Østfold University College

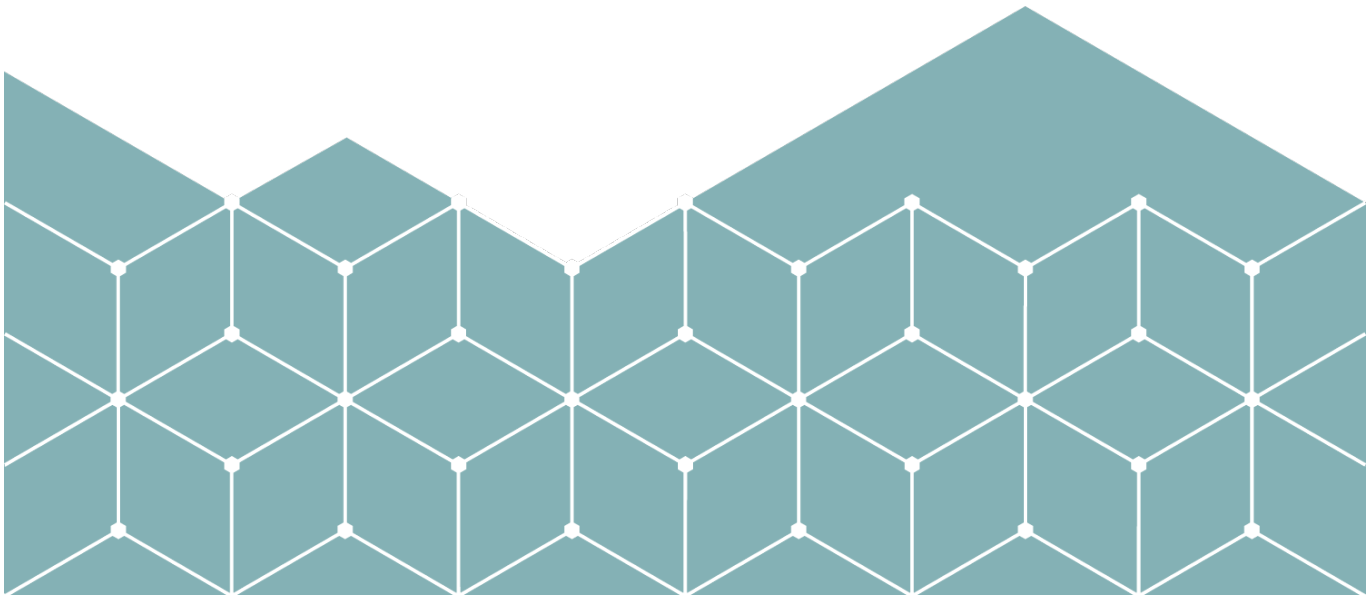
MASTER'S THESIS

Lecture Video Segmentation using Speech Content

Dipesh Chand

November 2020

Master in Applied Computer Science
Faculty of Computer Sciences



Abstract

Nowadays, lecture videos are valuable and useful resources for learning. The video that is captured in the lecture can be available and accessible online, as are a flexible resource to comparison with a textbook and classroom itself. Nevertheless, the adoption of lecture videos has been limited, primarily due to the difficulty of quickly finding the specific content of interest within a lecture video. Video segmentation, separating the video into a meaningful section, will significantly increase the usability.

In this thesis, we present a lecture video segmentation model based entirely on the speech content of the instructors. The objective of this research is to explore audio extracted from lecture videos to obtain Textual and Acoustic features and use them to segment the lecture video. One of the primary reason for doing so is that, unlike other sources which may or may not be available and can be utilized, lecture video always contains the audio track. To achieve this goal, we used different open source tools and algorithms like Audio extractor, VAD, ASR, Acoustic feature extractor, and segmentation algorithms because they are easily and freely available and there are always lots of resources available while utilizing them. To evaluate our proposed model, we create our own dataset containing a diverse set of 37 lecture videos and also manually created ground truth. The performance is measured by using metrics like precision, recall, and F-score and obtained 0.69, 0.58, and 0.63 respectively. We also compared our model with some previously known similar models where our model outperformed in all three metrics. The overall results of the study are presented as a lecture video segmentation pipeline, integrating various tools and techniques, and showing promising performance which we can further used for more detailed research in the content-based search and retrieval using speech content.

Keywords: Content-based search, Lecture video, Lecture video segmentation, E-learning, Speech content, VAD, ASR, NLP, Audio analysis, Information extraction.

Acknowledgement

First of all, I would like to express my deepest gratitude to my supervisor Dr. Hasan Ogul, for his excellent guidance, extensive knowledge, patience, and for providing me continuous support, flexibility, and motivation to complete my master's thesis. With his stepwise and simple to complex work approach, it has been a great learning period for me during this thesis. I could not have imagined having a better advisor and mentor for my thesis study.

I would also like to express my appreciation to all the professors and faculty members of the Faculty of Computer Science at Østfold University College who have given me a lot of valuable knowledge in the field of Computer Science.

Last but not least, I am grateful to my family who always motivated and supported me, all of my friends and colleagues who were always by my sides.

Tabel of Content

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, technology is widespread in various sectors, including government agencies, businesses, services, schools, and households. This allows us to do anything, anywhere at any moment by the use of information technology; the job becomes more effective because it just requires a little time to get information. With the rapid development and easy access to technology, there is tremendous growth in the popularity of e-learning [1], [2]. E-learning is a teaching approach focused on the evolutionary principle of knowledge access, which provides instruction and preparation for a diverse range of an audience, and which accommodates a greater number of learners than the conventional classroom [3]. Over the years, learning approaches change and adapts to new trends and circumstances. Nowadays, learning from online resources and specifically lecture videos is gaining lots of popularity. Online courses have become a popular source of learning because of its availability and easily accessible anytime, anywhere. And many education institutes are now being primarily focused on online and digital media as a teaching platform. In addition, there are now several Massively Open Online Courses (MOOC) that are popular globally for offering online lectures in various fields and are an excellent learning source. The most valuable benefit of a video that is captured in the lecture is that it is available everywhere. A key drawback of these types of lecture video is its failure to reach an important subject easily while we use the video

as a reference. It may take time to access the specific information within that lecture video and also not feasible to scan every lecture to get specific information.

1.1 Motivation

Various topic contents are often covered in the lecture video. The user may not be interested in all of these contents, but only in some specific content, and if there is no summary relating the topic to the video, the user will need to watch the video from the beginning until a topic of interest is found. Generally, the majority of platforms for making lecture videos available have an only topic of the lecture and nothing in this regard. In order to deal with this sort of problem, retrieving some specific parts of the lecture video, content-based retrieval comes into the picture. Retrieving the desired part of the video is still a very difficult and time-consuming process. Therefore, a browsing system based on content-based retrieval is needed to provide the desired lecture video part. The segmentation of the lecture video is thus focused specifically on the speech content of the videos because speech is always present in the lecture and this is the first step for the developing content-based browsing system.

1.2 Research Questions and Objectives

Lecture videos have many specific features that differentiate them from other types of videos, usually it contains text contents, video frames, and audio tracks [4]. The most significant of those features is that much of the content is based on the speech of the author. That is why the objective of this research is based on the speech of lecture videos, our method explores audio extracted from lecture video to obtain textual and acoustic features and utilize them to segment the lecture video. One of the main reason doing so is that, unlike other sources which may or may not be available and can be utilized, lecture video always contains the audio track.

This thesis is motivated by two main research questions:

1. How can we use speech content of lecture video to determine the transition of segments?
2. How can we use state of art tools to segment the lecture video based on the speech?

These questions require further examinations through these queries:

- How can we extract speech from the lecture video?
- How can we extract textual and acoustic content from the audio of the lecture video?
- Which tools should we use in this project?
- Which features of speech should we consider while segmentation?
- Can Automatic Speech Recognition (ASR) be used to extract the accurate text from the speech of the video?
- How to create dataset?
- How can we create a ground truth for evaluation?
- How can we evaluate our proposed model?

1.3 Organization of Thesis

The organization of this thesis is as follows. Chapter 1 briefly describes the motivation and scope of this project. Chapter 2 provides a fundamental understanding of Natural Language Processing and the literature review of existing approaches for Content-Based Search and Lecture Video Segmentation. Chapter 3 explains the method used for successfully completing this thesis. In Chapter 4, the experiment performed and the outcome result was illustrated. Chapter 5 discussed the overall experiment and results. Finally, the last Chapter 6 concludes and summarizes the thesis, and recommendations for future work are made with the following experimental results.

Chapter 2

Background and Review

This chapter explains the essential details that we used in this thesis to understand the background and the theory. This clarification allows all readers to better understand the research material and also enables the non-expert public to better understand the project's workflow in upcoming sections. This chapter will also cover the literature related to the research which we are going to perform and help us to understand what had been done up to now in this area of interest. We will discuss the concept of Natural Language Processing in Section 2.1. Section 2.2 includes an overview of literature reviews relevant to content-based search. And in the same way, Section 2.3 further addresses Lecture video segmentation and its literature review, which contributed to further investigation in this thesis.

2.1 Background in Natural Language Processing

Natural Language Processing (NLP) uses algorithms to grasp and analyze human natural language. This technology is one of the most widely used areas in machine learning. With the continuous development of Artificial Intelligence (AI), the demand for tools and technology related to NLP also continues to increase. NLP models can examine language and speech, reveal contextual patterns, and generate audio and text insights.

Basically, NLP implements text and language machine learning models.

The focus of the NLP is on training machines to understand what is written and spoken in real. An NLP algorithm is in operation every time you dictate something into your mobile phone and want it converted into text. You can predict whether the analysis is successful or poor using the NLP for a text review. In an article, you can use NLP to predict and segment certain categories. The book's genre can be predicted by using NLP. You can also use NLP to create an algorithm for the translator or voice recognition system and classify the language.

Let's go through a simple example to understand the general terms of NLP. Imagine we have two very simple documents.

Documents:

- Document A: "Black House"
- Document B: "White House"

Featurize based on word count:

- "Black House" → (black,white,house) → (1,0,1)
- "White House" → (black,white,house) → (0,1,1)

Here, the document is just Black house and then the second document is White House. That means it's just a document of basically a single sentence. So the first sentence is Black House document A and second sentence White houses document B. A simple way to featurize text documents is to featurize based on a word count. So we transform a black house into a vectorized word counts. We create a vector count of all the possible words through all the documents in this case they're black, white, and house and then we just count how many times those words occur in each document. That means in this case for document A Black House we get (1,0,1) since black occurs 1 times, white doesn't occurs anytime and house occurs once. Similarly in white house we get (0,1,1) because black occurs 0 times white once and house one time. A document represented as a vector of counts is called a bag of words.

- "Black House" → (black,white,house) → (1,0,1)
- "White House" → (black,white,house) → (0,1,1)

Once we have these bags of words vectors we can use cosine similarity on the vectors to determine similarity of the documents themselves. This is useful because we're treating each document as a vector of features meaning we can perform mathematical operations such as the cosine similarity taking their dot products and then dividing it by the multiplication of their magnitudes or other similarity metrics to figure out how similar two text documents are to each other. Following Equation 1 and Figure 1 defines and shows the cosine similarity respectively.

$$similarity(A, B) = \cos(\theta) = \frac{\mathbf{A}.\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \tag{1}$$

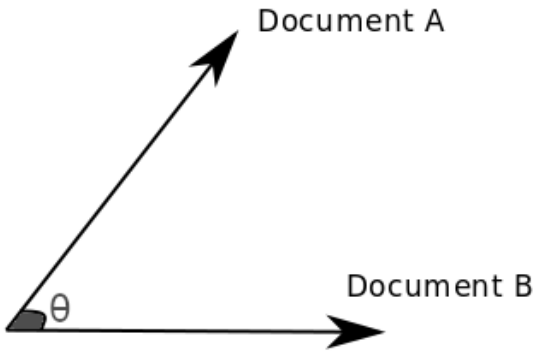


Figure 1: Illustration of cosine similarity

We can improve on bag of words by adjusting word counts based on their frequency in the corpus (the group of all the documents). We can use tf-idt (Term Frequency Inverse Document Frequency), which is the product of term frequency and inverse document frequency. Term frequency is the importance of the term within that document.

i.e. tf(t,d) = Number of occurrences of term t in document d.

And, the inverse document frequency which is the importance of the term

in the corpus itself.

i.e. idf(t)= log(D/ t), where D is the total number of documents and t is equal to a number of documents with the term.

Mathematically, tf-idf can be expressed as the following equation.

$$W_{x,y} = tf_{x,y} \times \log(\frac{N}{df_x}) \tag{2}$$

where,
$tf_{x,y}$= frequency of x in y
df_x= number of documents containing x
N = total number of documents

The reason we do this is so that we can get not just a word count but also some sort of notation on how important a word is not just relevant to the document but to the entire corpus of all the documents.

Word2vec

Word2vec is an NLP technique or a framework for learning word vector[5]. The word2vec algorithm implements a neural network model for learning words from a large corpus of text. Such a model can identify interchangeable terms once trained or recommend alternate terms for a partial sentence. Word2vec represents each word with a set of names that is called a vector. The vectors are carefully selected to show the degree of semantic similarity between the term represented by certain vectors in a basic mathematical function (the cosine similarity between the vectors).

Word2vec is a set of related models for word embedding generation. These are neural networks two-layer models, which can be equipped to recreate the speech environment of words. Word2vec uses a large text corpus as input which produces a vector space that is typically has hundreds of dimensions, and a corresponding vector is allocated to any single word in the corpus. The word vectors are in the vector space, so that words in the

corpus that share a common context are close together[1].

The basic idea how Word2vec works are as follows:

- We have a large corpus of text.
- Every word in a fixed vocabulary is represented by a vector.
- Go through each position t in the text, which has a center word c and context ("outside") word o.
- To calculate the probability o given c (or vice versa), use the similarity of word vectors for c and o.
- Continue to change word vectors to optimize this probability.

2.2 Literature Review

2.2.1 Content-based Search

Nowadays, digital audiovisual records are commonly used in learning for users to access online, independent of time and location. For a particular topic of interest, it is very hard to search for such videos. E-learning information must be generated efficiently so that lecture videos based on content can be found more effectively. For this, the keyword search in the lecture video needs an effective content-based retrieval system. The challenge is, however, not locating a lecture in a video archive, but to find the correct location in a video stream of the appropriate keyword. Content-based processing inside video data requires descriptive metadata to be generated manually or retrieved through automatic processing. Traditional Optical Character Recognition (OCR) techniques focused on high-resolution scans of written (text) records and Automatic Speech Recognition (ASR) concentrated to extract transcript from an audio track of a lecture video which must be enhanced and modified to apply for further processing. Image frames containing clear text data must be first detected in image OCR. And for ASR, the audio track should be clear to extract the transcript.

[1] https://en.wikipedia.org/wiki/Word2vec

The text must then be extracted from its context, and mathematical transformations must be introduced before the text is effectively processed in popular OCR algorithms or ASR algorithms. The method is still very difficult and time-consuming to retrieve a specific part of the video. Although various tools are available, there had been little work done on the audio-video section. So a more effective content-dependent retrieval system for video lectures is needed to promote the growth of e-learning.

2.2.1.1 Related Works on Content-based Search and Retrieval of Lecture Video

In recent years, many researchers have been conscious of the need to have content-based access to images and videos. Research efforts have contributed to methods for collecting images and video content. Such approaches are grounded in the understanding of computer vision, pattern recognition, speech detection, and machine learning. The techniques are used to classify the similarities in the audiovisual content of data derived from low-level functions. Those characteristics are then clustered to use in video retrieval. This section will describe the use of these types of models to provide an image and video retrieval through content-based in a previous study.

The study [6] presented an approach to content-based lecture video indexing and retrieval in a lecture video portal. Automatic video segmentation and keyframe recognition have been used, using OCR and ASR techniques, to automatically derive textual content-based metadata from keyframes and audio tracks of the lecture clips. For content-based video browsing and search functionality, a large-scale learning video archive has been set up using those metadata and consumer review has been done.

In the same way, [7] proposed a complementary video indexing and search integrated into a large video repository by using a novel approach and gives personalized results. Initially, they obtain relevant keyframes by segmenting videos and detecting keyframe. Secondly, to extract text keyword,

OCR and ASR algorithms are applied over the keyframe. The text detection the feature uses the SVM classification based on rich descriptors such as HOG, Gabor, and edge functions which improve performance and uses the PLS technique to minimize dimensional to increase the SVM rate. Color, Texture, and Edge features were obtained in the third stage. Finally, the search similarity calculation is taken on the extracted features and the output is presented to the users with personalized re-rank results as per interest.

A natural language approach for indexing and retrieving videos based on the content of video clips to meet user requirements is proposed by [8]. The authors developed a two-phase approach to content-based video-indexation and retrieval to classify video clips. Their method combines natural language processing, named retrieval, text, and video indexing based on frames and techniques for retrieval of data. A correlation between created questions templates and clip content tests the significance of video clips in terms of questions.

2.2.1.2 Benefits and Features of Content-based Search in Lecture Video

Several types of research and project had been proposed on content-based retrieval methods and based on those studies we can categorize the benefit of utilizing Content-based Search (CBS) in lecture videos. It could be grouped into three distinct categories.

a More Accuracy for the Search and Improves the Recognition Rate.

More recent research focuses on collecting information from audio and visual content of Lecture video so that the details of the clip are properly understood. The growing number of video lectures thus lead to automatic time segmentation and lecture description. Such automated description and segmentation will increase the search and retrieval of video lectures and maximize the relevance of content to the learner [9]. Automated segmentation and annotation involve content informative metadata extrac-

tion. Automated segmentation can reduce processing costs dramatically, thus reducing repetitive tasks [10]. The key features of most existing video recovery systems include color, texture, shape, motion, object, face, audio, genre, etc [11]. It is clear that the more features used for, the higher the video retrieval accuracy [12]. [13] found from their survey that rather than extracting text content from video files only, this allows more accuracy for the search if the extraction is performed for speech too.

According to [14], key-frame identification is essential for the indexing and search of content-based video search. Changes in a video were observed with various methods in their study. They choose two types of lecture videos as input for experimentation, type-1: video comprising only slides and type-2: video comprising slide view and presenter view. Their experimental results reveal, for Type 1 and Type 2 lecture videos for various segmentation periods, that global pixel variations and component-based approaches are better for both recall and precision values relative to all other methods mentioned in their study. For the slide change detection, it is advised to choose either Connected Component-based or Global Pixel Difference methods with a 4s time interval.

A useful tool for the indexation and retrieval of lecture video material is the technique of ASR. However, voice recognition is still an active field of research and virtually none of the existing voice recognition systems have achieved a good recognition rate. [15] tested the new software for speech recognition to find a way of transcribing German lecture videos automatically. They also developed an automated vocabulary extension method to add new vocabulary training resources and introduced technical terms relevant to topics to the training data. The research results show that the Word Error Rate (WER) has reduced by 12.8% when the language training period of the speaker has been increased by 1.6 hours.

A video retrieval framework based on content and text is introduced by [16]. Their approach uses both text-based retrieval and content-based retrieval

procedures. The technique includes a tag-based learning procedure and implements low-level feature computation based learning. In the training module, first, a list of visual objects known as frames is segmented into the video data, and each frame contains the corresponding tags. The tagged frames are then processed using the three different low-level feature computation techniques: the LBP for texture information, the canny edge detection technique for edge or object estimation, and the color grid movement for the color variation calculation of frames. Finally, for the classification of videos according to a user inquiry, the KNN classification is implemented. They examined their new working model and noticed that it is possible to improve the performance of traditional information retrieval techniques using this approach.

b Simple and Flexible Search Function

CBVR decreases the time burden as the user gets clips that include the most appropriate search query, helping to increase the overall user experience [17]. And also, it's sometimes hard for users to find parts of their immediate interest in a full lecture video clip or multiple videos. Video segmentation and Tagging methods can extract video subjects from the indexing process to remove these difficulties [18].

[19] developed a video analysis method used for content-based information retrieval and noticed that using content detection to extract the content line structure such as title, subtitle, key-point, etc., made search more flexible in a video retrieval system.

c Fast Retrieval and Efficient for Retrieving the Videos

As technology is increasingly used and the vast content on the Internet is accessible, a solution must be found to access this content through quicker and more efficient retrieval methods, so that the content can be looked at for less time and better understood. Video indexing is a method to mark and organize videos effectively to easily find and view them. Index-

ing optimization can reduce processing costs dramatically while reducing manual labor [12]. Though content-based search and retrieval have not yet achieved this position, but some work had been done to make better video retrieval.

[20] develop a system that can retrieve a related video according to the users keyword via a speech on the subject and found that proper indexed query handling in the database makes navigation easier and efficient. With the implementation of this content-based searching becomes faster and response time increases than the other existing video retrieval system.

[21] proposed the system, which optimized the searching of video based on video text content. They use a canny edge detector algorithm to preserve the frames for further process and histogram of the Gradient feature extraction method for extracting the feature from the frames to predict the frames which possess the text information. Finally, to classify the text frame from all detected frames, the multi SVM classifier is used. The performance and effectiveness of proposed indexing functionality are proved after evaluation.

According to [6], performance and learning effectiveness can be measurably enhanced by using video indexing tools. They suggest a method for automated video indexing and video search in large lecture video repositories. Text metadata are extracted through the application of video OCR technology on keyframes and ASR on audio tracks. For the detection of keywords, a video and segment-level keyword are used to browse and search through video content, using both the OCR and ASR transcripts as well as the identified text slide line forms. Evaluations show the reliability and effectiveness of the suggested indexing functions. In the same way, [22] suggested a video retrieval system and noticed that automatic annotations of the outcomes of OCR and ASR using Linked Open Data tools provides the ability to dramatically increase the amount of educational data connected. Therefore, in lecture video archives, a more powerful search and

recommendation system can be created.

[23] introduces a new visual interface for SBLV search and navigation via thin granular objects. In their approach, they first extract the embedded content objects from detected SBLV slides. When addressed during the lecture, each person is identified with their respective speech text in the lecture. Ultimately, the objects are displayed inside the user interface, along with other helpful hints, including cursor movements. Experimental results show that the new system could help digital learners search and locate content of interest in SBLV efficiently and effectively.

2.2.1.3 Data used in Content-based Search in Lecture Video

Content dependent search ensures that the video content is evaluated in the search. After a review of the primary studies, one can clearly see that data sources used in content-based video retrieval are Text content, video frame content, and audio content.

Table 1: The contents extracted from different data source

Data Source	Contents
Video Frame	Textual metadata, slide texts, colors, shapes, pixel contents of frames, bitmap properties, visual elements, and mathematical expressions included on lecture slides
Audio Tracks	Audio transcripts or textual metadata
Text Contents	Title, subtitle, video properties (extension, modified date, size, etc.)

Table 1 gives some details about data source and contents which can be extracted and utilized for content-based search. Lots of research has been performed based on the video frame and audio tracks extracted from the lecture video. The studies were more focused on the content extracted from the data source than the data source itself. So from our primary study, we can say that the textual metadata is by far the most relevant resource used for content-based search as we can also see in Table 1. Textual data can be extracted from both video frames as well as audio tracks [18]. Applying video Optical Character Recognition (OCR) technology on key-frames and ASR (Automatic Speech Recognition) on audible audio tracks can extract

textual metadata [14]. The OCR or ASR transcripts, as well as identified slide-line form of a text, can extract keywords, both on a visual or segment-level basis [22]. The content-oriented search approach will improve the user's browsing experience with numerous videos of interest.

2.2.1.4 Obstacles and Limitations of Content-based Search in Lecture Video

One of the key tasks in information management is data management. In order to correctly manage the data in different databases, appropriate information recovery techniques for the identification of user query relevant data should be developed. Nevertheless, the processing of unstructured data in contrast with standardized data formats is challenging. The video content is very complex among the various unstructured data formats such as web documents, text documents, pictures and others [16]. The videos have a much richer content with many raw data and very little structure previously used; it is difficult to search and retrieve videos [11]. Also, video retrieval takes too long because it usually takes too many attempts to look for and scan for a certain section of the video the user is interested in [20], [23].

The major limitations of the existence video retrieval systems are as follows:

1. Most current video retrieval system used the text metadata created manually. The creation of this metadata manually is a difficult task and is not enough to determine the pertinence of any video on the given topic [24], [25].
2. The issues that occurred during the development of the recorded videos for content-based retrieving include automatic segmentation, indexation and content-based retrieval from a lecture knowledge base with relevant data while selecting the video involved without looking into the Title or other global metadata [25].

Similarly to be specifically talking about the content-based video retrieval systems the major challenges we found in this study are:

1. Content-based methods collect metadata from the related video sections to construct a content searchable database. Such methods are tough to implement and time-consuming [18].
2. The extracted video content from OCR is from object selection and the recognition of the similarities between frames, while Video Lecture has homologous features between frames with many frames with the same information. So, the identification of distinct frames is crucial [13], [14].
3. The low quality of videos and text with different resolutions inside border boxes with a heterogeneous backdrop and a hard contrast ratio which often forbids accurate OCR result [26].
4. The background noise, changes in lighting, video compression, and occlusions caused by the teacher present a major challenge in automatically obtaining manually written content in Lecture video [24].
5. Dynamic adjustments on the camera can change the size, form, and luminosity of the slide; if the speaker steps in front of the slide, a partially obscured slack can be hindered and shifts in camera emphasis can also affect slide detection process [6].
6. Repetitions, errors, and rephrases in the SRT (Subtitle Resource Tracks) of lecture videos make it difficult to automatically tag, index, and content-based retrieval of appropriate information [26].
7. The technology for speech recognition for automated transcription of lecture video is poor inaccuracy at roughly 40-80% word error rates (WERs), which restricts the usefulness of CBS on the audio track of lecture video [15].

2.2.2 Lecture Video Segmentation

The goal of video segmentation is to divide the video stream into the basic elements of the index into a series of meaningful units. For various video

applications such as video browsing, retrieval, and summarization, this can be a very important step. However, because of the diversity of the underlying content structure, it has different meanings for various video genres for forming a set of meaningful units. One idea is to convert and build the video along with other lecture content to resolve this problem. Many online courses and e-learning systems, for example, use typical interfaces to allow students to view different topics in videos of other lectures. Figure 2 and Figure 3 show examples of how the segmentation is done with Lecture videos on online platforms.

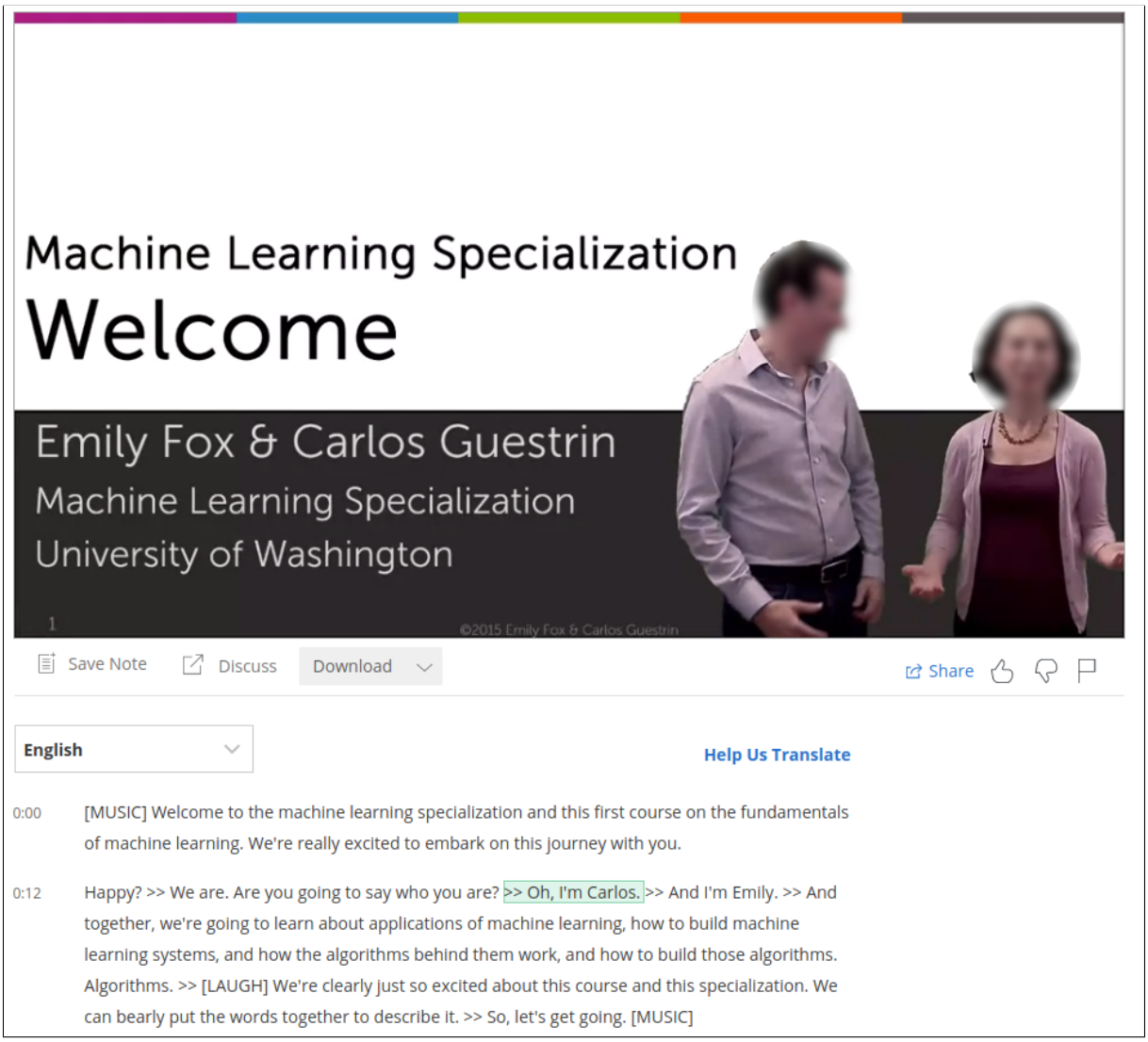


Figure 2: Lecture video from coursera.org

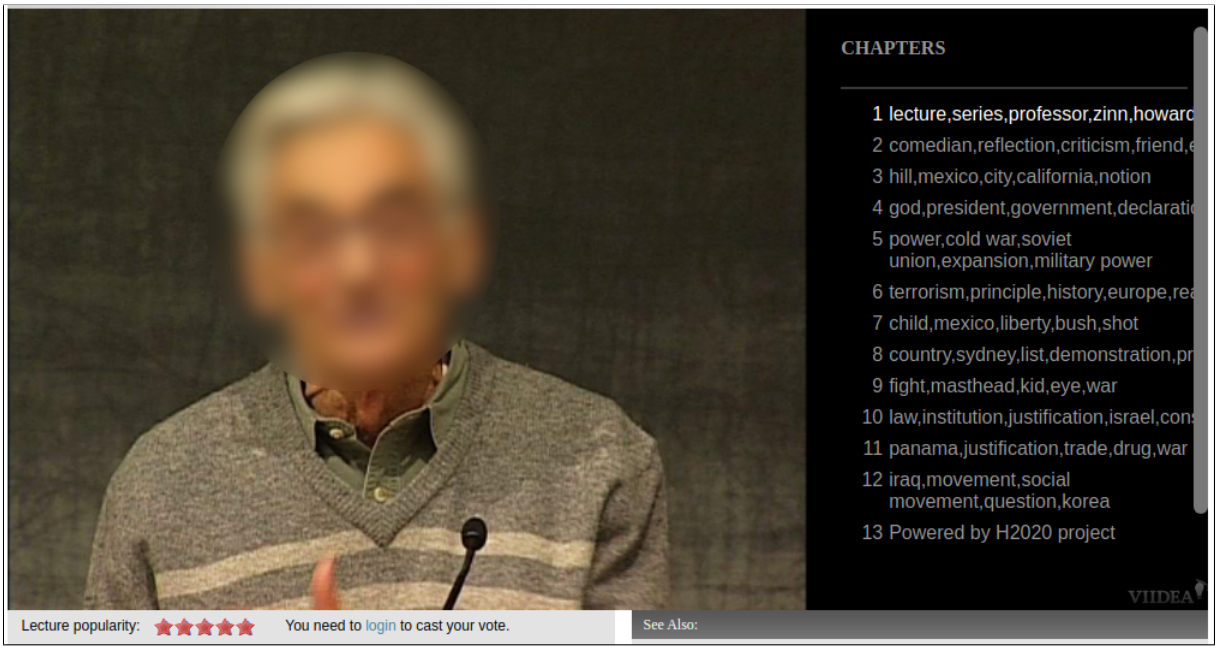


Figure 3: Lecture video from videolectures.net

We can see that the lecture video is like a lecture experience and that the transcript is divided into segments just below the video in Figure 2. Those segments have no defined subject, but when the subject is changed, the timing is displayed. This somehow allows the viewer to easily know that the change in topic. However lecture videos in another platform videolectures.net we can see that on the right side of the video there are some topics defined which segment the video and index like in Figure 3. Here the viewer can easily search through these indexes for their topic of interest and directly jump to those parts without viewing other parts of the lecture.

However, a critical pre-processing step must be taken to achieve such structured video lectures and to allow browsing and search functions: video segmentation. The video's knowledge structure can not be extracted and efficient browsing or searching is not possible without dividing an extended, continuous video into short, unobtrusive, and semantically internal segments.

Related Works on Lecture Video Segmentation

Some works related to the segmentation of lecture video are increasing

with a growing interest in this field. Up to now the widely used methods for segmentation of lecture videos typically involve keyframes or labels detection, text segmentation, segmentation based on slide change, and also some research based on audio contents. We totally understand that this topic is relatively new since we hardly see any research beyond a decade, but now the state of art and technology advancement has enabled us to do lots of research in this field. A framework of two module system is developed by [27]: a video segmentation/indexation module that decodes the educational video into images and creates automatically hierarchical indexes and a video browsing/query module to browse and scan for the video under certain request conditions. In order to minimize processing time, they apply OCR methods in the Area of Interest (AOI) section to retrieve text content from a video clip. A hypertext-assisted methodology has been implemented to exclude substantial human intervention from the OCR result. This method utilizes original lecture text, which was preserved in the medium of text files. They recognize the headline for each R-frame associated with a video screen to map the source of text into a video screen. After acknowledging the headline, it would map the text source headings to obtain the rest of the content. It ensures that the video content can be accessed from the source completely and reliably.

The TRACE method to perform the topic-specific video segmentation automatically based on a linguistic approach is presented by [26]. Experimental findings confirm that, considering video quality, the TRACE system can efficiently fragment the video to allow its content to be viewed and traced easily.

An interactive video content-related segmenting protocol that segments lecture video in subtopics based on speech signals is suggested by [28]. The text recognized by the ASR from the lecture speech was transformed into an index by means of Independent Components Analysis (ICA) rather than traditional tf-idf to represent the subtopics of video segments. This study has tried to use a dynamic programming segmentation approach that

minimizes the sum of cosine measurements between adjacent indexes. As a result of tests, they observed that the findings of tf-idf could be collected easily if indexes were used from the study of individual components.

In the field of lecture video segmentation with speech content, similar work has also been done as presented by this thesis. The purpose of video segmentation is to detect the main content change in the videos and split it. In a similar manner, [29] suggested a way to fragment lecture videos into meaningful pieces. They use video speech transcripts and interpret them and then use a word embedding for text representation. The precision, recall, and F-score of 0.465 and 0.491 and 0.477 were determined using their proposed system, respectively. In the same way, [30] proposed an optimization model of temporal video lecture segmentation using word2vec representation of transcripts and low-level acoustic features. The authors proposed an offline-based system which is basically using a combination of different individual tools to perform all the activities, i.e. they input in one tool and get the result and used that output to feed another tool. They extract the transcripts from the audio of the lecture video using Kaldi[2] ASR and removes the stop words and use Word2vec to calculate the word average vector to represent the transcripts. If the transcribed word is unsuccessful to find the topic transition they then used the extracted acoustic features from aubio[3] and finally used the segmentation algorithm to find the partition in the lecture video that best represents the topic boundaries. With their proposed method they got 0.40, 0.48, and 0.40 of average precision, recall, and F-score respectively. And in another research [31], the author presented a novel method for automatic topic segmentation of video lectures by using semantic annotation with knowledge base searches combined with the lower level feature of audio.

[2]https://kaldi-asr.org/
[3]https://aubio.org/

Chapter 3

Methodology

In this chapter, the methods and techniques used in this project are explained in detail. The objective of this thesis is to design a model that can segment the lecture video by only utilizing the audio source i.e. speech content in the lecture video. The goal of this project is to achieve the following outcome:

Input: A dataset containing a collection of lecture videos.

Output: Segmentation of those lecture videos along with the starting time of those segments.

3.1 Overview of Methodology

The following section provide an overview of how the experiments were designed and implemented and tested. The chapter is divided into two parts: the first part explains the design details. The second part gives the details of the implementation. In this work, the waterfall model is followed i.e. step-by-step approach where each component is partially or fully implemented to process the experiments. Figure 4 shows the waterfall model where each step is clearly separated and followed systematically to design and develop the system.

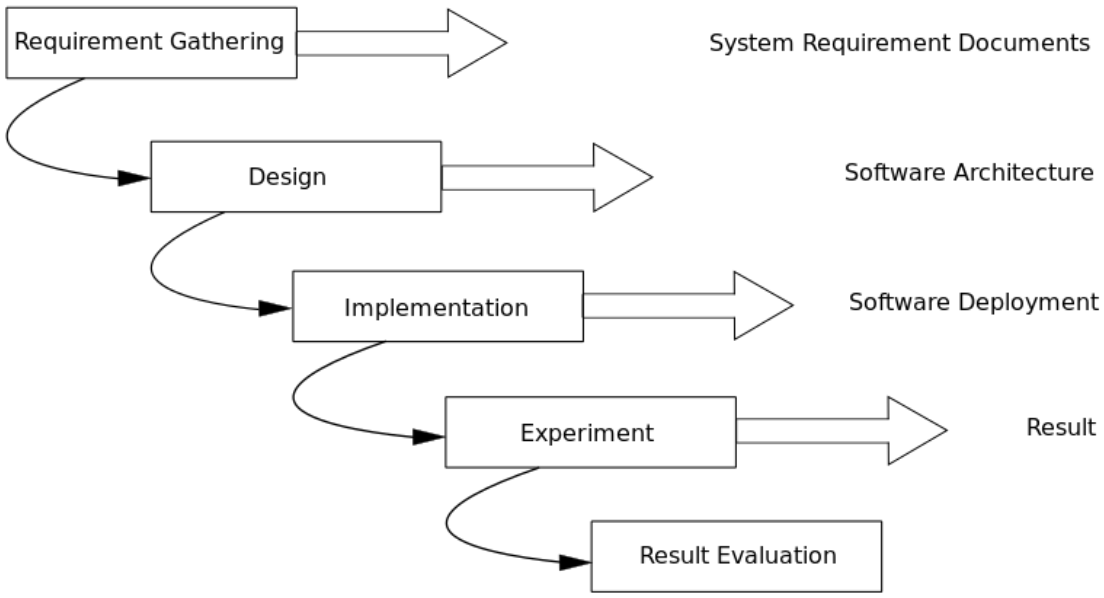


Figure 4: Waterfall model

The Waterfall Model is very simple to understand and use. Each phase must be completed before the next phase can begin and there is no overlapping in the phases. Each phase is briefly described below and this chapter will be more focused on Design and Implementation and the next chapter will be dedicated to Experiment and Result obtained.

a Requirement Gathering

The first step is the requirement gathering. All the requirements that are needed to develop the proposed system are gathered with the proper analysis of the objective of the work and based on the literature reviewed. This includes resources, proper planning, deadline time limit, hardware/software requirements, and tools selection.

b Design

Design and implementation are the major parts of this project, so most of the time and effort are also given to these. The aim is to make the model

simple and easy to use, through which the one can run the experiments with a single click.

c Implementation

It consists of the detailed execution of the design software in a real scenario. After completion of the design part next step is to implement in the real field with a real scenario. At first, the framework is designed with a single input, but it is validated and modified with a set of different inputs for actual and thorough implementation. The implementation is described briefly in the "Implementation of the Lecture Video Segmentation" section of the report.

d Experiment

It is a systemically established process of information collection and measurement for variables of interest, which allows one to answer stated research questions, to test the ground truth data and to evaluate results. The component of the data collection of research or the project is common to all areas of research including physical and social sciences, sciences, business, and so on. The emphasis on ensuring an accurate and honest collection continues to be the same, although methods are different in each discipline.

e Result Evaluation

Evaluation is important to continuously improve our practice. Evaluations provide examples of success to inspire others and improve our internal project performance. This is the final step to act upon the data collected after implementing the system. The collected data are now proceeded or tested with the expected results. The data evaluation depends upon how the user wants. In this project, the final conclusion are made by comparing the outcome of the experiment with ground truth data.

3.2 Architecture Design

This "Lecture Video Segmentation" architecture consists of several modules and components, each of which is responsible for a single stage of processing. The modules used are briefly described below:

- **API:** Entry point of this architecture where lecture videos are sent to be processed.
- **Message Broker:** Message broker used for integrating the processing modules.
- **Audio Extractor:** Module that extract the audio tracks from input lecture videos.
- **Voice Activity Detector:** Module that detects and splits the audio tracks into entirely voiced parts, reducing the duration of silence.
- **ASR:** Automatic Speech Recognition module that transcribe spoken speech into text from the audio tracks.
- **Acoustic Feature Extractor:** Module that extracts low-level features from audio tracks.
- **Feature Aggregator:** Module that aggregates the transcription and low-level features extracted from the audio tracks.
- **Segmentation:** Module that segments the lecture video based on the extracted speech contents.
- **Database:** Used to store the data from processing modules.

Figure 5 shows the design architecture of our proposed model.

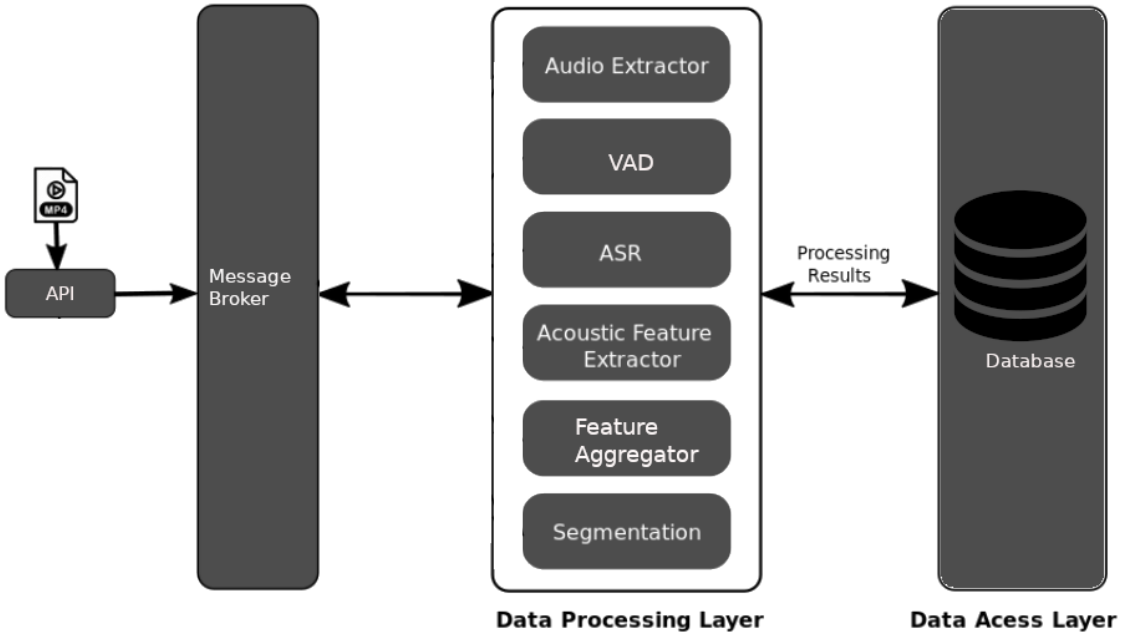


Figure 5: Architecture of lecture video segmentation model

Flow diagram is a diagram that visually displays interrelated information such as events, steps in a process, functions, etc., in an organized fashion, such as sequentially or chronologically. Flow diagram shows the step wise description of every component that is used in the system. It shows the work flow of the project. After visualizing the flow diagram it will be easier to understand the work-flow of this thesis. Figure 6 shows the flow chart of our proposed model.

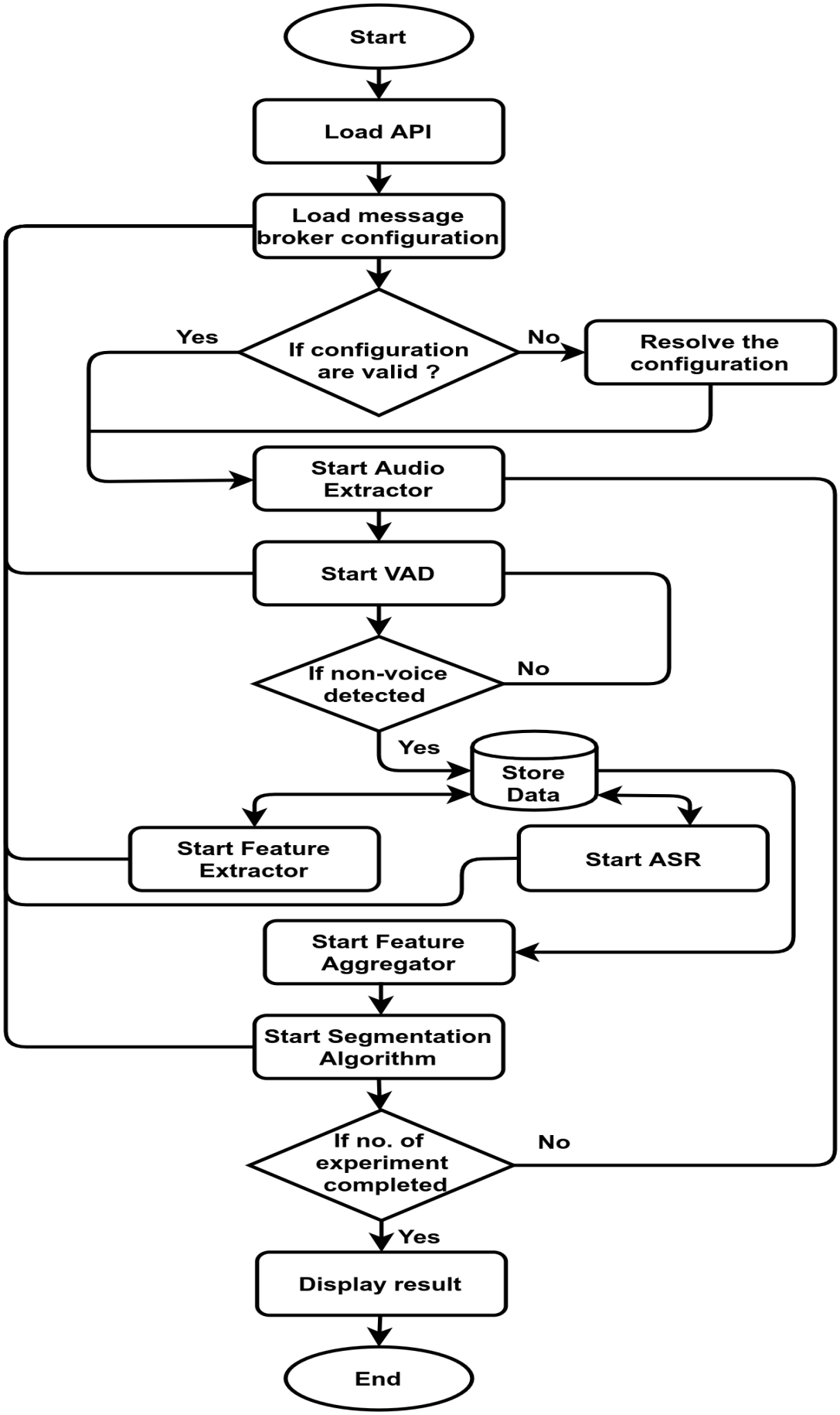


Figure 6: Flowchart of lecture video segmentation model

3.3 Implementation of Lecture Video Segmentation

Our Architecture is a basically a pipeline where modules are a group of data processing elements linked together to obtained the desired outcome. Figure 7 shows the block diagram of processing modules involve in our proposed lecture video segmentation model.

Figure 7: Block diagram of lecture video segmentation processing modules

The entire workflow can primarily be split into two parts: 1) the process of feature extraction, and 2) the process of segmentation. The feature extraction process comprises the extraction of textual and acoustic features from the lecture video and the segmentation process segments the lecture video using those features. The Feature extraction process is shown in Figure 8. And each module of the pipeline is clearly explained in this chapter below.

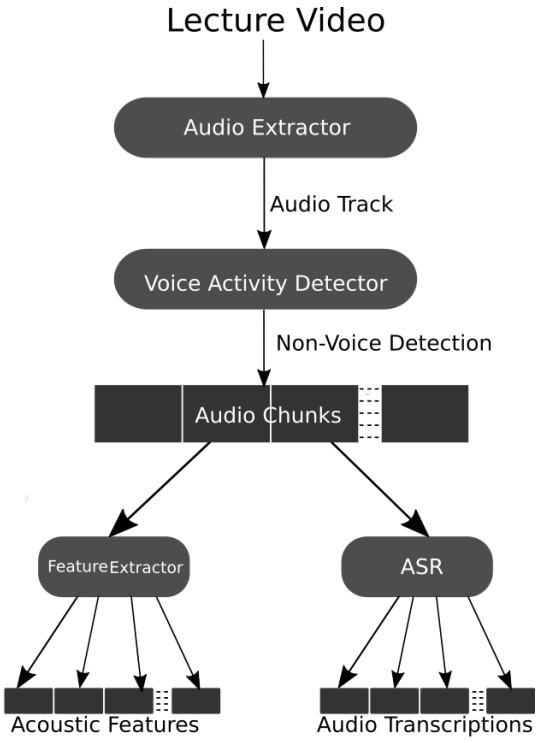


Figure 8: Feature extraction process from lecture video

3.3.1 Audio Extractor Module

Since our proposed model is based on the lecture video's speech content, visual content is not required. So the first thing which we need to take care of is to extract the audio tracks from the video clips. Here we focus on a lecture video that contains both image frames and audio as an input {IF, A} but we are only interested in audio track {A}. In this process, audio extraction is the result of removing all the image frames present in the video and just get its audio track. It is a rather simple process, and there are not many complexities involved to achieve it. Furthermore, there are plenty of free and open-source audio extraction tools to perform this task. Here we used Python bindings for FFmpeg[1]. Specifically, we focus on the functions for reading and writing files in a different format and only extract audio files without interfering with any other features of the lecture video file.

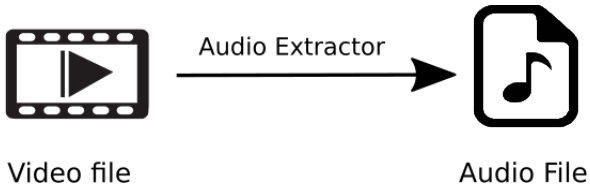


Figure 9: Extracting audio from lecture video

3.3.2 Voice Activity Detector Module

Voice Activity Detection (VAD) plays a leading role in our proposed model. VAD is described as the detection of voiced or non-voice portions of the speech, which is a key problem in many speeches/audio applications, such as speech recognition, speech enhancement, speech coding, audio classification, audio segmentation, and audio indexing [32], [33]. There are several VAD algorithms, but the basic task is to extract some measured features or quantities from the input signal and to equate these attributes with threshold variables, typically obtained from the sound and speech signal

[1]https://pypi.org/project/ffmpeg-python/

characteristics. The voice decision is taken if the values exceed the thresholds. The VAD requires a time-varying non-stationary noise threshold value. Usually, this value is measured in the inactive section of the voice. On the other hand, for signals dominated by voice-active segments, noise can differ before instant re-calibration at the next level of noise [34].

For an input signal x, voice activity detector objective is to determine whether it is speech or not. We express the VAD algorithm as a function y=VAD(x), where the desired target output is

$$y^* = \begin{cases} 1, & \text{if } x \text{ is speech} \\ 0, & \text{if } x \text{ is non-speech} \end{cases} \tag{3}$$

Correspondingly, the speech presence probability (SPP) is the probability that x is speech, SPP(x) = P(x is speech). A possible definition for the VAD is then

$$VAD(x) = \begin{cases} 1, & \text{if SPP}(x) \geq \theta \\ 0, & \text{if SPP}(x) < \theta \end{cases} \tag{4}$$

where θ is a scalar threshold.

In our proposed model we are implementing the Python interface to a VAD module developed by Google for the WebRTC project[2]. WebRTC VAD which is an open-source VAD based on the Gaussian mixture model that targets real-time performance, based on distributions of speech and non-speech features. Our VAD module uses multiple frequency band features with a pre-trained GMM classifier [35]. Given an audio file, our VAD module generates pulse-code modulation (PCM) audio data and used it to generate audio frames. Using these audio frames VAD filters out non-voiced audio frames and return only voiced audio. Basically, our VAD model produces two outputs: first with speech and non-speech segments,

[2]https://github.com/wiseman/py-webrtcvad

and second with 1's and 0's sequences with speech and non-speech frames [36]. Using these outputs, our VAD model compresses the silent packets of audio signals and separates the audio extracted from the lecture video into entirely voiced audio chunks. This allows obtaining pieces of audio that are consistent in their content as the speaker tends to take longer pauses to emphasize certain keywords [37], this is because a subject change is more apt to come after a break than in the middle of a continuous expression [38]. The reason to split the audio files into smaller chunks is that it will be easier to extract textual and acoustic features of small audio chunks rather than the longer audio file and those features can be further utilized in speech/ audio applications.

3.3.3 Automatic Speech Recognition Module

Automatic speech recognition (ASR) is seen as an essential part of human-computer interfaces build to use voice, to enable normal, universal, and widespread computing [39]. ASR refers to the method of transcribing an utterance, based on the waveform of the voice. It is an autonomous computer encoding and transcription mechanism for oral expression. A standard ASR program obtains speech input, analyzes them using a pattern, model, or algorithm, and produces a response typically in text type [40]. ASR is still a significant topic of study in the field of Natural Language Processing (NLP), but in the last couple of decades there have been significant improvements and many ASR tools have been developed to handle the speech and to achieve the best results. One such tool is the pocketsphinx[3] ASR, a lightweight open-source toolkit for speech recognition. Pocketsphinx is a python interface to CMU Spinx[4]. CMU Sphinx uses Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) to predict the phonemes in the utterance to specify the word or group of words spoken continuously [41].

In this model, we feed audio from our VAD model to pocketsphinx ASR

[3]https://github.com/cmusphinx/pocketsphinx
[4]https://cmusphinx.github.io/wiki/

and obtained the transcription of the input lecture videos. Since our audio input for ASR is split into audio chunks (from the VAD process), we also obtain the transcript as fragments. This process is conducted side by side with another process Acoustic Feature extractor so that inputs are identical for both our ASR and Acoustic Feature Extractor and output are also in the same shape.

3.3.4 Acoustic Feature Extractor Module

The general prospect of our suggested overall model is to extract two different features from the input lecture video, one being a textual feature in the transcript's form that we obtained using ASR, and the other is acoustic properties such as pitch, volume estimation of audio. These properties play an important role in defining audio and may help in further analysis. Since we already mentioned our input audio from VAD is in the form of audio chunks, these features depend entirely on those audio chunks. The combination of these smaller video fragments doesn't affect the features of the whole video but instead helps to better understand the lecture video. For this purpose, we used aubio[5], which is a set of algorithms and tools for marking and transforming music and sounds. It scans or listens to audio signals and tracks musical activities. The aubio functions are to segment audio file, pitch recognition, beat tapping, and creation of live audio midi streams.

In this process we feed the same audio chunks as we used in our ASR, those chunks being only voiced help our model to extract the exact acoustic properties which are useful. As we already described that the transcripts output from ASR will be in fragments, so does in this process. The final output from our Acoustic feature extractor is pitch, volume, pause rates, and the initial time of each audio chunk created by our VAD.

[5]https://github.com/aubio/aubio

3.3.5 Feature Aggregator Module

Up to this stage, our proposed model successfully extracts transcription and low-level acoustic features like pitch, volume, pause rates from the audio track of lecture video using our previously defined modules ASR and Acoustic Feature Extractor. But we need to aggregate the feature extraction results to be used by the segmentation module. So this module combines the two distinct features and feeds to the segmentation algorithm. In our model, this element also acts as the convergence point of two processes: the feature extraction process and the segmentation process.

3.3.6 Segmentation Module

In the lecture video, the segmentation algorithm is responsible for finding the series of partitions representing the subject boundaries with the audio track features. We have adopted the segmentation algorithm described in [30] with some modification to optimize the lecture video segmentation.

3.3.6.1 Multi-objective model:

The lecture video segmentation which we used is basically a multi-objective function. Here we consider the relationship of pitch and the volume [42], [43] i.e the mean loudness and mean fundamental frequency were correlated, so we must select the audio block accordingly to maximize the sum of the practical scores, while converting it as a topic and minimizing the number of digital partitions. Thus, the over-segmentation that will have the reverse result of a successful temporal segment is avoided. The utility score U_i of an audio chunk i is given by the equation:

$$U_i = \alpha(F_i + V_i) + \beta \cdot P_i + \gamma \cdot D_i \tag{5}$$

Where F_i, V_i, P_i are estimates of pitch, volume and pause rate respectively. These acoustic features are obtained from our previous module

Acoustic Feature extractor. And, D_i represents the cosine distance between the Word2vec representation of transcripts of audio chunks S_i and its two neighbors S_{i-1} and S_{i+1}, respectively. As we can see in Equation 6.

$$D_i = Dcos(i-1, i) + Dcos(i, i+1) \tag{6}$$

The constants α, β, and γare added for scaling purposes, which support not to prioritize one feature over another in the segmentation algorithm. Finally, our multi-objective function is given by:

$$maxT \sum_{i=1}^{n} U_i \cdot X_i - \sum_{i=1}^{n} X_i \tag{7}$$

where T is the solution set, an audio chunks subset that is chosen to optimize the Equation 7 as a topic transition. In addition, X_i is a decision variable of our problem, defined as:

$$X_i = \begin{cases} 1, & \text{if } S_i \in T \\ 0, & \text{if } S_i \notin T \end{cases}$$

From the multi-objective function we can represent segments of lecture video in terms of chromosomes as shown in Figure 10 below.

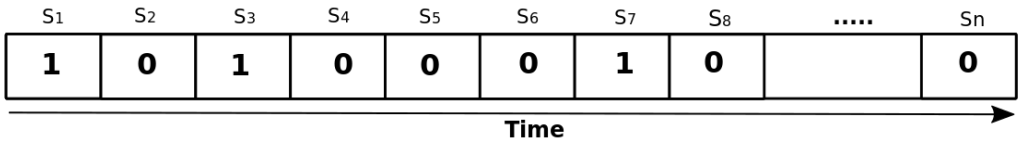


Figure 10: Representation of lecture video segment as a chromosome

3.3.6.2 Genetic Algorithm:

The Genetic Algorithm (GA) is a heuristic search approach based on Darwin's theory of natural evolution, which aims to find approximate solutions for search problems and optimization [44]. In GA the solution is called "individuals", together they form a "population", and each individual is represented by its chromosome, which typically makes up a one-dimensional

array, where each position of the array is one element of our problem. Moreover, every individual at GA has a fitness value, which shows how well the solution is for an individual problem. The fitness of an individual I_i in our case is given by Equation 7. The representation of individual chromosomes is assumed to be a binary array, in which the position i is equal to variable X_i in Equation 7.

We have an example in Figure 10 representing the segments of lecture video solution as chromosome. Here we can see the transitions in audio chunks S_1, S_3 and S_7. We can map it into a segment of lecture videos because audio chunks have timestamps of its appearance in the video.

The key attribute of an individual (solution) in GA have been clarified briefly. However, due to execution or the heterogeneity implemented, the GA measures responsible for converging solutions can be quite considerable. Since there are various GA varieties, we will clarify the one adopted in this project. The method of discovering solutions to the problem is:

1. We have a randomly created initial population.
2. A fitness function of each individual is assessed. And the individuals with the highest fitness score are submitted to local search.
3. Select individuals with better fitness scores for crossover. The chosen individuals are called "parents" in this stage They are chosen in pairs, and a new individual is formed from each pair of parents from their chromosomal combinations. In the next generation, the new individuals will be part of the population. We use the 2-point crossover approach [45] in this study.
4. Individuals with the lowest fitness level are excluded from the population.
5. Every individual has an opportunity to undergo a mutation, which is to alter a gene randomly in their DNA. This is an essential process to avoid premature convergence and to offer the variability of the

solutions. The mutation just flips a bit in our method. In other words, a gene chosen to be mutated with a value of 0 is converted into 1, and vice versa.

6. Repeat steps 2-4 by defining how many generations in the algorithm.

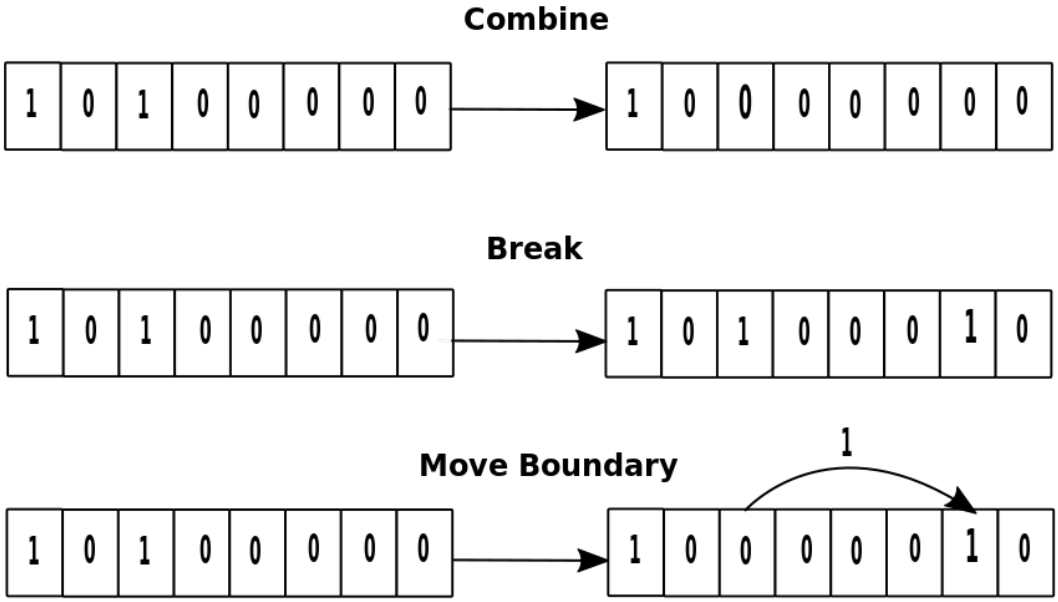


Figure 11: Illustration of local search movement

We also incorporated a local search process in our approach to adapting those movements to leading solutions that can enhance and discover more space. This method is expensive computationally, but we just required it to be used by the most influential individuals. We use an algorithm called Tabu Search (TS) [46] to do a local search. Each motion determines a neighborhood in TS such that the algorithm is attempted to find a better solution in the neighborhoods by the chained implementation of the movement to meet a stop criterion. In this work, we define 3 distinct movements: Combine, Break, and Move boundary. Two adjacent topics are merged into one in Combine movement. The Break movement is the opposite of Combine, a topic is divided into two new topics. Finally, in the Move Boundary movement, the topic boundary is moved to another audio section. The description of these movements is presented in Figure 11.

3.4 Dataset

In our development phase, we only used one lecture video for testing purpose but to assess our architecture that is not enough. To get a real insight into our proposed model, we need to make experiments on the collection of lecture videos. The proposed model is designed in such a way that it can handle single as well as multiple inputs, process them, and produce output simultaneously.

At first, we search for some dataset that have already been used in similar projects as ours. The motive for utilizing such a dataset is that we can save some time on creating ground truth parts of the input and rather more focus on the evaluation of the result. But we couldn't find any favorable dataset that can be used. We have therefore decided to create our own dataset. A total of 37 video lectures were taken from one of the Coursera courses. All the lecture videos had a different duration. The main reason for choosing these video lectures was because the lectures presentation format was well managed, and the Coursera also offers transcription (.txt) files, Web Video Text Tracks (.vtt), and one level of segmentation, which we can consider while creating a ground truth for further assessment. Table 2 is a list of lectures teaching different topics with different time duration and size. For ease, we renamed the original video name into ID format, other than that we haven't manipulated anything on these lecture videos.

Table 2: List of lecture videos used for evaluation

Video ID	**Original video name**	**Video length (mm:ss)**	**Video size (MB)**
Video_001	Welcome to this course and specialization	00:42	1.3
Video_002	Who we are	05:43	8.5

Video_003	Machine learning is changing the world	03:41	5.7
Video_004	Why a case study approach?	07:27	10
Video_005	Specialization overview	06:17	8.9
Video_006	How we got into ML	03:23	5.9
Video_007	Who is this specialization for?	04:01	5.5
Video_008	What you'll be able to do	00:57	1.7
Video_009	The capstone and an example intelligent application	06:31	7.6
Video_010	The future of intelligent applications	02:19	4.2
Video_011	Starting a Jupyter Notebook	05:30	5.4
Video_012	Creating variables in Python	07:15	6.9
Video_013	Conditional statements and loops in Python	08:08	7.8
Video_014	Creating functions and lambdas in Python	03:31	3.7
Video_015	Starting Turi Create & loading an Sframe	04:32	4.6
Video_016	Canvas for data visualization	04:09	4.1
Video_017	Interacting with columns of an Sframe	04:29	4.2
Video_018	Using .apply() for data transformation	05:17	5.1
Video_019	Predicting house prices: A case study in regression	01:22	1.7
Video_020	What is the goal and how might you naively address it?	03:47	3.9
Video_021	Linear Regression: A Model-Based Approach	05:34	5.2

Video_022	Adding higher order effects	04:11	4.1
Video_023	Evaluating overfitting via training/test split	06:19	6
Video_024	Training/test curves	04:22	3.9
Video_025	Adding other features	02:30	2.8
Video_026	Other regression examples	03:28	4.9
Video_027	Regression ML block diagram	05:55	5.4
Video_028	Loading & exploring house sale data	07:11	6.9
Video_029	Splitting the data into training and test sets	02:34	2.7
Video_030	Learning a simple regression model to predict house prices from house size	03:54	3.9
Video_031	Evaluating error (RMSE) of the simple model	02:29	2.7
Video_032	Visualizing predictions of simple model with Matplotlib	04:52	4.6
Video_033	Inspecting the model coefficients learned	01:18	1.6
Video_034	Exploring other features of the data	06:24	5.6
Video_035	Learning a model to predict house prices from more features	03:23	3.3
Video_036	Applying learned models to predict price of an average house	05:07	5.1
Video_037	Applying learned models to predict price of two fancy houses	07:20	7.2

The overall duration of lectures in our dataset is 2 hours, 45 minutes,

52 seconds and the total size is 182.6 MB and the videos are in MPEG-4 video (.mp4) format. The dataset used in this thesis are available at Google drive[6].

3.5 Ground Truth Creation and Evaluation Metrics

Ground truth heavily impacts the evaluation. It is therefore a very important step towards the overall concept of video segmentation. As discussed previously, our dataset comprises transcription (.txt), Web Video Text Tracks (.vtt), and one level of segmentation. Using all these we created ground truth manually, which can be used while evaluating our proposed model. Although Coursera provides one level of segmentation on their all lecture videos but we don't know on what ground this segmentation was defined and we are not sure that we can totally depend upon that, so we have to look at other sources as well to create our ground truth. A list of segments from Coursera for individual Lecture video is shown in Table 3.

Table 3: Segmentation from coursera for individual lecture

Video ID	Segmentation (mm:ss)	Number of segment
Video_001	00:00, 00:12	2
Video_002	00:00, 01:02, 01:54, 04:04	4
Video_003	00:00, 00:35	2
Video_004	00:00, 00:19, 00:35, 00:56, 01:21, 01:48, 03:33, 03:58, 05:34	9
Video_005	00:00, 03:25, 04:10, 05:46	4
Video_006	00:00, 00:52, 02:09, 03:08	4
Video_007	00:00, 00:11, 00:50, 02:48	4
Video_008	00:00	1
Video_009	00:00, 02:27, 03:12, 03:38, 04:07, 04:09, 04:39, 04:46, 05:06, 05:28, 05:37	11

[6]https://drive.google.com/drive/folders/1tjnRyoBh7OXYvmhlhTN29blPQRMNPKtw

Video_010	00:00	1
Video_011	00:00, 01:07, 02:00, 02:21, 03:15, 03:26, 03:29, 04:17, 05:21	9
Video_012	00:00, 01:22, 02:13, 02:22, 02:36, 02:48, 03:00, 03:27, 03:30, 03:41, 04:01, 04:48, 05:25, 05:39, 06:44	15
Video_013	00:00, 02:15, 02:46, 04:46, 06:01, 06:17, 06:25, 06:45, 07:08, 07:17	10
Video_014	00:00, 00:42, 01:35, 01:56, 02:16, 02:40, 03:18	7
Video_015	00:00, 00:42, 00:49, 02:05, 03:12, 03:20, 03:48	7
Video_016	00:00, 00:19, 00:46, 02:28	4
Video_017	00:00, 00:28, 00:45, 00:55, 01:36, 01:53, 02:06, 02:42	8
Video_018	00:01, 00:38, 00:58, 01:38, 01:41, 02:02, 04:24, 04:30, 04:48	9
Video_019	00:00	1
Video_020	00:03, 01:19, 01:28, 02:08, 03:08, 03:24	6
Video_021	00:00, 01:08, 01:44, 02:42, 02:56, 03:17, 04:21, 04:34, 04:41, 05:10	10
Video_022	00:00, 00:11, 00:56, 01:01, 01:11, 01:42, 02:02, 02:14, 02:21, 03:01, 03:16, 03:28, 03:40	13
Video_023	00:00, 00:53, 01:17, 01:47, 02:43, 02:59, 03:09, 03:35, 03:40, 04:53, 05:24, 05:40, 06:04	13
Video_024	00:00, 00:25, 00:39, 00:54, 03:16, 03:36, 04:00	7
Video_025	00:00, 00:43, 00:52	3

Video_026	00:00, 02:21	2
Video_027	00:00, 01:02, 01:13, 01:40, 02:12, 02:31, 02:37, 02:48, 02:54, 02:59, 03:37, 03:42, 03:45, 04:01, 05:27, 05:37	16
Video_028	00:00, 00:40, 01:38, 02:05, 03:42, 04:00, 04:02, 04:44, 04:48, 05:46, 06:01, 07:07	12
Video_029	00:00, 00:22, 00:42, 00:46, 01:25, 01:56, 02:30	7
Video_030	00:00, 00:25, 00:59, 01:45, 01:51, 02:05, 02:32, 02:41, 02:57, 03:12, 03:32	11
Video_031	00:00, 00:08, 00:38, 00:50, 01:32, 01:36	6
Video_032	00:00, 00:47, 01:17, 01:47, 02:01, 02:26, 02:46, 02:54, 03:00, 03:06, 03:32, 03:56, 04:36, 04:48	14
Video_033	00:00, 00:16, 00:20, 00:25, 01:14	5
Video_034	00:00, 00:28, 00:34, 01:03, 01:11, 01:20, 01:26, 01:32, 01:41, 02:04, 02:23, 02:31, 02:43, 03:40, 05:52	15
Video_035	00:00, 00:29, 00:41, 01:19, 01:33, 02:22, 02:25, 02:38	8
Video_036	00:00, 01:00, 01:36, 01:58, 02:37, 02:58, 03:34, 04:13, 04:18, 04:34, 05:03	11
Video_037	00:00, 00:24, 01:01, 02:03, 02:38, 03:39, 04:01, 04:19, 04:48, 04:51, 05:50, 06:01, 07:00	13

We have manually created the ground truth of correct segment boundaries by listening and analyzed the Web Video Text Tracks (WebVTT) file, which is used for the labeling of external timed text tracks for captioning video content [47]. It is the easiest method of subtitling video as it is usable for the screen reading applications and it also contains a text track with

related timing. Example of how the WebVTT file looks like is as follows:

WEBVTT

1
00:00:00.056 –>00:00:04.250
MUSIC

2
00:00:04.250 –>00:00:06.423
Welcome to the machine learning specialization and

3
00:00:06.423 –>00:00:09.210
this first course on the fundamentals of machine learning.

4
00:00:09.210 –>00:00:11.290
Were really excited to embark on this journey with you.

5
00:00:12.700 –>00:00:13.710
Happy?

6
00:00:13.710 –>00:00:14.280
>>We are.

7
00:00:14.280 –>00:00:15.640
Are you going to say who you are?

8
00:00:15.640 –>00:00:16.970
>>Oh, Im Carlos.

9
00:00:16.970 –>00:00:18.460

>>And Im Emily.

10
00:00:18.460 –>00:00:22.320
>>And together, were going to learn about applications of machine learning,

11
00:00:22.320 –>00:00:26.180
how to build machine learning systems, and how the algorithms behind them work, and

12
00:00:26.180 –>00:00:28.146
how to build those algorithms.

13
00:00:28.146 –>00:00:28.952
Algorithms.

14
00:00:28.952 –>00:00:33.760
>>LAUGH Were clearly just so excited about this course and this specialization.

15
00:00:33.760 –>00:00:36.540
We can bearly put the words together to describe it.

16
00:00:36.540 –>00:00:38.020
>>So, lets get going.

17
00:00:38.020 –>00:00:42.069
MUSIC

We retrieve the full sentences from every single lecture video from these

files and documented the starting time of those. In doing so, we take into account the observation that every time the segmentation on the lecture video begins from the start of a certain sentence [48], [49] i.e. the segment may consist of at least one complete sentence or combination of several sentences, but it always starts from the beginning of the sentence. And we also consider the original segmentation, which we get from coursera while creating the ground truth manually. The ground truth for evaluating our architecture is listed in the Table 4.

Table 4: Ground Truth for individual lecture video

Video ID	**Cue Count**	**Sentence Count**	**Segment Count**	**Start timings of segment (mm:ss)**
Video_001	17	13	4	00:00.056, 00:12.700, 00:28.952, 00:38.020
Video_002	109	77	12	00:00.000, 00:32.207, 01:02.640, 01:54.670, 02:40.020, 03:11.890, 03:43.870, 03:55.567, 04:04.160, 04:28.340, 04:55.619, 05:35.600
Video_003	60	34	8	00:00.463, 00:14.340, 00:35.590, 00:44.320, 01:58.210, 02:29.960, 03:18.840, 03:37.595

Video_004	122	67	33	00:00.000, 00:19.630, 00:35.340, 00:45.400, 00:56.320, 01:05.400, 01:21.730, 01:36.788, 01:48.350, 01:52.850, 02:03.900, 02:15.410, 02:26.290, 02:35.110, 02:47.963, 02:54.436, 03:33.180, 03:44.588, 03:58.050, 04:16.530, 04:18.720, 04:33.161, 04:36.630, 04:50.122, 05:02.100, 05:09.798, 05:18.835, 05:34.830, 05:46.920, 06:11.668, 06:53.760, 07:02.570, 07:23.691
Video_005	100	48	13	00:00.000, 00:13.750, 00:30.360, 03:25.240, 04:10.860, 05:05.118, 05:19.020, 05:30.340, 05:36.370, 05:38.290, 05:46.860, 05:55.020, 06:13.221
Video_006	66	38	11	00:00.000, 00:40.650, 00:52.080, 01:33.130, 02:05.650, 02:09.750, 02:29.630, 02:40.670, 03:08.740, 03:13.536, 03:19.539
Video_007	63	32	14	00:00.121, 00:11.570, 00:35.895, 00:50.370, 01:18.010, 01:26.370, 01:30.270, 01:49.150, 02:18.880, 02:48.310, 03:18.750, 03:38.310, 03:50.440,03:57.076
Video_008	18	10	1	00:00

Video_009	117	74	19	00:00.000, 00:18.550, 00:43.550, 00:50.550, 01:53.890, 02:27.570, 02:45.140, 03:08.680, 03:12.447, 03:26.800, 03:38.760, 04:07.940, 04:09.780, 04:23.440, 04:39.160, 04:46.930, 05:06.350, 05:28.478, 05:37.230
Video_010	52	42	10	00:00.633, 00:23.890, 00:40.110, 00:46.700, 00:55.310, 01:11.360, 01:34.710, 01:51.590, 02:03.550, 02:15.310
Video_011	95	66	21	00:00.000, 00:24.770, 00:45.000, 01:07.330, 01:16.000, 01:28.890, 01:42.230, 02:00.720, 02:06.350, 02:21.270, 02:52.660, 02:56.720, 03:15.280, 03:26.110, 03:29.980, 03:41.800, 04:00.950, 04:17.420, 04:31.318, 05:11.910, 05:21.070
Video_012	102	70	25	00:00.043, 00:18.531, 00:22.970, 00:43.685, 00:52.770, 01:06.300, 01:22.910, 01:26.030, 02:09.091, 02:16.170, 02:22.230, 02:29.180, 02:46.540, 03:00.410, 03:23.870, 03:27.490, 03:30.980, 03:41.720, 04:01.680, 04:24.210, 04:48.790, 05:25.490, 05:39.730, 06:37.820, 06:44.100

Video_013	124	97	38	00:00.093, 00:16.470, 00:35.750, 00:51.480, 01:01.770, 01:08.860, 01:17.490, 01:24.740, 01:30.500, 01:37.190, 01:43.970, 01:50.510, 02:15.780, 02:46.960, 02:57.420, 03:03.470, 03:05.220, 03:22.470, 03:28.220, 03:44.380, 03:47.550, 03:53.010, 03:55.470, 03:57.220, 04:23.538, 04:46.220, 04:52.500, 05:28.460, 05:30.130, 05:40.135, 05:57.380, 06:01.190, 06:17.240, 06:25.680, 06:45.590, 07:08.090, 07:17.250, 07:59.836
Video_014	51	29	12	00:00.000, 00:30.240, 00:42.200, 00:53.230, 01:35.550, 01:56.040, 02:14.790, 02:16.770, 02:40.430, 03:03.850, 03:18.610, 03:27.949
Video_015	66	47	18	00:00.726, 00:34.030, 00:42.790, 00:49.800, 01:17.460, 01:38.220, 01:54.940, 02:05.650, 02:11.920, 02:36.950, 02:46.170, 02:54.590, 02:59.240, 03:12.910, 03:20.630, 03:32.260, 03:48.640, 04:16.810
Video_016	61	43	14	00:00.025, 00:11.290, 00:19.830, 00:25.750, 00:33.530, 00:46.140, 01:02.321, 01:21.812, 02:06.210, 02:28.000, 03:04.370, 03:26.900, 03:44.890, 04:00.210

Video_017	69	53	16	00:00.000, 00:04.067, 00:28.145, 00:45.510, 00:55.830, 01:08.095, 01:21.542, 01:36.280, 01:53.550, 02:06.554, 02:22.412, 02:42.930, 02:53.591, 03:19.850, 03:33.981, 04:19.860
Video_018	79	53	19	00:01.042, 00:14.450, 00:38.180, 00:58.690, 01:17.681, 01:36.190, 01:38.880, 01:41.460, 01:53.380, 02:02.630, 02:31.100, 02:52.010, 02:58.962, 03:08.860, 03:33.190, 03:47.458, 04:07.100, 04:38.720, 04:48.705
Video_019	24	14	3	00:04.410, 00:11.650, 01:15.580
Video_020	56	35	12	00:03.961, 00:26.370, 01:03.200, 01:11.320, 01:19.050, 01:28.280, 01:43.187, 02:08.830, 02:46.342, 03:08.010, 03:19.500, 03:29.060
Video_021	84	53	16	00:00.383, 00:34.370, 01:08.560, 01:13.710, 01:27.470, 01:44.090, 02:29.010, 02:42.342, 02:56.950, 03:17.010, 03:50.470, 04:07.580, 04:21.620, 04:34.890, 04:41.390, 05:10.500

Video_022	76	61	23	00:00.148, 00:11.600, 00:33.950, 00:41.520, 00:44.944, 00:56.930, 01:01.180, 01:05.000, 01:11.010, 01:19.400, 01:23.060, 01:28.320, 01:29.990, 01:42.270, 01:51.910, 02:02.720, 02:14.590, 02:21.480, 03:01.500, 03:16.000, 03:28.760, 03:40.860, 04:08.037
Video_023	96	60	21	00:00.000, 00:53.630, 01:04.730, 01:17.240, 01:32.034, 01:47.050, 02:00.690, 02:16.080, 02:24.090, 02:43.270, 02:59.610, 03:09.020, 03:35.260, 03:40.920, 03:48.550, 04:31.714, 04:53.770, 05:24.840, 05:40.930, 06:04.520, 06:15.666
Video_024	66	37	15	00:00.000, 00:25.140, 00:34.200, 00:39.510, 00:54.470, 00:59.320, 01:37.936, 02:17.780, 02:34.308, 02:45.755, 03:04.480, 03:16.050, 03:36.028, 04:00.720, 04:19.256
Video_025	45	28	9	00:00.000, 00:10.620, 00:20.720, 00:33.650, 00:43.160, 00:52.140, 01:33.370, 01:50.260, 02:23.220
Video_026	54	25	10	00:00.209, 00:47.010, 01:23.290, 01:57.340, 02:14.275, 02:21.300, 02:34.230, 02:48.481, 03:15.078, 03:24.574

Video_027	87	51	21	00:00.000, 01:02.434, 01:09.330, 01:13.340, 01:25.000, 01:40.500, 02:12.133, 02:23.700, 02:31.705, 02:37.950, 02:48.820, 02:54.870, 02:59.610, 03:07.630, 03:20.080, 03:42.120, 03:45.880, 04:01.570, 04:52.960, 05:27.030, 05:37.700
Video_028	91	53	30	00:00.218, 00:09.374, 00:36.650, 00:40.630, 01:08.290, 01:20.453, 01:30.442, 01:38.750, 01:41.550, 01:46.150, 01:58.630, 02:05.510, 02:15.420, 02:22.670, 02:29.720, 02:41.560, 02:52.350, 03:19.430, 03:31.050, 03:34.580, 03:42.070, 03:48.861, 04:00.029, 04:02.750, 04:44.460, 04:48.960, 05:49.990, 06:01.590, 06:33.790, 07:07.385
Video_029	35	21	12	00:00.006, 00:22.040, 00:42.490, 00:46.450, 01:01.120, 01:22.285, 01:25.800, 01:32.420, 01:44.530, 01:56.930, 02:23.500, 02:30.366
Video_030	53	37	20	00:00.000, 00:25.510, 00:47.550, 00:49.930, 00:56.890, 00:59.590, 01:06.918, 01:37.730, 01:45.900, 01:51.189, 02:05.060, 02:21.020, 02:32.950, 02:41.980, 02:57.010, 03:12.830, 03:17.027, 03:22.400, 03:45.010, 03:50.422

Video_031	34	24	13	00:00.000, 00:08.430, 00:15.960, 00:28.900, 00:38.940, 00:47.210, 00:50.050, 01:03.230, 01:32.370, 01:36.140, 01:50.850, 02:01.496, 02:24.791
Video_032	64	45	18	00:00.133, 00:19.791, 00:47.150, 01:17.840, 01:47.930, 01:57.970, 02:01.690, 02:06.480, 02:26.190, 02:32.690, 02:46.320, 03:00.060, 03:06.090, 03:25.780, 03:56.530, 04:02.740, 04:36.670, 04:48.590
Video_033	19	17	7	00:00.265, 00:16.420, 00:20.900, 00:25.170, 00:38.436, 00:50.410, 01:14.515
Video_034	91	67	26	00:00.000, 00:28.350, 00:34.749, 00:46.120, 01:03.118, 01:11.290, 01:20.480, 01:26.210, 01:32.800, 01:41.310, 01:48.130, 02:04.990, 02:13.820, 02:23.710, 02:31.270, 02:43.290, 02:55.250, 03:40.475, 04:01.170, 04:07.230, 04:15.745, 04:55.130, 05:00.194, 05:44.970, 05:52.030, 06:04.794
Video_035	45	31	17	00:00.000, 00:23.462, 00:29.250, 00:41.040, 01:12.300, 01:19.280, 01:33.160, 01:51.535, 01:57.630, 02:05.423, 02:22.490, 02:32.870, 02:38.080, 02:46.470, 02:50.970, 02:52.830, 03:19.231

Video_036	67	46	24	00:00.025, 00:12.510, 00:51.180, 01:06.660, 01:19.251, 01:30.160, 01:48.430, 01:58.610, 02:11.050, 02:28.008, 02:37.862, 02:58.862, 03:07.090, 03:13.370, 03:20.473, 03:34.560, 03:46.370, 03:55.745, 04:13.430, 04:18.380, 04:34.300, 04:48.980, 04:54.520, 05:03.675
Video_037	110	90	31	00:00.000, 00:24.720, 00:28.584, 01:01.390, 01:20.181, 01:45.767, 02:03.370, 02:11.088, 02:22.630, 02:38.200, 02:44.620, 03:16.822, 03:36.620, 03:39.650, 03:49.328, 04:01.133, 04:19.352, 04:36.360, 04:48.540, 04:51.040, 05:01.868, 05:19.500, 05:21.137, 05:30.590, 05:38.021, 05:50.595, 06:01.020, 06:43.197, 06:52.911, 07:00.180, 07:10.720

In order to assess the feasibility of the proposed model, a thorough analysis of the output is important. For this thesis, we choose some metrics like precision, recall, and F-measure for the simple evaluation and quantitative comparison of the segmentation performance. Another reason for choosing these metrics is that we can compare with other proposed models, which we mentioned in the literature review section. The sample output from the prototype and the ground truth form the basis of these metrics. Precision is the ratio of the cumulative positive observations that are accurately predicted. Similarly, the Recall is the ratio of correctly predicted positive observations to all observations in the ground truth. And the F-score is the weighted average of Precision and Recall. Let S be the set of segments

belonging to our proposed method, and G be the segment sets of the ground truth, the evaluation metrics are defined by the Equation 8-Equation 10.

$$Precision = \frac{|S \cap G|}{|S|} \tag{8}$$

$$Recall = \frac{|S \cap G|}{|G|} \tag{9}$$

$$F_1 = 2.\frac{Precision.Recall}{Precision + Recall} \tag{10}$$

where,
$S \cap G$ (Matched Segment) = If starting timestamps of outcome segment and starting time of sentence in ground truth segment match.

We also check the performance of ASR to assess if the accuracy of the transcripts produced by speech recognition has any effect on our proposed model performance or not. To do so, we use a common metric of the performance of speech recognition, Word Error Rate (WER).

$$WER = \frac{S + D + I}{N} \tag{11}$$

where,
S is the number of substitutions word
D is the number of deleted word
I is the number of inserted word
N is the total number of spoken words

Chapter 4

Experiment and Results

In this chapter, we will first explain the necessary environment setup needed to run our proposed prototype for the lecture video segmentation model. By doing so, we are completely ready to illustrate the outcome of our system which is shown in experimental result section of this chapter.

4.1 Experimental Setup

In order to run our architecture, we used docker containers as it allows us to bundle and deploy an application containing all parts it needs, including libraries and other dependencies. Along with all the modules described in our implementation of lecture video segmentation section audio extractor, VAD, acoustic feature extractor, ASR, feature aggregator, and segmentation algorithm we also create containers of other additional modules required to successfully run our architecture such as API, message broker and databases. Besides these, we also have to do little set up on our local machine where we are performing our experiment. In our local machine, we need to get some models, the first one is the pocketsphinx toolkit and the second one is the word2vec model used by our segmentation algorithm.

Also, the segmentation algorithm has been fully implemented in python. The parameters of Equation 5 are chosen empirically so that the size range of each feature is the same and thus has the same importance in the utility

score. The proportional parameters α, β, and γ are set to 0.05, 1, and 10, respectively.

In the genetic algorithm, the number of generations is set to 1000, and the population size to 200 individuals. The best 35% of the individuals in a generation are chosen randomly in pairs for crossover. In the population, the likelihood of mutation is selected as 7%. In addition, we submit 35% of the best solution to the local search step.

4.2 Experimental Results

As we explained in our methodology section, our VAD uses the audio frames to generate entirely voiced audio chunks. Figure 12 shows the sequences of speech and non-speech frames generated by our VAD module. Our VAD labeled the speech frames as '1' and non-speech frames as '0'. If more than 90% of the frames are voiced in the window (as reported by the VAD) and in non triggered mode, the collector activates and begins collecting audio frames up to that point (i.e. positive values as seen in the figure). Then the collector waits until 90% of the frames in the window are unvoiced to untriggered. In the meantime, the collector begins collecting whatever audio frames can be collected up to that point (i.e. negative values in the figure). And this process continues until each frame is collected. Using these values our VAD produces the audio chunks which are used to extract the acoustic and textual features.

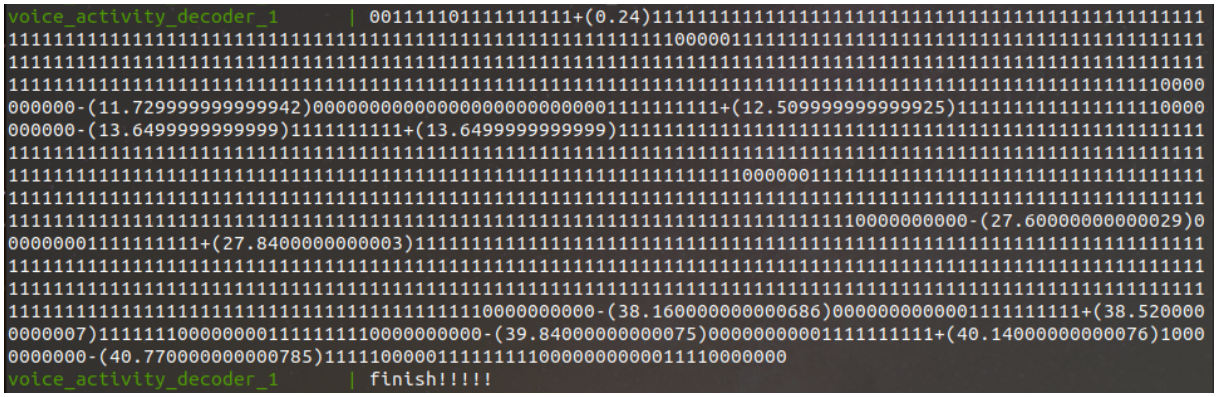


Figure 12: VAD processing output of single lecture video

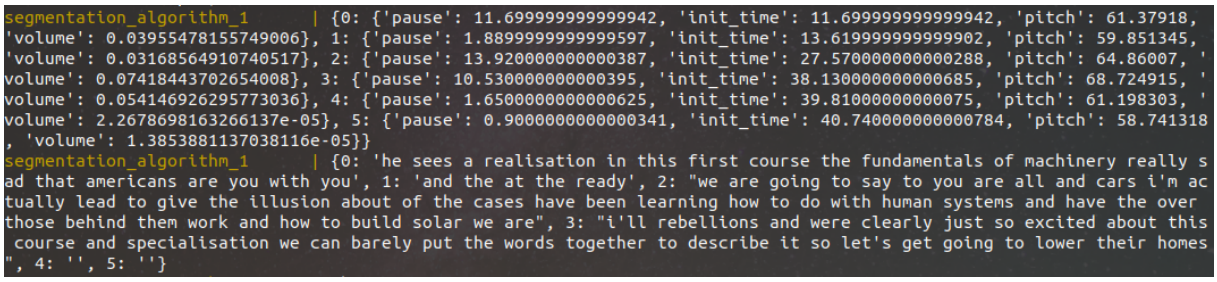


Figure 13: Segmentation algorithm processing input of single lecture video

Figure 13 displays the processing input of a single lecture video for our segmentation algorithm. The first part of the figure is the extracted acoustic features from our acoustic feature extractor module which has low-level acoustic features of individual audio chunks such as pause, initial timing of audio chunk, pitch, and volume. The second part is the textual feature extracted from our ASR module which includes the transcripts of each audio chunk. Since our acoustic feature extractor as well as ASR modules use the same audio chunks, we can see that the acoustic and textual features extracted are in the identical shape. Our segmentation algorithm uses thes features to generates the multi-objective function as mentioned in Equation 7, which our genetic algorithm uses to perform the final stage of lecture video segmentation.

The overall outcome of our proposed model can be seen in Figure 14 and Figure 15. Figure 14 shows the processing outcome which contains a set of final segments of individual lecture videos. When there are more than one lecture video, our model randomly selects the videos for processing so these outcomes are not in the order. Figure 15 shows the end results of our model, which is the combined outcome of all lecture videos. The final outcome is in the format of:

[{‘Video Name’: ‘Foldername/videoname.ext’, ‘Segmentation’: {‘segments’: [‘segment1’, ‘segment2’,...]}},{‘Video Name’: ‘Foldername/videoname.ext’, ‘Segmentation’: {‘segments’: [‘segment1’, ‘segment2’,...]}},...]

The start timing of segments and number of segments from each lecture video of our dataset is listed in Table 5.

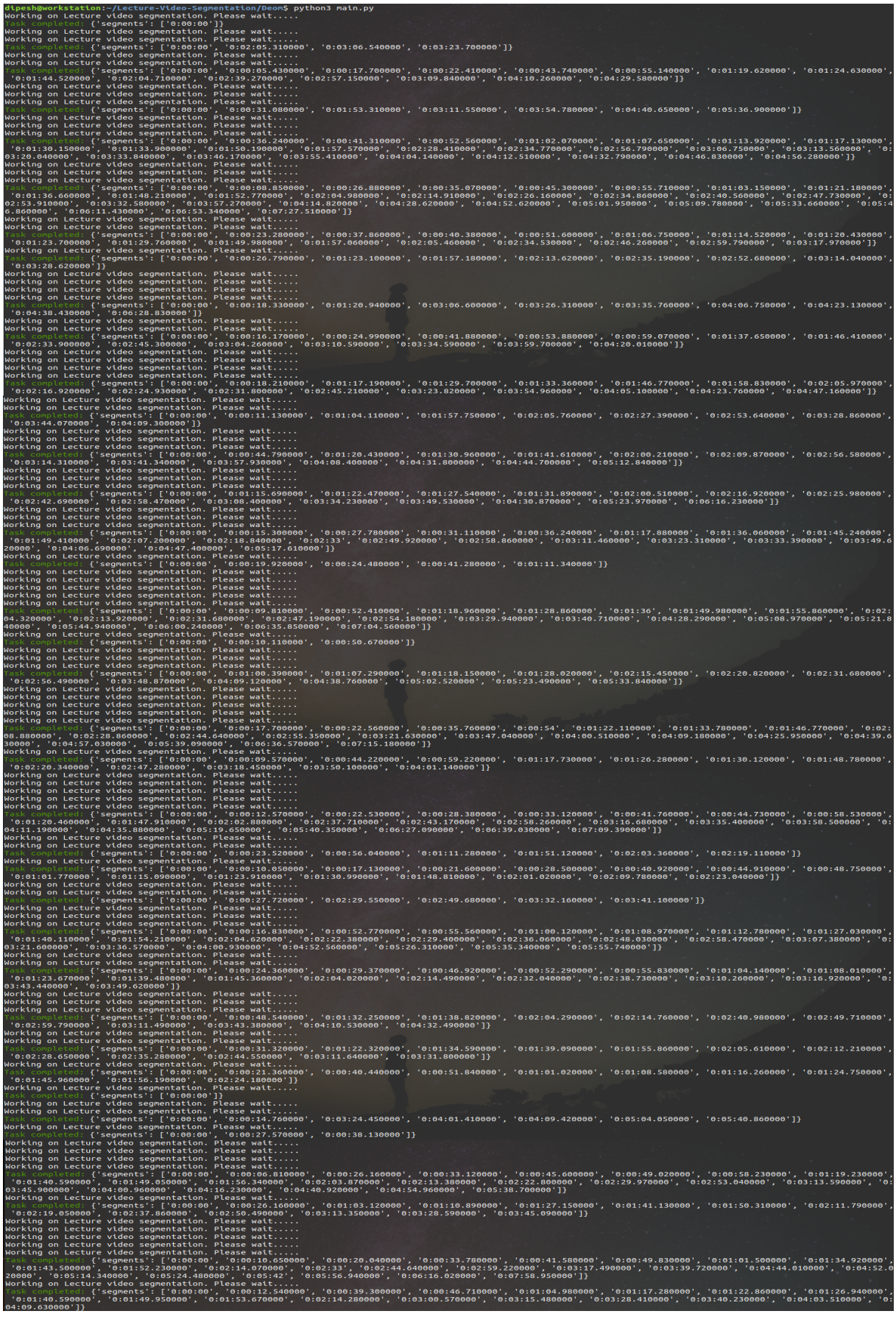


Figure 14: Output of the proposed model displaying the individual lecture videos processing

Figure 15: Final output of the proposed model displaying the combined result of all lecture videos

Table 5: Start timing of segment for individual lecture video from the proposed model

Video ID	Start timings of segments obtained (mm:ss)	Number of segment
Video_001	00:00.000000, 00:11.700000, 00:38.130000	3
Video_002	00:00.000000, 00:31.080000, 01:53.310000, 03:11.550000, 03:54.780000, 04:22.500000, 04:40.650000, 05:36.900000	8
Video_003	00:00.000000, 00:27.720000, 00:34.560000, 02:49.680000, 03:32.160000, 03:41.100000	6

Video_004	00:00.000000, 00:08.850000, 00:26.880000, 00:35.070000, 00:45.300000, 00:55.710000, 01:03.150000, 01:21.180000, 01:36.660000, 01:48.210000, 01:52.770000, 02:04.980000, 02:14.910000, 02:26.160000, 02:34.860000, 02:40.560000, 02:47.730000, 02:53.910000, 03:32.580000, 03:57.270000, 04:14.820000, 04:28.620000, 04:36.390000, 04:52.620000, 05:01.950000, 05:09.780000, 05:33.660000, 05:46.860000, 06:11.430000, 06:53.340000, 07:27.510000	31
Video_005	00:00.000000, 00:14.760000, 03:24.450000, 04:01.410000, 04:09.420000, 05:04.050000, 05:40.860000	7
Video_006	00:00.000000, 02:05.310000, 03:06.540000, 03:23.700000	4
Video_007	00:00.000000, 00:09.570000, 00:44.220000, 00:59.220000, 01:17.730000, 01:26.280000, 01:30.120000, 01:48.780000, 02:20.340000, 02:47.280000, 03:18.450000, 03:50.100000, 04:01.140000	13
Video_008	00:00.000000	1
Video_009	00:00.000000, 00:18.330000 01:20.940000, 03:06.600000, 03:26.310000, 03:35.760000, 04:06.750000, 04:23.130000, 04:38.430000, 06:28.830000	10
Video_010	00:00.000000, 00:23.520000, 00:56.040000, 01:11.280000, 01:51.120000, 02:03.360000, 02:19.110000	7

Video_011	00:00.000000, 00:44.790000, 01:20.430000, 01:30.960000, 01:41.610000, 02:00.210000, 02:09.870000, 02:56.580000, 03:14.310000, 03:41.340000, 03:57.930000, 04:08.400000, 04:31.800000, 04:44.700000, 05:12.840000	15
Video_012	00:00.000000, 00:17.700000, 00:22.560000, 00:35.760000, 00:54, 01:22.110000, 01:33.780000, 01:46.770000, 02:08.880000, 02:28.860000, 02:44.640000, 02:55.350000, 03:21.630000, 03:47.040000, 04:00.510000, 04:09.180000, 04:25.950000, 04:39.630000, 04:57.030000, 05:39.090000, 06:36.570000, 07:15.180000	22
Video_013	00:00.000000, 00:10.650000, 00:20.040000, 00:33.780000, 00:41.580000, 00:49.830000, 01:01.500000, 01:34.920000, 01:43.500000, 01:52.230000, 02:14.070000, 02:33, 02:44.640000, 02:59.220000, 03:17.490000, 03:39.720000, 04:44.010000, 04:52.020000, 05:24.480000, 05:42, 05:56.940000, 06:16.020000, 07:58.950000	23
Video_014	00:00.000000, 00:31.320000, 01:22.320000, 01:34.590000, 01:39.090000, 01:55.860000, 02:05.610000, 02:12.210000, 02:28.650000, 02:35.280000, 02:44.550000, 03:11.640000, 03:31.800000	13

Video_015	00:00.000000, 00:48.540000, 01:32.250000, 01:38.820000, 02:04.290000, 02:14.760000, 02:40.980000, 02:49.710000, 02:59.790000, 03:11.490000, 03:43.380000, 04:10.530000, 04:32.490000	13
Video_016	00:00.000000, 00:11.130000, 01:04.110000, 01:57.750000, 02:05.760000, 02:27.390000, 02:53.640000, 03:28.860000, 03:44.070000, 04:09.300000	10
Video_017	00:00.000000, 00:05.430000, 00:17.700000, 00:22.410000, 00:43.740000, 00:55.140000, 01:19.620000, 01:24.630000, 02:04.710000, 02:39.270000, 02:57.150000, 03:09.840000, 03:55.320000, 04:10.260000, 04:29.580000	16
Video_018	00:00.000000, 00:15.300000, 00:27.780000, 00:31.110000, 00:36.240000, 01:17.880000, 01:36.060000, 01:45.240000, 01:49.410000, 02:07.200000, 02:18.840000, 02:33, 02:49.920000, 02:58.860000, 03:11.460000, 03:23.310000, 03:33.390000, 03:49.620000, 04:06.690000, 04:47.400000, 05:17.610000	21
Video_019	00:00.000000	1
Video_020	00:00.000000, 00:26.160000, 01:03.120000, 01:10.890000, 01:27.150000, 01:41.130000, 01:50.310000, 02:11.790000, 02:19.050000, 02:37.860000, 02:50.490000, 03:13.350000, 03:28.590000, 03:45.090000	14

Video_021	00:00.000000, 01:00.390000, 01:07.290000, 01:18.150000, 01:28.020000, 02:15.450000, 02:20.820000, 02:31.680000, 02:56.490000, 03:10.890000, 03:48.870000, 04:09.120000, 04:38.760000, 05:02.520000, 05:23.490000, 05:33.840000	16
Video_022	00:00.000000, 00:12.540000, 00:39.300000, 00:46.710000, 01:04.980000, 01:17.280000, 01:22.860000, 01:26.940000, 01:40.590000, 01:49.950000, 01:53.670000, 02:14.280000, 03:00.570000, 03:15.480000, 03:28.410000, 03:40.230000, 04:03.510000, 04:09.630000	18
Video_023	00:00.000000, 01:15.690000, 01:22.470000, 01:27.540000, 01:31.890000, 02:00.510000, 02:25.980000, 02:42.690000, 02:58.470000, 03:08.400000, 03:34.230000, 03:49.530000, 04:30.870000, 05:23.970000, 06:16.230000	15
Video_024	00:00.000000, 00:16.170000, 00:24.990000, 00:41.880000, 00:53.880000, 00:59.070000, 01:37.650000, 01:46.410000, 02:33.900000, 02:45.300000, 03:04.260000, 03:10.590000, 03:34.590000, 03:59.700000, 04:20.010000	15
Video_025	00:00.000000, 00:10.110000, 00:50.670000	3
Video_026	00:00.000000, 00:26.790000, 01:23.100000, 01:57.180000, 02:13.620000, 02:35.190000, 02:52.680000, 03:14.040000, 03:28.620000	9

Video_027	00:00.000000, 00:16.830000, 00:52.770000, 00:55.560000, 01:00.120000, 01:08.970000, 01:12.780000, 01:27.030000, 01:40.110000, 01:54.210000, 02:04.620000, 02:22.380000, 02:29.400000, 02:36.060000, 02:48.030000, 02:58.470000, 03:07.380000, 03:21.600000, 03:36.570000, 04:00.930000, 04:52.560000, 05:26.310000, 05:35.340000, 05:55.740000	24
Video_028	00:00.000000, 00:09.810000, 00:52.410000, 01:18.960000, 01:28.860000, 01:36, 01:49.980000, 01:55.860000, 02:04.320000, 02:13.920000, 02:31.680000, 02:47.190000, 02:54.180000, 03:29.940000, 03:40.710000, 04:28.290000, 05:08.970000, 05:21.840000, 05:44.940000, 06:00.240000, 06:35.850000, 07:04.560000	22
Video_029	00:00.000000, 00:21.360000, 00:40.440000, 00:45.180000, 00:51.840000, 01:01.020000, 01:08.580000, 01:16.260000, 01:24.750000, 01:45.960000, 01:56.190000, 02:24.180000	12
Video_030	00:00.000000, 00:24.360000, 00:29.370000, 00:46.920000, 00:52.290000, 00:55.830000, 01:04.140000, 01:08.010000, 01:23.670000, 01:39.480000, 01:45.360000, 02:04.020000, 02:14.490000, 02:32.040000, 02:38.730000, 03:10.260000, 03:16.920000, 03:43.440000, 03:49.620000	19

Video_031	00:00.000000, 00:10.050000, 00:17.130000, 00:21.600000, 00:28.500000, 00:40.920000, 00:44.910000, 00:48.750000, 01:01.770000, 01:15.090000, 01:23.910000, 01:30.990000, 01:48.810000, 02:01.020000, 02:09.780000, 02:23.040000	16
Video_032	00:00.000000, 00:18.210000, 01:17.190000, 01:29.700000, 01:33.360000, 01:46.770000, 01:58.830000, 02:05.970000, 02:16.920000, 02:24.930000, 02:31.800000, 02:45.210000, 03:23.820000, 03:54.960000, 04:05.100000, 04:23.760000, 04:47.160000	17
Video_033	00:00.000000, 0:00:19.920000, 00:24.480000, 00:41.280000, 01:11.340000	5
Video_034	00:00.000000, 00:06.810000, 00:26.160000, 00:33.120000, 00:45.600000, 00:49.020000, 00:58.230000, 01:19.230000, 01:40.590000, 01:49.050000, 01:56.340000, 02:03.870000, 02:13.380000, 02:22.800000, 02:29.970000, 02:53.040000, 03:13.590000, 03:45.900000, 04:00.960000, 04:16.230000, 04:40.920000, 04:54.960000, 05:38.700000	23
Video_035	00:00.000000, 00:23.280000, 00:37.860000, 00:40.380000, 00:51.600000, 01:06.750000, 01:14.520000, 01:20.430000, 01:23.700000, 01:29.760000, 01:49.980000, 01:57.060000, 02:05.460000, 02:34.530000, 02:46.260000, 02:59.790000, 03:17.970000	17

Video_036	00:00.000000, 00:36.240000, 00:41.310000, 00:52.560000, 01:02.070000, 01:07.650000, 01:13.920000, 01:17.130000, 01:30.150000, 01:33.900000, 01:50.190000, 01:57.570000, 02:28.410000, 02:34.770000, 02:56.790000, 03:06.750000, 03:13.560000, 03:20.040000, 03:33.840000, 03:46.170000, 03:55.410000, 04:04.140000, 04:12.510000, 04:32.790000, 04:46.830000, 04:56.280000	26
Video_037	00:00.000000, 00:12.570000, 00:22.530000, 00:28.380000, 00:33.120000, 00:41.760000, 00:44.730000, 00:58.530000, 01:20.460000, 01:47.910000, 02:02.880000, 02:37.710000, 02:43.170000, 02:58.260000, 03:16.680000, 03:35.400000, 03:58.500000, 04:11.190000, 04:35.880000, 05:19.650000, 05:40.350000, 06:27.090000, 06:39.030000, 07:09.390000	24
Total		**518**

We compare the start timing of segments from the output of our model and the ground truth data i.e. Table 5 and Table 4 respectively. While comparing we consider a margin of +/- 2.5 seconds.

Table 6: Performance of our proposed model

Video ID	**Number of segments**	**Number of matched segments**	**Precision**	**Recall**	F_1
Video_001	3	3	1.00	0.75	0.86
Video_002	7	6	0.75	0.50	0.60
Video_003	6	2	0.33	0.25	0.29
Video_004	31	26	0.84	0.79	0.81

Video_005	7	5	0.71	0.38	0.50
Video_006	4	3	0.75	0.27	0.40
Video_007	13	10	0.77	0.71	0.74
Video_008	1	1	1.00	1.00	1.00
Video_009	10	8	0.80	0.42	0.55
Video_010	7	6	0.86	0.60	0.71
Video_011	15	11	0.73	0.52	0.61
Video_012	22	13	0.59	0.52	0.55
Video_013	23	15	0.65	0.39	0.49
Video_014	13	5	0.38	0.42	0.40
Video_015	13	7	0.54	0.39	0.45
Video_016	10	7	0.70	0.50	0.58
Video_017	16	6	0.40	0.38	0.39
Video_018	21	13	0.62	0.68	0.65
Video_019	1	1	1.00	0.33	0.50
Video_020	14	8	0.57	0.67	0.62
Video_021	16	7	0.44	0.44	0.44
Video_022	18	16	0.89	0.70	0.78
Video_023	15	13	0.87	0.62	0.72
Video_024	15	11	0.73	0.73	0.73
Video_025	3	3	1.00	0.33	0.50
Video_026	9	6	0.67	0.60	0.63
Video_027	24	17	0.71	0.81	0.76
Video_028	22	13	0.59	0.43	0.50
Video_029	12	9	0.75	0.75	0.75
Video_030	19	14	0.74	0.70	0.72
Video_031	16	13	0.81	1.00	0.90
Video_032	17	13	0.76	0.72	0.74
Video_033	5	3	0.60	0.43	0.50
Video_034	23	16	0.70	0.62	0.65

Video_035	17	12	0.71	0.71	0.71
Video_036	26	19	0.73	0.79	0.76
Video_037	24	14	0.58	0.45	0.51
Total	**518**	**355**	**0.69**	**0.58**	**0.63**

Table 7: Execution time and WER of our proposed model

Video ID	**Duration to execution**	**S+D+I**	**N**	**WER**
Video_001	01:07	66	99	0.67
Video_002	09:32	418	902	0.46
Video_003	03:01	172	606	0.28
Video_004	04:01	232	1238	0.19
Video_005	04:01	335	996	0.34
Video_006	03:01	323	591	0.55
Video_007	02:01	90	624	0.14
Video_008	00:41	48	149	0.32
Video_009	04:01	394	1175	0.34
Video_010	02:00	245	379	0.65
Video_011	04:00	401	841	0.48
Video_012	05:00	483	936	0.52
Video_013	06:01	630	1090	0.58
Video_014	03:00	216	487	0.44
Video_015	03:00	330	626	0.53
Video_016	03:00	331	614	0.54
Video_017	03:00	313	581	0.54
Video_018	03:00	349	728	0.48
Video_019	01:00	39	229	0.17
Video_020	02:00	126	601	0.21
Video_021	03:00	213	790	0.27
Video_022	03:00	207	693	0.30

Video_023	03:30	235	1036	0.23
Video_024	03:04	199	612	0.33
Video_025	02:00	70	445	0.16
Video_026	02:01	87	561	0.16
Video_027	03:34	174	882	0.20
Video_028	04:01	481	963	0.50
Video_029	02:00	211	361	0.58
Video_030	03:00	241	469	0.51
Video_031	02:00	127	276	0.46
Video_032	04:00	353	634	0.56
Video_033	02:00	76	175	0.43
Video_034	05:00	463	836	0.55
Video_035	03:00	223	400	0.56
Video_036	03:27	332	626	0.53
Video_037	04:45	494	881	0.56
Total	**01:59:53**	**9727**	**24132**	**0.40**

The performance of our proposed module is calculated based on the evaluation metrices defined in the methodology chapter and listed in Table 6 and in the same way WER and execution time of individual lecture videos are listed in Table 7. We will more discussed about all of our result in next chapter.

Chapter 5

Discussion

In this chapter, we will evaluate and discuss the overall aspect of the experiment performed and obtained results. And also the answers to the research question for this thesis will be briefly described.

In this study, we have successfully developed a segmentation system for lecture videos based only on the speech content, and the system seems promising looking over the result obtained. First, we have investigated the steps required for the development of a lecture video segmentation pipeline. This pipeline serves as a baseline for a prototype, which is the required outcome of our research. Then, we have applied various open-source tools and algorithms to find the best solution for the specific case for segmenting speech content of lecture video. Secondly, the recorded lectures of MOOC platforms were used to generate a dataset and the ground truth was also defined to interpret the outcome of the proposed prototype. These activities lead to answers all the related research questions of this thesis, which are here for the recall:

1. How can we use speech content of lecture video to determine the transition of segments?
2. How can we use state of art tools to segment the lecture video based on the speech?

Our prototype relies exclusively on the audio track of the lecture video to determine the outcome as segments of the video. The response to our first research question is obtained since we can see the outcome of our system. Furthermore, we can compare our outcome with different cases to analyze the effectiveness of our system. Some of the similar work done in the field of lecture video segmentation based on speech content as presented in this thesis is described in the literature review section above. Table 8 shows the comparison between our system and other similar systems.

Table 8: Comparison between our system and other systems

Method	Precision	Recall	F-score
Our Proposed System	0.690	0.580	0.630
System 1 [29]	0.465	0.491	0.477
System 2 [30]	0.400	0.480	0.400

This comparison clearly shows our model outperformed both of the other similar systems in all three metrics Precision, Recall, and F-score. As we already mentioned these metrics are very important factors based on an understanding and measure of relevance. Since we have a higher score on both Precision and Recall than other systems, this means our system is returning accurate results, as well as returning a majority of all positive results. With all these factors we can say our proposed method can efficiently work with the speech content to segment the lecture video.

And the answer to the second research question is achieved with the development of a prototype system based on the pipeline structure by using the start of art open-source tools. The development of the prototype has been carried out based on a software engineering architecture design where we implement the pipeline approach. We used Python bindings for FFmpeg to extract audio tracks from the input lecture video and then used WebRTC VAD module to detect non voiced portion in the speech. From this point, the audio feeds simultaneously to pocketsphinx ASR to extract

transcripts and aubio to extract acoustic features. Then we finally used Word2vec algorithm and Genetic algorithm to segment the lecture video entirely based on the speech content only.

Apart from answering only these research questions, we try to evaluate our proposed system as much as we can based on different cases, like the total execution time is almost 2 hours for our dataset. The compilation time may depend upon different factors like the size of the video, the number of audio chunks of the video but on average, our system takes around 45 seconds of execution time to compile a minute long lecture video. This may be a bit long compilation time if we have a huge dataset. For example, if we have the total duration of lecture videos around 100 hours it will take us approximately 75 hours to execution only for one time. This is totally not feasible while evaluating. That's why we limit our dataset to the current size of a diverse set of 37 lecture videos with different size and duration. Furthermore, dataset creation, analyzing each video, and creating ground truth manually is a time-consuming task. However the more example there are for the system, the more accurate evaluation will be provided. Thus, providing more number of Lectures videos to the dataset will increase the understanding and performance of the system. Moreover, the assessment of generalization performance will be more reliable. In the same way, we also calculate the WER of our ASR module is around 0.4 which is a bit higher in number but we don't see any adverse effect of the WER in our proposed system. Although they are not related to each other we notice that both the outcome performance and the WER are not that much in balance i.e the range has high differences. We can see the execution time of individual videos and WER in the Table 7.

Although we strive, with various tools and criteria, to test our entire system as much as possible. However, we could not evaluate all resources in modules due to the complicity and time constraints. But in case of genetic algorithm parameters, we perform numbers of trial based on the various parameters like population size, mutation rate, crossover rate, and number

of generations however we only get negligible performance differences so we eventually pick the best performing parameters. Basically, we run our model with the following different parameters:

- Population size of 100, 200, 300 and 400
- Mutation rate 0.030, 0.050, 0.065, 0.070, 0.075 and 0.080
- Number of generations 250, 500, 750, 1000 and 1250
- Crossover rate 0.50, 0.40, 0.350, 0.30 and 0.25

Chapter 6

Conclusion and Recommendation

The closing comments on the thesis are presented in this chapter, providing an overview and conclusion of the thesis endeavor. The thesis is examined in terms of both its contribution to the field and its limitations. And also based on the experiments and outcomes, some recommendations are made with regard to aspects that could be explored in future research.

We designed and tested a system for lecture video segmentation, which provides efficient results based on the speech content in a lecture video in this thesis. The system is capable of using open source tools and algorithms like Audio extractor, VAD, ASR, Acoustic feature extractor, segmentation algorithms so it is easily and freely available, and no problem in utilizing these. The proposed system is fully designed to handle any numbers of lecture videos and as well as no restriction in size and duration of videos, but we have to take consideration of execution time accordingly.

Through our experiments, we showed evidence that the proposed method can segment the video lectures by only utilizing the speech content. The outcome of the experiments performed shows the effectiveness of our proposed method since it surpasses the outcome of comparison models in all the 3 selected metrics Precision, Recall, and F-score.

Although our current system has some promising results, there are lots of

things to improve and update to make it more effective for maximum use. To do so, the following aspects could be explored in future work:

1. Implementation of topic segmentation.
2. To make architecture faster and efficient.
3. To analyze the performance of architecture by applying different parameters.
4. To make the end results with graphical representation automatically.
5. To analyze how an end-user can utilize this architecture in a real-word scenario.

This thesis should be only considered our small contribution and the first step towards a very important topic of content-based search and retrieval. This is a vast area of research to explore and we hope in the near future lot more research work will be performed in this area.

Bibliography

[1] H. W. Agius and M. C. Angelides, "Developing Knowledge-Based Intelligent Multimedia Tutoring Systems Using Semantic Content-Based Modelling," *Artificial Intelligence Review*, vol. 13, pp. 55–83, Feb. 1999.

[2] C. Baltes, "The E-learning balancing act: training and education with multimedia," *IEEE MultiMedia*, vol. 8, pp. 16–19, Oct. 2001. Conference Name: IEEE MultiMedia.

[3] S. Naidu, *E-Learning: A Guidebook of Principles, Procedures and Practices*. Commonwealth Educational Media Centre for Asia (CEMCA), 2006. Accepted: 2014-12-09T11:27:40Z.

[4] D. Chand and H. Ogul, "Content-Based Search in Lecture Video: A Systematic Literature Review," in *2020 3rd International Conference on Information and Computer Technologies (ICICT)*, pp. 169–176, Mar. 2020.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv:1301.3781 [cs]*, Sept. 2013. arXiv: 1301.3781.

[6] H. Yang and C. Meinel, "Content Based Lecture Video Retrieval Using Speech and Video Text Information," *IEEE Transactions on Learning Technologies*, vol. 7, pp. 142–154, Apr. 2014. Conference Name: IEEE Transactions on Learning Technologies.

[7] P. Chivadshetti, K. Sadafale, and K. Thakare, "Content based video retrieval using integrated feature extraction and personalization of results," in *2015 International Conference on Information Processing (ICIP)*, pp. 170–175, Dec. 2015.

[8] D. Zhang and J. Nunamaker, "A natural language approach to content-based video indexing and retrieval for interactive e-learning," *IEEE Transactions on Multimedia*, vol. 6, pp. 450–458, June 2004. Conference Name: IEEE Transactions on Multimedia.

[9] V. Balasubramanian, S. G. Doraisamy, and N. K. Kanakarajan, "A multimodal approach for extracting content descriptive metadata from lecture videos," *Journal of Intelligent Information Systems*, vol. 46, pp. 121–145, Feb. 2016.

[10] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys*, vol. 40, pp. 5:1–5:60, May 2008.

[11] A. Nasreen and S. G, "Parallelizing Multi-featured Content Based Search and Retrieval of Videos through High Performance Computing," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 5, p. 214, Jan. 2017.

[12] B. V. Patel and B. B. Meshram, "Content based video retrieval systems," *International Journal of UbiComp*, vol. 3, pp. 13–30, Apr. 2012. arXiv: 1205.1641.

[13] A. Y. Kothawade and D. R. Patil, "A Survey on Automatic Video Lecture Indexing," vol. 02, no. 02, p. 7.

[14] P. Eruvaram, K. Ramani, and C. S. Bindu, "An experimental comparative study on slide change detection in lecture videos," *International Journal of Information Technology*, vol. 12, pp. 429–436, June 2020.

[15] H. Yang, C. Oehlke, and C. Meinel, "German Speech Recognition: A Solution for the Analysis and Processing of Lecture Recordings,"

in *2011 10th IEEE/ACIS International Conference on Computer and Information Science*, pp. 201–206, May 2011.

[16] A. Sugandhi and D. Sharma, "Content based Video Retrieval using Text Annotation and Low Level Features Technique," *International Journal of Computer Applications*, vol. 145, pp. 11–16, July 2016.

[17] A. Sonawane, D. Mali, N. Shaikh, S. Shriya, and A. Gaidhani, "Sand search engine," *International Journal of Computer Applications & Information Technology*, vol. 8, no. 1, 2015.

[18] D. Patil and M. A. Potey, "Survey of Content Based Lecture Video Retrieval," *International Journal of Computer Trends and Technology*, vol. 19, no. 1, pp. 5–8, 2015. Publisher: Seventh Sense Research Group.

[19] G. Vigneshwari, "A Survey On Content Based Lecturing Video Retrieval," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 11, pp. 275–282, 2014.

[20] N. Purswani, R. Ramrakhyani, M. Makhija, and C. Mumbai, "Content based Multimedia Retrieval using Automatic Speech Recognition," *International Journal of Computer Applications*, vol. 118, no. 2, 2015.

[21] G. Vigneshwari and A. N. M. Juliet, "Optimized searching of video based on speech and video text content," in *2015 International Conference on Soft-Computing and Networks Security (ICSNS)*, pp. 1–4, Feb. 2015.

[22] "Pacify based Video Retrieval System," *International Journal for Rapid Research in Engineering Technology and Applied Science*, vol. 2, no. 1, 2016.

[23] B. Zhao, S. Xu, S. Lin, R. Wang, and X. Luo, "A New Visual Interface for Searching and Navigating Slide-Based Lecture Videos," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 928–933, July 2019. ISSN: 1945-788X.

[24] B. Urala Kota, K. Davila, A. Stone, S. Setlur, and V. Govindaraju, "Generalized framework for summarization of fixed-camera lecture videos by detecting and binarizing handwritten content," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 22, pp. 221–233, Sept. 2019.

[25] N. Poornima and B. Saleena, "Multi-modal features and correlation incorporated Naive Bayes classifier for a semantic-enriched lecture video retrieval system," *The Imaging Science Journal*, vol. 66, pp. 263–277, July 2018. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/13682199.2017.1419549.

[26] R. R. Shah, Y. Yu, A. D. Shaikh, and R. Zimmermann, "TRACE: Linguistic-Based Approach for Automatic Lecture Video Segmentation Leveraging Wikipedia Texts," in *2015 IEEE International Symposium on Multimedia (ISM)*, pp. 217–220, Dec. 2015.

[27] H. H.-S. Ip and S.-L. Chan, "Hypertext-assisted video indexing and content-based retrieval," in *Proceedings of the eighth ACM conference on Hypertext - HYPERTEXT '97*, (Southampton, United Kingdom), pp. 232–233, ACM Press, 1997.

[28] N. Kanedera, A. Sumida, T. Ikehata, and T. Funada, "Subtopic segmentation in the lecture speech," in *INTERSPEECH*, pp. 1821–1824, 2004.

[29] D. Galanopoulos and V. Mezaris, "Temporal Lecture Video Fragmentation Using Word Embeddings," in *MultiMedia Modeling* (I. Kompatsiaris, B. Huet, V. Mezaris, C. Gurrin, W.-H. Cheng, and S. Vrochidis, eds.), Lecture Notes in Computer Science, (Cham), pp. 254–265, Springer International Publishing, 2019.

[30] E. R. Soares and E. Barrére, "An optimization model for temporal video lecture segmentation using word2vec and acoustic features," in *Proceedings of the 25th Brazillian Symposium on Multimedia and the*

Web, pp. 513–520, Association for Computing Machinery, 2019.

[31] E. R. Soares and E. Barrére, "Automatic Topic Segmentation for Video Lectures Using Low and High-Level Audio Features," in *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web*, pp. 189–196, ACM Press, 2018.

[32] B. Atal and L. Rabiner, "A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, pp. 201–212, June 1976. Conference Name: IEEE Transactions on Acoustics, Speech, and Signal Processing.

[33] J. W. Seok and K. S. Bae, "Speech enhancement with reduction of noise components in the wavelet domain," in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1323–1326 vol.2, Apr. 1997. ISSN: 1520-6149.

[34] S. Tanyer and H. Ozer, "Voice activity detection in nonstationary noise," *IEEE Transactions on Speech and Audio Processing*, vol. 8, pp. 478–482, July 2000. Conference Name: IEEE Transactions on Speech and Audio Processing.

[35] S. Salishev, A. Barabanov, D. Kocharov, P. Skrelin, and M. Moiseev, "Voice Activity Detector (VAD) Based on Long-Term Mel Frequency Band Features," in *Text, Speech, and Dialogue* (P. Sojka, A. Horák, I. Kopeček, and K. Pala, eds.), Lecture Notes in Computer Science, (Cham), pp. 352–358, Springer International Publishing, 2016.

[36] L. J. Rodríguez-Fuentes, M. Peñagarikano, A. Varona, and G. Bordel, "GTTS-EHU Systems for the Albayzin 2018 Search on Speech Evaluation," in *IberSPEECH 2018*, pp. 249–253, ISCA, Nov. 2018.

[37] X. Che, S. Luo, H. Yang, and C. Meinel, "Sentence-Level Automatic Lecture Highlighting Based on Acoustic Analysis," in *2016 IEEE*

International Conference on Computer and Information Technology (CIT), pp. 328–334, Dec. 2016.

[38] G.-J. Poulisse, M.-F. Moens, T. Dekens, and K. Deschacht, "News story segmentation in multiple modalities," *Multimedia Tools and Applications*, vol. 48, no. 1, pp. 3–22, 2010.

[39] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. Senior, "Recent advances in the automatic recognition of audiovisual speech," *Proceedings of the IEEE*, vol. 91, pp. 1306–1326, Sept. 2003. Conference Name: Proceedings of the IEEE.

[40] J. Lai and N. Yankelovich, "Conversational speech interfaces," in *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*, pp. 698–713, USA: L. Erlbaum Associates Inc., Jan. 2002.

[41] A. Dhankar, "Study of deep learning and CMU sphinx in automatic speech recognition," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2296–2301, Sept. 2017.

[42] P. Gramming, J. Sundberg, S. Ternström, R. Leanderson, and W. H. Perkins, "Relationship between changes in voice pitch and loudness," *Journal of voice*, vol. 2, no. 2, pp. 118–126, 1988.

[43] J. G. Neuhoff, J. Wayand, and G. Kramer, "Pitch and loudness interact in auditory displays: Can the data get lost in the map?," *Journal of Experimental Psychology: Applied*, vol. 8, no. 1, pp. 17–25, 2002.

[44] P. Bajpai and M. Kumar, "Genetic algorithm–an approach to solve global optimization problems," *Indian Journal of computer science and engineering*, vol. 1, no. 3, pp. 199–206, 2010.

[45] K. A. De Jong and W. M. Spears, "A formal analysis of the role of multi-point crossover in genetic algorithms," *Annals of Mathematics and Artificial Intelligence*, vol. 5, pp. 1–26, Mar. 1992.

[46] F. Glover and M. Laguna, "Tabu Search," in *Handbook of Combinatorial Optimization: Volume1–3* (D.-Z. Du and P. M. Pardalos, eds.), pp. 2093–2229, Boston, MA: Springer US, 1998.

[47] S. Pfeiffer and I. Hickson, "Webvtt: The web video text tracks format," *Draft Community Group Specification, W3C*, 2013.

[48] M. Lin, J. F. Nunamaker, M. Chau, and H. Chen, "Segmentation of lecture videos based on text: a method combining multiple linguistic features," in *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*, pp. 9–pp, IEEE, 2004.

[49] N. Yamamoto, J. Ogata, and Y. Ariki, "Topic segmentation and retrieval system for lecture videos based on spontaneous speech recognition," in *Eighth European Conference on Speech Communication and Technology*, pp. 961–965, 2003.