

**Video/Text Summarization and Video Key-Framing for Educational
Media**

Capstone Project Proposal

Submitted by:

Harshit Vishwakarma (101917204)

Shruty (101917187)

Akshat Sharma (101903191)

Vernica Beohar (101903182)

BE Third Year- COPC/COE

CPG No. 168

Under the Mentorship of

Dr. Jasmeet Singh

Lecturer



Computer Science and Engineering Department

Thapar Institute of Engineering and Technology, Patiala





March 2022

TABLE OF CONTENTS

• Mentor Consent Form	3
• Project Overview	4
• Problem Statement	4-5
• Need Analysis	5-6
• Literature Survey	6-13
• Objectives	14
• Methodology	15
• Work Plan	15-16
• Project Outcomes & Individual Roles	16
• Course Subjects	17
• References	18-21

Mentor Consent Form

I hereby agree to be the mentor of the following Capstone Project Team

Project Title: Video/Text Summarization and Video Key-Framing for Educational Media		
Roll No	Name	Signatures
101917204	Harshit Vishwakarma	
101917187	Shruty	
101903191	Akshat Sharma	
101903182	Vernica Beohar	

NAME of Mentor:

Dr. Jasmeet Singh

SIGNATURE of Mentor:



1. Project Overview

With the age of the internet, there is an abundance of data in the form of videos, articles, research papers, and podcasts. Among hundreds of thousands of resources on a topic, it becomes increasingly complex to find quality resources. Media sharing platforms such as Youtube, Vimeo, Spotify, etc aim towards providing its users with a good experience by allowing them to easily search through a high volume of data comparable to finding a needle in a haystack.

Due to the unfortunate COVID pandemic in 2020, offline activities such as teaching and other forms of experiential learning opportunities were adversely affected. We leveraged the power of the internet and tried to substitute lectures, labs, workshops in the form of recorded videos or live streams, this created a boom in Ed-tech startups and the educational content produced by them grew exponentially.

We want to develop a solution to index these ever-increasing video content, especially for the educational domain with the help of Artificial Intelligence and improve the overall searching and browsing experience of the learner. We propose a video summarization and video indexing tool for educational videos.

2. Problem Statement

Extensive use of video streaming platforms without proper labeling makes the learning experience quite difficult, our project aims to resolve this problem by providing a concise and apt title for various video resources.

Apart from that, a lot of time is utilized by the user while navigating the whole video to arrive at the desired topic of learning. Hence our proposed idea of Smart Video Indexing

would help narrow down the search for content inside the video by partitioning the video into chunks of small topics being taught and saving precious time for users.

3. Need Analysis

Due to the advancement of web technologies and the popularity of video capture devices in the past few decades, the amount of video data has dramatically increased. On average, a person watches 6 hours 48 minutes of video per week [1] and the rate is even higher for the youth. The advent of the Covid-19 virus and subsequent lockdown forced educational institutions to shut down and carry on teaching in online mode.

During online teaching and assessments, the primary source of preparation were the recorded video lectures and textual materials. In most of the courses, videos are not labeled or tagged meaningfully, or coherently, or in a relevant manner for the topics they include and it becomes quite cumbersome to find the desired topic or concept quickly. For example, Lec01, Lec02, Lec03, ... and so on do not indicate the topics that are taught but only the order to be followed. Therefore, one has to go through all the videos to access the required material. There is a need for a solution that is not only limited to recorded lectures but is also desired in various video-sharing platforms or conferences. Our proposed solution is to design a smart combination of the Video Naming model, Video Summarization, and Video Key-Framing (Indexing) model.

Smart Video Naming would assist lecturers and teachers name their videos appropriately, would help content creators (YouTube, Vimeo, Instagram) to put appropriate captions for their videos. Some parts of Smart Video Naming can be used for other summarization tasks as well, like naming research papers, poems, or essays. Smart Video Key-Framing (Indexing) would help narrow down the search for content inside the video and save time for users. YouTube actively uses smart key-framing models to search for the videos that

contain the desired content rather than the title, description, or other metadata. It would also create a navigable index of very long videos (over 30 mins) which would again save users' time. Video Summarization would point out the important key points in the entire duration of the video and give a quick overview of the contents of the video.

The aim of our proposed solution is to speed up browsing and searching of a large collection of video data and achieve efficient access and representation of the video content. By reading the video title and using the key-frames, users can make quick decisions on the usefulness of the video.

4. Literature Survey

1. Speech to Text Methods

Technique	Description	Result
Artificial Neural Network Classifier (ANN) based Cuckoo Search Optimization	<p>ASR is built for a better interface of human and machine interaction. For the same, a three-step process is followed:</p> <ul style="list-style-type: none"> • Pre-processing of the speech signals is the most important part of speech recognition which is executed to remove avoidable waveforms of the signal. • Two kinds of acoustic features are extracted from the speech signal. They are 	ASR with Cuckoo Search Optimization technique is used for better communication, better recognition and to remove unwanted noise.

	<p>Mel Frequency Cep- strum Coefficients (MFCC) and Linear Predictive Coding coefficients (LPCC).</p> <ul style="list-style-type: none"> • Classification: In this, an artificial neural network is used as the classifier. The input layer consists of two inputs having two features extracted which are MFCC and LPCC features. These features are given as input in which networks get trained and it produces a corresponding output. 	
--	---	--

Table 4.1.1

2. Text Summarization

Technique	Description	Result
Graph-based approaches	Each sentence in the text is represented as a vertex and a graph is constructed around all the sentences, where the edges correspond to the interconnections between the sentences.	Classical approaches did not perform optimally, further advancements in techniques like LexRank [2] and TextRank[3] were used by Google to rank web page in their search engine.

	LexRank [2] and TextRank [3] are two such techniques.	
Machine learning-based approaches	Document summarization can be converted to a supervised or semi-supervised learning problem. In supervised learning approaches, hints or clues such as key-phrases, topic words, blacklist words, are used to label the sentences as positive or negative classes, or the sentences are manually tagged (which is not scalable). Once the labels are established, a binary classifier can be trained for obtaining the scores or summary likelihood scores pertaining to each sentence.	Classification-based approaches generalize well, however they are not efficient in extracting document-specific summaries. If the document level information is not provided then these approaches provide the same prediction irrespective of the document.
Abstractive summarization	Less prevalent in the literature than extractive ones. The two common abstraction techniques are structured and semantic [4, 5], both of which mostly are either graph/tree-based or ontology and rule (e.g. template) based.	It is much harder because it involves re-writing the sentences which if performed manually, is not scalable and requires natural language generation techniques.
Seq2Seq techniques-based approaches	Used to efficiently map the input sequences (description/document) to the output sequence (summary), however, they require large	It is found that Seq2Seq models currently work well for smaller document summaries (one-two lines of the document mapping to

	amounts of data. The model tends to learn the mapping between the input sequence and output sequence and generate more efficient summaries corresponding to the input document.	headlines/phrase representation) [6, 7]. Even though Seq2Seq models are providing benchmark results in Machine Translation and Speech Recognition tasks [8, 9, 10] they have not yet performed well for summarization tasks, dialog systems, and evaluation of dialog systems [11, 12, 13] and are facing many challenges (e.g. summarizing long documents).
--	---	--

Table 4.2.1

3. Corpus-based Similarity

Technique	Description	Result
Hyperspace Analogue to Language (HAL) [14,15]	A word-by-word matrix is formed with each matrix element is the strength of association between the word represented by the row and the word represented by the column. As the text is analyzed, a focus word is placed at the beginning of a ten word window that records which neighboring words are counted as co-occurring. Matrix values are accumulated by weighting the co-occurrence inversely proportional to the distance from the focus word; closer neighboring	Creates a semantic space from word co-occurrences. HAL also records word-ordering information by treating the cooccurrence differently based on whether the neighboring word appeared before or after the focus word.

	words are thought to reflect more of the focus word's semantics and so are weighted higher.	
Latent Semantic Analysis (LSA) [16]	Assumes that words that are close in meaning will occur in similar pieces of text. A matrix containing word counts per paragraph (rows represent unique words and columns represent each paragraph) is constructed from a large piece of text and a mathematical technique called singular value decomposition (SVD) is used to reduce the number of columns while preserving the similarity structure among rows. Words are then compared by taking the cosine of the angle between the two vectors formed by any two rows.	LSA can only partially capture polysemy (i.e., multiple meanings of a word) because each occurrence of a word is treated as having the same meaning due to the word being represented as a single point in space.
Explicit Semantic Analysis (ESA) [17]	A measure used to compute the semantic relatedness between two arbitrary texts. The semantic relatedness between two terms (or texts) is expressed by the cosine measure between the corresponding vectors.	The Wikipedia-Based technique represents terms (or texts) as high-dimensional vectors; each vector entry presents the TF-IDF weight between the term and one Wikipedia article.
Pointwise Mutual Information - Information Retrieval (PMI-IR) [18]	A method for computing the similarity between pairs of words, it uses AltaVista's Advanced Search query/ syntax to	In computational linguistics, PMI has been used for finding collocations and associations between

	calculate probabilities. The more often two words co-occur near each other on a web page, the higher is their PMI-IR similarity score.	words.
Second-order co-occurrence pointwise mutual information (SCO-PMI) [19,20]	A semantic similarity measure using pointwise mutual information to sort lists of important neighbor words of the two target words from a large corpus.	The advantage of using SOC-PMI is that it can calculate the similarity between two words that do not co occur frequently, because they co-occur with the same neighboring words.

Table 4.3.1

5. Objectives

- Generate automatic transcripts from videos.
- Build a summary using the transcript as a description of the video and assign meaningful title to the video
- Partition of the transcript into segments such that each segment holds a different topic/context from the adjacent partitions and construct a summary of each segment and give an appropriate title to each video segments.
- Build a Web interface to integrate the above objectives and add an option to upload the video files to process them through the AI pipeline.

6. Methodology

- **Data Collection and Preprocessing:** We will pick an educational domain-specific dataset and pre-process it before feeding it to our Summarisation Neural Network Model and Speech-to-Text generation Model.
- **Transcript Generation:** We will use Meta's wav2vec [21] model for generating transcripts from the audio sample of the video.
- **Summarization Pipeline:** We will build an abstractive Summarization model fine-tuned for educational videos, articles, and conferences.
- **Video Description and Title:** We will run our Summarization pipeline on the transcript generated by the wav2vec model and assign the output as video description. We will then perform sentence scoring and pick the most significant sentence as the title of the video.
- **Video Key-Framing:** From the ordered set of paragraphs generated from the transcript we will design an algorithm to partition the set such that each partition holds a topic/concept other than its adjacent partitions using Doc2Vec [22] vectorization of paragraphs.
- **Web Interfacing:** We will create REST based architecture of our AI server combining the above models deployed on the Cloud for guaranteed up-time.

7. Work Plan

- **Data Collection:** Firstly we will be collecting textual data from various openly available platforms like Kaggle, TVSum, WikiHow.
- **Data Preprocessing:** After data collection, we will start with data preprocessing and the creation of the data preprocessing pipeline.
- **Text Summarization Model:** The data will then be used to train text summarization models like BERT.
- **Paragraph clustering Model:** Partition of the set of paragraphs based on Doc2Vec vector space for array partition problem.
- **Joint Learning Model:** We will combine two models to summarize and index a given video.
- **Web-based integration:** During this phase, we will develop the UI of our website and integrate our AI model with a flask server.
- **Website Deployment:** This is the last phase of our project and in this, we will deploy our website.

S.No.	Activity	Month	February		March				April				May				June			
		Week	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Data Collection	Plan																		
		Actual																		
2	Data Pre-processing	Plan																		
		Actual																		
3	Summarization Model	Plan																		
		Actual																		
4	Paragraph Clustering Model	Plan																		
		Actual																		
5	Combining and Implementing Models	Plan																		
		Actual																		
6	Web based Integration	Plan																		
		Actual																		
7	Website Deployment	Plan																		
		Actual																		

Fig 7.1

8. Project Outcomes & Individual Roles

At the end of this project, we will be delivering a one-stop solution to suggest an appropriate video title, create meaningful and contextual partitions of educational videos, and generate appropriate titles for those partitions. This will not only help students to search a particular topic in a lengthy video but will also help them in skipping a particular section of the video easily. In order to achieve this task we will be dividing tasks as given below:

- **Akshat Sharma:** Data Collection, Data Preprocessing, Text Summarization Model, Website Deployment.
- **Harshit Vishwakarma:** Data Collection, Data Preprocessing, Paragraph Clustering Model, Website Deployment.
- **Shruty:** Data Collection, Data Preprocessing, Joint Learning Model, Web-based integration.
- **Vernica Beohar:** Data Collection, Data Preprocessing, Joint Learning Model, Web-based integration.

Student\Task	Data Collection	Data Pre-processing	Summarization Model	Paragraph Clustering Model	Combining and Implementing Models	Web based Integration	Website Deployment
Harshit Vishwakarma							
Akshat Sharma							
Shruty							
Vernica Beohar							

Fig 8.1

9. Course Subjects

S. NO.	SUBJECT	SUBJECT CODES
1	Machine Learning	UML 501
2	Software Engineering	UCS 503
3	Data Analytics	UCS 543
4	Natural Language Processing	UCS 664
5	Probability and Statistics	UCS 410

10. References

- [1] Joe Mandese ‘Time Spent Watching Online Video Expanding To 100 Minutes Daily, Ad Budgets Set To Follow’. MediaPost - 2019
- [2] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004.
- [3] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. *Association for Computational Linguistics*, 2004.
- [4] I. F. Moawad and M. Aref. Semantic graph reduction approach for abstractive text summarization. In *Computer Engineering & Systems (ICCES), 2012 Seventh International Conference on*, pages 132–138. IEEE, 2012.
- [5] A. M. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.
- [6] Text summarization with tensorow. <https://ai.googleblog.com/2016/08/text-summarization-with-tensorow.html>. Accessed: 2017-10-23.
- [7] S. Chopra, M. Auli, A. M. Rush, and S. Harvard. Abstractive sentence summarization with attentive recurrent neural networks. *Proceedings of NAACL-HLT16*, pages 93–98, 2016.
- [8] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

- [9] I. Sutskever, J. Martens, and G. E. Hinton. Generating text with recurrent neural networks. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), pages 1017–1024, 2011.
- [10] S. Wiseman and A. M. Rush. Sequence-to-sequence learning as beam-search optimization. arXiv preprint arXiv:1606.02960, 2016.
- [11] F. Guo, A. Metallinou, C. Khatri, et al. Topic-based evaluation for conversational bots. arXiv preprint arXiv:1801.03622, 2017
- [12] A. Ram, R. Prasad, C. Khatri, and A. Venkatesh. Conversational ai: The science behind the alexa prize. arXiv preprint arXiv:1801.03604, 2017.
- [13] Lund, K., Burgess, C. & Atchley, R. A. (1995). Semantic and associative priming in a high-dimensional semantic space. Cognitive Science Proceedings (LEA), 660-665.
- [14] Lund, K. & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical cooccurrence. Behavior Research Methods, Instruments & Computers, 28(2),203-208.
- [15] Landauer, T.K. & Dumais, S.T. (1997). A solution to plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge", Psychological Review, 104.
- [16] Gabrilovich E. & Markovitch, S. (2007). Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis, Proceedings of the 20th International Joint Conference on Artificial Intelligence, pages 6–12.
- [17] Turney, P. (2001). Mining the web for synonyms: PMIIR versus LSA on TOEFL. In Proceedings of the Twelfth European Conference on Machine Learning (ECML).
- [18] Islam, A. and Inkpen, D. (2008). Semantic text similarity using corpus-based word similarity and string similarity. ACM Trans. Knowl. Discov. Data 2, 2 (Jul. 2008), 1–25.
- [19] Islam, A. and Inkpen, D. (2006). Second Order Cooccurrence PMI for Determining the Semantic Similarity of Words, in Proceedings of the International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy, pp. 1033–1038.
- [20] Steffen Schneider, Alexei Baevski, Ronan Collobert, Michael Auli. WAV2VEC: UNSUPERVISED PRE-TRAINING FOR SPEECH RECOGNITION Sept. 2019
- [21] Andrew M. Dai, Christopher Olah, Quoc V. Le. Document Embedding with Paragraph Vectors. July 2015