

# YOUTUBE TRANSCRIPT SUMMARIZER

Dr.T.Bala Murali Krishna<sup>1</sup>, Sujitha Mannepalli<sup>2</sup>, Swapna Muthireddy<sup>3</sup>, Perni Sahithi Priya<sup>4</sup>, Swetha Vedantam<sup>5</sup>,  
Gowtham Yedida<sup>6</sup>

<sup>1</sup>Professor,<sup>2,3,4,5,6</sup> Department of Computer Science and Engineering, Dhanekula Institute of Engineering and  
Technology Ganguru, India

**Abstract**— This paper presents the development of an innovative video summarization system utilizing Natural Language Processing (NLP) and Machine Learning techniques. The exponential growth of video content on platforms like YouTube has posed a significant challenge in efficient content consumption. To address this challenge, we propose a YouTube transcript summarizer that generates concise and informative summaries of video transcripts, enabling users to grasp key content material without the need for complete viewing. Unlike images, where data extraction is feasible from a single frame, understanding the context of a video typically requires viewing the entire content. Our study seeks to alleviate this issue by reducing the transcript length while preserving its comprehensiveness. Leveraging NLP and Machine Learning algorithms, such as Logistic Regression, we extract transcripts from user-provided video links and summarize the content into a precise representation using Word2Vec and Logistic Regression. By distilling the essence of YouTube video transcripts while retaining their pivotal elements, our system offers users a more streamlined and effective way to consume video content.

**Keywords:** Transcript Summarizer, YouTube, NLP, Word2Vec, Logistic Regression, Summary.

## I. INTRODUCTION

The proliferation of online video has democratized the distribution of information, allowing individuals and organizations to share knowledge, express creativity, and connect with audiences around the world but the explosive growth of video data presents a formidable challenge users looking to derive meaningful insights from these massive databases With millions of hours of video uploaded, manual methods of analysing and summarizing video are not feasible done again. The amount of content gets overwhelming for users, and they often struggle to find relevant videos and prioritize them in the noise. In this context, automated summary creation tools represent an important step towards empowering users to better navigate and extract value from online video

The advent of natural language processing (NLP) and machine learning has changed the landscape of text mining, enabling computers to understand insights and extract insights from unstructured text When NLP techniques are applied to video transcriptions a, researchers and developers seek to automate the process of summarizing video content Increased Capacity This automated summary systems use speech analysis, statistical modelling, and machine learning algorithms a are used in combination to extract key information from video text and present it in a concise and coherent manner

Our paper contributes to this developing body of studies by using imparting the improvement of a YouTube Transcript Summarizer that leverages modern NLP techniques to routinely generate concise summaries of video transcripts. By integrating superior text evaluation and machine learning algorithms, our gadget offers a scalable and green answer for summarizing YouTube movies, permitting users to quickly draw close the primary content material of motion pictures and make informed choices about which movies to watch. By automating the summarization process, our system reduces the time and effort required for users to extract treasured insights from video content material, thereby improving the general user revel in and facilitating know-how acquisition within the digital age.

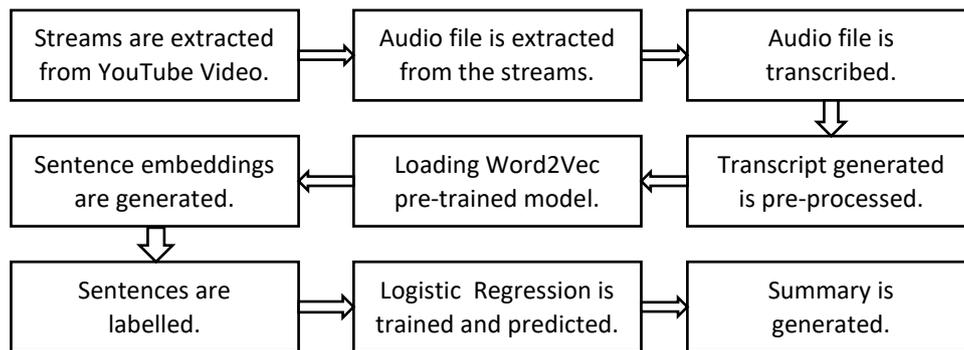


Figure 1 Process Flow Diagram.

## II. LITERATURE SURVEY

[1] In this paper, they proposed the usage of the NLP to summarize the YouTube Transcript. They used Hugging Face's Transformers library in Python to generate the summary of the original transcript and displayed the summary through a User Interface from python library called Tkinter.

[2] In this paper, they proposed a video summarizing system through Hugging Face Transformers. Their motivation is, to give user a quick glance of what is in the video with the shortened summary. They proposed this project using LSA which is a Natural Language Computing Algorithm which doesn't require Training Data.

[3] They proposed an automatic summarization method for YouTube video transcription text using term frequency-inverse document frequency (TF-IDF). By analysing the frequency of terms and their importance within the corpus, the system can identify and prioritize key information, enabling users to obtain a comprehensive overview of video content without the need for manual review.

[4] In this paper, abstractive summarization and deep neural networks are used to summarize video sequences. In comparison to prior approaches in the context, a joint model is proposed that enables users to discriminate between relevant and irrelevant information and produce superior outcomes.

[5] This paper proposed an application of LSTM-CNN based deep learning architectures for abstractive text summarization. They used LSTM-CNN based ATS framework (ATSDDL) that can construct new sentences by exploring more fine-grained fragments than sentences, namely, semantic phrases.

[6] They proposed this system using Flask framework and NLP. They generated summary by using the YouTube Subtitles and those subtitles are summarized through CNN model in their Flask Backend Server.

[7] They explored semantic hierarchical indexing for online video lessons using Natural Language Processing (NLP) techniques. By organizing video lessons into semantic hierarchies based on linguistic features and context, the system enables users to explore and access relevant content easily, thereby enhancing the learning experience in online educational platforms.

[8] In this paper, distil Bert is used to convert the words into numbers and pipelines were used to abstract the complex words and for displayed the summary through FLASK REST API which is the chrome extension.

[9] This paper gives a comprehensive survey on automatic text summarization techniques. The study provides an overview of various approaches to text summarization, including extractive and abstractive methods. The authors discuss the challenges and opportunities in automatic text summarization and analyse the strengths and limitations of existing algorithms.

[10] They proposed an abstraction-based multi-document summarization framework that can construct new sentences by exploring more fine-grained syntactic units than sentences, namely, noun/verb phrases. They employ integer linear optimization for conducting phrase selection and merging simultaneously to achieve the global optimal solution for a summary.

### III. PROPOSED SYSTEM

The proposed system for YouTube Transcript Summarizer is developed using Whisper API and the text is summarized using Word2Vec word embeddings and logistic regression.

#### A. PyTube Library and Whisper API:

A Python package called Pytube gives a simple interface for viewing and interacting with YouTube videos. Pytube makes easy to achieve functions like download videos, extract audio streams, get video metadata, and use the YouTube Data API. OpenAI's Whisper API gives users access to cutting-edge natural language processing (NLP) models for text creation and analysis. With the help of Whisper API, the audio is converted to corresponding text.

#### B. Word2Vec:

Words from a vocabulary are mapped to continuous vector representations in a high-dimensional space using the Word2Vec word embedding approach. We utilized Word2Vec to generate embeddings for words within the video transcripts. By training the Word2Vec model on a large corpus of text data, such as Google News data, we can map words to dense vector representations based on their contextual usage. This enables us to capture the meaning and semantic relationships between words, facilitating more accurate analysis and summarization. Once the Word2Vec model is trained, we can use the generated word embeddings as input features for our summarization algorithms. By considering the semantic similarity between words, our summarization process can better capture the essence of the content and produce more coherent summaries

#### C. Logistic Regression:

A basic statistical method that is frequently applied in machine learning for binary classification applications is logistic regression. Logistic regression determines the possibility that an instance belongs to a specific class, usually expressed as either 0 or 1. In contrast to linear regression, which predicts continuous numerical values, logistic regression predicts the probability. We used logistic Regression for the prediction of sentences whether important or unimportant.

### IV. IMPLEMENTATION

This system follows a systematic methodology encompassing multiple stages. The process begins with the retrieval of video content and the audio component of the video is transcribed into text format. The transcript undergoes preprocessing then sentence embeddings are generated using pre-trained Word2Vec models. The embeddings are then utilized to train a logistic regression model for sentence classification based on importance. Finally, important sentences are selected to construct a coherent summary of the video content.

#### A. Extracting transcript from YouTube video

The methodology begins with the installation of the PyTube library, a Python package designed for interacting with the YouTube API and downloading video content. The "Whisper" library, developed by OpenAI, is then installed to enable speech-to-text transcription. The procedure involves using PyTube to download the audio from a particular YouTube video, then loading the Whisper model for transcription. After the audio is transcribed, the text is saved to a text file as specified in figure 1 and printed to the console for additional examination.



Figure 2 Extracted Audio and Transcript files.

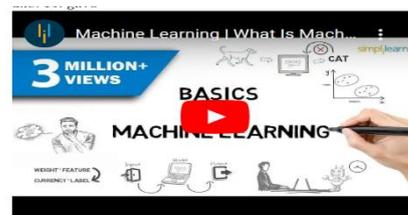


Figure 3 Video Used.

### B. Pre-processing the transcript

The next step is extracting pre-processed transcripts. Firstly, the dataset containing transcripts is loaded, which contains TEDx talk transcripts. Each transcript is then subjected to a preprocessing algorithm that includes tokenization, lowercasing, punctuation removal, and stop word elimination. Tokenization separates the transcripts into sentences, and lowercasing makes all characters lowercase to standardize the content. Non-essential symbols are removed from punctuation, and popular terms with limited semantic value are removed from stop words. The raw transcripts are converted into clean, structured text data through various preprocessing stages, making them ready for additional analysis.

### C. Loading Word2Vec model and generating sentence embeddings

The process starts with obtaining a pre-trained Word2Vec model, which is a popular word embedding method that performs well at capturing semantic relationships between words. The Gensim library is used to load the Word2Vec model, giving users access to high-quality word embeddings that have been acquired through a huge text corpus. Next, a custom function is built that averages the word embeddings of the constituent terms in each word to produce sentence embeddings. The pre-trained Word2Vec model's word embeddings are retrieved, input sentences are tokenized, and the average embedding for each sentence is calculated. Sentence embeddings that are produced capture the contextual information and semantic meaning of the relevant sentences, making tasks involving downstream analysis easier.

```
[[ [ 0.      0.      0.      ... 0.      0.
    [ 0.      ]
  [ 0.      0.      0.      ... 0.      0.
    [ 0.      ]
  [-0.20800781  0.03417969  0.02575684 ... 0.09570312 -0.046875
    0.23730469]
  ...
  [ 0.      0.      0.      ... 0.      0.
    [ 0.      ]
  [ 0.      ]
  [ 0.      ]
  [ 0.      0.      0.      ... 0.      0.
    [ 0.      ]]]
```

Figure 4 Word Embeddings for the transcript.

### D. Training the Logistic Regression

Then the definition of labels for sentences extracted from sentence embeddings is given. In this context, a simple labelling strategy is employed, where sentences exceeding a certain length threshold are labelled as 'important' (1), while shorter sentences are labelled as 'not important' (0). This labelling strategy is based on the intuition that longer sentences may contain more substantial information or convey key messages within the transcript context. Once the sentences are labelled, the dataset is split into training and testing sets using a standard train-test split approach. The sentence embeddings generated from pre-trained word embeddings serve as input features, while the corresponding labels represent the target variable for logistic regression modelling. The logistic regression model is trained on the training set to learn the underlying patterns and relationships between the sentence embeddings and their corresponding labels.

### E. Generating Summary

This methodology begins with the preprocessing of transcripts to tokenize sentences. Next, pre-trained word embeddings, specifically the Word2Vec model, are utilized to create dense vector representations of words, capturing semantic relationships and contextual information. These word embeddings serve as the basis for generating sentence embeddings, where each sentence is represented as a vector by averaging the embeddings of its constituent words. Following feature extraction, logistic regression classification is employed to classify sentences into 'important' and 'not important' categories based on their semantic content. The classification model is trained on labelled data, where 'important' sentences are identified using a predefined criterion such as length or other domain-specific characteristics. Once the classification model is trained, it is applied to new transcripts to predict the 'importance' of each sentence. 'Important' sentences are selected based on the model predictions, and a summary is generated by concatenating these selected sentences.

We know humans learn from their past experiences and machines follow instructions given by humans. So that's Paul. He either likes them or dislikes them. So here, tempo is on the x-axis, ranging from relaxed to fast, whereas intensity is on the y-axis, ranging from light to soaring. More the data, better the model, higher will be the accuracy. Suppose your friend gives you one million coins of three different currencies. Each coin has different weights. Here, your weight becomes the feature of coins. Based on weight of the new coin, your model will predict the currency. Suppose you have cricket data set of various players with their respective scores and the wickets taken. Hence, the learning with unlabelled data is unsupervised learning. You must determine whether the given scenarios use supervised or unsupervised learning. Also, the memory handling capabilities of computers have largely increased, which helps them to process such huge amount of data at hand without any delay.]

Figure 5 Generated Summary for the Video.

## V. RESULTS

A wide range of datasets including video transcripts from different domains were used to evaluate the YouTube Transcript Summarizer. Qualitative analysis demonstrated how the system might highlight important details and extract important insights to improve the user experience. The summary generated is displayed using FLASK API. Figure 6 shows the interface before giving the video link and figure 7 shows the interface after the summary generated. Both the transcript and summary are generated.

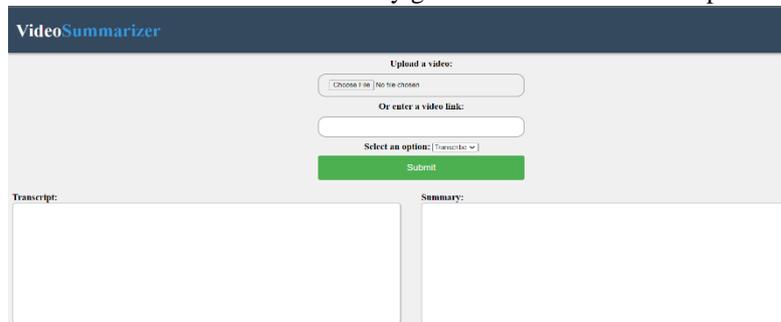


Figure 6 User input

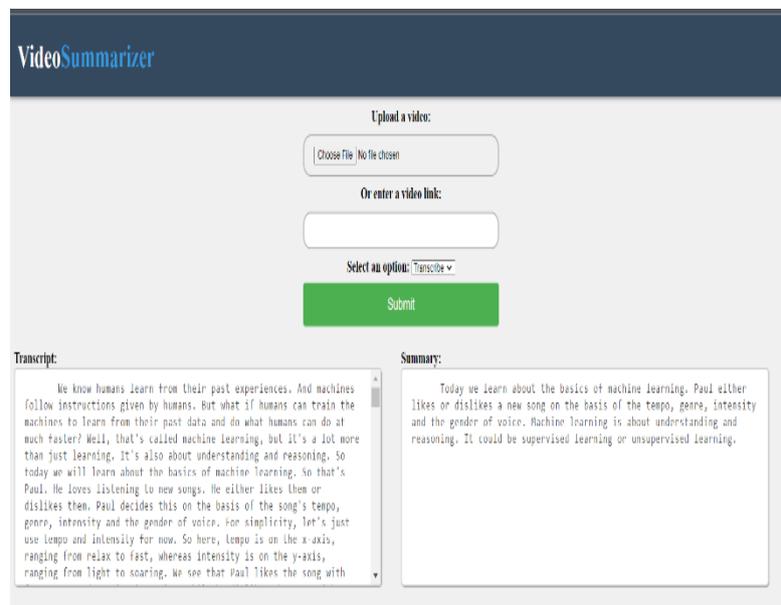


Figure 7 User output

## VI. CONCLUSION AND FUTURE WORK

In conclusion, this paper presents a comprehensive YouTube Transcript Summarizer leveraging advanced NLP techniques. As the need for effective information retrieval from online multimedia content grows, the system provides a workable option for summarizing YouTube video transcripts. Our technology makes it easier for users to browse the vast number of online platforms videos by automating the summarizing process. This facilitates knowledge acquisition and increases productivity. Subsequent investigations will concentrate on enhancing the summarization algorithms and investigating new features to augment the user experience. This study only summarizes the English language videos. So, this can be expanded by taking all languages videos into consideration from different industries.

## VII. REFERENCES

- [1] Prof. S. H. Chaflekar, Achal Bahadure, Hosanna Bramhapurikar, Ruchika Satpute, Rutuja Jumde, Sakshi A. Bakhare, Shivani Bhirange. "YouTube Transcript Summarizer using Natural Language Processing."
- [2] Atluri Naga Sai Sri Vybhavi, Laggiseti Valli Saroja, Jahnavi Duvvuru, Jayanag Bayana. "Video Transcript Summarizer." IEEE, 2022.
- [3] Rand Abdulwahid Albeer, Huda F. Al-Shahad, Hiba J. Aleqabie, Noor D. Al-shakarchy. "Automatic summarization of YouTube video transcription text using term frequency-inverse document frequency." Department of Computer Science, Faculty of Computer Science and Information Technology, University of Kerbala, Karbala, Iraq.
- [4] Aniq Dilawari, Muhammad Usman Ghani Khan. "ASoVS: Abstractive Summarization of Video Sequences."
- [5] Shengli Song, Haitao Huang, Tongxiao Ruan. "Abstractive text summarization using LSTM-CNN based deep learning."
- [6] P. Vijaya Kumari, M. Chenna Keshava, C. Narendra, P. Akanksha, K. Sravani. "YouTube Transcript Summarizer Using Flask and Nlp."
- [7] Marco Arazzi, Marco Ferretti, Antonino Nocera. "Semantic Hierarchical Indexing for Online Video Lessons Using Natural Language Processing."
- [8] Sourav Biswas, Atul Kumar Patel. "YouTube Transcript Summarizer to Summarize the content of YouTube."
- [9] El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2021). "Automatic Text Summarization: A Comprehensive Survey." *Expert Systems with Applications*, 165.
- [10] Bing L, Li P, Liao Y et al (2015) Abstractive multi-document summarization via phrase selection and merging[J]. arXiv preprint arXiv:1506.01597