

**AIM: USING NLP TO COMPLETE ANALYTICAL TASKS SUCH AS GENERATING DOCUMENT  
ABSTRACTS(TEXT SUMMARIZATION)**

**TEXT SUMMARIZATION:**

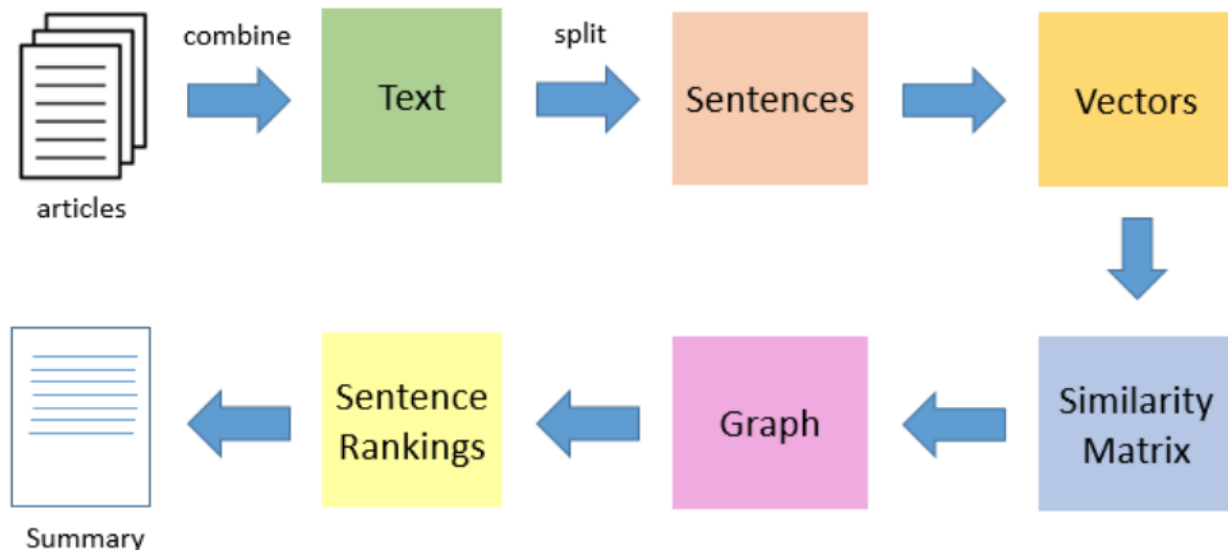
Text summarization is a very useful and important part of Natural Language Processing (NLP). First let us talk about what text summarization is. Suppose we have too many lines of text data in any form, such as from articles or magazines or on social media. We have time scarcity so we want only a nutshell report of that text. We can summarize our text in a few lines by removing unimportant text and converting the same text into smaller semantic text form.

Now let us see how we can implement NLP in our programming. We will take a look at all the approaches later, but here we will classify approaches of NLP.

**TEXT SUMMARIZATION**

In this approach we build algorithms or programs which will reduce the text size and create a summary of our text data. This is called automatic text summarization in machine learning.

Text summarization is the process of creating shorter text without removing the semantic structure of text.



**A) ENGLISH:**

**SOURCE CODE:**

```

# importing libraries
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize

nltk.download('punkt')
nltk.download('stopwords')
  
```

```
# Input text - to summarize
text = "In late summer 1945, guests are gathered for the wedding reception of Don Vito
Corleones " + \
    "daughter Connie (Talia Shire) and Carlo Rizzi (Gianni Russo). Vito (Marlon Brando)," +
    \
    "the head of the Corleone Mafia family, is known to friends and associates as Godfather. "
+ \
    "He and Tom Hagen (Robert Duvall), the Corleone family lawyer, are hearing requests
for favors " + \
    "because, according to Italian tradition, no Sicilian can refuse a request on his daughter's
wedding " + \
    "day. One of the men who asks the Don for a favor is Amerigo Bonasera, a successful
mortician " + \
    "and acquaintance of the Don, whose daughter was brutally beaten by two young men
because she" + \
    "refused their advances; the men received minimal punishment from the presiding
judge. " + \
    "The Don is disappointed in Bonasera, who'd avoided most contact with the Don due to
Corleone's" + \
    "nefarious business dealings. The Don's wife is godmother to Bonasera's shamed
daughter, " + \
    "a relationship the Don uses to extract new loyalty from the undertaker. The Don agrees
" + \
    "to have his men punish the young men responsible (in a non-lethal manner) in return
for " + \
    "future service if necessary."
```

```
# Tokenizing the text
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] > (1.2 * average)):
        summary += " " + sentence
print(summary)
```

**OUTPUT:**

```
❏ In late summer 1945, guests are gathered for the wedding reception of Don Vito Corleones daughter Connie (Talia Shire) and Carlo Rizzi (Gianni Russo). He and Tom
[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!
```

In late summer 1945, guests are gathered for the wedding reception of Don Vito Corleones daughter Connie (Talia Shire) and Carlo Rizzi (Gianni Russo). He and Tom Hagen (Robert Duvall), the Corleone family lawyer, are hearing requests for favors because, according to Italian tradition, no Sicilian can refuse a request on his daughter's wedding day. One of the men who asks the Don for a favor is Amerigo Bonasera, a successful mortician and acquaintance of the Don, whose daughter was brutally beaten by two young men because she refused their advances; the men received minimal punishment from the presiding judge.

**B) HINDI:****SOURCE CODE:**

```
import nltk
from nltk import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
import math

nltk.download('punkt')
nltk.download('stopwords')
```

text = "एकनाथ शिंदे ने दावा किया है कि उनके पास 40 से अधिक विधायकों का समर्थन है. यह आंकड़ा दलबदल विरोधी कानून को मात देने के लिए जरूरी दो-तिहाई की आवश्यकता को पूरा करता है. ऐसे में अगर शिंदे की मांग को स्वीकार नहीं किया जाता है, तो शिंदे डिप्टी स्पीकर झिरवाल से मांग करेंगे कि उनके गुट को असली शिवसेना के रूप में मान्यता दी जाए. अगर ऐसा हो जाता है तो शिवसेना दो हिस्सों में बंट जाएंगी. अभी सवाल है कि डिप्टी स्पीकर क्या कदम उठाएंगे. उनके सामने क्या चुनौती आती है और वे क्या निर्णय लेते हैं, यह देखने वाली बात होगी."

```
#print(text)
stop=open('hindi-stopwords.txt')
#stop=open('hindi-stop-words-2.txt')
stopwords=[]
for x in stop:
    stopwords.append(x)

def createfrequencytable(text_string) -> dict:
    stopWords = set(stopwords)
    words = word_tokenize(text_string)
    ps = PorterStemmer()

    freqTable = dict()
    for word in words:
        word=str(word)
        word = ps.stem(word)
        if word in stopWords:
            continue
        if word in freqTable:
            freqTable[word] += 1
        else:
            freqTable[word] = 1
    return freqTable

ft=createfrequencytable(text)
print(ft)

#tokenization of sentences
sentences = sent_tokenize(text) # NLTK function
total_documents = len(sentences)
print(sentences)
print(total_documents)

def scoresentences(sentences, freqTable) -> dict:
    sentenceValue = dict()
    for sentence in sentences:
        word_count_in_sentence = (len(word_tokenize(sentence)))
        for wordValue in freqTable:
            if wordValue in sentence.lower():
                if sentence[:10] in sentenceValue:
```

```
sentenceValue[sentence[:10]] += freqTable[wordValue]
else:
    sentenceValue[sentence[:10]] = freqTable[wordValue]
sentenceValue[sentence[:10]] = sentenceValue[sentence[:10]] // word_count_in_sentence
return sentenceValue

sentence_val=scoresentences(sentences, ft)
print(sentence_val)

def findaverage_score(sentenceValue) -> int:
    sumValues = 0
    for entry in sentenceValue:
        sumValues += sentenceValue[entry]
    # Average value of a sentence from original text
    average = int(sumValues / len(sentenceValue))
    return average

thresh=findaverage_score(sentence_val)
print(thresh)

def _generate_summary(sentences, sentenceValue, threshold):
    sentence_count = 0
    summary = ""
    for sentence in sentences:
        if sentence[:10] in sentenceValue and sentenceValue[sentence[:10]] > (threshold):
            summary += " " + sentence
            sentence_count += 1
    return summary

summary = _generate_summary(sentences, sentence_val, 1.5 * thresh)
print(summary)
```

**OUTPUT:**

```
C:\> [एकनाथ]: 1, 'शिंदे': 3, 'ने': 1, 'दावा': 1, 'किया': 2, 'है': 7, 'कि': 3, 'उनके': 3, 'पास': 1, '40': 1, 'से': 2,
[एकनाथ शिंदे ने दावा किया है कि उनके पास 40 से अधिक विधायकों का समर्थन है, "यह आंकड़ा दलबदल विरोधी कांग्रेस को मत देने के लिए अपनी दो-तिहाई की अवधि\u200dयकता को पूरा करता है", "ऐसे में अगर शिंदे की मांग को स्\u200dअसली विरोधी स्\u200dअसली विरोधी स्
6
[एकनाथ शिंदे': 41, 'यह आंकड़ा ': 38, 'ऐसे में क्या': 65, 'अगर ऐसा हो': 33, 'अभी सवाल है': 27, 'उनके समर्थन': 2]
34
ऐसे में अगर शिंदे की मांग को स्वीकार नहीं किया जात है, तो शिंदे किसी स्वीकार किए गए से मांग करेगे कि उनके टुक को असली विरोधी के रूप में मान्यता दी जाए.
[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!
```

```
{'एकनाथ': 1, 'शिंदे': 3, 'ने': 1, 'दावा': 1, 'किया': 2, 'है': 7, 'कि': 3, 'उनके': 3, 'पास': 1, '40': 1, 'से': 2,
'अधिक': 1, 'विधायकों': 1, 'का': 1, 'समर्थन': 1, '': 6, 'यह': 2, 'आंकड़ा': 1, 'दलबदल': 1, 'विरोधी': 1,
'कानून': 1, 'को': 4, 'मात': 1, 'देने': 1, 'के': 2, 'लिए': 1, 'जरूरी': 1, 'दो-तिहाई': 1, 'की': 2,
'आवश्\u200dयकता': 1, 'पूरा': 1, 'करता': 1, 'ऐसे': 1, 'मैं': 3, 'अगर': 2, 'मांग': 2, 'स्\u200dवीकार': 1,
'नहीं': 1, 'जाता': 2, '': 2, 'तो': 2, 'डिप्\u200dटी': 2, 'स्\u200dपीकर': 2, 'झिरवाल': 1, 'करेंगे': 1, 'गुट': 1,
'असली': 1, 'शिवसेना': 2, 'रूप': 1, 'मान्\u200dयता': 1, 'दी': 1, 'जाए': 1, 'ऐसा': 1, 'हो': 1, 'दो': 1,
'हिस्\u200dसों': 1, 'बंट': 1, 'जाएंगी': 1, 'अभी': 1, 'सवाल': 1, 'क्\u200dया': 3, 'कदम': 1, 'उठाएंगे': 1,
```

'सामने': 1, 'चुनौती': 1, 'आती': 1, 'और': 1, 'वे': 1, 'निर्णय': 1, 'लेते': 1, 'हैं': 1, 'देखने': 1, 'वाली': 1, 'बात': 1, 'होगी': 1}

[ 'एकनाथ शिंदे ने दावा किया है कि उनके पास 40 से अधिक विधायकों का समर्थन है.', 'यह आंकड़ा दलबदल विरोधी कानून को मात देने के लिए जरूरी दो-तिहाई की आवश्यकता को पूरा करता है.', 'ऐसे में अगर शिंदे की मांग को स्वीकार नहीं किया जाता है, तो शिंदे डिप्टी स्पीकर झिरवाल से मांग करेंगे कि उनके गुट को असली शिवसेना के रूप में मान्यता दी जाए.', 'अगर ऐसा हो जाता है तो शिवसेना दो हिस्सों में बंट जाएंगी.', 'अभी सवाल है कि डिप्टी स्पीकर कदम उठाएंगे.', 'उनके सामने चुनौती आती है और वे निर्णय लेते हैं, यह देखने वाली बात होगी. ']

6

{ 'एकनाथ शिंदे': 41, 'यह आंकड़ा': 38, 'ऐसे में अगर': 65, 'अगर ऐसा हो': 33, 'अभी सवाल है': 27, 'उनके सामने': 2 }

34

#### SUMMARIZED TEXT:

ऐसे में अगर शिंदे की मांग को स्वीकार नहीं किया जाता है, तो शिंदे डिप्टी स्पीकर झिरवाल से मांग करेंगे कि उनके गुट को असली शिवसेना के रूप में मान्यता दी जाए.

#### C) LEXRANK:

LexRank is an unsupervised approach to text summarization based on graph-based centrality scoring of sentences. The main idea is that sentences "recommend" other similar sentences to the reader. Thus, if one sentence is very similar to many others, it will likely be a sentence of great importance. The importance of this sentence also stems from the importance of the sentences "recommending" it. Thus, to get ranked highly and placed in a summary, a sentence must be similar to many sentences that are in turn also similar to many other sentences. This makes intuitive sense and allows the algorithms to be applied to any arbitrary new text.

#### SOURCE CODE:

```
from lexrank import STOPWORDS, LexRank
from path import Path

documents = []
documents_dir = Path('/content/drive/MyDrive/bbc-fulltext/bbc/politics')

for file_path in documents_dir.files('*.txt'):
    with file_path.open(mode='rt', encoding='utf-8') as fp:
        documents.append(fp.readlines())
```

```
lrx = LexRank(documents, stopwords=STOPWORDS['en'])

sentences = [
    'One of David Cameron\'s closest friends and Conservative allies, '
    'George Osborne rose rapidly after becoming MP for Tatton in 2001.',

    'Michael Howard promoted him from shadow chief secretary to the '
    'Treasury to shadow chancellor in May 2005, at the age of 34.',

    'Mr Osborne took a key role in the election campaign and has been at '
    'the forefront of the debate on how to deal with the recession and '
    'the UK\'s spending deficit.',

    'Even before Mr Cameron became leader the two were being likened to '
    'Labour\'s Blair/Brown duo. The two have emulated them by becoming '
    'prime minister and chancellor, but will want to avoid the spats.',

    'Before entering Parliament, he was a special adviser in the '
    'agriculture department when the Tories were in government and later '
    'served as political secretary to William Hague.',

    'The BBC understands that as chancellor, Mr Osborne, along with the '
    'Treasury will retain responsibility for overseeing banks and '
    'financial regulation.',

    'Mr Osborne said the coalition government was planning to change the '
    'tax system \'to make it fairer for people on low and middle '
    'incomes\', and undertake \'long-term structural reform\' of the '
    'banking sector, education and the welfare state.',
]

# get summary with classical LexRank algorithm
summary = lrx.get_summary(sentences, summary_size=2, threshold=.1)
print(summary)

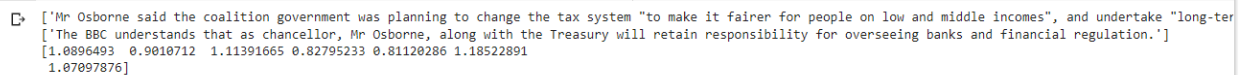
# ['Mr Osborne said the coalition government was planning to change the tax '
# 'system "to make it fairer for people on low and middle incomes", and '
# 'undertake "long-term structural reform" of the banking sector, education and '
# 'the welfare state.',
# 'The BBC understands that as chancellor, Mr Osborne, along with the Treasury '
# 'will retain responsibility for overseeing banks and financial regulation.']

# get summary with continuous LexRank
summary_cont = lrx.get_summary(sentences, threshold=None)
print(summary_cont)

# ['The BBC understands that as chancellor, Mr Osborne, along with the Treasury '
# 'will retain responsibility for overseeing banks and financial regulation.']
```

```
# get LexRank scores for sentences
# 'fast_power_method' speeds up the calculation, but requires more RAM
scores_cont = lxr.rank_sentences(
    sentences,
    threshold=None,
    fast_power_method=False,
)
print(scores_cont)
```

**OUTPUT:**



```
['Mr Osborne said the coalition government was planning to change the tax system "to make it fairer for people on low and middle incomes", and undertake "long-term structural reform" of the banking sector, education and the welfare state.', 'The BBC understands that as chancellor, Mr Osborne, along with the Treasury will retain responsibility for overseeing banks and financial regulation.']
[1.0896493 0.9010712 1.11391665 0.82795233 0.81120286 1.18522891 1.07097876]
```

['Mr Osborne said the coalition government was planning to change the tax system "to make it fairer for people on low and middle incomes", and undertake "long-term structural reform" of the banking sector, education and the welfare state.', 'The BBC understands that as chancellor, Mr Osborne, along with the Treasury will retain responsibility for overseeing banks and financial regulation.']

['The BBC understands that as chancellor, Mr Osborne, along with the Treasury will retain responsibility for overseeing banks and financial regulation.']

[1.0896493 0.9010712 1.11391665 0.82795233 0.81120286 1.18522891 1.07097876]

**CONCLUSION:**

From this practical, I have learned & implemented the text summarization in python.