

Word --> From Polyglot and From BPE

जाता --> ['जा', 'ता'] and ['जाता']

द्वारा --> ['द्वारा'] and ['द्वारा']

जैसे --> ['जैस', 'े'] and ['जैसे']

कि --> ['कि'] and ['कि']

किसी --> ['किस', 'ी'] and ['किसी']

For random 5 words from 100 least frequent

Word --> From Polyglot and From BPE

दरजी --> ['दर', 'जी'] and ['दर', 'जी']

आधीनस्थ --> ['आध', 'ी', 'न', 'स्थ'] and ['आध', 'ी', 'न', 'स्थ']

एक्सटीरियर --> ['एक्स', 'टी', 'रियर'] and ['ए', 'क्स', 'टी', 'रि', 'यर']

मूर्खतापूर्ण --> ['मूर्ख', 'ता', 'पूर्ण'] and ['मू', 'र्', 'ख', 'त', 'ाप', 'ूर्', 'ण']

मोर्गेसों --> ['मोर्', 'गे', 'स', 'ों'] and ['मो', 'र्', 'गे', 'सों']

Note: In all the tasks one-fourth of the provided corpora is used to avoid memory related issues while processing (also suggested in assignment).