NATURAL LANGUAGE PROCESSING

LAB ASSIGNMENT - 2

Course Code: CSE3015

Course Title: Natural Language Processing

Professor: Prof.V SRIKANTH REDDY

Slot: L47+L48

Name: P. Sathwika

Reg. No.: 21BCE8118

```
#1.Write a program to slit sentences in a document
import nltk
nltk.download('punkt')
def split_sentences(document):
    sentences = nltk.sent_tokenize(document)
    return sentences

para = "It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness"
sentences = split_sentences(para)
print(sentences)
```

```
['It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness']
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
#2. Perform tokenizing and stemming by reading the input string?

from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
nltk.download('punkt')
def tokenize_and_stem(sentence):
    words = word_tokenize(sentence)
    stemmer = PorterStemmer()
    stemmed_words = [stemmer.stem(word) for word in words]
    return stemmed_words

line = "VIT IS TOP EMERGING COLLEGE IN INDIA."
tokenized_and_stemmed = tokenize_and_stem(line)
print(tokenized_and_stemmed)
```

```
['vit', 'is', 'top', 'emerg', 'colleg', 'in', 'india', '.']
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
#3. Remove the stopwords and rare words in the document?

from nltk.corpus import stopwords
from collections import Counter
nltk.download('stopwords')
def remove_stopwords_and_rare_words(sentence):
    stop_words = set(stopwords.words('english'))
    words = word_tokenize(sentence)
    filtered_words = [word for word in words if word.lower() not in stop_words]
    word_counts = Counter(filtered_words)
    filtered_words = [word for word in filtered_words if word_counts[word] > 1]
    return filtered_words

input_sentence = "It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness"
filtered_words = remove_stopwords_and_rare_words(input_sentence)
print(filtered_words)
```

```
     ['times', ',', 'times', ',', 'age', ',', 'age']
     [nltk_data] Downloading package stopwords to /root/nltk_data...
     [nltk_data]   Package stopwords is already up-to-date!
```

```
#4. Identify the parts of speech in the document?

from nltk import pos_tag
from nltk.tokenize import word_tokenize
nltk.download('averaged_perceptron_tagger')
def identify_parts_of_speech(sentence):
    words = word_tokenize(sentence)
    pos_tags = pos_tag(words)
    return pos_tags

input_sentence = "raj has just gone out to the market. "
pos_tags = identify_parts_of_speech(input_sentence)
print(pos_tags)
```

```
     [('raj', 'NN'), ('has', 'VBZ'), ('just', 'RB'), ('gone', 'VBN'), ('out', 'RP'), ('to', 'TO'), ('the', 'DT'), ('market', 'NN'), ('.', '.')]
     [nltk_data] Downloading package averaged_perceptron_tagger to
     [nltk_data]     /root/nltk_data...
     [nltk_data]   Package averaged_perceptron_tagger is already up-to-
     [nltk_data]       date!
```

```
#5. Implement the N-gram tagger?

from nltk.util import ngrams
def ngram_tagger(sentence, n=2):
    words = word_tokenize(sentence)
    n_grams = list(ngrams(words, n))
    return n_grams
input_sentence = "I have been in Pune for one week."
n_grams = ngram_tagger(input_sentence, n=2)
print(n_grams)
```

```
     [('I', 'have'), ('have', 'been'), ('been', 'in'), ('in', 'Pune'), ('Pune', 'for'), ('for', 'one'), ('one', 'week'), ('week', '.')]
```