e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

SUMMARY AND KEYWORD EXTRACTION FROM YOUTUBE VIDEO TRANSCRIPT

Shraddha Yadav [*1], Arun Kumar Behra [*2], Chandra Shekhar Sahu[*3], Nilmani Chandrakar[*4]

[*1]Faculty, Department of Computer Science and Engineering, Government Engineering College, Raipur, Chhattisgarh, India.

[*2,3,4]Student, Department of Computer Science and Engineering, Government Engineering College, Raipur, Chhattisgarh, India.

ABSTRACT

On YouTube, lots of videos are uploaded every single day around the world which are very long, mostly educational. And if you want to find some useful information from those videos, it is an almost impossible task to achieve since most of the videos contain so much useless buffer content and less useful content, because of this you have to watch the whole video which probably wastes your lots of time and efforts or watch multiple videos to extract that useful information. So generating the summaries of those video transcripts will save you lots of time and you will quickly gather more useful and important information from it, which will surely save your efforts and time to watch the whole video. The project which we have made is extracting the summaries from video transcripts and also generating important keywords from it which will use Natural Language Processing methods for extractive and abstractive summarization.

Keywords: Natural Language Processing, Extractive Summarization, Abstractive Summarization, Machine Learning, Youtube, Keyword Extraction.

I. INTRODUCTION

Natural Language Processing (NLP) is a field of Artificial Intelligence that focuses mainly on the study of the interaction between human languages and machines. Generating summaries of video transcripts is the process of generating short, fluent, and most importantly accurate summaries of longer videos. The main idea behind it is to be able to find a short subset of the most essential information from the entire set and present it in a human-readable format. As online textual data grows, automatic summarization of text methods has the potential to be very helpful because more useful information can be read in a short time. Generating summaries of transcripts is in the field of NLP because machines are required to understand what humans have written and produce human-readable outputs. Nowadays, it has become very easy to watch videos on Youtube for anything, be it educational or entertainment, there are lots of videos of those kinds of genres and because of the number, it is very important for us to find out the exact content that we want to consume. And sometimes because of network issues, we aren't able to watch text-based YouTube videos in high resolution, since it makes the text blurry. There are also those long and click bait videos that are only made to earn profit that contain not a single bit of useful information which wastes our time. So, by removing the useless part of the videos, skipping ads, and getting summaries, we can directly jump to the information that interests us and save lots of time and effort. The main objective of our project is to save the time of a user and increase work efficiency. There are countless situations when a user lures into the catchy title and thumbnail of the video and wastes all its time in consuming useless information. Students browse youtube videos before their exams so that they can get the most of the information in less time because of that they usually watch videos at double speed and because of this, it doubles the confusion about totally new topics. Sometimes people want to know what is happened in a long meeting in short, if the meeting is recorded and has a transcript we can find out its summary which will save lots of time for the people of the organization. This is where our project is pointing out, it will create a summary of the transcript or the captions of whatever video you are watching. The most important part of our project is to collect the most necessary information and concentrate it into a small paragraph. Getting all the necessary information about the topic which interests you in the form of that small paragraph will save lots of time and effort for other things, which was wasted when you were lured into click bait videos and wasted your whole half or one hour or more. Our main goal is to save the time of the user who was wasting time on finding useful information about the topic which they are interested in and to save them from click baited videos.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

II. METHODOLOGY

First, we need to get the subtitles or transcript for a given Youtube video id by using the python API known as youtube_transcript_api. Since there are three types of transcript that we can extract - manually generated transcript, automatically generated transcript, and the videos that contain no transcript. We are not considering videos that do not have transcripts.

Secondly, when we get the transcript of a given Youtube video since it does not contain any punctuations like comma(,), full stops(.) which is very important for us in finding the boundaries of a sentence, so we will restore punctuations from our extracted transcript by using the python library known as "punctuator".

Now we will apply the text preprocessing methods to clean the extracted transcript by tokenizing the sentences as well as the words, lowercasing it, removing stop words like a, an, the, etc, removing punctuations, and stemming or lemmatization to generate the root form of inflected words.

Performing text summarization: This task consists of shortening a large form of text into a precise summary that keeps all the necessary information intact and preserves the overall meaning.

For this purpose in NLP for text summarization, there are two types of methods used:

Extractive Summarization: In this type of text summarization, the output is only the important phrases and sentences that the model identifies from the original text.

For the purpose of extractive summarization, we have used the TF-IDF model with Text Rank Algorithm.

TF-IDF(Term Frequency - Inverse Document Frequency)

After the cleaning process, we have to convert the words into it's vectorized form so that our algorithm will process it by using **TF-IDF**.

This is a technique to measure the quantity of a word in documents, we compute a weight to each word which signifies the importance of the word in the document and corpus.

TF(Term Frequency): TF calculates the frequency of a word in a document.

TF = No. of repetition of the word in the sentence / No. of words in a sentence

IDF(Inverse Document Frequency): IDF is the inverse of the document frequency which measures the informativeness of term t.

IDF = log(No. of sentences / No. of sentences containing words)

After this, we will multiply both matrices to obtain the vectorized form which tells us which words are the most important.

Text Rank Algorithm

It is based on the PageRank algorithm which calculates the rank of web pages which is used by search engines such as Google. Using the concept of this we will rank the most important sentences in the text and generate a summary.

For the task of automated summarization, TextRank models any document as a graph using sentences as nodes.

A function is computing the similarity of sentences to build edges in between. This function is used to weight the graph edges, the higher the similarity between the sentences the more important the edge between them will be in the graph.

TextRank determines the relation of similarity between two sentences based on the content that both share. This overlap is calculated simply as the number of common lexical tokens between them, divided by the length of each to avoid promoting long sentences.

Cosine similarity of two words A & B is computed using following formula:

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

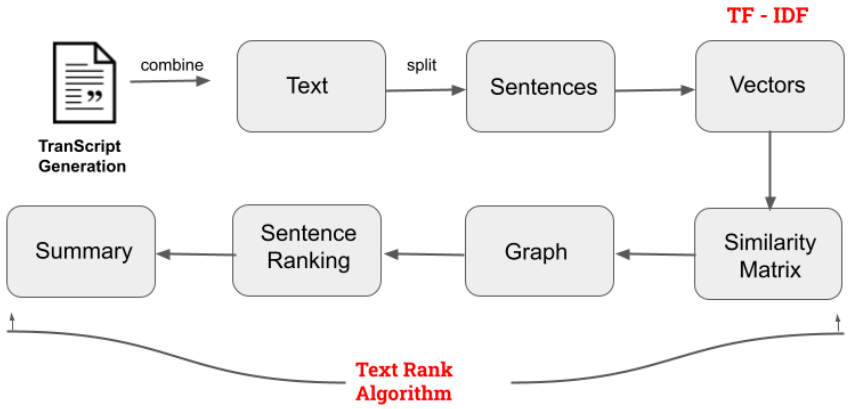


Figure1: Flow of summarization

Abstractive Summarization: In this type of text summarization, the output is a different text which is shorter than the original, is generated as a new sentence in a new form by the model. It is just like some human generating summary.

In this step, we have used the Encoder-Decoder Seq2Seq model with an attention mechanism to perform abstractive summarization of the text in a transcript after the cleaning process.

Sequence-to-Sequence Model:

A Sequence-to-Sequence model takes some sequence of information as an input and generates a sequence of information as an output like mapping a fixed-length input with a fixed-length output where the length of input and output may differ.

Figure 2: Seq2Seq Model

Before giving a sequence of input(words) to the seq2seq model we convert the words into it's vectorized form for that we use some word embedding techniques like Word2Vec, Glove, etc., in our project we have created an embedding layer that converts words into some vector representation so that our model can able to generalize the words to do any kind of prediction or summary generation.

The model composed of an encoder and decoder:

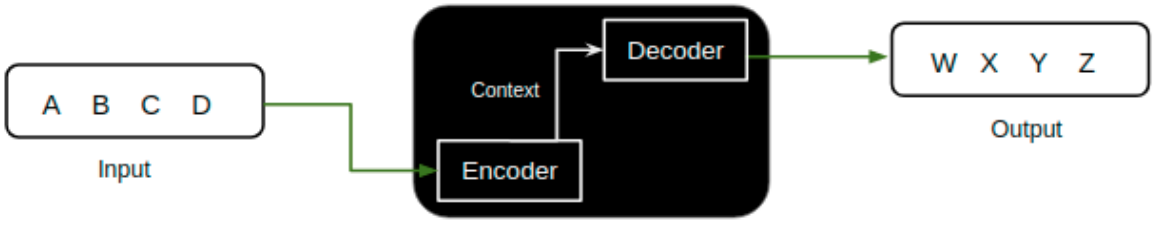


Figure 3: Seq2Seq Model

1.Encoder

- It consists of several recurrent units such as LSTM, which captures the context of the input sequence in the form of a hidden state vector and generates a final embedding at the end of the sequence which is known as context vector, this is then forward to the decoder.
- In our project, we have used three encoders consisting of recurrent unit LSTM.

- The hidden state h(i) are computed using the formula:

$$h_t = f(W^{(hh)}h_{t-1} + W^{(hx)}x_t)$$

- The formula represents the result of LSTM. we tend to simply apply the acceptable weights to the previous hidden state *h(t-1)* and also the input vector *x(t)*

Context Vector: Its aim is to encapsulate the data for all input elements so as to assist the decoder make correct predictions.

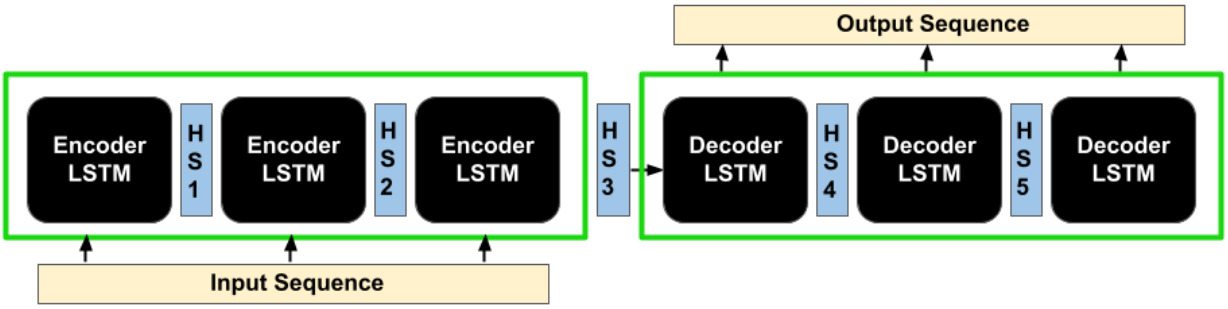


Figure 4: Encoder-Decoder Model for Seq2Seq Modelling

2.Decoder

- A context vector is then sent to the decoder which also consists of recurrent unit LSTM, which then uses it to predict the output sequence y(t) at a time step t, and after each successive prediction, it uses the previous hidden state to predict the subsequent sequence.
- The hidden sate h(i) is computed using the formula:

$$h_t = f(W^{(hh)}h_{t-1})$$

- We are using the previous hidden state to work out the subsequent one.
- The output y(t) at the time step t is calculated using the formula:

$$y_t = softmax(W^S h_t)$$

- We are calculating the outputs using the hidden state at the present time step alongside the individual weight W(S). Softmax is used to create a probability vector which can facilitate us to determine which word is to incorporate within the summary.

Attention Mechanism

Problem with encoder-decoder is that its output heavily depends on the context defined by the hidden state in the final output of the encoder, since only final layer of encoder generates output which is a context vector, so it makes more challenging for the model to deal with the longer sentences. If any long sequences came in the input, there is a high probability that the initial context has been lost by the end of the sequence.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

So the answer is to use a method known as "Attention" that permits our model to target completely different components of the input sequence at each stage of output sequence i.e. considering outputs of each encoder layer allowing the context to remain intact from beginning to end. Since a single hidden state vector at the end of the encoder wasn't enough, we sent as many as hidden state vectors.

Now the decoder will get the hidden states of all the instances of the input during the phase of decoding.

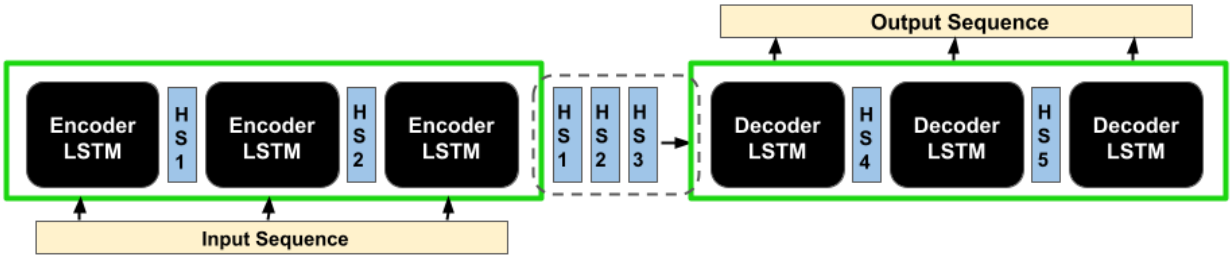


Figure 5: Seq2Seq with Attention - Incomplete

Context vector is generated for each time instance within the output sequences. At each step, the context vector is a weighted add of the input hidden states.

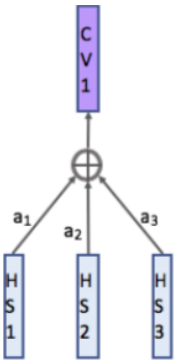


Figure 6: Context Vector

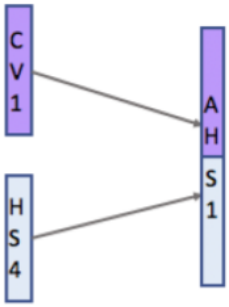


Figure 7: Attention Hidden State

But how is that the context vector utilized in the prediction? And how a1, a2, a3 weights are decided?

The generated context vector is combined with the hidden state vector by concatenation and this new attention hidden vector is employed for predicting the output at that point instance. Note that this attention vector is generated for each time instance within the output sequence and currently replaces the hidden state vector.

Attention scores are the output of another neural network model, the alignment model, that is trained collectively with the seq2seq model at first. The alignment model scores how well an input (represented by its hidden state) matches with the previous output (represented by attention hidden state) and will match this matching for each input with the previous output. Then a softmax is taken of these scores and the resulting number is the attention score for every input.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

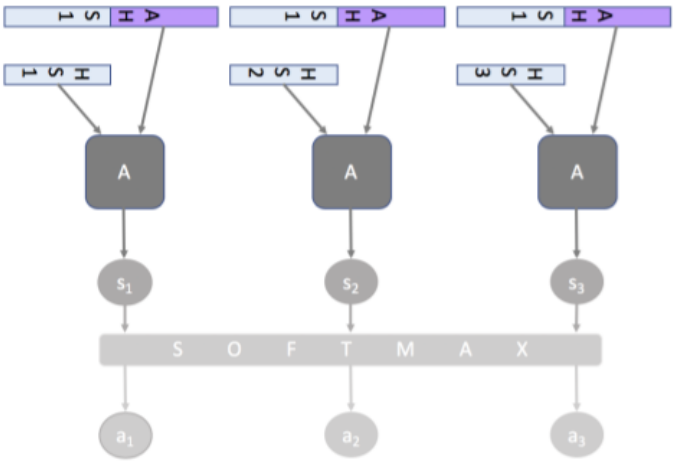


Figure 8: Attention Scoring

Hence, we have a tendency to currently apprehend that a part of the input is most vital for the prediction of each of the instances within the output sequence. Within the training section, the model has learned a way to align numerous instances from the output sequence to the input sequence. Below is an illustrated example of a computational linguistics model, shown in an exceedingly matrix kind. Note that each of the entries within the matrix is the attention score related to the input and also the output sequence.

So currently we've got the ultimate and complete model:

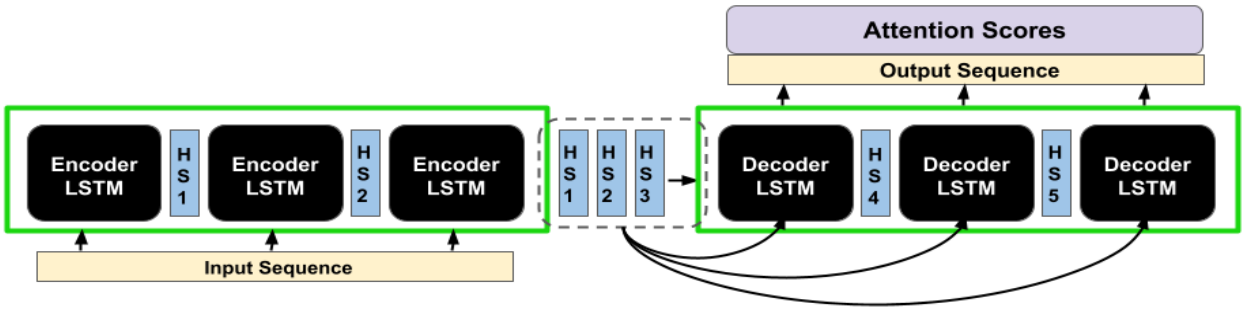


Figure 8: Seq2Seq Attention Based Model

Now we are going to train the created model and reserve it for the convenience of the long run use, and using this we will generate our video transcripts summary.

Now the **Keyword generation** part, for this we have used TF-IDF through which most important words are extracted whose values are most in the TF-IDF vector.

In the next step, we will build an API based on the Flask backend for the user to be exposed to the summarization service. API stands for Application Programming Interface which is the program that allows two applications to talk to each other so that one can access the data and features of other applications and services. There are many types of API that are used to implement various types of applications. And we can set up API as per our needs. And in next step we will build a React app which is used to show our extracted summary and important keywords, which uses an API to request the summary and important keywords of a particular video whose url is provided to our flask backend where our model will process the transcripts and sends summary and keywords to the user interface of our React app using an API.

III. MODELING AND ANALYSIS

These are the steps through which a user can obtain the summary and important keywords:

- Open our React app and paste YouTube video's URL on the input tag and click Get summary and keywords.
- Request Transcript for the Youtube video's id.
- Return the Transcript.
- Perform the summarization process.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

- Display summary and important keywords in the React app.

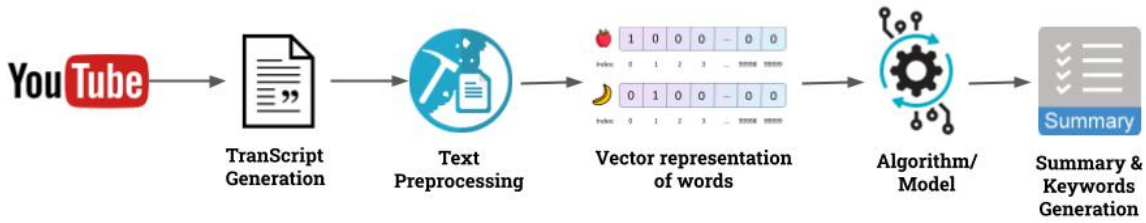


Figure 9

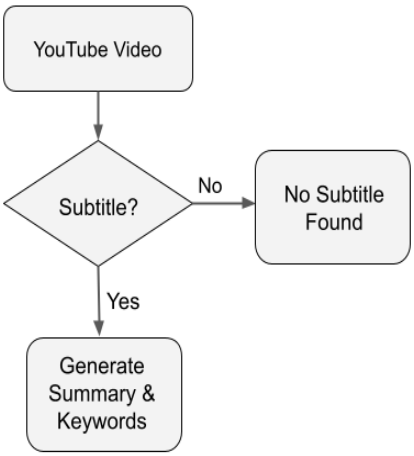


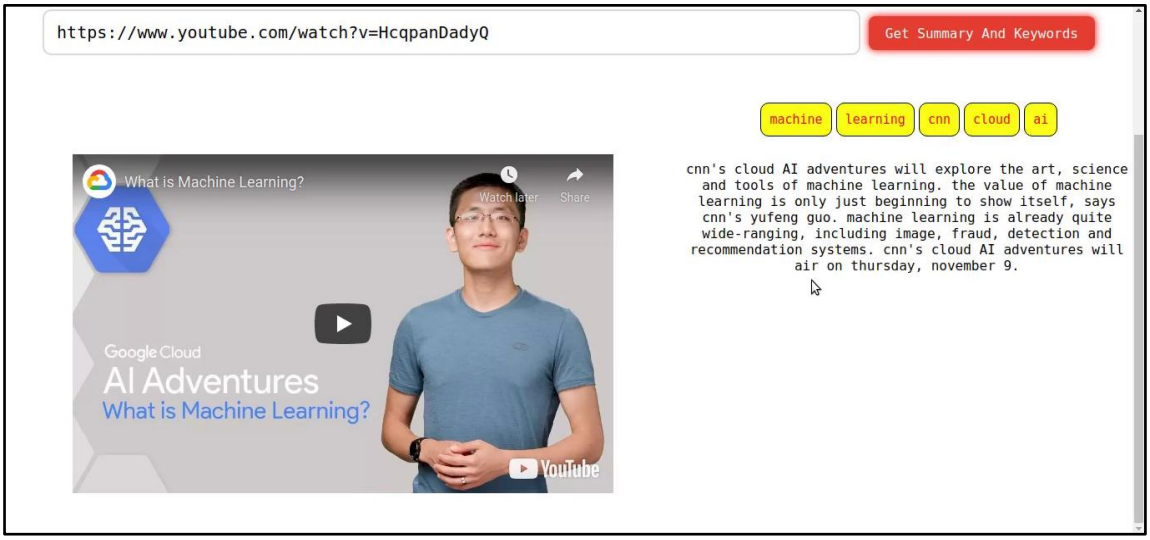
Figure 10

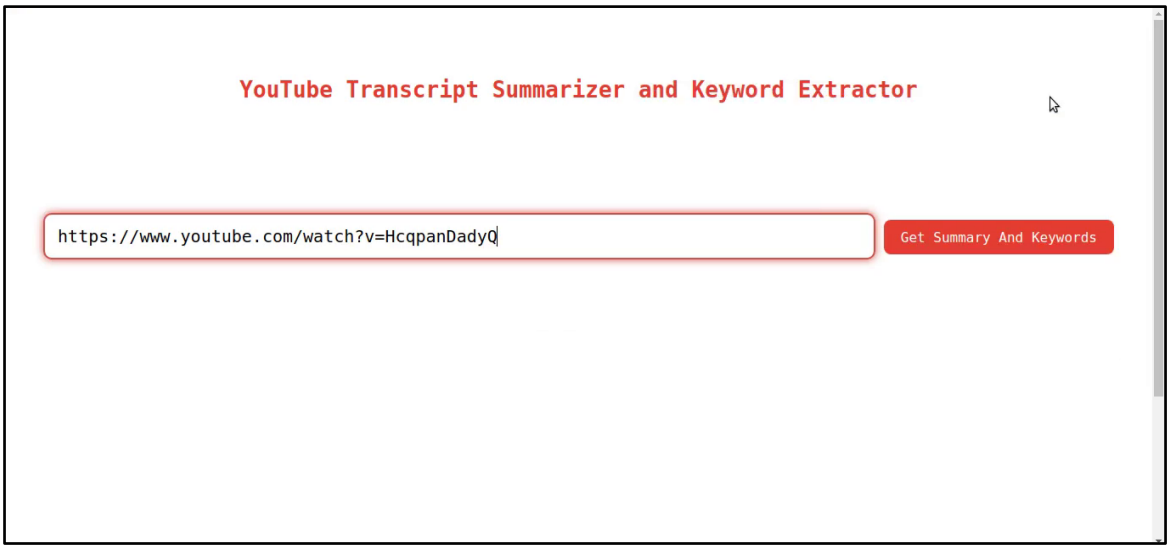
IV. RESULTS AND DISCUSSION

This project is used for following functions -

- Conferences – a fast outline can be generated from your team conferences and video conferences, to be employed in more conferences or additional intricate discussions.
- Patents research - how to extract the foremost vital claims across patents/research papers.
- Crash-Course – Students that watch YouTube videos to find out a theme topic quickly and concisely will get a quick read of the video and check if a connected course programme is present or not.
- Online Classes – Students that have missed classes during this online era of education will build notes from the summary of the video lecture. Also, notes can be simply obtained to be distributed to all the students.

The result of the project as follows:





V. CONCLUSION

In this paper, we propose two different methods to generate summary and important keywords from the given Youtube video - extractive and abstractive. We have made a simple user interface through which users can easily get their summaries through these methods, and surely find it easy to interact with our user interface and get what they want. We are sure that our project will surely satisfy the users and solve all the problems that it's supposed to tackle which is saving time and efforts, by providing only the useful information about the topic which interests them so that they don't have to watch those long videos and the time that saved can be used in gaining more knowledge.

VI. REFERENCES

[1] Anuj Gupta,Bodhisattwa Prasad Majumder,Harshit Surana and Sowmya Vajjala."Practical Natural language Processing" A Comprehensive guide to build Real World NLP System.1st edition. O'Reilly Media, Inc.(June,2020)

[2] Ankit Kumar, Zixin Luo & Ming Xu,"Text Summarization using Natural Language Processing". WPI Juniper Networks.(March,2018)

[3] Gaurav Sharma, Shaba Parveen Khan, Shivanshu Sharma and Syed Ubed Ali (2021)."Summarizer For Easy Video Assessment(SEVA)".Vol.03. e-ISSN: 2582-5208. (April,2021)

[4] Hamza Shabbir Moiyadi, Harsh Desai, Dhairya Pawar, Geet Agrawal, and Nilesh M. Patil.(2016)."NLP Based Text Summarization Using Semantic Analysis". IJAEMS. ISSN: 2454-1311.(Oct,2016)

[5] Korra Sathya Babu,Santosh Kumar Bharti and Sanjay Kumar Jena.2017. "Automatic Keyword Extraction for Text Summarization: A Survey".(Feb,2017).

[6] Pratik K. Biswas, Aleksander Lakubovich. (2020)" Extractive Summarization On Call Transcript".AI & Data Science, Global Network and Technology, Verizon Communications, New Jersey, USA.

[7] Cuneyt M. Taskiran , Arnon Amir, Dulce Ponceleon and Edward J. Delp. 2009. "Automated Video Summarization Using Speech Transcripts".Video and Image Processing Laboratory (VIPER).(Jan, 2009)

[8] Feifan Liu, Deana Pennell, Fei Liu and Yang Liu. 2009. "Unsupervised Approaches for Automatic Keyword Extraction Using Meeting Transcripts". Computer Science Department The University of Texas at Dallas Richardson, USA. (June 2009).

[9] Rada Mihalcea, Paul Tarau. "TextRank: Bringing Order into Texts". Department of Computer Science University of North Texas.

[10] Gang Liu, Jiabao Guo. 2019. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. School of Computer Science, Hubei University of Technology, Wuhan , China. (Feb,2019).

[11] Shengli Song, Haitao Huang and Tongxiago Ruan.2018. "Abstractive Text summarization using LSTM-CNN based deep learning. Springer Science+Business Media. (Feb, 2018).