	Bansilal Ramnath Agarwal Charitable Trust's Vishwakarma Institute of Information Technology **Department of Artificial Intelligence and Data Science**

Name: **Siddhesh Dilip Khairnar**		
Class: **TY**	Division: **B**	Roll No: **372028**
Semester: **6th**	Academic Year: **2023-24**	
Subject Name & Code: **Natural Language and Processing & ADUA32203**		
Title of Assignment: **Perform various pre-processing tasks like tokenization, stemming, lemmatization, stop word removal etc. using inbuilt functions. and using regular expressions**		
Date of Performance: **27-01-2024**	Date of Submission: **08-02-2024**	

ASSIGNMENT NO: - 2

Aim: Perform various pre-processing tasks like tokenization, stemming, lemmatization, stop word removal etc. using inbuilt functions and using regular expressions

❖ **THEORY:**

- **Text Preprocessing:** Text preprocessing is a crucial step in Natural Language Processing (NLP) that involves cleaning and transforming raw text data into a format suitable for analysis. Various tasks are performed to enhance the quality of text data and improve the performance of NLP models.
- **Tokenization:**

- **Definition:** Tokenization is the process of breaking down a text into individual units, typically words or phrases (tokens).
- **Inbuilt Functions:** Most NLP libraries provide built-in functions for tokenization, allowing easy extraction of tokens from a given text.
- **Regular Expressions:** Regular expressions can be employed to define custom tokenization rules, providing flexibility in handling specific patterns.

- **Stemming and Lemmatization:**

- **Stemming:** Reducing words to their root or base form by removing suffixes.
- **Lemmatization:** Like stemming but aims to reduce words to their canonical form (lemma), considering the context.
- **Inbuilt Functions:** NLP libraries often offer functions for both stemming and lemmatization, helping standardize and normalize words.
- **Regular Expressions:** Custom regular expressions can be designed to achieve stemming and lemmatization based on specific patterns.

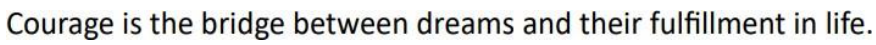
- **Stop Word Removal:**

- **Definition:** Stop words are common words (e.g., "the," "is," "and") that are often removed during text preprocessing to focus on content carrying words.
- **Inbuilt Functions:** Libraries provide predefined lists of stop words and functions for their removal.

```python
import nltk
from nltk.tokenize import line_tokenize, word_tokenize, TweetTokenizer
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk.corpus import stopwords
from PyPDF2 import PdfReader

# Function to read PDF file and extract text
def read_pdf(pdf_path):
    with open(pdf_path, 'rb') as file:
        pdf_reader = PdfReader(file)
        pdf_text = ''
        for page_num in range(len(pdf_reader.pages)):
            page = pdf_reader.pages[page_num]
            pdf_text += page.extract_text()
    return pdf_text

# Example PDF file path
pdf_path = r'D:\VS Code\Sem 6 Assignments\NLP\nlp2files\corpus.pdf'

# Read content from PDF
pdf_text = read_pdf(pdf_path)

# Perform line, space, word, and tweet tokenization
lines_tokens = line_tokenize(pdf_text)
space_tokens = pdf_text.split()
word_tokens = word_tokenize(pdf_text)
tweet_tokenizer = TweetTokenizer()
tweet_tokens = tweet_tokenizer.tokenize(pdf_text)


# Display tokenized outputs
print("Line Tokenization:", lines_tokens)
print("\nSpace Tokenization:", space_tokens)
print("\nWord Tokenization:", word_tokens)
print("\nTweet Tokenization:", tweet_tokens)

# Perform stemming using Porter Stemmer
porter_stemmer = PorterStemmer()
stemmed_tokens = [porter_stemmer.stem(token) for token in word_tokens]

# Display stemmed tokens
print("\nStemmed Tokens:", stemmed_tokens)

# Perform lemmatization using WordNet
wordnet_lemmatizer = WordNetLemmatizer()
lemmatized_tokens = [wordnet_lemmatizer.lemmatize(token) for token in word_tokens]

# Display lemmatized tokens
print("\nLemmatized Tokens:", lemmatized_tokens)

# Remove stop words using NLTK's stop-word corpus
stop_words = set(stopwords.words('english'))
filtered_tokens = [token for token in word_tokens if token.lower() not in stop_words]

# Display filtered tokens after stop-word removal
print("\nTokens after Stop-word Removal:", filtered_tokens)

```

INPUT:

PDF



OUTPUT:

```
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

D:\VS Code\Sem 6 Assignments\NLP>python -u "d:\VS Code\Sem 6 Assignments\NLP\nlp2files\NLP2pdf.py"
Line Tokenization: ['Courage is the bridge between dreams and their fulfillment in life.  ']

Space Tokenization: ['Courage', 'is', 'the', 'bridge', 'between', 'dreams', 'and', 'their', 'fulfillment', 'in', 'life.']

Word Tokenization: ['Courage', 'is', 'the', 'bridge', 'between', 'dreams', 'and', 'their', 'fulfillment', 'in', 'life', '.']

Tweet Tokenization: ['Courage', 'is', 'the', 'bridge', 'between', 'dreams', 'and', 'their', 'fulfillment', 'in', 'life', '.']

Stemmed Tokens: ['courag', 'is', 'the', 'bridg', 'between', 'dream', 'and', 'their', 'fulfil', 'in', 'life', '.']

Lemmatized Tokens: ['Courage', 'is', 'the', 'bridge', 'between', 'dream', 'and', 'their', 'fulfillment', 'in', 'life', '.']

Tokens after Stop-word Removal: ['Courage', 'bridge', 'dreams', 'fulfillment', 'life', '.']
```

```python
import nltk
from nltk.tokenize import line_tokenize, word_tokenize, TweetTokenizer
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk.corpus import stopwords

# Updated file path
corpus_path = r'D:\VS Code\Sem 6 Assignments\NLP\nlp2files\cricket_corpus.txt'

with open(corpus_path, 'r', encoding='utf-8') as file:
    user_corpus = file.read()

# Perform line, space, word, and tweet tokenization
lines_tokens = line_tokenize(user_corpus)
space_tokens = user_corpus.split()
word_tokens = word_tokenize(user_corpus)
tweet_tokenizer = TweetTokenizer()
tweet_tokens = tweet_tokenizer.tokenize(user_corpus)

# Display tokenized outputs
print("Line Tokenization:", lines_tokens)
print("\nSpace Tokenization:", space_tokens)
print("\nWord Tokenization:", word_tokens)
print("\nTweet Tokenization:", tweet_tokens)

# Perform stemming using Porter Stemmer
porter_stemmer = PorterStemmer()
stemmed_tokens = [porter_stemmer.stem(token) for token in word_tokens]

# Display stemmed tokens
print("\nStemmed Tokens:", stemmed_tokens)

# Perform lemmatization using WordNet
wordnet_lemmatizer = WordNetLemmatizer()
lemmatized_tokens = [wordnet_lemmatizer.lemmatize(token) for token in word_tokens]

# Display lemmatized tokens
print("\nLemmatized Tokens:", lemmatized_tokens)

# Remove stop words using NLTK's stop-word corpus
stop_words = set(stopwords.words('english'))
filtered_tokens = [token for token in word_tokens if token.lower() not in stop_words]

# Display filtered tokens after stop-word removal
print("\nTokens after Stop-word Removal:", filtered_tokens)
```

INPUT:

Text:

```
Cricket is a popular bat-and-ball sport played between two teams,
each consisting of 11 players, where the objective is to score
runs by hitting the ball and running between wickets. The game is
divided into innings, and teams take turns batting and bowling,
with the team scoring the most runs declared the winner.
```

Output:

```
D:\VS Code\Sem 6 Assignments\NLP>python -u "d:\VS Code\Sem 6 Assignments\NLP\nlp2files\NLP2text.py"
Line Tokenization: ['Cricket is a popular bat-and-ball sport played between two teams, each consisting of 1
1 players, where the objective is to score runs by hitting the ball and running between wickets. The game i
s divided into innings, and teams take turns batting and bowling, with the team scoring the most runs decla
red the winner.']

Space Tokenization: ['Cricket', 'is', 'a', 'popular', 'bat-and-ball', 'sport', 'played', 'between', 'two',
'teams,', 'each', 'consisting', 'of', '11', 'players,', 'where', 'the', 'objective', 'is', 'to', 'score', '
runs', 'by', 'hitting', 'the', 'ball', 'and', 'running', 'between', 'wickets.', 'The', 'game', 'is', 'divid
ed', 'into', 'innings,', 'and', 'teams', 'take', 'turns', 'batting', 'and', 'bowling,', 'with', 'the', 'tea
m', 'scoring', 'the', 'most', 'runs', 'declared', 'the', 'winner.']

Word Tokenization: ['Cricket', 'is', 'a', 'popular', 'bat-and-ball', 'sport', 'played', 'between', 'two', '
teams', ',', 'each', 'consisting', 'of', '11', 'players', ',', 'where', 'the', 'objective', 'is', 'to', 'sc
ore', 'runs', 'by', 'hitting', 'the', 'ball', 'and', 'running', 'between', 'wickets', '.', 'The', 'game', '
is', 'divided', 'into', 'innings', ',', 'and', 'teams', 'take', 'turns', 'batting', 'and', 'bowling', ',',
'with', 'the', 'team', 'scoring', 'the', 'most', 'runs', 'declared', 'the', 'winner', '.']

Tweet Tokenization: ['Cricket', 'is', 'a', 'popular', 'bat-and-ball', 'sport', 'played', 'between', 'two',
'teams', ',', 'each', 'consisting', 'of', '11', 'players', ',', 'where', 'the', 'objective', 'is', 'to', 's
core', 'runs', 'by', 'hitting', 'the', 'ball', 'and', 'running', 'between', 'wickets', '.', 'The', 'game',
'is', 'divided', 'into', 'innings', ',', 'and', 'teams', 'take', 'turns', 'batting', 'and', 'bowling', ',',
 'with', 'the', 'team', 'scoring', 'the', 'most', 'runs', 'declared', 'the', 'winner', '.']

Stemmed Tokens: ['cricket', 'is', 'a', 'popular', 'bat-and-bal', 'sport', 'play', 'between', 'two', 'team',
 ',', 'each', 'consist', 'of', '11', 'player', ',', 'where', 'the', 'object', 'is', 'to', 'score', 'run', '
by', 'hit', 'the', 'ball', 'and', 'run', 'between', 'wicket', '.', 'the', 'game', 'is', 'divid', 'into', 'i
nning', ',', 'and', 'team', 'take', 'turn', 'bat', 'and', 'bowl', ',', 'with', 'the', 'team', 'score', 'the
', 'most', 'run', 'declar', 'the', 'winner', '.']


Lemmatized Tokens: ['Cricket', 'is', 'a', 'popular', 'bat-and-ball', 'sport', 'played', 'between', 'two', '
e', 'is', 'divided', 'into', 'inning', ',', 'and', 'team', 'take', 'turn', 'batting', 'and', 'bowling', ','
, 'with', 'the', 'team', 'scoring', 'the', 'most', 'run', 'declared', 'the', 'winner', '.']

Tokens after Stop-word Removal: ['Cricket', 'popular', 'bat-and-ball', 'sport', 'played', 'two', 'teams', '
,', 'consisting', '11', 'players', ',', 'objective', 'score', 'runs', 'hitting', 'ball', 'running', 'wicket
s', '.', 'game', 'divided', 'innings', ',', 'teams', 'take', 'turns', 'batting', 'bowling', ',', 'team', 's
coring', 'runs', 'declared', 'winner', '.']
```

```python
import nltk
from nltk.tokenize import line_tokenize, word_tokenize, TweetTokenizer
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk.corpus import stopwords
from bs4 import BeautifulSoup

# Function to read local HTML file and extract text
def read_local_html(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        html_content = file.read()
        soup = BeautifulSoup(html_content, 'html.parser')
        # Extract text from HTML content
        website_text = ' '.join([p.get_text() for p in soup.find_all('p')])
    return website_text

# Example HTML file path
html_file_path = r'D:\VS Code\Sem 6 Assignments\NLP\nlp2files\corpus.html'

# Read content from the HTML file
website_text = read_local_html(html_file_path)

# Perform line, space, word, and tweet tokenization
lines_tokens = line_tokenize(website_text)
space_tokens = website_text.split()
word_tokens = word_tokenize(website_text)
tweet_tokenizer = TweetTokenizer()
tweet_tokens = tweet_tokenizer.tokenize(website_text)

# Display tokenized outputs
print("Line Tokenization:", lines_tokens)
print("\nSpace Tokenization:", space_tokens)
print("\nWord Tokenization:", word_tokens)
print("\nTweet Tokenization:", tweet_tokens)

# Perform stemming using Porter Stemmer
porter_stemmer = PorterStemmer()
stemmed_tokens = [porter_stemmer.stem(token) for token in word_tokens]

# Display stemmed tokens
print("\nStemmed Tokens:", stemmed_tokens)

# Perform lemmatization using WordNet
wordnet_lemmatizer = WordNetLemmatizer()
lemmatized_tokens = [wordnet_lemmatizer.lemmatize(token) for token in word_tokens]

# Display lemmatized tokens
print("\nLemmatized Tokens:", lemmatized_tokens)

# Remove stop words using NLTK's stop-word corpus
stop_words = set(stopwords.words('english'))
filtered_tokens = [token for token in word_tokens if token.lower() not in stop_words]

# Display filtered tokens after stop-word removal
print("\nTokens after Stop-word Removal:", filtered_tokens)
```

INPUT:

Website:

Opportunities are often disguised as hard work, so people miss them.

Website link:
file:///D:/VS%20Code/Sem%206%20Assignments/NLP/nlp2files/corpus.html

OUTPUT:

```
D:\VS Code\Sem 6 Assignments\NLP>python -u "d:\VS Code\Sem 6 Assignments\NLP\nlp2fil
es\NLP2website.py"
Line Tokenization: ['Opportunities are often disguised as hard work, so people miss
them.']

Space Tokenization: ['Opportunities', 'are', 'often', 'disguised', 'as', 'hard', 'wo
rk,', 'so', 'people', 'miss', 'them.']

Word Tokenization: ['Opportunities', 'are', 'often', 'disguised', 'as', 'hard', 'wor
k', ',', 'so', 'people', 'miss', 'them', '.']

Tweet Tokenization: ['Opportunities', 'are', 'often', 'disguised', 'as', 'hard', 'wo
rk', ',', 'so', 'people', 'miss', 'them', '.']

Stemmed Tokens: ['opportun', 'are', 'often', 'disguis', 'as', 'hard', 'work', ',', '
so', 'peopl', 'miss', 'them', '.']

Lemmatized Tokens: ['Opportunities', 'are', 'often', 'disguised', 'a', 'hard', 'work
', ',', 'so', 'people', 'miss', 'them', '.']

Tokens after Stop-word Removal: ['Opportunities', 'often', 'disguised', 'hard', 'wor
k', ',', 'people', 'miss', '.']
```

```python
import nltk
from nltk.tokenize import line_tokenize, word_tokenize, TweetTokenizer
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk.corpus import stopwords
from docx import Document

# Function to read DOCX file and extract text
def read_docx(docx_path):
    doc = Document(docx_path)
    doc_text = ''
    for paragraph in doc.paragraphs:
        doc_text += paragraph.text + ' '
    return doc_text

# Example DOCX file path
docx_path = r'D:\VS Code\Sem 6 Assignments\NLP\nlp2files\corpus.docx'

# Read content from DOCX
docx_text = read_docx(docx_path)

# Perform line, space, word, and tweet tokenization
lines_tokens = line_tokenize(docx_text)
space_tokens = docx_text.split()
word_tokens = word_tokenize(docx_text)
tweet_tokenizer = TweetTokenizer()
tweet_tokens = tweet_tokenizer.tokenize(docx_text)

# Display tokenized outputs
print("Line Tokenization:", lines_tokens)
print("\nSpace Tokenization:", space_tokens)
print("\nWord Tokenization:", word_tokens)
print("\nTweet Tokenization:", tweet_tokens)

# Perform stemming using Porter Stemmer
porter_stemmer = PorterStemmer()
stemmed_tokens = [porter_stemmer.stem(token) for token in word_tokens]

# Display stemmed tokens
print("\nStemmed Tokens:", stemmed_tokens)

# Perform lemmatization using WordNet
wordnet_lemmatizer = WordNetLemmatizer()
lemmatized_tokens = [wordnet_lemmatizer.lemmatize(token) for token in word_tokens]

# Display lemmatized tokens
print("\nLemmatized Tokens:", lemmatized_tokens)

# Remove stop words using NLTK's stop-word corpus
stop_words = set(stopwords.words('english'))
filtered_tokens = [token for token in word_tokens if token.lower() not in stop_words]

# Display filtered tokens after stop-word removal
print("\nTokens after Stop-word Removal:", filtered_tokens)
```

INPUT:

Docx file:

Courage is the bridge between dreams and their realization in life.

OUTPUT:

```
D:\VS Code\Sem 6 Assignments\NLP>python -u "d:\VS Code\Sem 6 Assignments\NLP\nlp2files\NLP2word.py"
Line Tokenization: ['Courage is the bridge between dreams and their realization in life. ']

Space Tokenization: ['Courage', 'is', 'the', 'bridge', 'between', 'dreams', 'and', 'their', 'realization', 'in', 'life.']

Word Tokenization: ['Courage', 'is', 'the', 'bridge', 'between', 'dreams', 'and', 'their', 'realization', 'in', 'life', '.']

Tweet Tokenization: ['Courage', 'is', 'the', 'bridge', 'between', 'dreams', 'and', 'their', 'realization', 'in', 'life', '.']

Stemmed Tokens: ['courag', 'is', 'the', 'bridg', 'between', 'dream', 'and', 'their', 'realiz', 'in', 'life', '.']

Lemmatized Tokens: ['Courage', 'is', 'the', 'bridge', 'between', 'dream', 'and', 'their', 'realization', 'in', 'life', '.']

Tokens after Stop-word Removal: ['Courage', 'bridge', 'dreams', 'realization', 'life', '.']
```