Experiment No. 2

Title: Implementation of removal of punctuations, stop words, extra white spaces, URLs and HTML code from Text

Batch: B2 **Roll No.:16010420061** **Experiment No.: 2**

Aim: To implement removal of punctuations, stop words, extra white spaces, URLs and HTML code from Text.

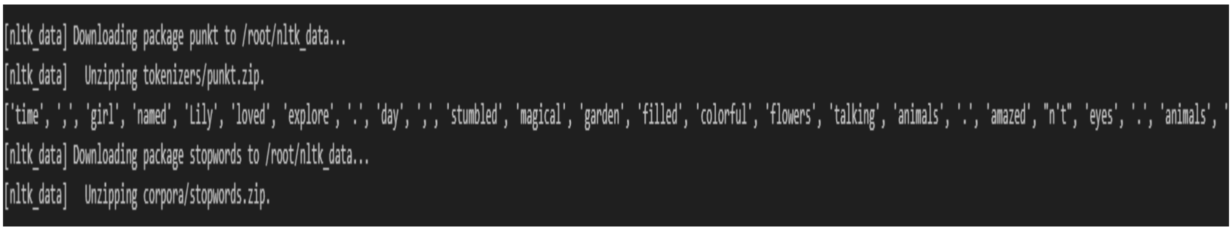
Resources needed: Text Editor, Python Interpreter

Activity:

1. Add custom list of stop words to English language stop words and use this list of stop words to remove stop words from text

```
import nltk
from nltk.corpus import stopwords
nltk.download('punkt')
nltk.download('stopwords')

# Add custom stop words to the existing list
custom_stop_words = ["0o", "0s", "3a", "3b", "3d", "6b", "6o", "a", "a1",
"a2", "a3", "a4", "ab", "able", "about", "above", "abst", "ac", "accordance",
"according", "accordingly", "across", "act", "actually", "ad", "added", "adj",
"ae", "af", "affected", "affecting", "affects", "after", "afterwards", "ag",
"again", "against", "ah", "ain", "ain't", "aj", "al", "all", "allow",
"allows", "almost", "alone", "along", "already", "also", "although", "always",
"am", "among",…..,"wherever", "whether", "which", "while", "whim", "whither",
"who", "whod", "whoever", "whole", "who'll", "whom", "whomever", "whos",
"who's", "whose", "why", "why's", "wi", "widely", "will", "willing", "wish",
"with", "within", "without", "wo", "won", "wonder", "wont", "won't", "words",
"world", "would", "wouldn", "wouldnt", "wouldn't", "www", "x", "x1", "x2",
"x3", "xf", "xi", "xj", "xk", "xl", "xn", "xo", "xs", "xt", "xv", "xx", "y",
"y2", "yes", "yet", "yj", "yl", "you", "youd", "you'd", "you'll", "your",
"youre", "you're", "yours", "yourself", "yourselves", "you've", "yr", "ys",
"yt", "z", "zero", "zi", "zz"]

# Get the standard English stop words
english_stop_words = set(stopwords.words("english"))

# Combine the standard and custom stop words
all_stop_words = english_stop_words.union(custom_stop_words)

# Your text
text = "Once upon a time, there was a little girl named Lily who loved to
explore the world around her. One day, she stumbled upon a magical garden
filled with colorful flowers and talking animals. She was amazed and couldn't
believe her eyes. The animals welcomed her with open arms and showed her
around the garden. They even taught her how to talk to the flowers and make
them grow. Lily was overjoyed and spent hours playing and learning in the
```

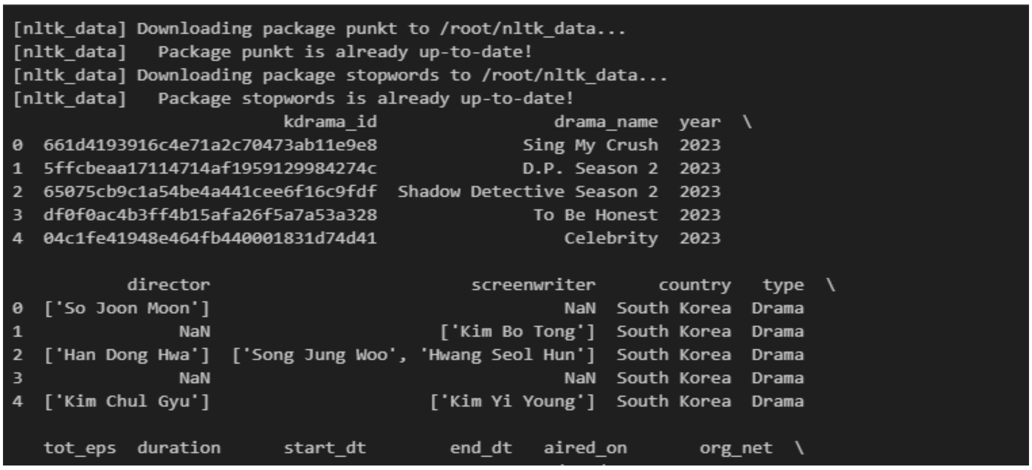
```
garden. From that day on, she visited the garden every day and made many new
friends. It was a magical place that she would never forget."

# Tokenize the text
words = nltk.word_tokenize(text)

# Remove stop words
final_list = [word for word in words if word.lower() not in all_stop_words]

print(final_list)
```



```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
['time', ',', 'girl', 'named', 'Lily', 'loved', 'explore', '.', 'day', ',', 'stumbled', 'magical', 'garden', 'filled', 'colorful', 'flowers', 'talking', 'animals', '.', 'amazed', "n't", 'eyes', '.', 'animals', '
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

2. Apply stop word removal, punctuation removal, space removal, URL and HTML code removal to a dataset of technical discussion forum such as dataset of stack overflow.

```
import re
import pandas as pd
from bs4 import BeautifulSoup
import nltk
from nltk.corpus import stopwords

# Download NLTK resources
nltk.download('punkt')
nltk.download('stopwords')


df= pd.read_csv('/content/korean_drama.csv')
print(df.head())

def preprocess_text(text):
    if isinstance(text, str):
        # Remove HTML tags
        text = BeautifulSoup(text, "html.parser").get_text()

        # Remove URLs
        text = re.sub(r"http\S+|www\S+", "", text)

        # Remove punctuation
        text = re.sub(r'[^\w\s]', '', text)

        # Tokenize the text
        words = nltk.word_tokenize(text)

        # Remove stop words
        english_stop_words = set(stopwords.words("english"))
        filtered_words = [word for word in words if word.lower() not in 
english_stop_words]

        # Remove extra spaces and convert to lowercase
        cleaned_text = ' '.join(filtered_words).lower()

        return cleaned_text
    else:
        return ""

# Iterate over each row and apply preprocessing to the "synopsis" column
for index, row in df.iterrows():
      cleaned_synopsis = preprocess_text(row["synopsis"])
      df.at[index, "cleaned_synopsis"] = cleaned_synopsis

# Save the DataFrame with the cleaned data
df.to_csv("cleaned_korean_dramas.csv", index=False)
```



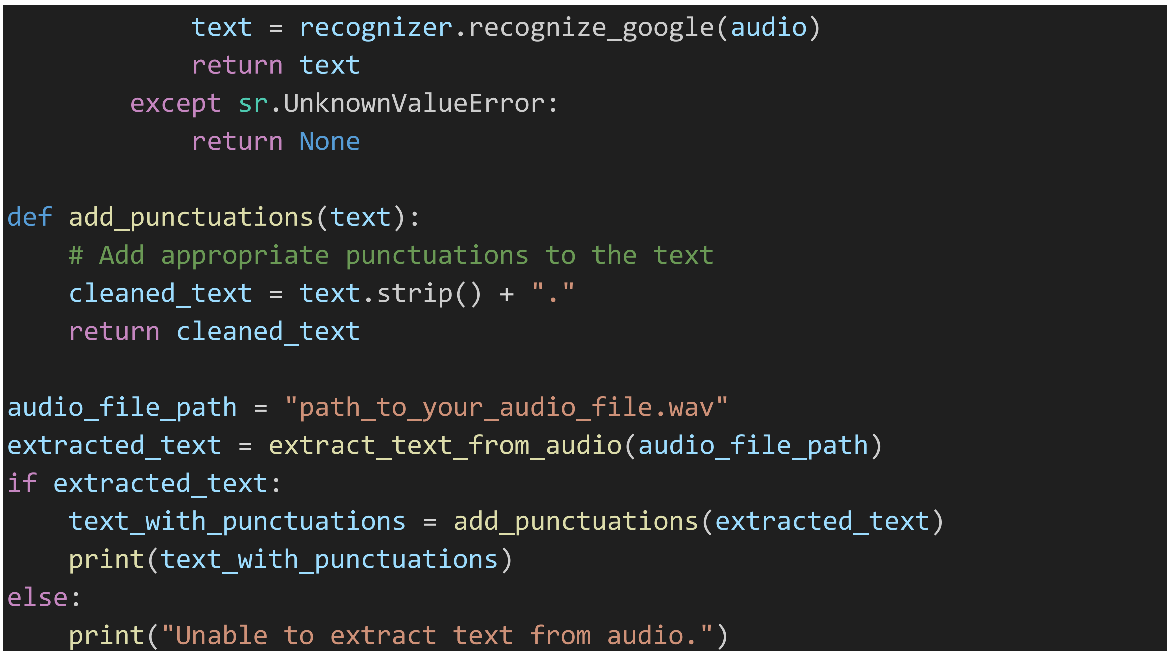
```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
                          kdrama_id                 drama_name  year  \
0  661d4193916c4e71a2c70473ab11e9e8              Sing My Crush  2023   
1  5ffcbeaa17114714af1959129984274c              D.P. Season 2  2023   
2  65075cb9c1a54be4a441cee6f16c9fdf  Shadow Detective Season 2  2023   
3  df0f0ac4b3ff4b15afa26f5a7a53a328               To Be Honest  2023   
4  04c1fe41948e464fb440001831d74d41                  Celebrity  2023   

           director                          screenwriter      country   type  \
0  ['So Joon Moon']                                   NaN  South Korea  Drama   
1               NaN                     ['Kim Bo Tong']  South Korea  Drama   
2  ['Han Dong Hwa']  ['Song Jung Woo', 'Hwang Seol Hun']  South Korea  Drama   
3               NaN                                   NaN  South Korea  Drama   
4  ['Kim Chul Gyu']                    ['Kim Yi Young']  South Korea  Drama   

   tot_eps  duration      start_dt        end_dt   aired_on       org_net  \
```

Questions:

1. As discussed there might be need to add punctuations or retain URLs in the given Text.
    a. Write the sample python code for extracting text from audio and add appropriate punctuations to it

    For extracting text from audio, you can use a speech-to-text library like SpeechRecognition. Here's how you can use it along with the string library to add appropriate punctuations to the extracted text:

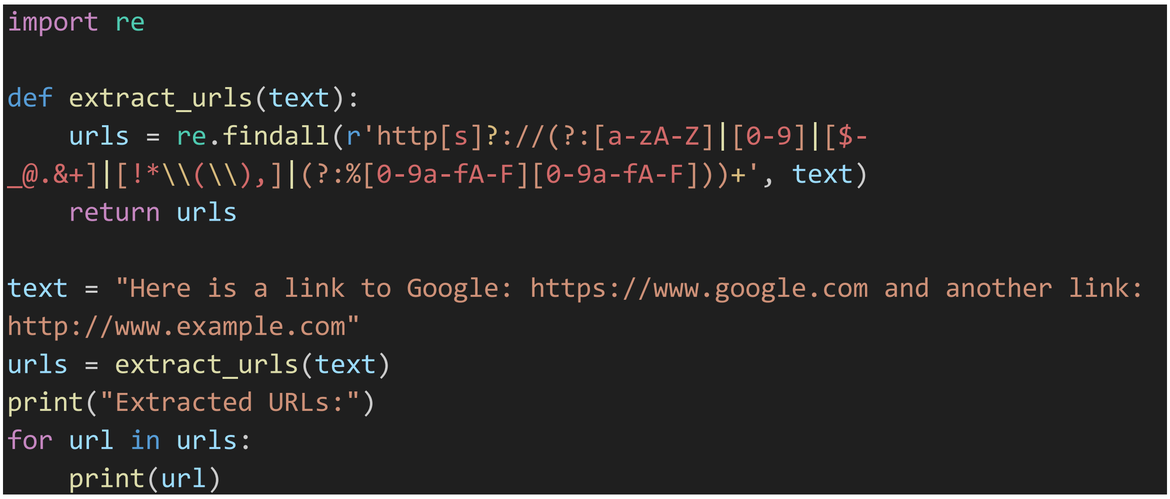
```
import speech_recognition as sr
import string

def extract_text_from_audio(audio_file_path):
    recognizer = sr.Recognizer()
    with sr.AudioFile(audio_file_path) as source:
        audio = recognizer.record(source)
        try:
```



```python
            text = recognizer.recognize_google(audio)
            return text
        except sr.UnknownValueError:
            return None

def add_punctuations(text):
    # Add appropriate punctuations to the text
    cleaned_text = text.strip() + "."
    return cleaned_text

audio_file_path = "path_to_your_audio_file.wav"
extracted_text = extract_text_from_audio(audio_file_path)
if extracted_text:
    text_with_punctuations = add_punctuations(extracted_text)
    print(text_with_punctuations)
else:
    print("Unable to extract text from audio.")
```

b. Write a sample python code to identify the URLs from the text data and extract URLs from text data

For identifying and extracting URLs from text data, you can use regular expressions. The re library in Python provides functions to work with regular expressions. Here's a sample code to extract URLs from a given text:



```python
import re

def extract_urls(text):
    urls = re.findall(r'http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\\(\\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+', text)
    return urls

text = "Here is a link to Google: https://www.google.com and another link: http://www.example.com"
urls = extract_urls(text)
print("Extracted URLs:")
for url in urls:
    print(url)
```

Outcomes: CO1: Understand fundamentals of NLP

Conclusion: (Conclusion to be based on the outcomes achieved)

Successfully implemented and understood the implementation of removal of punctuations, stop words, extra white spaces, URLs and HTML code from Text.