TEXT SUMMARIZER

Sri Lakshmi Prasanna Koneru
CSE
IIIT Sri City
Andhra Pradesh, India
Email: srilakshmiprasanna.k19@iiits.in

Pedapalli Saam Prasanth Deeven
CSE
IIIT Sri City
Andhra Pradesh, India
Email: saamprasanthdeevan.p19@iiits.in

Pidakala Abhinandan Babu
CSE
IIIT Sri City
Andhra Pradesh, India
Email: abhinandanbabu.p19@iiits.in

Prema Vignesh
CSE
IIIT Sri City
Andhra Pradesh, India
Email: prema.g19@iiits.in

Abstract - Text summarization automatically produces a summary containing important sentences and includes all relevant important information from the original document. There are two approaches, when viewed from the summary results, an extractive summarization and an abstractive summarization. Our project is extractive summarization. In this Term document, you will come to know about stopwords, word tokenization, sentence tokenization, word frequencies, sentence frequency and text summarization

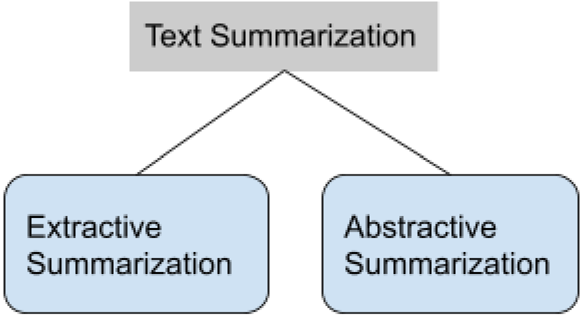
Introduction

NLP text summarizer is one of the cool applications of NLP (Natural Language Processing), which shortens the long document into a shorter one while retaining all important information from the document. Language is the primary communication tool used by everyone in daily life as a means to convey information.

Predominantly information is stored in the form of text. People don’t always have the time to read large text documents. In order to solve this problem, we can summarize the text and reduce the reading time. Text summarization is the process of creating a short, coherent, and fluent summary of a longer text document and involves the outlining of the text’s major points.

Summarisation reduces reading time. When researching documents, summaries make the selection process easier. Automatic summarization improves the effectiveness of indexing. Automatic summarization algorithms are less biased than human summarizers. Personalized summaries are useful in question-answering systems as they provide personalized Information.

There are two types of Text Summarizations, they are-



Figure(i) Text Summarization types

Extractive Summarization:

These methods rely on extracting several parts, such as phrases and sentences, from a piece of text and stack them together to create a summary. Therefore, identifying the right sentences for summarization is of utmost importance in an extractive method.

Abstractive Summarization:

These methods use advanced NLP techniques to generate an entirely new summary. Some parts of this summary may not even appear in the original text.

Motivation

Nowadays, Short news is popular everywhere because people can get the important information in a very short time, hence this saves time for the people. Abstractive summarization methods aim at producing a summary by interpreting the text using advanced natural language techniques in order to generate a new shorter text.

State of the art/Background

We will look into various ways for implementing this project. There are many ways to implement text summarization.

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) Automatic Text Summarization using Machine learning

2)Output Model Idea

3)Computational Analysis and Understanding of Natural Languages: Principles, Methods and Applications

4) Literature review of automatic single document text summarization using NLP

5) NLP based Machine Learning Approaches for Text Summarization

6) A survey of text summarization extractive techniques

7) Jurafsky, Martin.-Speech and Language Processing_ An Introduction to Natural Language

8) Recent automatic text summarization techniques

9) Automatic text summarization: Past, present and future

10)Trends and Applications of Text Summarization Techniques

Proposed System

Our Proposed system is Text Summarisation “with” using NLP techniques and “without” using any Machine learning and Deep learning Libraries. For our project, we have used the NLP techniques like Text cleaning, Word Tokenization, Sentence Tokenization, Word Frequency and Summarisation.

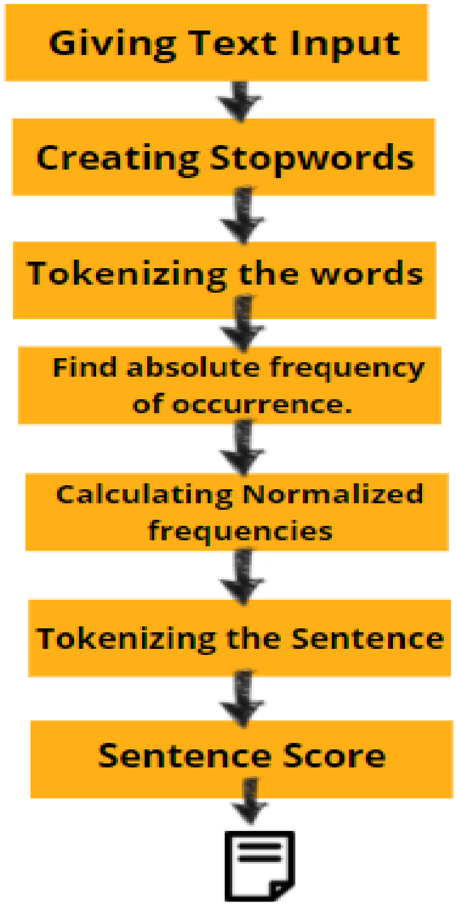
Working model of our project:

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. Sentence Score.
8. Summarizing the Text.

Architecture of our project



Giving Text Input :

In our project the user can enter the data in text format without any limitations like number of words or sentences.

Creating Stopwords :

Stop words are function (filler) words, which are words with little or no meaning that help form a sentence. These words do not have much role in retrieving information or getting to know which sentence is more important. We ignore these stopwords so that we can get to know how important a sentence really is.

Tokenizing the words :

Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens. We have used the Split function for this process. In this process we are replacing the punctuation marks with a space in a split function, so that each sentence/ paragraphs/entire document splits into single words/Tokens.These tokens help in understanding the context or developing the model for the NLP. The tokenization helps in interpreting the meaning of the text by analyzing the sequence of the words.Tokenization can be done to either separate words or sentences.

Finding Absolute frequency of occurrence :

Creating a dictionary for the word frequencies. Taking the data from the text, we calculate the term frequency of words, we will not calculate the term frequencies for the Stopwords and Punctuation marks. If any word is introduced for the first time then the frequency of that word will be 1, if the word is being occured for the second time then it will be incremented by one.

Calculating Normalized Frequency :

(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 :

Sentence tokenization is the process of splitting text into individual sentences. These tokenizers work by separating the words using punctuation and spaces. These tokens help in understanding the context or developing the model for the NLP. The tokenization helps in interpreting the meaning of the text by analyzing the sequence of the words.

Sentence Scores :

We calculated the sentence scores by adding up the normalized word frequencies of the words present in the sentence after ignoring the stopwords.

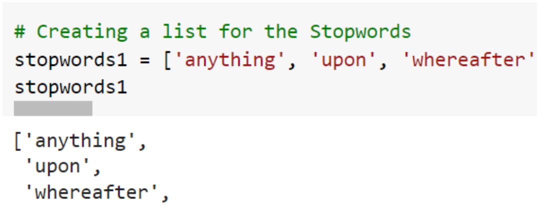
Text Summarization :

We are generating the summary by initializing it with the first sentence. We are adding the sentences with high sentence scores to the summary and finally adding the last sentence to the summary.

Results

Text summarization automatically produces a summary containing important sentences and includes all relevant important information from the original document.

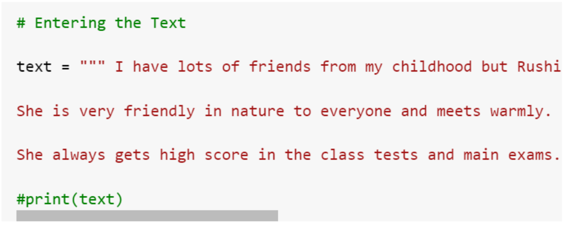
Creating a List of Stopwords :



```
# Creating a list for the Stopwords
stopwords1 = ['anything', 'upon', 'whereafter'
stopwords1
```

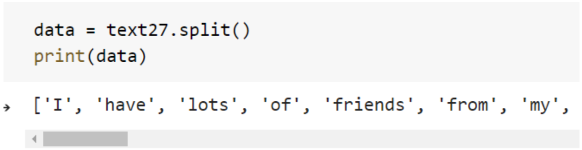
```
['anything',
 'upon',
 'whereafter',
```

Entering the Text :



```
# Entering the Text

text = """ I have lots of friends from my childhood but Rushi

She is very friendly in nature to everyone and meets warmly.

She always gets high score in the class tests and main exams.

#print(text)
```

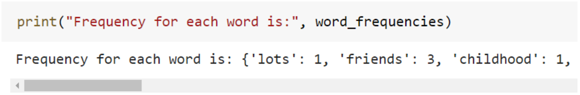
Tokenization of Words :



```
data = text27.split()
print(data)
```

```
['I', 'have', 'lots', 'of', 'friends', 'from', 'my',
```

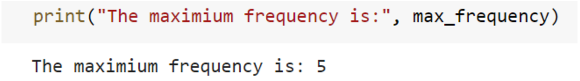
Calculating Word Frequencies :



```
print("Frequency for each word is:", word_frequencies)
```

```
Frequency for each word is: {'lots': 1, 'friends': 3, 'childhood': 1,
```

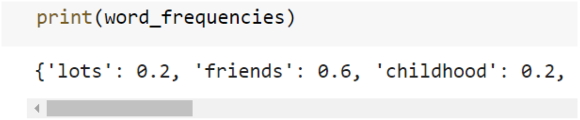
Calculating Maximum Frequency :



```
print("The maximium frequency is:", max_frequency)
```

```
The maximium frequency is: 5
```

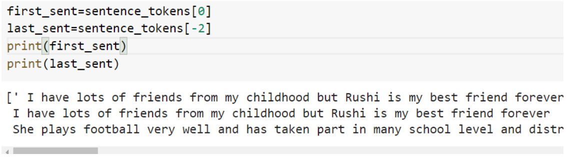
Normalized Frequency :



```
print(word_frequencies)
```

```
{'lots': 0.2, 'friends': 0.6, 'childhood': 0.2,
```

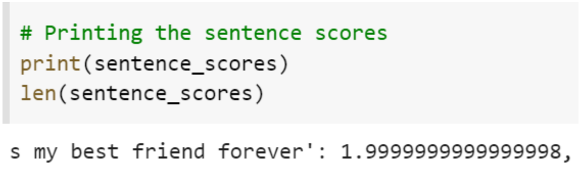
Sentence Tokenization :



```
first_sent=sentence_tokens[0]
last_sent=sentence_tokens[-2]
print(first_sent)
print(last_sent)
```

```
[' I have lots of friends from my childhood but Rushi is my best friend forever
 I have lots of friends from my childhood but Rushi is my best friend forever
 She plays football very well and has taken part in many school level and distr
```

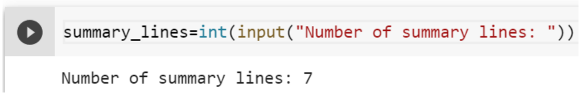
Calculating Sentence Score :



```
# Printing the sentence scores
print(sentence_scores)
len(sentence_scores)
```

```
s my best friend forever': 1.9999999999999998,
```

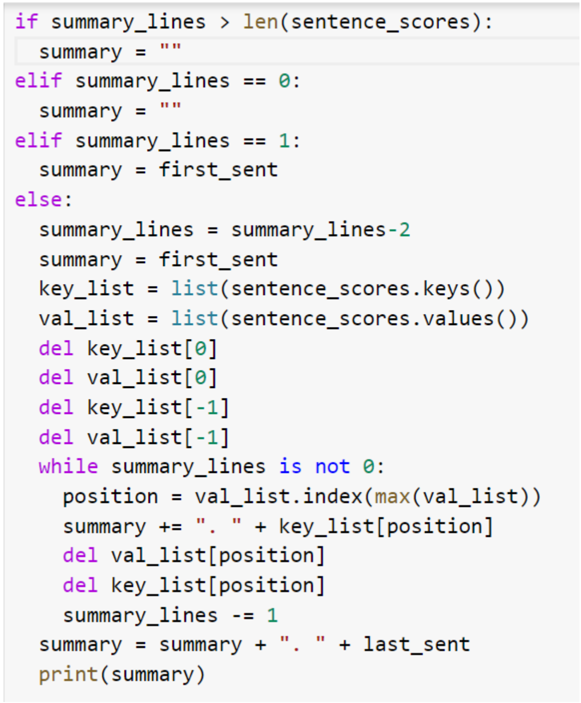
Number of Lines to be Summarized :



```
summary_lines=int(input("Number of summary lines: "))
```

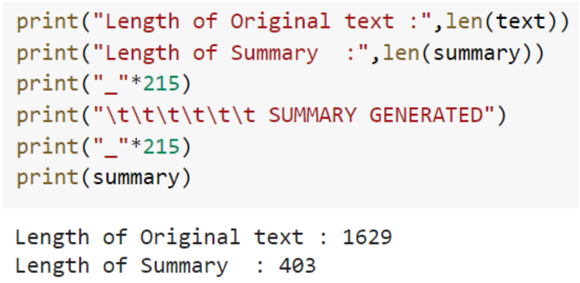
```
Number of summary lines: 7
```

Summarization :



```
if summary_lines > len(sentence_scores):
  summary = ""
elif summary_lines == 0:
  summary = ""
elif summary_lines == 1:
  summary = first_sent
else:
  summary_lines = summary_lines-2
  summary = first_sent
  key_list = list(sentence_scores.keys())
  val_list = list(sentence_scores.values())
  del key_list[0]
  del val_list[0]
  del key_list[-1]
  del val_list[-1]
  while summary_lines is not 0:
    position = val_list.index(max(val_list))
    summary += ". " + key_list[position]
    del val_list[position]
    del key_list[position]
    summary_lines -= 1
  summary = summary + ". " + last_sent
  print(summary)
```

Output :



```
print("Length of Original text :",len(text))
print("Length of Summary  :",len(summary))
print("_"*215)
print("\t\t\t\t\t\t\t SUMMARY GENERATED")
print("_"*215)
print(summary)
```

```
Length of Original text : 1629
Length of Summary  : 403
```

Conclusion and Future Work

Text Summarization Saves Time:

By generating automatic summaries, text summarization helps content editors save time and effort, which otherwise is invested in creating summaries of articles manually.

Text Summarization gives Instant Response:

It reduces the user's effort involved in extracting the relevant information. With automatic text summarization, the user can summarise an article in just a few seconds by using the software, thereby decreasing their reading time.

Text Summarization Increases Productivity Level:

Text Summarization enables the user to scan through the contents of a text for accurate, brief, and precise information. Therefore, the tool saves the user from the workload by reducing the size of the text and increases the productivity level of the user who can channel their energy to other critical things.

Text Summarization ensures that all Important Facts are Covered:
The human eye can miss crucial details; however, automatic software does not. What every reader requires is to be able to pick out what is beneficial to them from any piece of content. The automatic text summarization technique helps the user gather all the essential facts in a document with ease.

The field of text summarization is experiencing rapid growth, and specialized tools are being developed to tackle more focused summarization tasks. With open-source software and word embedding packages becoming widely available, users are stretching the use case of this technology.

Automatic text summarization is a tool that enables a quantum leap in human productivity by simplifying the sheer volume of information that humans interact with daily. This not only allows people to cut down on the reading necessary but also frees up time to read and understand otherwise overlooked written works. It is only a matter of time that such summarizers get integrated so well that they create summaries indistinguishable from those written by humans.

Future work:
We will implement Abstractive Summarisation and update code. We will also look to update the code.