1. **Introduction**
   1. **Project Overview**
   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., DailyMotion, 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 s/he is 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 coronavirus 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 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 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. **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. **Problem 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 were 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, 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.

* 1. **Assumptions and Constraints**

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

|  |  |
| --- | --- |
| **S. No.** | **Constraints** |
|  | //Size of video |
|  | The computational power available to fetch the data and train the model. |
|  | The system currently works only for the English language. |
|  | //more |

* 1. **Standards**
  2. **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 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.
  1. **Methodology**
* **Data Collection and Pre-processing:** We will pick an educational domain-specific dataset from various openly available platforms like Kaggle, TVSum, WikiHow.
* **Data Pre-Processing:** We will pre-process the dataset and the 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 [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. We will combine the text summarization and paragraph clustering models to summarize and index a 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. **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 under 15 words.
* Should output the transcript of speech detected.
* Should divide the transcript into segments of topic they are based on.
  1. **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.

1. **Requirement Analysis**
   1. **Literature Survey**
      1. **Theory Associated with Problem Area**

**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.

**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), convolutional neural network (CNN), as well as LSTM, NNLM, AMR, GRU, AE network models.

**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 express 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.

**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.

**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).

* + 1. **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.
* **//TABLE**
  + 1. **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.

* + 1. **Problem Identified**
    2. **Survey of Tools and Technologies Used**

1. **Methodology Adopted**
   1. **Investigative Techniques**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Investigative Project Techniques** | **Investigative Techniques Description** | **Investigative Projects Examples** |
|  | Descriptive |  |  |
|  | Comparative |  |  |
|  | Experimental |  |  |

* 1. **Proposed Solution**

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 and Video Key-Framing (Indexing) model. We will create a meaningful summary of the video based on the text transcript generated by sampling the audio. We will also analyse the video to generate key-frames, then generate a textual summary of each key-frame.

Smart Video Naming would assist teachers in naming their videos appropriately using text summarization, as well as 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. 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.

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.

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. We will use Meta’s wav2vec model for generating transcripts from the audio sample of the video. We will build an abstractive summarization model fine-tuned for educational videos, articles, and conferences. 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. 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 create REST based architecture of our AI server combining the above models deployed on the Cloud for guaranteed up-time.

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. **Work Breakdown Structure**
  + **Data Collection:** Firstly, we will be collecting textual data from various openly available platforms like Kaggle, TVSum, WikiHow.
  + **Data Pre-processing:** After data collection, we will start with data pre-processing and the creation of the data pre-processing 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.

// Image & Diagram

* 1. **Tools and Technology**
* **Hugging Face:** Hugging Face is an open-source provider of natural language processing (NLP) models. It is a community and data science platform that provides tools that enable users to build, train and deploy ML models based on open source (OS) code and technologies.
* **Transformers:** The Transformers library is the natural language processing library and one of the most popular attractions Hugging Face offers. It is backed by deep learning libraries – [PyTorch](https://analyticsindiamag.com/crypten-a-research-tool-for-secure-and-privacy-preserving-machine-learning-in-pytorch/) and TensorFlow. Transformers provides APIs to easily download and train state-of-the-art pretrained models that enable a developer to perform various NLP tasks like text classification, information retrieval, abstractive and extractive summarization, name-entity recognition, image captioning, etc.
* **Colab Notebook:** Colab notebooks are Jupyter notebooks that run in the cloud and are highly integrated with Google Drive, making them easy to set up, access, and share. it does not require a setup and the notebooks that you create can be simultaneously edited by your team members - just the way you edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in your notebook.
* **Flask (Web Framework):** We will use Flask library for implementing HTTP server and model deployment. Flask is an API of Python that allows us to build up web-applications. Flask is based on WSGI (Web Server Gateway Interface) toolkit and Jinja2 template engine. Unlike the Django framework, Flask is very Pythonic. It doesn’t have a huge learning curve, it has less base code to implement a simple web-Application and it is very explicit, which increases readability.
* **React:** React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “components”. It is used to build single-page applications.
* **AWS EC2 Linux Instance**: We will use Amazon Elastic Compute Cloud (Amazon EC2) and use a Linux instance to deploy the server. An instance is a virtual server in the AWS Cloud. We can set up and configure the operating system and applications that run on our instance.
* **AWS RDS Instance:** We will use AWS RDS instance (Amazon Relational Database Service) to host cloud SQL database for user credentials. Amazon RDS is a managed Database-as-a-Service (DBaaS) that makes it easy for IT administrators to set up, operate, and scale relational databases in the cloud. It also supports Amazon’s database platform, MySQL and PostgreSQL compatible relational database.
* **Wav2Vec:** We will use Meta’s Wav2Vec for Automatic Speech Recognition. It is one of the current state-of-the-art models for ASR due to a self-supervised training which allows us to pre-train a model on unlabelled data which is always more accessible. It uses convolutional layers to pre-process raw waveform and then it applies transformer to enhance the speech representation with context.
* **Google Pegasus:** The Pegasus model was proposed in [PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization](https://arxiv.org/pdf/1912.08777.pdf). Pegasus’ pretraining task is intentionally similar to summarization: important sentences are removed/masked from an input document and are generated together as one output sequence from the remaining sentences, similar to an extractive summary.

**Conclusions and Future Scope**

**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.

**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.

//dataset, training, testing, results, evaluation, accuracy

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.

**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.

**Future Work Plan**