**CONTENT BASED LECTURE VIDEO SEGMENTATION INTO TOPIC UNITS**

**Final Report Draft**

S. S. M. S. Fernando

( IT16001862)

BSc. (Honors) in Information Technology

Specializing in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

August 2019

# **DECLARATION**

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other university or institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology the nonexclusive right to reproduce and distribute our dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature : …..............................................................

S. S. M. S. Fernando

The above candidate has carried out research for the B.Sc Dissertation under my supervision.

Signature of the supervisor: …...................................... Date: ….........................

Signature of the co-supervisor: …................................. Date: ….........................

# **ABSTRACT**

Recorded lecture videos have gained widespread popularity as a method of delivering lecture content because of its distinct advantages. This approach gives leaners the ability to follow lectures without location and time constraints and consume lectures at their own pace. Although recorded lectures have many advantages, they tend to be lengthy and tedious to watch. They also prove cumbersome when specific information needs to be extracted from them. Therefore, segmenting lengthy lecture videos based on topic transitions is a key feature that could improve the overall learning experience of the student. Though there have been researches done in the field of segmenting text based on topics, only few have been done in the area of segmenting/ tagging videos according to topic transitions. Some of the existing platforms allow content creators to identify and tag videos based on topic transitions, however this need to be done manually and hence consumes lot of time and effort.

VTutor is a smart web platform that aims to address these drawbacks by introducing an automated process for identifying and combining the lecture material to create an enhanced user experience. Specifically, VTutor allows users to navigate through lecture videos using sub topic, its corresponding slides and code samples. Furthermore it is also equipped with ability to automatically generate questions based on lecture content thus improving learner engagement with the videos. In VTutor, we follow a linguistic approach for identifying topics within a video based on machine learning algorithms which analyses videos transcripts and identifies segmentation points (timestamps) where topic transitions occur.

# **ACKNOWLEDGEMENT**

The work described in this research paper was carried out as part of the final year research project for the subject Comprehensive Design Analysis Project. The completed final project is the result of combining all the hard work of the group members and the encouragement, support and guidance given by many others. Therefore, it is our duty to express our gratitude to all who supported us in this endeavor. We would like to thank Dr. Nuwan Kodagooda and Ms. Kushnara Suriyawansa our supervisor and co-supervisor, for the invaluable advice and guidance they provided. We are also extremely grateful to Mr. Jayantha Amararachchi, Senior Lecturer/ Head-SLIIT Centre for Research who gave and confirmed the permission to carry out this research. We also wish to thank all our colleagues and friends for their support. Finally, we would like to thank all others whose names are not listed particularly but have given their support in many ways and encouraged us to make this project a success.

**Table of Contents**

[**DECLARATION** ii](#_Toc16545835)

[**ABSTRACT** iii](#_Toc16545836)

[**ACKNOWLEDGEMENT** iv](#_Toc16545837)

[**List of Abbreviations** vi](#_Toc16545838)

[**List of Figures** vii](#_Toc16545839)

[**List of Tables** viii](#_Toc16545840)

[**1.** **Introduction** 1](#_Toc16545841)

[**1.1.** **Background Literature** 2](#_Toc16545842)

[**1.2.** **Research Gap** 3](#_Toc16545843)

[**1.3.** **Research Problem** 4](#_Toc16545844)

[**1.4.** **Research Objectives** 5](#_Toc16545845)

[**1.4.1.** **General Objective** 5](#_Toc16545846)

[**1.4.2.** **Specific Objectives** 5](#_Toc16545847)

[**2.** **Methodology** 6](#_Toc16545848)

[**2.1.** **Methodology** 6](#_Toc16545849)

[**2.1.1.** **Transcribing Videos** 6](#_Toc16545850)

[**2.1.2.** **Topic Modelling** 8](#_Toc16545851)

[**2.1.3.** **Text Segmentation** 10](#_Toc16545852)

[**2.2.** **Commercialization aspect of the product** 12](#_Toc16545853)

[**2.3.** **Testing and Implementation** 13](#_Toc16545854)

[**3.** **Results and Discussions** 15](#_Toc16545855)

[**3.1.** **Results** 15](#_Toc16545856)

[**3.2.** **Discussion** 18](#_Toc16545857)

[**CONCLUSION** 19](#_Toc16545858)

[**References** 21](#_Toc16545859)

[**APPENDICES** 22](#_Toc16545860)

# **List of Abbreviations**

|  |  |
| --- | --- |
| TDT | Topic Detection and Tracking |
| LDA | Latent Dirichlet Allocation (topic modelling algorithm) |
| LMS | Learning Management System |
| API | Application Programming Interface |
| Topic Tiling | An algorithm for text segmentation |

# 

# **List of Figures**

[Figure 2.1 Sample time stamped transcript from Assembly AI 7](https://mysliit-my.sharepoint.com/personal/it16001862_my_sliit_lk/Documents/Final%20Report%20Individual.docx#_Toc19559370)

[Figure 2.2 Plate notation for LDA with Dirichlet-distributed topic-word distributions 9](#_Toc19559371)

[Figure 2.3 Sample output from topic tiling algorithm 11](https://mysliit-my.sharepoint.com/personal/it16001862_my_sliit_lk/Documents/Final%20Report%20Individual.docx#_Toc19559372)

[Figure 2.4 Overall process flow of content based video segmentation into topics 12](#_Toc19559373)

[Figure 2.5 Business model canvas 13](#_Toc19559374)

[Figure 2.6 VTutor video review UI 14](#_Toc19559375)

[Figure 2.7 VTutor homepage 14](#_Toc19559376)

[Figure 2.8 VTutor video playback page 15](#_Toc19559377)

[Figure 4.1 High level architecture of VTutor Platform 22](https://mysliit-my.sharepoint.com/personal/it16001862_my_sliit_lk/Documents/Final%20Report%20Individual.docx#_Toc19559378)

# **List of Tables**

[Table 3.1 Transcription results from Assembly AI 15](#_Toc19559445)

[Table 3.2 List of lowest Pk values for the Choi data set for different algorithms. 16](#_Toc19559446)

[Table 3.3 Sample 1 text segmentation results 17](#_Toc19559447)

[Table 3.4 Sample 2 text segmentation results 17](#_Toc19559448)

[Table 3.5 Sample 3 text segmentation results 18](#_Toc19559449)

# **Introduction**

Today, e-learning has become an essential component of higher education for both teachers and students. According to a study on the effectiveness of e-learning on education, it was found that students nowadays are more satisfied with web enhanced learning when compared to a traditional classroom environment [1].  Therefore, it is commonplace to see universities and higher education institutes adopting some form of e-learning to assist their students and many institutes use their own customized version of a Learning Management System (LMS) to provide online course material.

Online education is beneficial in many ways. For instance, as lecture content is always available online, the possibility of missing a lecture is low and teachers can ensure that students have access to course material irrespective of time and location [1]. Recorded lecture videos also enable students with different styles of learning and different levels of understanding to obtain a better grasp of the subject. For example, those who are familiar with the work can skip ahead to a section of interest while those that need more time to understand the concepts can pause and rewind to digest the lecture at their own pace [2, 3].

Many LMSs enable lecturers to upload course material such as tutorials, lab sheets, lecture slides and recorded lecture videos. Whilst videos are effective because they address both the visual and auditory aspects of teaching [4], many students find it tedious to watch recorded lecture videos because of its duration, which normally lasts around 1 - 3 hours and its lack of interconnectivity and relevance to other course material [5].

Although platforms such as LearnWorlds[[1]](#footnote-2), Echo360[[2]](#footnote-3) and Techsmith Relay[[3]](#footnote-4) allow users to create interactive course material, the methods used by them are time consuming as they require the lecturer to spend time editing and making the video interactive. Other applications like Camstatia[[4]](#footnote-5) allow users to create interactive learning material only if the videos are recorded using their own software, which is quite impractical in a normal lecture hall.

V-Tutor is a smart web platform which is capable of automatically transforming raw lecture material into content which is more interactive while saving the time and effort of the content creators. It provides many interactive ways for leaners to navigate through lecture videos such as using topics, slides and specific lines in code samples. In addition, it is also capable of generating content-based questions from the videos which can be utilized by the learners to check their knowledge on the key points of a specific lecture video.

## **Background Literature**

Segmentation of long lecture videos into cohesive topic units is highly beneficial since it makes the search for information easier, minimizes learning time and improves the overall learning experience. Research has been carried out in relation to segmenting videos based mainly on three areas - visual content, audio and text (transcripts). However, limited research has been conducted in relation to the specific domain of segmenting lecture videos into topic units.

The most commonly used mechanism for segmenting videos is based on scene changes. A research conducted in 2000 used color histogram distance computation between successive images to detect scene changes [6].  Another research done by Zhang and Smoliar, proposed a system for detecting progressive transitions based on both motion and statistical analysis [7]. However, this mechanism of segmentation based on scene changes is not applicable to lecture videos, as they have very few scene changes and even these scene changes do not match with topic transitions [4].

Another feature considered by researchers for lecture video segmentation is slide matching. A research done in 2013, focused on segmenting lecture videos into topics by analyzing its supplementary synchronized slides. They used OCR to extract content from lecture slides and identify different subtopics according to their logical relevance. Slides were then synchronized with the video stream to identify different topic changes. As stated in their paper [8] the mechanism was approximately 90% accurate, except that it always assumes that the slides are synchronized with the video streams, which, in practice might not always be the case. Furthermore, since their method was solely based on matching slide content with the video, its accuracy is limited only for certain lecture video types.

Moreover, segmentation methods utilizing transcribed text or closed captions have also been researched on. The main motivational factor for work in this area was the Topic Detection and Tracking (TDT) initiative conducted in 1998 [6]. TDT is defined as the task of segmenting transcribed speech into topically cohesive stories. Their algorithm is trained mainly using broadcast and news domain data sets where, formal presentation format and cue phrases are used to improve segmentation accuracy. Unlike in the broadcast domain, speeches in lecture videos are often unscripted and spontaneous. Furthermore, a large training dataset is used for many methods in TDT, which is not available for lecture videos [4] [6] [9].

Inspired by the work on TDT, in 2004 Michael Chau, and his team members, conducted another research to identify topic changes based on multiple linguistic features like noun phrases, topic noun phrases, verb classes, word stems, combined features, cue phrases, and pronouns [4]. Using automatic speech recognition software, they retrieved timestamps that synchronize with the video stream and then the results from transcribed text segmentation is then mapped back to video segmentation. However, they only developed a set of algorithms each considering one specific feature out of the list of segmentation features mentioned above. They then compared the results from these algorithms to identify the most salient feature for lecture video segmentation. Work in this area can be considered as a potentially successful solution and can be improved further as audio and the transcribed text extracted from lecture videos provide rich content information for topic change detection [4].

## **Research Gap**

Even though there are several products already available with similar objectives, they mostly focus on the use of manual processes that involve human intervention to make the content more interactive and increase the searchability. Our proposed solution aims to reduce the amount of human interaction needed for this process by introducing a platform which will analyze and augment the content automatically.

When considering about content based lecture video segmentation into topic units, many of the existing platforms allow users to define split points and segment video and then name the segments according to topics manually. This requires the user to traverse through the whole video and identify the segmentation points which a very time consuming process which requires lot of effort. Also there are other video editing platforms which is capable of automatically segment video based on scene changes. However this is not applicable for lecture videos, because unlike in many other types of videos, recorded lectures contains few scene changes and these do not directly correspond to topic transitions. The goal of our system is to develop an automated process which analyses video transcripts and automatically identifies segmentation points within a lecture video based on topic transitions.

## **Research Problem**

Nowadays, video lectures have become increasingly popular and many education institutes use Learning Management Systems that support video content. Whilst video lectures have benefits such as giving learners remote access to lectures, there are a few drawbacks such as poor searchability through the video and less interaction with the learner.

To overcome these drawbacks many lecture platforms have introduced tools such as web-based video editors that allow lecturers to add captions, divide the video into discussed topics, link lecture slides and embed questions into the video. However, these tools require human intervention which is time-consuming. When considering the domain of computer science, none of the available platforms provide a tool to map programming language code segments with their occurrences in the video.

Because there is a need for video lectures to be more interactive and searchable, but that a significant amount of time is taken by lecture creators to make them so, we conclude that a platform which will analyze and augment the content automatically will be useful.

## **Research Objectives**

### **General Objective**

The proposed system is a research study to improve the method of delivery for video lectures and increase the engagement of the learner. The main objective of the research is to develop an automated platform which provides a quick and efficient way for lecture creators to deliver video lectures which are more interactive and have increased searchability. Specific objectives of this research are as follows.

### **Specific Objectives**

* Develop a mechanism for transcribing uploaded lecture videos and store in cloud storage
* Develop and train a topic model which can effectively identify topics from video transcripts.
* Automatically identify main topic transitions of a given lecture video based on the topic model and segment the video according to the identified positions.
* Design an interactive user interface for students to navigate through a given lecture using sub topics.

# **Methodology**

## **Methodology**

As elaborated in the previous section of this paper, people have tried different mechanisms for identifying topic transitions within lecture videos. In this research we focus on a linguistic approach based on video transcripts. Video transcripts provide rich content information and will be ideal for detecting topic transitions. Also, generally computing audio and visual features is a very time-consuming process, while computing text takes comparatively less amount of time. Our approach consists of two main parts: Topic modelling and text segmentation based on the topic model.

### **Transcribing Videos**

A key factor which affects the overall accuracy of topic modelling is how accurately the videos are transcribed. For transcribing videos, a third party API called AssebyAI[[5]](#footnote-6) is used. With AssembyAI a time stamped transcript can be extracted for a given video URL, which contains the time (its occurrence in the video) in milliseconds for each and every word in the video. A sample time stamped transcript generated by AssemblyAI API is shown in Figure 2.2. This time stamped transcript is later used to identify the segmentations point in the video, in other words times where the video need to be segmented.

The transcript of the video need to be extracted automatically whenever a new video is uploaded to cloud storage through the web app. To trigger this flow following cloud architecture is used when deploying the web app

Figure . Sample time stamped transcript from Assembly AI

**{ "status": "completed",**

**"acoustic\_model": "assemblyai\_default",**

**"text": "Look at collection classes specifically, we look at an introduction to a list and also what action classes are really. So basically you have a framework, you have a framework of classes in Java which allows you to do pretty sophisticated stuff.",**

**"format\_text": true,**

**"punctuate": true,**

**"audio\_url": "https://cdap-bucket.s3.ap-south-1.amazonaws.com/2019-OOP-Revision-ArrayList-1-MP4.mp4",**

**"words": [**

**{**

**"text": "Look",**

**"confidence": 0.73,**

**"end": 140,**

**"start": 0**

**},**

**{**

**"text": "at",**

**"confidence": 0.86,**

**"end": 240,**

**"start": 100**

**},**

**{**

**"text": "collection",**

**"confidence": 0.87,**

**"end": 640,**

**"start": 220**

**},**

**],**

**"language\_model": "computer-science-model-2",**

**"id": "898o9jwx-01a9-4fc2-b7c9-c5595540adf3",**

**"confidence": 0.894341138659315,**

**"utterances": null,**

**"audio\_duration": 2341.10548752834,**

**"webhook\_status\_code": 200,**

**"webhook\_url": "http://52.66.30.76:5000/vtutor-transcriptions-api/v1/get-transcript",**

**"dual\_channel": null**

**}**

.

### **Topic Modelling**

Our approach for topic modelling is based on a state of art algorithm known as Latent Dirichlet allocation (LDA). LDA is a form of unsupervised learning and is one of the most popular generative, probabilistic text modelling techniques in machine learning. LDA works with the assumption that each document was generated by picking a set of topics and then for each topic picking a set of words [10]. Input to LDA is a collection of documents, which in our case is the video transcripts.

LDA considers each document is in the form of a ‘bag of words’ [10]. Therefore, in order to transform raw transcripts into a bag of words structure, it need to undergo data pre-processing. Standard data pre-processing for LDA consists of the following steps:

* *Tokenization* - Split text into words, lowercase the words and remove punctuation
* *Remove stopwords* - stopwords are most commonly used words in a language, for an example word ‘the’ in English language. Stopwords are usually not relevant to any topics hence, removing these from the corpus supports LDA to extract topics more accurately.
* *Lemmatization* - third person words are changed to first person, and verbs in different tenses are converted to present tense.
* *Stemming* - words are reduced to their root form.

Once preprocessing is done the corpus is subjected to LDA. The generative process of LDA is as follows:

For each document w in a corpus D:

1. Choose N Poisson (ξ).

2. Choose Θ Dir (α).

3. For each of the N words wn:

a. Choose a topic zn Multinomial (Θ).

Choose a word wn from 𝑝(𝑤𝑛 |𝑧𝑛, 𝛽), a multinomial probability conditioned on the topic zn. [11]



Figure 2.2 Plate notation for LDA with Dirichlet-distributed topic-word distributions

Source: <https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation>

α - per-document topic distributions,

β - per-topic word distribution,

θ - topic distribution for document m,

φ - word distribution for topic k,

z - topic for the n-th word in document m

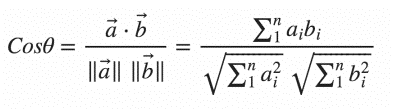
w - specific word [12]

LDA was originally designed for extracting topics from written text, such as newspaper articles and etc. However video transcripts contain spoken words which is less saturated with words related to the topics. Therefore when training the model hyper parameters α and β need to be chosen carefully. In this research LDA model was trained using β = 0.05 and default α value which is 50/K (K is the number of topics).

### **Text Segmentation**

Text segmentation is achieved through a LDA based segmentation algorithm known as topictiling. Topic Tiling[[6]](#footnote-7) was first introduced by Riedl and Biemann in 2012 and was inspired by the text tiling algorithm developed by Mari Hearst. Unlike in text tiling which segments text based on words, topic tiling algorithm is based on topic IDs assigned by Bayesian Inference method of LDA. When compared with other segmentation methods based on LDA , topic toiling is computationally less expensive as it performs segmentation in linear time. [13]

In topic tiling , initially the document is split into units based on sentence boundaries. Each sentence is represented by a N dimensional vector where N is the number of topics represented in the topic model. Then the cosine similarity between two adjacent sentences are computed using Equation 1 to calculate the coherence [14].



Equation Cosine similarity

Where **a**i and **b**i are the vectors of two adjacent sentences.

Values close to one indicates a substantial connectivity of the two sentences whereas, values close to 0 indicate a maginal relatedness. The coherence are then plotted to trace the local minima, which is utilized in identifying the possible segmentation boundaries. [15]

The final outcome of the topic tiling algorithm will be a XML file with text segments as shown in Figure 2.3.

Figure 2.3 Sample output from topic tiling algorithm

**<documents>**

**<document>**

**<documentName>input.txt</documentName>**

**<segments>**

**<segment>**

**<depthScore>0.0</depthScore>**

**<text>………………..</text>**

**</segment>**

**<segment>**

**<depthScore>0.012</depthScore>**

**<text>………………..</text>**

**</segment>**

**</segemnts>**

**</document>**

**</documents>**

This XML file is then read by another algorithm and matched with a time stamped transcript of the video to extract the timestamps where the segmentations occur. Final outcome of the topic modelling component will be a set timestamps along with set of key words suggested from the topic model.

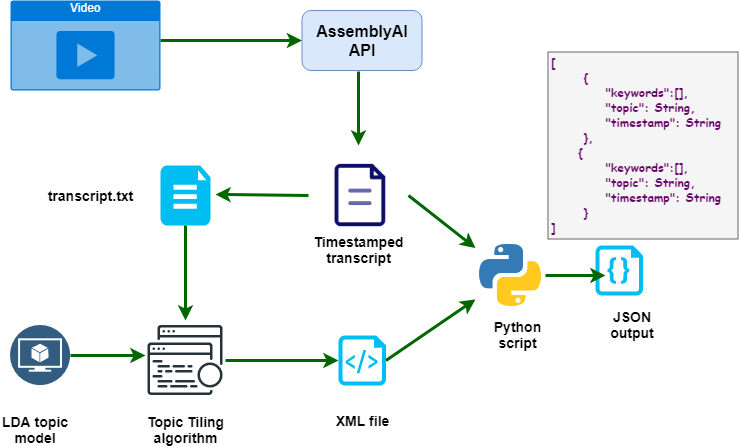


Figure 2.4 Overall process flow of content based video segmentation into topics

## **Commercialization aspect of the product**

As shown in the literature survey and research gap, none of the available commercial products provide automated unique features introduced in our platform. Considering the unique features provided in a web-based nature, our platform has a unique business potential in eLearning field. Few of the business product approaches that is available for the platform is mentioned below.

* Standalone platform for video lectures provided as SaaS product, where customers can get license and use the features.
* Free platform for video lectures with payable content that viewers must pay purchase or subscribe to view premium content.
* Integration to existing MOOCs and LMSs as a plugin.

Figure 2.5 shows the proposed business model for our system

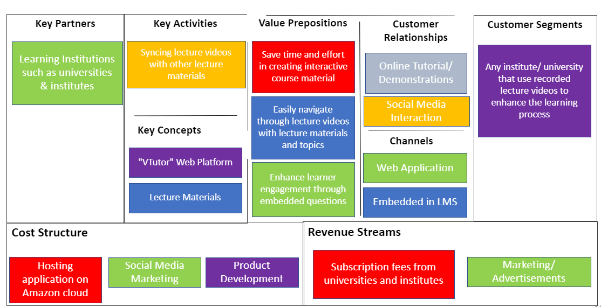


Figure 2.5 Business model canvas

## **Testing and Implementation**

The overall topic modelling and video segmentation process can only suggest a set key words for each topic and the timestamps where possible topic transitions might occur. It cannot predict the exact topic for a given segment, because topic is something vague and may differ according to an individual’s perspective. Therefore once a lecture video is processed, it should be reviewed by the lecturer before publishing it for students. Vtutor web app consists of a video review UI Figure 2.6 where the lecturers can review and edit the topics and segmentation points suggested by the system, before publishing it.

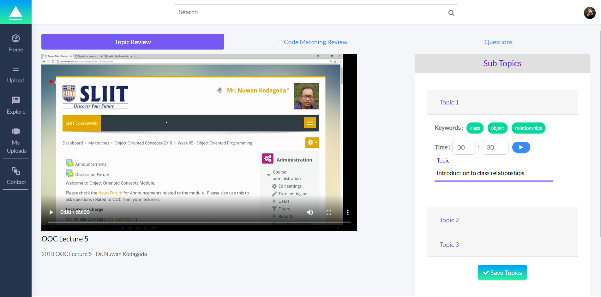


Figure 2.6 VTutor video review UI

VTutor homepage and video playback page is shown in the Figure 2.7 and Figure 2.8 .

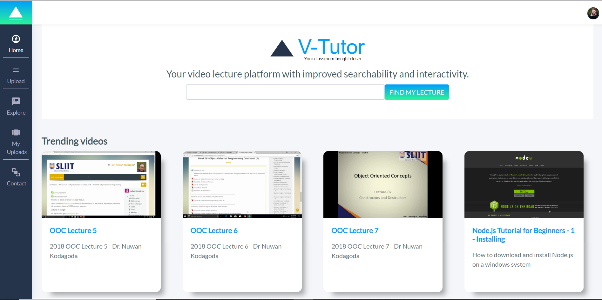


Figure 2.7 VTutor homepage

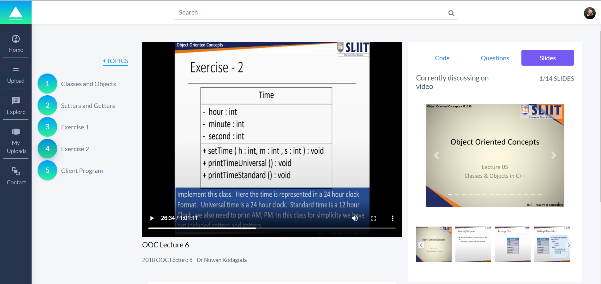


Figure 2.8 VTutor video playback page

# **Results and Discussions**

## **Results**

This section presents the results and evaluations of the component content based video segmentation into topic units of VTutor platform.

Transcribing lecture video as accurately as possible is a key feature that will affect the accuracy of the topic model as well as text segmentation process. As discussed in the earlier section of this document lecture videos are transcribed using AsemblyAI. Table 3.1 shows the average confidence level of transcribing raw mp4 lecture videos, without any editing or noise removal.

|  |  |  |  |
| --- | --- | --- | --- |
| **Video Name** | **Video Duration (hh:mm:ss)** | **Approx. Duration for transcribing (minutes)** | **Average confidence of the transcript** |
| OOC Lecture 5 | 00:59:09 | 17 | 0.8543 |
| OOC Lecture 6 | 01:01:11 | 23 | 0.8231 |
| OOC Lecture 7 | 00:59:27 | 19 | 0.8711 |
| OOC Lecture 9 | 00:57:31 | 14 | 0.8972 |

Table 3.1 Transcription results from Assembly AI

Apart from the topic tiling algorithm used in this research for text segmentation, there are various other algorithms capable of segmenting text into subtopics. Pk value is a metric that is popularly used for evaluating text segmentation algorithms. It based on the window size and the segmentation boundaries of the algorithm. Lower the Pk value, better the performance of the algorithm. Table 3.2 shows a list of Pk values for Choi data set for different text segmentation algorithms in literature when number of segments are provided and unprovided.

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Segments** | |
| **Provided** | **Un provided** |
| TextTiling (TT) | 44.48 | 49.51 |
| C99 | 11.20 | 12.73 |
| U00 (Utiyama and Isahara, 2001) | 9 | 10 |
| F04 (Fragkou et al., 2004) | 5.39 |  |
| M09 (Misra et al., 2009) | 2.73 |  |
| C99LDA | 2.69 | 3.24 |
| TTLDA | **1.04** | 2.89 |
| Topic Tiling | 1.06 | **1.39** |

Table 3.2 List of lowest Pk values for the Choi data set for different algorithms.

As seen in the table, topic tiling algorithm outperforms all other algorithms when the number of segments are not given. Hence it is the best algorithm for segmenting text based on topic when compared with other existing algorithms.

Table 3.3, Table 3.4 and Table 3.5 shows the results of topic tiling algorithm for three sample videos compared to manual segmentation of the same video in a human perspective in two scenarios: number of segments provided and number of segments not provided.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Video Title - OOC Lecture 5** | | | | |
| **Number of segments not provided** | | | | |
|  | **Topic Tiling Algorithm** | | **Manual Segmentation** | |
| **No. of segments** | 5 | | 3 | |
| **Number of segments provided** | | | | |
|  | **Timestamps (hh:mm:ss)** | | | |
| **Segmentation points identified by the algorithm** | **Segmentation points identified by manual segmentation** | | **Time difference in minutes** |
| **Segment 1** | 00:04:45 | 00:02:33 | | 2.2 |
| **Segment 2** | 00:17: 54 | 00:19:31 | | 1.62 |
| **Segment 3** | 00:42:11 | 00:39:28 | | 2.71 |

Table 3.3 Sample 1 text segmentation results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Video Title - OOC Lecture 6** | | | | |
| **Number of segments not provided** | | | | |
|  | **Topic Tiling Algorithm** | | **Manual Segmentation** | |
| **No. of segments** | 8 | | 5 | |
| **Number of segments provided** | | | | |
|  | **Timestamps (hh:mm:ss)** | | | |
| **Segmentation points identified by the algorithm** | **Segmentation points identified by manual segmentation** | | **Time difference in minutes** |
| **Segment 1** | 00:02:10 | 00:03:37 | | 1.45 |
| **Segment 2** | 00:10:40 | 00:08:32 | | 2.14 |
| **Segment 3** | 00:16:12 | 00:13:43 | | 2.48 |
| **Segment 4** | 00:25:56 | 00:26:34 | | 0.64 |
| **Segment 5** | 00:53:39 | 00:50:22 | | 3.28 |

Table 3.4 Sample 2 text segmentation results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Video Title - OOC Lecture 6** | | | | |
| **Number of segments not provided** | | | | |
|  | **Topic Tiling Algorithm** | | **Manual Segmentation** | |
| **No. of segments** | 7 | | 5 | |
| **Number of segments provided** | | | | |
|  | **Timestamps (hh:mm:ss)** | | | |
| **Segmentation points identified by manual segmentation** | **Segmentation points identified by manual segmentation** | | **Time difference in minutes** |
| **Segment 1** | 00:03:42 | 00:02:37 | | 1.03 |
| **Segment 2** | 00:15:34 | 00:18:30 | | 2.93 |
| **Segment 3** | 00:27:12 | 00:25:58 | | 1.23 |
| **Segment 4** | 00:37:50 | 00:36:28 | | 1.30 |
| **Segment 5** | 00:46:10 | 00:42:59 | | 3.19 |

Table 3.5 Sample 3 text segmentation results

On average there is time difference of 2.02 minutes between the timestamps suggested by the algorithm and when the same process is done manually by a human.

## **Discussion**

According to the figures in **Error! Reference source not found.** , transcribing a video using AssemblyAI takes around 20-30% of the duration of the video, which is tolerable delay because all the processing of the videos are dome in the backend and user do not have to wait until the videos is processed. Once a user uploads lecture materials through the VTutor web app it is automatically subjected for processing including transcription extraction and topic modelling. On successful completion of processing the user (lecturer) will notified through the web app and the users can then review the processed video, edit and publish for students to view.

Furthermore most of the transcripts obtained using AssemblyAI have a confidence level of approximately between 0.8 and 0.9, and the major factor which affects the drop in confidence level is the accent and the background noises. All the videos subjected for testing is done by Sri Lankan lecturers with an Asian accent and most of these lectures includes background noise such as student voices which is commonly seen in a typical lecture hall. When considering all these factors, it can be concluded that AssemblyAI is a good for our platform because it transcribes lectures with a fairly high level of confidence within a reasonable period of time.

A main contributing factor for text segmentation is how well the LDA model is trained. Since LDA is an unsupervised learning technique, lot of data is required to train the model effectively. Many of the available data sets on internet used for training LDA are not related to lecture video transcripts, instead many of these are based on newspaper articles and other written text. None, these datasets are suitable for training the LDA model in this context because unlike written text, transcripts contain spoken words which are spontaneous and less saturated with topic related words. Hence, one of the main challenges in this research is to train the LDA model effectively, due to lack of data. The results can be further improved by training the model with a larger data set.

# **CONCLUSION**

Nowadays many students prefer online learning rather than traditional classroom based learning. In order to cater this growing desire of the modern generation, many universities and higher educational institutes adapt some kind of a Learning Management System to deliver online course materials for their students. One of the commonly used techniques for online delivery of lectures is the use of recorded lecture videos.

However students face many problems when using recorded lecture due to its lack of interactivity. One such problem is the difficulty to find specific information within the video. There are many platforms which allows lecturer to create well organised interactive videos , but most of these require either manual editing which consumes lot of time or lectures need to be recorded using specific software.

If the videos van be segmented into topic units automatically it would be very beneficial for both the student and the lecturers. This paper elaborates a feasible way of automatically segmenting videos into topics based on video transcripts. Currently the system is tested only for computer domain specific lectures, however it can be extend for other domains as well. Aslo, the accuracy and performance of the proposed method can be further increased by training the machine learning model with more data.

# **References**

|  |  |
| --- | --- |
| [1] | “Importance and Effectiveness of E-learning,” 1 December 2015. [Online]. Available: https://higheredrevolution.com/importance-and-effectiveness-of-e-learning-9513046ed46c. [Accessed 6 March 2019]. |
| [2] | M. Brandsteidl, T. Mayerhofer, M. Seidl and C. Huemer, “Replacing traditional classroom lectures with lecture videos: an experience report,” in *Proceedings of the 8th edition of the Educators' Symposium*, 2012. |
| [3] | M. Maher, H. Lipford and V. Singh, “Flipped classroom strategies using online videos,” UNC Charlotte, North Carolina, 2013. |
| [4] | M. Chau, M. Lin, J. F. Nunamaker and H. Chen, “Segmentation of Lecture Videos Based on Text: A Method Combining Multiple Linguistic Features,” in *Proceedings of the 37th Hawaii International Conference on System Sciences*, Hawaii, 2004. |
| [5] | Z. Woolfitt, “The effective use of video in higher education,” 2015. [Online]. Available: https://www.inholland.nl/media/10230/the-effective-use-of-video-in-higher-education-woolfitt-october-2015.pdf. [Accessed 6 March 2019]. |
| [6] | M. Lin, C. B. R. Diller, N. Forsgren, Y. Huang and J. J. F. Nunamaker, “Segmenting Lecture Videos by Topic: From Manual to Automated Methods,” in *Eleventh Americas Conference on Information Systems*, Omaha, 2005. |
| [7] | H. J. Zhang and S. W. Smoilar, “Developing power tools for video indexing and retrieval,” in *SPIE'94 Storage and Retrieval for Video Databases*, San Jose, CA, USA, 1994. |
| [8] | X. Che, H. Yang and C. Meine, “Lecture Video Segmentation by Automatically Analyzing the Synchronized Slides,” in *MM’13*, Barcelona, Spain, 2013. |
| [9] | J. Allan, J. Carbonel, G. Doddington, J. Yamron and Y. Yang, “Topic detection and tracking pilot study: Final report,” in *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*. |
| [10] | T. Doll, “LDA Topic Modeling: An Explanation,” 25 June 2018. [Online]. Available: https://towardsdatascience.com/lda-topic-modeling-an-explanation-e184c90aadcd. [Accessed 4 August 2019]. |
| [11] | A. Onan, S. Korukoglu and H. Bulut, “LDA-based Topic Modelling in Text Sentiment Classification: An Empirical Analysis,” *International Journal of Computational Linguistics and Applications,* vol. VII, pp. 101-119, 2016. |
| [12] | “Latent Dirichlet allocation,” wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Latent\_Dirichlet\_allocation. [Accessed 4 August 2019]. |
| [13] | “Topitiling,” Github, [Online]. Available: https://github.com/riedlma/topictiling. [Accessed 4 August 2019]. |
| [14] | M. Riedl and C. Biemann, “How Text Segmentation Algorithms Gain from Topic Models,” in *2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Montreal, Canada, 2012. |
| [15] | S. Kaur and K. K. Mangat, “COMPARATIVE ANALYSIS OF C99 AND TOPICTILING TEXT,” *International Journal of Research in Engineering and Technology,* vol. II, pp. 529-534, 2013. |

# **APPENDICES**

A screenshot of a cell phone

Description automatically generated

Figure . High level architecture of VTutor Platform

1. https://www.learnworlds.com/ [↑](#footnote-ref-2)
2. https://echo360.com/ [↑](#footnote-ref-3)
3. https://www.techsmith.com/lecture-capture.html [↑](#footnote-ref-4)
4. https://www.techsmith.com/video-editor.html [↑](#footnote-ref-5)
5. <https://www.assemblyai.com/> [↑](#footnote-ref-6)
6. <https://github.com/riedlma/topictiling> [↑](#footnote-ref-7)