TEXT SUMMARIZER:

Both the methods work similar, And the method 2 is more accurate than method 1.

METHOD 1:

# Tokenizing the text

from spacy.lang.en.stop\_words import STOP\_WORDS

from string import punctuation

from collections import Counter

from heapq import nlargest

from nltk.corpus import stopwords

stopWords = set(stopwords.words("english"))

words = word\_tokenize(text)

# Creating a frequency table to keep the score of each word.

freqTable = dict()

for word in words:

word = word.lower()

if word in stopWords:

continue

if word in freqTable:

freqTable[word] += 1

else:

freqTable[word] = 1

# Creating a dictionary to keep the score of each sentence.

sentences = sent\_tokenize(text)

sentenceValue = dict()

for sentence in sentences:

for word, freq in freqTable.items():

if word in sentence.lower():

if sentence in sentenceValue:

sentenceValue[sentence] += freq

else:

sentenceValue[sentence] = freq

sumValues = 0

for sentence in sentenceValue:

sumValues += sentenceValue[sentence]

# Average value of a sentence from the original text

average = int(sumValues / len(sentenceValue))

# Storing sentences into our summary.

summary = ''

for sentence in sentences:

if (sentence in sentenceValue) and (sentenceValue[sentence] > (average)):

summary += " " + sentence

print("Summarised Text: \n")

print(summary)

METHOD 2:

import re

import nltk

# nltk.download('stopwords')

# nltk.download('punkt')

from nltk.corpus import stopwords

RawText=RawText.lower()

RawText

Text = re.sub('[^.,a-zA-Z0-9]', ' ', RawText)

Text = re.sub('\s+', ' ', Text)

Text

Sentence = nltk.sent\_tokenize(Text)

Sentence

stopwords = nltk.corpus.stopwords.words('english')

word\_frequencies = {}

for word in nltk.word\_tokenize(Text):

    if word not in stopwords:

        if word not in word\_frequencies:

            word\_frequencies[word] = 1

        else:

            word\_frequencies[word] += 1

maximum\_frequency = max(word\_frequencies.values())

for word in word\_frequencies:

    word\_frequencies[word] = word\_frequencies[word] / maximum\_frequency

sentence\_scores = {}

for sentence in Sentence:

    for word in nltk.word\_tokenize(sentence):

        if word in word\_frequencies and len(sentence.split(' ')) < 30:

            if sentence not in sentence\_scores:

                sentence\_scores[sentence] = word\_frequencies[word]

            else:

                sentence\_scores[sentence] += word\_frequencies[word]

print(word\_frequencies)

print(sentence\_scores)

# get top 5 sentences

import heapq

summary = heapq.nlargest(2, sentence\_scores, key=sentence\_scores.get)

print(" ".join(summary))

KEYPHRASE EXTRACTION:

METHOD 1:

from rake\_nltk import Rake

r=Rake()

r.extract\_keywords\_from\_text(summary)

Phrase=r.get\_ranked\_phrases()[:4]

print(Phrase)