2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI) | 979-8-3503-1590-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ACCAI58221.2023.10201219

Learning to Summarize YouTube Videos with Transformers: A Multi-Task Approach

R.Sudhan, D.R.Vedhaviyassh and G.Saranya*

Department of Networking and Communications, Faculty of Engineering and Technology, SRM Institute of Science & Technology, Kattankulathur,Tamilnadu– 603203, India

E-mail : sr6424@srmist.edu.in vd8264@srmist.edu.in saranyag3@srmist.edu.in
Corresponding author: G.Saranya (saranyag3@srmist.edu.in)

Abstract-The wide-range availability of online video content has made people consume more time to keep up with the amount of information available. In this paper, we present a system for summarizing YouTube videos that use NLP and machine learning techniques to retain the important details without any data loss. There are a few techniques for summarization on YouTube, the first approach is YouTube API which has been a popular method used for video summarization. However, it has certain limitations, such as limited accuracy, language barriers, and inability to summarize caption-less videos. To overcome these limitations, a new approach using transformers has been proposed. The proposed approach involves downloading the video' audio, converting it to WAV format, performing speech-to-text conversion using the Hugging Face Automatic Speech Recognition model, and then using transformers and pipeline for summarization. For evaluation we have used Rogue (Recall-oriented understudy for Gisting evaluation) which is a popular metric used for summarization tasks with an f1 score of 0.6. The approach has been tested on multiple videos with different video lengths and has shown good results in terms of accuracy and efficiency. The proposed approach can be used for a wide range of applications, including online education, video marketing, and false thumbnail videos.

Keywords - Transformers, Hugging Face, ASR (Automatic Speech Recognition)

I.INTRODUCTION

With over 122 million users per day, YouTube has become an important part of our day-to-day lives. YouTube offers a wide range of various video content, including short and long videos. Popular video types on YouTube include tutorials, lecture videos, product reviews, educational content and so on[1,2]. YouTube's popularity has increased rapidly which helped to raise many creators and social influencers who make a living from the platform, with some earning marque numbers in a year. Due to the rapid increase in users and content creators, the quality of video and standards is increasing day by day. In addition, the pandemic has greatly expanded the usage of online education. Even so, users may find it difficult to keep up with the content due to the platform's huge amount of data, especially those with limited time or resources. By automatically producing brief summaries of a video's key ideas, video summarization offers a viable answer to this problem.

Video summarization offers benefits beyond the educational domain and is widely applicable in fields such as journalism, law, and business, where it can be leveraged to extract key insights and data from large volumes of video footage. Several methods can be used to summarize a video, including using Python transcript API, which directs to the source video's caption and summarizes the transcript. Another approach involves obtaining the video's URL, downloading the audio, using the appropriate language's automatic speech recognition (ASR) technology, and segmenting the audio to obtain the transcript. The transcript offers an accurate representation of the video's content and can be analyzed to extract the most critical sentences and phrases. However, conventional methods for text summarization rely on handcrafted features and heuristics, which can limit their effectiveness. Recent advancements in deep learning, particularly transformer architecture, have shown remarkable results in various natural language processing tasks, including text summarization.

For summarization, the classic method followed is a recurrent neural network[3] which is a sequential data processing method. There are issues with long-term dependency sequences to overcome this LSTM (long short-term memory) method. This model is

Authorized licensed use limited to: KLE Technological University. Downloaded on September 06,2023 at 04:50:49 UTC from IEEE Xplore. Restrictions apply.

trained to predict the next word in a sequence given by the previous words. long short-term memory is particularly useful for speech-to-text, summarization because it can capture long-term dependencies in the sequence, allowing them to predict the probability of the next word based on the length of the context but there is a drawback is that they can be expensive and slow to train the model compared to other neural network architectures, such as feedforward networks or convolutional neural networks (CNNs). This is because LSTMs require a large number of parameters to be learned during training, and the recurrent connections can lead to issues such as vanishing or exploding gradients, which can make optimization difficult. On the other hand, Transformers are based on a self-attention mechanism that allows the network to attend to different parts of the input sequence when generating its output. This makes Transformers particularly effective for tasks that require the model to consider the entire input sequence at once, such as machine translation, summarization, and question-answering. Transformers can also handle input sequences of variable length, without the need for recurrent connections that make RNNs slower to train.

In this paper, we propose a YouTube video transcript summarizer using the transformer architecture [4,6]. Our goal is to use transformers to accurately and briefly summarize the video's content from its transcript. The reasons for creating a system for summarization are plenty such as it can save viewers time who don't have the luxury amount of time to watch an entire video, it will improve accessibility for those who are deaf or hard of hearing, different language videos can be seen by a wide audience, By briefly and clearly summarizing key ideas, it helps improve understanding. Moreover, it can provide a personalized service based on the user's interests and preferences. Our system will be able to process large volumes of text and generate high-quality summaries that capture the key information of the video and offer a valuable solution for people who want to learn and stay informed in a fast and efficient way.

II. Related Work:

Related work shows Natural Language Processing (NLP) and Machine Learning techniques are used for summarizing YouTube videos. Firstly, by retrieving transcripts with the help of YouTube API, then the transcripts are briefed by using the hugging face transformers model [1] which has a bundle of pre-trained models that are used in Deep Learning, Computer Vision related tasks. Our model at first converts the audio file format to .wav file for further computation, [2] uses compressed wav files to watermark an audio file by random sampling method. RNN-based methods are used to summarize the video in a sequence-to-sequence manner. Videos are divided into frame shots for summarization, it is important to summarize both audio and video information in parallel in order to summarize a video efficiently so, a hierarchical multimodal transformer is built by [3] to summarize this type of task. [4] uses the ML model to predict the transitions of a video with a novel method of fade-in, fade-out, and dissolve for comprehending video.

Supervised video summarization techniques embody the sequence learning method which makes it tough for a long-duration video to store all the parameters for the full-length sequence, so a U-shaped transformer is developed by [5] which makes it easy to summarize a long sequence video and reduces parameters. Long sequence summarization takes more memory and computation cost, [6] proposes an hourglass model a hierarchical transformer a state-of-the-art model on the ImageNet32 dataset. [7] summarizes a video on a supervised learning approach where a similar set of video datasets are summarized by having a summarized data of a subset of videos from the dataset.

Long videos are split into short segments and the word frequency to duration ratio is calculated for each segment, these scores are then ranked by and concatenated to form a meaningful summary [8]. [9] does an abstractive summarization of these video sequences which makes users distinguish between relevant and irrelevant videos needed for them.

DNNs are used to extract frame shots. For summarizing the video frames a content-based recommendation system is used by [10] which gives a score for the importance of the video segment. Video summarization is not only restricted to summarizing recorded video it can be done in real-time as well i.e., Live video. [11] proposes a real-time video summarization technique that can be used on mobile.

Authorized licensed use limited to: KLE Technological University. Downloaded on September 06,2023 at 04:50:49 UTC from IEEE Xplore. Restrictions apply.

Indexing of videos plays a vital role in displaying the right video users look for when they search through YouTube videos. To do this job [12] proposes an auto-indexing of YouTube videos by using similarity metrics on YouTube-generated auto-caption and transcript of the video. Furthermore, different video summarization techniques, algorithms, and their accuracy scores are compared in [13-15].

III. PROPOSED METHODOLOGY:

In this segment, we will see the approach used, methods taken to build the model, and the architecture diagram of the system. In this architecture diagram(Fig.2), the YouTube video's URL is fed into a video processing and feature extraction module, which extracts key features from the videos such as audio, visual, and semantic information. These features are then fed into a transformer-based summarization module, which generates a summary of the video content. Video Summarization is a diverse concept where in every generation new technology and algorithms is being implemented[13-15]. As a whole, the most implemented method is using YouTube API (Application Programming Interface. With the YouTube API, developers can access a wide range of collection of videos from the YouTube library including their titles, descriptions basically all the text formats but there are a few flaws in this method such as limited accuracy, language barriers, and the most important issue is if the video is caption less than that particular video cannot be retrieved and summarized. Here comes our method of summarization using transformers by getting real-time YouTube video URLs and summarizing them irrespective of caption or caption-less video

The model was developed using the following modules:

- Download the YouTube video and convert it to Wav format.
- Automatic Speech Recognition (ASR) was employed to convert the speech to text.
- Text summarization techniques were applied to generate a summary of the transcript.

We were able to create an effective and precise video summarizing model by using this modular approach. The model can extract important information from YouTube videos and display it in a clear and understandable way. Summarizing video footage can assist save time in a variety of applications, including educational films, news broadcasts, and marketing content. Fig.1 states the workflow of the system which is majorly divided into three segments that lead to generating the summarized text.

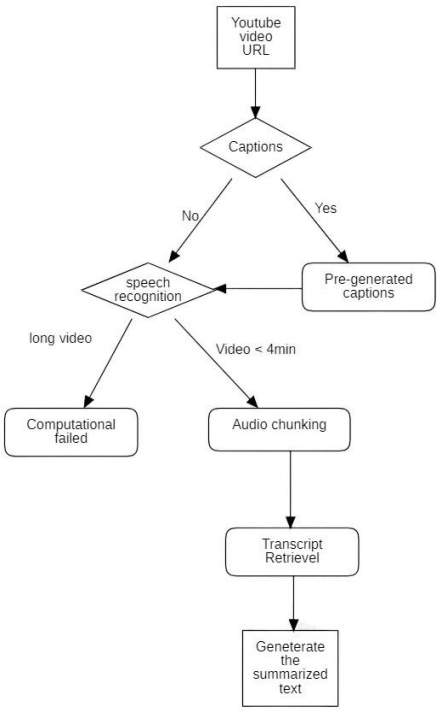

Fig.1. Proposed Workflow·

A. ACCESSING VIDEO:

For video summarization there are certain library should be implemented and we have used pytube library. Pytube is a python library used for downloading YouTube video streams in a simple and effective manner with different formats like mp4,3gp and various qualities. Additionally, there is an advantage of this library is it supports a YouTube playlist, which can be used to download multiple videos in a single run and helps to monitor download progress. With the help of this library YouTube videos can be downloaded using its URL as a first step Video's URL is assigned to a variable and using the filter option in python with certain constraints like only audio is true, the file type is mp4 the video' audio is downloaded in mp4 format. As the next step to proceed the mp4 format should be changed to .wav format which is (Waveform Audio File Format) is a popular and widely-used audio file format. It is a lossless format, which implies that no audio information is lost during compression,

Authorized licensed use limited to: KLE Technological University. Downloaded on September 06,2023 at 04:50:49 UTC from IEEE Xplore. Restrictions apply.

producing a sound of a high caliber. FFmpeg, a well-known and widely-used open-source program used for managing multimedia files, including audio, video, and photos, converts wav files. This command-line utility can convert between various file formats, compress and decompress audio and video streams, extract audio and video from video files, and carry out a variety of other operations on multimedia files. With a sampling rate of 16000 and save it as a .wav file

B. SPEECH RECOGNITION SYSTEM:

As a next step in this model is to import the hugging sound library which is an open-source library built upon the transformer's library. It gives access to pre-defined models and tools for fine-tuning the automatic speech recognition (ASR) and text to Speech application to make it simple and efficient. In this model, we used automatic speech recognition based on the English language to achieve clear audio from the wav file which is created. There are several benefits in the speech recognition model that are accessibility, user allows to perform multi-tasks simultaneously, with advancements in machine learning and deep learning ASR system is becoming increasingly accurate in recognizing speech and converting it into text. In our model, we used Fine-tuned XLSR-53 large model for speech recognition. After the speech recognition module is imported from the hugging sound library the model has been set to cuda (compute unified Device Architecture) this is because when the model is trained upon GPU it gives more accuracy and reduces the time complexity.

Even though the ASR model consists of many benefits, in our model, we found an issue with the length of the video, especially with a long video which is known as an out-of-memory error (OOM) which affects the model accuracy and retrieving time. To overcome this problem, we came up with a solution known as audio chunking. Audio chunking is a technique used to divide longer audio streams into smaller, more manageable segments or chunks. Audio chunking allows parallel processing which leads to faster time, small audio chunks can be easily stored and retrieved from the memory which improves accuracy. Basically, in simple terms, audio chunking is if a video consists of three minutes and our model is given a constraint of block length of 30 seconds then the three min video is divided into 6 wav files respectively by which we can retrieve audio without any data loss. Overall, audio chunking is a powerful technique for improving the performance and efficiency of audio processing systems, particularly those that deal with longer audio streams or real-time audio inputs. A problem is raised in the audio chunking when audio is divided it cuts the ending words into two parts and each part is stored in a different wav file to overcome this we have implemented a voice activity detector in our project. VAD is an open-source tool used to detect speech activity in an audio signal and it is designed to work with large audio files and real-time applications making it faster and more efficient. This VAD is implemented by importing a library called Librosa. Finally, after this process, the transcript of the audio file is retrieved from each wav file and ready to proceed with summarization.

C. TEXT SUMMARIZATION:

This is the important part of the model which is text summarization using transformers. Transformers is a type of deep learning model used for natural languages processing tasks such as translational, summarization, text-to-speech, and sentiment analysis. Transformers makes itself different from other models is relying on the self-attention mechanism, which is the latest mechanism before follows attention mechanism that allows the model to selectively focus on specific parts of the input sequence when processing each element and it works on a set of weights that are determined to each element in the input sequence while processing an element. The main difference between both mechanisms lies in the way the queries, keys, and values are calculated.

While the queries, keys, and values in self-attention all originate from the same input, which is essentially the output of the previous transformer's layer, whereas in the attention mechanism, they do not originate from the same input. Each word in the sentence, therefore, pays attention to every other word in the sentence as part of the self-attention process. In contrast, in the attention mechanism, the summary vector pays attention to the input sequence to produce the context vector. From the Transformers library, we are importing a pipeline which is a high-level interface for accessing pre-trained models on a given input. Basically, pipelines in transformers process a sequence of steps that are executed next to perform specific tasks. It takes input data passing through the more transformer models and showcases the output which is basically the input of the next pipeline. Pipelines are used in

Authorized licensed use limited to: KLE Technological University. Downloaded on September 06,2023 at 04:50:49 UTC from IEEE Xplore. Restrictions apply.

different tasks like speech recognition, summarization, computer vision, etc. After importing the pipeline, the full transcript is started to summarize by counting the transcript length and processed in a certain iteration with constraints for starting and ending sequence, minimum length, maximum length and finally summarized text in printed. According to our model, there are two types of summarization possible. The first approach is after the audio chunking process the transcript text from different wav files is first chunked and chunked transcript proceeds with summarization whereas instead of text chunking and summarization in this model every specific chunk file is summarized first and later all the summarized files are chunked which shows the good result as compared to the first approach.

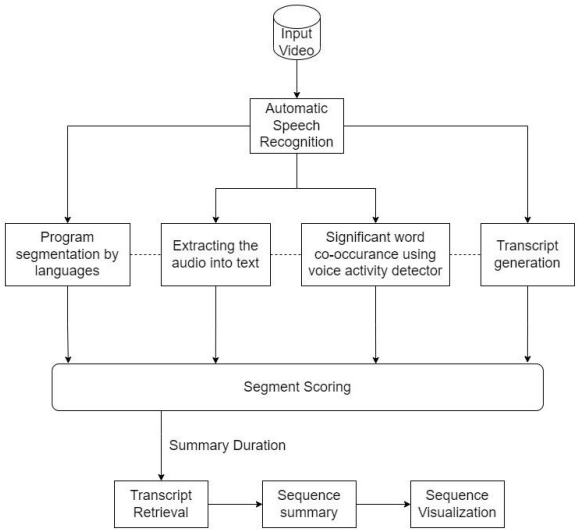

Fig.2. Architecture Diagram

IV. Experimental Results:

The accuracy of our model is evaluated by Rogue (Recall-Oriented Understudy for Gisting Evaluation) metrics. It is a popular metric best used for evaluating summarization-related tasks. This metric mainly distinguishes between the final summarized text of our model and a man-generated summary or we can call it a reference summary. So, we built a custom dataset with a set of videos along with their reference summaries, then the precision (1), recall(2), and F1 score(3) are calculated. These scores are calculated under three types of Rogue metrics namely Rogue-1 (comparing single word at a time), Rogue-2 (comparing two words simultaneously), and Rogue-L (comparing Longest common subsequence) for finding how our model's accuracy will behave over different metrics. Thus Table 1. Shows that the Hugging face ASR model shows a good accuracy score on the Rogue-L model with an F-score of greater than 0.5.

$$Precision = \frac{no\ of\ words\ in\ common}{no\ of\ words\ in\ summarized\ text} \quad (1)$$

$$Recall = \frac{no\ of\ words\ in\ common}{no\ of\ words\ in\ reference\ text} \quad (2)$$

$$F1\ Score = \frac{2(Precision * Recall)}{Precision + Recall} \quad (3)$$

Table 1. Evaluation metrics

Evaluation/Rogue metrics	Rogue -1	Rogue -2	Rogue - L
Precision	0.93	0.89	0.94
Recall	0.50	0.43	0.51
F-score	0.60	0.52	0.61

V. Conclusion:

We proposed a YouTube video summarization system using a hugging face transformer as earlier stages the model works on short-duration videos and the efficiency of the model is increased with respect to the local system's GPU. The evaluation of the system is done on a metric known as Rogue which is an Ultimate performance metric in natural language processing (NLP). Where, the precision, recall, and F1 scores are compared over different Rogue metrics which implies a high F1 score of 0.61 which is a good accuracy score. These methods are also in charge of eliminating unnecessary phrases. During the evaluation process, only English-language YouTube videos were used. The scope of this study could be increased by examining a vast array of videos in other fields and languages.

Authorized licensed use limited to: KLE Technological University. Downloaded on September 06,2023 at 04:50:49 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] Vybhavi, A.N.S.S., Saroja, L.V., Duvvuru, J. and Bayana, J., 2022, March. Video Transcript Summarizer. In 2022 International Mobile and Embedded Technology Conference (MECON) (pp. 461-465). IEEE.

[2] Anand, R., Shrivastava, G., Gupta, S., Peng, S.L. and Sindhwani, N., 2018. Audio watermarking with reduced number of random samples. In Handbook of Research on Network Forensics and Analysis Techniques (pp. 372-394). IGI Global.

[3] Zhao, B., Gong, M. and Li, X., 2022. Hierarchical multimodal transformer to summarize videos. Neurocomputing, 468, pp.360-369.

[4] Ren, W. and Zhu, Y., 2008, August. A video summarization approach based on machine learning. In 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (pp. 450-453). IEEE.

[5] Chen, Y., Guo, B., Shen, Y., Zhou, R., Lu, W., Wang, W., Wen, X. and Suo, X., 2022. Video summarization with u-shaped transformer. Applied Intelligence, pp.1-17.

[6] Nawrot, P., Tworkowski, S., Tyrolski, M., Kaiser, Ł., Wu, Y., Szegedy, C. and Michalewski, H., 2021. Hierarchical transformers are more efficient language models. arXiv preprint arXiv:2110.13711.

[7] Dhingra, H., Diwakar, S., Saranya, G. and Kumar, M., 2018. Fusion model for traffic sign detection, tracking and recognition. Journal of Electronic Systems Volume, 8(2), p.73.

[8] Taskiran, C.M., Amir, A., Ponceleon, D.B. and Delp III, E.J., 2001, December. Automated video summarization using speech transcripts. In Storage and Retrieval for Media Databases 2002 (Vol. 4676, pp. 371-382). SPIE.

[9] Dilawari, A. and Khan, M.U.G., 2019. ASoVS: abstractive summarization of video sequences. IEEE Access, 7, pp.29253-29263.

[10] Jiang, Y., Cui, K., Peng, B. and Xu, C., 2019. Comprehensive video understanding: Video summarization with content-based video recommender design. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (pp. 0-0).

[11] Choudhary, P., Munukutla, S.P., Rajesh, K.S. and Shukla, A.S., 2017, July. Real time video summarization on mobile platform. In 2017 IEEE International Conference on Multimedia and Expo (ICME) (pp. 1045-1050). IEEE.

[12] Jaiswal, S. and Misra, M., 2018, February. Automatic indexing of lecture videos using syntactic similarity measures. In 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 164-169). IEEE.

[13] Basak, J., Luthra, V. and Chaudhury, S., 2008, December. Video summarization with supervised learning. In 2008 19th International Conference on Pattern Recognition (pp. 1-4). IEEE.

[14] Geetha, G., Safa, M., Saranya, G. and Subburaj, R., 2017, May. An effective practices, strategies and technologies in the service industry to increase customer loyalty using map indicator. In 2017 International Conference on IoT and Application (ICIOT) (pp. 1-6). IEEE.

[15] Apostolidis, E., Adamantidou, E., Metsai, A.I., Mezaris, V. and Patras, I., 2021. Video summarization using deep neural networks: A survey. Proceedings of the IEEE, 109(11), pp.1838-1863.

[16] PRIYANKA, G. and MEENA, M.P., 2020. Survey and Evaluation on Video Summarization Techniques. Journal of Critical Reviews 7.8.

Authorized licensed use limited to: KLE Technological University. Downloaded on September 06,2023 at 04:50:49 UTC from IEEE Xplore. Restrictions apply.