International Journal of Future Generation Communication and Networking
Vol.14, No. 1, (2021), pp. 3830 - 3835

Video Summarization Using Subtitles

Abhinav Verma, Anuj Soni, Anuj Prajapati

Department of Computer Science and Engineering, Delhi Technological University

abhinavverma_2k17co09@dtu.ac.in

anujsoni_2k17co70@dtu.ac.in

anujprajapati_2k17co69@dtu.ac.in

Abstract

Generating compact and intuitive representations of video sequences that can be grasped easily by the users and that let the users quickly browse large amounts of videos is becoming one of the most important topics in content-based video processing. Such representations are called video summaries and these video summaries provide the user with important information about the content of the particular data while preserving the essential message. Here we propose a method to generate video summaries automatically using speech transcripts. We divide the full video into segments based on pause detection (the time duration for which there is a gap in speech) and derive a score for each segment, based on the frequencies of the words and bi-grams it contains. Then, a summary is generated by selecting the segments with the highest score to duration ratios while at the same time maximizing the coverage of the summary over the full program. We developed an experimental design and a user study to judge the quality of the generated video summaries.

Keywords— **text summarization, video summarization, speech transcripts, NLP, script vector**

I. INTRODUCTION

In this world of information technology and the internet, huge amounts of data are generated every day. Most of the data is in the unstructured form i.e not stored in a systematic manner. We know a lot of video content is being generated from innumerous multimedia sources which arise for the need of ways to develop analysis of data. The data includes videos, documents, homemade videos, audio-visual presentations and entertainment videos such as television shows, sitcoms and movies. In present modern times TV shows, movies and youtube videos constitute a major portion of the entertainment industry. Every year around 9,000 motion pictures are released worldwide spanning approximately 18,000 hours of videos[1]. India has the largest number of movies released in a year spanning around 2500 in a year. With the development of networking technology and digital videos and online streaming platforms like Netflix, Amazon Prime, Hotstar, etc, video content is increasing and people are demanding for live content . Such a large amount of data or information (content) implies a need for effective and efficient methodologies to organize and analyse the data.In the past, methods taken in relation to video summaries (i.e. video analysis and indexing) were developed using motion information and low-level design [2]. However, when working with video data, people are more interested in browsing semantic keywords than in low-level features [10]. Therefore, knowing how to extract semantic information from videos is a very important and necessary task. And the most common among these methods is mechanical reading and human annotation [3]. These methods are completely ineffective and complex. People need to train initial data and work with different types of videos. In addition, another strategy for working with video captions provides direct approval for the semantic portion of video content on the grounds that the semantic data is very well captured in the caption record. So it seems wise to use this fact to use semantic data, rather than build sophisticated video preparation for statistics.The first step is to extract the time stamps from the subtitle files associated with the video. These subtitle files

ISSN: 2233-7853 IJFGCN

can be found in many YouTube videos. Now the next step is to divide the text into sections [5]. Each part of a given script is converted into a vector presentation that will be used as a semantic index. Now, these vector-based indicators can be used for summarizing and retrieving.

II. RELATED WORK

A. KSUMM A Compressed Domain Technique for Video Summarization using Partial Decoding of Videos (2018)

This research was based on two different types of video techniques. One is an uncompressed or pixel based compression technique and the other is a compressed video summarization technique. In uncompressed video has to be fully decoded and after that it undergoes summarization but in compressed method there is no need of fully decoding video. Video can be summarized from the partial decoding of original video and then extracting the keyframes from the partially decoded video content. In this method the performed K-means clustering on forming classes based on extracting features.

Dataset :

VSUMM

This dataset contains 50 videos from OpenVideo. All videos are in MPEG format. It is not limited to a single gentre. It contains various videos from genres including educational, sports, history etc. This dataset contains many videos having many frames. Videos are 1 to 4 minutes long .

Result:

Performing the technique on this dataset, in this technique, there is reliance on change of scenes in content of video. After performing this technique 358 frames were reduced to 32 frames.

Summaries generated through this technique were given for rating to many individuals and average of all their scores was calculated and it overall had good experience.

B. Unsupervised object-level video summarization with online motion auto-encoder(2018)

It was an unsupervised method for video summarization. Earlier researchers did not consider semantic and motion information for generating video summaries. This did consider this fact by performing an object-level video summarization with an online motion autoencoder. It focuses on creating a storyboard type of summary. It is a pixel-based compression technique.

Dataset:

Usage of four datasets like Orangeville, CoSum, SumMe and TVSet.

SumMe consists of 25 videos, each annotated with at least 15 human summaries (390 in total).

Cosum: The dataset is collected from YouTube using 10 queries, in total 51 videos of 147m40s. We release the video URLs, preprocessed shot indices and annotations for reproducibility of our results.

The method they followed was based on LSTM encoder. LSTM is Long ShortTerm Memory. An LSTM autoencoder contains an encoder as well as a decoder. The encoder takes each feature

ISSN: 2233-7853 IJFGCN

repeatedly and feeds it to LSTM three layers and then the decoder decodes the vector generated by the encoder and obtains a sequence.

III. SEMANTIC VIDEO INDEXING

The process of identifying video sequences extracted from a script file into YouTube videos is done in three stages, namely extracting the script, separating the script and the script from the vector map..

A. Script Extraction

Regular YouTube videos have different script files and subtitles for each frame in the video sequence. There are 2 types of subtitle files.

a- When texts are recorded as strings in the text file.
b- where text is stored such as bitmap images are drawn on the screen while playing multimedia / video. Text-based subtitles are more flexible and smaller than bitmaps[8]. The advantage of using subtitles based on the text is that it is not only understandable to the person, but the user can also change the appearance of the displayed text. In addition to this there is easily accessible software (such as 'VOBSUB') [4] that converts bitmap images into text. Our main focus will therefore be on script extraction using subtitle-based files.

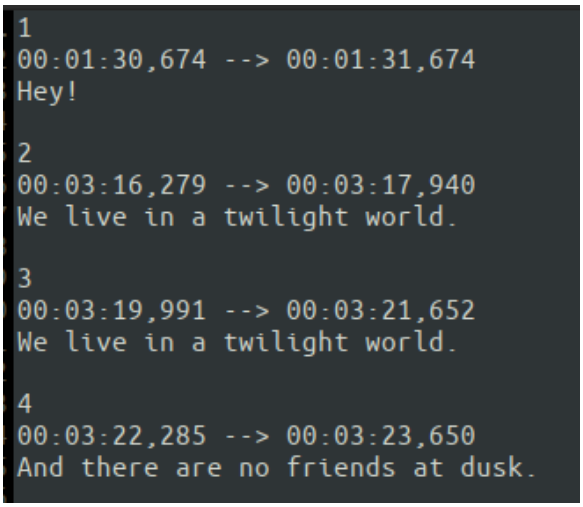

Fig. 1 Subtitle script of Tenet (2021)

Each text in the file contains a guide, start time and end of script for the video startup and synchronized text. The text file has been moved to script_elements}, where each script_elements} has the following 3 characters: start_time ',' end_time ', and' text '.

B. Script Partition

The goal of separating a script is to collect those scripts that share the same semantic thread that passes through them. It is quite clear that most of the time near Script_elements has been collected; some script_elements contain only a few words, and do not convey any semantic meaning itself [11]. This leads us to the question of how to determine which Script_elements should be grouped together to formulate the whole text.

ISSN: 2233-7853 IJFGCN

Here we will be using the time gap between Script_elements as a script demo. This time space is also called the Script_element gap and is defined as the difference between the previous 'end time' of Script_element and the current 'start time' of Script_element. Often, in the video, there may be some conversations or long stories that extend to many frames. In that case, the Script_element gap is too small. It also seems fair to assume that Script_elements with extended narratives will also have a "great interaction" between them. Thus, it appears that the Script_element gap is a useful tool for grouping the appropriate Script_element, and thus creating text fragmentation. In the following way, Script_elements split is done by closing the Script_element gap. Each section is called Script_segment.

C. Script Vector Representation

After the text is broken down into sections, the index is created for each part of the script. We use a space model for vector frequency inverse document (tfidf) vector space. This model is widely used to retrieve semantic reference details for script components [9]. Stopwords are those words that do not express the true feeling about a text i.e. what kind of meaning (good or bad) text conveys. So the first step here involves the removal of stopwords e.g. “Me”, “going”, “you” etc. Obstructing the process of finding root words in a collection of words that have the same meaning [12]. We used the Potter Stemming algorithm to find the stem for each word, e.g. the stem of the word “read” means “read”. All titles are stored in a dictionary, which is used to create a vector script for each component.

IV. PROPOSED SYSTEM ARCHITECTURE

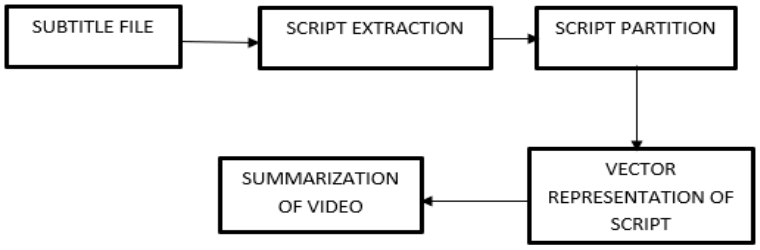

Fig. 2 Block diagram of video summarization process

V. RESULTS AND DISCUSSIONS

The proposed system was run and tested on different kinds of videos of various genres and of varying sizes. Different videos from a popular video-sharing platform were picked up and were made to run on the proposed system. The experimental results of the original video and the summarized video are calculated.

TABLE I
RESULT TABLE

S.No	Duration	Summarized
1.	11:20	4:59
2.	11:57	6:13

ISSN: 2233-7853 IJFGCN

3.	23:51	14:20
5.	10:36	3:45
6.	8:00	2:17
7.	14:54	4:42
8.	17:19	7:10
9.	8:36	3:23
10.	6:48	1:16
11.	4:45	2:55
12.	28:12	9:41

It is seen that there is a significant difference in the duration of the original video and summarized video. We can see the proposed system has saved upto 75% of the user's time. It also helps in presenting the video in a concise manner saving the user's time and energy.

We have gathered another set of results. In this experiment, we took a documentary on astronomy and space research. The system downloads the given video and subtitle of the corresponding video. The subtitle file is then converted into a text file. The initial subtitle file of the video contained 120 sentences. Script partition was performed by forming a vector representation of script and then later it was summarized to 25 sentences.

VI. Conclusion And Future Scope

In present times video summarization is an essential part of various video applications, including video ordering and retrieval. Brief and astutely generated video edited compositions would enable users to access a large amount of video content in an effective and compelling way. Here in this paper, a new approach is provided by us to solve the problem of video indexing. After analysing the subtitle files of the youtube videos, the semantic video index is extracted and represented by the vector-based model. The results we got after doing the semantic video indexing and retrieval demonstrate the effectiveness of the approach proposed by us. In future, we would consider other video retrieval methods and include the extracted video summary into MPEG-7 standard representation. The above-proposed algorithm takes into consideration only the text content of the programs in the generation of the summaries. If we need to improve the quality and effectiveness of the summaries, we need to take into account several other modalities, for example, the audio and image content could also be taken into consideration. The summarized videos can be used for storing important information which is hard to obtain for a rather lengthy video. Storing shorter videos helps in better memory management as many summarized videos containing relevant data can be stored. Although this system has shown a different method on summarizing videos, there are various features that can be upgraded as an extension of the system.

ISSN: 2233-7853 IJFGCN

Summarization of videos in different languages can be added in the system to extend the use of the system. Passive voice generation functionality can be upgraded in the system.

ACKNOWLEDGMENT

This project is supported by the Department of Computer Science and Engineering with Delhi Technological University.

REFERENCES

[1] H. D. Wactlar. *The challenges of continuous capture, contemporaneous analysis and customized summarization of video content*. CMU.
[2] S. Smoliar and H. Zhang. *Content-based video indexing and retrieval*. IEEE Multimedia, 1:62–72, 1994.
[3] C.-Y. Lin, B. L. Tseng, and J. R. Smith. *VideoAnnEx: IBM MPEG-7 annotation tool for multimedia indexing and concept learnin*g. In IEEE International Conference on Multimedia & Expo, Baltimore, USA, July 2003.
[4] Kansagara R, Thakore D and Josh M: A Study on Video Summarization Techniques(2014)
[5] M. M. Yeung and B.-L. Yeo, *"Video visualization for compact presentation and fast browsing of pictorial content*," IEEE Trans. Circuits
[6] K. Ratakonda, I. M. Sezan, and R. J. Crinon, "Hierarchical video summarization," in Proc. SPIE Conf. Visual Communications and Image Processing, San Jose, CA, Jan. 1999, vol. 3653, pp. 1531–1541.
[7] "*Video Summarization using Subtitles*", International Journal of Emerging Technologies and Innovative Research (www.jetir.org | UGC and issn Approved), ISSN:2349-5162
[8] M.W.Berry, S.T.Dumais, and G.W.O'Brien. *Using linear algebra for intelligent information retrieval*. SIAM Review, 37:301–328, 1995.
[9] H. Jing, R. Barzilay, K. McKeown, and M. Elhadad, "*Summarization, evaluation methods: experiments and analysis*," in Proc. AAAI Symp. Intelligent Summarization, Palo Alto, CA, March 23–25, 1998 [On-line]. Available: http://citeseer.nj.nec.com/jing98summarization.htmi
[10] A. M. Ferman and A. M. Tekalp, "*Two-stage hierarchical video summary extraction to match low-level user browsing preferences*," IEEETrans. Multimedia, vol. 5, no. 2, pp. 244–256, Jun. 2003.
[11] Y. Rubner, L. Guibas, and C. Tomasi, "*The earth mover's distance, multi-dimensional scaling, and colour-based image retrieval*," in Proc. ARPA Image Understanding Workshop, May 1997.
[12] B. Li, H. Pan, and I. Sezan, "*A general framework for sports video summarization with its application to soccer*," in Proc. IEEE Int. Conf. Acoustic, Speech and Signal Processing, Hong Kong, Apr. 6–10, 2003, pp. 169–172.

ISSN: 2233-7853 IJFGCN