

**DEPARTMENT OF COMPUTER ENGINEERING**

**CSL804 Computational Lab II**

**Eighth Semester, 2021-2022 (Even Semester)**

**Name of Student :** Saurav Kumar

**Roll No.** 23

**Division :** BE – CMPN

**Day/ Session :** Monday/Afternoon

**Venue :** SLRTCE Lab 305

**Experiment No.** 2

**Title of Experiment :** To study & implement preprocessing of document (Stop Word Removal, Stemming, Lemmatisation)

**Date of Conduction :**

**Date of Submission :**

|  |  |  |
| --- | --- | --- |
| **Particulars Max. Marks Marks Obtained** | | |
| Preparedness and Efforts(PE) | **3** |  |
| Knowledge of tools(KT) | **3** |  |
| Debugging and results(DR) | **3** |  |
| Documentation(DN) | **3** |  |
| Punctuality & Lab Ethics(PL) | **3** |  |
| **Total** | **15** |  |

**Grades – Meet Expectations (3 Marks), Moderate Expectations (2 Marks), Below Expectations (1 Mark)**

**Checked and Verified by Name of Faculty :** Prof. Neelam Kulkarni

**Signature :**

**Date :**

EXPERIMENT NO: 2

PREPROCESSING OF DOCUMENT

**AIM:** To study and implement preprocessing of document (Stop Word Removal, Stemming, Lemmatisation)

**SOFTWARE:** Python, NLTK, CLTK

# THEORY:

Take any document in English and perform following preprocessing steps:

* Stop Word Removal
* Stemming using (Porter, Lancaster, Snowball) stemmers
* Lemmatization

Take any document in natural language (Hind / Marathi) and perform:

* Stop word removal
* Stemming

# IMPLEMENTATION:

STOPWORDS

Stopwords are words that do not contribute to the meaning of a sentence. Hence, they can safely be removed without causing any change in the meaning of the sentence. The NLTK library has a set of stopwords and we can use these to remove stopwords from our text and return a list of word tokens.

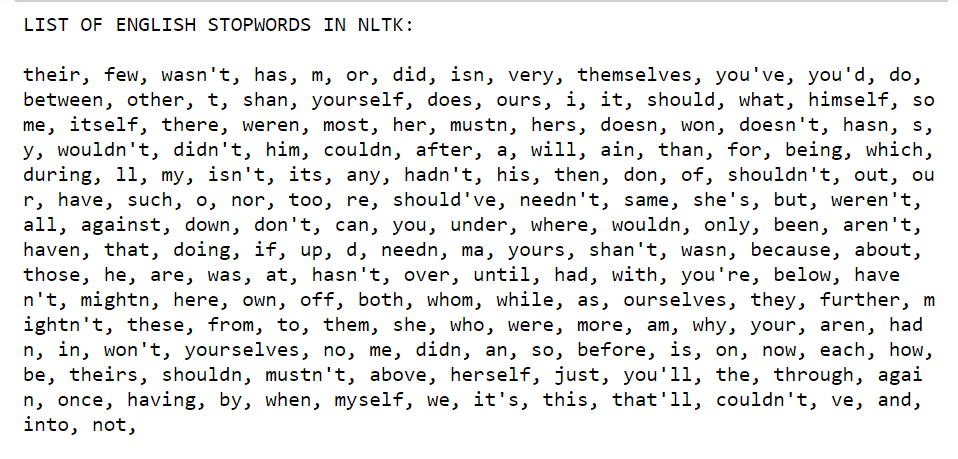
CODE

import nltk

from nltk.corpus import stopwords

stops = set(stopwords.words('english')) print(stops)

OUTPUT



REMOVAL OF STOPWORDS

To remove stop words from a text.

CODE

import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

# remove stopwords function def remove\_stopwords(text):

stop\_words = set(stopwords.words("english")) word\_tokens = word\_tokenize(text)

filtered\_text = [word for word in word\_tokens if word not in stop\_words]

return filtered\_text

example\_text = "This is a sample sentence and we are going to remove the stopwords from this."

print(remove\_stopwords(example\_text))

OUTPUT

Text

Description automatically generated

MARATHI STOPWORDS

List of Marathi stop words.

CODE

import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

from cltk.stop.marathi.stops import STOP\_LIST print(STOP\_LIST)

OUTPUT



REMOVAL OF STOPWORDS IN MARATHI

Remove stopwords in a Marathi text.

CODE

import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

from cltk.stop.marathi.stops import STOP\_LIST

# remove stopwords function def remove\_stopwords(text):

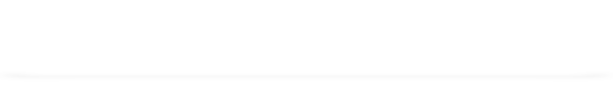
word\_tokens = word\_tokenize(text)

filtered\_text = [word for word in word\_tokens if word not in STOP\_LIST]

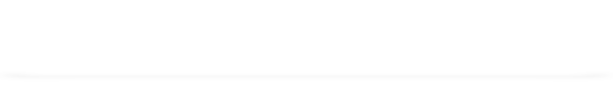
return filtered\_text

example\_text = "N"# hTNी (e˛ +T" print(remove\_stopwords(example\_text))

OUTPUT

Text

Description automatically generated

STEMMING

Stemming is the process of getting the root form of a word. Stem or root is the part to which inflectional affixes (-ed, -ize, -de, -s, etc.) are added. The stem of a word is created by removing the prefix or suffix of a word. So, stemming a word may not result in actual words.

If the text is not in tokens, then we need to convert it into tokens. After we have converted strings of text into tokens, we can convert the word tokens into their root form. There are mainly three algorithms for stemming. These are the Porter Stemmer, the Snowball Stemmer and the Lancaster Stemmer. Porter Stemmer is the most common among them.

CODE

import nltk

from nltk.stem.porter import PorterStemmer from nltk.tokenize import word\_tokenize stemmer = PorterStemmer()

# stem words in the list of tokenised words def stem\_words(text):

word\_tokens = word\_tokenize(text)

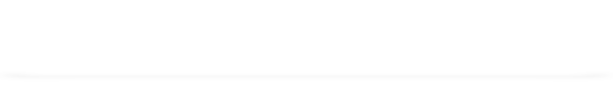
stems = [stemmer.stem(word) for word in word\_tokens] return stems

text = 'data science uses scientific methods algorithms and many types of processes'

print(stem\_words(text))

OUTPUT





PORTER VS LANCASTER STEMMER

CODE

import nltk

from nltk.stem.porter import PorterStemmer from nltk.tokenize import word\_tokenize

raw = """DENNIS: Listen, strange women lying in ponds distributing swords

... is no basis for a system of government. Supreme executive power derives from

... a mandate from the masses, not from some farcical aquatic ceremony."""

tokens = word\_tokenize(raw) porter = nltk.PorterStemmer()

lancaster = nltk.LancasterStemmer()

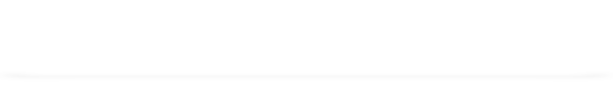
print(f"PORTER STEMMER: \n {[porter.stem(t) for t in tokens]} \n") print(f"LANCASTER STEMMER: \n {[lancaster.stem(t) for t in tokens]}

\n")

OUTPUT

Text

Description automatically generated

LEMMATIZATION

Like stemming, lemmatization also converts a word to its root form. The only difference is that lemmatization ensures that the root word belongs to the language. We will get valid words if we use lemmatization. In NLTK, we use the WordNetLemmatizer to get the lemmas of words. We also need to provide a context for the lemmatization. So, we add the part-of-speech as a parameter.

CODE

import nltk

from nltk.stem import WordNetLemmatizer from nltk.tokenize import word\_tokenize lemmatizer = WordNetLemmatizer()

# lemmatize string

def lemmatize\_word(text):

word\_tokens = word\_tokenize(text)

# provide context i.e. part-of-speech

lemmas = [lemmatizer.lemmatize(word, pos ='v') for word in word\_tokens]

return lemmas

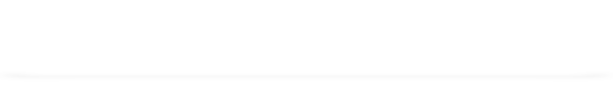
text = 'data science uses scientific methods algorithms and many types of processes'

print(lemmatize\_word(text))

OUTPUT

Text

Description automatically generated



# CONCLUSION:

Thus we have studied and implemented preprocessing of document (Stop Word Removal, Stemming, Lemmatisation).