An Adaptive Time Reduction Technique

for

Video Lectures

by

Sreenivas Purushothama Shenoy

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2016 by the
Graduate Supervisory Committee:

Ashish Amresh, Chair
John Femiani
Erin Walker

ARIZONA STATE UNIVERSITY

May 2016

ABSTRACT

Lecture videos are a widely used resource for learning. A simple way to create videos is to record live lectures, but these videos end up being lengthy, include long pauses and repetitive words making the viewing experience time consuming. While pauses are useful in live learning environments where students take notes, I question the value of pauses in video lectures. Techniques and algorithms that can shorten such videos can have a huge impact in saving students' time and reducing storage space. I study this problem of shortening videos by removing long pauses and adaptively modifying the playback rate by emphasizing the most important sections of the video and its effect on the student community. The playback rate is designed in such a way to play uneventful sections faster and significant sections slower. Important and unimportant sections of a video are identified using textual analysis. I use an existing speech-to-text algorithm to extract the transcript and apply latent semantic analysis and standard information retrieval techniques to identify the relevant segments of the video. I compute relevance scores of different segments and propose a variable playback rate for each of these segments. The aim is to reduce the amount of time students spend on passive learning while watching videos without harming their ability to follow the lecture. I validate the approach by conducting a user study among computer science students and measuring their engagement. The results indicate no significant difference in their engagement when this method is compared to the original unedited video.

DEDICATION

To God, without whom I wouldn’t have reached anywhere. To my mother and father who always want to see me succeeding.

ACKNOWLEDGMENTS

I would like to thank my thesis committee chair Dr. Ashish Amresh for his guidance in the entire course of this project. Additionally I would like to thank my committee members Dr. John Femiani and Dr. Erin Walker for taking their time to provide me with valuable feedback. I am thankful to Mr.Ruben Acuna for providing data which was very helpful in the project. I would also like to thank my brotherly figure, Dr.Rajeev V Shenoi for helping me achieve this dream. I am very grateful to my sister and brother-in-law for their immense help. My acknowledgment would be incomplete if I do not mention and thank my best friend Ramya Sivaraman for supporting me through out this research work. I am very grateful for the unconditional love and support my family, roommates and friends have extended and to all the users who participated in the user survey.

TABLE OF CONTENTS

CHAPTER Page

APPENDIX

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

1.1 Motivation

With the popularity of personal video cameras, video processing software and availability of broadband connections, we are entering an era where people depend widely on classroom lecture videos or professionally shot Khan-style videos (Guo, Kim, and Rubin 2014) for educational purposes. A Khan-style video is a full-screen video of an instructor drawing freehand on a digital tablet, a style popularized by Khan Academy (*Khan Academy*). A screen shot for Khan-style video is shown in figure 1. A classroom lecture, as the name suggests, is a video shot in a classroom setup. A screen shot for a classroom lecture is shown in figure 2. Although many research studies were carried out to understand how to improve the effectiveness of these videos, there still exists a need for techniques with which we can shorten videos and save time while maintaining the audio and video clarity and keeping the ability to understand the lecture at par.

A lecture video shot in an academic environment comprises segments such as long pauses, noises such as speaker clearing his throat and murmurs, which take up time and bandwidth. We posit these as undesirable segments which can be removed. A Khan-style video, even though having a better production quality, can have uneven pace due to the way the speech is recorded as well as the presence of pauses. Additionally, many videos contain segments, which do not contribute much to learning. For instance, in a Khan-style video present in (*A sample Khan style video*), the video segment from

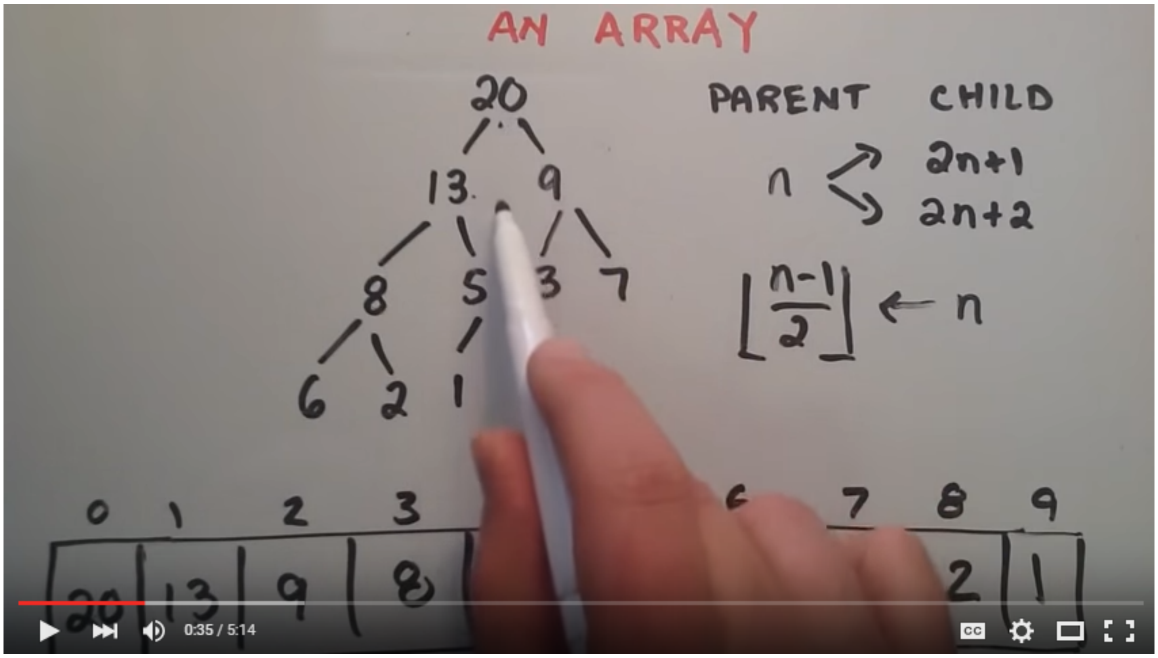


Figure 1: Khan-Style Video

5:07 to 5:14 can be transcribed to “Anyways thank you guys for watching. Have an excellent day and if you have not already, do not forget to subscribe.“ This video segment can be safely sped up to a higher rate and thereby saving 7 seconds of a student watching the lecture.

In this research, we aim to discuss this challenge and present a novel technique of temporal compression of lecture videos by giving emphasis on important sections that are identified using silence detection and speech-to-text analysis. We use an existing speech-to-text algorithm (*IBM Watson Developer Cloud - Speech to Text*) to extract the speech and use the findings of (Gong and Liu 2001) to find the relevance of the extracted text segments. The associated video segments are then mapped to a playback rate range. We analyze how users respond to such a video, in which all the pauses are removed and sped up on a step function through a user study. We validate our findings through a statistical test and present our results.

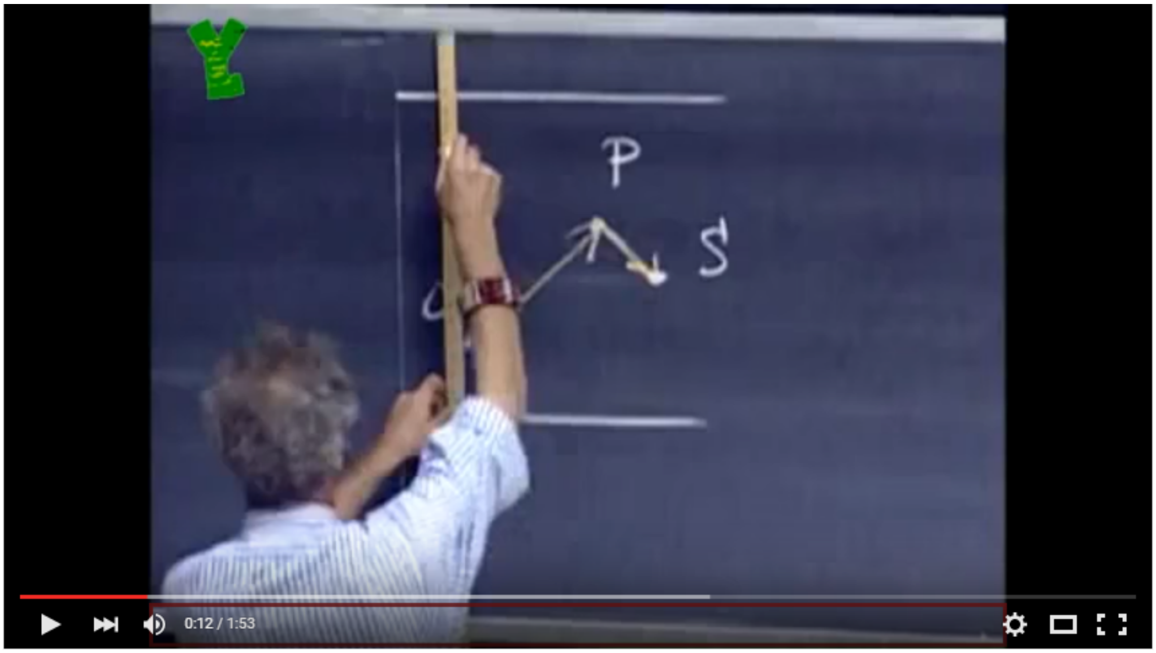


Figure 2: Classroom Lecture Video

To summarize, the novel contributions being made in this research are as follows:

1. We introduce the idea of a varying playback rate for a recorded lecture video.
2. We demonstrate that the speech between two long pauses can be classified and tested for its relevancy.
3. We demonstrate an effective application of standard information retrieval (IR) (Gong and Liu 2001) and latent semantic analysis (Gong and Liu 2001) techniques on lecture videos in modifying the playback rate.
4. Finally we prove that effectiveness of the method is not statistically different from the original video

Chapter 2

RELATED WORK

The closest related work (Galbraith and Spencer 2001) is by Galbriath, Joel D., and Steven G. Spencer, who studied the deployment of variable-speed playback capability in a media player application. In (Galbraith and Spencer 2001) it is suggested that, "Most users preferred an accelerated speed of at least 1.5 times normal, and several users reported watching the lecture material comfortably at speeds of twice normal and higher." The study is different from ours as our study introduces the playback rate not as a constant but as a function of relevance of the content. Applications such as Camtasia (*Camtasia*) and Audacity (*Audacity*) perform well in removing silence but require manual efforts. The web application Laconia Trim (*Laconia Trim*) trims audio and removes pauses automatically but it is still in its early phases of development. Another closely related work is by Guo, Philip J and Kim, Juho and Rubin, Rob (Guo, Kim, and Rubin 2014). The work provides many findings such as "shorter videos are much more engaging" and recommends that lecture videos should be segmented to less than 6 minutes. It also suggests that faster-speaking videos are more appealing. Another seminal work on compressing lecture video was by, He, L., Sanocki, E., Gupta, A., and Grudin, J (He et al. 1999) which uses three techniques namely pitch and pause information of audio signal, knowledge of slide transition points and information about access patterns of previous users. The speech skimmer developed by B Arons (Arons 1997) temporally compresses a video by introducing an idea of pause removal and interactively skimming recorded speech. But to our knowledge, our study is the first to introduce the idea of automatically varying the playback rate in lecture videos and studying its effect.

Chapter 3

METHODOLOGY

We took an experimental approach: We first analyzed how students respond to a fluctuating playback rate through a survey(User study-1). Based on the results obtained from User study-1, we designed a robust approach using standard IR and latent semantic analysis techniques discussed in (Gong and Liu 2001) to compute the relevance scores of different segments present in a video. The playback rate for each of these segments is then modified based on the relevance score to create a temporally compressed video. We performed statistical validation using another survey (User study-2) to study the effectiveness of such a video.

3.1 Dead-time Removal

Dead-time with respect a lecture video refers to segments in the video in which no important events are happening. It can be identified as segments with long pauses or vague speech such as lecturer clearing his throat or murmuring. The observations made in (Guo, Kim, and Rubin 2014) regarding removal of pauses and filler sounds by edX (*edX*) supports our hypothesis - removal of dead-time in lecture videos is not detrimental to user engagement.

An ideal solution does not have to remove all dead-time but it must not remove the critical parts of the instructional video. An ideal solution should also be of an ideal speed for the students. If there are two equally prepared students, student A watching the original uncompressed video, and student B watching the video processed using our technique, then student B should recall as much as student A. The output should not produce distracting or unpleasant artifacts such as changes in pitch, obvious discontinuities or other artifacts which make the learning experience less pleasant for student B when compared to student A.

Our application was developed in Python because of the availability of various open source libraries which we could customize for the situation. FFmpeg (*FFmpeg*), an open-source tool which provides excellent results in removing silence is the heart of our application. We used a trial-and-error technique for finding out the maximum duration of silence, which could be removed and a corresponding noise tolerance value which could keep the audio intelligible. From the input obtained by performing a focus group study, the noise tolerance value was set to 0.1 dB and silence duration, 0.5 seconds. Removing silence of duration 0.5 seconds temporally compressed the classroom lecture video (*A sample classroom lecture video*) to 64.3% of the original duration and Khan-style video (*A sample Khan style video*) to 89.8%. This is more than the typical 15-20% time reduction presented in (Gan and Donaldson 1988). The difference in time compression obtained is expected as more dead-time is present in classroom lecture compared to professionally shot Khan-style video.

3.1.1 Dead-time Removal Using FFmpeg

For dealing with silence present in audio stream, FFmpeg provides two filters

- *silencedetect* filter
- *silenceremove* filter

The filter *silencedetect* detects silence present in an audio stream. As per (*FFmpeg filters*), "This filter logs a message when it detects that the input audio volume is less or equal to a noise tolerance value for a duration greater or equal to the minimum detected noise duration. The printed times and duration are expressed in seconds". The filter accepts silence duration and noise tolerance as parameters.

The filter *silenceremove* removes silence present in an audio stream. The filter accepts many options such as start periods, start duration, threshold, etc. Though it works perfectly fine for all the audio streams, we used *silencedetect* filter, as we wanted to get the timestamps of silence present.

The timestamps for the start and end of pauses of duration 0.5 seconds was obtained as follows.

$$\begin{multline} ffmpeg \quad -i \quad videofile \quad -af \\ silencedetect = noise = 0.01 : \\ duration = 0.5 \quad -f \quad null \quad - \tag{3.1} \end{multline}$$

3.2 User Study-1

This section explains how we used a survey(User study-1) to analyze how students responded to a fluctuating playback rate.

3.2.1 Varying Playback

The key idea presented in this paper is a varying playback rate, where significant segments are played slower and insignificant segments faster. Before validating or introducing the concept, we performed a user study to analyze how students perceive a video with a fluctuating playback. We edited the video (*A sample Khan style video*) and a segment of duration 8 minutes and 17 seconds sliced from (*A sample classroom lecture video*) as given in algorithm 1.

Algorithm 1: Algorithm for Dead-time removal and fluctuating playback

```
Data: A lecture video
Result: Temporally compressed video with fluctuating playback rate
Find the time stamps of silence start and silence end of the input video using
  FFmpeg silence detect filter with duration 0.5 sec and noise tolerance 0.1 dB
Split the video based on the timestamps into n different segments,
  $V = \{v_1, v_2, ..., v_n\}$ , $v_i \epsilon V$ is a segment with no silence
foreach $v_i$ do
    if $duration(v_i) < 2$ seconds then
        set playback rate to meet a speaking rate of 265
        continue
    end
    Select $k$ where $k \leq duration(v_i)$ is an odd number
    Split $v_i$ into $\{v_i^1, v_i^2, v_i^3, ..., v_i^k\}$ with duration($v_i^1$) = ... = duration($v_i^k$)
    Change the playback rate of $v_i^{(k+1)/2}$ by a multiplicative factor $x$ to meet the
      speaking rate of 254 words per minute (wpm)
    Set playback rate of $v_i^{(k+1)/2+1}$ and $v_i^{(k+1)/2-1}$ to $x + 0.1$
    Set playback rate of $v_i^{(k+1)/2+2}$ and $v_i^{(k+1)/2-2}$ to $x + 0.2$ and so on till setting
      a playback rate for $v_i^1$ and $v_i^k$ to $x + 0.1 * (k - 1)/2$
end
Combine the segments in order and output the video
```

Algorithm 1 sets a fluctuating playback rate for video segments identified between two pauses and combines them together. The minimum playback rate was set in such a way that speaking rate of 254 words per minute (wpm) was met. It was selected based on the observations in (Guo, Kim, and Rubin 2014), where it was asserted that students were able to follow lectures in which speakers spoke at that rate with no loss in student engagement. If the duration of a segment between two pauses is less than 2 seconds, we assume it as incapable of conveying something meaningful and is sped up to meet a much higher speaking rate of 265 wpm. The modified playback for the video used in User study-1 is shown in figure 3. X-axis of the graph represents time stamp of the video and Y-axis represents the multiplication factor for the playback rate. The plot Y=1 represents the normal playback rate of the video with pauses removed, with a speaking rate of 202 wpm. The step function in the graph shows the way in which the playback rate was modified for the video. The graph shows the rate for the first 23 seconds only.

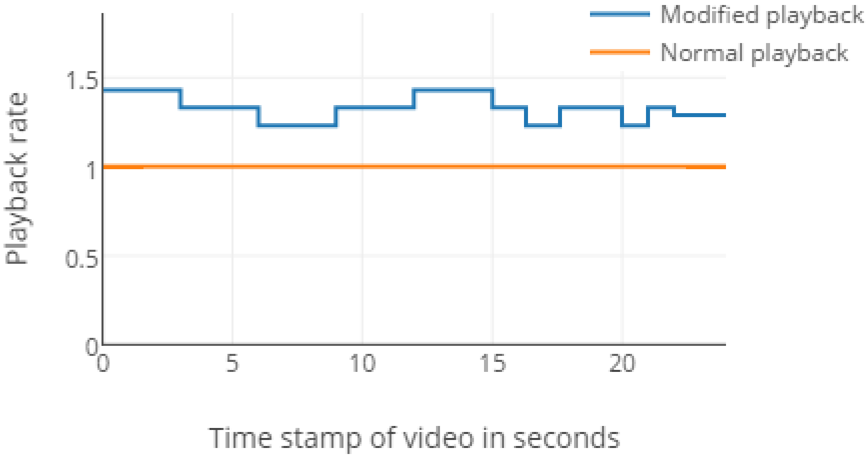


Figure 3: Modified Playback for User Study-1

3.2.2 Modifying the Playback Rate Using FFmpeg

FFmpeg was used for splitting the video at different timestamps marked by pauses and to change the playback rates. If the end of first pause is found out at $timestamp1$ and start of second pause is found at $timestamp2$, the segment in between those two pauses are obtained using the command given below.

$$ffmpeg \quad -i \quad videofile \quad -ss \quad timestamp1 \quad -t \\ timestamp2 - timestamp1 \quad outputfile \qquad (3.2)$$

In the above command, $timestamp2 - timestamp1$ gives the duration of the segment.

If the speaking rate of the speaker in the video after dead-time is removed is x, and the speaking rate that we are planning to achieve is y, the playback rate of the video is modified as shown below.

$$\begin{multline} ffmpeg \quad -i \quad videofile \quad -filter\_complex \\ \text{“}[0:v]setpts = y/x * PTS[v]; [0:a]atempo = x/y[a]\text{”} \\ -map \quad \text{“}[v]\text{”} \quad -map \quad \text{“}[a]\text{”} \quad outputfile \tag{3.3} \end{multline}$$

Once all the segments are modified to corresponding playbacks, the video segments are concatenated using the command below.

$$\begin{multline} ffmpeg \quad -i \quad concat : \text{“}segment1|segment2|...|segmentn\text{”} \\ -c \quad copy \quad outputfile \tag{3.4} \end{multline}$$

The video obtained using the above technique was then analyzed using the responses obtained from students as explained in the next section.

3.2.3 Analysis

In User study-1, we used the original video along with three different modifications of a Khan-style video and a classroom lecture video to study effectiveness.

- Video 1: The original video without any modifications.
- Video 2: The video with all the dead-time removed.
- Video 3: The video with all the dead-time removed and playback rate modified at a constant rate of 254 wpm.

- Video 4: The video with all the dead-time removed and playback rate modified using algorithm 1.

The compression rates obtained for a Khan-style video are provided in table 1.

Compression of Videos	
Video Type	Duration in seconds
Video 1	314
Video 2	279
Video 3	236
Video 4	208

Table 1: Time Reduction - Khan-Style Lecture Videos

The compression rates obtained for a classroom lecture video are provided in table 2.

Compression of Videos	
Video Type	Duration in seconds
Video 1	497
Video 2	321
Video 3	256
Video 4	232

Table 2: Time Reduction - Classroom Lecture Videos

For a Khan-style video lecture, we obtained a time compression rate of 33.75% and for classroom lecture video, a compression rate of 53.3%. Which means, the duration of the Khan-style video lecture obtained using our technique is 66.24% of the original and that of class room lecture video is 46.67% of the original. The difference of 19.5% in compression rates for Khan-style video lectures and classroom lecture videos is expected because the former is professionally shot with minimal dead-time and at a playback rate pleasing to the student community.

To understand the effectiveness of the above eight videos a user study was conducted in which 219 computer science graduate and undergraduate students participated. We created eight different survey forms for each of the eight videos, and a randomly selected form was shown to a student. The student was then asked to rate the experience on a scale of 1 to 5 for the metrics - satisfaction on current playback rate, audio clarity, video clarity, ability to understand what is being taught in the lecture and prior knowledge of the topic presented. A rating of 5 means the student is completely satisfied and a rating of 1 means the student is totally dissatisfied. The videos for the survey were carefully chosen, which accurately modelled the styles of each of the classes. The ability to understand the lecture and audio clarity were the two metrics which were given utmost importance. The mean values were calculated for responses obtained and bar graphs were plotted. The mean values for each of these metrics are plotted along Y-axis and metrics along X-axis. The bar graph for a Khan-style video is shown in figure 4. The responses obtained for classroom lecture videos were also plotted as bar graphs as shown in figure 7.

Using the graph for the original video as the baseline, we compare and interpret the user experience results obtained for the videos developed by algorithm 1. For a detailed study of the graph given in figure 4, the graph was restructured into two :-

1. Responses from students who indicated they have little knowledge of the topic, figure 5.
2. Responses from students who indicated they have been previously exposed to the topic, figure 6.

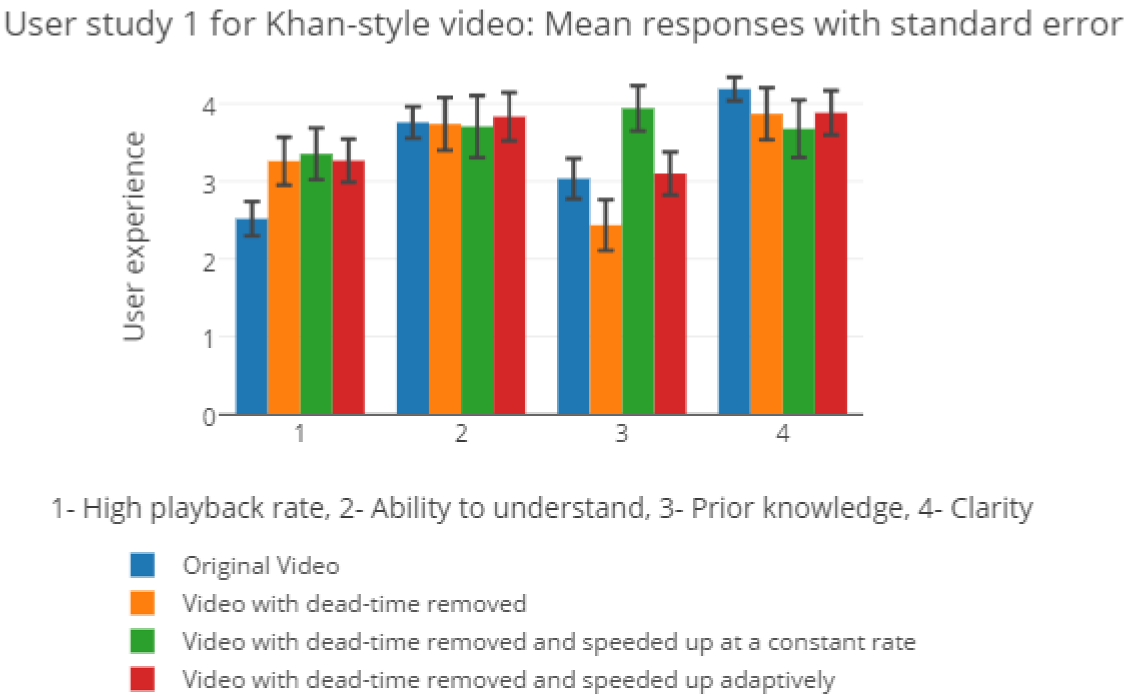


Figure 4: Mean Video Response with Standard Error: Khan-Style Video

Using the graph for the original classroom video as the baseline, we compare and interpret the user experience results obtained. For a detailed study of the graph given in 7, it was restructured into two:-

1. Responses from students who indicated they have little knowledge of the topic, figure 8.
2. Responses from students who indicated they have been previously exposed to the topic, figure 9.

From an initial glance of the bar graphs plotted, it looks like the mean values are different from each other. But, a robust statistical approach is needed to conclude something from the graph. The statistical test which we performed is provided in the next section.

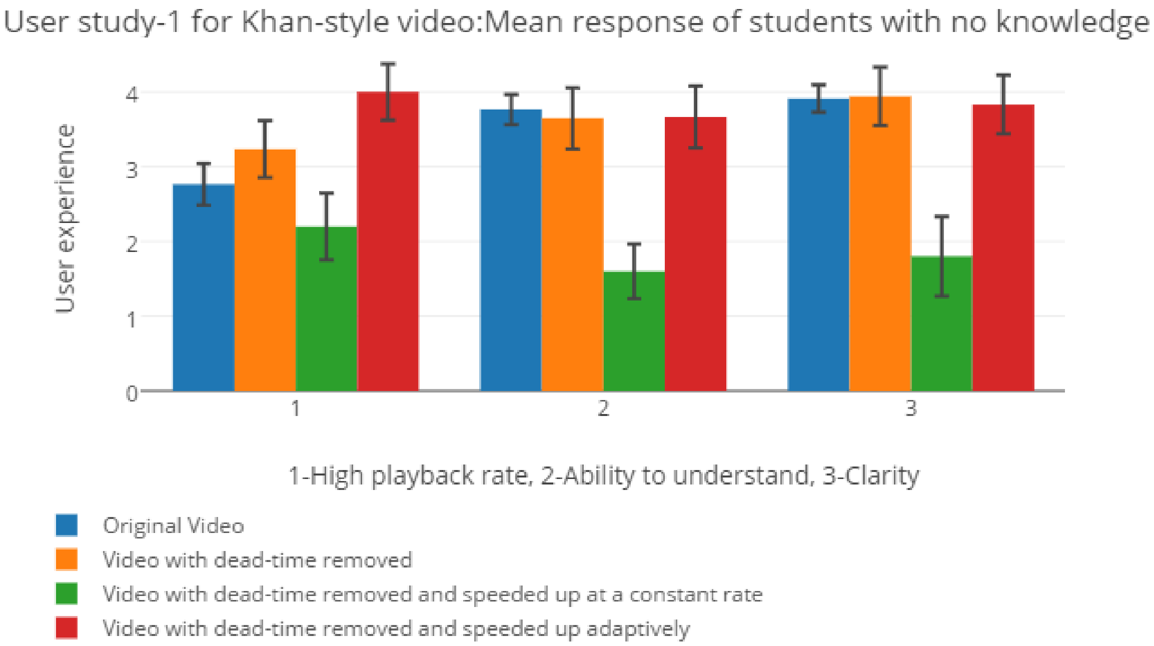


Figure 5: Video Response to Khan-Style Video: Students with Little Knowledge on the Topic with Standard Error

3.2.4 Statistical Validation

The mean values obtained for each of these video viewing sections need to be validated against each other through a statistical test. A t-test is the most used technique in statistics for this purpose. We explain what a t-test is in detail and then explain how this was used on our data.

Consider figure 10, taken from (*t-test*) for illustration purposes. In all the three situations, the difference between means is the same, even though the situations look different. In all the three cases, the variability of normal distributions are different. First one with moderate variability, second with high variability and third with low variability. From the figure, it is clear that for two populations with two different variability, it is not possible to compare the means directly. So, a t-test needs to be

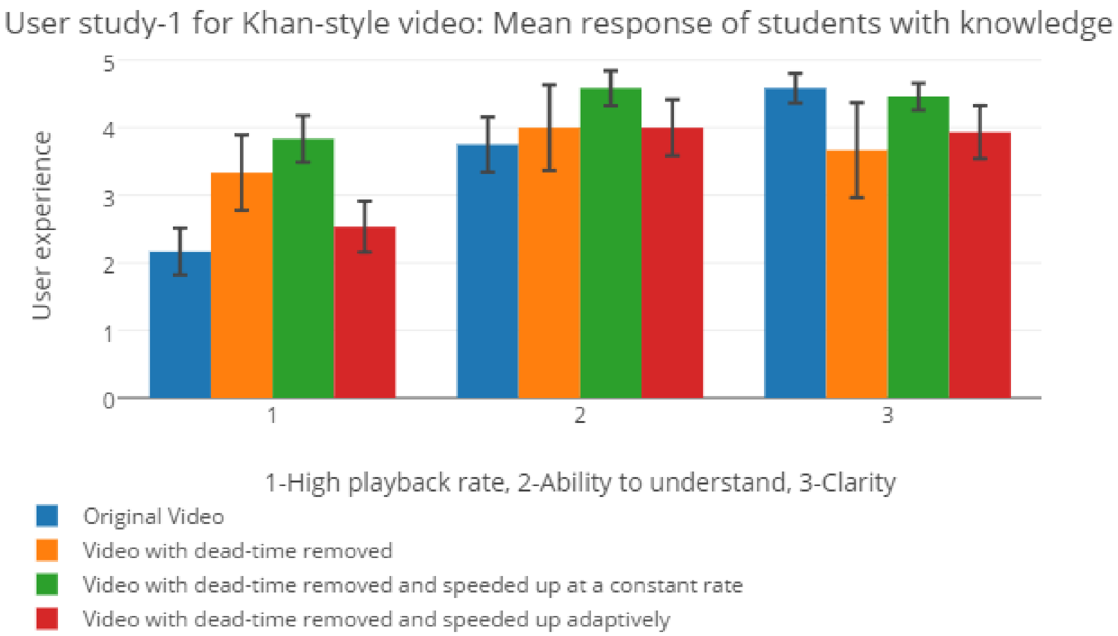


Figure 6: Video Response to Khan-Style Video: Students with Knowledge on the Topic with Standard Error

used. A t-test examines whether two samples are significantly different from each other. In our case, the variances are not known correctly as the sample size is small. So a t-test gives the idea if the population is different or not.

For each of the videos, Video 1, Video 2, Video 3 and Video 4, we performed t-test and results are presented from tables 3 to 8. The t-test was performed at a confidence value of p<0.01.

Table 3: T-Test Values for Khan-Style Lecture Videos: Video 1 and Video 2.

Video 1 and Video 2					
Playback Rate		Ability to Understand		Clarity	
t-value	p-value	t-value	p-value	t-value	p-value
-2.01123	0.024854	0.05153	0.479554	0.93406	0.17738

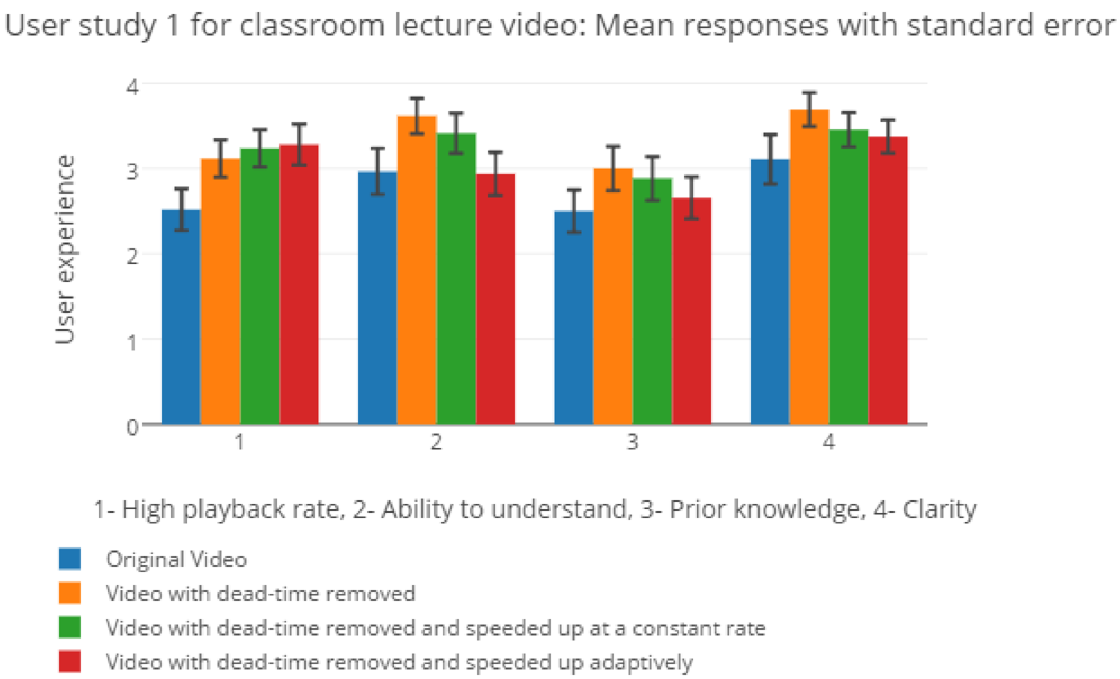


Figure 7: Mean Video Response with Standard Error: Classroom Video

Table 4: T-Test Values for Khan-Style Lecture Videos: Video 1 and Video 3.

Video 1 and Video 3					
Playback Rate		Ability to Understand		Clarity	
t-value	p-value	t-value	p-value	t-value	p-value
-2.18277	0.017216	0.13065	0.448324	1.4935	0.071222

Table 5: T-Test Values for Khan-Style Lecture Videos: Video 1 and Video 4.

Video 1 and Video 4					
Playback Rate		Ability to Understand		Clarity	
t-value	p-value	t-value	p-value	t-value	p-value
-2.09937	0.020112	-0.19984	0.421157	0.93586	0.176647

The t-test shows that there is no significant difference between the video produced by the technique and the original video at a significance value of $p<0.01$. Though the metrics used are not accurate indicators of how much the students were able to grasp, the result shows that a video having varying playback rate at different segments do not hinder student engagement.

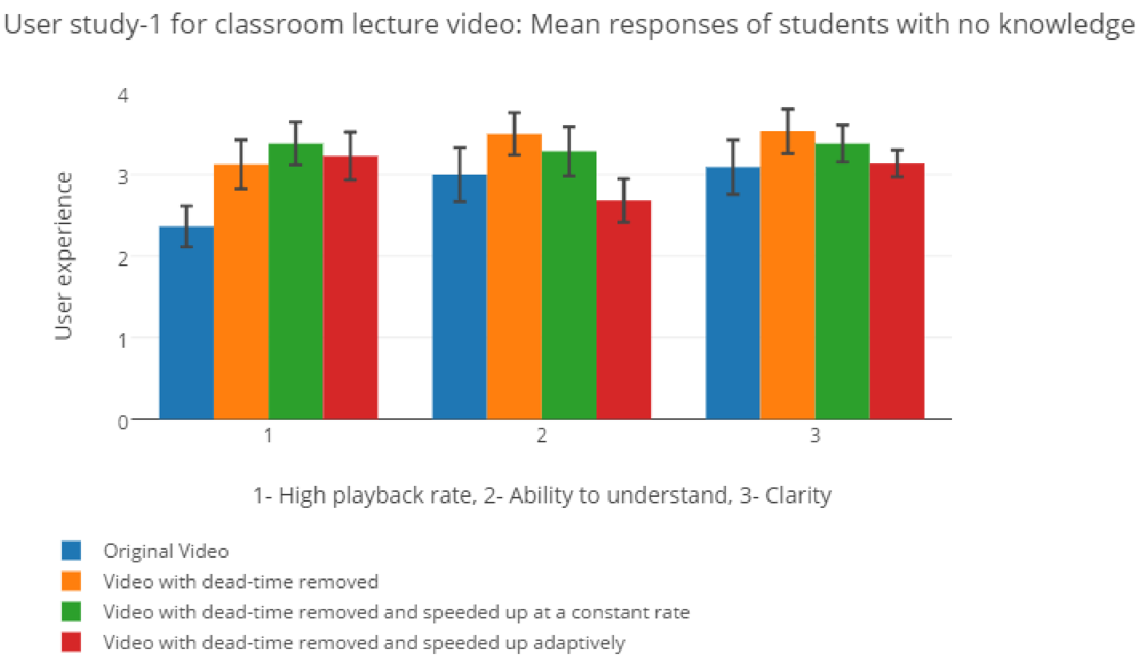


Figure 8: Video Response to Classroom Video: Students with Little Knowledge on the Topic with Standard Error

Table 6: T-Test Values for Classroom Lecture Videos: Video 1 and Video 2.

Video 1 and Video 2					
Playback Rate		Ability to Understand		Clarity	
t-value	p-value	t-value	p-value	t-value	p-value
-1.55042	0.063553	-1.8909	.032107	-1.64031	0.053489

Table 7: T-Test Values for Classroom Lecture Videos: Video 1 and Video 3.

Video 1 and Video 3					
Playback Rate		Ability to Understand		Clarity	
t-value	p-value	t-value	p-value	t-value	p-value
-1.91819	.029924	-1.25461	0.107243	-1.01686	0.156652

3.3 Ranking Segments

Results obtained from User study-1 indicate that having a variable playback does not affect student engagement. Contrary to the conventional way of having a constant

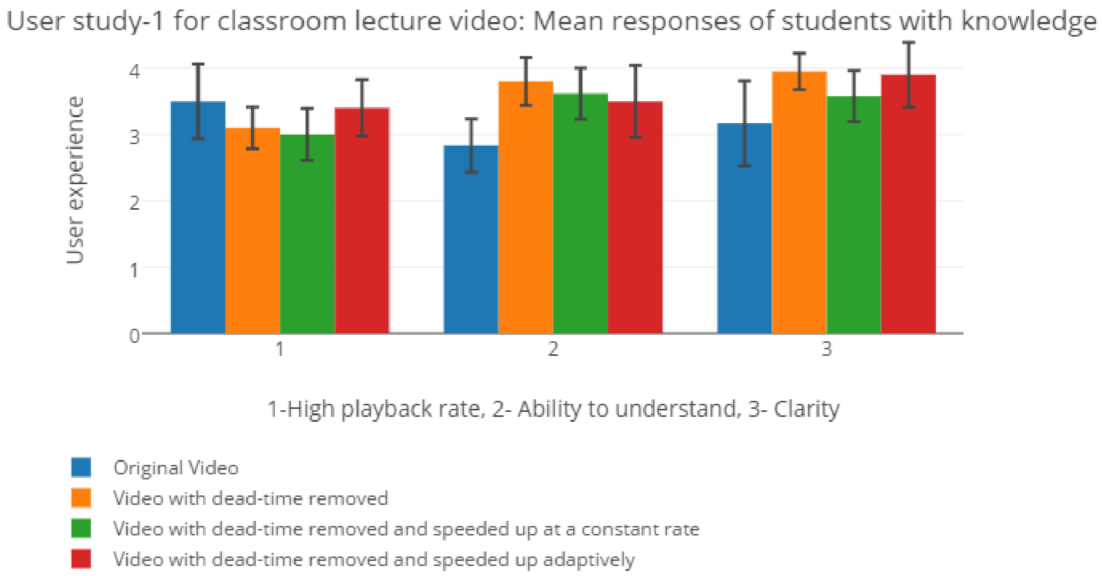


Figure 9: Video Response to Classroom Video: Students with Knowledge on the Topic with Standard Error

Table 8: T-Test Values for Classroom Lecture Videos: Video 1 and Video 4.

Video 1 and Video 4					
Playback Rate		Ability to Understand		Clarity	
t-value	p-value	t-value	p-value	t-value	p-value
-1.97262	0.026656	0.07231	0.471301	-0.78177	0.218764

playback rate, say a multiplication factor of 1.5(1.5x) or 2(2x) of the original rate, we claim that having a varying playback rate can increase compression rate. For instance, if a lecture can be clearly followed at 1.5 times the original rate, we propose a playback rate varying between 1.5x and a higher value, with the most relevant segments played at 1.5x.

The video was first sliced into segments whenever a pause of duration 0.5 seconds or more was found. The duration of 0.5 seconds which we identified using trial-and-error, closely matches with the findings provided in (Yang 2003). (Yang 2003) makes an observation that an average pause duration that indicates a phrasal boundary is 0.461908 seconds. Based on the observation, we assume that each of the segments

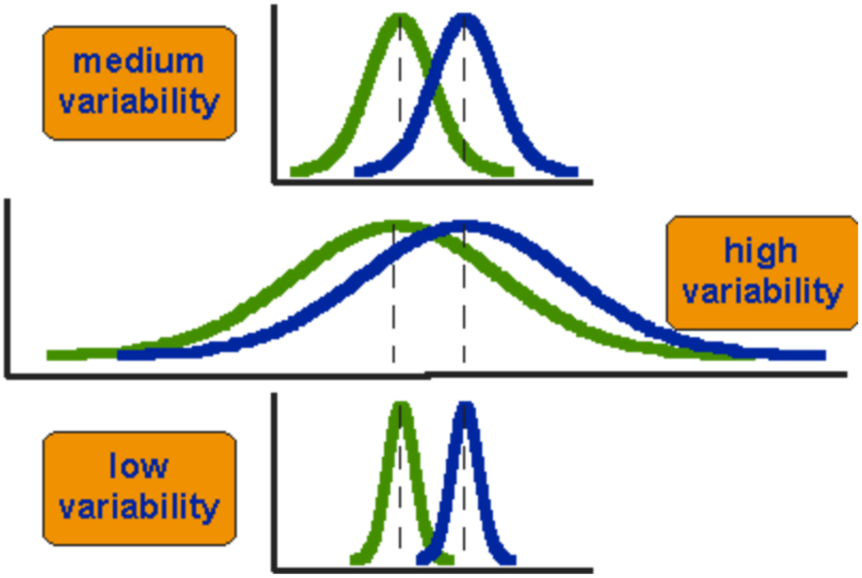
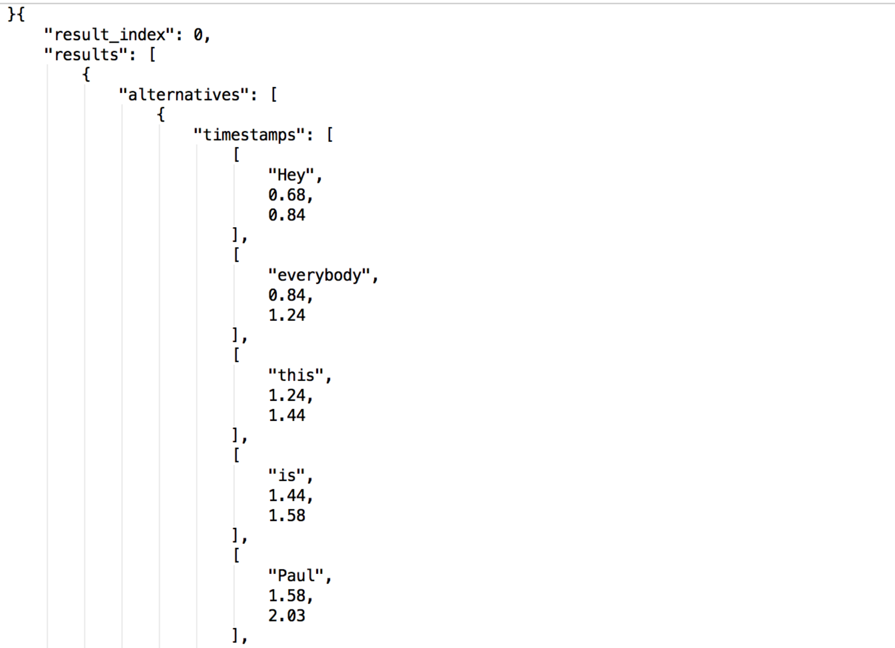


Figure 10: Different Scenarios for Differences Between Means. Image courtesy:- (*t-test*)

contains a meaningful phrase and can be ranked based on importance. Each of the segments was then transcribed using IBM Watson API (*IBM Speech to Text API*) into a text file. We applied two techniques mentioned in (Gong and Liu 2001) for finding out relevance of the phrases marked by pauses. (Gong and Liu 2001) is a seminal research which discusses two algorithms for text summarization - 1) using standard IR technique and 2) using latent semantic analysis(LSA) or singular value decomposition(SVD) method.

3.3.1 Transcribing Using IBM Watson

IBM Watson speech-to-text service is one of the state-of-the-art techniques available for transcribing. The service is accessed through a WebSocket connection or REST API (*IBM Watson API Reference*). A username and password was created for this purpose, which works as service credentials. For each of the segments marked by pauses, we extracted the audio in wav file format (*wav format documentation*). The API was then used for transcribing each of these wav files. The API provides the output in JSON format as shown in figure 11.



```
}{
    "result_index": 0,
    "results": [
        {
            "alternatives": [
                {
                    "timestamps": [
                        [
                            "Hey",
                            0.68,
                            0.84
                        ],
                        [
                            "everybody",
                            0.84,
                            1.24
                        ],
                        [
                            "this",
                            1.24,
                            1.44
                        ],
                        [
                            "is",
                            1.44,
                            1.58
                        ],
                        [
                            "Paul",
                            1.58,
                            2.03
                        ],
```

Figure 11: Output Obtained from IBM Watson Speech-to-text API in JSON

The content is extracted out from JSON format and the data is validated manually by comparing with the lecture. We found that for crisp and clear videos the text extracted was an exact match but for the video at hand, a minor manual editing of less was than fifteen minutes was needed to clean up the data. The text is extracted from each of these segments (phrases) and the two algorithms explained in (Gong and Liu 2001) are used for ranking the phrases.

3.3.2 Ranking Using Standard IR Technique

The standard IR algorithm discussed in (Gong and Liu 2001) selects sentences based on relevance measure for summarization. This method first decomposes a document, which needs to be summarized into individual sentences, and creates a weighted term-frequency vector for each of the sentences. If $T_i = [t_{1i} \quad t_{2i} \quad .... \quad t_{ni}]$ is the term-frequency vector for passage i, element t_{ji} denotes the frequency in which term j occurs in passage i.

The operation of the algorithm explained in (Gong and Liu 2001) is as follows, "Decompose the document into individual sentences, and use these sentences to form the candidate sentence set S. Create the weighted term-frequency vector A_i for each sentence $i \epsilon S$, and the weighted term-frequency vector D for the whole document. For each sentence $i \epsilon S$, Compute the relevance score between A and D, which is the inner product between A_i and D. Select sentence k that has the highest relevance score, and add it to the summary. Delete k from S, and eliminate all the terms contained in k from the document. Recompute the weighted term-frequency vector D for the document. If the number of sentences in the summary reaches the predefined value, terminate the operation; otherwise go to step 3"

The algorithm given above is modified for the situation at hand as follows,

1. Split the video into segments, whenever a pause of duration 0.5 sec or more is identified.
2. Extract the speech as text for each of these segments i and represent the extracted phrase as term-frequency vector A_i. D represents the term-frequency vector of the whole text.

3. For each of these phrases, find out the relevance as inner product between A_i and D.
4. Select the phrase that has the highest relevance and remove it from the text.
5. Go to step 3 until all the phrases are scored.

The major modifications made for our research are - 1) There is no predefined value for the number of phrases; 2) No removal of terms from the text even after a phrase is selected. The main concern of our research is to find out which phrases are more important than others and not summarization of phrases. So using a predefined number of phrases is not a good idea, as this can result in irregularities in the video. The terms from a phrase already selected are not removed because we believe if a speaker repeats a phrase, it has some major content.

Once the relevance scores are obtained, the scores are linearly mapped to a range of minimum and maximum playback rates. The linear function associates the video segment associated with the most relevant phrase to the minimum playback rate and the least relevant to the maximum playback rate. All the other segments are linearly mapped to the values in the selected interval. The playback rate of the video segments are then modified using FFmpeg and combined.

3.3.3 Ranking Using Latent Semantic Analysis

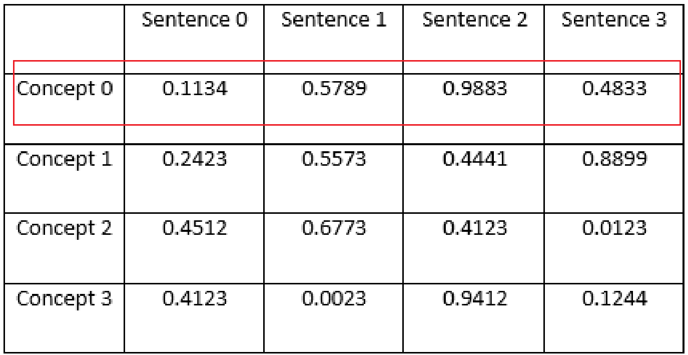
(Landauer, Foltz, and Laham 1998) explains latent semantic analysis (LSA) as “a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text. The underlying idea is that the aggregate of all the word contexts, in which a given word does and does not appear provides a set of mutual constraints that largely determines the similarity of meaning of words and sets of words to each other."

The first step in LSA, as given in (Landauer, Foltz, and Laham 1998), is "to represent the text as a matrix, in which each row stands for a unique word and each column stands for a text passage or other context. Each cell contains the frequency with which the word of its row appears in the passage denoted by its column. Next, the cell entries are subjected to a preliminary transformation, whose details we will describe later, in which each cell frequency is weighted by a function that expresses both the word's importance in the particular passage and the degree to which the word type carries information in the domain of discourse in general. Next, LSA applies singular value decomposition (SVD) to the matrix. This is a form of factor analysis, or more properly the mathematical generalization of which factor analysis is a special case. In SVD, a rectangular matrix is decomposed into the product of three other matrices. One component matrix describes the original row entities as vectors of derived orthogonal factor values, another describes the original column entities in the same way, and the third is a diagonal matrix containing scaling values such that when the three components are matrix-multiplied, the original matrix is reconstructed."

(Gong and Liu 2001) uses this technique to summarize documents. The method discussed is as follows,

"Decompose the document D into individual sentences, and use these sentences to form the candidate sentence set S, and set $k = 1$. Construct the terms by text segment matrix A for the document D. Perform the SVD on A to obtain the singular value matrix $\sum$, and the right singular vector matrix V^T. In the singular vector space, each sentence i is represented by the column vector $\psi_i = [v_{i1}v_{i2}...v_{ir}]^T$ of V^T Select the k'th right singular vector from matrix V^T. Select the sentence which has the largest index value with the k'th right singular vector, and include it in the summary. If k reaches the predefined number, terminate the operation; otherwise, increment k by one, and go to Step 4."

The working of the above algorithm is explained in detail in (Ozsoy, Cicekli, and Alpaslan 2010). As per (Ozsoy, Cicekli, and Alpaslan 2010), V^T matrix is used for selecting the most important sentences required for summary. The columns of the matrix represent the sentences and rows represent the concepts. The importance of the concepts decreases as we go down the columns. Cells of the matrix indicates the relationship of the sentence with a concept. Higher the cell value, higher is the relationship of the sentence with the concept. For instance, if a V^T matrix is as given in figure 12, the importance of concepts decreases as concept 0 > concept 1 > concept 2 > concept 3. For concept 0, sentence 2 is the most related one. For concept 1, sentence 3 is the most related sentence and so on.



	Sentence 0	Sentence 1	Sentence 2	Sentence 3
Concept 0	0.1134	0.5789	0.9883	0.4833
Concept 1	0.2423	0.5573	0.4441	0.8899
Concept 2	0.4512	0.6773	0.4123	0.0123
Concept 3	0.4123	0.0023	0.9412	0.1244

Figure 12: A Sample V^T Matrix

For this situation, the algorithm is modified to match our needs as follows,

1. Split the video into segments whenever a pause of duration 0.5 sec or more is identified.

2. Extract the speech as text for each of these segments i and let D represent the whole text.
3. Construct the terms by phrase matrix A for D.
4. SVD is performed on A to obtain the singular value matrix $\sum$ and the right singular vector matrix V^T as explained in (Gong and Liu 2001).

For each of the concepts present in V^T, the most relevant phrase is found out, which gives the order of importance. If a phrase was already selected, another phrase closest to the concept was chosen. The video segments associated with the phrases are then linearly mapped to a playback rate interval and modified using FFmpeg and combined.

3.3.4 Mapping to a Playback Interval

Once the relevance scores of segments are found out or ranked, the segments are mapped to a preselected playback interval. Identifying the best playback rate range is out of scope of this research. Even though the study (Williams 1998) suggests an optimal rate of speech is 160 wpm, (Guo, Kim, and Rubin 2014) puts forward that students were able to follow even a rate of 254 wpm. But, we believe that finding out a range for playback rate which is the best for all types of lectures, subjects and students who come from different cultural backgrounds with varying intellectual capabilities is difficult.

If $x \epsilon \, [A, B]$ and we need to map x to $[P, Q]$ then,

$$y = (x - A) * (P - Q)/(B - A) + P, y \epsilon [P, Q]$$

.

In our case, $[A, B]$ is the relevance intervals and $[P, Q]$ is the playback rate interval selected. Since we need to map most relevant segment to lowest playback and lower relevant segment to highest playback, we use the linear mapping with a minor modification as shown below,

$$y = P - (x - B) * (Q - P)/(B - A)$$

This makes sure that value A is mapped to Q and B is mapped to P and other values accordingly. This modification of playback rates based on relevance is the main contribution of this research.

Chapter 4

EVALUATION

In this section, we describe how we evaluated the effectiveness of the video created using our technique through a user study (User study-2). The videos created using the IR technique and latent semantic analysis were of comparable duration and therefore we chose to only validate videos created using the IR technique.

4.1 User Study-2

On analyzing the extracted segments, we found that some segments did not contain a completely meaningful phrase. Therefore, the text was manually divided into conceptual units conveying a meaning. These units were then scored with the IR technique. Using the timestamps obtained using IBM Watson API, the video was split into segments, with each segment now conveying a stand-alone idea. Playback rate of each of these segments was then modified using relevance scores and combined. As the video used for evaluation purpose dealt with heaps and arrays, we requested one of the data structures instructors to score the manually identified phrases and the phrases marked by pauses. We used these ranks to create one more video. In brief, we used 5 videos for User study-2, namely

- Video 1: The original video without any modifications.
- Video 2: The video with dead-time removed and playback modified for automatically identified phrases as ranked by IR technique.

- Video 3: The video with dead-time removed and playback modified for manually identified phrases ranked by IR technique.
- Video 4: The video with dead-time removed and playback modified for automatically identified phrases as ranked by the instructor.
- Video 5: The video with dead-time removed and playback modified for manually identified phrases as ranked by the instructor.

The duration of each of these videos are provided in table 9.

Table 9: Duration of Videos Used for User Study-2

Duration of Video in Seconds	
Video 1	314
Video 2	182
Video 3	177
Video 4	206
Video 5	193

The manually extracted phrases with phrase numbers are provided in table 10

Table 10: Manually Identified Phrases

Phrase #	**Manually Extracted Phrases from Khan-style Video**
1	Hey everybody this is Paul.
2	Welcome to part three in the introduction to a heap data structure.
3	So in this tutorial am basically going to be showing you how we can translate a tree structure like this a heap as a tree into a heap as an array structure.
4	So basically I just went ahead

Continued on next page

Table 10 – *Continued from previous page*

Phrase #	**Manually Extracted Phrases from Khan-style Video**
5	And I did this already and am gonna explain to you how I filled in this array.
6	So basically you just start with the root here and the root is going to go in index zero of the array.
7	I got twenty through an index zero and then to fill in the rest of the array we simply just go down the tree from left to right and fill in the values.
8	So thirteen and nine would come next.
9	Thirteen and nine are there.
10	And then eight five three and seven eight five three and seven eight five three and seven
11	And then we go down to the bottom row six two and one six two and one six two and one
12	So it is pretty simple to see the relationship between the tree and the array when you look at that way.
13	But we need a way to be able to let our computer know or program know what the relationship is between the parent and the child.
14	So it is really easy to see in this tree structure but not so easy to see in this array structure.

Continued on next page

Table 10 – *Continued from previous page*

Phrase #	Manually Extracted Phrases from Khan-style Video
15	So basically i have written down the algorithms that basically tell us which child belongs to which parent and what children the parents have.
16	Basically for looking for the children of a certain parent n is going to be the index of our array n is going to be the index of our array
17	And if we are looking at some certain index of our array then it will have children at two times that index plus one and another child at two times the index plus two.
18	For example if we wanted to find the children of our root node here you basically look at its corresponding index.
19	So the root has an index of zero.
20	And so we are going to use that index number as our n here.
21	So basically array index zero will have children at two times zero plus one and two times zero plus two.
22	Two times zero plus one is one two times zero plus two is two.
23	So we end up with a one and a two which tell us which indexes hold the children.
24	So one and two are the indexes and they hold the children thirteen and nine.
25	So twenty is the parent of thirteen and nine just like we see in our tree here.

Continued on next page

Table 10 – *Continued from previous page*

Phrase #	Manually Extracted Phrases from Khan-style Video
26	So if we were to try to figure out what the children of the number nine is nine is held in index two so we use that index number as our n
27	So two times two which is our index number two time two is four plus one is five
28	So index two has a child in index five.
29	Nine has a child of three and their it is on our tree.
30	So if we look at the other child here two times two the index we are looking at is two
31	Two times two is four plus two is six.
32	So index two has a child in index six.
33	So index two holds nine index six holds seven nine has a child of seven and we can see that in our tree as well.
34	We can use this all the way down the tree.
35	And this relationship right here basically tells us if we have some child index where is its parent located.
36	so the parent of a child is going to be located at the child's index minus one divided by two.
37	And then we take the floor of that.

Continued on next page

Table 10 – *Continued from previous page*

Phrase #	**Manually Extracted Phrases from Khan-style Video**
38	So this is just a mathematical notation called the floor which basically means that if we have some numbers say like three point five for example we just round it down.
39	So that is really what the floor means is we are just going to round this number down to the nearest integer.
40	So for example if we were looking at index seven well seven holds the value six so in our tree we are looking at this part right here
41	And we want to figure out who is the parent of index seven.
42	So plugging in the index seven here seven minus one is going to be six.
43	Six divided by two is three.
44	And so then we basically say okay we will use index seven we got index three as our outcome.
45	That means that six is the child of eight. and we can see that in our tree here.
46	And we can also look at if we wanted to this one right here
47	Two is located in index eight.
48	Hence we plug our index eight in here.
49	Eight minus one is going to be seven.
50	Seven divided by two is three point five.
51	We take the floor of that number which becomes three

Continued on next page

Table 10 – *Continued from previous page*

Phrase #	**Manually Extracted Phrases from Khan-style Video**
52	We ended up with three as our result when we plug in the value eight.
53	Index eight is the child of whatever is in index three.
54	Or the value in index eight is the child of the value in index three.
55	So two is the child of eight just like we see in our graph here.
56	Anyway that is basically how you can describe a heap as an array and basically that if we have some numbers say like three point five for example we just round it down between the child and a parent once they are in the array elements.
57	Anyways thank you guys for watching.
58	Have an excellent day and if you have not already do not forget to subscribe.

The phrases identified by the application using silence detection of 0.5 seconds is given in table 11.

Table 11: Automatically Identified Phrases

Phrase #	Automatically Extracted Phrases from Khan-style Video
1	hey everybody this is Paul welcome to part three in the introduction to heap data structure so in this tutorial I am basically going to be showing you how we can translate a tree structure like this a heap as a tree into a heap as an array structure
2	so basically I just went ahead and did this already
3	and I am going to explain to you how I filled in this array
4	so basically you just start with the root here
5	and the root is going to go in index zero of the array I have got twenty through an index zero
6	and then to fill in the rest of the array we simply just go down the tree from left to right and fill in the value
7	so thirteen and nine would come next thirteen and nine are there
8	and then eight five three and seven eight five three and seven and then we go down to the bottom row six two one six two and one
9	so it is pretty simple to see the relationship between the tree and the array when you look at at that way
10	but
11	we need a way to be able to let our computer know or our program know
12	what the relationship is between the parent and the child

Continued on next page

Table 11 – *Continued from previous page*

Phrase #	Automatically Extracted Phrases from Khan-style Video
13	so it is really easy to see in this tree structure but not so easy to see in this array structure
14	so basically I have written down the algorithm that basically tells us which child belongs to which parent and what children the parents have
15	so basically for looking for the children of a certain parent
16	n is going to be the index of our array
17	and if we are looking at some certain index of array
18	then it will have children at two times that index plus one and another child at two times the index plus two
19	so for example if we wanted to find the children of our root node here
20	you basically look at its corresponding index
21	so the root has an index of zero
22	and so we are going to use that index number as our n here
23	so basically array index zero will have children at two times zero plus one and two times zero plus two
24	two times zero plus one is one two times zero plus two is two
25	so we end up with a one and a two which tell us which indexes
26	hold these children

Continued on next page

Table 11 – *Continued from previous page*

Phrase #	**Automatically Extracted Phrases from Khan-style Video**
27	so one and two are the indexes and they hold the children thirteen and nine
28	so twenty is the parent of thirteen and nine just like we see in our tree here
29	so if we were to try to figure out what the children of the number nine is
30	nine is held in index two so use that index number as our n
31	so two times two which is our index number two times two is four plus one is five
32	so index two
33	has a child
34	in index five
35	nine has a child of three and there it is on our tree
36	so if we look at the other child here
37	two times two the index that we are looking here is two two times two is four plus two is six so index two
38	has a child in index six
39	so index two holds nine index six holds seven nine has a child of seven and we can see that in our tree as well
40	we can use this all the way down the tree and then this relationship right here basically tells us

Continued on next page

Table 11 – *Continued from previous page*

Phrase #	Automatically Extracted Phrases from Khan-style Video
41	if we have some child index where is its parent located
42	so the parent of a child is going to be located at each child index minus one divided by two
43	and then we take the floor of that so this is just a mathematical notation called the floor which basically means that if we have some numbers say like three point five
44	for example we just round it down so that is really what the floor means is just we are going to round this number down to the nearest integer
45	so for example if we were looking at index seven
46	well seven holds the value six so in our tree we are looking at this part right here
47	and we want to figure out who is the parent of index seven
48	so plugging in the index seven here seven minus one is going to be six six divided by two is three
49	and so then we basically say okay we used index seven we got index three as our outcome that means that six is the child of eight and we can see that in our tree here
50	and we can also look at if we wanted to this one right here
51	two is located in index eight

Continued on next page

Table 11 – *Continued from previous page*

Phrase #	**Automatically Extracted Phrases from Khan-style Video**
52	hence we plug our index eight in here eight minus one is going to be seven seven divided by two is three point five we take the floor of that number which becomes three
53	we ended up with three as a result when we plug in the value eight
54	index eight is the child
55	of whatever is in index three
56	or the value an index eight is the child of the value in index thre
57	so
58	two is the child of eight just like we see in our tree here so anyway that is basically how you can describe a heap as an array
59	and basically how you translate between the tree and the array and the relationship between the child and a parent once they are in the array elements
60	anyway thank you guys for watching and have an excellent day
61	and that if you have not already do not forget to subscribe

Comparison of the relevance scores obtained using standard IR technique and manual scoring for manually identified phrases are given in table 12.

Table 12: Comparison of Relevance Scores of Manually Identified Phrases - IR Technique vs Manual

Phrase #	**Manual Relevance Score**	**Relevance Score Using IR Technique**
1	0	0.27139
2	2	1.6283
3	57	29.8529
4	2	0.2326
5	48	9.4986
6	48	16.8649
7	48	27.9919
8	48	0.5040
9	48	0.2713
10	48	4.4585
11	48	7.1724
12	2	10.4291
13	5	14.0347
14	4	8.2967
15	56	14.9652
16	52	20.4705
17	54	43.5775
18	53	7.5213
19	55	0.3877
20	38	3.6443

Continued on next page

Table 12 – continued from previous page

Phrase #	Manual Relevance Score	Relevance Score Using IR Technique
21	38	9.7700
22	3	4.2647
23	38	4.8850
24	38	13.2593
25	38	6.4358
26	38	33.7687
27	38	11.6697
28	38	2.4425
29	38	2.7139
30	38	17.9117
31	3	1.6283
32	38	2.4425
33	38	19.1911
34	40	1.2018
35	42	12.8716
36	41	12.4451
37	39	0.6203
38	3	25.3556
39	3	15.9732
40	28	22.4478
41	28	3.2566
42	28	6.0868

Continued on next page

Table 12 – continued from previous page

Phrase #	Manual Relevance Score	Relevance Score Using IR Technique
43	4	0.4264
44	27	10.8943
45	27	7.9090
46	24	3.6443
47	24	0.8141
48	24	0.5040
49	3	0.6590
50	3	1.0080
51	3	0.9304
52	4	1.7446
53	24	5.3890
54	24	8.9171
55	5	3.0240
56	2	58.0
57	0	0.2713
58	1	1.2018

Comparison of the relevance scores obtained using standard IR technique and manual scoring for automatically identified phrases are given in table 13.

Table 13: Comparison of Relevance Scores of Automatically Identified Phrases - IR Technique vs Manual

Phrase #	**Manual Relevance Score**	**Relevance Score Using IR Technique**
1	60	61.0
2	0	4.8086
3	51	5.6328
4	51	0.6869
5	51	18.1351
6	51	22.9436
7	51	3.9842
8	51	31.4159
9	11	17.2192
10	7	0.0458
11	59	1.5571
12	59	6.6861
13	6	11.8611
14	59	21.0203
15	11	5.9536
16	55	2.5646
17	11	8.8844
18	57	27.2485
19	53	3.9384
20	53	1.6486

Continued on next page

Table 13 – continued from previous page

Phrase #	Manual Relevance Score	Relevance Score Using IR Technique
21	52	1.6029
22	16	6.5488
23	11	13.1892
24	52	2.3814
25	52	3.7553
26	52	0.1832
27	52	16.1201
28	52	10.2125
29	51	9.4797
30	51	7.8311
31	11	9.8461
32	38	0.2748
33	51	0.1832
34	51	0.1832
35	51	3.1141
36	51	0.9617
37	11	24.6839
38	51	1.1449
39	51	24.6839
40	54	13.5556
41	54	4.2132
42	54	15.2042

Continued on next page

Table 13 – continued from previous page

Phrase #	Manual Relevance Score	Relevance Score Using IR Technique
43	60	41.5368
44	11	26.7905
45	49	2.7935
46	46	8.3806
47	49	4.4879
48	49	14.2425
49	49	46.8949
50	16	7.4647
51	48	0.9159
52	48	1.7446
53	48	2.015
54	48	0.4122
55	45	0.6411
56	48	10.8078
57	16	0.0458
58	21	22.2568
59	16	44.2387
60	0	4.7169
61	1	1.7860

Using latent semantic analysis gives ranks instead of relevance scores. Comparison of manual ranks and ranks obtained using latent semantic analysis for manually identified phrases is given in table 14 and automatically identified phrases in table 15.

Table 14: Comparison of Ranks of Manually Identified Phrases - Latent Semantic Analysis vs Manual

Phrase #	**Manual Ranks**	**Rank Using Latent Semantic Analysis**
1	58	56
2	56	26
3	1	38
4	56	22
5	10	3
6	10	17
7	10	7
8	10	15
9	10	10
10	10	44
11	10	27
12	56	40
13	53	45
14	54	2
15	2	58
16	6	18
17	4	14
18	5	33

Continued on next page

Table 14 – continued from previous page

Phrase #	Manual Ranks	Rank Using Latent Semantic Analysis
19	3	25
20	20	35
21	20	13
22	55	32
23	20	50
24	20	5
25	20	52
26	20	51
27	20	57
28	20	30
29	20	41
30	20	11
31	55	53
32	20	34
33	20	36
34	18	29
35	16	12
36	17	47
37	19	42
38	55	46
39	55	6
40	30	24

Continued on next page

Table 14 – continued from previous page

Phrase #	Manual Ranks	Rank Using Latent Semantic Analysis
41	30	48
42	30	16
43	54	54
44	31	20
45	31	55
46	34	8
47	34	37
48	34	1
49	55	49
50	55	31
51	55	19
52	54	4
53	34	28
54	34	9
55	53	43
56	56	23
57	58	39
58	57	21

Table 15: Comparison of Ranks of Automatically Identified Phrases - Latent Semantic Analysis vs Manual

Phrase #	Manual Ranks	Rank Using Latent Semantic Analysis
1	1	1
2	61	59
3	10	37
4	10	18
5	10	23
6	10	39
7	10	56
8	10	14
9	50	44
10	54	13
11	2	9
12	2	30
13	55	43
14	2	15
15	50	11
16	5	29
17	50	24
18	4	31
19	8	47
20	8	48

Continued on next page

Table 15 – continued from previous page

Phrase #	Manual Ranks	Rank Using Latent Semantic Analysis
21	9	53
22	45	40
23	50	4
24	9	25
25	9	61
26	9	5
27	9	20
28	9	60
29	10	35
30	10	3
31	50	7
32	10	41
33	10	28
34	10	27
35	10	46
36	10	2
37	50	6
38	10	17
39	10	58
40	7	50
41	7	38
42	7	26

Continued on next page

Table 15 – continued from previous page

Phrase #	Manual Ranks	Rank Using Latent Semantic Analysis
43	11	22
44	50	12
45	12	19
46	15	45
47	12	51
48	12	55
49	12	32
50	45	36
51	13	34
52	13	54
53	13	16
54	13	57
55	16	21
56	13	33
57	45	49
58	40	8
59	45	42
60	61	10
61	60	52

One of the limitations that was observed in the user study was the inability for the user to adjust the minimum and maximum playback rates. The videos resulting from our technique were set to meet a speaking rate varying between 254 wpm and 354 wpm. We set the minimum rate as 254 wpm based on the observations provided in (Guo, Kim, and Rubin 2014) and to analyze how students respond to very fast videos.

We created five different survey forms for each of these videos with eight questions (See Appendix B). The questions were designed to obtain information about the users' native language, confidence level about heap data structure, previous exposure to the subject and understanding of the content presented. Questions numbered 5 to 7 in the questionnaire examine students' mastery of the content. Out of the five forms, a randomly selected one was shown to the participant. The participant was then requested to watch the video and answer the questions that followed. A total of 110 students participated in the survey. Mean values of all the responses were computed and plotted as shown in figure 13. First metric indicates satisfaction of playback rate, second metric implies confidence level about the content presented and the third metric 'Test score' indicates how well the students answered questions 5 to 7. The response for metric 'Test score' was calculated based on the answers for these questions. A correct answer carried one point and an incorrect zero points. And therefore, maximum value possible for this metric was 3.

As we could not figure out if the difference of the mean was significant, a two-tailed t-test was performed on the data to compare Video 2 with other videos. Table 16 shows the results of the statistical test on first metric - playback rate. It indicates a dissatisfaction of range of rate selected compared to the original video, Video 1. An interview with a few students who indicated dissatisfaction, mentioned that they felt the speed was fast in spite of understanding the lecture clearly. This is where giving the user the ability to vary the range of playback rate would have helped. The t-test

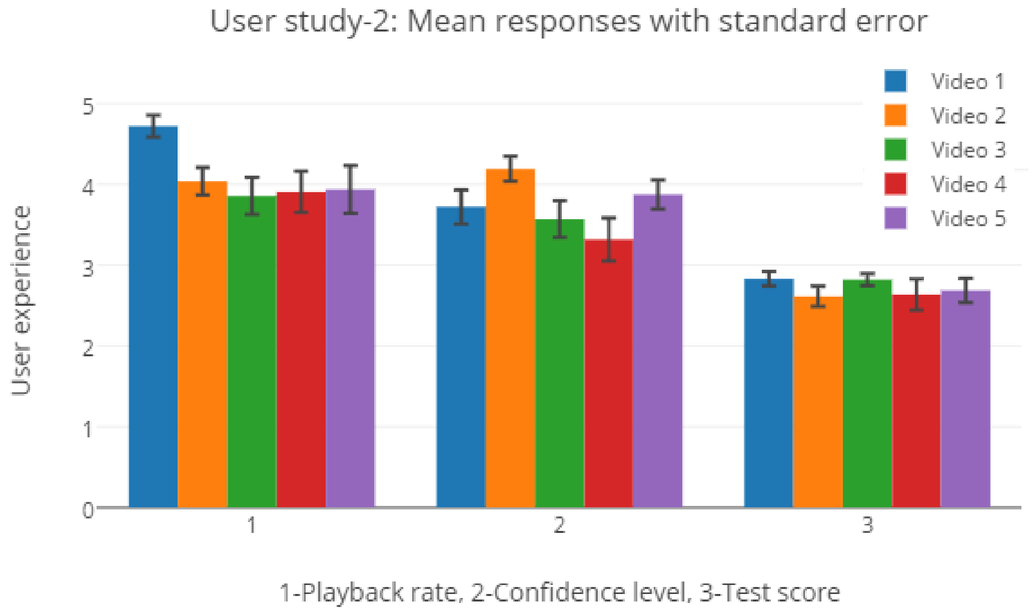


Figure 13: Video Experience Survey for Video Modified Using IR Technique

conducted on the third metric ‘Test score’ reinforced their statements and is given in table 17.

Table 16: T-Test on Playback Rate Satisfaction

T-Test on Playback Rate	t-value	p-value
Video 1 and Video 2	-2.91541	.005676
Video 3 and Video 2	0.62827	.53258
Video 4 and Video 2	0.43368	.666546
Video 5 and Video 2	0.31815	.752029

Table 17: T-test on Test Score

T-Test on Test Scores	t-value	p-value
Video 1 and Video 2	-1.29498	.2024
Video 3 and Video 2	-1.44463	.154562
Video 4 and Video 2	-0.0941	.925437
Video 5 and Video 2	-0.36353	.718125

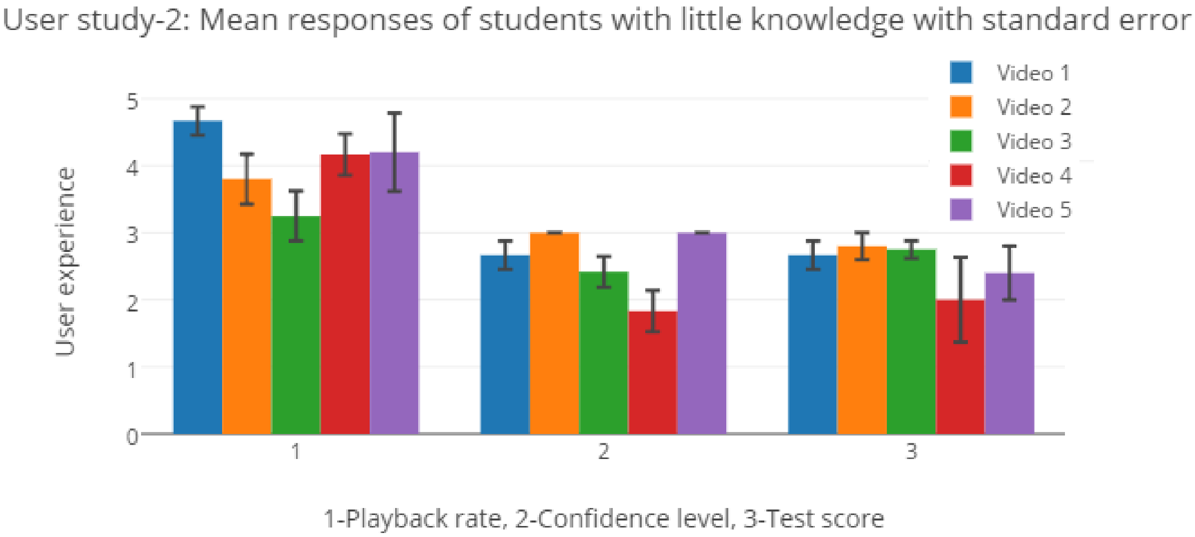


Figure 14: Video Experience Survey for Video Modified Using IR Technique - Participants with No Prior knowledge of the Topic

Table 18: T-Test on Test Score - Students with No Prior Knowledge of the Topic

t-test on test scores	t-value	p-value
Video 1 and Video 2	0.45227	.66178
Video 3 and Video 2	0.20831.	.837787
Video 4 and Video 2	1.10782	.296665
Video 5 and Video 2	0.89443	.397204

Table 17 signals that there is no significant difference in the mean values. But, table 17 cannot be used as the sole indicator to claim that students who watched our video were able to learn as much as those who watched the original video. This is because of the possibility that a student might have been proficient with the content much before taking the survey. In order to nullify such a possibility, we extracted the responses provided by students who indicated a low confidence about their knowledge and those who have not taken a related class and plotted as figure 14. The results of t-test on the data is shown in table 18, which indicates students were able to understand our video as much as that of the original video.

Chapter 5

CONCLUSION AND FUTURE WORK

In this thesis, we introduced a novel idea of temporally compressing an educational lecture video by varying playback rate. Initially, we proved that introducing a varying playback rate in a video does not affect the student engagement. This was studied using a user survey, in which, the playback rate was modified according to a step function and measuring various metrics, such as, clarity of the video, ability to follow, clarity of audio and playback rate satisfaction. It was statistically validated through a t-test. Finally, we studied various text mining algorithms using latent semantic analysis and standard information retrieval techniques. These techniques were successfully applied to identify relevant segments of a video. Then the playback rates were modified using a linear mapping. A user study was conducted with videos, with a speaking rate varying between 254 words per minute and 354 words per minute. The results were really promising, which showed no significant difference in student engagement between the video created using our technique and the original video. This technique can be easily employed by educators to save time and bandwidth, while preserving and at times enhancing the student experience. Incorporating a configurable minimum and maximum playback rate in a video instead of conventional playback rate change, as in YouTube or media players, will be more beneficial to students. Students will be able to understand the lecture in less time. In case, a student wants to watch the video at constant playback rate, it can be achieved by setting the minimum and maximum playback rate to same value.

REFERENCES

A sample classroom lecture video. https://goo.gl/gj2dKf.

A sample Khan style video. https://goo.gl/QrmzMu.

Arons, Barry. 1997. "SpeechSkimmer: a system for interactively skimming recorded speech." *ACM Transactions on Computer-Human Interaction (TOCHI)* 4 (1): 3–38.

Audacity. http://www.audacityteam.org/.

Camtasia. https://www.techsmith.com/camtasia.html.

edX. https://www.edx.org/.

FFmpeg. https://www.ffmpeg.org/.

FFmpeg filters. https://ffmpeg.org/ffmpeg-filters.html.

Galbraith, Joel D, and Steven G Spencer. 2001. "Variable Speed Playback of Digitally Recorded Lectures: Evaluating Learner Feedback." *Provo, Utah: Center for Instructional Design, Brigham Young University.*

Gan, Cheong K, and Robert W Donaldson. 1988. "Adaptive silence deletion for speech storage and voice mail applications." *Acoustics, Speech and Signal Processing, IEEE Transactions on* 36 (6): 924–927.

Gong, Yihong, and Xin Liu. 2001. "Generic text summarization using relevance measure and latent semantic analysis." In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval,* 19–25. ACM.

Guo, Philip J, Juho Kim, and Rob Rubin. 2014. "How video production affects student engagement: An empirical study of mooc videos." In *Proceedings of the first ACM conference on Learning@ scale conference,* 41–50. ACM.

He, Liwei, Elizabeth Sanocki, Anoop Gupta, and Jonathan Grudin. 1999. "Auto-summarization of audio-video presentations." In *Proceedings of the seventh ACM international conference on Multimedia (Part 1),* 489–498. ACM.

IBM Speech to Text API. https://speech-to-text-demo.mybluemix.net/.

IBM Watson API Reference. http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/speech-to-text/api/v1/.

IBM Watson Developer Cloud - Speech to Text. http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/speech-to-text.html.

Khan Academy. https://www.khanacademy.org/.

Laconia Trim. http://laconiatrimvideo.com/.

Landauer, Thomas K, Peter W Foltz, and Darrell Laham. 1998. “An introduction to latent semantic analysis.” *Discourse processes* 25 (2-3): 259–284.

Ozsoy, Makbule Gulcin, Ilyas Cicekli, and Ferda Nur Alpaslan. 2010. “Text summarization of turkish texts using latent semantic analysis.” In *Proceedings of the 23rd international conference on computational linguistics,* 869–876. Association for Computational Linguistics.

t-test. http://www.socialresearchmethods.net/.

wav format documentation. http://soundfile.sapp.org/doc/WaveFormat/.

Williams, James R. 1998. “Guidelines for the use of multimedia in instruction.” In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting,* 42:1447–1451. 20. SAGE Publications.

Yang, Li-chiung. 2003. “Duration and pauses as phrasal and boundary marking indicators in speech.” In *Proceedings of 15th ICPhS,* 1791–1794.

APPENDIX A

QUESTIONS - USER STUDY-1

Please answer the questions below. (1- Highly dissatisfied, 5- Highly satisfied)

1. I felt the video was too fast
   - A. 1
   - B. 2
   - C. 3
   - D. 4
   - E. 5
2. I felt the video was too slow
   - A. 1
   - B. 2
   - C. 3
   - D. 4
   - E. 5
3. I was able to understand the speaker and did not feel the need to rewind the video
   - A. 1
   - B. 2
   - C. 3
   - D. 4
   - E. 5
4. I had a good understanding of the heap data structure before I watched the video
   - A. 1
   - B. 2
   - C. 3
   - D. 4
   - E. 5
5. I was satisfied with the video clarity
   - A. 1
   - B. 2
   - C. 3
   - D. 4
   - E. 5
6. I was satisfied with the audio clarity
   - A. 1

B. 2

C. 3

D. 4

E. 5

7. If a node falls into array index 1, the children of the node fall into array indices

A. 2 and 3

B. 2 and 4

C. 3 and 4

D. I was not able to understand the concept

E. I was able to understand but unable to recollect

8. If a node is present in array index 7, parent of that node is present in

A. 3

B. 4

C. 2

D. I was not able to understand the concept

E. I was able to understand but unable to recollect

9. You are satisfied with the playback rate of the video and is good enough to follow. (1- Highly dissatisfied , 5- Highly satisfied)

A. 1

B. 2

C. 3

D. 4

E. 5

APPENDIX B

QUESTIONS - USER STUDY-2

1. You are
    - A. A graduate student
    - B. An undergraduate student
2. You have taken a class related to data structures
    - A. True
    - B. False
3. What is your confidence level about heap data structures (1- Low confidence, 5- High confidence)
    - A. 1
    - B. 2
    - C. 3
    - D. 4
    - E. 5
4. English is your native language
    - A. True
    - B. False
5. The tutorial deals with
    - A. Translating a heap as a tree to heap as an array structure
    - B. Translating a heap as an array structure to heap as a tree structure
    - C. I was not able to understand what the tutorial was about
6. If a node falls into array index 1, the children of the node fall into array indices
    - A. 2 and 3
    - B. 2 and 4
    - C. 3 and 4
    - D. I was not able to understand the concept
    - E. I was able to understand but unable to recollect
7. If a node is present in array index 7, parent of that node is present in
    - A. 3
    - B. 4
    - C. 2
    - D. I was not able to understand the concept
    - E. I was able to understand but unable to recollect
8. You are satisfied with the playback rate of the video and is good enough to follow. (1- Highly dissatisfied , 5- Highly satisfied)
    - A. 1
    - B. 2

C. 3

D. 4

E. 5