Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020)
IEEE Xplore Part Number: CFP20V66-ART; ISBN: 978-1-7281-4108-4

# Automatic Text Summarization and Keyword Extraction using Natural Language Processing

Avinash Payak
*Department of Information Technology*
*RAIT, Mumbai University*
Navi Mumbai, India
avinashpayak@gmail.com

Saurabh Rai
*Department of Information Technology*
*RAIT, Mumbai University*
Navi Mumbai, India
saurabh.jeevan9@gmail.com

Kanishka Shrivastava
*Department of Information Technology*
*RAIT, Mumbai University*
Navi Mumbai, India
kanishkashrivastava786@gmail.com

Reshma Gulwani
*Department of Information Technology*
*RAIT, Mumbai University*
Navi Mumbai, India
reshma.gulwani@rait.ac.in

***Abstract*—The process of gaining and absorbing the knowledge from various sources is a time-consuming process where people, mainly youth spend time surfing over the internet for relevant information. The proposed system mainly focuses on scraping the data from websites and providing the summary as well as keywords from the information extracted from various websites giving the user flexibility to select the website of their choice. The proposed system for the text summarization and keyword extraction undergoes a sequence of steps starting from data extraction from a website link, removal of outliers and irrelevant information, emphasizing on the importance of particular data extracted from the website and creating a summary of the extracted data. For the selection of relevant information from the extracted data, it is necessary to use natural language processing. The proposed project helps its users to reduce their surfing time and gives summary prepared from multiple website links and documents or keywords from a particular website or a document.**

***Index Terms*—multi-webpage summarization, multi-document summarization, keyword extraction, natural language processing, Text rank algorithm***

## I. INTRODUCTION

Summarization of any data plays a vital role in integrating central ideas in a meaningful way and to ignore irrelevant information. Keywords drawn out of the summary helps in un-declining the main idea of the document. It saves adequate time for different domains of use cases like marketing, institutions, education, business etc. Data mining involves the process of generating new data by evaluating already existing large data sets. Classification, clustering, regression, association, outlier detection, prediction, tracking sequential patterns are the techniques used for data mining. The raw data is converted into useful information using these techniques.

Web mining is the procedure of one of the data mining techniques which emphasize on the World Wide Web and its components as the primary source of data. It discovers patterns and evokes valid information from documents. It is used to find a pattern in web pages and web documents by collecting and analyzing information to gain insight into the overall data. It aims to extract/mine useful information or knowledge from the web page content. Keyword extraction, being the most important, is a process of highlighting important words, phrases and expressions in a particular content. It is done using Natural Language Processing (NLP).

## II. RELATED WORK

J.N Madhuri et al. proposed a text summarization technique [1] based on a ranking algorithm which gives the important sentences which are collected to form an audio summary. There are two techniques for doing the text summarization:

1) Abstractive
2) Extractive

This is done by using extractive text summarization which helps in important sentence selection using the linguistic or statistical features. Prakhar Sethi et.al. proposed a summarization algorithm[2] which uses lexical chains and thesaurus to generate a text summary of the news. It compared various summaries of the content and hence the scoring parameters. The sentence is given the score based on repeated nouns in the article. Thus, the sentence with the maximum score is considered in the summary of the article. Hua Yuan et.al. proposed a summarization method for tourism blogs[3] which focuses on removing the noise efficiently and was tested over a Chinese tourism blog. Yutong Wu et.al. Proposed a method with semantic and context-based analysis for multi-document summarization [4] which measures the important information covered in a sentence. Yan-Xiang He et.al. Proposed a method for multi-document summarization[5] which tries to reduce the sentence size and combines similar sentences to create new sentences. Dragomir Radev et.al. proposed a centroid- based summarization method[6] for multiple documents which finds the most important words from the documents and then selects the sentences that largely represent the context of the document. N. Moratanch, S. Chitrakala et.al. surveyed various methods for text summarization [7]. They chose extractive

summarization over the other processes because it has high precision and coherence along with having low cohesion. Arpita Sahoo et.al. proposed a review paper on extractive text summarization[8] which gives the idea of each document in a very short paragraph to reduce the time consumed by the reader while going through various documents. S. Mohamed Saleem et.al. studied text summarization using extractive methods[9], in which he focuses on the importance of extractive methods in summarization for better and efficient performance K. Vanisri et.al. proposed an integrated approach to web document summarization using semantic similarity[10], which mainly focuses on the ranking of clusters using the K-means clustering algorithm and enhances the quality of the obtained summary. Shai Erera et.al. proposed a summarization system for scientific documents[11] which uses IBM Science Summarizer that allows the user to get the summaries of scientific documents. N. Vijay Kumar et.al. performed a survey on real-time accumulative short text summarization on comment streams[12], which uses an algorithm IncreSTS, which can incrementally update the clustering results to provide an effective summary. Thus these papers help in understanding how summarization of single and multiple documents, blogs and single website page is done and how noise is removed from sentences.

Papis Wongchaisuwat et. al proposed a paper an algorithm for keyword extraction [13] based on Textrank algorithm which gives a better-averaged precision and F1-score as compared to Textrank algorithm. In this paper, vectors containing words are used to calculate a similarity measure. This similarity measure is then used as the weight of the edge. The scores of sentences are calculated using the text rank algorithm. The sentences with the maximum score are further considered for the extraction of useful keywords. Bhavneet Kaur et al. proposes a method for catchphrase extraction[14] using natural language processing that enhances the f-score. The main technique used is web content mining which used to find a pattern in web pages and web documents by collecting and analysing information to gain insight into the overall data. The main aim is to maximise accuracy and precision. It also focuses on increasing the number of useful instances amongst all the other present instances. It involves the connection of web content mining with machine learning. Akshi Kumar et al. gives an analysis of three different keyword extraction algorithms [15] namely, lex-rank, latent semantic analysis and text-rank used in articles and compared their results with handwritten summaries. It uses ROUGE-1 to score the keywords and hence the keywords with the maximum score are considered for future analysis. The algorithm with the most efficiency is identified and hence is considered for the keyword extraction. Once the algorithm is identified, steps are taken to improve the accuracy of the algorithm. These documents mainly focus on how keywords are extracted from documents and how to improve the f-score.

As compared to the above-mentioned papers regarding their methods and implementation which mainly focus on providing summary from a single document or a single website link, our proposed system emphasizes on providing:

1) Summarization of not only a single website link but multiple links through Textrank algorithm. User may input single or multiple website links and the proposed system visits the link(s) and scrapes all the textual data from the website(s).
2) Summarization of not only single but multiple documents(pdf file or word file) through Textrank algorithm. User may upload single or multiple documents which are temporarily stored until the proposed system scrapes the data from the documents.
3) Keyword extraction is also implemented with the help of Textrank algorithm. User may input a website link or upload a document and keywords will be extracted. A total number of keywords extracted depends upon the size of the document.
4) The proposed system also gives the flexibility to the user to input direct text in the text area through which keywords can be extracted or text can be summarized based on the user's choice.

## III. Proposed Methodology

### A. Problem Definition

The proposed system should be designed in such a manner that it will alleviate these limitations. The problem definition of our proposed system is as follows:

- To extract textual data from any link chosen by the user and display its summary. Websites might contain irrelevant data and information which is not important. This project emphasizes on removing outliers to display a summary of only important data and also to remove redundant information gathered from multiple websites.
- To summarize textual data from multiple web pages at once. Existing summarizing software, browser extensions can either summarize multiple documents at once or a single web page. This project mainly focuses on summarizing multiple web pages and documents.
- To let the user decide if a summary of one link or multiple links is required and to give the choice to produce either a combined summary or a separate summary of the chosen website links.
- To implement keyword extraction as functionality to help the user to know the context of the textual information acquired from the link or document quickly.
- To implement multi-document summarization to generate summary out of documents present in different formats such as pdf and word.

### B. System Architecture

The proposed system is built using the Flask framework and MySQL database at the back-end. Figure 1 shows the working diagram of the system for summarization. When the user enters a website link or uploads a document for summarization, the application starts the process and scrapes all the textual data from the link(s) or the document(s) using BeautifulSoup text mining python module. Cleaning step

involves removal of outliers and stop-words which is done using NLTK(Natural Language Toolkit) package available for python. Tokenization of sentences involves each word to get stored in an array where sentence acts as a node and each node has a weight assigned to it. Each sentence is a node for Textrank algorithm. Weight calculation is same as implemented in keyword extraction and is debriefed in implementation. HeapQ algorithm used in natural language processing retrieves the thirty percent that is three-tenth of the total number of sentences of the extracted text. This forms the summary of the extracted textual data. User has the option to choose whether the required summary should be a combined summary of all the documents or links given as an input or a separate summary of each input link or document is required by the user. In the case of combined summary, textual data from each link or document is combined first and it is summarized.

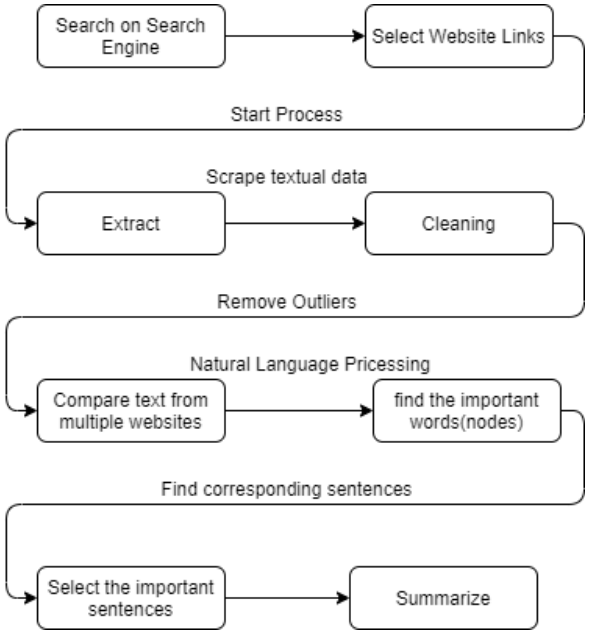


Fig. 1. Working diagram of the proposed system of summarization

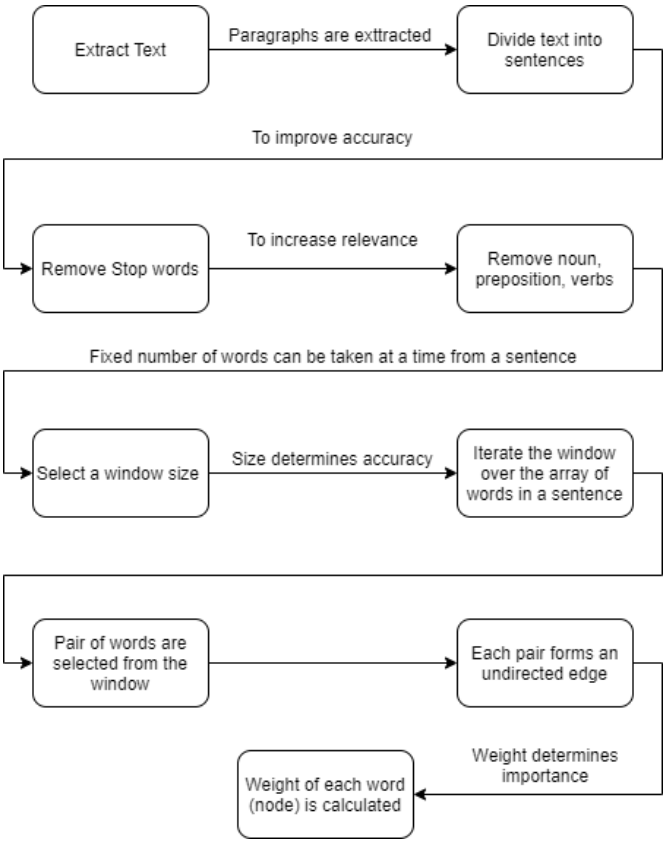


Fig. 2. Working diagram of the proposed system of keyword extraction

Keyword extraction is based on Textrank algorithm and begins with dividing the text into sentences. These sentences are divided into words. Stopwords are removed from the words and then these words are assigned specific Part-Of-Speech tags. SpaCy is used for generating these tags. Words of the sentences in pair form edge of the directed graph and Textrank graph is plotted. This is explained in the implementation of keyword extraction in detail.

### *C. System Design*

The proposed system is a website consisting of Front-End with which the user interacts and a Back-End which processes all the textual information depending upon the request of the user.

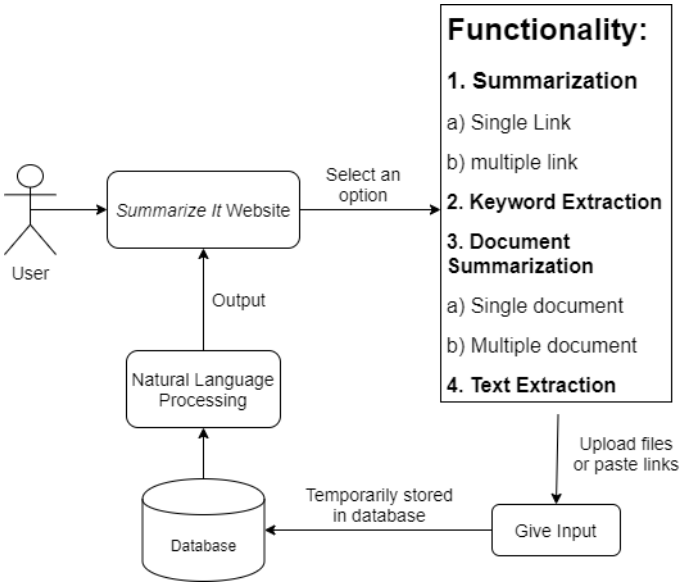


Fig. 3. Use case diagram

Figure 3 shows the use case diagram of the proposed system and the components of the proposed system. The user visits the website and selects one of the many features provided by the website. The input taken from the user is stored temporarily in the database until all the processes are performed and the final output is displayed on the website.

*D. Implementation*

The complete project is divided into small tasks and all the components related to the project are given below:

1) **Extraction**
   - **Website Text Extraction:** For summarization and extraction of keywords it is necessary to extract the data first from the website websites. The websites links are temporarily stored as a buffer which this then provided to the system one at a time to extract.
   - **Pdf and Word Data Text Extraction:** Data can also be extracted from different sources like pdf and word files which require uploading and temporarily storing the file for extraction.

2) **Summarization** Text summarization is creating a summarized data from the given text which has high relevance. Therefore it helps to reduce the user's effort to read and summarize a given text which could take hours differing from person to person. Summarization is done for creating a smaller version of the text which gives an overview of what the overall content conveys and saves time to read, understand and then infer from the complete text.

   Natural Language Processing (NLP) is used for the text summarization. For text summarization using NLP, Natural Language Toolkit (NLTK) library is used. NLTK module is used for Natural Language Processing(NLP). NLP is a process of getting a computer to understand natural language and usually this in the form of written language and sometimes it can be in the form of spoken language. But usually spoken language gets converted to written language and then to numbers. Following are the steps are taken for text summarization:

   a) **Splitting paragraphs into sentences:** When a paragraph is split into sentences, do it based on encountering a full stop. This is very important for further data cleaning process.
   b) **Cleaning the text:** After splitting paragraphs into sentences, the text needs to be cleaned. For cleaning the data, all the special characters, numerals, stopwords need to be deleted.
   c) **Tokenization:** Tokenizing means breaking the sentences into words. These words are later retrieved in an array separated by commas.
   d) **Counting frequency:** After tokenizing the sentences into tokens, the next important step is to find the frequency of the words. Frequency is the number of times the word has occurred in the particular text. After finding the frequency, weighted frequency of the word is calculated. Weighted frequency is the ratio of the frequency of the word and the frequency of the word which has occurred the most number of times. The weighted frequency of the stopwords will be zero.
   e) **Renewing the words in the sentences by the weighted frequency:** After calculating the weighted frequency, the words in the sentences are renewed with the weighted frequency. After this process, the sum of all the weighted frequencies of the words in the sentence is calculated. The sums hence obtained are stored in the array.
   f) **Reverse sorting:** The sums obtained in the above process are then sorted in the reverse order of their corresponding value. That is, look for the sentences which have the highest sum. If two sentences have the same calculated sum, the sentence which appears first in the text is considered first. After selecting the sentence with the highest sum, the sentence with the second-highest sum is then attached to the first sentenced to make the summary more relevant.

Above mentioned two algorithms are explained in brief:

a) **Textrank:** Textrank algorithm is a specific application of the Pagerank algorithm. The first step of the algorithm is to clean up the text. This is because the text is often noisy and full of irregularities. One can apply character filters to remove all the stopwords. Once the text is cleaned, the next step is to break that text up into tokens or possible keywords. Words are broken up into tokens based on whitespaces and punctuations. Once the words are broken into tokens, the next step is to filter out some of those tokens which would not make good keywords. For this purpose, part-of-speech tagging is used. In part-of-speech tagging, sentences are split into words. These words are categorised as noun, pronoun, adverb, verb. This is followed by stopwords removal, which is done by stopword filter which removes the common stopwords. The minimum length token filter removes the words that have two characters or less. Once the text is filtered, a graph is constructed. The graph is made based on the dependency between the two words when they co-occur. It consists of nodes which are considered as tokens.

b) **HeapQueue:** A heap is a semi-ordered tree-based data structure where either:

   i) Each parent's key is greater than its children(a max heap-largest element on top).
   ii) On each parent's key is less than its children(a min heap-smallest element on top).

   Often these trees have a max number of children(per parent) of 2, in which case they are known as binary heaps. The property of this data structure is that each time the smallest of the heap element is popped and whichever element is pushed orpopped, the heap structure is maintained. Heap push function is used to push the element into the heap. Heap pop function is used to remove and return the smallest element from the heap. The order is adjusted so as the heap structure is

maintained. To convert the list into a heap, heapify function is used.

3) **Keyword Extraction:** Summarization and keyword extraction both make use of text rank algorithm. Implementation of keyword extraction through Textrank algorithm is shown through the following example. Example:

a) **Step 1:** Text is taken as an input. Text = "*Coronathe virus is a deadly disease which is spreading at an alarming rate everywhere around the globe. The patients diagnosed with Corona-virus do not always show heavy symptoms. Sometimes, mild symptoms like coughing, sneezing, difficulty in breathing can also be a symptom of this virus. It can easily spread through the open contact with eyes, nose and mouth. As prevention is better than cure, it is better to prevent the spread of this virus by simply washing hands thoroughly for about 20 seconds with soap and water. One should also cover their mouths while sneezing so as not to let others get acquainted with the virus.*"

*b)* **Step 2:** This text is then divided into five sentences.
Sentence 1: *"Corona-virus is a deadly disease which is spreading at an alarming rate everywhere around the globe."*
Sentence 2: *"The patients diagnosed with Coronavirus do not always show heavy symptoms."*
Sentence 3: *"Sometimes, mild symptoms like coughing, sneezing, difficulty in breathing can also be a symptom of this virus."*
Sentence 4: *"It can easily spread through the open contact with eyes, nose and mouth."*
Sentence 5: *"As prevention is better than cure, its better to prevent the spread of this virus by simply washing hands thoroughly for about 20 seconds with soap and water."*
Sentence 6: *"One should also cover their mouths while sneezing so as not to let others get acquainted with the virus."*

c) **Step 3:** These sentences consists of words which do not have high relevance and are supposed to be removed. So, the important parts of speech are taken into account which includes Noun, Preposition, etc.
Output: [[Corona-virus, deadly, disease, spreading, alarming, rate, everywhere, globe], [patients, diagnosed, Corona-virus, always, show, heavy, symptoms], [mild, symptoms, coughing, sneezing, breathing, virus], [spread, through, open, contact, eyes, nose, mouth], [prevention, cure, better, prevent, spread, virus, washing, hands, thoroughly, soap, water], [cover, mouths, sneezing, acquainted, virus]]

d) **Step 4:** Every word is considered as a node. A window size can be fixed as n. It means only a combination of n words will be considered from each array of words each time. Say, n=4. For example, got five windows if consider; [Corona- virus, deadly, disease, spreading, alarming, rate, everywhere, globe].
Output: [Corona-virus, deadly, disease, spreading], [deadly, disease, spreading, alarming], [disease, spreading, alarming, rate], [spreading, alarming, rate, everywhere], [alarming, rate, everywhere, globe]

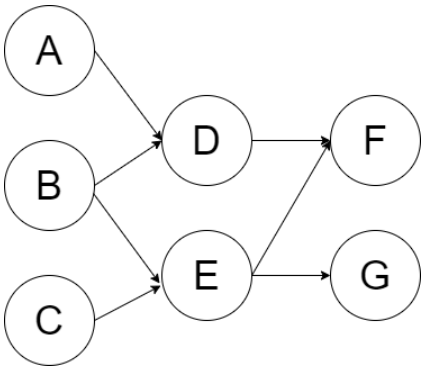


Fig. 4. Nodes representing words in a sentence

e) **Step 5:** From every window, two words form a pair. This pair forms a directed edge of the graph. Therefore, have (Corona-virus, deadly), (disease, spreading), (alarming, rate), etc.
Based on this, weight each word i.e. node. The words with the highest weights can be considered as relevant and marked as keywords.
Output: [Corona-virus, patients, mouth, nose, spread, alarming, acquainted]

$$S(V_i) = (1-d) + d * \sum_{j \in In(v_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

Fig. 5. PageRank Algorithm weight calculation formula

- S(Vi) - the weight of webpage i
- d - damping factor, in case of no outgoing links
- In(Vi) - inbound links of i, which is a set
- Out(Vj) - outgoing links of j, which is a set
- Absolute value of Out(Vj) - the number of outbound links

To get weight of the word the following function is used:

$$S(V_e) = (1-d) + d * \left( S(V_a) + \frac{1}{2} S(V_b) \right)$$

Fig. 6. Function to calculate weight

## IV. Results and Discussion

The below mentioned table shows the following parameters:

- Input link from which data was scraped
- Total number of words present in the extracted data.
- Total number of words after summarization of the extracted data.
- Keywords present in the extracted data. Keywords are displayed based on their scores and scores are not shown as the part of output.

Summarized data is not shown to avoid plagiarism. Summary includes sentences in the order of their importance in the document or website link. If summary of factual data as present on wikipedia or other informational websites or documents is required then order of the sentences in the summary does not matter. For summarization of news articles, blog posts, textual tutorial based websites etc, the order of the sentences in the original document or the website link should match the order of sentences in the summary therefore user has to choose whether the uploaded document or the chosen website link contains factual information or not. Based on that input the summary sentences are displayed in the required order.

Keyword extraction gives the important words from the document or link regardless of the type of textual information it contains.

| | |
|---|---|
| Link 1 | https://en.wikipedia.org/wiki/Mango |
| Total Words | 3345 |
| Words after Summarization | 1003 |
| Keywords | ['mango', 'fruit', 'cultivar', 'india', 'south', 'trees', 'cultivars', 'century', 'flavor', 'leaves', 'pulp', 'sauce', 'dermatitis', 'asia', 'salt', 'alphonso', 'variety', 'world', 'contact', 'chili'] |
| Link 2 | https://www.theguardian.com/world/2020/feb/25/what-is-coronavirus-symptoms-wuhan-covid-19 |
| Total Words | 617 |
| Words after Summarization | 185 |
| Keywords | ['cases', 'people', 'coronavirus', 'china', 'deaths', 'outbreak', 'health', 'flu', 'virus', 'syndrome', 'fever', 'human', 'sars', 'rate', 'korea', 'symptoms', 'vaccine', 'devlin', 'taiwan', 'spread', 'pneumonia'] |

## V. Conclusion

The proposed system implements website link and document summarization using natural language processing which helps the users to save time. The user is given the liberty to choose multiple links of their choice from any search engine. Multi-document and multi-webpage summarization support enables the user to use the functionality even more efficiently. It gives the summary of the individual links and also the combined summary of the links as per the user's requirement. This is what makes it different from the already existing systems. The keyword extraction feature also plays a vital role in providing the user with the gist of the complete document or website within seconds. The size of the summary is thirty percent of the total extracted text in the first step. The functionality to add direct text as input for summarization helps the user to obtain summary of blog posts, any other post from social media sites or particular textual data which they want to summarize.

## References

[1] *"Extractive Text Summarization Using Sentence Ranking"* - J.N Madhurt, Ganesh Kumar.R., 2019.
[2] *"Automatic Text Summarization of News Articles"* - Prakhar Sethi, Sameer Sonawane, Saumitra Khanwalker, R. B. Keskar, 2017.
[3] *"Towards Summarizing Popular Information form massive Tourism Blogs"* -Hua Yuan,Hualin Xu,Yu Qian,Kia Ye, 2016.
[4] *"Mining Topical Relevant Patterns for Multidocument Summarization"* - Yutong Wu, Yang Gao, Yuefeng Li, Yue Xu, 2015.
[5] *"A Multi-document Summarization System Based On Genetic Algorithm"* - Yan-xiang He, De-xi Liu, Dong-hong Ji3, Hua Yang, Chong Teng, 2006.
[6] *"A Scalable Multi-document Centroid-based Summarizer"* - Dragomir Radev , Timothy Allison, Matthew Craig , Stanko Dimitrov , Omer Kareem , Michael Topper , Adam Winkel , and Jin Y, 2004.
[7] *"A Summary On Extractive Text Summarization"* - N. Moratanch , S. Chitrakala , 2017.
[8] *"Review Paper On Extractive Text Summarization"* - Arpita Sahoo, Dr. Ajit Kumar Nayak, 2018
[9] *" Study On Text Summarization Using Extractive Methods"* - S.Mohamed Saleem, R.Krithiga, S.K.Rani, S.Celin Sindhya, 2015.
[10] *"An Integrated Approach to Web Document Summarization Using Semantic Similarity"* - K .Vanisri, P. Ponnila, J. Jeejovetharaj, 2014.
[11] *"A Summarization System for Scientific Documents"* - Shai Erera, Michal Shmueli-Scheuer, Guy Feigenblat, Ora Peled Nakash, Odellia Boni, Haggai Roitman, Doron Cohen, Bar Weiner, Yosi Mass, Or Rivlin,Guy Lev, Achiya Jerbi, Jonathan Herzig, Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, Debasis Ganguly, David Konopnicki, 2019.
[12] *"A survey on Real-Time Accumulative Short Text Summarization on Comment Streams"* - N. Vijay Kumar, Dr.M.Janga Reddy, 2017.
[13] *"Automatic Keyword Extraction Using Textrank"* - Papis Wongchaisuwat, 2019.
[14] *"Keyword Extraction Using Machine Learning Approaches"* - Bhavneet Kaur, Dr.Sushma Jain, 2017
[15] *"Performance Analysis of Keyword Extraction Algorithms Assessing Extractive Text Summarization"*- Akshi Kumar, Aditi Sharma, Sidhant Sharma, Shashwat Kashyap, 2017.