Journal Homepage: - www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI: 10.21474/IJAR01/14876
DOI URL: http://dx.doi.org/10.21474/IJAR01/14876

RESEARCH ARTICLE

TEXT SUMMARIZATION USING PYTHON NLTK

Dr. Bharti Deshmukh
Assistant Professor, Prerna College of Commerce, Nagpur.

Manuscript Info

Manuscript History
Received: 10 April 2022
Final Accepted: 14 May 2022
Published: June 2022

Key words:-
Natural Processing Language (NLP), Natural Language Tookkit (NLTK), Extractive Summarization, Abstractive Summarization

Abstract

Text summarization is basically summarizing of the given paragraph with the use of natural language processing and machine learning. There has been an explosion in the quantity of textual content records from lot of sources. This quantity of textual content is a useful supply of facts and information which needs to be efficiently summarized to be useful. In this paper, the primary tactics to computerized textual content summarization were described. The distinctive approaches for summarization and the effectiveness and shortcomings of the distinctive methods were described. The machine works through assigning rankings to sentences withinside the document to be summarized, and the use of the maximum scoring sentences in the summary. Score values are primarily based totally on functions extracted from the sentence. A linear mixture of function rankings was used. Almost all the mappings from function to score and the coefficient values withinside the linear mixture were derived from a training corpus. Some anaphor decision was performed. In addition to primary summarization, a strive was made to address the issue of targeting the text at the user. The meant user was taken into consideration to have little history information or analyzing ability. The machine enabled through simplifying the individual words or phrases used in the summary and through drawing the pre-needful history facts from the web.

Introduction:-

1.1 Overview

In today's fast emerging world of facts and information text summarization is very vital and critical and required tool for understanding text information or textual content facts. There is a lot of textual content and file, documents available on the internet which provides information or facts past necessities and creates the scenario called "infobesity". To pick out information or facts from huge quantity of information or facts from different sources is difficult for human beings. Due to the large quantity of information and un-structuredness of facts, to manually summarize facts to be had at the net is really challenging, complex and tough task.

Before going to the Text summarization, first we, should realize or know that what a summary is. A summary is a textual content that is constructed from one or more texts, that conveys essential information or facts in the original text, and it is of a shorter form. The purpose of automated textual content summarization is providing the source

Corresponding Author:- Dr. Bharti Deshmukh
Address:- Prerna College of Commerce, Nagpur.

textual content right into a shorter model with semantics. The most important benefit of using a summary is, it reduces the reading time.

Problem Statement

In our busy schedule, it is very difficult for us to go through the entire article or document. So, we prefer to read summary. In this paper, summarization of large text into short summary is done which reduces the reading time for users. For implementation, Natural Language Toolkit (NLTK) through Natural Language Processing (NLP) algorithm is used. The prerequisite to use this Graphical User Interface (GUI) is that the user must have little knowledge about English language.

Objectives:-

The intention of textual content summarization is to lessen the size of the textual content even as keeping its crucial data and overall meaning. With the provision of internet, data or records are developing leaps and limit. It is far nearly not possible summarizing all this records manually. Automatic summarization may be classified as extractive and abstractive summarization. For abstractive summarization we should know the meaning of textual contents and then create a shorter model which nice expresses the meaning, while in extractive summarization we pick out sentences from given records itself which includes maximum information or data and fuse the ones sentences to create an extractive summary.

The goal of this paper is to apprehend the principles of natural language processing and developing a tool or device for textual content summarization. The difficulty in summarization is growing extensively, so the manual work is removed. The paper concentrates to develop a device which summarizes the document.

Contributions

The research has motivated on feature extraction for finding good combination of features for achieving good content coverage, minimum redundancy and possible cohesiveness. The thesis presented an approach where sentences are represented as a set of features. The features capture the statistical aspects. An extensive analysis of the feature sets was conducted to understand their impact on capturing information. The number of features were reduced to optimize the size of rule-base used as decision module.[1] With the reduction of features, the quality of summary generated is found to be much better than Baseline & MS-Word in terms of precision recall and F-measure. A good combination of features consisting of statistics of document like TF-ISF/IDF, sentence length, sentence position score, numerical data, sentence centrality, Thematic-words score and proper noun is used.[2] This thesis focuses on the text summarization system using python NLTK. This thesis has highlighted similarity measure of text documents to reduce the redundant sentences from summary.

Review Of Literature:-

Overview

Deepali K. Gaikwad et al have discussed various features of text summarization as follows:

1. Term Frequency
2. Location
3. Cue method
4. Sentence length

Approaches of various text summarization techniques have also been briefly described as follows:

1. Abstractive Approach:
   a. Structure based approach:
   Structured based approach encodes most important information from the document through cognitive schemes such as templates, extraction rules and other structures such as tree, ontology, lead and body phrase structure.
   b. Semantic based approach:
   In Semantic based approach, semantic representation of document is used to feed into natural language generation (NLG) system. This method focuses on identifying noun phrase and verb phrase by processing linguistic data.
2. Extractive Approach
   Performance comparison of various text summarizers for Indian languages have also been done. [3]

Pradeepika Verma et al have discussed about how text summarization can be done using various approaches. Challenges like Problem of redundancy, Problem of irrelevancy, Problem of loss of coverage, Problem of non-readability and less cohesive content that are faced while summarizing text have also been discussed.
Also, a good number of works related to extractive text summarization are discussed:
1. Graph based methods
2. Maximal Marginal Relevance based methods
3. Meta-heuristic based methods, Several Other methods have also been discussed.

In this paper, the authors have presented a technical background for document summarization. This paper has also discussed several challenges as well as surveys of the existing summarization methods. From these discussions, they have observed that many techniques suffer from various challenges, for example, the graph-based methods have imitation in data size, the clustering-based methods require prior knowledge of the number of clusters, the MMR approaches have uncertainty for the coverage and non-redundancy aspects in the summary, etc. So, it is imperative that further research is required in this field to develop more effective methods for document. [4]

Tripti Sharma et al have discussed past works that have been done in the field of text summarization. Some of them are mentioned below:
1. Title word method
2. Fixed phase feature
3. Paragraph method
4. Uppercase word feature

Description of python libraries have also been given.
a. Genism: Genism is a python library dependent on NumPy and SciPy libraries. Genism workflow have also been discussed.
b. NLTK: NLTK stands for Natural Language Toolkit. It was developed jointly with a computational linguistics course at the University of Pennsylvania in the year 2001 by Edward Loper and Steven Bird. It is the most popular python library used to work with human language data. It contains a variety of text processing libraries for classification, tokenization, stemming, tagging, parsing etc. It uses TF-IDF algorithm for text summarization.
NLTK involves follow steps:
1) Tokenize the sentence: it divides a text into a series of tokens.
2) Matrix of all words sentence by sentence is created called frequency matrix. It stores the number of times a word is appeared in each sentence.
3) Calculate Term Frequency and generate a matrix.
Sorting techniques and experimental results have also been discussed.[5]

Workdone
Text Summarization using Python NLTK has been focused here. Natural Language Toolkit (NLTK) is a text processing library that is widely used in Natural Language Processing (NLP). It supports the high-performance functions of tokenization, parsing, classification, etc.[6] The NLTK term initially released it in 2001.

Natural language processing (NLP) is a field of artificial intelligence in which computers analyze, understand, and derive meaning from human language in a smart and useful way.[6] By utilizing NLP, developers can organize and structure knowledge to perform tasks such as automatic summarization, translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation.

NLP is commonly used for summarization of texts text mining, machine translation, and automated question answering. In this project, we have used the frequency method approach of NLP algorithm.

There are two different approaches for text summarization:
1. Extractive Method
2. Abstractive method

Extractive method is focused more. This method functions by identifying important sentences or excerpts from the text and reproducing them as part of the summary. In this approach, no new text is generated, only the existing text is used in the summarization process. [7]

Implementation

Algorithm:

Step 1: The first step is to import the required libraries. There are two NLTK libraries that are necessary for building an efficient text summarizer.

1. From nltk.corpus import stopwords.
2. From nltk.tokenize import word_tokenize, sent_tokenize

Terms Used:

Corpus

A collection of text is known as Corpus. This includes both data sets such as bodies of work by an author, poems by a particular poet, etc. A data set of predetermined stop words is used here.

Tokenizers

This divides a text into a series of tokens. In Tokenizers, there are three main tokens – sentence, word, and regex tokenizer. Only the word and the sentence tokenizer are used here.

Step 2: Remove the Stop Words and store them in a separate array of words.

Stop Words

Words such as **is, an, a, the, for** that do not add value to the meaning of a sentence. For example, let us take a look at the following sentence:

Contribution of Subhas Chandra Bose to the Independence of India **is** significant.

After removing the stop words in the above sentence, we can narrow the number of words and preserve the meaning as follows:

[‘Contribution’, ‘Subhas’, ‘Chandra’, ‘Bose’, ‘Independence’, ‘India**’, ‘**significant**’,‘.’]**

Step 3: Create a frequency table of the words.

A Python Dictionary can keep a record of how many times each word will appear in the text after removing the stop words. We have used this dictionary over each sentence to know which sentences have the most relevant content in the overall text.

```
1. stopwords = set (stopwords.words(“english”))
2. words = word_tokenize(text)
3. freqTable = dict()
```

Step 4: Depending on the words it contains and the frequency table, we will assign a score to each sentence.

Here, we have used the **sent_tokenize()** method, that can be used to create the array of sentences. We will also need a dictionary to keep track of the score of each sentence, and we can later go through the dictionary to create a summary.

```
1. sentences = sent_tokenize(text)
2. sentenceValue = dict()
```

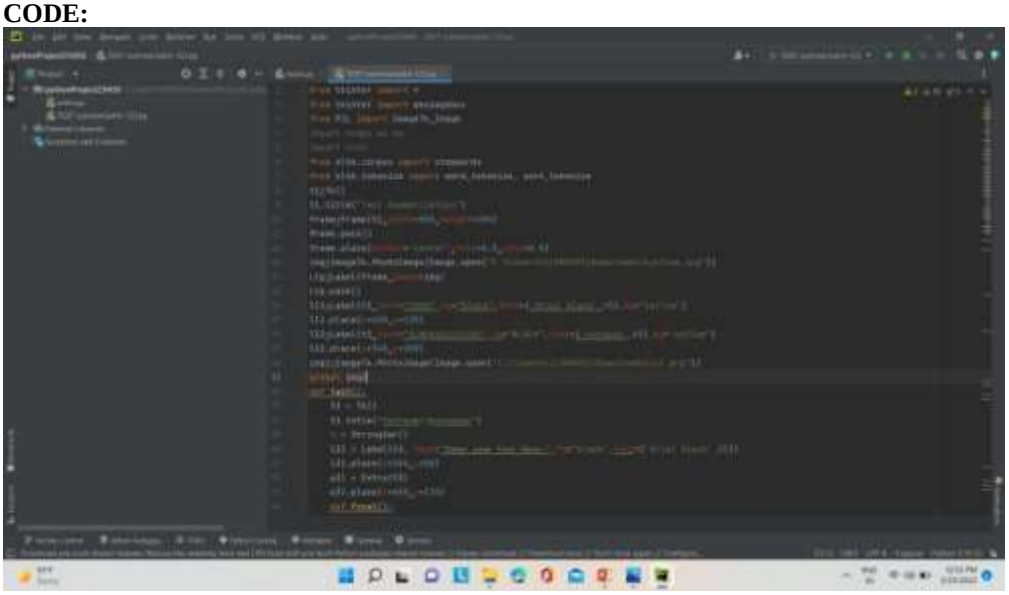
Step 5: To compare the sentences within the text, assign a score.

One simple approach that is used to compare the scores is to find an average score of a particular sentence. This average score can be a good threshold.

```
1. sumValues = 0
2. for sentence in sentenceValue:
3. sumValues += sentenceValue[sentence]
4. average = int(sumValues/len(sentenceValue))
```

Apply the threshold value and then store sentences in an order into the summary.[8]

CODE:



Testing

During this phase, the developed GUI was verified. All the functionalities that were causing errors or problems were removed as the final result of the system is a very high priority and important.

TOOLS

The tools / software used during the implementation of the GUI:

1. Pycharm & python 3.10
2. Modules:

a) Tinker message box
b) PIL
c) ImageTK
d) Numpy
e) NLTK
f) NLTK.corpus
g) NLTK.Tokenize
h) Stop words
i) Word Tokenize
j) Sent Tokenize

Results And Discussion:-

Results

Step 1: Click on **"START"**

Fig 4.1:-

Step 2: Enter your details & select the language of text to be summarized.

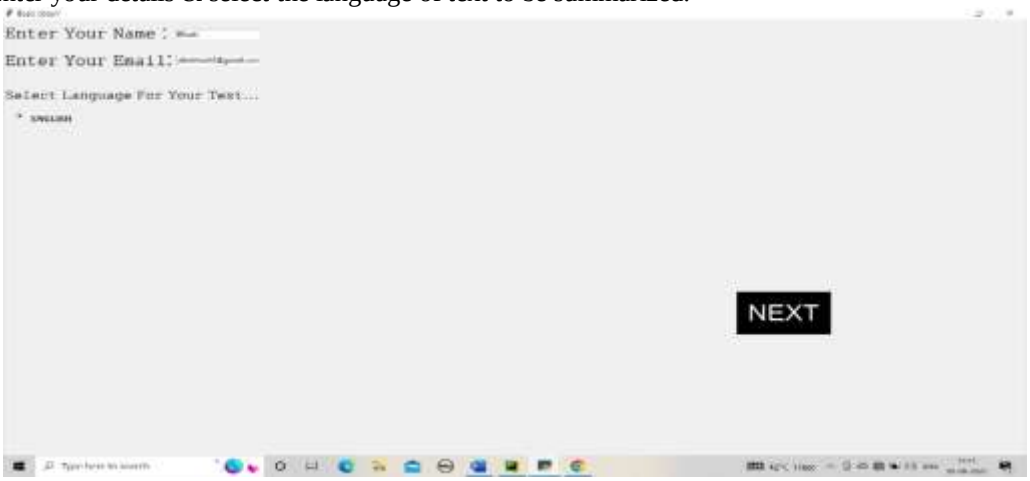


Fig 4.2:-

Step 3: Enter the text to be summarized.

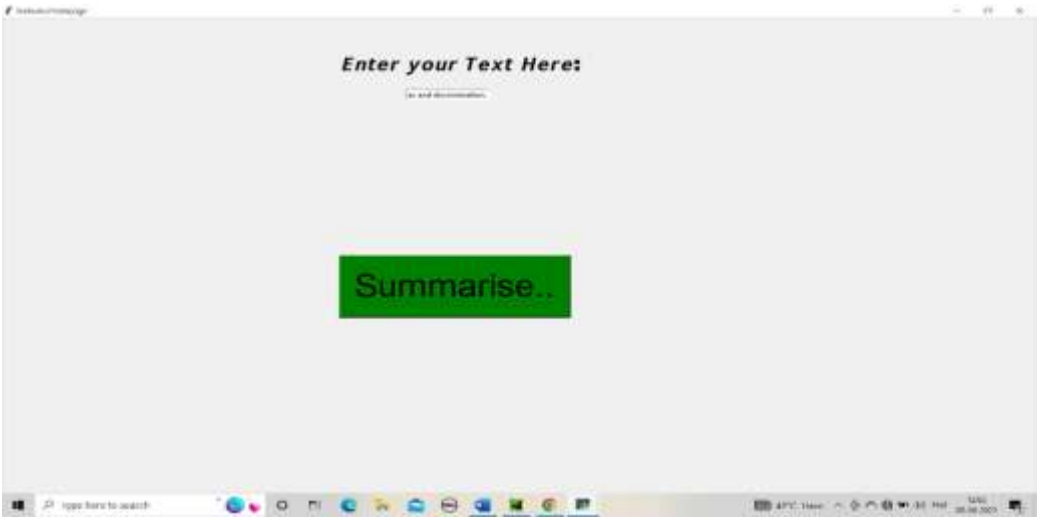


Fig 4.3:-

Step 4: Text will be summarized

Fig 4.4:-

Original Text:
Born and raised in a Hindu family in coastal Gujarat, Gandhi trained in the law at the Inner Temple, London, and was called to the bar at age 22 in June 1891. After two uncertain years in India, where he was unable to start a successful law practice, he moved to South Africa in 1893 to represent an Indian merchant in a lawsuit. He went on to live in South Africa for 21 years. It was here that Gandhi raised a family and first employed nonviolent resistance in a campaign for civil rights. In 1915, aged 45, he returned to India and soon set about organising peasants, farmers, and urban labourers to protest against excessive land-tax and discrimination.[9]

Summarized Text:
Born and raised in a Hindu family in coastal Gujrat, Gandhi trained in the law at the Inner Temple, London, and was called to the bar at age 22 in June 1891. After two uncertain years in India, where he was unable to start a successful law practice, he moved to South Africa in 1893 to represent an Indian merchant in a lawsuit.

Discussion:-
As per the research, it is quite evident that extractive based summarizing implementations have resulted more effectively than abstractive based. However, even though the implementations within the bounds of the domains to which the studies have been restricted have been successful, they are still not as accurate as would be expected to a normal user of that system. As far as the research on abstractive summarization is considered, successful implementations are a rarity, though the research conducted on it, at least theoretically, proves that if a successful implementation is attained, the summary generated will make more sense than the summary from an extraction-based summary.

Summary And Conclusions:-
Summary
To summarize with, in this paper GUI is developed using python NLTK that summarizes texts. Also, the concept of extractive approach and the implementation of NLP algorithm using Frequency method approach is made clear. Various researches which have been done in the field of text summarization have also helped to write this paper.

Conclusion:-
Text summarization is increasingly recognized as sub-branch of NLP. Due to large quantity of textual content records or information available on net, the demand for compressed but meaningful information has increased. Thus, text summarization is today's need and used by almost all. Text summarization is used for both commercial purpose as well as research purpose. An abstractive summarization requires deep learning and reasoning. A technical background for document summarization is presented in this paper. A basic text summarizer using nltk library using python is used and it is proved that it works on small documents. Frequency method approach of NLP algorithm has successfully worked and has helped to reduce the redundant sentences from summary.

References:-
[1] Ayesha Yaseen, "CONCLUSION AND FUTURE SCOPE" , https://www.academia.edu/40614929/CONCLUSION_AND_FUTURE_SCOPE
[2] By Richard Nordquist, "Sentence Length", May 23, 2018, https://www.thoughtco.com/sentence-length-grammar-and-composition-1691948
[3] Deepali K. Gaikwad and C. Namrata, "A Review Paper on Text Summarization", International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 5, Issue 4, April 2018
[4] Pradeepika Verma and Anshul Verma, "A Review on Text Summarization Techniques", Volume 64, Issue 1, 2020 Journal of Scientific Research
[5] Tripti Sharma, Anmol Ashri, Navin Kumar, Shubham Pal, Rajat, "Evaluation of Python Text Summarization Libraries", International Journal of Engineering Development and Research (IJEDR), ISSN:2321-9939, Volume.9, Issue 1, pp.159-164, January 2021
[6] Kapil Khangaonkar, "Introduction to Natural Language Processing (NLP)" 2016, https://www.linkedin.com/pulse/introduction-natural-language-processing-nlp-2016-kapil-khangaonkar
[7] DARSHANKUMAR VANOL, "YOUTUBE TRANSCRIPT SUMMARIZER", HTTPS://MEDIUM.COM/@DARSHANVANO009/YOUTUBE-TRANSCRIPT-SUMMARIZER-4AD5618C8C25

[8] Nitin Kumar,Apr 15, 2021, “Text Summarization in Python”, https://www.mygreatlearning.com/blog/text-summarization-in-python/
[9] https://en.wikipedia.org/wiki/Mahatma_Gandhi.