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

Capstone Project Report

MID-SEMESTER EVALUATION

Submitted by:

Harshit Vishwakarma	**(101917204)**
Shruty	**(101917187)**
Akshat Sharma	**(101903191)**
Vernica Beohar	**(101903182)**

BE Fourth Year, COE

CPG No: 168

Under the Mentorship of

Dr. Jasmeet Singh

Assistant Professor

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology,

Patiala

July 2022

ABSTRACT

Due to the unfortunate pandemic, offline activities such as teaching and interaction were put to a halt. With the age of the Internet, we tried to mimic offline behaviour as online content such as recorded lectures and articles but soon it all piled up and searching for specific content in videos and other media formats became really difficult.

Given the problem, we aim to devise a smart indexing and summarization tool for educational videos so that learners can jump to whichever section they want and get a summary of the topic being taught without actually watching the entire length of the video.

DECLARATION

We hereby declare that the design principles and working prototype model of the project entitled **CrackleAI** is an authentic record of our own work carried out in the Computer Science and Engineering Department, TIET, Patiala, under the guidance of Dr. Jasmeet Singh.

Date: 24-08-2022

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

Counter Signed By:

Faculty Mentor:
Dr. Jasmeet Singh

Jasmeet Singh

____

Lecturer
CSED,
TIET, Patiala

ACKNOWLEDGEMENT

We would like to express our thanks to our mentor Dr. Jasmeet Singh. He has been of great help in our venture, and an indispensable resource of technical knowledge. He is truly an amazing mentor to have.

We are also thankful to Dr. Shalini Batra, Head, Computer Science and Engineering Department, the entire faculty and staff of the Computer Science and Engineering Department, and also our friends who devoted their valuable time and helped us in all possible ways towards successful completion of this project. We thank all those who have contributed either directly or indirectly to this project.

Lastly, we would also like to thank our families for their unyielding love and encouragement.

They always wanted the best for us and we admire their determination and sacrifice.

Date: 24/08/2022

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

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	**Elaboration**
JS	JavaScript
TF	Tensorflow
BERT	Bidirectional Encoder Representations from Transformers
LDA	Latent Dirichlet allocation
MB	Mega-Bytes
SRS	Software Requirement Specification
NLP	Natural Language Processing
COVID	Corona Virus
ANN	Artificial Neural Network
ASR	Automatic Speech Recognition System
TF-IDF	Term Frequency-Inverse Document Frequency
OAuth	Open Authentication

INTRODUCTION

1.1 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 contents, 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.

The entire system is divided into 3 major modules

- **Automatic Speech Recognition (ASR)**
- **Topic Segmentation**
- **Text Summarization**

ASR Module is responsible for extracting audio from video of various formats and divides the entire audio segment into smaller chunks to allow parallel processing of Speech Recognition Model.

Topic Segmentation is responsible for finding the number of optimal distinct topics the entire video should be divided into, and tag each document obtained from ASR a topic.

Text Summarization will be used to generate the name of the topic and suggest title of the video based on the ASR transcript generated.

1.2 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 and the rate is even higher for the youth. In July 2015, YouTube revealed that it receives over 400 hours of video content every single minute, which translates to 65.7 years' worth of content uploaded every day. Since then, we are experiencing an even stronger engagement of consumers with both online video platforms and devices. According to newer estimates, YouTube now receives 500 hours of video per minute; and YouTube is just one of the many video hosting platforms (e.g., Daily Motion, Vimeo), social networks (e.g., Facebook, Twitter, Instagram), and online repositories of media and news organizations that host large volumes of video content. So, how is it possible for someone to efficiently navigate within endless collections of videos, and find the video content that they are looking for? Given the plethora of video content on the Web, effective video summarization facilitates viewers' browsing of and navigation in large video collections, thus increasing viewers' engagement and content consumption.

The advent of the Covid-19 resulted in schools shut all across the world. Globally, over 1.2 billion children were out of the classroom. As a result, education has changed dramatically, with the distinctive rise of e-learning, whereby teaching is undertaken remotely and on digital platforms. Research suggests that online learning has been shown to increase retention of information, and take less time, meaning the changes corona virus have caused might be here to stay. During online teaching and assessments, the primary sources of preparation are the recorded video lectures and textual materials. In most of the courses, videos are not labelled 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 lecture 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 and Video Key-Framing (Indexing) model. 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.

1.3 Research Gaps

Although various techniques for speech summarisation have been proposed, there is still a considerable gap between the quality of automatic speech summarisation and manual summarisation by humans.

- Despite their potential usefulness, there has been little research on abstractive summarisation. This is partially due to the lack of suitable resources, corpora, and reference summaries in the speech domain.
- Another gap is the scarcity of extrinsic or task-based evaluations, which indicates that most studies focussed on traditional summarisation without paying attention to the usefulness for a specific task.
- Factors such as audio quality, structured speech, and number of speakers, affect the quality of the speech-to-text conversion, selection of methods and/or features, and the overall quality of summarisation. Lectures are less structured, speakers are usually not trained, and speaking styles and/or accents can vary widely.
- In terms of the speech features used, there was substantial variation, suggesting that the choice of feature types depends on the task, dataset, method applied, and language characteristics. In lecture summarisation, recent studies shifted from sentence ranking-based method to rhetorical information-based methods in a shallow or deep structure, due to their higher performance.
- Another observation was the lack of agreement between subjective and objective evaluations on the performance of lexical and acoustic features, for lecture summarisation, possibly due to the relatively large number of fillers included in a lecture.

- A potential issue is “gaming” where focus on discrete metrics increases score without an actual increase in readability or relevance. Several articles made comments relating to this: a single metric can be detrimental to the model quality.

1.4 Product Definition and Scope

In this new era, where tremendous information is available on the Internet, it is most important to provide the improved mechanism to extract the information quickly and efficiently. During online teaching and assessments, the primary source of preparation was the recorded video lectures and textual materials. In most of the courses, videos are not labelled or tagged meaningfully for the topics they include and it becomes quite cumbersome to find the desired concept quickly. For example, Lec01, Lec02, Lec03 ... and so on does 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. It also becomes very difficult to manually extract the summary of long videos or recorded lectures.

In order to solve the above problem, text summarization is very much necessary. Text summarization is the process of identifying the most important meaningful information in a document or set of related documents (here, audio transcripts) and compressing them into a shorter version preserving its overall meanings. Text Summarization methods can be classified into extractive and abstractive summarization. Abstractive Text Summarization is the task of generating a short and concise summary that captures the salient ideas of the source text. The generated summaries potentially contain new phrases and sentences that may not appear in the source text.

Our proposed solution is to design a smart combination of the Video Naming model and Video Key-Framing (Indexing) model. Smart Video Naming would assist teachers in naming their videos appropriately using text summarization, as well as help content creators (YouTube, Vimeo, and 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 minutes) which would again save users' time.

1.5 Assumption and Constraints

Assumptions

S. No.	Assumptions
1.	It is assumed that the lectures are being delivered in English language only.
2.	It is assumed that the user has Internet access and Internet browser
3.	It is assumed that the user has good bandwidth to upload the video to server seamlessly.
4.	It is assumed that the video has some speech in it.

Table 1.5.1

Constraints

S. No.	Constraints
1.	Size of video must not exceed 35MB
2.	The computational power available to fetch the data and train the model.
3.	The system currently works only for the English language.

Table 1.5.2

1.6 Standards

IEEE standards are followed for the complete development process of our projects.

Browser	Minimum Version	Restrictions
Chrome	71	Size of picture uploaded must not exceed the upper threshold
Firefox	64	Size of picture uploaded must not exceed the upper threshold
Microsoft Edge	42	Size of picture uploaded must not exceed the upper threshold
Opera	53	Size of picture uploaded must not exceed the upper threshold

Table 1.6.1

SRS building process completely followed from IEEE guide to build an SRS (830-1998-IEEE). CASE tools are used for constructing different software diagrams.

1.7 Approved 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 segment.
- 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.

1.8 Methodology

- **Data Collection and Pre-processing:** We will pick an educational domain-specific dataset from various openly available platforms like Kaggle, TV Sum, and Wiki How.
- **Data Pre-Processing:** We will pre-process the dataset and then create the data pre-processing pipeline. The data will then be used to train text summarization models like BERT.
- **Transcript Generation:** We will use Meta's wav2vec 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 vectorization of paragraphs. We will combine the text summarization and paragraph clustering models to summarize and index given video.
- **Web Interfacing:** We will create REST based architecture of our AI server combining the above models deployed on the Cloud for guaranteed up-time.

1.9 Project Outcomes and Deliverables

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.

Deliverables:

- A web application, which will have an interface to upload video.
- The output will be automatically displayed on the same page.
- Should provide accurate title in fewer than 15 words.
- Should output the transcript of speech detected.
- Should divide the transcript into segments of topic they are based on.

1.10 Novelty of Work

The main objective of this project 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.

Even though there exist many tools and many researchers have worked on text summarization, very few have used text summarization to create titles for educational videos. Our product will also provide Smart Indexing, which is absent in recorded lectures even if they have relevant titles.

Though there exists similar work, our institute currently does not have any facility to provide titles and summarizations for recorded lectures and videos. Therefore, we hope our project will prove to be useful to our institution and pave a smoother way for online education.

REQUIREMENT ANALYSIS

2.1 Literature Survey

2.1.1 Theory Associated with Problem Area

2.1.1.1 Natural Language Processing

NLP is a branch of artificial intelligence that deals with analysing, understanding, and generating the languages that humans use naturally in order to interface with computers in both written and spoken contexts using natural human languages instead of computer languages.

Some researchers combine NLP with deep learning where they "encode linguistic information" including POS (parts of speech) and NER (named entity recognition) tags as the lexical features as part of the neural encoder-decoder neural network. A step towards building more accurate summarization systems is to combine summarization techniques with knowledge bases and semantic-based or ontology-based summarizers. A trend that can be seen in the comparison matrix is a pivot away from NLP and more towards Deep Learning.

2.1.1.2 Deep Learning

Deep learning models have historically proven effective for machine translation and speech recognition. Now summarization is treated as a training and classification problem as well. Google, FaceBook, IBM, Microsoft and other companies are developing successful models based on Recurrent Neural Network (RNN), convolution neural network (CNN), as well as LSTM, NNLM, AMR, GRU, AE network models.

2.1.1.3 Text Summarization

Automatic text summarization is the task of producing a concise and fluent summary while preserving key information content and overall meaning. The goal of automatic text summarization is presenting the source text into a shorter version with semantics. The most important advantage of using a summary is that it reduces the reading time. There are broadly two different approaches that are used for text summarization:

- **Extractive Summarization:** An extractive summarization method consists of selecting important sentences, paragraphs etc. from the original document and concatenating them into shorter form.
- **Abstractive Summarization:** An Abstractive summarization is an understanding of the main concepts in a document and then expresses those concepts in clear natural language.

Abstraction is harder than extraction for humans and computers. Abstractive summarization can require “Prior knowledge, natural language processing and understanding”. In prior decades’ research abstractive solutions were rarer, but with advances in deep learning these systems are more commonplace.

2.1.1.4 Supervised and Unsupervised Learning

In machine learning, supervised learning uses datasets to train, whereas unsupervised learning does not (or uses latent features). Supervised and unsupervised approaches can be categorized into the following groups: “latent topic models” for unsupervised techniques, and “classification and regression” as the supervised techniques. Extractive summarization is almost always achieved. This is likely because these solutions are largely traditionally algorithmic or NLP-based. Abstractive techniques often (but not always) use supervised learning, since custom rubrics are still required for abstraction.

2.1.1.5 Metrics

Fitness of summarization needs to be measured. Measurement in itself can be challenging if the summarization involves abstractions. Automatically generated summaries have been evaluated using both subjective/qualitative metrics such as readability, coherence, usefulness, completeness, and objective/quantitative metrics such as ROUGE, Precision, Recall, F-measure, word accuracy, and Pyramid. A metric ideally works for different types of summaries or languages. The most popular measure is called “Rouge” which measures recall and how much the words appear in the reference. Another method is called “Bleu” which measures precision (how words match the reference summaries).

2.1.2 Existing Systems and Solutions

MicroFocus IDOL: It offers Unified text analytics, speech analytics and video analytics. The IDOL probabilistic model is capable of extracting meaning from human information in any language or format. It does not rely on an intimate knowledge of a language’s grammatical structure, but rather derives its understanding through the context of the words’ occurrence. This is particularly beneficial when analysing spoken or informal language that does not follow the linguistic rules of pure NLP systems. In addition, the ability to extract information from around 1000 file types—including audio and video—makes this sophisticated technique a very powerful tool that can add great value.

2.1.2.1 Speech to Text Methods

Technique	Description	Result
Artificial Neural Network Classier (ANN) based Cuckoo Search Optimization	ASR is built for a better interface of human and machine interaction. For the same, a three-step process is followed: • 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 Mel Frequency Cep-strum Coefficients (MFCC) and Linear Predictive Coding coefficients (LPCC). • 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	ASR with Cuckoo Search Optimization technique is used for better communication, better recognition and to remove unwanted noise.

	LPCC features. These features are given as input in which networks get trained and it produces a corresponding output.	

2.1.2.2 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. LexRank [2] and TextRank [3] are two such techniques.	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.
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 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.	It is found that Seq2Seq models currently work well for smaller document summaries (one-two lines of the document mapping to 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).

2.1.2.3 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 words are thought to reflect more of the focus word's semantics and so are weighted higher.	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.
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 calculate probabilities. The more often two words co-occur near each other on a web page, the	In computational linguistics, PMI has been used for finding collocations and associations between

	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.

2.1.3 Research Findings for Existing Literature

This literature review contrasted and synthesized recent developments in speech-to-text methods and automatic text summarization. Advances in abstractive summarizers and deep learning systems are observed. Extractive techniques continue to achieve top fitness scores, while a progressing metric trend for abstraction is closing the gap. Opportunity areas include improving unsupervised learning for diverse sources, blending NLP vs knowledge-based insights, and improving measurement metrics.

We have also discussed several challenges as well as surveys of the existing summarization methods. From these discussions, we have observed that many techniques suffer from various challenges, for example, the graph-based methods have imitation in data size, the clustering-based methods require prior knowledge of the number of clusters, etc. So, it is imperative that further research is required in this field to develop more effective methods for document summarization.

The wide variety of approaches, tasks and study designs limits our ability to genuinely compare the effectiveness of much of the published research. For this reason, future research should report in a more standardised way, and use standard public corpora to assist with performance comparisons.

2.1.4 Problem Identified

With the ever increasing rate of internet usage, educational content is usually in a video format. Quality lectures, tutorials and self-help videos are hard to find and need a proper way to search the content that the video cover.
Most of the time, lectures have meaningless title attached to them and doesn’t portray the contents inside the video.

With the help of modern-day transformers and advancements in AI, we can improve the overall user experience of searching the relevant educational videos.

2.1.5 Survey of Tools and Technologies Used

- **Client Side**: Cross-platform UI library ReactJs was implemented to make development process modular and created various components to achieve re-usability. Languages and Tools used were JavaScript and Axios.
- **Server Side**: The Server is implemented in Flask library for quick development and lightweight process. Languages and Technologies used in development are Python, Tensorflow, HuggingFace, PyTorch, Flask
- **Experiments and Analysis**:For testing and comparison of various models and evaluation metric Google Colab was used with WikiHow Dataset. Experimentation was done in Python.

2.2 Software Requirement Specification

2.2.1 Introduction

2.2.1.1 Purpose

The influence of digital videos on our everyday culture is undeniable. Online video sharing sites boast monthly audience numbers in the millions. Every other person nowadays depends on video resources either for learning or entertainment purpose.
In Educational sector, the importance of video lectures in life of students is undeniable. The corona virus pandemic has accelerated the transition to online content and off-campus learning. Most of the content was delivered online, enabling students to progress in their studies from the safety of their own home, or even from their home country.

On its face, this shift to online Video lectures is a good thing. Online or recorded lectures require less physical infrastructure and can be reused, reducing time and costs. The material can be accessed remotely, at any time of day — potentially making education accessible to more people. And we've all got better at exploiting digital technologies.

But at the same time most the video lectures lack at some areas. Some of the video lectures lack proper labelling or title whereas in some videos proper timestamp of various topics are missing making it difficult for a user to find and switch at required areas of interest. In long videos generally more than one topic is covered but all of the topics are not mentioned explicitly.

Our project covers all these drawbacks and works on video lectures to improve its usage. The main purpose of our project is to enhance the experience and scope of educational video lectures so as to provide ease and benefit to its audience as well as creators. This project will be a boon to many Ed Tech platforms.

2.2.1.2 Intended Audience and Reading Suggestions

The intended audience of our project is:

- **Teachers/ Professors**: The project will help teachers to make their content more efficient by providing all the necessary information and details which will make their videos more descriptive and self-explanatory.
- **Students**: The features added to the video lectures by our project will help students to look for required topic of interest. The appropriate titles, timestamps, explicit mentioning of topics and text summarization of video lectures will save a lot of user's time.
- **Other Content Creators**: It can also be used by other content creators on Video streaming platforms.

2.2.1.3 Project Scope

With loads of video lectures uploading every day on various video streaming and Ed Tech platforms our project will help the creators and its users in every way possible. The videos will be made self-explanatory and the users will find it easy to search and look for required topic without playing each and every video available and then finding the required section. This will further save time of content creators as they will be able to focus more on content rather than its labelling and detailing task as all of those tasks will be handled by our project.

This project can be further enhanced so as to use in variety of areas of research and development.

2.2.2 Overall Description

This segment depicts the highlights and different interfaces of the framework. It additionally manages the kind of correspondence the framework has with other outside elements.

2.2.2.1 Product Perspective

Our Project aims to ease and increase the efficiency of video lectures so as to help the teachers and students. The project will help teachers or creators to produce the appropriate tittle according to the content, produce the topic highlights or names explicitly and summarizing the audio lectures into text. This will help students as well as text summarization component can be used to extract hand written notes and find the appropriate and required time stamp.

Thus, this project will benefit the video lectures quality and its effective usage by students and teachers will tend to save lot of their crucial time.

2.2.2.2 Product Features

1. **Performance**: Our cutting-edge transformers are reliable and secure, which takes into consideration the user’s privacy and security. The model in its own might take a while to analyse the

video and display the result on the user dashboard. It is recommended to let the model load for a couple minutes to see the result.

2. **Maintainability**: The software is extremely easy to maintain. The source code of the whole project has been written following the best programming practices and conventions, making it easy for anyone to read, understand and maintain the code. The model will remain static with the same code once completed so it will experience no issues in the future.
3. **Security**: Only the users that are authenticated via email & password or Google/Twitter OAuth will be able to access the core features of the website. User interests and the posts they analyse will not be made public and will only be visible to the user itself. Apart from this, the user will have the option to completely delete their profile from the portal for increased customer satisfaction.
4. **Availability**: Our system is available for users at any point of time. Therefore, has a 100% uptime and 24*7 operation.

2.2.3 External Interface Requirements

2.2.3.1 User Interfaces

The web app will work in the browser itself, hence there are no external interface requirements apart from a device (mobile, tablet, laptop or PC) to access the web application.

2.2.3.2 Hardware Interfaces

Since our project is entirely based on software, it doesn’t require any hardware interfaces. The computer or the device they are working on must be fast enough so as to allow smooth functioning of the application.

2.2.3.3 Software Interfaces

Any Web browser that supports rendering of the following:

- HTML
- JavaScript
- CSS

2.2.4 Other Non-functional Requirements

2.2.4.1Performance Requirements

- The system should be available 365/24/7 days, the performance should not be degraded.
- The server must be capable of supporting all types of computers and shall provide no limit on how many devices are in the system.
- The server must lose no order under any circumstances.

2.2.4.2 Safety Requirements

- The system shall be capable of restoring itself to its previous state in the event of failure (e.g., a system crash or power loss).
- The security/safety of each user is provided with login id and password.

2.2.4.2 Security Requirements

- The system shall be able to do encryption and decryption of data for password which is given by the user for login.

2.3 Cost Analysis

The project is entirely software based and doesn't involve any hardware components. The libraries and datasets used are all free of cost only the deployment costs vary depending on the cloud services being used (AWS).

S.No.	Service	Hourly Rate
1.	General Cloud Compute	$0.0056
2.	Database/File System	$0.042
TOTAL	**---**	$0.0476

Table 2.3.1

Hourly rate of $0.0476 (*Rs. 3.80*) which amounts to monthly cost of *Rs. 2774.*

2.4 Risk Analysis

In order to do the risk analysis, we first need to identify risk and then do a qualitative analysis of each identified risk. Once this is done, we define a risk management plan to track breach and develop a service routine to maintain the normal functioning of development process.

S. No.	Hazards	Effects	Control Measures
1.	Poor Internet Connectivity	Slow video uploads to server and poor user experience	Ensure strong Internet connectivity
2.	Video contains no audio	No textual output and waste of resources	Provide guidelines to user to upload video containing speech
3.	Server downtime	Website becomes unresponsive	Rely on AWS up-time guarantee

Table 2.4.1

METHODOLOGY ADOPTED

3.1 Investigative Techniques

The proposed solution is to design completely novel system that index videos and give suitable title to those, it requires an experimental investigative technique which allows us to test hypothesis on various AI models for required tasks.

The machine learning models involved in tasks such as summarization, topic segmentation, and headline generation were tested on certain hypothesis and the best performing models were implemented.

The methodology above proves the point that Experimental Investigative Technique is the most apt methodology.

3.2 Proposed Solution

Through this project, we aim to develop a system capable of appropriately name an educational video and indexing the entire video with the topics being covered in it. We followed the below workflow to achieve the objectives of the Project:

1. Accept video from User in multiple formats, which allows portability in the system.
2. Extract audio from video and divide the audio into smaller chunks tagged with timeframe for leveraging parallel processing
3. Use Wav2Vec transformer model for automatic speech recognition system to convert audio chunks into English text
4. Use sentence scoring transformer for creating embedding vector of the transcript generated from automatic speech recognition
5. Partition the transcript into segments of same topic by using similarity metrics
6. Assign the segment a topic name using Headline generation transformer
7. Tag the video timeframe with topic name for each segment.
8. Generate summary of video using the transcript
9. Generate headline of video using the transcript

3.3 Work Breakdown Structure

The project has been broken down into several manageable modules, and at the end all of them will be interfaces as per the dataflow of the project. The Gantt chart below specifies the duration of each activity that sums up to the overall completion of the project.

Module 1: Identification, formation, and planning of project

Module 2: Study and analysis of Wav2Vec, BERT, T5, sentence-scoring

Module 3: Corpus Generation and Finalization

Module 4: Development of Models

Module 5: Model Optimization

Module 6: Accuracy Analysis

Module 7: Model Improvement

Module 8: Development of Dashboard, Deployment and final integration

Module 9: Result Evaluation

Module 10: Final Report

3.4 Tools and Technology Used

Software tools used by this system are:

- Python
- Tensorflow
- Flask
- HuggingFace
- PyTorch
- JavaScript
- React

DESIGN SPECIFICATIONS

4.1 System Architecture

S. No.	Diagram Name	Description
1.	**BLOCK DIAGRAM**	Block diagrams show a high-level view of the product under development and their interaction with different components including the sensors, actuators and servers.
2.	**SEQUENCE DIAGRAM**	A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram.
3.	**USE CASE DIAGRAM**	A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. Note: **-** The logout/login use case use same as normal/master mode.
4.	**ACTIVITY DIAGRAM**	Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.
5.	**DATA FLOW DIAGRAM**	A data-flow diagram (DFD) is a way of representing a flow of a data of a process or a system (usually an information system).

6.	**ENTITY RELATIONSHIP DIAGRAM**	Entity Relationship (ER) model is a high-level conceptual data model diagram. ER modelling helps you to analyse data requirements systematically to produce a well-designed database. The Entity-Relation model represents real-world entities and the relationship between them. It is considered a best practice to complete ER modelling before implementing your database.
7.	**GANTT CHART DIAGRAM**	A Gantt chart is a type of bar chart that illustrates a project schedule, named after its inventor, Henry Gantt, who designed such a chart around the years 1910–1915. Modern Gantt charts also show the dependency relationships between activities and current schedule status
8.	**CLASS DIAGRAM**	Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.
9.	**STATE DIAGRAM**	State Diagram is a diagram that depicts the different states the project has and their transitions on receiving particular inputs. It is a model that depicts the functionality of the system.
10.	**COMPONENT DIAGRAM**	Component Diagram is a pictorial representation of various components that the product has and it also highlights the various dependencies between the components.

Table 4.1.1

4.2 Design Level Diagrams

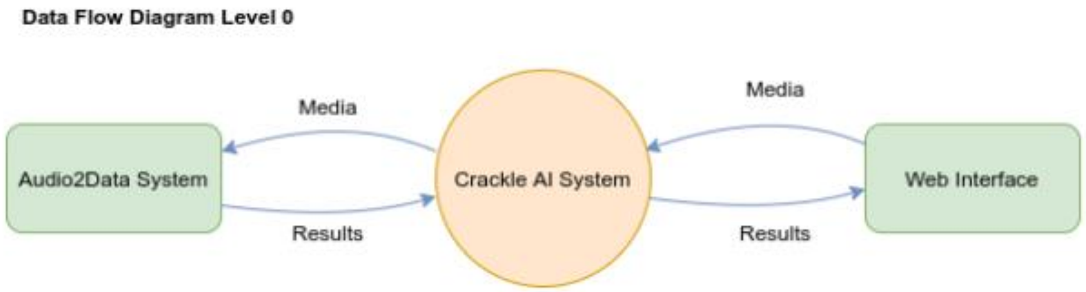


Fig 4.2.1, DFD Level 0

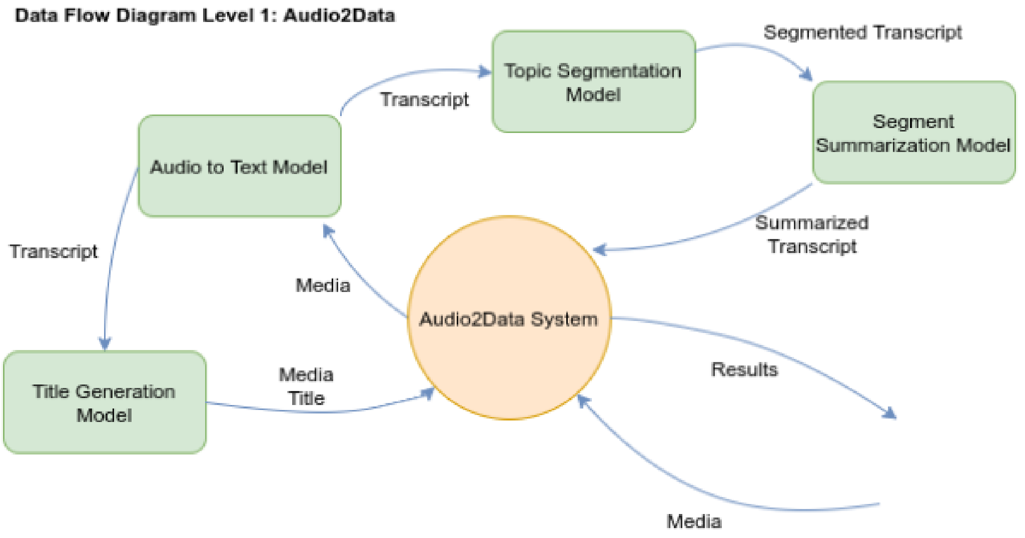


Fig 4.2.2, DFD Level 1

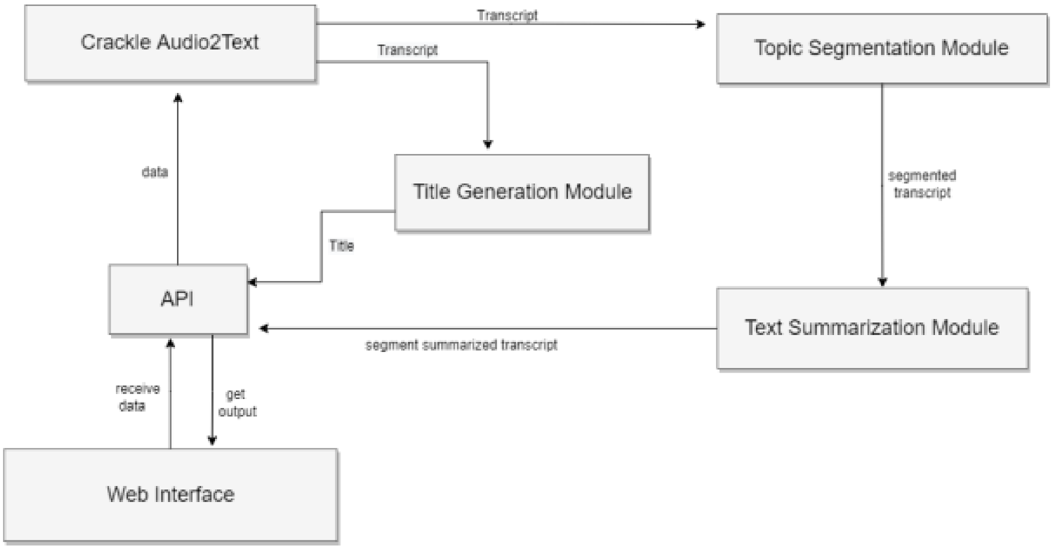


Fig 4.2.3, Block Diagram

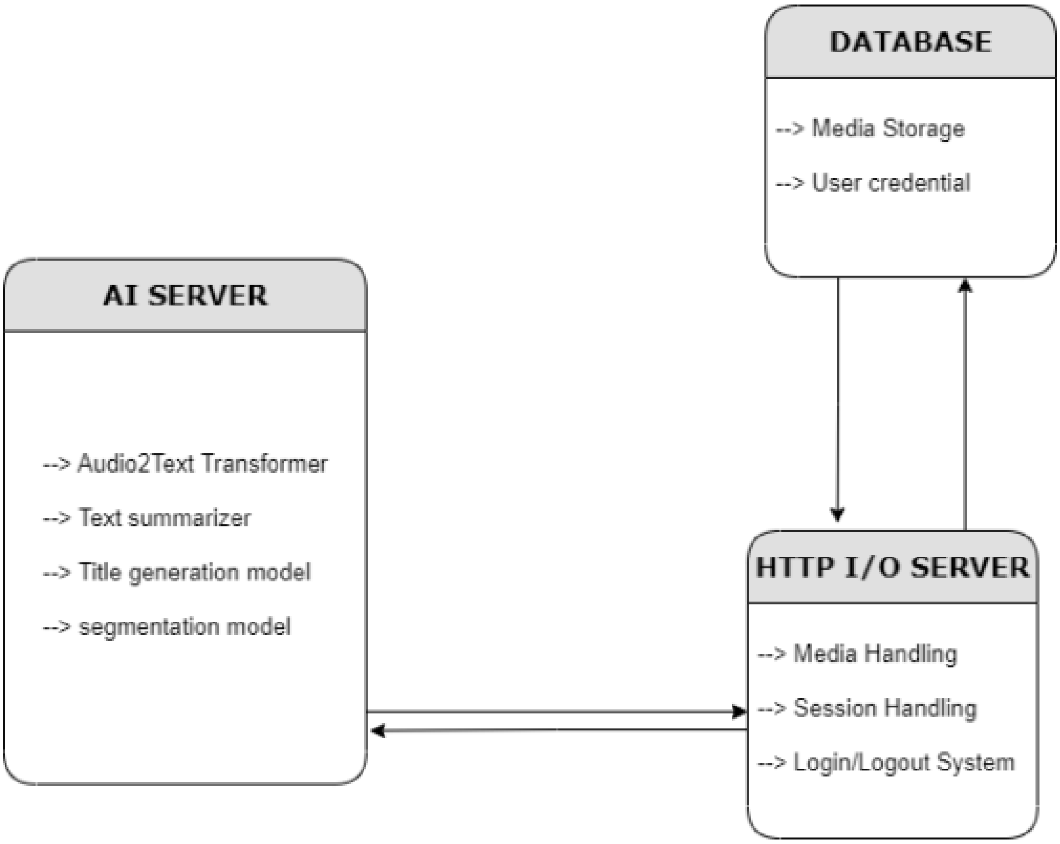


Fig 4.2.4, Component Diagram

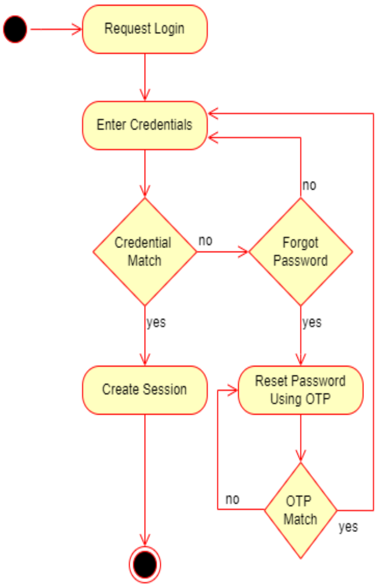


Fig 4.2.5, Activity Diagram (Login)

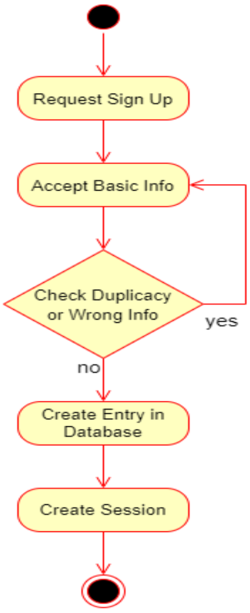


Fig 4.2.6, Activity Diagram (Sign Up)

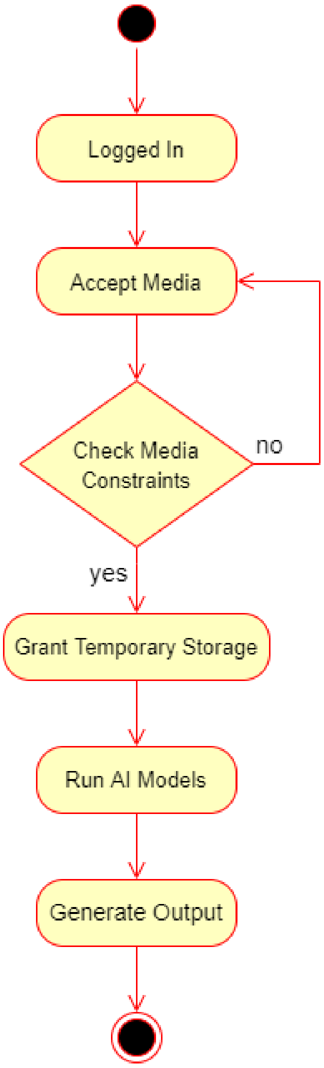


Fig 4.2.7, Activity Diagram (Summarization Service)

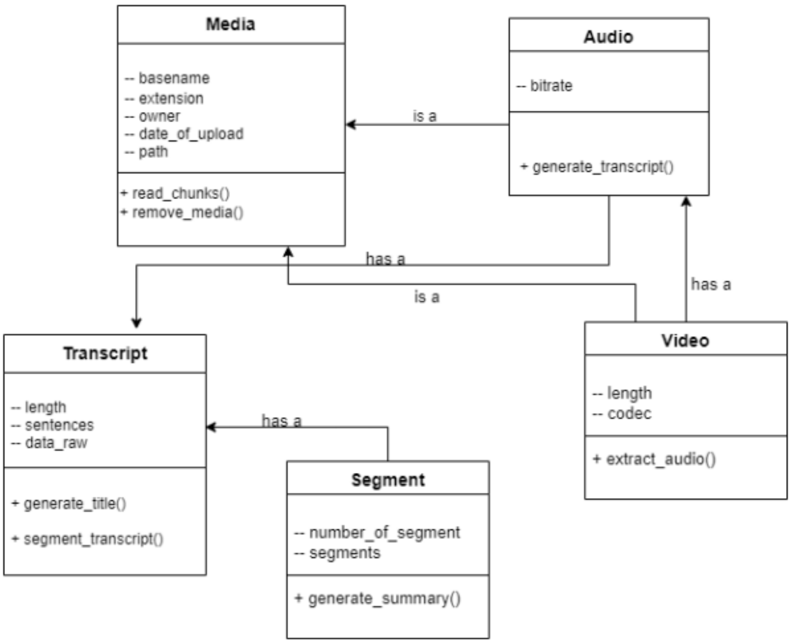


Fig 4.2.8, Class Diagram

4.3 User Interface Diagrams

4.3.1 Login Use case

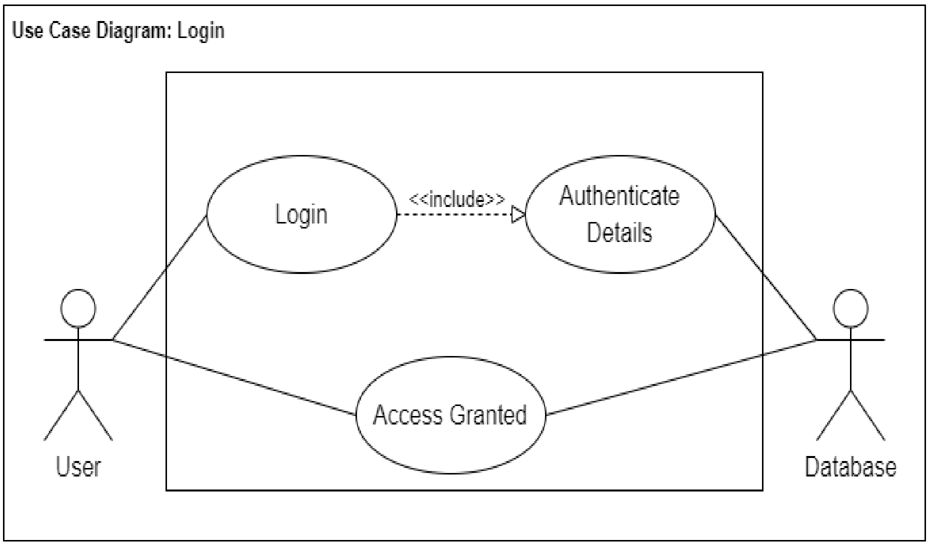


Fig 4.3.1.1, Login Use Case

1. Use Case ID	1
2. Use Case Title	Login
3. Actors	User, Database
4. Purpose: To allow the user to log in to their account	
5. Description: Authenticate user details to grant them access to database	
6. Pre-conditions: 1. Internet Access 2. Have an existing account in the database	
7. Task Sequence: 1. User opens the website 2. User enters details/credentials 3. User clicks on the login button 4. User is logged in	
8. Alternate Flow: No alternate scenario	
9. Post-conditions: 1. User is logged in to their account	
Modification History: 10-June-2022	
Author: Harshit Vishwakarma, Vernica Beohar, Shruty, Akshat Sharma	

Table 4.3.1.1 Login Template

4.3.2 Sign Up Use case

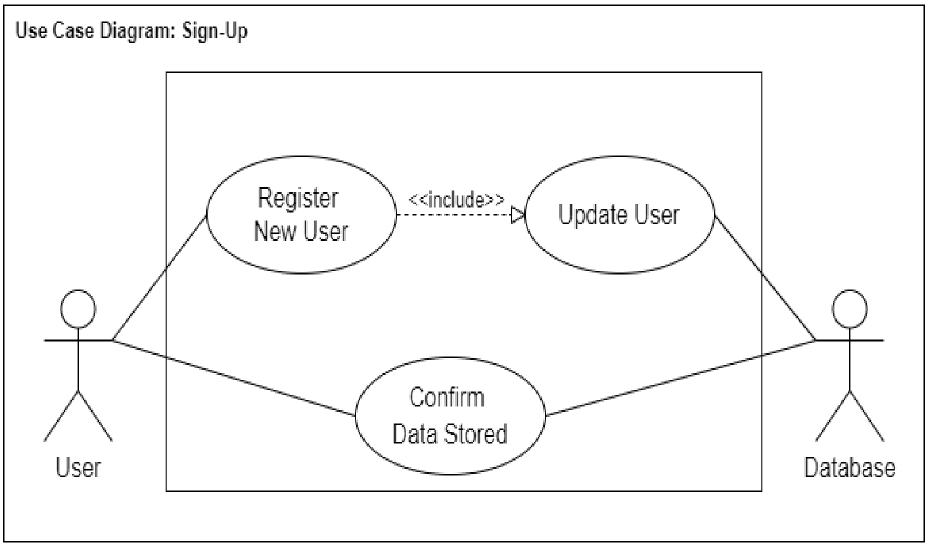


Fig 4.3.2.1 Sign Up Use case

1. Use Case ID	2
2. Use Case Title	Sign Up
3. Actors	User, Database
4. Purpose: To allow the user to sign up and create an account	
5. Description: Update user details and store them in the database	
6. Pre-conditions: 1. Internet Access	
7. Task Sequence: 1. User opens the website 2. User enters details/credentials 3. User clicks on the sign-up button 4. User is registered in the database	
8. Alternate Flow: No alternate scenario	
9. Post-conditions: 1. User's account is created and stored in the database	
Modification History: 10-June-2022	
Author: Harshit Vishwakarma, Vernica Beohar, Shruty, Akshat Sharma	

Table 4.3.2.1 Sign Up Template

4.3.3 Summary Service Use case

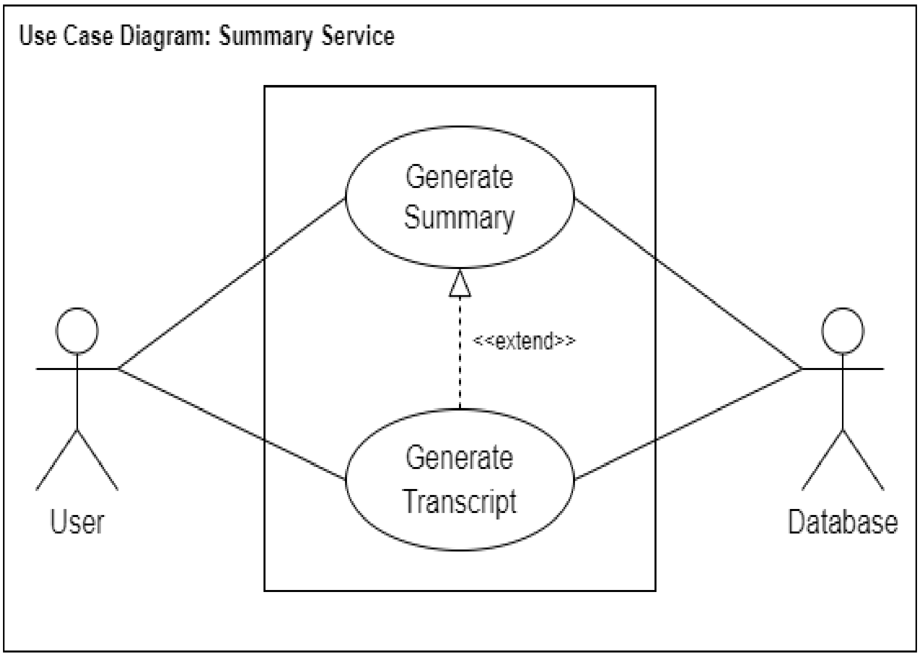


Fig 4.3.3.1 Summary Service Use case

1. Use Case ID	3
2. Use Case Title	Summary Service
3. Actors	User, Database
4. Purpose: To provide an apt summarized title for the video uploaded	
5. Description: Generate audio transcript and generate its summary to create title	
6. Pre-conditions: 1. Internet Access 2. Have an existing account in the database	
7. Task Sequence: 1. User opens the website 2. User enters details/credentials 3. User uploads a video 4. A summary is generated for the video transcript	
8. Alternate Flow: No alternate scenario	
9. Post-conditions: 1. An apt title for the video is shown to the user	
Modification History: 10-June-2022	
Author: Harshit Vishwakarma, Vernica Beohar, Shruty, Akshat Sharma	

Table 4.3.3.1 Summary Service Template

4.4 User Interface

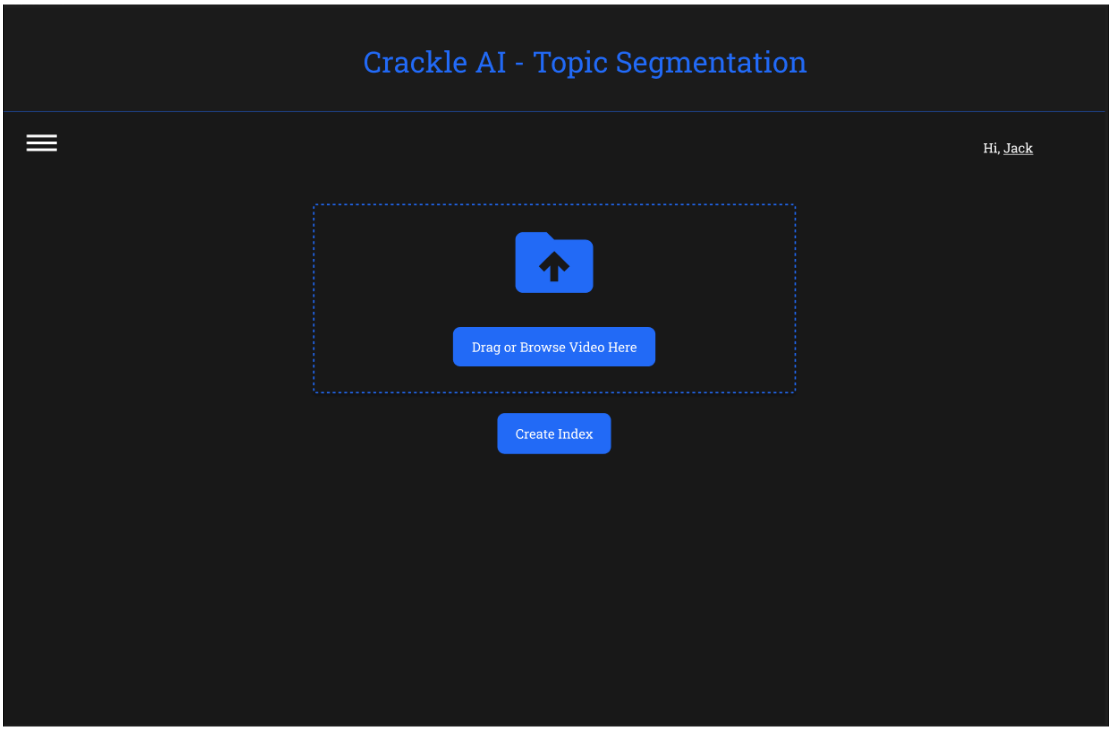
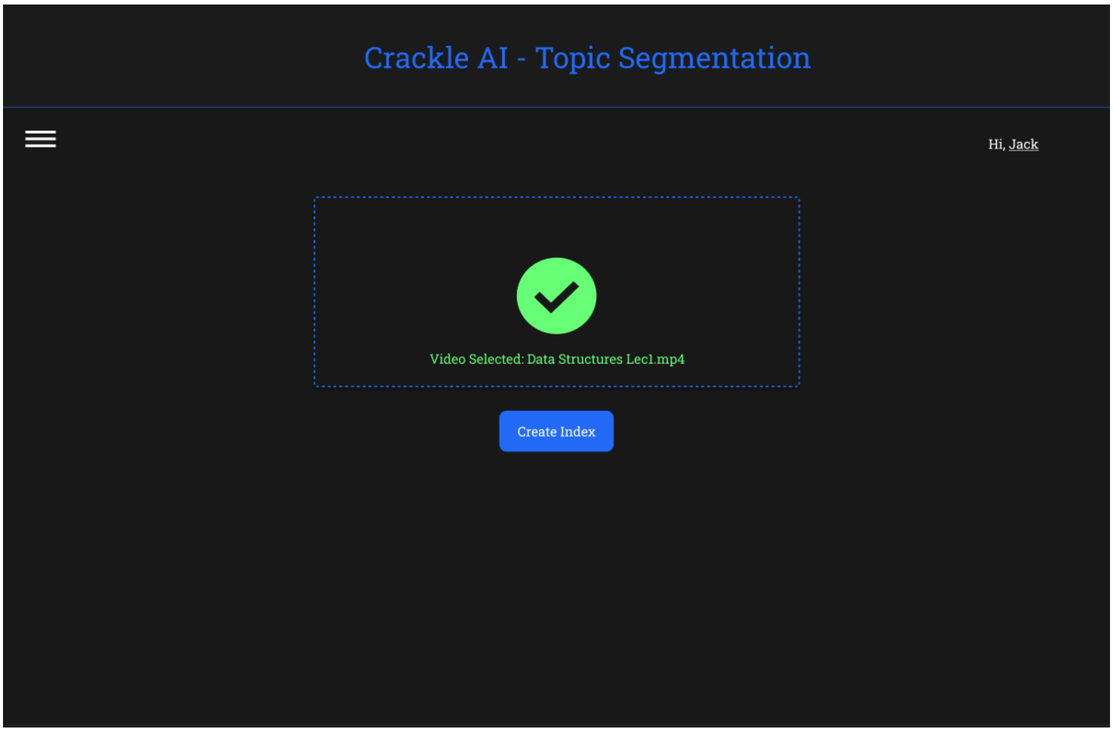
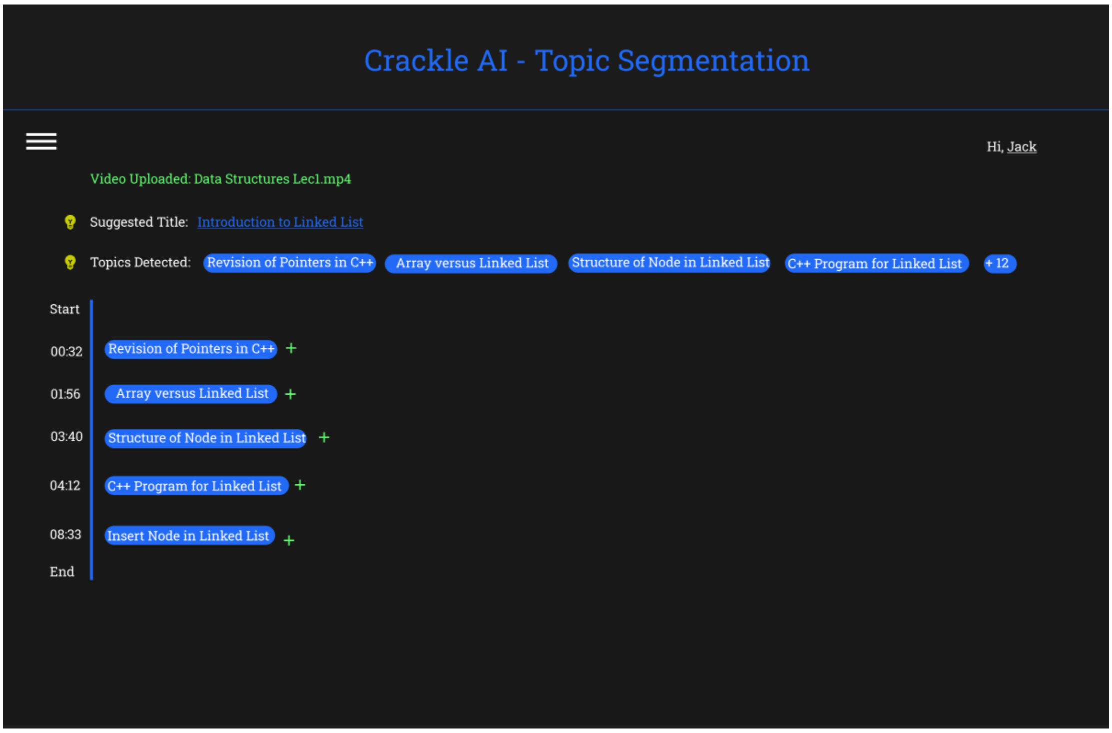


Fig 4.4.1 File Upload



Video Selected: Data Structures Lec1.mp4

Fig 4.4.2 Upload Success



Crackle AI - Topic Segmentation

Hi, Jack

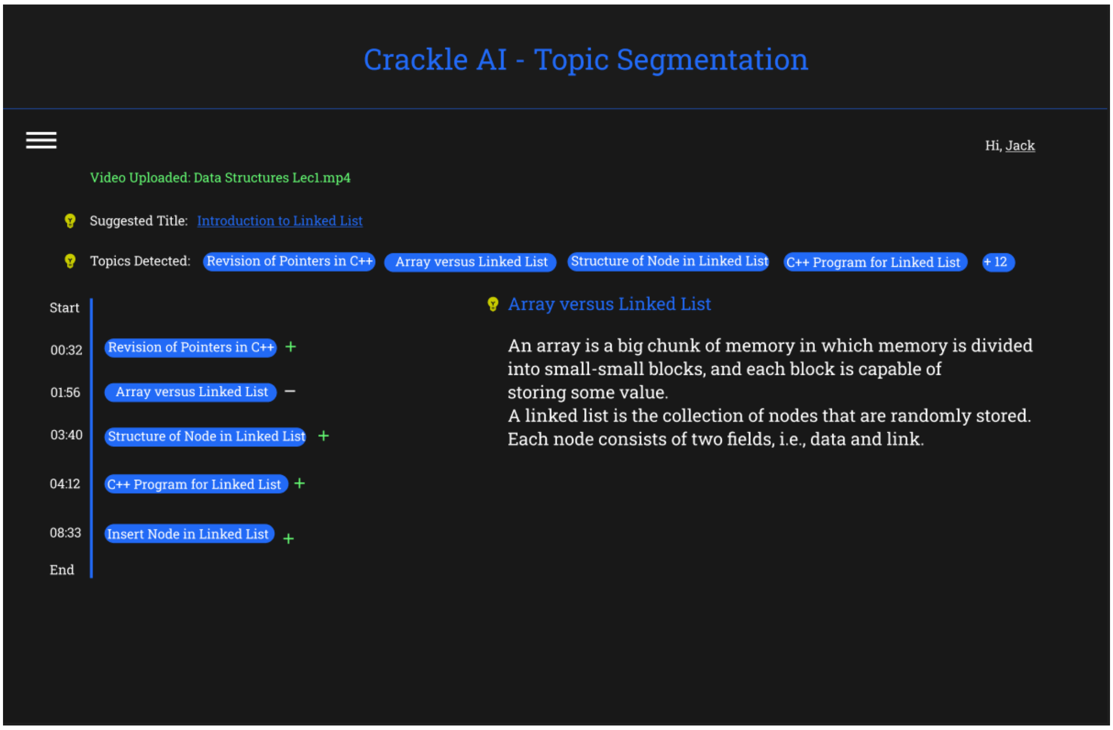
Video Uploaded: Data Structures Lec1.mp4

Suggested Title: Introduction to Linked List

Topics Detected: Revision of Pointers in C++ | Array versus Linked List | Structure of Node in Linked List | C++ Program for Linked List | + 12

Start
00:32 Revision of Pointers in C++ +
01:56 Array versus Linked List +
03:40 Structure of Node in Linked List +
04:12 C++ Program for Linked List +
08:33 Insert Node in Linked List +
End

Fig 4.4.3 Topics found inside the video



Crackle AI - Topic Segmentation

Hi, Jack

Video Uploaded: Data Structures Lec1.mp4

Suggested Title: Introduction to Linked List

Topics Detected: Revision of Pointers in C++ | Array versus Linked List | Structure of Node in Linked List | C++ Program for Linked List | + 12

Start
00:32 Revision of Pointers in C++ +
01:56 Array versus Linked List –
03:40 Structure of Node in Linked List +
04:12 C++ Program for Linked List +
08:33 Insert Node in Linked List +
End

Array versus Linked List

An array is a big chunk of memory in which memory is divided into small-small blocks, and each block is capable of storing some value.
A linked list is the collection of nodes that are randomly stored. Each node consists of two fields, i.e., data and link.

Fig 4.4.4 Topic Summary using abstractive technique

CONCLUSIONS AND FUTURE SCOPE

5.1 Work Accomplished

- Generated automatic transcripts from videos.
- Built a summary using the transcript as a description of the video and assigned meaningful title to the video.
- Made partition of the transcript into segments such that each segment holds a different topic/context from the adjacent partitions and constructed a summary of each segment and gave an appropriate title to each video segments.
- Built a Web interface to integrate the above objectives and add an option to upload the video files to process them through the AI pipeline.

5.2 Conclusions

The main problem we faced during online learning during Covid-19 was that in most of the courses, videos were not labelled or tagged meaningfully, or coherently, or in a relevant manner for the topics they include and it became quite cumbersome to find the desired topic or concept quickly. Therefore, we developed a system to provide apt and brief titles to long lecture videos as well as provide key-framing (video indexing) to easily navigate within topics in a video.

We generated transcripts of lecture videos and put them through abstractive summarization model to produce summary and hence, titles for the videos. We also used topic segmentation algorithm for video indexing. The aim was to speed up browsing and searching of a large collection of video data and achieve efficient access and representation of the video content.

Speech or text summarisation is a fast-growing field of research that has the potential to contribute to many application domains and tasks. At present however, the evidence for their effectiveness remains limited. This work indicated several research directions towards further advancing the performance of video summarization systems. Besides these proposals for future scientific work, we believe that further efforts should be put towards the practical use of summarization algorithms, by integrating such technologies into tools that support the needs of modern media organizations for time-efficient video content adaptation and re-use.

5.3 Benefits

The general time span used by individuals in the global communities to read texts or watch videos is reducing by the day. People are looking for every avenue to read documents and watch videos without having to encounter unnecessary information. This problem is solved with Automatic Text Summarization which makes it easy for people to extract information quickly because the central idea has already been summarized.

- Instant Response: You can save time and get other jobs done by relying on the advantages that the computer has brought into how you get data.
- Easy Navigation: Relevant titles and video indexing will help to go on the desired topics.
- Increases Productivity Level: Instead of going through content that you do not need it saves you the stress by reducing the title of the video to just 15 words or less. This way, your productivity level would increase and you would be able to channel your energy to other crucial things.
- Varied and Extensive Potential: NLP and automatic text summarization have become a lifesaver when it comes to summarizing long and tedious documents, be it technical, financial, legal, medical, or even literary. From academia to businesses, every sector can reap its own benefits. And we are still just scratching the surface of its true potential.

5.4 Future Work Plan

In future, several improvements in transformers and Architectural patterns will result in much more resilient system that can handle several thousand of requests efficiently. We hope to improve upon the speed of the entire pipeline, and the amount of data to work upon.

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 Articial 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).

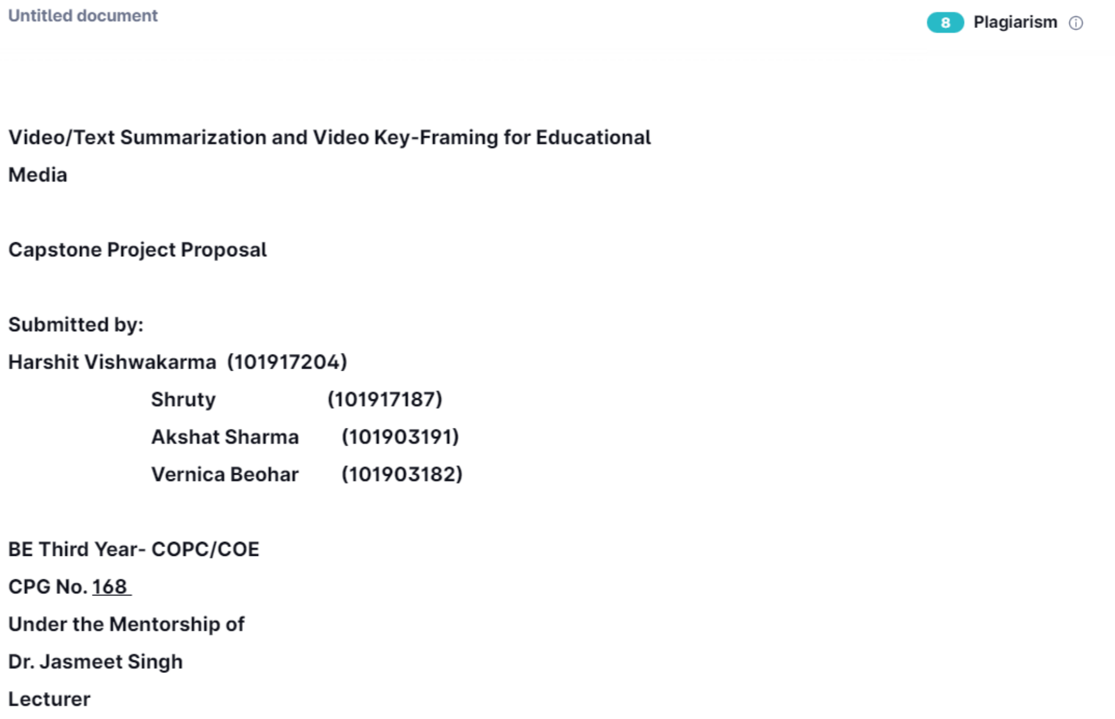
[18] Islam, A. and Inkpen, D. (2008). Semantic text similarity
using corpus-based word similarity and string similarity.
ACM Trans. Knowl. Discov. Data 2, 2 (Jul. 2008), 1–25.

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

Plagiarism Report



Untitled document

8 Plagiarism ⓘ

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

Fig B1, Plagiarism check by Grammarly with 8%