NATURAL LANGUAGE PROCESSING

PRACTICAL NO 1

A. CONVERT SPEECH TO TEXT

```
C:\Users\sushmita>pip install  nltk
Requirement already satisfied: nltk in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages
Requirement already satisfied: regex>=2021.8.3 in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from nltk)
Requirement already satisfied: tqdm in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from nltk)
Requirement already satisfied: click in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from nltk)
Requirement already satisfied: joblib in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from nltk)
Requirement already satisfied: colorama; platform_system == "Windows" in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from tqdm->nltk)
Requirement already satisfied: importlib-metadata; python_version < "3.8" in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from click->nlt
k)
Requirement already satisfied: typing-extensions>=3.6.4; python_version < "3.8" in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from impo
rtlib-metadata; python_version < "3.8"->click->nltk)
Requirement already satisfied: zipp>=0.5 in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from importlib-metadata; python_version < "3.8"-
>click->nltk)
You are using pip version 9.0.1, however version 22.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

```
C:\Users\sushmita>pip install gtts
Requirement already satisfied: gtts in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages
Requirement already satisfied: click in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from gtts)
Requirement already satisfied: six in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from gtts)
Requirement already satisfied: requests in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from gtts)
Requirement already satisfied: colorama; platform_system == "Windows" in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from click->gtts)
Requirement already satisfied: importlib-metadata; python_version < "3.8" in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from click->gtt
s)
Requirement already satisfied: idna<4,>=2.5; python_version >= "3" in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from requests->gtts)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from requests->gtts)
Requirement already satisfied: charset-normalizer~=2.0.0; python_version >= "3" in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from requ
ests->gtts)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from requests->gtts)
Requirement already satisfied: typing-extensions>=3.6.4; python_version < "3.8" in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from impo
rtlib-metadata; python_version < "3.8"->click->gtts)
Requirement already satisfied: zipp>=0.5 in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages (from importlib-metadata; python_version < "3.8"-
>click->gtts)
You are using pip version 9.0.1, however version 22.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

Pract-1-nlp.py - D:\master\Part2\New folder\NLP\Pract-1-nlp.py (3.6.0)

File Edit Format Run Options Window Help

```
import speech_recognition as sr
filename="file1.wav"
r=sr.Recognizer()
with sr.AudioFile(filename) as source:
    audio_data=r.record(source)
    text=r.recognize_google(audio_data)
    print(text)
```

OUTPUT

```
========== RESTART: D:\master\Part2\New folder\NLP\Pract-1
talking nonsense
>>>
```

NATURAL LANGUAGE PROCESSING

B. CONVERT TEXT TO SPEECH

```
C:\Users\sushmita>pip install playsound
Requirement already satisfied: playsound in c:\users\sushmita\appdata\local\programs\python\python36\lib\site-packages
You are using pip version 9.0.1, however version 22.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

Pract 1b-nlp.py - D:\master\Part2\New folder\NLP\Pract 1b-nlp.py (3.6.0)

File Edit Format Run Options Window Help

```
from playsound import playsound
from gtts import gTTS
mytext="Welcome to Natural Language Processing"
language="en"
myobj=gTTS(text=mytext,lang=language,slow=False)
myobj.save("myfile.mp3")
playsound("myfile.mp3")
```

OUTPUT

NATURAL LANGUAGE PROCESSING

PRACTICAL NO 2

A. STUDY OF VARIOUS CORPUS (BROWN CORPUS) WITH DIFFERENT METHODS LIKE FIELDS, RAW,WORDS, SENTENCE, CATEGORY

Practical 2A Study of Brown corpus with various methods like fields,words,sent **etc**

```
[ ] import nltk
```

```
[ ] from nltk.corpus import brown
```

```
[ ] nltk.download('brown')
```

```
[nltk_data] Downloading package brown to /root/nltk_data...
[nltk_data]   Unzipping corpora/brown.zip.
True
```

```
[ ] print(brown.fileids())
```

```
['ca01', 'ca02', 'ca03', 'ca04', 'ca05', 'ca06', 'ca07', 'ca08', 'ca09', 'ca10', 'ca11', 'ca12', 'ca13', 'ca14',
```

```
[ ] ca01=brown.words('ca01')
```

```
[ ] print(ca01)
```

```
['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', ...]
```

```
[ ] print(len(ca01))
```

```
2242
```

```
[ ] cr07=brown.words('cr07')
```

```
[ ] print(cr07)
```

```
['One', 'day', ',', 'the', 'children', 'had', 'wanted', ...]
```

```
[ ] print(len(cr07))
```

```
2456
```

NATURAL LANGUAGE PROCESSING

```
[ ] print(brown.categories())

['adventure', 'belles_lettres', 'editorial', 'fiction', 'government', 'hobbies', 'humor', 'learned', 'lore', 'mystery', 'ne
```

```
[ ] for fileid in brown.fileids():
      num_chars=len(brown.raw(fileid))
      num_words=len(brown.words(fileid))
      num_sents=len(brown.sents(fileid))
      print(fileid, '\t\t',num_chars,'\t\t',num_words,'\t\t',num_sents)

ca01            20187           2242            98
ca02            20357           2277            98
ca03            20214           2275            112
ca04            20608           2217            88
ca05            19748           2244            84
ca06            20137           2263            99
ca07            20921           2270            120
ca08            19599           2187            101
ca09            20331           2234            113
ca10            20128           2282            106
ca11            19728           2259            100
ca12            19633           2338            104
ca13            19059           2241            108
```

B. SIMULATIG SENTENCE TOKENIZATION AND WORD TOKENIZATION

```
2B-Study of tokenization

[ ] from nltk import tokenize

[ ] nltk.download('punkt')

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True

[ ] para='Tokenization breaks the raw text into words, sentences called tokens. These tokens help in understanding the context or developing the model for the NLP. T

[ ] sents=tokenize.sent_tokenize(para)
    print(sents)

['Tokenization breaks the raw text into words, sentences called tokens.', 'These tokens help in understanding the context or developing the model for the NLP.',

[ ] nltk.download('words')

[nltk_data] Downloading package words to /root/nltk_data...
[nltk_data]   Unzipping corpora/words.zip.
True
```

```
[ ] for index in range(len(sents)):
      words=tokenize.word_tokenize(sents[index])
      print(words)

['Tokenization', 'breaks', 'the', 'raw', 'text', 'into', 'words', ',', 'sentences', 'called', 'tokens', '.']
['These', 'tokens', 'help', 'in', 'understanding', 'the', 'context', 'or', 'developing', 'the', 'model', 'for', 'the', 'NLP', '.']
['The', 'tokenization', 'helps', 'in', 'interpreting', 'the', 'meaning', 'of', 'the', 'text', 'by', 'analyzing', 'the', 'sequence', 'of', 'the', 'words', '.']
```

NATURAL LANGUAGE PROCESSING

C. WRITE A PROGRAM TO FIND THE MOST FREQUENT NOUN TAGS

2c- WAP to find the most frequent noun tags

```
[ ] from collections import defaultdict
```

```
[ ] text=nltk.word_tokenize("Nick likes to play football. Nick does not like to play cricket")
```

```
[ ] text
```

```
['Nick',
 'likes',
 'to',
 'play',
 'football',
 '.',
 'Nick',
 'does',
 'not',
 'like',
 'to',
 'play',
 'cricket']
```

```
[ ] nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
True
```

```
tags=nltk.pos_tag(text)
print(tags)
```

```
:k', 'NNP'), ('likes', 'VBZ'), ('to', 'TO'), ('play', 'VB'), ('football', 'NN'), ('.', '.'), ('Nick', 'NNP'), ('does', 'VBZ'), ('not', 'RB'), ('like', 'V
```

```
[ ]
addNounWords=[]
count=0
for words in tags:
  val=tags[count][1]
  if(val=='NN' or val=='NNS' or val=='NNPS' or val=='NNP'):
    addNounWords.append(tags[count][0])
  count+=1
```

```
[ ] print(count)
```

```
13
```

```
[ ] print(addNounWords)
```

```
['Nick', 'football', 'Nick', 'cricket']
```

```
[ ] temp=defaultdict(int)
```

```
[ ] for sub in addNounWords:
      for wrd in sub.split():
        temp[wrd]+=1
```

NATURAL LANGUAGE PROCESSING

```
[ ]  res=max(temp,key=temp.get)

[ ]  print(temp)

     defaultdict(<class 'int'>, {'Nick': 2, 'football': 1, 'cricket': 1})

[ ]  print(str(res))

     Nick
```

D. MAP WORDS TO PROPERTIES USING PYTHON DICTIONARIES

Map words to Properties using pythn dictonaries

```
[ ]  thisdict={
         "brand":"Ford",
         "model":"Mustang",
         "year":"1964"
     }

[ ]  print(thisdict)

     {'brand': 'Ford', 'model': 'Mustang', 'year': '1964'}

[ ]  print(thisdict["year"])

     1964

[ ]  print(len(thisdict))

     3

[ ]  print(type(thisdict))

     <class 'dict'>
```

E. IMPLEMENT DEFAULT TAGGER

Default Tagger

```
[ ]  from nltk.tag import DefaultTagger

[ ]  exptagger=DefaultTagger('NN')

[ ]  print(exptagger.tag_sents([['hi',','],['HOW','are','you','?']]))

     [[('hi', 'NN'), (',', 'NN')], [('HOW', 'NN'), ('are', 'NN'), ('you', 'NN'), ('?', 'NN')]]
```

NATURAL LANGUAGE PROCESSING

F. IMPLEMENT REGULAR EXPRESSION TAGGER

Regular expression Tagger

```
[ ] from nltk.tag import RegexpTagger
```

```
test_sent=brown.sents(categories='news')[0]
print(test_sent)
```

```
ulton', 'County', 'Grand', 'Jury', 'said', 'Friday', 'an', 'investigation', 'of', "Atlanta's", 'recent', 'primary', 'election', 'produced', '``', 'no', 'evidence',
```

```
[ ] regexptagger=RegexpTagger(
        [(r'.*ble$','JJ'),
         (r'.*ness$','NN'),
         (r'.*ly$','RB'),
         (r'.*ed$','VBD'),
         (r'.*tion$','NN'),
         (r'.*s$','NNS'),
         (r'.*(The|the|A|a|An|an)$','AT'),
         (r'.*','NN')
         ]
    )
```

```
[ ] print(regexptagger)
```

```
<Regexp Tagger: size=8>
```

```
print(regexptagger.tag(test_sent))
```

```
'NN'), ('County', 'NN'), ('Grand', 'NN'), ('Jury', 'NN'), ('said', 'NN'), ('Friday', 'NN'), ('an', 'AT'), ('investigation', 'NN'), ('of', 'NN'), ("Atlanta's", 'NN
```

NATURAL LANGUAGE PROCESSING

PRACTICAL NO 3

A. STUDY OF WORD NET DICTIONARY

3 a Study of word net dictionary with methods as Synsets,Definitions,Examples, Antonyms

```
import nltk
from nltk.corpus import wordnet
```

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

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
True
```

```
[ ] print(wordnet.synsets("computer"))
```

```
[Synset('computer.n.01'), Synset('calculator.n.01')]
```

```
[ ] print(wordnet.synset('computer.n.01').definition())
```

```
a machine for performing calculations automatically
```

```
[ ] print(wordnet.synset('computer.n.01').examples())
```

```
[]
```

```
[ ] print(wordnet.lemma('buy.v.01.buy').antonyms())
```

```
[Lemma('sell.v.01.sell')]
```

B. STUDY OF LEMMA , HYPONYMS & HYPERNYMS

```
[ ] import nltk
    from nltk.corpus import wordnet
```

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

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
True
```

```
[ ] print(wordnet.synsets('computer'))
```

```
[Synset('computer.n.01'), Synset('calculator.n.01')]
```

NATURAL LANGUAGE PROCESSING

```
[ ] print(wordnet.synset('computer.n.01').lemma_names())

    ['computer', 'computing_machine', 'computing_device', 'data_processor', 'electronic_computer', 'information_processing_system']
```

```
[ ] for p in wordnet.synsets('computer'):
      print(f'{p}-------->{p.lemma_names()}')

    Synset('computer.n.01')-------->['computer', 'computing_machine', 'computing_device', 'data_processor', 'electronic_computer', 'information_processing_system']
    Synset('calculator.n.01')-------->['calculator', 'reckoner', 'figurer', 'estimator', 'computer']
```

```
print(wordnet.synset('computer.n.01').lemmas())

'computer.n.01.computer'), Lemma('computer.n.01.computing_machine'), Lemma('computer.n.01.computing_device'), Lemma('computer.n.01.data_processor'), Lemma('comput
```

```
[ ] print(wordnet.lemma('computer.n.01.computing_machine').synset())

    Synset('computer.n.01')

[ ] print(wordnet.lemma('calculator.n.01.figurer').synset())

    Synset('calculator.n.01')

[ ] print(wordnet.lemma('computer.n.01.computing_machine').name())
    print(wordnet.lemma('calculator.n.01.figurer').name())

    computing_machine
    figurer
```

Hyponyms give abstract concept of the word that are much more specific

```
[ ] syn=wordnet.synset('computer.n.01')

[ ] syn

    Synset('computer.n.01')
```

```
[ ] print(syn.hyponyms())

    [Synset('analog_computer.n.01'), Synset('digital_computer.n.01'), Synset('home_computer.n.01'), Synset('node.n.08'), Synset('number_cruncher.n.02'), Synset('par
```

NATURAL LANGUAGE PROCESSING

```
[ ] print(wordnet.synsets('vehicle'))

    [Synset('vehicle.n.01'), Synset('vehicle.n.02'), Synset('vehicle.n.03'), Synset('fomite.n.01')]

[ ] for p in wordnet.synsets('vehicle'):
      print(f'{p}--------->{p.lemma_names()}')

    Synset('vehicle.n.01')--------->['vehicle']
    Synset('vehicle.n.02')--------->['vehicle']
    Synset('vehicle.n.03')--------->['vehicle']
    Synset('fomite.n.01')--------->['fomite', 'vehicle']
```

```
vsyn=wordnet.synset('vehicle.n.01')
print(vsyn.hyponyms())

Synset('bumper_car.n.01'), Synset('craft.n.02'), Synset('military_vehicle.n.01'), Synset('rocket.n.01'), Synset('skibob.n.01'), Synset('sled.n.01'), Synset('steami
```

Hypernym is a word the general meaning Hyponym is in a type of relationship with hypernym and Lemma is just meaning

```
[ ] print(syn.hypernyms())

    [Synset('machine.n.01')]
```

C. COMPARE TWO NOUNS

3c- Compare two nouns

```
[ ] import nltk
    from nltk.corpus import wordnet

[ ] nltk.download('wordnet')

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

[ ] syn1=wordnet.synsets("football")
    syn2=wordnet.synsets("soccer")

    for s1 in syn1:
      for s2 in syn2:
        print("Path similarity for:")
        print(s1,'(',s1.pos(),')','[',s1.definition(),']')
        print(s2,'(',s2.pos(),')','[',s2.definition(),']')
        print(s1.path_similarity(s2))
        print()
```

```
Path similarity for:
Synset('football.n.01') ( n ) [ any of various games played with a ball (round or oval) in which two teams try to kick or carry or propel the ball into each other
Synset('soccer.n.01') ( n ) [ a football game in which two teams of 11 players try to kick or head a ball into the opponents' goal ]
0.5

Path similarity for:
Synset('football.n.02') ( n ) [ the inflated oblong ball used in playing American football ]
Synset('soccer.n.01') ( n ) [ a football game in which two teams of 11 players try to kick or head a ball into the opponents' goal ]
0.05
```

NATURAL LANGUAGE PROCESSING

D. WRITE A PROGRAM TO FIND SYNONYM & ANTONYM OF WORD ACTIVE

3d WAP to find snonym and antonym of word active

```
[ ] from nltk.corpus import wordnet

[ ] print(wordnet.synsets("active"))

    [Synset('active_agent.n.01'), Synset('active_voice.n.01'), Synset('active.n.03'), Synset('active.a.01'), Synset('active.s.02'), Syn

[ ] print(wordnet.synset("active_voice.n.01").lemma_names())

    ['active_voice', 'active']

[ ] print(wordnet.lemma("active.a.01.active").antonyms())

    [Lemma('inactive.a.02.inactive')]
```

E. ADDING AND REMOVING STOPWORDS USING SPACY LIBRARY

3e Adding and Removing Stop Words using Spacy

```
[ ] !pip install spacy

    Requirement already satisfied: spacy in /usr/local/lib/python3.7/dist-packages (2.2.4)
    Requirement already satisfied: plac<1.2.0,>=0.9.6 in /usr/local/lib/python3.7/dist-packages (from spacy) (1.1.3)
    Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from spacy) (3.0.6)
    Requirement already satisfied: catalogue<1.1.0,>=0.0.7 in /usr/local/lib/python3.7/dist-packages (from spacy) (1.0.0)
    Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (0.9.1)
    Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (4.64.0)
    Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from spacy) (2.0.6)
    Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (1.21.6)
    Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (2.23.0)
    Requirement already satisfied: srsly<1.1.0,>=1.0.2 in /usr/local/lib/python3.7/dist-packages (from spacy) (1.0.5)
    Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (1.0.7)
    Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from spacy) (57.4.0)
    Requirement already satisfied: thinc==7.4.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (7.4.0)
    Requirement already satisfied: blis<0.5.0,>=0.4.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (0.4.1)
    Requirement already satisfied: importlib-metadata>=0.20 in /usr/local/lib/python3.7/dist-packages (from catalogue<1.1.0,>=
    Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata
    Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata>=0.20->catalog
```

```
[ ] import spacy

[ ] import nltk

[ ] from nltk.tokenize import word_tokenize

[ ] sp=spacy.load('en_core_web_sm')

[ ] all_stopwords=sp.Defaults.stop_words
```

NATURAL LANGUAGE PROCESSING

```
[ ]  all_stopwords
```

```
{"'d",
 "'ll",
 "'m",
 "'re",
 "'s",
 "'ve",
 'a',
 'about',
 'above',
 'across',
 'after',
 'afterwards',
 'again',
 'against',
 'all',
 'almost',
 'alone',
 'along',
 'already',
 'also',
 'although',
 'always',
 'am',
 'among',
```

```
[ ]  all_stopwords.add ('play')
```

```
all_stopwords
```

```
 'ourselves',
 'out',
 'over',
 'own',
 'part',
 'per',
 'perhaps',
 'play',
 'please',
 'put',
 'quite',
 'rather',
 're',
 'really',
 'regarding',
 'same',
 'say',
```

NATURAL LANGUAGE PROCESSING

```
[ ]  text="Rohan likes to play football,but he is not too fond of tennis"
```

```
[ ]  nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
[ ]  text_tokens=word_tokenize(text)
```

```
[ ]  text_tokens
```

```
['Rohan',
 'likes',
 'to',
 'play',
 'football',
 ',',
 'but',
 'he',
 'is',
 'not',
 'too',
 'fond',
 'of',
 'tennis']
```

```
[ ]  tokens_without_sw=[word for word in text_tokens if not word in all_stopwords]
```

```
[ ]  tokens_without_sw
```

```
['Rohan', 'likes', 'football', ',', 'fond', 'tennis']
```

```
[ ]  all_stopwords.remove('not')
```

```
[ ]  tokens_without_sw=[word for word in text_tokens if not word in all_stopwords]
```

```
[ ]  tokens_without_sw
```

```
['Rohan', 'likes', 'football', ',', 'not', 'fond', 'tennis']
```

NATURAL LANGUAGE PROCESSING

PRACTICAL NO 4

A. TEXT TOKENIZATION USING PYTHONS SPLIT FUNCTION

```
4A Text Tokenization using pythons split function

[ ] text="This tool is a beta storage. Alexa developers can use Get Metrics API to seamlessly analyse m

[ ] data=text.split('.')

[ ] for i in data:
      print(i)

    This tool is a beta storage
     Alexa developers can use Get Metrics API to seamlessly analyse metric
     It also supports custom skill model
     You can use this tool for model creation
```

B. TEXT TOKENIZATION USING REGULAR EXPRESSION

```
4B Text Tokenization using Regular Expression

[ ] from nltk.tokenize import RegexpTokenizer

[ ] tk=RegexpTokenizer('\s+',gaps=True)

[ ] str="I like to study machine learning based subjects in python"

[ ] tokens=tk.tokenize(str)

[ ] tokens

    ['I',
     'like',
     'to',
     'study',
     'machine',
     'learning',
     'based',
     'subjects',
     'in',
     'python']
```

NATURAL LANGUAGE PROCESSING

C. TOKENIZATION USING NLTK

4C Tokenization using nltk

```
[ ] import nltk

[ ] from nltk.tokenize import word_tokenize

[ ] nltk.download('punkt')

    [nltk_data] Downloading package punkt to /root/nltk_data...
    [nltk_data]   Unzipping tokenizers/punkt.zip.
    True

[ ] str="I like to study machine learning based subjects in python"
```

```
[ ] tokens2=word_tokenize(str)

    tokens2

    ['I',
     'like',
     'to',
     'study',
     'machine',
     'learning',
     'based',
     'subjects',
     'in',
     'python']
```

NATURAL LANGUAGE PROCESSING

D. TOKENIZATION USING SPACY LIBRARY

4 D Tokenization using spacy library

```
[ ] import spacy

[ ] nlp=spacy.blank("en")

[ ] str="I like to study machine learning based subjects in python"

[ ] doc=nlp(str)

[ ] for word in doc:
      print(word.text)
```

```
I
like
to
study
machine
learning
based
subjects
in
python
```

E. TOKENIZATION USING KERAS

```
[ ] !pip install keras

Requirement already satisfied: keras in /usr/local/lib/python3.7/dist-packages (2.8.0)

[ ] !pip install tensorflow

Requirement already satisfied: tensorflow in /usr/local/lib/python3.7/dist-packages (2.8.0)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.14.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (4.2.0)
Requirement already satisfied: gast>=0.2.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.5.3)
Requirement already satisfied: absl-py>=0.4.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.0.0)
Requirement already satisfied: tensorboard<2.9,>=2.8 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (2.8.0)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.3.0)
Collecting tf-estimator-nightly==2.8.0.dev2021122109
  Downloading tf_estimator_nightly-2.8.0.dev2021122109-py2.py3-none-any.whl (462 kB)
     |████████████████████████████████| 462 kB 5.1 MB/s
```

NATURAL LANGUAGE PROCESSING

```
[ ] import keras

[ ] from keras.preprocessing.text import text_to_word_sequence

    text="This tool is a beta storage. Alexa developers can use Get Metrics API to seamlessly analyse metric. It also supports custom skill model.

[ ] text

    'This tool is a beta storage. Alexa developers can use Get Metrics API to seamlessly analyse metric. It also supports custom skill model. You c
    or model creation. '

[ ] tokens=text_to_word_sequence(text)

[ ] print(tokens)

    ['this', 'tool', 'is', 'a', 'beta', 'storage', 'alexa', 'developers', 'can', 'use', 'get', 'metrics', 'api', 'to', 'seamlessly', 'analyse', 'me
```

F. TOKENIZATION USING GENSIM

```
[ ] !pip install gensim

    Requirement already satisfied: gensim in /usr/local/lib/python3.7/dist-packages (3.6.0)
    Requirement already satisfied: scipy>=0.18.1 in /usr/local/lib/python3.7/dist-packages (from gensim) (1.4.1)
    Requirement already satisfied: numpy>=1.11.3 in /usr/local/lib/python3.7/dist-packages (from gensim) (1.21.6)
    Requirement already satisfied: six>=1.5.0 in /usr/local/lib/python3.7/dist-packages (from gensim) (1.15.0)
    Requirement already satisfied: smart-open>=1.2.1 in /usr/local/lib/python3.7/dist-packages (from gensim) (6.0.0)

[ ] from gensim.utils import tokenize

    str="I like to study machine learning subjects"

[ ] list(tokenize(str))

    ['I', 'like', 'to', 'study', 'machine', 'learning', 'subjects']
```

NATURAL LANGUAGE PROCESSING

PRACTICAL NO 5

A. IMPORT NLP LIBRARIES FOR WORKING WITH INDIAN LANGUAGES AND PERFORM A- WORK TOKENIZATION FOR HINDI TEXT

5 Import NLP libraries for working with Indian Languages and perform A- Work tokenization for Hindi text

```
[ ] ! pip install torch==1.3.1+cpu -f https://download.pytorch.org/whl/torch_stable.html

Looking in links: https://download.pytorch.org/whl/torch_stable.html
Collecting torch==1.3.1+cpu
  Downloading https://download.pytorch.org/whl/cpu/torch-1.3.1%2Bcpu-cp37-cp37m-linux_x86_64.whl (111.8 MB)
     |████████████████████████████████| 111.8 MB 1.2 MB/s
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from torch==1.3.1+cpu) (1.21.6)
Installing collected packages: torch
  Attempting uninstall: torch
    Found existing installation: torch 1.11.0+cu113
    Uninstalling torch-1.11.0+cu113:
      Successfully uninstalled torch-1.11.0+cu113
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This
torchvision 0.12.0+cu113 requires torch==1.11.0, but you have torch 1.3.1+cpu which is incompatible.
torchtext 0.12.0 requires torch==1.11.0, but you have torch 1.3.1+cpu which is incompatible.
torchaudio 0.11.0+cu113 requires torch==1.11.0, but you have torch 1.3.1+cpu which is incompatible.
Successfully installed torch-1.3.1+cpu
```

```
[ ] !pip install inltk

Collecting async-timeout>=3.0.1
  Downloading async_timeout-4.0.2-py3-none-any.whl (5.8 kB)
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (from inltk) (1.3.5)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from inltk) (1.4.1)
Requirement already satisfied: spacy>=2.0.18 in /usr/local/lib/python3.7/dist-packages (from inltk) (2.2.4)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from inltk) (3.2.2)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from inltk) (2.23.0)
Collecting sentencepiece
  Downloading sentencepiece-0.1.96-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.2 MB)
     |████████████████████████████████| 1.2 MB 16.7 MB/s
Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from inltk) (7.1.2)
Requirement already satisfied: fastprogress>=0.1.19 in /usr/local/lib/python3.7/dist-packages (from inltk) (1.0.2)
Collecting aiohttp>=3.5.4
  Downloading aiohttp-3.8.1-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_64.whl (1.1 MB)
     |████████████████████████████████| 1.1 MB 46.1 MB/s
```

```
[ ] !pip install tornado==4.5.3

Collecting tornado==4.5.3
  Downloading tornado-4.5.3.tar.gz (484 kB)
     |████████████████████████████████| 484 kB 8.1 MB/s
Building wheels for collected packages: tornado
  Building wheel for tornado (setup.py) ... done
  Created wheel for tornado: filename=tornado-4.5.3-cp37-cp37m-linux_x86_64.whl
  Stored in directory: /root/.cache/pip/wheels/a2/45/43/36ec7a893e16c1212a6b15
Successfully built tornado
Installing collected packages: tornado
  Attempting uninstall: tornado
    Found existing installation: tornado 5.1.1
    Uninstalling tornado-5.1.1:
      Successfully uninstalled tornado-5.1.1
ERROR: pip's dependency resolver does not currently take into account all the
google-colab 1.0.0 requires tornado~=5.1.0; python_version >= "3.0", but you ha
bokeh 2.3.3 requires tornado>=5.1, but you have tornado 4.5.3 which is incompat
Successfully installed tornado-4.5.3
```

NATURAL LANGUAGE PROCESSING

```
[ ] from inltk.inltk import setup

[ ] setup('hi')

    Downloading Model. This might take time, depending on your internet connection. Please be patient.
    We'll only do this for the first time.
    Downloading Model. This might take time, depending on your internet connection. Please be patient.
    We'll only do this for the first time.
    Done!

[ ] hindi_text="कपड़े  सुखा दो।"

[ ] from inltk.inltk import tokenize

[ ] tokenize(hindi_text,"hi")

    ['_कपड़े', '_सुखा', '_दो', '।']
```

B. GENERATE SIMILAR SENTENCES FROM A GIVEN HINDI TEXT INPUT

```
[ ] !pip install torch==1.2.0+cu92 torchvision==0.4.0+cu92 -f https://download.pytorch.org/whl/torch_stable.html

    Looking in links: https://download.pytorch.org/whl/torch_stable.html
    Collecting torch==1.2.0+cu92
      Downloading https://download.pytorch.org/whl/cu92/torch-1.2.0%2Bcu92-cp37-cp37m-manylinux1_x86_64.whl (663.1 MB)
         |████████████████████████████████| 663.1 MB 1.7 kB/s
    Collecting torchvision==0.4.0+cu92
      Downloading https://download.pytorch.org/whl/cu92/torchvision-0.4.0%2Bcu92-cp37-cp37m-manylinux1_x86_64.whl (8.8 MB)
         |████████████████████████████████| 8.8 MB 24.1 MB/s
    Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from torch==1.2.0+cu92) (1.21.6)
    Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from torchvision==0.4.0+cu92) (1.15.0)
    Requirement already satisfied: pillow>=4.1.1 in /usr/local/lib/python3.7/dist-packages (from torchvision==0.4.0+cu92) (7.1.2)
    Installing collected packages: torch, torchvision
      Attempting uninstall: torch
        Found existing installation: torch 1.11.0+cu113
        Uninstalling torch-1.11.0+cu113:
          Successfully uninstalled torch-1.11.0+cu113
      Attempting uninstall: torchvision
        Found existing installation: torchvision 0.12.0+cu113
        Uninstalling torchvision-0.12.0+cu113:
          Successfully uninstalled torchvision-0.12.0+cu113
    ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is th
    torchtext 0.12.0 requires torch==1.11.0, but you have torch 1.2.0+cu92 which is incompatible.
    torchaudio 0.11.0+cu113 requires torch==1.11.0, but you have torch 1.2.0+cu92 which is incompatible.
    Successfully installed torch-1.2.0+cu92 torchvision-0.4.0+cu92
```

```
[ ] pip install inltk

    Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from inltk) (7.1.2)
    Collecting fastai==1.0.57
      Downloading fastai-1.0.57-py3-none-any.whl (233 kB)
         |████████████████████████████████| 233 kB 52.2 MB/s
    Requirement already satisfied: torch>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from fastai==1.0.57->inltk) (1.2.0+cu92)
    Requirement already satisfied: torchvision in /usr/local/lib/python3.7/dist-packages (from fastai==1.0.57->inltk) (0.4.0+cu92)
    Collecting multidict<7.0,>=4.5
      Downloading multidict-6.0.2-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (94 kB)
         |████████████████████████████████| 94 kB 3.5 MB/s
    Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.7/dist-packages (from aiohttp>=3.5.4->inltk) (21.4.0)
    Collecting asynctest==0.13.0
      Downloading asynctest-0.13.0-py3-none-any.whl (26 kB)
    Collecting yarl<2.0,>=1.0
```

NATURAL LANGUAGE PROCESSING

```
!pip install tornado==4.5.3
```

```
Collecting tornado==4.5.3
  Downloading tornado-4.5.3.tar.gz (484 kB)
     |████████████████████████████████| 484 kB 30.1 MB/s
Building wheels for collected packages: tornado
  Building wheel for tornado (setup.py) ... done
  Created wheel for tornado: filename=tornado-4.5.3-cp37-cp37m-linux_x
  Stored in directory: /root/.cache/pip/wheels/a2/45/43/36ec7a893e16c1
Successfully built tornado
Installing collected packages: tornado
  Attempting uninstall: tornado
    Found existing installation: tornado 5.1.1
    Uninstalling tornado-5.1.1:
      Successfully uninstalled tornado-5.1.1
ERROR: pip's dependency resolver does not currently take into account
google-colab 1.0.0 requires tornado~=5.1.0; python_version >= "3.0", b
bokeh 2.3.3 requires tornado>=5.1, but you have tornado 4.5.3 which is
Successfully installed tornado-4.5.3
```

```
[ ] from inltk.inltk import get_similar_sentences
```

```
[ ] from inltk.inltk import setup
```

```
[ ] setup('hi')
```

```
Downloading Model. This might take time, depending on your internet connection. Please be patient.
We'll only do this for the first time.
Downloading Model. This might take time, depending on your internet connection. Please be patient.
We'll only do this for the first time.
Done!
```

```
[ ] output=get_similar_sentences(' मैं आज बहुत खुश हूँ',5,'hi')
```

```
/usr/local/lib/python3.7/dist-packages/torch/serialization.py:453: SourceChangeWarning: source code of cla
  warnings.warn(msg, SourceChangeWarning)
/usr/local/lib/python3.7/dist-packages/torch/serialization.py:453: SourceChangeWarning: source code of cla
  warnings.warn(msg, SourceChangeWarning)
/usr/local/lib/python3.7/dist-packages/torch/serialization.py:453: SourceChangeWarning: source code of cla
  warnings.warn(msg, SourceChangeWarning)
```

```
[ ] print(output)
```

```
['मैं आज काफ़ी खुश हूँ', 'मैं आज काफी खुश हूँ', 'मैं सदैव बहुत खुश हूँ', 'मैं आज अत्यंत खुश हूँ', 'मैं आज अत्यंत खुश हूँ']
```

C. WRITE A PROGRAM TO IDENTIFY INDIAN LANGUAGE OF A TEXT

5c- WAP to identify Indian language of a text

```
setup('bn')
```

```
Downloading Model. This might take time, depending on your internet connection. Please be patient.
We'll only do this for the first time.
Downloading Model. This might take time, depending on your internet connection. Please be patient.
We'll only do this for the first time.
Done!
```

```
from inltk.inltk import identify_language
```

```
identify_language('তুমি খুবই মিষ্টি')
```

```
Downloading Model. This might take time, depending on your internet connection. Please be patient.
We'll only do this for the first time.
Downloading Model. This might take time, depending on your internet connection. Please be patient.
We'll only do this for the first time.
/usr/local/lib/python3.7/dist-packages/torch/serialization.py:453: SourceChangeWarning: source code of class 'torch.nn.modules
  warnings.warn(msg, SourceChangeWarning)
/usr/local/lib/python3.7/dist-packages/torch/serialization.py:453: SourceChangeWarning: source code of class 'fastai.text.lear
  warnings.warn(msg, SourceChangeWarning)
/usr/local/lib/python3.7/dist-packages/torch/serialization.py:453: SourceChangeWarning: source code of class 'fastai.text.mode
  warnings.warn(msg, SourceChangeWarning)
```

```
  warnings.warn(msg, SourceChangeWarning)
/usr/local/lib/python3.7/dist-packages/torch/serialization.py:453: SourceChangeWarning: so
  warnings.warn(msg, SourceChangeWarning)
/usr/local/lib/python3.7/dist-packages/torch/serialization.py:453: SourceChangeWarning: so
  warnings.warn(msg, SourceChangeWarning)
/usr/local/lib/python3.7/dist-packages/torch/serialization.py:453: SourceChangeWarning: so
  warnings.warn(msg, SourceChangeWarning)
/usr/local/lib/python3.7/dist-packages/torch/serialization.py:453: SourceChangeWarning: so
  warnings.warn(msg, SourceChangeWarning)
/usr/local/lib/python3.7/dist-packages/torch/serialization.py:453: SourceChangeWarning: so
  warnings.warn(msg, SourceChangeWarning)
/usr/local/lib/python3.7/dist-packages/torch/serialization.py:453: SourceChangeWarning: so
  warnings.warn(msg, SourceChangeWarning)
/usr/local/lib/python3.7/dist-packages/torch/serialization.py:453: SourceChangeWarning: so
  warnings.warn(msg, SourceChangeWarning)
/usr/local/lib/python3.7/dist-packages/torch/serialization.py:453: SourceChangeWarning: so
  warnings.warn(msg, SourceChangeWarning)
/usr/local/lib/python3.7/dist-packages/torch/serialization.py:453: SourceChangeWarning: so
  warnings.warn(msg, SourceChangeWarning)
/usr/local/lib/python3.7/dist-packages/torch/serialization.py:453: SourceChangeWarning: so
  warnings.warn(msg, SourceChangeWarning)
'bengali'
```

NATURAL LANGUAGE PROCESSING

PRACTICAL NO 6

STUDY OF DIFFERENT STEMMERS AND WORDNETLEMMATIZER

Practical 6 Study of different stemmers and wordnetlemmatizer

```
[ ] import nltk

[ ] from nltk.stem import PorterStemmer

    word_stemmer=PorterStemmer()

[ ] print(word_stemmer.stem('naturing'))

    natur
```

LancasterStemmer

```
[ ] import nltk

[ ] from nltk.stem import LancasterStemmer

[ ] word_stemmer1=LancasterStemmer()

[ ] print(word_stemmer1.stem('naturing'))

    nat
```

Regular Expression

```
[ ] import nltk

[ ] from nltk.stem import RegexpStemmer

[ ] reg_stemmer=RegexpStemmer('ing$|s$|e$|able',min=4)

[ ] print(reg_stemmer.stem('capable'))

    cap
```

NATURAL LANGUAGE PROCESSING

SnowballStemmer

```
[ ] import nltk

[ ] from nltk.stem import SnowballStemmer

[ ] eng_stemmer=SnowballStemmer('english')

[ ] print(eng_stemmer.stem('writing'))

    write
```

WordNetLemmatizer

```
[ ] import nltk

[ ] from nltk.stem import WordNetLemmatizer

[ ] nltk.download('wordnet')

    [nltk_data] Downloading package wordnet to /root/nltk_data...
    [nltk_data]   Unzipping corpora/wordnet.zip.
    True
```

```
[ ] lemmatizer=WordNetLemmatizer()

[ ] from nltk.classify.rte_classify import lemmatize
    print("word:\tlemma")
    print("rocks:",lemmatizer.lemmatize("rocks"))
    print("corpora:",lemmatizer.lemmatize("corpora"))
    print("writing:",lemmatizer.lemmatize("writing",pos="v"))
    print("better:",lemmatizer.lemmatize("better",pos="a"))

    word:   lemma
    rocks: rock
    corpora: corpus
    writing: write
    better: good
```

NATURAL LANGUAGE PROCESSING

PRACTICAL NO 7

ILLUSTRATE POS TAGGING

A. SENTENCE TOKENIZATION,WORD TOKENIZATION,POS TAGGING AND CHUNKING

```
[ ]  import nltk
     from nltk import tokenize
     nltk.download('punkt')
     from nltk import tag
     from nltk import chunk
```

```
     [nltk_data] Downloading package punkt to /root/nltk_data...
     [nltk_data]   Unzipping tokenizers/punkt.zip.
```

```
[ ]  nltk.download('averaged_perceptron_tagger')
     nltk.download('maxent_ne_chunker')
     nltk.download('words')
```

```
     [nltk_data] Downloading package averaged_perceptron_tagger to
     [nltk_data]     /root/nltk_data...
     [nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
     [nltk_data] Downloading package maxent_ne_chunker to
     [nltk_data]     /root/nltk_data...
     [nltk_data]   Unzipping chunkers/maxent_ne_chunker.zip.
     [nltk_data] Downloading package words to /root/nltk_data...
     [nltk_data]   Unzipping corpora/words.zip.
     True
```

```
[ ]  para="Hello! My name is Patra Sushmita. Today we're learning concepts on NLP"
```

```
[ ]  sents=tokenize.sent_tokenize(para)
```

```
[ ]  sents
```

```
     ['Hello!',
      'My name is Patra Sushmita.',
      "Today we're learning concepts on NLP"]
```

NATURAL LANGUAGE PROCESSING

```
tagged_words=[]
tree=[]
for index in range(len(sents)):
  words=tokenize.word_tokenize(sents[index])
  tagged_words.append(tag.pos_tag(words))
  tree.append(chunk.ne_chunk(tagged_words[index]))
  print(words)
```

```
['Hello', '!']
['My', 'name', 'is', 'Patra', 'Sushmita', '.']
['Today', 'we', "'re", 'learning', 'concepts', 'on', 'NLP']
```

```
[ ] tagged_words

[[('Hello', 'NN'), ('!', '.')],
 [('My', 'PRP$'),
  ('name', 'NN'),
  ('is', 'VBZ'),
  ('Patra', 'NNP'),
  ('Sushmita', 'NNP'),
  ('.', '.')],
 [('Today', 'NN'),
  ('we', 'PRP'),
  ("'re", 'VBP'),
  ('learning', 'VBG'),
  ('concepts', 'NNS'),
  ('on', 'IN'),
  ('NLP', 'NNP')]]
```

```
[ ] tree

[Tree('S', [Tree('GPE', [('Hello', 'NN')]), ('!', '.')]),
 Tree('S', [('My', 'PRP$'), ('name', 'NN'), ('is', 'VBZ'), Tree('PERSON', [('Patra', 'NNP'), ('Sushmita', 'NNP')]), ('.', '.')]),
 Tree('S', [('Today', 'NN'), ('we', 'PRP'), ("'re", 'VBP'), ('learning', 'VBG'), ('concepts', 'NNS'), ('on', 'IN'), Tree('ORGANIZATION', [('NLP', 'NNP')])])]
```

NATURAL LANGUAGE PROCESSING

B. NAMED ENTITY RECOGNITION USING USER DEFINED TEXT

```
[ ] !pip install -U spacy

    Requirement already satisfied: spacy in /usr/local/lib/python3.7/dist-packages (2.2.4)
    Collecting spacy
      Downloading spacy-3.3.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (6.2 MB)
         |████████████████████████████████| 6.2 MB 5.3 MB/s
    Requirement already satisfied: blis<0.8.0,>=0.4.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (0.4.1)
    Collecting catalogue<2.1.0,>=2.0.6
      Downloading catalogue-2.0.7-py3-none-any.whl (17 kB)
    Collecting srsly<3.0.0,>=2.4.3
      Downloading srsly-2.4.3-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (457 kB)
         |████████████████████████████████| 457 kB 54.7 MB/s
    Collecting langcodes<4.0.0,>=3.2.0
      Downloading langcodes-3.3.0-py3-none-any.whl (181 kB)
         |████████████████████████████████| 181 kB 58.0 MB/s
    Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from spacy) (2.0.6)
    Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from spacy) (3.0.6)
    Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (21.3)
    Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.7/dist-packages (from spacy) (2.23.0)
```

```
[ ] !python -m spacy download en_core_web_sm

    Collecting en-core-web-sm==3.3.0
      Downloading https://github.com/explosion/spacy-models/releases/download/en
         |████████████████████████████████| 12.8 MB 5.3 MB/s
    Requirement already satisfied: spacy<3.4.0,>=3.3.0.dev0 in /usr/local/lib/py
    Requirement already satisfied: jinja2 in /usr/local/lib/python3.7/dist-packa
    Requirement already satisfied: pydantic!=1.8,!=1.8.1,<1.9.0,>=1.7.4 in /usr/
    Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/d
    Requirement already satisfied: blis<0.8.0,>=0.4.0 in /usr/local/lib/python3.
    Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/pyt
    Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/p
    Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.9 in /usr/local/lib/
```

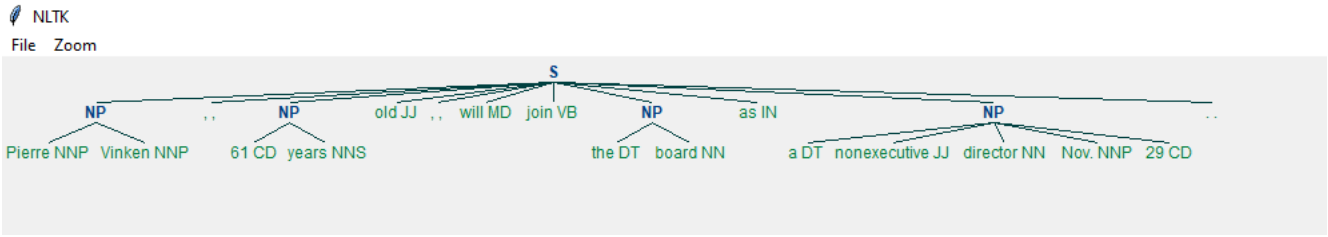
```
[ ] import spacy

[ ] nlp=spacy.load("en_core_web_sm")

[ ] text=("When Sebastin thrun started working on self driving cars at "
    "Google in 2007, few people outside of the company took him"
    "seriously")

[ ] text

    'When Sebastin thrun started working on self driving cars at Google in 2007, few people outside of the company took himseriously'

[ ] doc=nlp(text)

[ ] print("Noun phrases:",[chunk.text for chunk in doc.noun_chunks])
    print("Verbs:",[token.lemma_ for token in doc if token.pos_ == 'VERB'])

    Noun phrases: ['Sebastin thrun', 'self driving cars', 'Google', 'few people', 'the company']
    Verbs: ['start', 'work', 'drive', 'take']
```

NATURAL LANGUAGE PROCESSING

C. DEMONSTRATION OF STATISTICAL PARSER TO CREATE PARSE TREE

File Edit Format Run Options Window Help

```
import nltk
#nltk.download('treebank')
from nltk.corpus import treebank_chunk
treebank_chunk.tagged_sents()[0]
treebank_chunk.chunked_sents()[0]
treebank_chunk.chunked_sents()[0].draw()
```



NATURAL LANGUAGE PROCESSING

PRACTICAL NO 8

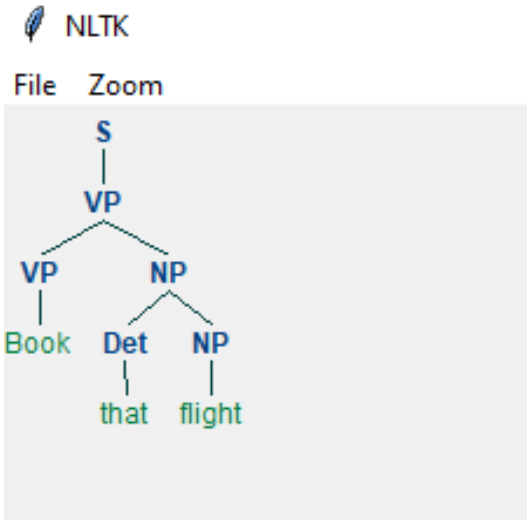
DEFINE GRAMMER USING NLTK. ANALYSE A SENTENCE USING THE SAME

pract-8- nlp.py - D:\master\Part2\New folder\NLP\pract-8- nlp.py (3.6.0)

File Edit Format Run Options Window Help

```
#Define grammer using nltk. Analyse a sentence using the same
import nltk
from nltk import tokenize
grammar1 = nltk.CFG.fromstring("""
S -> VP
 VP -> VP NP
 NP -> Det NP
 Det -> 'that'
 NP -> singular Noun
 NP -> 'flight'
 VP -> 'Book'
 """)
sentence="Book that flight"
nltk.download('punkt')
for index in range (len(sentence)):
    all_tokens=tokenize.word_tokenize(sentence)
print(all_tokens)

parse=nltk.ChartParser(grammar1)
for tree in parse.parse(all_tokens):
    print(tree)
    tree.draw()
```

```
========== RESTART: D:\master\Part2\New folder\NLP\pract-8- nlp.py ======
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\sushmita\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
['Book', 'that', 'flight']
(S (VP (VP Book) (NP (Det that) (NP flight))))
```



NATURAL LANGUAGE PROCESSING

PRACTICAL NO 9

MWE IN NLP

Practical 9 MWE in NLP

```
[1] import nltk
```

```
[2] from nltk.tokenize import MWETokenizer
```

```
[3] from nltk import sent_tokenize,word_tokenize
```

```
[4] nltk.download('punkt')
```

```
    [nltk_data] Downloading package punkt to /root/nltk_data...
    [nltk_data]   Unzipping tokenizers/punkt.zip.
    True
```

```
[5] s="Good cake cost Rs.1500\kg in Mumbai.Please buy me one of them.\n\nThanks."
```

```
[6] s
```

```
    'Good cake cost Rs.1500\\kg in Mumbai.Please buy me one of them.\n\nThanks.'
```

```
[7] mwe=MWETokenizer([('New' 'York'),('Hong' 'Kong')],separator='_')
```

```
[8] for sent in sent_tokenize(s):
      print(mwe.tokenize(word_tokenize(sent)))
```

```
    ['Good', 'cake', 'cost', 'Rs.1500\\kg', 'in', 'Mumbai.Please', 'buy', 'me', 'one', 'of', 'them', '.']
    ['Thanks', '.']
```