TELUGU TEXT SUMMARIZATION

K. VENKAT SUBHASH
S20200010081
CSE
IIIT Sri City
Andhra Pradesh, India
Email: venkatsubhash.k20@iiits.in

P.M.S SATYA KARTHIK
S20200010158
CSE
IIIT Sri City
Andhra Pradesh, India
Email: srisatyakarthik.p20@iiits.in

P.SAI SHARAN
S20200010174
CSE
IIIT Sri City
Andhra Pradesh, India
Email: saisharan.p20@iiits.in

Y C SAI THAPAN
S20200010236
CSE
IIIT Sri City
Andhra Pradesh, India
Email: chandrasekhar.s20@iiits.in

Abstract -

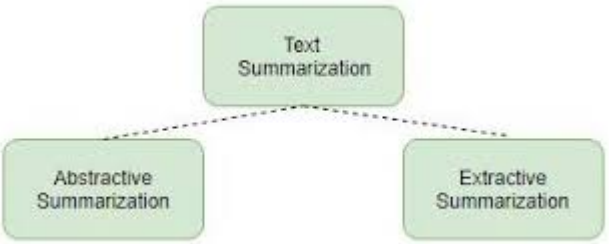
Text summarizing is the act of condensing a document's text to produce a description that retains the key ideas from the original text, also known as a text summary. Text summarizing is the act of condensing a document's text to produce a description that retains the key ideas from the original text, also known as a text summary. Given the daily increase of data, it is becoming more and more challenging for humans to manually summarize lengthy texts. The two methods of record summarizing that are examined in this research are abstract and extract text summarization

I.Introduction-

One of the cool NLP (Natural Language Processing) applications is the text summarizer, which condenses long documents into shorter ones while keeping all of the crucial information. Everyone in the world uses language as their major means of communication. Text is the primary format used to store information. Large text documents can be time-consuming to read. We can condense

the text and shorten the reading time to address this issue. Text summarizing entails outlining the main ideas of the text in order to provide a concise, coherent, and fluid summary of a lengthy text document. Summarizing cuts down on reading time. Summaries facilitate the selection of documents for research. Indexing performance is improved via automatic summarization. Compared to human summarizers, automatic systems are less prejudiced. Because they offer individualized Information, personalized summaries are helpful in question-answering systems.

There are two types of text summarization:



1)***Extractive summarization:***

These techniques focus on extracting various textual components, including phrases and sentences, then stacking them. Collectively to provide a summary. As a result, choosing the appropriate sentences to summarize is crucial when using an extractive technique.

2)***Abstractive summarization:***

These methods create an entirely new summary using cutting-edge NLP algorithms. It's possible that certain portions of this summary weren't even in the original text.

I.1.Motivation-

Because people can obtain the most crucial information quickly and thereby save time, short news is now widely read everywhere. In order to create a new, shorter text, abstractive summarization techniques aim to produce a summary by interpreting the text using sophisticated natural language techniques. The fundamental goal of an automated text summarizer is to eliminate the challenges associated with manually summarizing vast amounts of information from several sources.

II.State of the art/Background-

We'll investigate numerous options for carrying out this project. Text summarization can be applied in a variety of ways:

Few approaches for this task are as follows :

1. Text Summarization using Bert's Google model.
2. Text summarization using T5 transformer model.
3. Text Summarisation in nlp using NLTK library in Python.
4. Text Summarization in NLP using spaCy.

Reference links:

1. Approaches to Text Summarization: An Overview - KDnuggets
2. Text Summarization for Telugu Document
3. Understanding Automatic Text Summarization-1: Extractive Methods | by Abhijit Roy | Towards Data Science
4. Text Summarization with NLP: TextRank vs Seq2Seq vs BART | by Mauro Di Pietro | Towards Data Science
5. Text summarization using NLP

III.Proposed system-

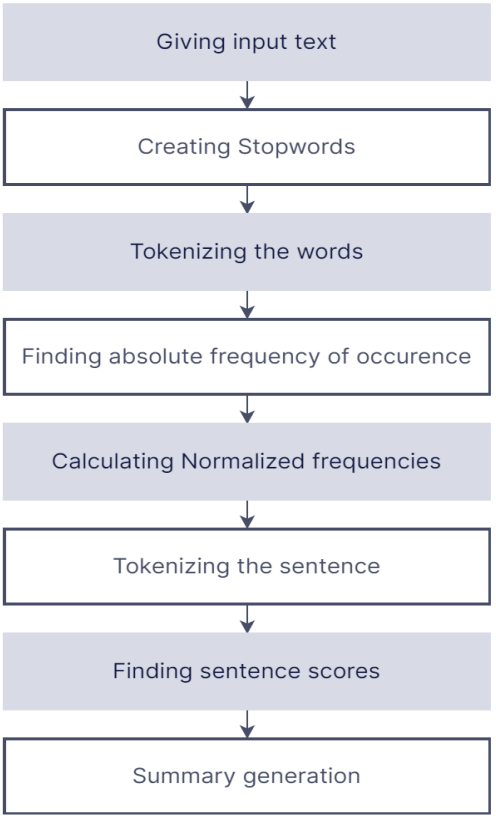
Our suggested approach is capable of text summarization "with" and "without" machine learning and deep learning libraries. We used NLP methods like text cleaning, word tokenization, sentence tokenization, word frequency, and summarization for our project.

Working model of our project:

The steps Involved in our project are as follows -

Working Architecture of our project:

1. Giving Text Input.
2. Creating Stopwords.
3. Tokenizing the words.
4. Find absolute frequency of occurrence.
5. Calculating Normalized frequencies
6. Tokenizing the Sentence.
7. Finding Sentence Score.
8. Summary Generation.



Giving Text Input:

Our technology allows users to enter data in text format without any restrictions on word count or sentence length, and also data is stored in json format to train the model for abstractive summarization.

Creating Stopwords:

Stop words are function (filler) words—words with little to no semantic content that support the grammatical structure of a phrase. These words have little of a part in information retrieval or determining which sentence is more crucial. We disregard these filler words so that we can determine how significant a sentence actually is.

Tokenization:

Tokenization is the process of breaking down a phrase, sentence, paragraph, or even an entire text document into simpler components, like individual words or phrases. Tokens are the name for each of these smaller components. For this procedure, we have been using the Split function. In this procedure, the punctuation marks are swapped out for spaces in the split function, causing each sentence, each paragraph, and the entire document to be divided into single words or tokens. These tokens aid in context comprehension or model development for NLP. By examining the word order in the text, tokenization aids in comprehending the text's meaning. You can tokenize individual words or entire sentences.

Finding absolute frequency of occurrence:

The development of a dictionary of word frequencies. Using information from the text, we determine the term frequency of the words. We do not determine the term frequency of stopwords or punctuation. When a word is used for the first time, its frequency will be 1, and when it is used for the second time, its frequency will be increased by one.

(i) Maximum Frequency

Obtaining the word with highest frequency from the word Frequency.

(ii) Normalized frequency

We will divide each word frequency with maximum frequency, this is to bring the normalized frequency.

Tokenizing the sentences:

Text is divided into sentences through the process of sentence tokenization. These tokenizers operate by dividing the words with spaces and punctuation. These tokens aid in context comprehension or model development for NLP. By examining the word order, the tokenization aids in deciphering the text's meaning.

Finding Sentence Scores:

After removing stopwords from the phrase, we added up the normalized word frequencies of the remaining words to determine the sentence scores.

Summary Generation:

The summary is being created by initializing it with the opening phrase. We add sentences with high-quality scores to the summary and then add the final phrase to the summary.

IV.Results :

1.*Giving Text Input:*

```
import string
import re
InputText = "పాకిస్థాన్•తో జరిగిన మ్యాచ్•లో అసాధారణ పోరాటంతో భారత్•కు విజయాన్ని అందించిన విరాట్ కోహ్లి(Virat Kohli)పై ప్రపంచవ్యాప్తంగా
```

2.*Creating Stopwords:*

We are creating stop words manually. (telugu stop words).

```
stopwords = ["పోరాటంతో", "స్టేడియంలోని","బంతుల్లో", "ల్లో", "వరకు", "పై", "అదే"]
```

3.*Tokenization:*

```
def remove_sw(sentences):
    all_words_without_sw = []
    for sentence in sentences:
        words = sentence.split(' ')
        words_without_sw = [word for word in words if not word in stopwords]
        all_words_without_sw = words_without_sw+all_words_without_sw
    for w in all_words_without_sw:
        if ((w == ' ') or (w == '')):
            all_words_without_sw.remove(w)
    return all_words_without_sw
```

```
--------------------------------------------------------------------------
All words without stop words
--------------------------------------------------------------------------

['కానీ', 'గౌతమ్', 'అనుకుంటున్నట్టుగా', 'అతని', 'తల్లిదండ్రులను', 'ఎవరూ', 'చంపేయలేదని', 'తల్లిదండ్రులు', 'లేని', 'అనాథ', 'అ
యిన', 'గౌతం', 'సృష్టించుకున్న', 'ఊహల్లో', 'మాత్రమే', 'అతని', 'తల్లిదండ్రులు', 'వారిని', 'చంపిన', 'హంతకులు', 'ఉన్నారని',
'అందరి', 'నమ్మకం.', 'వాళ్ళని', 'చంపాలనే', 'పగతో', 'ఆ', 'ముగ్గురిని', 'గుర్తు', 'పెట్టుకుంటాడు', 'గౌతం', 'చిన్నప్పుడు', 'కొన్ని',
'కారణాల', 'వల్ల', 'అతని', 'తల్లిదండ్రులని', 'ముగ్గురు', 'వ్యక్తులు', 'కలసి', 'చంపుతారు', 'గౌతంకి', 'అతని', 'తల్లిదండ్రులు', 'ఎ
లా', 'ఉంటారో', 'గుర్తు', 'ఉండదు', 'గౌతమ్', 'పదేళ్ళ', 'వయసులో', 'అతని', 'తల్లి', 'దండ్రులను', 'ఎవరో', 'చంపేయడంతో', 'అ
నాథ', 'శరణాలంలో', 'పెరుగుతాడు', 'గౌతంకి', 'మెదడుకి', 'సంబంధిచిన', 'ఇంటిగ్రేషన్', 'డిజార్డర్', 'మెదడు', 'గుర్తు', 'పెట్టుకునే',
'సామర్థ్యం', 'తక్కువగా', 'జబ్బు', 'ఉంటుంది', 'చిత్ర', 'కథ', 'విషయానికి', 'వస్తే', '', 'గౌతం', 'ఒక', 'రాక్', 'స్టార్']

--------------------------------------------------------------------------
```

4.*Finding Term Frequencies:*

```python
def assign_words_tf():
    unique_words.append(tockens_without_sw[0])
    tf_unique_words.append(1)
    for i in range(len(tockens_without_sw)):
        if(i > 0):
            for j in range(len(unique_words)):
                if (tockens_without_sw[i].__eq__(unique_words[j])):
                    tf_unique_words[j] = tf_unique_words[j]+1
                    flag = 0
                    break
                else:
                    flag = 1
            if(flag == 1):
                unique_words.append(tockens_without_sw[i])
                tf_unique_words.append(1)
```

```
--------------------------------------------------------------------------
Words - TF
--------------------------------------------------------------------------

కానీ-1 , గౌతమ్-2 , అనుకుంటున్నట్టుగా-1 , అతని-5 , తల్లిదండ్రులను-1 , ఎవరూ-1 , చంపేయలేదని-1 , తల్లిదండ్రులు-3 , లేని-1 , అ
నాథ-2 , అయిన-1 , గౌతం-3 , సృష్టించుకున్న-1 , ఊహల్లో-1 , మాత్రమే-1 , వారిని-1 , చంపిన-1 , హంతకులు-1 , ఉన్నారని-1 , అందరి-
1 , నమ్మకం.-1 , వాళ్ళని-1 , చంపాలనే-1 , పగతో-1 , ఆ-1 , ముగ్గురిని-1 , గుర్తు-3 , పెట్టుకుంటాడు-1 , చిన్నప్పుడు-1 , కొన్ని-1 , కార
ణాల-1 , వల్ల-1 , తల్లిదండ్రులని-1 , ముగ్గురు-1 , వ్యక్తులు-1 , కలసి-1 , చంపుతారు-1 , గౌతంకి-2 , ఎలా-1 , ఉంటారో-1 , ఉండదు-1 , ప
దేళ్ళ-1 , వయసులో-1 , తల్లి-1 , దండ్రులను-1 , ఎవరో-1 , చంపేయడంతో-1 , శరణాలంలో-1 , పెరుగుతాడు-1 , మెదడుకి-1 , సంబంధిచిన-
1 , ఇంటిగ్రేషన్-1 , డిజార్డర్-1 , మెదడు-1 , పెట్టుకునే-1 , సామర్థ్యం-1 , తక్కువగా-1 , జబ్బు-1 , ఉంటుంది-1 , చిత్ర-1 , కథ-1 , విష
యానికి-1 , వస్తే-1 , -1 , ఒక-1 , రాక్-1 , స్టార్-1

--------------------------------------------------------------------------
```

7.*Normalized Term Frequencies:*

```python
print('\n----------------------------------------------------------------------------\nWords - Normallized TF\n--------------------
for i in range(len(unique_words)):
    print(unique_words[i] + '-' + str(normalized_tf_unique_words[i]))
```

```
--------------------------------------------------------------------------
Words - Normallized TF
--------------------------------------------------------------------------

కానీ-0.2 , గౌతమ్-0.4 , అనుకుంటున్నట్టుగా-0.2 , అతని-1.0 , తల్లిదండ్రులను-0.2 , ఎవరూ-0.2 , చంపేయలేదని-0.2 , తల్లిదండ్రులు-0.6 ,
లేని-0.2 , అనాథ-0.4 , అయిన-0.2 , గౌతం-0.6 , సృష్టించుకున్న-0.2 , ఊహల్లో-0.2 , మాత్రమే-0.2 , వారిని-0.2 , చంపిన-0.2 , హంత
కులు-0.2 , ఉన్నారని-0.2 , అందరి-0.2 , నమ్మకం.-0.2 , వాళ్ళని-0.2 , చంపాలనే-0.2 , పగతో-0.2 , ఆ-0.2 , ముగ్గురిని-0.2 , గుర్తు-0.6 ,
పెట్టుకుంటాడు-0.2 , చిన్నప్పుడు-0.2 , కొన్ని-0.2 , కారణాల-0.2 , వల్ల-0.2 , తల్లిదండ్రులని-0.2 , ముగ్గురు-0.2 , వ్యక్తులు-0.2 , కలసి-0.2 ,
చంపుతారు-0.2 , గౌతంకి-0.4 , ఎలా-0.2 , ఉంటారో-0.2 , ఉండదు-0.2 , పదేళ్ళ-0.2 , వయసులో-0.2 , తల్లి-0.2 , దండ్రులను-0.2 , ఎవరో-
0.2 , చంపేయడంతో-0.2 , శరణాలంలో-0.2 , పెరుగుతాడు-0.2 , మెదడుకి-0.2 , సంబంధిచిన-0.2 , ఇంటిగ్రేషన్-0.2 , డిజార్డర్-0.2 , మెద
డు-0.2 , పెట్టుకునే-0.2 , సామర్థ్యం-0.2 , తక్కువగా-0.2 , జబ్బు-0.2 , ఉంటుంది-0.2 , చిత్ర-0.2 , కథ-0.2 , విషయానికి-0.2 , వస్తే-0.2 ,
-0.2 , ఒక-0.2 , రాక్-0.2 , స్టార్-0.2
```

8.*Finding Sentence Scores:*

```
print('----------------------------------------------------------------------------\nSentence - Score\n---------------------------
while (count > 0):
    MaxValue = max(sentence_score)
    index = sentence_score.index(max(sentence_score))
    print(sentences[index]+'----'+str(sentence_score[index])+'\n')
    if(flag1 == 1):
        final_summary = sentences[index]
        flag1 = 0
    else:
        final_summary = final_summary+'. '+sentences[index]
    sentence_score.remove(MaxValue)
    count -= 1
```

```
----------------------------------------------------------------------------
Sentence - Score
----------------------------------------------------------------------------

నాగ్పూర్ వేదికగా న్యూజిలాండ్ జరిగిన తొలి మ్యాచ్లో కోహ్లి 23 పరుగులకే ఔట్ కాగా 127 పరుగుల లక్ష్య చేధనలో భారత్ 79 పరుగులకే ఆలౌటయ్యింది----8.0

ఐదో టీ20 వరల్డ్ కప్ ఆడుతున్న విరాట్ కోహ్లి పొట్టి ఫార్మాట్ ప్రపంచ కప్లలో పది మ్యాచ్లో ఛేజింగ్కు దిగగా 135 స్ట్రైక్ రేట్తో 541 పరుగులు చేశాడు----7.666666666666668

2016లో నాగ్పూర్లో చేసిన 23 పరుగులే టీ20 వరల్డ్ కప్ లక్ష్య చేధనలో కోహ్లీకి అత్యల్ప స్కోరు కావడం గమనార్హం----7.333333333333334

కోల్కతా వేదికగా పాకిస్థాన్తో జరిగిన మ్యాచ్లో కోహ్లీ 37 బంతుల్లో 55 రన్స్తో అజేయంగా నిలిచాడు----7.333333333333333
```

9.*Number of Lines to be Summarized :*

```
count=input("Enter the final number of sentences to be there in the paragraph : ")
count=int(count)
print('----------------------------------------------------------------------------\nFinal Sentences - Score\n-------------------

print(sentences[0]+'----'+str(sentence_score[0])+'\n')
```

10.*Summary Generation:*

```
print(final_summary)
```

```
----------------------------------------------------------------------------
Final Summary
----------------------------------------------------------------------------

నాగ్పూర్ వేదికగా న్యూజిలాండ్ జరిగిన తొలి మ్యాచ్లో కోహ్లి 23 పరుగులకే ఔట్ కాగా 127 పరుగుల లక్ష్య చేధనలో భారత్ 79 పరుగులకే ఆలౌటయ్యింది. ఐదో టీ20 వరల్డ్ కప్ ఆడుతున్న విరాట్ కోహ్లి పొట్టి ఫార్మాట్ ప్రపంచ కప్లలో పది మ్యాచ్లో ఛేజింగ్కు దిగగా 135 స్ట్రైక్ రేట్తో 541 పరుగులు చేశాడు. 2016లో నాగ్పూర్లో చేసిన 23 పరుగులే టీ20 వరల్డ్ కప్ లక్ష్య చేధనలో కోహ్లీకి అత్యల్ప స్కోరు కావడం గమనార్హం. కోల్కతా వేదికగా పాకిస్థాన్తో జరిగిన మ్యాచ్లో కోహ్లీ 37 బంతుల్లో 55 రన్స్తో అజేయంగా నిలిచాడు
```

V. Conclusion:

Text Summarization Saves Time:

Text summarization helps content editors save time and effort by generating automatic summaries rather than manually creating summaries of articles.

Text Summarization Increases Productivity Level:

Through the use of text summarization, a user is able to quickly and accurately scan a text's contents for pertinent information. So, by making the text smaller, the technology relieves the user's workload and boosts their productivity so they can focus their attention on other important tasks.

Through the simplification of the enormous amount of information that people engage with on a daily basis, automatic text summarization is a technique that permits a quantum jump in human productivity. This not only enables people to reduce the amount of reading required, but also frees up time to read and comprehend written works that might otherwise go unnoticed. Such summarizers will eventually become so well-integrated that the summaries they produce will be impossible to tell from those authored by humans.

VI. Future work:

We have implemented some sort of abstractive by changing the least normalized frequency words with nearest synonyms. We will implement the whole Abstractive Summarisation and update code in the future.