Ex.no:1 Functions in NLTK Packages
Date:16-12-2024

AIM:
To implement basic functions in NLTK Packages.

PROCEDURE:
Import NLTK.
Download the packages from NLTK .
View the Packages inside NLTK.
Implement the basic functions .

CODE:
Install and import NLTK:

```
In [1]: !pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\praveena\anaconda3\lib\site-packages (3.8.1)
Requirement already satisfied: click in c:\users\praveena\anaconda3\lib\site-packages (from nltk) (8.0.4)
Requirement already satisfied: joblib in c:\users\praveena\anaconda3\lib\site-packages (from nltk) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\praveena\anaconda3\lib\site-packages (from nltk) (2022.7.9)
Requirement already satisfied: tqdm in c:\users\praveena\anaconda3\lib\site-packages (from nltk) (4.65.0)
Requirement already satisfied: colorama in c:\users\praveena\anaconda3\lib\site-packages (from click->nltk) (0.4.6)
```

```
In [2]: import nltk
```

Download the Packages:

```
In [16]: nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Praveena\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

```
Out[16]: True
```

View the Packages using :

```
In [4]: dir(nltk)
```

```
Out[4]: ['ARLSTem',
 'ARLSTem2',
 'AbstractLazySequence',
 'AffixTagger',
 'AlignedSent',
 'Alignment',
 'AnnotationTask',
 'ApplicationExpression',
 'Assignment',
 'BigramAssocMeasures',
 'BigramCollocationFinder',
 'BigramTagger',
 'BinaryMaxentFeatureEncoding',
 'BlanklineTokenizer',
 'BllipParser',
 'BottomUpChartParser',
 'BottomUpLeftCornerChartParser',
 'BottomUpProbabilisticChartParser',
 'Boxer',
```

Implement basic functions :

1) TOKENIZATION:

```
In [7]: from nltk.tokenize import word_tokenize
```

```
In [10]: text="what is your name"
         tokens=word_tokenize(text)
         tokens
```

```
Out[10]: ['what', 'is', 'your', 'name']
```

2) POS-TAGGING:

In [18]:
```
from nltk import pos_tag
pos=pos_tag(tokens)
pos
```

Out[18]:
```
[('what', 'WP'), ('is', 'VBZ'), ('your', 'PRP$'), ('name', 'NN')]
```

3) LEMMATIZATION:

In [28]:
```
#nltk.download('wordnet')
from nltk.stem import WordNetLemmatizer

# create an object of class WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
print(lemmatizer.lemmatize("better", 'a'))
```

```
good
```

4) STEMMING:

In [30]:
```
from nltk.stem import PorterStemmer
stemmer = PorterStemmer()
words = ["running", "flies", "easily", "connected"]
stemmed_words = [stemmer.stem(word) for word in words]
print(stemmed_words)
```

```
['run', 'fli', 'easili', 'connect']
```