Grenze International Journal of Engineering and Technology, June Issue

YouTube Video Transcript Summarizer using NLP

Hemil Sheth[1], Kalash Vishwakarma[2], Sobiya Shaikh[3], Prapti Sonawane[4] and Prof. Vaishali Hirlekar[5]
[1-5]Shah and Anchor Kutchhi Engineering College/Computer, Mumbai, India
Email: hemilsheth7@gmail.com, kalash8286@gmail.com, sobiyas363@gmail.com, parpti2002@gmail.com
vaishali.hirlekar@sakec.ac.in

***Abstract*—This research suggests utilizing natural language processing to summarize the YouTube video transcript in a way that preserves important details. Around one-third of YouTube users in India who view videos on their mobile devices spend more than 48 hours each month on the platform, according to a Google study. It has gotten tougher to sit through these films, some of which may go longer than they should. Our efforts can even be in naught if we can't find the information we need in them. It can take a lot of time and effort to find movies that provide the information we are really seeking for. There are a lot of online videos, for instance, where the speaker wants to capture the audience's attention, but we won't understand the message unless we watch the entire video. The built-in model accepts links to YouTube videos and creates summaries for each one. It also gives the user the ability to choose the algorithm and the percentage of summaries they want to create. Another feature that the user has access to is the ability to transform the generated summary from text to speech and to translate it into the Hindi language.**

***Index Terms*— Natural Language Processing , YouTube API, Machine Learning, TF-IDF, Text Summarization, Transcript, Gensim.**

I. INTRODUCTION

The Covid-19 pandemic has developed into the greatest menace to every aspect of society. In this scenario, the educational system is not an exception. Due to this epidemic, education has been the sector most severely impacted. Online classes are the sole hope for reaching students in this predicament. Since the pandemic, the majority of Indian universities and schools have adopted this strategy. They use a variety of online teaching techniques, including live and/or pre-recorded classes. Several systems, including YouTube videos, Cisco WebEx, Microsoft Teams, Google Meet, and Zoom have saved them. One such initiative is the National Programme on Technology Enhanced Learning (NPTEL) of the Indian Ministry of Education. Even before the pandemic, it was there. Several students register in these courses in order to advance their knowledge or to satisfy academic requirements. Occasionally, due to a lack of time, students find it challenging to handle both their academic work and these online courses. In this case, they require a system that lessens their workload by summarizing the video lesson for them without them having to watch it or read a computer-generated transcript. In this case, a video transcript summarizer has a wide range of applications. It draws attention to the crucial points in the video. The concept of summarizing videos can be applied to videos for online courses. The concept of summarizing videos can be applied to videos for online courses. Students will benefit from the time savings because they can quickly grasp the main points of the lesson by reading the summary rather than watching the video. Our approach focuses on the creation of a module that uses Python and Natural Language Processing to

Grenze ID: 01.GIJET.9.2.14_1

synthesize an online lecture video.

Millions of new films are uploaded to websites like YouTube, Vimeo, and Facebook every day, and there has been an explosion in the amount of video content available online in recent years. As a result, absorbing and comprehending all of this content has become a time-consuming process. Researchers have created a number of automated approaches for summarizing video transcripts to address this difficulty. These summarizing techniques can offer a succinct and instructive overview of a video's content, minimizing the time needed to comprehend its meaning.

Forms of Video Transcript Summarization: Extractive and abstractive summarization are the two primary methods of video transcript summarization.

The most significant sentences from a video transcript are identified, extracted, and combined using extractive summarizing techniques to provide a summary. This kind of summary is simple because it only includes highlighting the most crucial passages from the transcript. Contrarily, abstractive summarizing techniques create a new, reduced version of the video transcript by figuring out the text's underlying meaningand creating a new summary that does so. This kind of summary is more difficult since it calls for the capacity to comprehend the transcript's content and produce new language that is both brief and coherent.

Common Techniques for Summarizing Video Transcripts:

A. Extractive Summarization

In extractive summarizing techniques, the sentences in a video transcript are rated according to importance, and the top-ranked phrases are then chosen to create the summary. The TF-IDF method and the TextRank algorithm are two common techniques for extractive summarization.

B. Abstractive Summarization

Advanced NLP methods, like recurrent neural networks and attention-based models, are used by abstractive summarization methods to create a new, condensed transcript of the video. These techniques often entail developing a model on a big body of text data and then optimizing it for the particular purpose of summarizing video transcripts. Popular abstractive summarization models include the GPT-3 language model and Transformer-based approaches. Nltk and Gensim are methods for summarizing video transcripts.

II. Literature Survey

Ravali Boorugu, Dr. G. Ramesh [1] they talked about a survey on text summarization techniques, which are important for online shopping as people rely on reviews to pick the best products. With an increase in online shopping, it has become necessary to reduce long reviews to short sentences that convey the same meaning. The paper discusses various text summarization techniques, from basic to advanced, and recommends using a seq2seq model with LSTM and attention mechanism for increased accuracy.

Domain-Aware Abstractive Text Summarization in this paper [2] they talked about This paper introduces a deep-reinforced, abstractive summarization model for biomedical publications. While extractive approaches have been used in the past, this model uses deep learning to generate one-sentence headlines or titles. The model is capable of producing domain-aware, abstractive summaries using reinforcement learning reward metrics based on biomedical expert tools such as UMLS Metathesaurus and MeSH. Additionally, the model can learn domain-specific information without the use of expert tools, using a reward metric based on TF-IDF. The paper uses pointer-generator networks and deep reinforcement learning.

Natural Language Processing (NLP) based Text Summarization - A Survey [3] they talked about the need for text summarization to transform vast amounts of raw information on the internet into useful information for human consumption. The two main categories of text summarization, extractive and abstractive methods, are explained. Extractive methods are favored by NLP researchers due to their ability to choose relevant sentences from the text using linguistic and statistical characteristics. The paper analyzes both extractive and abstractive methods and highlights the importance of producing less repetitive and more concentrated summaries.

Krishna Kulkarni, Rushikesh Padaki [4] they talked about the development of a module using Natural Language Processing (NLP) algorithms to summarize online course videos. The model takes the URL of a video from the user and uses NLP techniques such as Term Frequency-Inverse Document Frequency (TF-IDF) and Gensim to obtain a summary. The model also provides flexibility to the user to decide on the percentage of summary needed compared to the original lecture. The paper presents two summarization techniques - cosine similarity and ROUGE score - and evaluates their efficiency. The cosine similarity approach performs better than the ROUGE score, with an efficiency of over 90% using both TF-IDF and Gensim. The paper highlights the

potential of video transcript summarization for online education, enabling students to save time and quickly understand the gist of a lecture.

III. Problem Statement

A. Problem Statement

People can now access a variety of instructional resources online, including YouTube videos, in today's world. The main issue here is how time-consuming it is to watch hours of videos to comprehend fundamentalideas. As a result, our software enables the students to read and comprehend a condensed text version of the video's key points. The goal is to examine the text that has been created by summarizing YouTube videos into a format that only displays their essential content.

B. Objectives

1. Making a website where the YouTube link and summary are posted.
2. To record and pull the YouTube video's transcript.
3. To assess the summary produced by various algorithms.
4. To assess which algorithm produces better outcomes.
5. Based on the percentage that the user has entered, summarize the text.

C. Scope

We can combine summarizing techniques to learning sites like YouTube by using five algorithms to efficiently construct summaries of YouTube videos. The five algorithms are Frequency Based (NLTK), Latent Semantic Analysis Based (Sumy), Luhn Algorithm Based (Sumy), and Text Rank Based (Sumy)Viewers can download the summary of the video by selecting the download option we provide.
This downloaded movie is stored on the user's computer as a text file with the.txt extension. We are now creating an application for Computers and phones with lower specifications, so we can design an extension for them. We will benefit from this because the extension will only work on browsers and will be stored in other locations. We can increase the potential options for the summary.

IV. Methodology

The suggested method for text-based online course summarizing With two algorithms, videos is implemented. TF-IDF and Gensim are them. A user's input for the model is the URL of a video. Our summary procedure has been put in place two algorithms are used to aid. Using a term's frequency and its inverse document frequency, the TF-IDF information retrieval algorithm can find relevant material. Topic modeling is covered by the NLP software package Gensim. Additionally, the approach allows the user to choose the amount of summary that is required relative to the original presentation, giving them more freedom.
The machine receives the YouTube video lecture's URL as input. The user must select the model to be used for summarization—TF-IDF or what percentage of the original video lecture should be summarized—as wellas the local disc location, or where the output summary has to be kept, in addition to the URL. The procedureis started by the video summarizer. The output summary file is automatically stored in the location specified by the user once the summarization procedure is complete. The Open Folder button on the GUI allows users to view the output by opening a local folder.

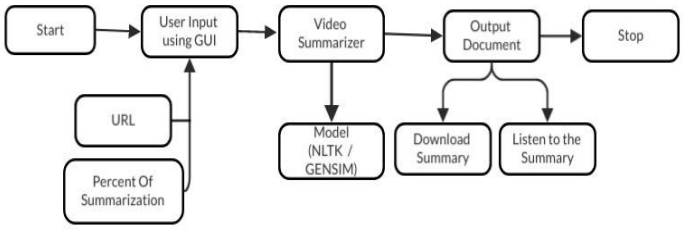

Fig. 1. System Architecture

A. AlgorithmsTF-IDF

The TF-IDF algorithm is the result of multiplying two algorithms together. Term Frequency (TF) and Inverse Document Frequency are two examples (IDF). TF is calculated as the number of times a word appears in a manuscript divided by the total number of words. IDF provides a definition of a word's rarity [8].

The actions taken to put this algorithm into practise are:

1. Sentence tokenization: In this case, we tokenize sentences rather than individual words. We then assign these sentences weight.
2. Calculating the frequency of the words in each phrase and creating a matrix of those frequencies. The end result is a key-value pair, where the key is a sentence and the value is a word frequency dictionary.
3. TF calculation and matrix generation: We determine the TF for every word in a paragraph.
4. Development of a table for documents per word: Here, we count the number of sentences that each word appears in. This is referred to as a documents per words matrix.
5. Determine IDF and produce a matrix: In this step, we determine IDF for every word in a paragraph.
6. Determine TF-IDF and produce a matrix: The following step is simple now that we have both matrices. We create a new matrix by multiplying the values from the two matrices, TF and IDF.
7. Score the following phrases: Using the TF- IDF score of words in a phrase, sentences are scored to provide weight to the paragraph.
8. Locate the boundary: At this step, we average the sentence score.
9. Create the summary: A sentence with a sentence score higher than the average should be chosen for a summary.

B. Text Rank Algorithm

Using the Python Gensim module, we put the text rank algorithm into practise. It is a free toolbox for topic and vector space modeling. It is carried out utilizing packets like Cython, SciPy, and NumPy. We make use of the gensim library's summarization.summarizer. The summary reads based on rankings of text phrases produced by a variant of the TextRank algorithm, a graph-based, all-purpose ranking algorithm for NLP. It isa method of automatic summarization. A vertex's relevance in a graph can be determined using graph-based ranking algorithms, which draw global information recursively from the entire graph.

The procedures taken to put this algorithm into practise are as follows:

1. Pre-processing the corpus: Moreover, stemming Eliminate all stop words and punctuation.
2. Create a network with the sentences from the corpus as the vertices.
3. In the network, the similarity between two vertices is indicated by the edge connecting them.
4. Apply the PageRank algorithm to the weighted graphs created after including the edge weights.
5. Add the sentences that correspond to the vertices with the greatest scores to the summary.
6. The number of vertices that will be used is decided based on the fraction value.

V. IMPLEMENTATION

A. The submit Module

The project's UI is located in the on-submit module. The user's information is received by the interface. The "URL," the "Summarization algorithm," and the necessary size of the summary expressed as a "Percentage" of the summary make up the input. The programme begins summarizing when the button labeled "Summarize" is clicked.

When the "summarize" button is clicked, this module sequentially summons the next two modules, which wewill see now. The "Download Summary" button is enabled in the user interface (UI) once the final module, the Summarizing module, has effectively completed execution. This will allow the user to download the summary.

B. The get-transcript module

The submit module uses the get-transcript module first. The URL is provided to this function as input. The correctness of the URL is then verified by this module. If the URL is wrong, a person is shown an error message. The Python module "youtube_transcript_api" is used to retrieve the Youtube transcript. If the URL is valid, the module extracts the "transcript" of the YouTube movie. Additionally, the module shows the transcript's legible text on the user interface, which we will use as the foundation for our summary.

C. The Summarising Module

The second module is the submit module's dispatch to the summarizer module. The user-selected summarization algorithm, the transcript produced by the get-transcript module, and the percentage that will determine the size of the summary as a percentage of the module are the three inputs to this module. Depending on the option of summarizing algorithm, i.e., "TFIDF- based Module" or "Gensim-based Module," the summarizer refers to either of the two modules, tfidf-based or genism-based. The transcript and the percentage satisfy the selected

algorithm. The summarizer module sends the summary back to the on-submit module after getting it from one of these modules so that it can be displayed on the application's user interface.

D. The programme for Text to Speech

The user interface's "Listen Summary" button becomes active following the effective conclusion of the summarization. For the viewer, this module reads out the summary. The user has the option to pause and restart speaking whenever is most convenient.

E. The downloading mechanism

After the summarization has been completed effectively, the "Download Button" becomes active. The personcan download the summary to their local computer by clicking the button.

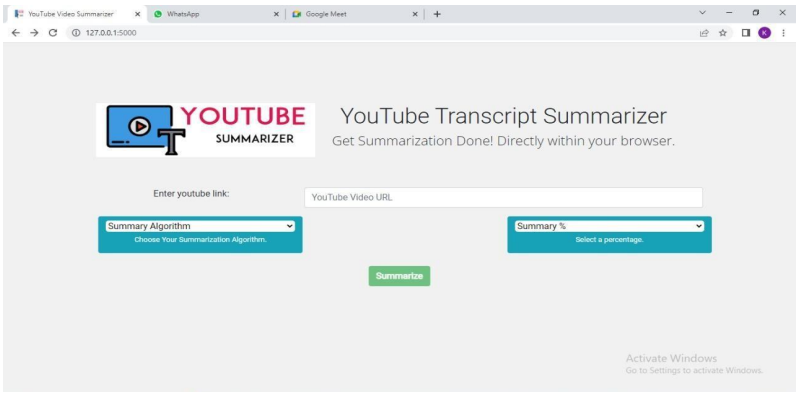

Fig. 2. Input Interface

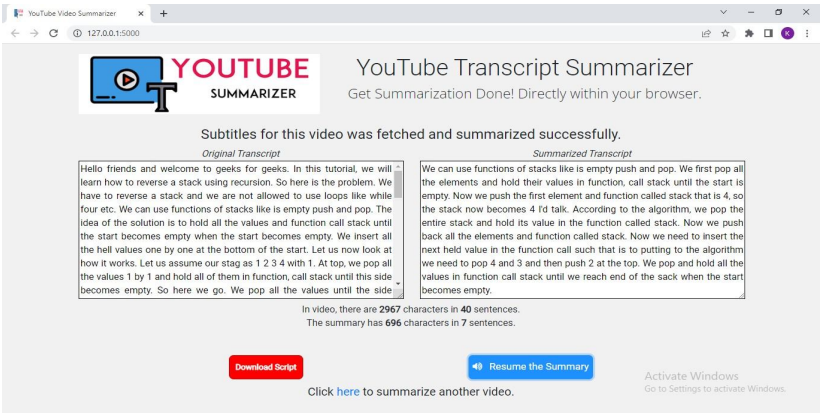

Fig. 3. Output Interface

VI. Challenges and Future Scope

Even if the accuracy and effectiveness of video transcript summarization have advanced significantly in recent years, there are still a number of issues that need to be resolved. The capacity to manage variances in the language used in video transcripts, including various accents, dialects, and languages, is one of the key issues. The capacity to manage the various lengths and complexity of video transcripts which can range from a few words to many pages of text presents another difficulty. Last but not least, techniques for assessing the quality of video transcript summaries need to be developed because they may not be effectively reflected by current metrics in practical applications.

VII. Conclusion

Nltk and Gensim algorithms were used to summarize video transcripts in this work, providing a technical description of the process. Using the use of TF-IDF, TextRank, and LSA, we have shown how to develop extractive and abstractive summarization methods and contrasted their effectiveness on a sample dataset.

According to the findings, LSA, TF-IDF, and TextRank were the three methods that performed the best overall. Although these libraries offer a wide variety of NLP tools and algorithms, using Nltk and Gensim makes it simple to perform video transcript summarizing. The efficiency and effectiveness of video content analysis might be considerably increased by using these techniques, which would make it possible for consumers to swiftly and simply comprehend the content of many different videos.

REFERENCES

[1] H. Gupta and M. Patel, "Method Of Text Summarization Using Lsa And Sentence Based Topic Modelling With Bert," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 511-517, doi: 10.1109/ICAIS50930.2021.9395976.

[2] Awasthi, K. Gupta, P. S. Bhogal, S. S. Anand and P. K.Soni, "Natural Language Processing (NLP) Text Summarization - A Survey," 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 1310-1317, doi: 10.1109/ICICT50816.2021.9358703.

[3] S. JUGRAN, A. KUMAR, B. S. TYAGI and V. ANAND, "Extractive Automatic Text Summarization using SpaCy in Python NLP," 2021International Conference on Advance Computing and Innova- tive Technologies in Engineering (ICACITE), 2021, pp. 582-585, doi: 10.1109/ICACITE51222.2021.9404712.

[4] Rahul, S. Adhikari and Monika, "NLP based Machine Learning Approaches for Text Summarization," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), 2020, pp.535-538, doi: 10.1109/ICCMC48092.2020.ICCMC-00099.

[5] Verma, Pradeepika Verma, Anshul, "Accountability of NLP Tools in Text Summarization for Indian Languages" Journal of scientific research, 2020, 64, 258-263. 10.37398/JSR.2020.640149.

[6] P. Batra, S. Chaudhary, K. Bhatt, S. Varshney and S. Verma,"A Review: Abstractive Text Summarization Techniques using NLP," 2020 International Conference on Advances in Computing, Communication Materials (ICACCM), 2020, pp. 23-28, doi: 10.1109/ICACCM50413.2020.9213079.

[7] W. H. Ong, K. G. Tay, C. C. Chew and A. Huong, "A Com- parative Study of Extractive Summary Algorithms Using Natural Language Processing," 2020 IEEE Student Conference on Research and Development (SCOReD), 2020, pp. 406-410, doi: 10.1109/SCOReD50371.2020.9251032.

[8] R. Boorugu and G. Ramesh, "A Survey on NLP based Text Summariza-tion for Summarizing Product Reviews," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA) 2020, pp. 352-356, doi: 10.1109/ICIRCA48905.2020.9183355.

[9] Supreetha D, Rajeshwari S B and Jagadish S Kallimani, "Abstractive Text Summarization Techniques", International Journal of Engineering, Science and Computing, Vol. 10, Issue No. 7, 2020

[10] Divya Santwani, Akash Bedi and Mohit Bahrani, "Textizer", International Journal of Engineering Research and Technology, ISSN: 2278- 0181, Vol. 9 Issue 07, July-2020.