Automated Segmentation and Tagging of Lecture Videos

Submitted in partial fulfillment of the requirements for the degree of

Master of Technology

by

Ravi Raipuria
Roll No: 113050077

Under the guidance of:

Prof. D. B. Phatak

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Mumbai

2012
Acknowledgement

I would like to express my sincere thanks to my guide Prof. D. B. Phatak for his invaluable guidance. Without his deep insight into this domain and his valuable time for this project, it would not have been possible for me to move ahead properly.
Abstract

Recorded video lectures are used more frequently by students for E-learning purpose. Generally, duration of a lecture video is about 90 to 120 min. So, it is cumbersome for student to search a particular area of interest in an entire video lecture due to lack of explicit annotation. However, Manual annotation is very time consuming process. It is desirable to have a system that take keywords as user-input and provide related section of video lecturer. We proposed an approach for automatic tagging of video lectures based on speech transcription of its audio layer. Firstly, The Video is converted in audio file and speech transcription is generated using open source Speech Recognition Engine. We processed the speech transcription for generating index for video lecture.
Contents

1 Introduction .......................................................... 1
  1.1 Motivation ......................................................... 1

2 Indexing Technique for Lecture Videos .......................... 2
  2.1 Manual ......................................................... 2
  2.2 Tagging ......................................................... 2
  2.3 Slides .......................................................... 4
  2.4 Speech Transcription ............................................ 6
  2.5 Chain Indexing of Speech Transcription ....................... 7

3 Existing Lecture Video Repository ............................... 9

4 Problem Definition and Solution Approach ....................... 11
  4.1 Problem Statement .............................................. 11
  4.2 Our Approach .................................................. 11
  4.3 Audio Extraction .............................................. 12

5 Conclusion and Future Work .................................... 13
List of Figures

2.1 MPEG-7 Basic Elements ................................................. 3
2.2 Elements of VideoSegment Tag ......................................... 4
2.3 Elements of MediaReview ............................................... 5
2.4 Chain Index ................................................................. 8
2.5 Accuracy of Result depend upon distance $d$ [17] .................. 8
4.1 Proposed Solution Approach ............................................. 12
Chapter 1

Introduction

Nowadays, many universities provide free lecture videos for distance learning education. It is very difficult to browse within those videos for a particular topic of interest. Generally, video lecture duration ends up about 90 to 120 min. To make best use of this, an efficient content-retrieval mechanism is required for searching the keyword in lecture videos. However, the problem is not to find lecture in video archive rather than finding the proper position of desired keyword in video stream. The rapid growth of e-learning data required more efficient content-based retrieval mechanism for video lectures. The main challenge is to provide automatic indexing of lecture videos and find semantically appropriate part of the lecture video.

1.1 Motivation

There is speedy growth in the amount of generating e-learning data for distance education in recent years. Content-based retrieval comes into picture, when a user asks to retrieve specific part of the video lecture. To retrieve a desired part of video is still a very difficult and time-consuming process. So, a browsing system is required based on content-based retrieval to provide desired part of the lecture video.
Chapter 2

Indexing Technique for Lecture Videos

Indexing is the process of creating access point to facilitate retrieval of information. Lecture video indexing comprises two steps: first, divide the video into coherent segments called topics and second, segments have to specify a topic with keyword or semantic description. There are four main indexing technique for video lecture indexing.

2.1 Manual

Here, topic segmentation and annotations are done manually. This process is done in very time-consuming process. A person who is familiar with topic can do the segmentation and then associate segments with its semantic description. Accuracy of this type of tagging is obviously high, and depends upon the knowledge of a person about the topic. It perform better for small lecture videos but not practically possible for large archive.

2.2 Tagging

Tagging is a process of annotating a document with an unstructured list of keywords. Collaboration tagging can be used for generation and enhancement of video meta-data to support content based retrieval[20]. It is very unlikely to tag or bookmark a document, if it’s no use of reader. But, tagging has impact to all who will read this document. This technique needs lot of time to annotating a document and it depend on many learners who have ability to annotate.

Author of [20] proposed an approach for automated tagging of synchronized multimedia contents. Firstly, Lecture video is synchronized with corresponding desktop presentation, which is use for generating MPEG-7 metadata. MPEG-7 is an XML based markup language
which is used for annotation and description of multimedia documents. Textual information of presentation slides is used to annotate each part of the video. After that you can search a video by querying a keyword used in annotating process. In this process, MPEG-7 is used to maintain the collaborative tagging information.

```xml
<Mpeg7 xmlns=".....">
  <Description xsi:type="ContentEntityTypeDef">
    ...
    <MultimediaContent xsi:type="VideoType">
      <Video>
        <MediaInformation>
          ...
        </MediaInformation>
        <TemporalDecomposition>
          <VideoSegment>...</VideoSegment>
          <VideoSegment>...</VideoSegment>
          ...
        </TemporalDecomposition>
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

Figure 2.1: MPEG-7 Basic Elements

Figure 2.1 represents the basic element of MPEG-7. `<TemporalDecomposition>` element of MPEG-7 description schema can be used to annotate the video by storing various information. `<TemporalDecomposition>` facilitate to annotate a specific part of video rather than whole video stream. A video segment can be identified by `<VideoSegment>` tag. `<VideoSegment>` associate with `<MediaReview>` element to provide collaborative tagging information.

To integrate collaborative tagging information into MPEG-7 description schema, we need to add an extra element `<MediaReview>` with each video segment. We need to store the tagging information for each user. So, the collaborative tagging information can be stored in a tuple

```
({tagset}, username, date, [rating])
```

where a set of tags is submitted by the user with rating of a specific video segment. Each tag has date and time associated with it. tagset denote all tags submitted by distinctive user for a specific video segment. It is encoded in `<FreeTextReview>` element with coma separated value. Media Review element are shown in Figure 2.3.

MediaReview element provides all necessary information needed for collaborative tagging. Date stores the last modified date of tags for each user and Reviewer element stores all user
A system is implemented [20] for collaborative tagging in video lectures. A user interface is provided where a user can add a tag for a particular time duration of lecture videos. If you are retrieving a part of lecture video, all tags associated with this part are displayed. It facilitate similarity feature, so you can easily access similar video segment. This technique is also extended to text-document, where you can perform all the operation provided by this system.

The accuracy of this approach is high, but manual annotating of lecture videos requires lot of time. It also depend upon users, who have the knowledge of a particular domain.

2.3 Slides

Slides represent most of the information in video segments. Slides can also be use for indexing of related video lectures. For synchronizing between slides and videos, a log file is maintained during the live lecture recording which store timestamps(start and finish) of each slide[21]. The task of synchronizing desktop presentation slides with lecture video is done manually. The textual content of presentation slide is used to generate MPEG-7 meta-data information.
To integrate MPEG-7 with lecture videos meta-data, slides converted into MHTML and textual information is extracted from the slides. After eliminating all stop word that doesn’t have semantic meaning, all words are stemmed. Porter algorithm is used for word stemming. The timestamps stored in a log file are used to partition the video into smaller segments, So you have one video segment corresponding to particular slide. Now, you can extract outline and headings from HTML file and integrate it with MPEG-7 <FreeTextAnnotation> and <KeywordAnnotation> element of a video segment.

MPEG-7 is XML based markup language, so searching in MPEG-7 is just required to parse the XML file, which follows document object model. Their are mainly two problems, one is parsing the XML file is slow downs the system and second, ranking of user query result. To overcome the first issue, an index technique is used for MPEG-7 document. During search process index must contain all the information provided by MPEG-7 document. For second one, the are taking word frequency for as a ranking parameter. If a video segment has a word that has higher word frequencies, it will be put at higher level.
This technique also require lot of manual work, so not reliable. And, nobody cares about maintaining the log file at the time of presentation.

2.4 Speech Transcription

Speech can also be used as resource of index for lecture videos. State-of-art SRE is used to transforming a speech into text[15]. Now the task is to divide the text into smaller parts called topics. A transition(for e.g., pause) in a vocabulary could mark a segment boundary, based on we can divide the video into multiple segments. For each video segment an index is generated, for searching the specific video segment. The development of text segment is a very challenging task of natural language process.

After generating the text stream, some standard text algorithm are implemented to check whether it is possible to recognize the topic segment based on speech transcription[18].

- **Linear**: The linear distribution of slide transition during the presentation time is implemented. It assumes that number of topic is given for a particular lecture video.

- **Pause**: The duration of silence(pause in speech) is used as feature for detecting the segment boundary. The longest pause in a lecture video used as a segment boundaries.

- **C99, LCseg and MinCut**: The number of segment boundary is given to this algorithm and sentence length is defined as 10 words. A slide transition is assumed to be segment boundary, and detected by this algorithm.

- **SlidingWindow**: It is based on TextTiling algorithm[14]. In this, a sliding window (120 words) moves across the text stream over a certain interval(20 words) and neighboring window is compared by cosine measure. After post-processing, lowest similarity points will become the segment boundaries.

- **VoiceSeg**: In this algorithm, first text stream is passed to stemmer, removed all stop words and similar words are grouped into clusters. Now vector element is filled by the value of cluster data and build the vectors. Similarity between two Adjacent vector are calculated by cosine measure and segment boundaries are detected based on these similarity.

The result of these algorithm is shows that determining the segment boundary is nearly impossible(*First Test*). If the segment count is not given to algorithm, the result could be much worse. It needs to count the number of segment boundary to work it properly.
If the transcript is error-free, it will improve the performance of this algorithm (Second Test). Further the segmentation result could be better if we have the real segment boundary, rather than assuming pause as segment boundary. The overall result shows that detecting the segment boundary is very complex task.

### 2.5 Chain Indexing of Speech Transcription

Chaining is a process of detecting cohesive areas in lecture video. A linguistic research shows that word repetition in a speech transcription of lecture video is hint for creating thematic cohesion[19]. Text stream can contain all parts of speech. First, all stop words are removed and then it is passed to stemmer. A stemmed word in a text stream is called term[16]. All distinctive term are stored in a list $L$. If we have $n$ terms the list will have $n$ distinct terms[17].

$$L = \{T_1, T_2, T_3, \ldots, T_n\}$$

A chain is constructed for each term in list $L$. Chain contains each appearance of the term from beginning to end in text stream with timestamps. Now, the chain is divided into subparts if there is large distance $d$ between two successive appearance. Store each segment in database with term count in each segment. The whole process will work as follows:

1. Take a term $T_n$ from list $L$.
2. build segments: divide the chain in subparts, count the occurrence in text stream, store start and end time for each segment.
3. store the data in database as chain index for the lecture video.
4. take the next term $T_{n+1}$ from list $L$ and repeat go to step 2.

There may be chances of overlap of chains because they are generated for each distinct term separately. Segments have attached weight (no of occurrence) called TN, to reduce ambiguity. If a segment have higher score, it will be more relevant.

Figure 2.4 represents an example of the chain index. If you query for sorting in chain index database. It has two appearance in the table. First appearance has 10 occurrences in speech transcription and second appearance has 4 occurrences. If a term has higher occurrences, it will be given higher priority. In this case, First appearance will be given higher priority.

#### The Distance d

The distance $d$ is varied from 0.5 to 10 minutes, for finding the proper position of breaking the chain. and measure the accuracy for 10 keywords. These keyword are randomly selected from
<table>
<thead>
<tr>
<th>Term(T)</th>
<th>Start Time</th>
<th>End Time</th>
<th>TN(No Of Occurrences)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sorting</td>
<td>1600s</td>
<td>2400s</td>
<td>10</td>
</tr>
<tr>
<td>Sorting</td>
<td>2600s</td>
<td>3300s</td>
<td>4</td>
</tr>
<tr>
<td>QuickSort</td>
<td>2100s</td>
<td>2800s</td>
<td>8</td>
</tr>
<tr>
<td>MergeSort</td>
<td>1500s</td>
<td>2200s</td>
<td>5</td>
</tr>
<tr>
<td>Searching</td>
<td>100s</td>
<td>1500s</td>
<td>20</td>
</tr>
</tbody>
</table>

Figure 2.4: Chain Index

the domain lexicon. Result of this analysis is shown in Figure 2.5. It may be vary because analysis is done for only 10 keywords. It is still debatable whether this interval (2 to 8 minutes) will always provide accurate result.

Figure 2.5: Accuracy of Result depend upon distance d [17]
Chapter 3

Existing Lecture Video Repository

In this chapter, we will talk about features of existing lecture video repository. These repository compared on the basis of following parameter:

1. **Video Retrieval**: It is a process of obtaining related lecture videos from the repository. There are many repository on the web, which provides free lecture videos. They have large number of videos in their repository. If somebody want to access a particular lecture video (e.g. Algorithm), they should have some retrieval mechanism to search in the video repository. It can be done in two ways:

   - **Meta-data Based**: Meta-data is the textual data associated with the video such as title, description, comments etc. This method use these meta-data to search the video in the repository.
   
   - **Content Based**: Content of a lecture video can be retrieved from speech transcript, slide etc. These content is used as index to search a video in the repository.

2. **Video Navigation**: Navigation within a video is important aspect, if you want to retrieve a particular topic from a large duration videos. It can also be done in two ways:

   - **Key-frames**: Key-frames are the screen-shot of slides from lecture video. A index is generated using these key-frames to navigate a video for a particular slides. Presentation slides can be additionally used to improve the performance of the system.

   - **Speech Transcription**: Most of the content in lecture video is a speech. Speech transcript can be used as main resource of indexing. Speech is converted into text and then stemming operation is performed to reduce the redundancy of keywords. Now, an index is generated based on these keywords for navigate a particular topic in lecture video.
<table>
<thead>
<tr>
<th>Repository</th>
<th>Retrieval</th>
<th>Navigation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Meta-data</td>
<td>Content</td>
</tr>
<tr>
<td>CDEEP[3]</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>NPTEL[11]</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>MIT Open CourseWare [7]</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Coursera[4]</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Academic Earth[1]</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Free Video Lectures[6]</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>VideoLectures.net[12]</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>MIT Lecture Browser[13]</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 3.1: Comparison of Lecture Video Repositories

Table 3.1 shows the comparison between existing lecture video repository based on the above parameter.
Chapter 4

Problem Definition and Solution Approach

4.1 Problem Statement

Now a days, E-Learning is the main resource of distance education. Many repository provides free lecture videos on the Internet for distance education purposes. Duration of these videos around 90-120 minutes, so it is very difficult to browse a specific topic within the lecture video. To efficient browsing within lecture videos, an tagging mechanism is required for content based retrieval. Our aim is to develop a content-based retrieval tool for providing efficient browsing within lecture videos.

4.2 Our Approach

Our proposed solution provide search and navigation feature based on the content of lecture video. A lecture video contains audio part and visual part i.e. frames. Firstly, frames are extracted from lecture video. Now, we are processing these frames to obtain text slides. Optical character recognition is used to extract content from slides. Most of the slides have a title at top portion, that will used in segmentation of lecture videos. We are storing the title and description of each slides with the timestamps at which it occur in lecture video.

We are extracting the audio layer of lecture video. Audio is passed to speech recognition engine to produce text stream of whole lecture with time stamps associated with it. Text stream can also have stop words that have no use in segmentation or indexing. First, all stop words removed and then it is passed to stemmer. Stemming is done in pre-processing phase. We are using these stemmed word for producing index of lecture video. We are using open source tool to develop this system.
4.3 Audio Extraction

There are several open source multimedia tools available. Some of them are listed below:

- MPlayer [10]
- FFMpeg [5]
- AviSynth [2]
- MKVToolnix [8]
- MP4Box [9]

Each of them have its own feature, but the most versatile utility is FFMpeg[5]. Only FFMpeg provide Java wrapper for performing operation on multimedia files. This is main reason for choosing FFMpeg for converting video to audio file. It support all multimedia file formats for audio and video file. Below is the command that converts a video file into audio file.

```
ffmpeg -i "inputFile" -ar 16000 -ac 1 "outputfile"
```

Here, we are using lossless compression for better quality audio. FFMpeg used ar and ac option for defining audio frequency and audio channel respectively.
Chapter 5

Conclusion and Future Work

A Literature survey of existing technique for segmentation and tagging of lecture video is presented in Chapter 2. These technique have some drawbacks such as Technique 2.1 require lot of manual processing to do segmentation and tagging. Technique 2.2 is quite effective compared to 2.2, but it requires expert who have the knowledge of related domain. Technique 2.3 require a log file of timestamps maintained during presentation, it is not possible for all lecture videos. Technique 2.4 perform automatic segmentation and tagging, but there is debate on how to choose the time difference between two topic segments.

We have proposed an approach for automatic segmentation and tagging of lecture videos. Initially, we are using the slides (video frames) and speech for segmentation of lecture video. We are thinking of using clustering algorithm for finding out the topic segment in a lecture video. The architecture of proposed system presented in Figure 4.1, which shows the important module of our system.
Bibliography


