KJSCE/IT/LYBETCH(AI-HONOURS)/SEM-VII/NLP/2023-2024

**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."

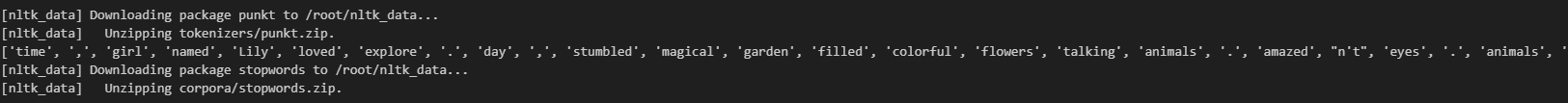
# Tokenize the text

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)



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

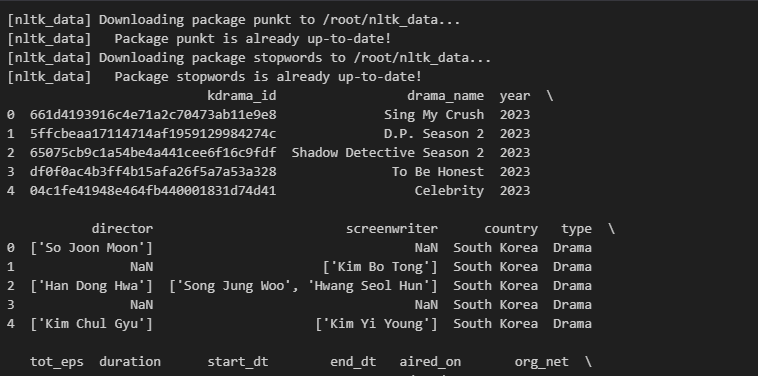
for index, row in df.iterrows():

      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)

****

**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:

            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:

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.