Assignment No - 2

Natural Languages Processing

Aim :- Using programming language Python and suitable libraries perform fundamental Language processing for three different languages.

Theory:-

NLP stands for Natural Language Processing, which is a part of Computer Science, Human language, and Artificial Intelligence.

It is the technology that is used by machines to understand, analyse, manipulate, and interpret human's languages.

It helps developers to organize knowledge for performing tasks such as translation, automatic summarization, Named Entity Recognition (NER), speech recognition, relationship extraction, and topic segmentation.

Steps Of NLP - 1) Lexical Analysis

2) Syntactic Analysis

3) Semantic Analysis

4) Discourse Integration

5) Pragmatic Analysis

NLP Libraries -

1) NLTK –

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, etc. This library provides a practical introduction to programming for language processing. NLTK has been called “a wonderful tool for teaching and working in computational linguistics using Python,” and “an amazing library to play with natural language.”

Features Of NLTK :-

Helps with text classification

Helps with tokenization

Helps with parsing

Helps with part-of-speech tagging

Helps with stemming

The most well-known and full NLP library.

Plenty of approaches to each NLP task.

It Supports a significant number of languages.

2) iNLTK –

The iNLTK library is the Indian language equivalent of the popular NLTK Python package. This library is built with the goal of providing features that an NLP application developer will need.iNLTK provides most of the features that modern NLP tasks require, like generating a vector embedding for input text, tokenization, sentence similarity etc. in a very intuitive and easy API interface.

Features Of iNLTK -

Tokenization

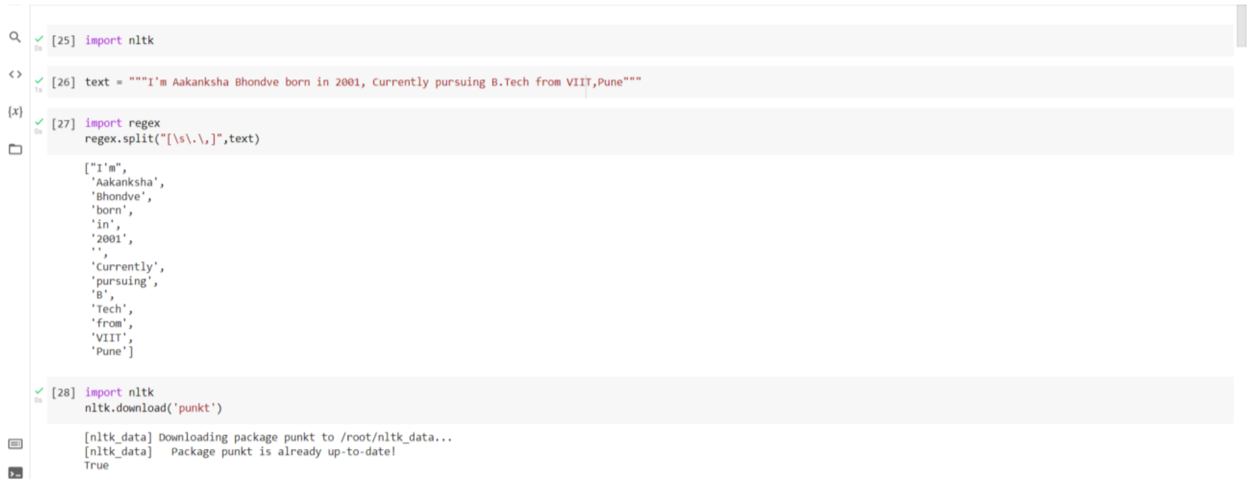
Word Embeddings

Text Completion

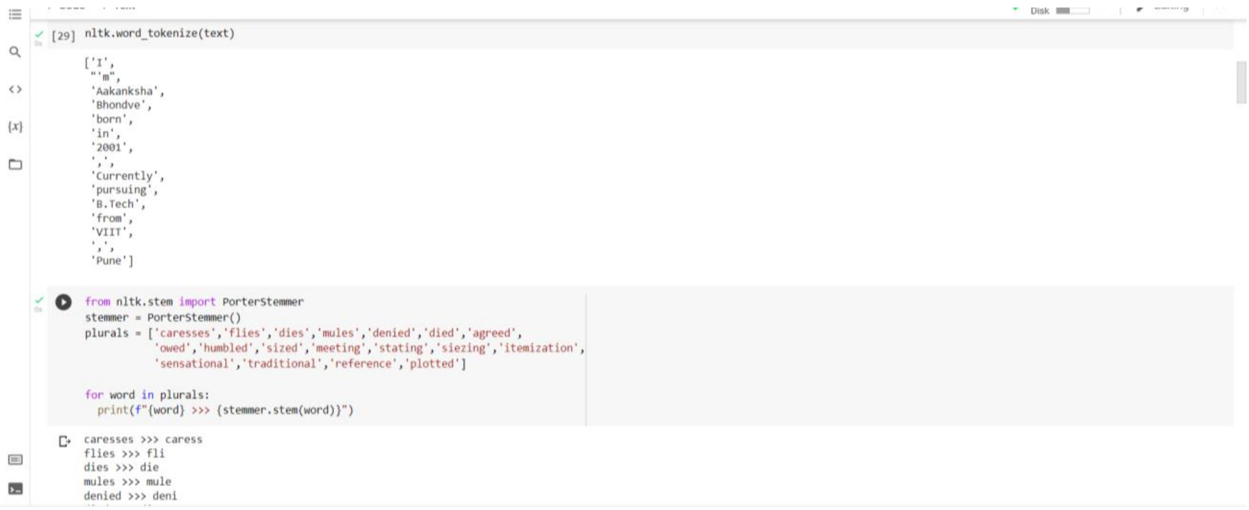
Similarity of sentences

This feature of iNLTK is very useful for text data augmentation as we can just multiply the sentences in our training data by populating it with sentences that have a similar meaning.

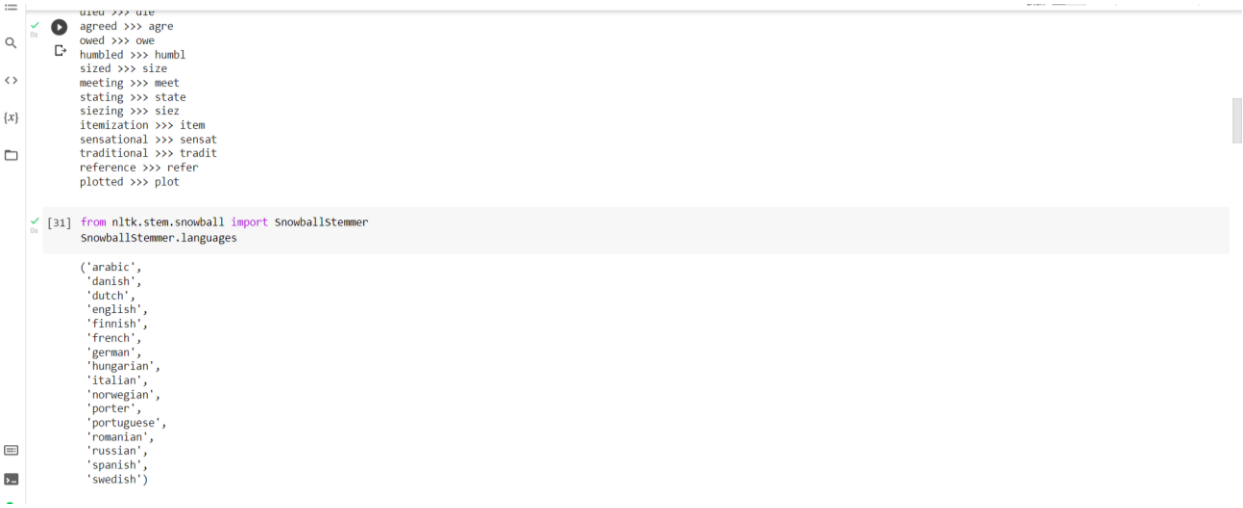
Code and output -



```
[25] import nltk

[26] text = """I'm Aakanksha Bhondve born in 2001, Currently pursuing B.Tech from VIIT,Pune"""

[27] import regex
     regex.split("[\s\.\,]",text)

     ["I'm",
      'Aakanksha',
      'Bhondve',
      'born',
      'in',
      '2001',
      '',
      'Currently',
      'pursuing',
      'B',
      'Tech',
      'from',
      'VIIT',
      'Pune']

[28] import nltk
     nltk.download('punkt')

     [nltk_data] Downloading package punkt to /root/nltk_data...
     [nltk_data]   Package punkt is already up-to-date!
     True
```



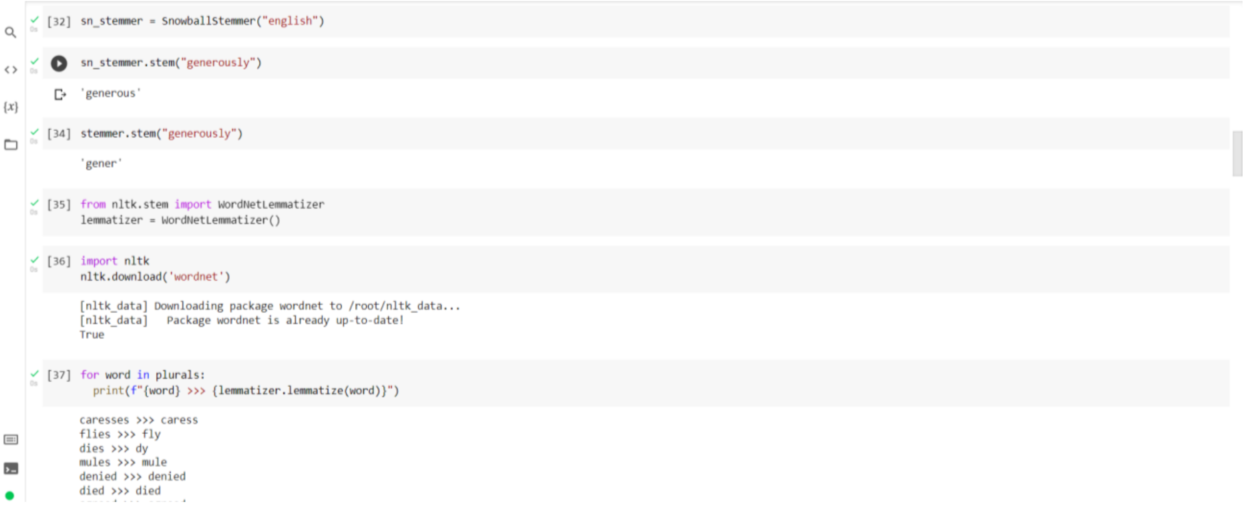
```
[29] nltk.word_tokenize(text)

     ['I',
      "'m",
      'Aakanksha',
      'Bhondve',
      'born',
      'in',
      '2001',
      ',',
      'Currently',
      'pursuing',
      'B.Tech',
      'from',
      'VIIT',
      ',',
      'Pune']

     from nltk.stem import PorterStemmer
     stemmer = PorterStemmer()
     plurals = ['caresses','flies','dies','mules','denied','died','agreed',
                'owed','humbled','sized','meeting','stating','siezing','itemization',
                'sensational','traditional','reference','plotted']

     for word in plurals:
       print(f"{word} >>> {stemmer.stem(word)}")

     caresses >>> caress
     flies >>> fli
     dies >>> die
     mules >>> mule
     denied >>> deni
```



```
died >>> die
agreed >>> agre
owed >>> owe
humbled >>> humbl
sized >>> size
meeting >>> meet
stating >>> state
siezing >>> siez
itemization >>> item
sensational >>> sensat
traditional >>> tradit
reference >>> refer
plotted >>> plot
```

```
[31] from nltk.stem.snowball import SnowballStemmer
     SnowballStemmer.languages
```

```
('arabic',
 'danish',
 'dutch',
 'english',
 'finnish',
 'french',
 'german',
 'hungarian',
 'italian',
 'norwegian',
 'porter',
 'portuguese',
 'romanian',
 'russian',
 'spanish',
 'swedish')
```



```
[32] sn_stemmer = SnowballStemmer("english")
```

```
sn_stemmer.stem("generously")
```

```
'generous'
```

```
[34] stemmer.stem("generously")
```

```
'gener'
```

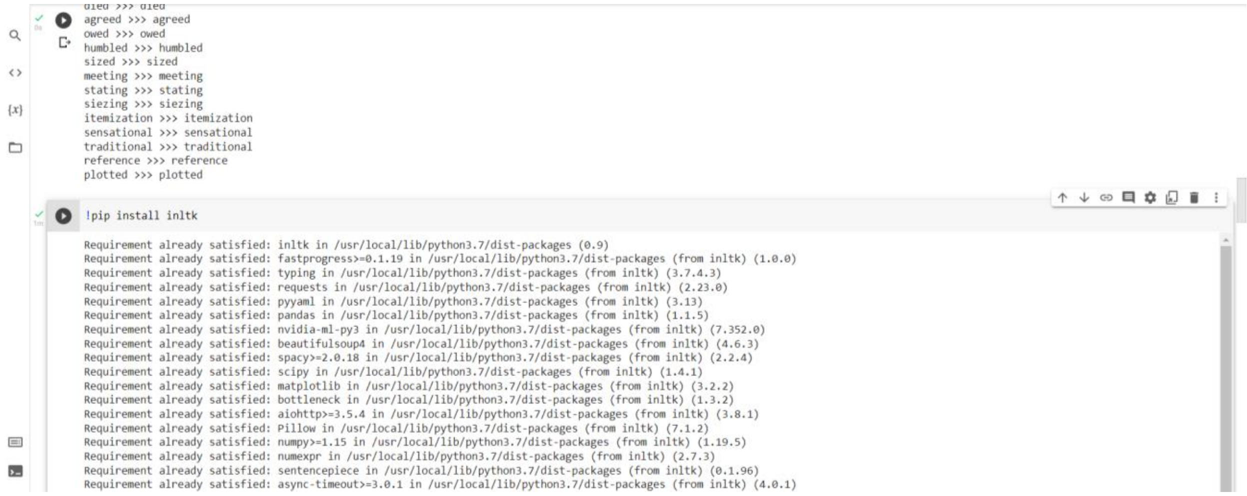
```
[35] from nltk.stem import WordNetLemmatizer
     lemmatizer = WordNetLemmatizer()
```

```
[36] import nltk
     nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
True
```

```
[37] for word in plurals:
       print(f"{word} >>> {lemmatizer.lemmatize(word)}")
```

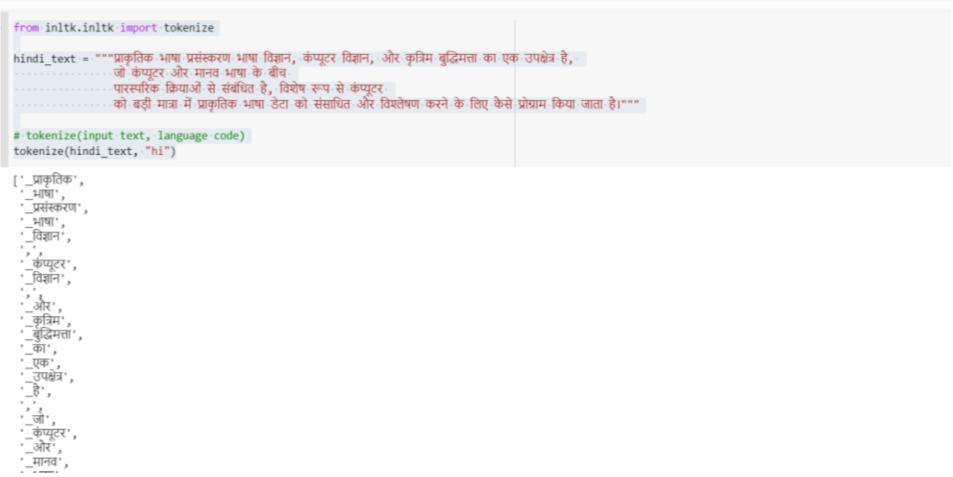
```
caresses >>> caress
flies >>> fly
dies >>> dy
mules >>> mule
denied >>> denied
died >>> died
```



```
died >>> died
agreed >>> agreed
owed >>> owed
humbled >>> humbled
sized >>> sized
meeting >>> meeting
stating >>> stating
siezing >>> siezing
itemization >>> itemization
sensational >>> sensational
traditional >>> traditional
reference >>> reference
plotted >>> plotted
```

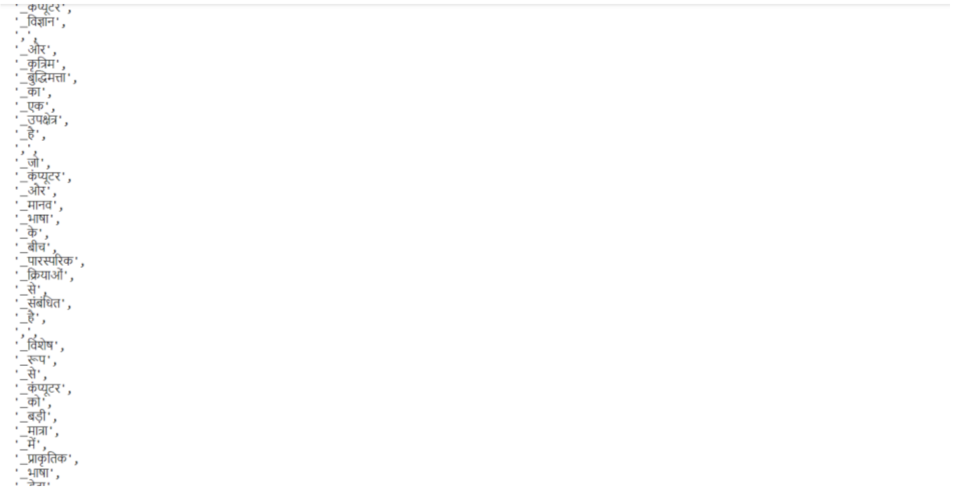
```
!pip install inltk
```

```
Requirement already satisfied: inltk in /usr/local/lib/python3.7/dist-packages (0.9)
Requirement already satisfied: fastprogress>=0.1.19 in /usr/local/lib/python3.7/dist-packages (from inltk) (1.0.0)
Requirement already satisfied: typing in /usr/local/lib/python3.7/dist-packages (from inltk) (3.7.4.3)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from inltk) (2.23.0)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages (from inltk) (3.13)
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (from inltk) (1.1.5)
Requirement already satisfied: nvidia-ml-py3 in /usr/local/lib/python3.7/dist-packages (from inltk) (7.352.0)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.7/dist-packages (from inltk) (4.6.3)
Requirement already satisfied: spacy>=2.0.18 in /usr/local/lib/python3.7/dist-packages (from inltk) (2.2.4)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from inltk) (1.4.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from inltk) (3.2.2)
Requirement already satisfied: bottleneck in /usr/local/lib/python3.7/dist-packages (from inltk) (1.3.2)
Requirement already satisfied: aiohttp>=3.5.4 in /usr/local/lib/python3.7/dist-packages (from inltk) (3.8.1)
Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from inltk) (7.1.2)
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.7/dist-packages (from inltk) (1.19.5)
Requirement already satisfied: numexpr in /usr/local/lib/python3.7/dist-packages (from inltk) (2.7.3)
Requirement already satisfied: sentencepiece in /usr/local/lib/python3.7/dist-packages (from inltk) (0.1.96)
Requirement already satisfied: async-timeout>=3.0.1 in /usr/local/lib/python3.7/dist-packages (from inltk) (4.0.1)
```

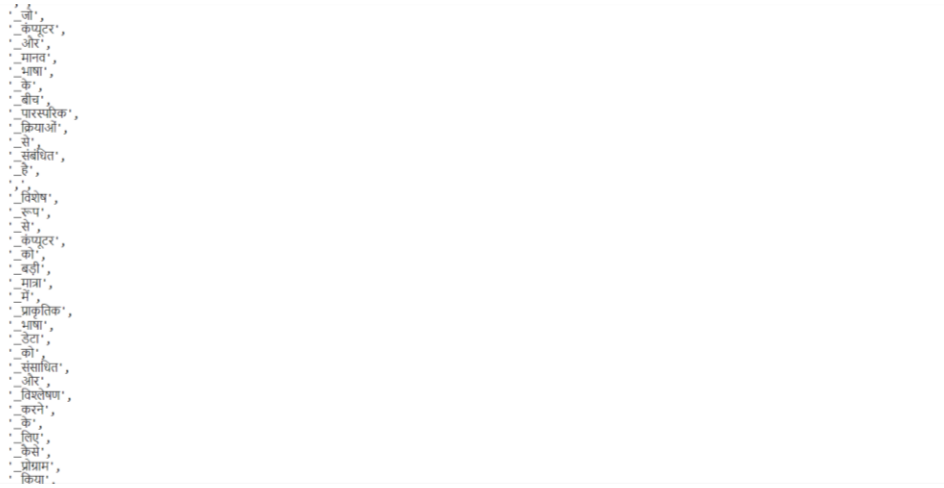


```
from inltk.inltk import tokenize

hindi_text = """प्राकृतिक भाषा प्रसंस्करण भाषा विज्ञान, कंप्यूटर विज्ञान, और कृत्रिम बुद्धिमत्ता का एक उपक्षेत्र है,
                जो कंप्यूटर और मानव भाषा के बीच
                पारस्परिक क्रियाओं से संबंधित है, विशेष रूप से कंप्यूटर
                को बड़ी मात्रा में प्राकृतिक भाषा डेटा को संसाधित और विश्लेषण करने के लिए कैसे प्रोग्राम किया जाता है।"""

# tokenize(input text, language code)
tokenize(hindi_text, "hi")
```

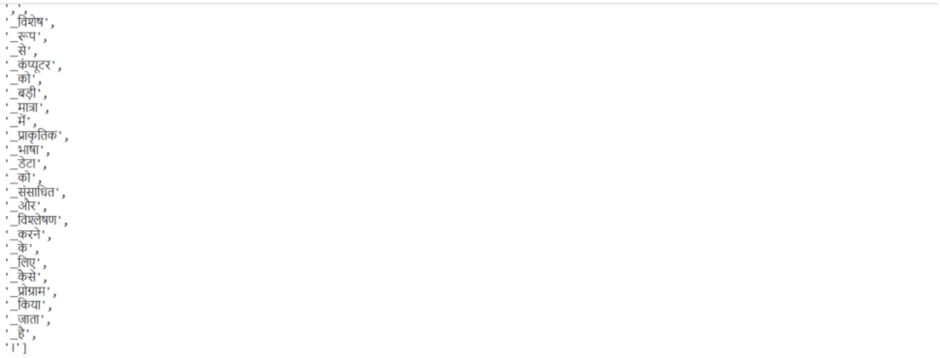
```
['_प्राकृतिक',
 '_भाषा',
 '_प्रसंस्करण',
 '_भाषा',
 '_विज्ञान',
 ',',
 '_कंप्यूटर',
 '_विज्ञान',
 ',',
 '_और',
 '_कृत्रिम',
 '_बुद्धिमत्ता',
 '_का',
 '_एक',
 '_उपक्षेत्र',
 '_है',
 ',',
 '_जो',
 '_कंप्यूटर',
 '_और',
 '_मानव',
```



```
 '_कंप्यूटर',
 '_विज्ञान',
 ',',
 '_और',
 '_कृत्रिम',
 '_बुद्धिमत्ता',
 '_का',
 '_एक',
 '_उपक्षेत्र',
 '_है',
 ',',
 '_जो',
 '_कंप्यूटर',
 '_और',
 '_मानव',
 '_भाषा',
 '_के',
 '_बीच',
 '_पारस्परिक',
 '_क्रियाओं',
 '_से',
 '_संबंधित',
 '_है',
 ',',
 '_विशेष',
 '_रूप',
 '_से',
 '_कंप्यूटर',
 '_को',
 '_बड़ी',
 '_मात्रा',
 '_में',
 '_प्राकृतिक',
 '_भाषा',
```



```
',',
'_जो',
'_कंप्यूटर',
'_और',
'_मानव',
'_भाषा',
'_के',
'_बीच',
'_पारस्परिक',
'_क्रियाओं',
'_से',
'_संबंधित',
'_है',
',',
'_विशेष',
'_रूप',
'_से',
'_कंप्यूटर',
'_को',
'_बड़ी',
'_मात्रा',
'_में',
'_प्राकृतिक',
'_भाषा',
'_डेटा',
'_को',
'_संसाधित',
'_और',
'_विश्लेषण',
'_करने',
'_के',
'_लिए',
'_कैसे',
'_प्रोग्राम',
'_किया',
```



```
',',
'_विशेष',
'_रूप',
'_से',
'_कंप्यूटर',
'_को',
'_बड़ी',
'_मात्रा',
'_में',
'_प्राकृतिक',
'_भाषा',
'_डेटा',
'_को',
'_संसाधित',
'_और',
'_विश्लेषण',
'_करने',
'_के',
'_लिए',
'_कैसे',
'_प्रोग्राम',
'_किया',
'_जाता',
'_है',
'।']
```

Conclusion –

In this assignment, we have done natural language processing on three different languages using different python libraries.