

# Text Summarization with Sentimental Analysis

Kummari Shiva Kumar<sup>1</sup>, M Priyanka<sup>2</sup>, M Rishitha<sup>3</sup>, D Divya Teja<sup>4</sup>, and Nallamothu Madhuri<sup>5</sup>

<sup>1,2,3,4</sup> Students, Department of Computer Science & Engineering, Dhanekula Institute of Engineering and Technology, Vijayawada, Andhra Pradesh, India

<sup>5</sup> Assistant professor, Department of Computer Science & Engineering, Dhanekula Institute of Engineering & Technology/JNTUK/, Vijayawada, Andhra Pradesh, India

Correspondence should be addressed to K. Shiva Kumar: shivakumar98669@gmail.com

Copyright © 2021 Made K Shiva Kumar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**ABSTRACT-** In today's world, Modern organizations deal with terabytes of text, such as email, that often plays a significant role in their day to day operations. The user has to face a task of identifying useful information from these data which is difficult and it requires some amount of time. One possible means is to use text summarization. Text summarization is the process of identifying the most valued/meaningful information in a document and compressing that information into a shorter version preserving its overall meaning. Sentiment analysis is about determining the text given by the user whether it is Positive, Negative or Neutral. We used Gensim Algorithm for generating text summary. This algorithm automatically summarizes the given text, by extracting one or more important sentences from the text. This project is about text summarization which includes sentiment analysis. In UI, a text box will be displayed, which is used to take the input text from the user which need to be summarize. Then it will pre-process the text and show the summarized content. It will be taking input as URL of an article and it is going to provide Title, Author, Publication Date, Sentimental analysis, Keywords, URL of an Article.

**KEYWORDS -** Text Summarization, Natural Language Processing, Sentimental Analysis.

## I. INTRODUCTION

Before getting into detail about the Text summarization, first we have to know about what a summary is? A summary is a text that is produced from one or more texts, that conveys important information in the original text, and it is of a shorter form. The goal of automatic text summarization is presenting the source text into a shorter version with semantics. The most important advantage of using summary is, it saves the time for the users. Text Summarization has two methods namely extractive and abstractive summarization. An extractive summarization method focuses on the sentence present in the original document itself and concatenating them into shorter form. An Abstractive summarization tries to give summarized output based on the trained data set. [4]

## II. LITERATURE SURVEY

Our project focuses on providing a system design which could prepare a summary for the user by scanning through

multiple articles. If a reader wants to get an information in detail, he/she has to go through multiple sources such as blogs, articles, etc. which is very time consuming and needs a lot of patience. So we are developing a system to come up with summary for the user. We will apply the Text Rank algorithm on a dataset of scraped articles with the aim of creating a nice and concise summary [2][3]. Also, that this would be essentially a single domain multiple documents summarization task, i.e., we will take multiple articles as input and generate a single bullet point summary. The first step is to concatenate all the text data in the document. Then split the text into individual sentences. Then do pre-processing on them. In the next step, we will find vector representation (word embedding's) for each and every sentence. Similarities between sentence vectors are then calculated and stored in a matrix. The similarity matrix is then converted into a graph, with sentences as vertices and similarity scores as edges, for sentence rank calculation. [1]

## III. EXISTING SYSTEM

There are many text summarization applications in the web, one such application is INSHORTS. It provides its users the summarized content of the news articles. As it is confined to news articles, it may not be useful for the users who searches for a text summarizer. Moreover the existing system fails to provide the sentiment analysis to the news articles. Therefore, these two drawbacks from the existing system helped us develop an idea of providing users text summarization with sentiment analysis. [5]

### Disadvantages:

- The existing system is confined to only news articles.
- Sentiment analysis is absent in the existing system.

## IV. PROPOSED SYSTEM

Reading the large volumes of text data available online is challenging for the users as it is time taking process. The proposed system, Text Summarization is much more practical and applicable in real time. We are going to design a system in two ways: UI, GUI. **While working with UI**, the input is taken as text. The input text is processed using natural language processing and the input text is tokenized and then normalization is done by removing the stop words After that all the words in the text data will be converted into lowercase. Now, each word weight is calculated with

help of word frequency. Threshold should be 80% of the highest valued sentence. Percentage can be changed in the UI. While working with the GUI, a text summarization graphical user interface is created with the help of tkinter, gensim and nltk the input should be provided in the form of URL of any news article. From the given URL, the GUI will extract the text data and summarizes it. To the summarized text, the sentiment analysis is applied which helps the user to be more informative about the news. And it's going to provide the following: Title, Author, Publication Date, Sentimental analysis, Keywords, URL of an Article. We got this idea from the existing systems, but we observed a drawback of not having any sentiment analysis. Therefore, we felt like proposing a system which include the sentiment analysis to the summarized text. So that user can determine the news as positive, negative or neutral. In the below Figure 1, we can easily understand how the flow of text summarization for a text data is done.

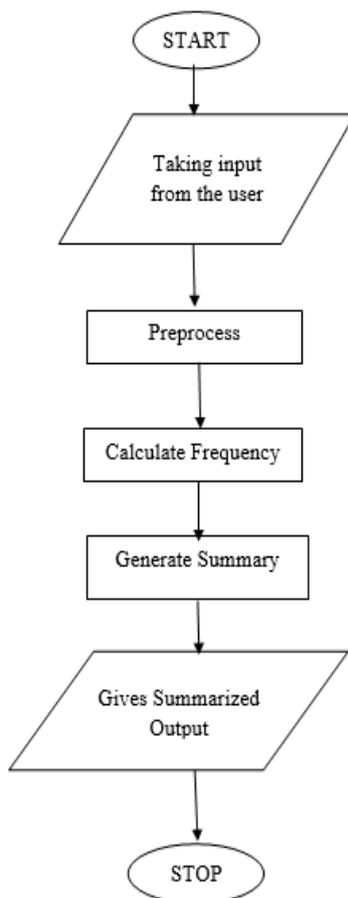


Fig. 1: Flowchart diagram of Text Summarization

### A. Gensim Algorithm

We used Gensim Algorithm for generating text summary. This algorithm automatically summarizes the given text, by extracting one or more important sentences from the text. In order to obtain the text summarization we used Gensim algorithm which is a free open-source Python library for representing documents as semantic vectors, as efficiently as possible. Gensim is designed to process raw, unstructured digital texts (“plain text”) using unsupervised machine learning algorithms. The algorithms which are

present in Gensim, such as Word2Vec, FastText, Latent Semantic Indexing etc., automatically discover the semantic structure of documents by examining statistical co-occurrence patterns within a corpus of training documents. These algorithms are unsupervised, that means no human input is necessary and we only need a corpus of plain text documents.

- **Corpus:** A collection of text documents.
- **Vector:** Form of representing text.
- **Model:** Algorithm used to generate representation of data.
- **Topic Modelling:** It is an information mining tool which is used to extract semantic topics from documents.
- **Topic:** A repeating group of words frequently occurring together.

As Gensim is efficient in topic modelling, we chose to include gensim in our project.

## V. IMPLEMENTATION

Steps involved in Extractive Summarization:

### A. Implementing Using UI

- Convert paragraph to sentences
- Text preprocessing
- Tokenization
- Evaluate the weighted frequency of words
- Substitute words with their weighted frequencies
- Based up on the maximum word frequency the generated words are sent to the gensim algorithm to get Topic modelled over the generated words

### B. Implementing using GUI

In this implementation we will be building a text summarization graphical user interface with tkinter, gensim and nltk. It will be taking URL as input of any news article and it's going to provide below mentioned format

- Title
- Author
- Publication Date
- Sentimental analysis
- URL of an Article

From the given URL, the GUI will extract the text data and summarizes it. To the summarized text, the sentiment analysis is applied which helps the user to be more informative about the news.

## VI. DESIGN

### A. UI Implementation

We developed an interface for the user to give the text data in order to summarize. The user is allowed to place the text data in the text box given and he/she need to click on the predict button to get the summarized text.

### B. GUI Implementation

We developed a Graphical User Interface in order to take the URL as input from the user.

This is the Graphical User interface that asks the user for the URL of any news article that needs to be summarized.

## VII. RESULTS/OUTPUT

### A. UI Output

After clicking the predict button, a result page is displayed as above figure which shows the summary of the input text

to the user. The below Figure 2 is taken when we limit the ratio of the summary to 20%.

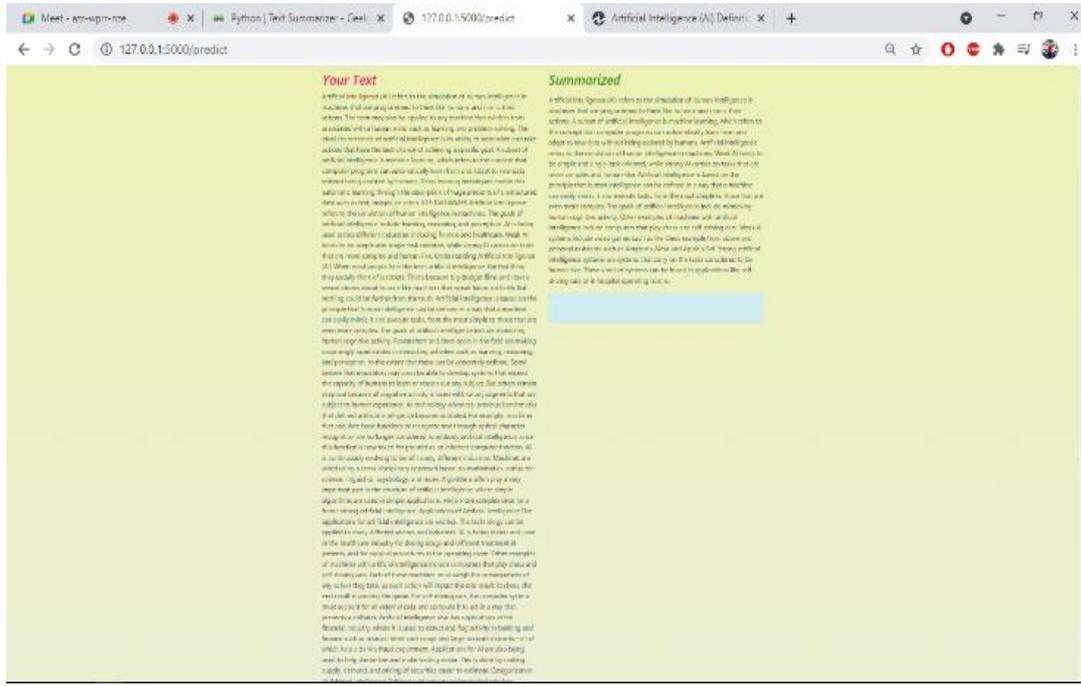


Fig. 2: UI Output page

### B. GUI Output

This below figure 3 is taken when the users provide an URL from India Today Website:

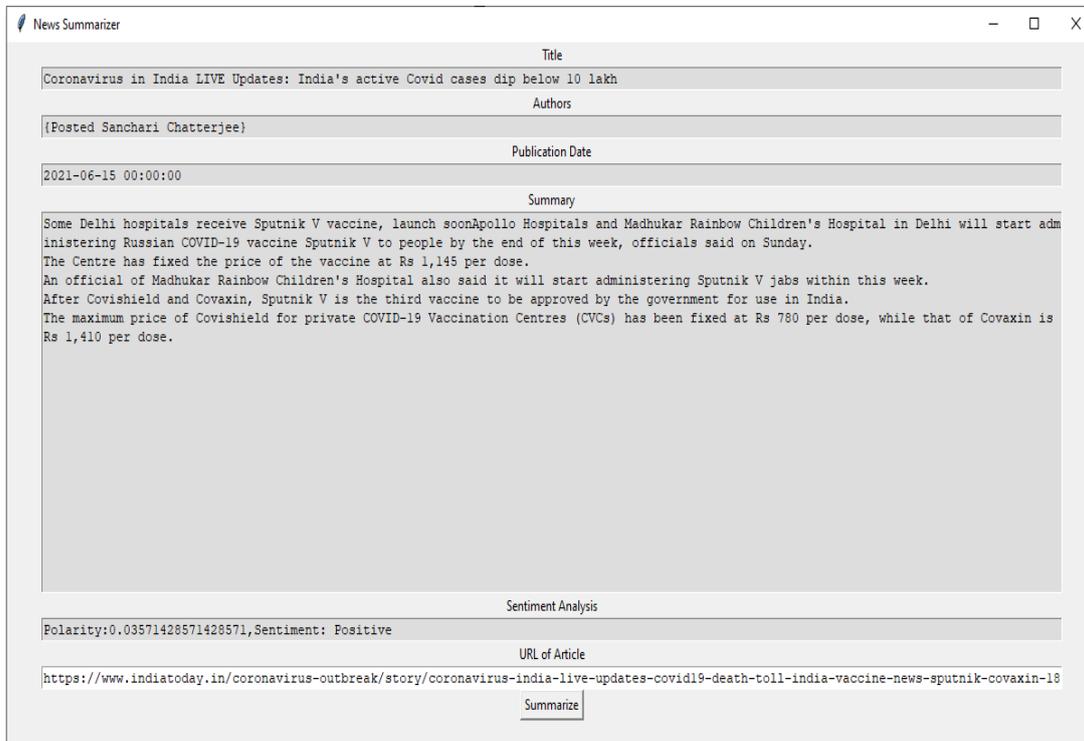


Fig. 3: GUI Output page

## VIII. FUTURE SCOPE

In present busy lifestyle of humans, they got used to read everything in summarized (concise) form. So according to their lifestyle, we developed an application that saves their time that is allotted for reading news articles as a daily routine. As a future scope, the images in the news articles may also be considered, which helps the user to be more informative about the news article. Where as in UI implementation, we can also work on abstractive summarization that provides meaningful summary, which includes the knowledge of the trained system.

## IX. CONCLUSION

The main objective of our project is to create summaries for the huge text data which results in time saving and preserving its overall meaning. It helps in concise understanding of the textual data. Our project also handles the limit of the summary by giving the choice to the user in terms of ratio/word count. The GUI implementation of our project is confined only to news articles. The objective of the GUI is to provide the summary of the news article along with the specific details like title, author, published date and sentiment analysis. Therefore, our project is more effective compared to other news summarizers as it has sentiment analysis.

## CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

## REFERENCES

- [1] Dazhi Yang\_ and Allan N. Zhang Singapore Institute of Manufacturing Technology "Title of the paper performing literature review using text mining, Part III: Summarizing articles using Text Rank".
- [2] Wengen Li and Jiabao Zhao School of management and engineering title "Text Rank algorithm by exploiting Wikipedia for short text keywords extraction."
- [3] Sonya Rapinta Manalu, Willy School of Computer Science title "Stop Words in Review Summarization Using Text Rank".
- [4] BlogVidhyaAnalytics:<https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/>  
AzharIqbal.,<https://www.inshorts.com/en/news/indianorigin-vandi-verma-drives-rover-on-mars-from-nasas-lab-calls-jezero-incredible-1625275849898>.

## ABOUT THE AUTHORS



**Kummari Shiva Kumar**, A Under Graduate Final Year student seeking his degree in Computer Science Engineering at Dhanekula Institute of engineering and Technology, Vijayawada, Andhra Pradesh.



**Matta Priyanka**, A Under Graduate Final Year student seeking her degree in Computer Science Engineering at Dhanekula Institute of engineering and Technology, Vijayawada, Andhra Pradesh.



**Marre Rishitha**, A Under Graduate Final Year student seeking her degree in Computer Science Engineering at Dhanekula Institute of engineering and Technology, Vijayawada, Andhra Pradesh.



**Dunna Divya Teja**, A Under Graduate Final Year student seeking her degree in Computer Science Engineering at Dhanekula Institute of engineering and Technology, Vijayawada, Andhra Pradesh.



**Nallamothe Madhuri**, Assistant Professor in Computer Science & Engineering department at Dhanekula Institute of engineering and Technology, Vijayawada, Andhra Pradesh.