YouTube Video Content Extraction and Summarization with Automation

Y. V. Sravan Kumar Reddy, T. Yeswanth Sai, T. V. Premchand, T. Sasikala
Department of Computer Science and Engineering, Amrita School of Engineering, Bengaluru
Amrita Vishwa Vidyapeetham, India
yvsravan2000@gmail.com, yeswanth1809@gmail.com, venkataprem1999@gmail.com,
t_sasikala@blr.amrita.edu

***Abstract* – Data analysis and mining are playing a crucial role in today's business world of innovative organizations. The data is one of the essential factors to analyze trends and marketing strategies in business. Moreover, most of the data available on earth are unstructured sources, i.e., audio, images, images in motion(videos), and many followed. These unstructured data sources also contain information that is furthermore used for data analysis and learning. Most of the online videos we see in our day-to-day life have more data to get information on a specific range of topics. YouTube videos can also be used as unstructured sources to gain knowledge by formatting them in a structured way. In this project, we created a software program that can extract information from the YouTube video. Mainly, there are two types of content collected from the YouTube video: video details like title of the video, views, likes and dislikes count, tags linked, description attached to the video, and many other elements seen in YouTube videos webpage. The other type of content is a summary of the video based on the text obtained from speech attached to the YouTube video.**

***Keywords* – YouTube; Automation; Summarization; Selenium; Sentence Ranking Algorithm; Natural Language Processing; Web Scraping; Graph-Based method.**

I – INTRODUCTION

Social media is the platform where most people across the globe share their thoughts, opinions, etcetera in the form of text, speech, images, video, and many other formats. Moreover, this social media is acting as a medium to express their ideas and experiences to the world, and videos one among those: YouTube, one of the numerous notable online video sharing and streaming applications. Video or collection of images with added motion is one of the great examples of unstructured data sources. Videos are sometimes combined with audio to enrich their values. There are many sources online to access different types of videos, and YouTube is no exception to that. YouTube, the most desirable for entertainment, video creating, uploading, and viewing platform. It has a vast database containing videos of different categories for different types of users.

YouTube, a platform that allows billions of people to connect, share, inform, and motivate other people by creating videos and sharing them with the public. According to Google's statistics in 2019, there are more than two billion active users for YouTube. YouTube is the second most widespread social medial stage, with almost 75% of daily internet users having a YouTube account. YouTube, a platform that includes video uploading, sharing, and downloading. It is also a marketing stage for many products and innovative ideas. Substantially, the video content uploaded every 60 minutes on YouTube is about 30,000 play hours on an average. Every day, people on the internet watch one billion hours of video content from YouTube, which says how people need to gather information from videos.

The users are of different kinds with different mindsets. Everyone has their thoughts of experience in seeking information. YouTube is way more ahead by creating different types of categories to satisfy every user's needs. YouTube contains a wide range of categories as Film & Animation, People & Blogs, Autos & Vehicles, Comedy, Music, Entertainment, Pets & Animals, News & Politics, Sports, How to & Style, Gaming, Science & Technology, Education, Nonprofits and Activism.

In this paper, we focus on the YouTube platform to extract the information and make it easier for every user on the internet to make their work more comfortable and efficient. We spend some noticeable amount of our time watching YouTube videos every day, be it for education, sports, entertainment, or exploring our interests. Moreover, we know that we are watching it for information. We want to develop a software program that automates information extraction from a video and of the video.

One who creates YouTube videos is called a YouTuber or a Creator. Users have many goals while achieving their tasks and information received through any source. As stated, the system will generate two types of outputs that can be used by two sets of beneficiaries.

Output type-1: YouTube Video details; these are extracted from the webpage and stored under string format. Youtubers, data analysts, and many others can get benefit from this output of video details. Considering a YouTuber scenario, a YouTuber wants to rank their video on top of the search results within a specific category. The solution for this case can be the system output proposed in this paper, using which they can see how the video is described, hashtags used, location (considerably country) it got posted, under which category video got posted, how well the title is represented and many other factors.

Output type-2: YouTube video content summary, this summary is generated from the text produced using the speech signals of YouTube video. Students, investors, reviewers (product, film, etcetera), and many others can benefit from this content summary output. Considering a scenario of the student; a student wants to prepare a last-minute note for revision from the YouTube videos using which they studied for a test-X. The solution for this case can be the system output proposed in this paper, using which he can get a summary of the content present in the YouTube video for preparing the revision notes, and the system was able to generate the text from the video, which can be used when there is enough time. Using these outputs, the user can save his time, data, and effort to gather information from any YouTube video.

II – Prior Work

In this modern world with numerous technologies, there is a large set of applications for web scraping. Information extraction from the online web pages helps us to have detailed information about the data. The framework used in collecting raw data from the web is gathered and extracted as per the user requirements. Retrieval of data from the web using a scraper and other tools is useful. The data extraction plays a significant role in many applications like Data Mining etc. [1]. Regular Expressions or patterns that are to be fetched from the web page can be made using simple parsing techniques implementing libraries such as Beautiful Soup, etc. In this part, the first instruction to be mentioned will be the extraction using a particular library constructor [2]. The technique of extracting web data automatically refers to web scraping. In this, parsing the HTML part of the web pages using programs is the primary step. Some preamble server pages have adaptive techniques for the specific task, making the work easier by directly updating the web server connection [3]. The filtering or cleaning of the text document for the sake of important information is called summarization. The different ranking approaches like graph-based ranking approach provide us with the briefest and exact data for the software's raw text [4]. Using various scraping techniques, we can collect a massive amount of data in a shorter time on the Internet. This data is either related to it or the data that connects through the world. We can later apply machine learning techniques on the retrieved data for a specific use [5]. In this, TF (Term frequency) and IDF (Inverse document frequency) plays the crucial role for scoring the sentences and will be using TF-IDF measure for assigning a score to the sentences and making out the cleanest sentence possible [6]. The text containing sentences is split into words when it is ready to be summarized. Later, they applied the text rank to the sentences based on the sentence score that is taken earlier. The graph's vertices are considered in place of text. They will be ranked accordingly, which results in a highly ranked tokens/sentences in the graph. It provides us with the most ranked sentences in which the semantics will be considered [7]. The provided input text under a string variable and the string of sentences will be divided into tokens. The process of cleaning the sentences is done by removing stop words in the early part. The system will assign the token to POS tags that fall under weights of the tokens and ranks are calculated for the sentences. The highly ranked will be considered as the summarized sentences [8]. In this, the study of different string methods useful for extracting data on the web plays a key role. Analysis of other functional parsers helps to gain knowledge, particularly in the efficiency of the extraction that we are performing [9]. In text-summarization word frequency, length of the sentence, sentence position, title similarity, heading similarity, and sentence to sentence cohesion and several other aspects play a significant role in sentence ranking algorithm. Also, they bring out a clear idea about the text summarization [10]. The HTML parser working using python library (Beautiful Soup) helps us extract the desired content by removing the HTML tags in the web page's skeleton structure [11]. We can do this preprocessing of the text by obtaining the tokens and removing stop words with the help of two efficient methods, the clustering and support vector machine algorithms, and the improved summary obtained from both techniques [12]. Extraction of the applause and non-applause parts in an audio clip using the particular audio signal's autocorrelation features and that detection algorithm plays a significant role in extracting the audio features for the input audio format [13]. An extractive summarized version of the text is provided after undergoing data collection, preprocessing, and extraction of relevant sentences from the raw text. To obtain the correct cluster of sentences, the DBSCAN algorithm helps analyze the cluster count [14]. To convert data that the system can understand is called preprocessing. Irrelevant data like stop words takes more time and space, which is entirely unnecessary. NLTK tool kit helps us in removing these words [15].

III – IMPLEMENTATION

The process from taking input from the user till generating the output information involves several essential steps and sub-process to make the system work efficiently and produce better results. The brief workflow model of the implementation is described in Figure 1 given below.

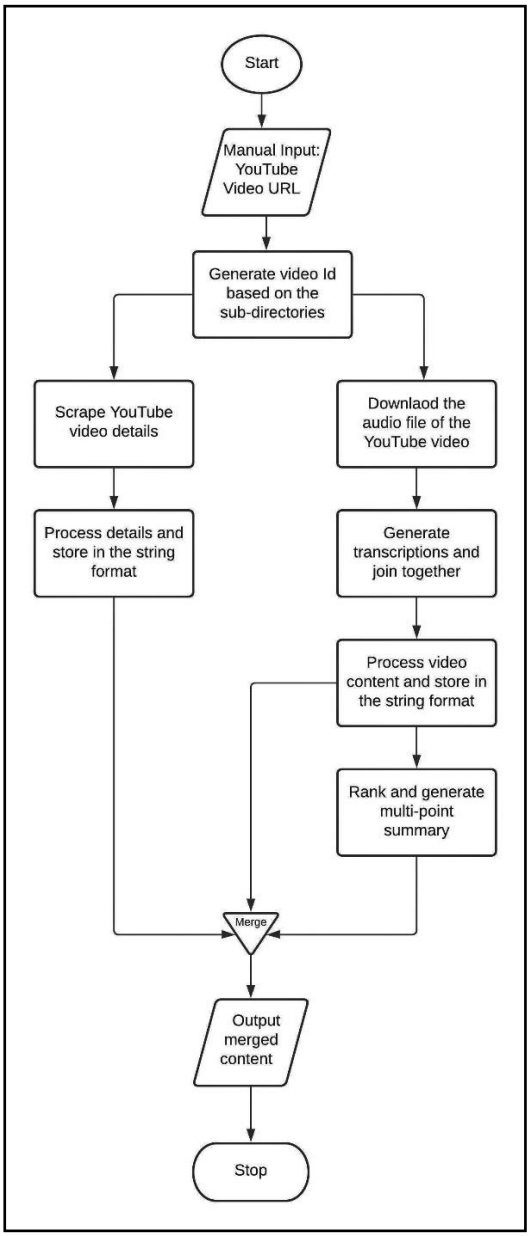

Figure 1: Basic workflow model for YouTube video content extraction and summarization with automation

To make the system capable of producing the outputs, the system needs an input termed as the URL of any YouTube video. Users should input the YouTube video URL of their choice and give the system some time to run and produce the specific URL of the YouTube video. Based on the input-YouTube video URL link and regular expressions, we will separate the video ID from the URL and store it in the new string variable for further process.

After extracting the YouTube video ID, the successive step is scraping the YouTube video details parallel to the video content generation.

Web scraping is the processing of scraping the data present on any web page on the worldwide web, and it widely used technique for data mining. Any web scraper has mainly two parts as web crawler and data extractor. Data extraction can be done in many ways like HTTP programming, web wrapping, tree-based techniques, semantic annotation, human copy and paste, HTML parsing, etcetera. Based on the type of data to extract and from which web page to collect will decide which technique to be used for that respective problem. To scrape the video details from the YouTube video webpage, web scraping and HTML parsing as data extraction technique can be implemented for better results as shown in the Figure 2.

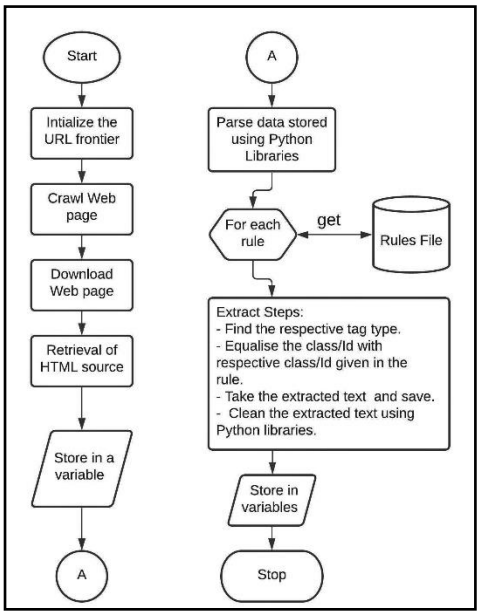

Figure 2: Workflow of web scraping model

In the first step, the URL frontier is initialized with the input URL given by the user. The web page crawling starts based on the given input, and the web page gets downloaded and saved in the local storage/database. The web page stored is parsed using the Python library – BeautifulSoup. As webpages are created using the markup languages like HTML – Hypertext Markup Language or XML – Extensible Markup Language, they also contain useful information in the form of text. Web browsers help to extract data from the world wide web. Selenium, an automation tool that plays a significant part in automating web application processes. Using the BeautifulSoup and web driver, the HTML code of the respective webpage will get parsed. This BeautifulSoup library is used to clean the data from the document after isolating the links, text, headings, etcetera from HTML tags. For every extracted form of data, pre-defined rules are applied to collect the

video details: video name/title, video duration, published data, views count, likes count, dislikes count, description, channel name, category, location and hashtags used. For all the extracted elements and tags, the rules are applied based on the class and Id. After finding the respective tag type, match the individual class or Id values, successively after matching the text is extracted. All the video details are cleaned by removing extra whitespaces and next line characters using inbuilt necessary python libraries and BeautifulSoup and stored in a string variable dictionary.

TABLE I. SAMPLE URL LIST

URL	Sample URL for scraping details
URL – 1	https://youtu.be/opixrHfHVDU
URL – 2	https://youtu.be/Aj9OWdtwDX4
URL – 3	https://youtu.be/83R9sFBt3hE
URL – 4	https://youtu.be/ukzFI9rgwfU
URL – 5	https://youtu.be/MPWn_or_gvI
URL – 6	https://youtu.be/Sazgd9KU-Mg
URL – 7	https://youtu.be/dhYOPzcsbGM

The executions time for scraping all the YouTube video detail from the sample URLs given in Table 1 are visualized as a graph in Figure 3.

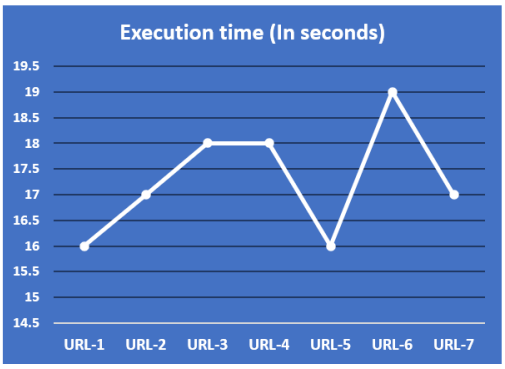

Figure 3: Graphical representation of URLs and execution time for scraping YouTube video details

To download the audio file related to any specific YouTube video Python library - youtube_dl is used. The specifications to be set with corresponding values for downloading the audio for processing, as shown in Table 2.

TABLE II. SPECIFICATIONS FOR AUDIO FORMATTING

Specification	Value
format	bestaudio
key	FFmpegExtractAudio
preferredcodec	wav
preferredquality	192
prefer_ffmpeg	True
keepvideo	False

Transcription is the process of converting speech signals into text format. Here, the speech signals can either be recorded or provided at real-time. The Python library - youtube_transcript_api can be used for getting transcriptions of any YouTube video. There is another method wherein the system needs audio as input. The audio is converted into text using the Google application program interface – API or python library speech recognition. To get another output: video content or text, the above methods can be used. After joining all the individual transcripts, it is stored as a string variable as text. This text is cleaned and processed to get the summary using natural language techniques and sentence ranking method.

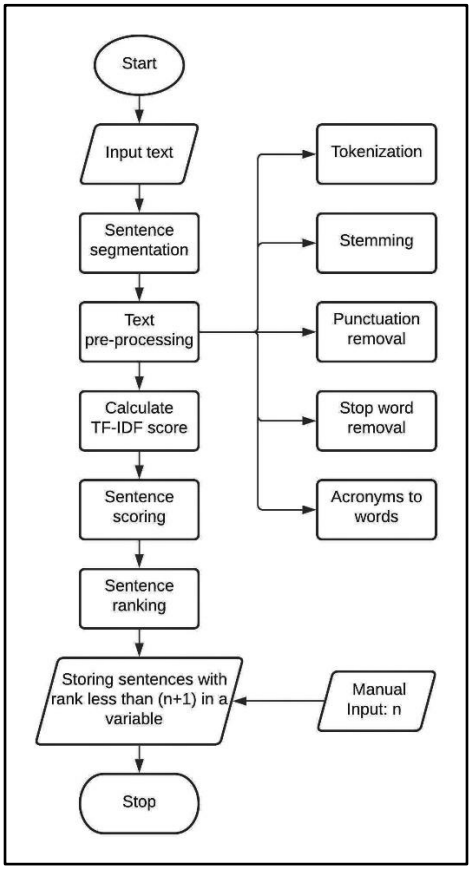

Figure 4: Workflow of Sentence ranking algorithm

The sentence ranking algorithm is used to generate a multi-point summary based on the given text as explained in Figure 4. To create a summary, the first step is to do sentence segmentation from the input text. Which is then followed by text pre-processing techniques which include sub-processes like:

A. Tokenization

It is a method of dividing a text into smaller units called tokens, which are the text corpora's unique elements.

B. Stemming

It is used for obtaining the root words from different forms of a root word by removing the suffixes, prefixes.

C. Punctuation removal

It is a technique used in pre-processing the text data to generalize the text, where punctuations (like comma, pull-stop, question mark) are removed.

D. Stop word removal

It is a technique used in pre-processing the text data by removing the stop words (which are more frequently occurred in the text and adds only a little value).

E. Acronyms to words

Acronyms are the type of abbreviations where every word's initial letter is clubbed together as OMG is the acronym of Oh My God.

After pre-processing the text, the cleaned text will be prepared for sentence scoring using TF-IDF scores. TF stated as term frequency, and IDF inverse document frequency provides a statistical approach by calculating it for every word present in the text document.

TF(Word-X) = (Word-X count in document) / (Total count of words in document)

IDF(Word-X) = loge (Total count of documents / Count of documents with Word-X in it)

TF-IDF(Word-X) = TF(Word-X) * IDF(Word-X)

After finding the TF-IDF score for each word, the sentences containing these words are scored based on the individual scores of words in the sentences. Based on the given input value – n, a threshold is set to value of n. The first n-ranked sentences are given as multi-point summary. The executions time for extracting video content as text and producing text summary from the sample URLs given in Table 1 are visualized as a graph in Figure 5.

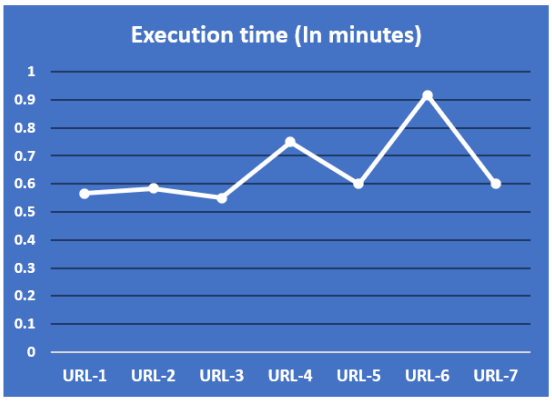

Figure 5: Graphical representation of URLs and execution time for summarization

After implementing all the required natural language processing techniques, web scraping, and sentence ranking algorithm using selenium to automate the process of YouTube video content extraction and summarization will produce better results to analyze the data.

IV – RESULTS

The results for the given input link: https://youtu.be/1Hx8_BAfgj8 are represented as output shown in Figure 6, Figure 7 and Figure 8.

```
Video Name        :   What is Machine Learning - in
3 minutes
Video Duration    :   2:14
Published Date    :   Mar 9, 2017
Views Count       :   10177
Likes Count       :   126
Dislikes Count    :   3
Channel Name      :   Raj Ramesh
Hash tags         :
Description       :
 Here's my attempt to demystify machine learning b
y explaining a simple version of machine learning.
```

Figure 6: Output – Video details

```
Simple Machine learning we hear so [ much ] about machine learning And I th
ought I'd illustrate how machines actually learn. So here is one of the sim
plest forms of machine learning. That'S based on a data set., I'm looking f
or a used car but buying a car is a harrowing experience. So I decide to us
e machine learning to tell me what price I should pay [ for ] a car. Let'S
say: here's the data for used cars sold in my neighborhood; It shows how ol
d the car is and what price it sold for. If I plot this with the price agai
nst how old the car is, I get something like this. As You can see, there's
somewhat of a pattern, I Can draw a line [ like ] this to approximate how m
uch I should pay for say a seven year old car Which turns out to be about s
eventeen thousand dollars Using basic Algebra. I can write down an equation
for the line. Looking at the line that I just drew, it looks like the y int
ercept or the point at which the line cuts the y axis is 33. Also, the slop
e of the Line is about Negative 3. So the equation of this line is y equals
negative 3x + 33 or In other words the price is equal to negative 3 times t
he age of the car + 33. If I can program a computer to read these data poin
ts and come up with an equation, Then, given any value of x, which is the a
ge of the car, the computer can come up with the price. Essentially, it has
learned from this data. How does the computer actually come up with the equ
ation based on this data? Here'S how first it assumes an arbitrary equatio
n, then, for that equation it computes an error term Which basically tells
it how far it is from the data points. If the error is too high, Then it ad
justs the two parameters, the slope and the y-intercept. In a way that redu
ces this error, Then it does it again and again and again until it comes up
to a close approximation of the data, Then it stops. The equation is the mo
del that's being created Using this model, it can tell the price of the car
Given the year. That'S it Now, the same principle can be extended to add mo
re dimensions to the Data such as the car mileage, Zip code, car model and
so on. So [ here ] are the things that matter for any machine learning The
Data it has to be sufficient. The model it has to be well chosen and the Al
gorithm It'll find the parameters of the model in this video.. I just wante
d to demystify machine learning, Thanks for watching
```

Figure 7: Output – Video content as text

```
--> Let'S say: here's the data for used cars
sold in my neighborhood; It shows how old the
car is and what price it sold for.
--> If I can program a computer to read these
data points and come up with an equation, The
n, given any value of x, which is the age of
the car, the computer can come up with the pr
ice.
```

Figure 8: Output – Video Content Summary

The above-shown results: video details and the video content/text and text summary of two points are generated in less than a minute time with the best accuracy in the meantime.

V – CONCLUSION

Scraping the YouTube video details and generating the summary along the text generated from the YouTube video based on the specific user query helps to get the video's overall thought and allows the user save their time, data, and energy.

VI – FUTURE SCOPE

Acknowledging the research did in this paper and work did so far, the system described can be improved and developed using several other artificial learning technologies. The possibilities include: generating the keywords present in the video content, determining how well the content matches the category, title, and thumbnail attached based on computer vision concepts. Also, scraping comments from the video webpage will give more scope of how well the audience is connected based on sentiment analysis.

REFERENCES

[1] David Mathew Thomas, Sandeep Mathur, " **Data Analysis by Web Scraping using Python**," International Conference on Electronics Communication and Aerospace Technology, [ICECA], Noida, 2018, doi: 19.05.18.

[2] Thivaharan. S., Srivatsun. G., Sarathambekai.S, "**A Survey on Python Libraries Used for Social Media Content Scraping**", e International Conference on Smart Electronics and Communication, (ICOSEC), Coimbatore, 2020 doi: 19.10.20./; ISBN: 978-1-7281-5461-9.

[3] Sanjay Kumar Malik, SAM Rizvi, **"Information Extraction using Web Usage Mining, Web Scrapping and Semantic Annotation**", International Conference on Computational Intelligence and Communication Systems, New Delhi, doi: 2011.

[4] Neelima Bhatia, Arunima Jaiswal, "**Automatic Text summarization and its methods – A Review**", International Conference - Cloud System and Big Data Engineering, Noida, doi: 18.6.2016.

[5] Deborah Kurniawati, Deny Triawan, "**Increased Information Retrieval capabilities on e-commerce websites using scraping techniques**" International conference on Sustainable Information Engineering and Technology, (SIET), Yogakarta,Indonesia, doi: 2017.

[6] K Usha Manjari, Syed Rousha, Dasi Sumanth, Dr. J Sirisha Devi, "**Extractive Text Summarization from Web pages using Selenium and TF-IDF algorithm**", Proceedings of the Fourth International Conference on Trends in Electronics and Informatics, (ICOEI), Hyderabad, doi:2020.

[7] Khushboo S. Thakkar, Dr. R. V. Dharaskar, M. B. Chandak, "**Graph-Based Algorithms for Text Summarization**," Third International Conference on Emerging Trends in Engineering and Technology, /ICETET, Nagpur, doi: 10.1109.

[8] J.N.Madhuri., Ganesh Kumar.R,, "**Extractive Text Summarization Using Sentence Ranking**", IEEE, Bangalore, doi:19.03 2019.

[9] ERDINÇ UZUN, " **A Novel Web Scraping Approach Using the Additional Information Obtained From Web Pages**",IEEE volume 8,Tekridag, Turkey doi: 31/3/2020.

[10] P.Krishnaveni, Dr.S. R. Balasundaram, "**Automatic Text Summarization by Local Scoring and Ranking for Improving Coherence**", International Conference on Computing Methodologies and Communication, (ICCMC), Tiruchirapalli, doi: 17.08.2017.

[11] Momin Saniya Parvez, Shivankar Sneha Rajendra, Khan Shaista Agah Tasneem, Kalpana R. Bodke, "**Analysis of Different Web Data Extraction Techniques**", IEEE, New Panvel, doi: 2018.

[12] K. Mab Shiva Kumar and Soumya, Rab, "**Text summarization using clustering technique and SVM technique**", International Journal of Applied Engineering Research, vol. 10, pp. 25511-25519, Mysore, doi: 2015.

[13] C. Manoj, S. Magesh, Aditya Sriram Sankaran, Dr. M. Sabarimalai Manikandan, "**Novel Approach for Detecting Applause in Continuous Meeting Speech",** IEEE, Department of Electronics and Communication Engineering Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore doi:11.03.2011.

[14] Gayathri Venu Madhuri Chandu, Amritha Premkumar , Sai Susmitha K , Nalini Sampath, "**Extractive Approach For Query Based Text Summarization**", 2019 2nd International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)/Amrita Vishwa VidyaPeetham ,Bangalore, doi: 2019.

[15] Sai Srihitha Yadlapalli, Rakesh Reddy R, T. Sasikala, "**Twitter Sentiment Analysis using Machine Learning Techniques"** IEEE/Department of Computer Science and Engineering, Amrita School of Engineering, Bengaluru.