Terna Engineering College
Computer Engineering Department
Program: Sem VIII

Course: Natural Language Processing

Experiment No. 2

A.1 Aim: Implement stemming and lemmatization operations for a corpus.

PART B
(PART B: TO BE COMPLETED BY STUDENTS)

Roll No. 50	**Name:** AMEY THAKUR
Class: BE COMPS B	**Batch:** B3
Date of Experiment: 31/01/2022	**Date of Submission:** 31/01/2022
Grade:	

B.1 Software Code written by a student:

- **Install NLTK**

```
pip install nltk
```

```
import nltk
```

```
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('omw-1.4')
```

- **Stemming**

```
from nltk.stem import PorterStemmer
e_words= ["walk", "walks", "walked", "walking"]
ps =PorterStemmer()
for w in e_words:
  rootWord=ps.stem(w)
  print(rootWord)
```

- **Applying Stemma on a complete sentence**

```
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize
sentence="Let me sleep through the chaos, let me escape the reality one
more time, I’m tired"
words = word_tokenize(sentence)
ps = PorterStemmer()
for w in words:
  rootWord=ps.stem(w)
  print(rootWord)
```

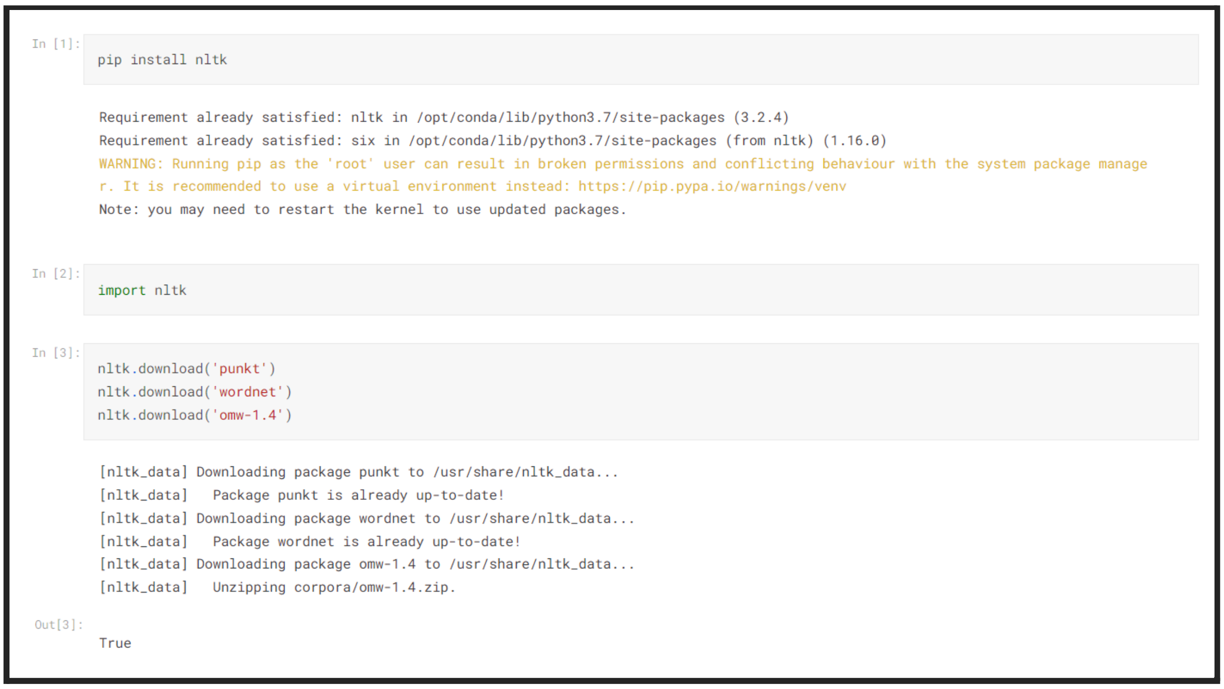
- **Applying Stemma on a complete sentence**

```
import nltk
from nltk.stem.porter import PorterStemmer
porter_stemmer = PorterStemmer()
text = "walking running plays studies sleeps worked"
tokenization = nltk.word_tokenize(text)
for w in tokenization:
  print("Stemming for {} is {}".format(w,porter_stemmer.stem(w)))
```

```
import nltk
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
text = "walking running plays studies sleeps worked"
tokenization = nltk.word_tokenize(text)
for w in tokenization:
             print("Lemma       for       {}       is       {}".format(w,

wordnet_lemmatizer.lemmatize(w)))
```

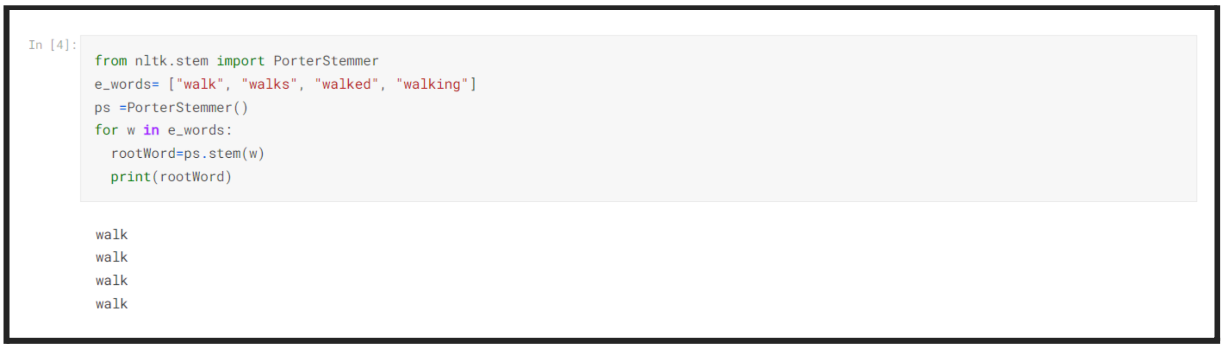
B.2 Input and Output:

- **Install NLTK**



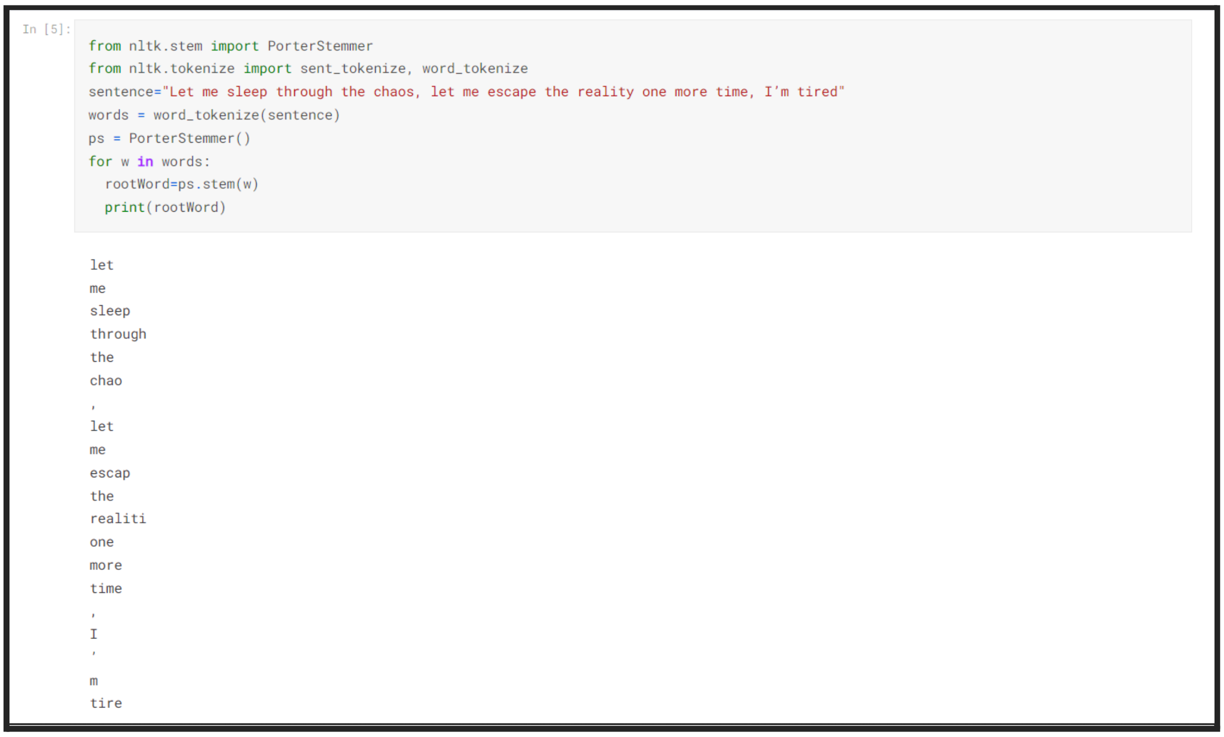
```
In [1]: pip install nltk

Requirement already satisfied: nltk in /opt/conda/lib/python3.7/site-packages (3.2.4)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages (from nltk) (1.16.0)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manage
r. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
Note: you may need to restart the kernel to use updated packages.

In [2]: import nltk

In [3]: nltk.download('punkt')
        nltk.download('wordnet')
        nltk.download('omw-1.4')

[nltk_data] Downloading package punkt to /usr/share/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /usr/share/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /usr/share/nltk_data...
[nltk_data]   Unzipping corpora/omw-1.4.zip.

Out[3]: True
```

- **Stemming**



```
In [4]: from nltk.stem import PorterStemmer
        e_words= ["walk", "walks", "walked", "walking"]
        ps =PorterStemmer()
        for w in e_words:
          rootWord=ps.stem(w)
          print(rootWord)

walk
walk
walk
walk
```

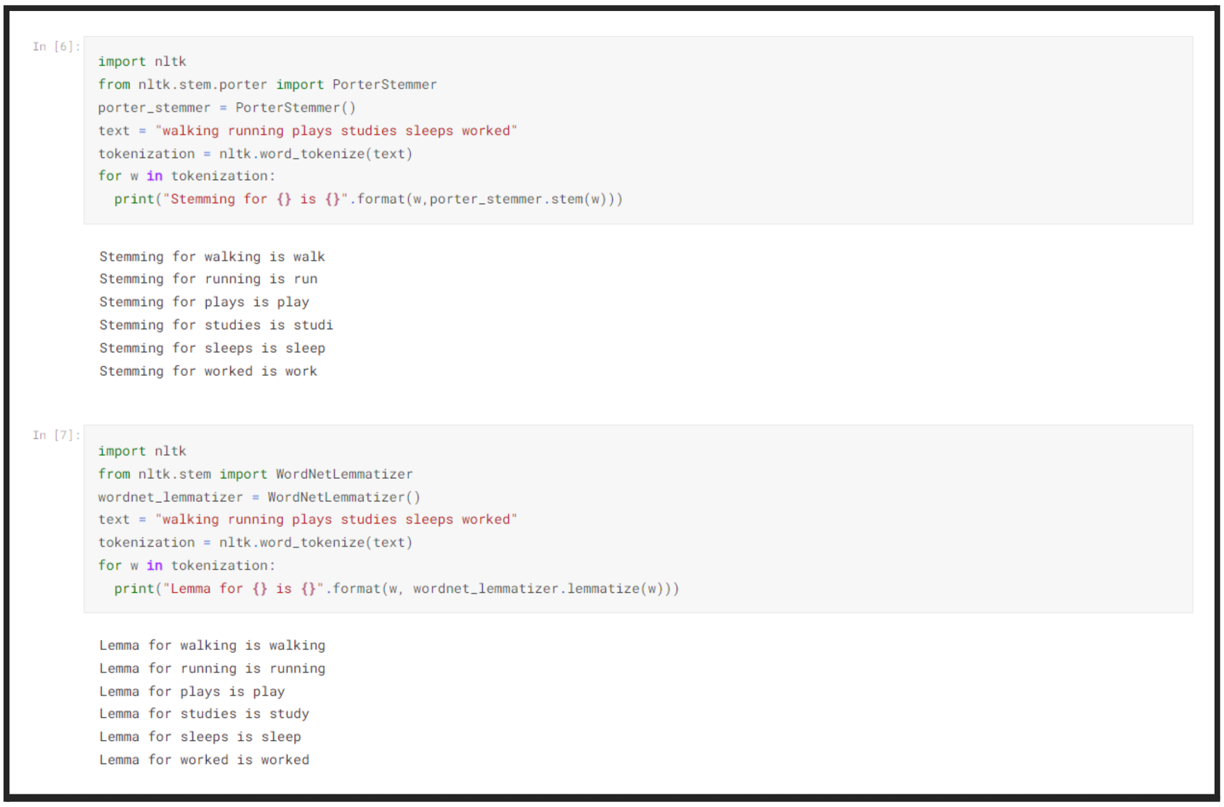
- **Applying Stemma on a complete sentence**



In [5]:
```
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize
sentence="Let me sleep through the chaos, let me escape the reality one more time, I'm tired"
words = word_tokenize(sentence)
ps = PorterStemmer()
for w in words:
  rootWord=ps.stem(w)
  print(rootWord)
```

```
let
me
sleep
through
the
chao
,
let
me
escap
the
realiti
one
more
time
,
I
'
m
tire
```

- **Applying Stemma on a complete sentence**



In [6]:
```
import nltk
from nltk.stem.porter import PorterStemmer
porter_stemmer = PorterStemmer()
text = "walking running plays studies sleeps worked"
tokenization = nltk.word_tokenize(text)
for w in tokenization:
  print("Stemming for {} is {}".format(w,porter_stemmer.stem(w)))
```

```
Stemming for walking is walk
Stemming for running is run
Stemming for plays is play
Stemming for studies is studi
Stemming for sleeps is sleep
Stemming for worked is work
```

In [7]:
```
import nltk
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
text = "walking running plays studies sleeps worked"
tokenization = nltk.word_tokenize(text)
for w in tokenization:
  print("Lemma for {} is {}".format(w, wordnet_lemmatizer.lemmatize(w)))
```

```
Lemma for walking is walking
Lemma for running is running
Lemma for plays is play
Lemma for studies is study
Lemma for sleeps is sleep
Lemma for worked is worked
```

B.3 Observations and learning:

- Text normalization approaches for Natural Language Processing are stemming and lemmatization in Python NLTK.
- In Natural Language Processing, stemming is a method of word standardisation. It is a method of converting a set of words in a sentence into a sequence to decrease the lookup time. The words that have the same meaning but differ depending on the context or sentence are normalised using this method.
- In NLTK, lemmatization is the algorithmic process of determining a word's lemma based on its meaning and context. In most cases, lemmatization refers to the morphological analysis of words to remove inflectional endings. It aids in the retrieval of the lemma, or basic or dictionary form, of a word.

B.4 Conclusion:

As a result, we've been able to effectively create a variety of functions to grasp the ideas of stemming and lemmatization.

B.5 Question of Curiosity

Q1. Explain porters' stemming algorithm with examples.
ANS:
Porter's Stemmer algorithm

- It is one of the most popular stemming methods proposed in 1980. It is based on the idea that the suffixes in the English language are made up of a combination of smaller and simpler suffixes.
- This stemmer is known for its speed and simplicity. The main applications of
- Porter Stemmer include data mining and Information retrieval. However, its applications are only limited to English words.
- Also, the group of stems is mapped onto the same stem and the output stem is not necessarily a meaningful word.
- The algorithms are fairly lengthy and are known to be the oldest stemmer.
- Example: EED -> EE means "if the word has at least one vowel and consonant plus EED ending, change the ending to EE" as 'agreed' becomes 'agree'.

Q2. What are stop words? Explain it with an example in NLTK.
ANS:

- Python for Natural Language Processing (NLP) is a field of study that faces several obstacles, including natural language comprehension.
- Stopwords are a collection of highly common words that don't give meaningful information for most text analysis algorithms.
- Stop words like 'the,' 'is,' and 'are' may appear in the text. Stop words in the text to be processed can be filtered out. In NLP research, there is no uniform list of stop words; nevertheless, the nltk module provides a list of stop words.

- While it aids in the comprehension of phrase structure, it does not aid in the comprehension of sentence semantics. Here's a list of the most widely used English words: N = ['stop', 'the', 'to', 'and', 'a', 'in', 'it', 'is', 'I', 'that', 'had', 'on', 'for', 'were', 'was']
- You don't have to declare each stop word manually using nltk. Stop words are words that are regularly used but have little significance. Stop words are terms that are so prevalent that tokenizers usually ignore them.
- NLTK (Natural Language Toolkit) comes with a list of 40 stop words by default, including "a," "an," "the," "of," "in," and so on.

Q3.How is the dataset accessed by NLTK? Explain wordnet lemmatizer in NLTK.
ANS:

- The nltk.data module includes routines for loading NLTK resource files such corpora, grammars, and stored processing objects.
- Loading data files
  The function nltk.data.load(), which takes as its first parameter a URL specifying which file should be loaded, is used to load resources. The NLTK data distribution is loaded via the nltk: protocol:
  ```
  >>> tokenizer =
  nltk.data.load('nltk:tokenizers/punkt/english.pickle')
  >>> tokenizer.tokenize('Hello. This is a test. It works!')
  ['Hello.', 'This is a test.', 'It works!']
  ```

Wordnet lemmatizer

- Lemmatizer reduces ambiguity in the text.
- Examples include bicycles and bicycles, which are both transformed to the basic word bicycle.
- In essence, it will return all words with the same meaning but different representations to their original form.
- It helps to provide correct features for the training machine by reducing the word density in the given text. The more precise and clever your machine learning model is, the cleaner the data is.
- The NLTK Lemmatizer saves both memory and computational time.