Automated Video Indexing

Arpan Mukherjee MT17007, IIITD
Kunal Suryavanshi MT17023, IIITD
Shubhi Tiwari MT17057, IIITD
{arpan17007, kunal17023, shubhi17057}@iiitd.ac.in

Abstract

Automatic generation of video indexes remains one of the few manual tasks related to videos. Indexing videos that are large(in terms of play duration) can be access easily if they are indexed. In this project we present a new domain-independent,training-free and automatic approach to generation of indexes of videos. We generate indexes for a video in terms of start time, end time and the main topic discussed in the video in that duration. we show our approach to be a more effective and practical than ones that used previous keyword extraction and supervised learning.

***Keywords*—** indexing, indexing video by content, video segmentation, similarity measure

1 Introduction

Video has become an integral part of multimedia computing and communication environments, with applications as varied as broadcasting, education, publishing, and military intelligence. With the explosion of internet-based learning and online courses, the use of videos has increased even more. However, the video will only become an active part of every-day computing environments when we can use it with the same facility that we currently use text.

A lot of course lecture videos from prestigious universities like Stanford, MIT and IITs are available online free for everyone to share and learn. But these videos are very long, the approx 1-2 hour in duration. Even if one wants to watch only a specific topic in the video, one has to go through the entire video to find it. This not only eats up the precious time of the student but also leads to disinterest in the course. If these long videos are indexed, it would also provide the learners, an opportunity to see the video structure beforehand and decide if the video is useful to them or not. Also, a structured view that represents the video contents in smaller chunks is less intimidating for a new learner than an entire video that has no other information but a title alone.

Thus it would be beneficial and time-saving if indexes are available for these videos just like indexes in a book. An index could specify the time duration (start time and end time) and the topic(s) discussed in that duration. Unfortunately, there does not exist any automatic tool or software that can perform this task of index generation. All indexes that do exist are manually created. Hence we present a domain-independent,training-free and automatic indexer for videos that can generate indexes for fairly large videos.

We divide our problem of video indexing into two main components. First, we need to find the time slices or video sections that talk about a single topic. Second, we need to mine the crucial topics from these video sections. We have designed a novel algorithm for determining the video sections. For topic mining from these video sections, we are using LDA and NMF.

2 Related Work

As per our survey, very less work has been done in the field of automatic video indexing. Our initial approach was to take ideas from ”Can Back-of-the-Book Indexes be Automatically Created” - Zhaohui Wuy, Zhenhui Liz, Prasenjit Mitrazy, C. Lee Gileszy, where the author discusses how indexes can automatically be generated for books using the text itself. Many online tools are also available which can also index a book, but they require the user to provide the topics explicitly. We have attempted to build a fully automatic indexer, that does not depend on any book any other text material for topic mining.

Content-Based Video Indexing and Retrieval by Stephen W. Smoliar and Hongliang Zhang was another approach which was discussed in 1994 and hence did not use many new tools like speech to text and automatic subtitles which were used in a significant amount in our project.

3 Dataset

We have collected our data set from two sources. First, we are using youtube subtitles as our dataset. We have parsed over 220 videos and extracted their subtitles. The parsing was done by using a web crawler, and subtitles were downloaded using an online tool. The dataset consisted of SRT files from 3 NPTEL video courses. The motivation behind choosing this dataset was the average length of the videos. The NPTEL videos that we chose had an average duration over 40 mins.

But we had a bottleneck that ground truth was not available in case of the videos mentioned above. Thus we chose subtitles from a course on BIG DATA from coursera.org. The reason behind selecting these course subtitles was that each video in this course was short (5-10 minutes in duration). Thus we can treat each of these videos as an individual section, and the title of the video can represent the topics that are discussed in the section. We combine five such short videos and treat it as a complete video just like the longer youtube videos. Then we run our indexer on this combined video.

4 Proposed Methodology

The following approach has been used to accomplish the requisite of the project.

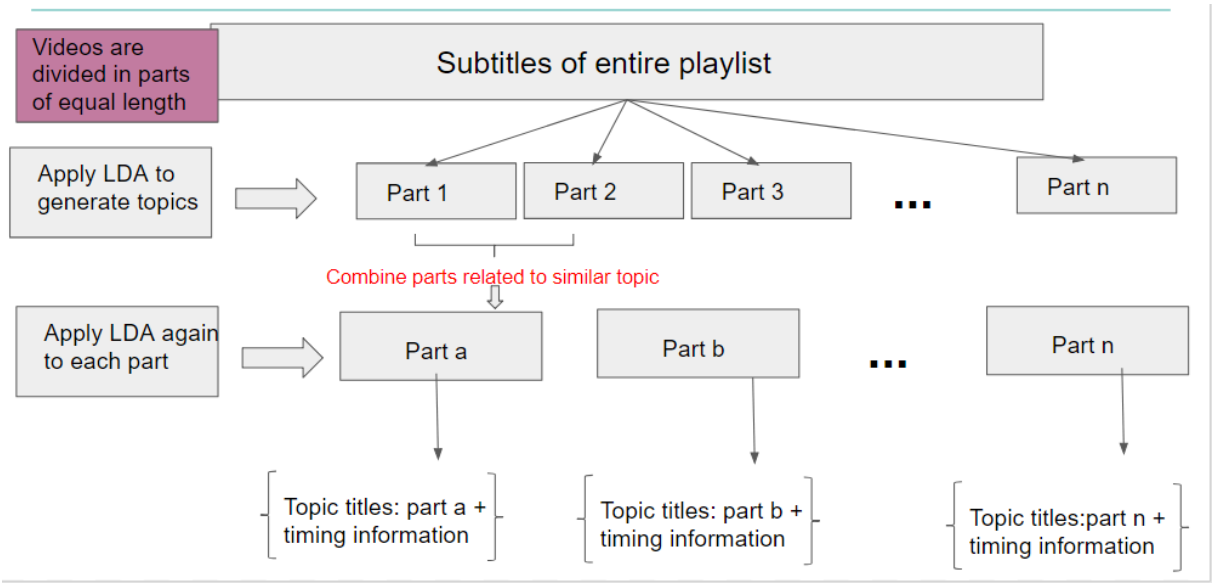

Figure 1: Tree based LDA model

4.1 Data Preprocessing

The pre-processing of the data consisted of the following steps.

- Used python module pysrt for processing of subtitles.
- Corrected the timing information of combined videos(5 videos combined in one) so that it can be treated as a single video srt file
- The basic text pre processing steps were done like lemmatization, stop words removal, stemming etc on subtitle text
- POS tagging to extract the nouns for template based model.
- We also planned to take into consideration the video name and video description as it contained a significant amount of information related to the video content. From this information, we planned to extract important nouns related to the course. But this step could not be preformed as video description was not available for all videos.

4.2 Models

We tried building two models to generate the indices in the video. Both the models were similar in terms of the pre-processing needed. The models were based on simple observation of how books and other material are indexed.

4.2.1 Template based model

This model is a simple rule-based model wherein a video would be divided into parts and put into predefined partition as per a template. The template would be made after studying the basic structure of various videos. As we focused did not focus on a single genre of videos coming up with a template was not that easy.

The basic structure of the video would be predefined in this approach, and each video segment would be fit into one of the structure. Rules were to be manually crafted that could help us identify potential topics in the video. E.g., using the introduction part of the video the basic structure and organization of the video could be determined, and corresponding rules be generated. But coming up with rules that could help us to determine the topics was not possible because every video did not have an introduction section to it. Also, another demerit of this approach is that every video had their way of knowledge representation and spread and hence generalization would not work.

The generalization could only work if the template size were too small, i.e., the number of substructures the video is being divided into is very small. This is not an optimum approach as we didn't get the basic topic information from the video. The demerit of this approach gave us a hint to make a new model using topics modeling.

4.2.2 Tree based LDA and NMF based model

We propose a novel tree-based approach to this problem of video indexing.
We first divide the subtitles in a video into segments of 20 units each. Each unit in a subtitle file is a small section of 4-5 secs and the text spoken in that time duration. The pre-processing steps were done as mentioned before. An extra step of pre-processing was done by using regex to replace many incoherent common terms. Then we perform topic mining on each of these segments. We perform topic mining with very powerful algorithms like LDA and NMF.

The tf-idf vectoriser was used as a parameter for both LDA and NMF model.The LDA and NMF model was made using the sklearn library. Number of words/topics for LDA model was 25 as it gave the most optimal answer .This would change with the size of the video. After mining topics from each segment, we merge neighbouring segments on the basis of Jaccard similarity. A threshold was defined for merging these segments and segments with similarity greater than the threshold were merged. During merge we also merge the timing values of the videos appropriately. We finally obtain the number of segments that our indexer finally divides the given video into.
To finally define the topics that can represent each of the merged segments, another iteration of LDA/NMF was used on each of the segment. This gave us top 10 topics for each of the video segments along with time slice information.

Ground Truth		Predicted (indexer output)	
Timing	**Topics**	**Timing**	**Topics**
0:0:0 - 0:2:21	What, why,sorting, arrangement, algorithm	0:0:0 - 0:2:49	Apply, applications , data, required, sorting
0:2:22 - 0:3:45	Sorting, applications,examples	0:2:49 - 0:5:32	element', 'largest', 'array', 'unsorted', 'sorting
0:3:46 - 0:11:16	Selection, sort, algorithm, array, example,code	0:5:32 - 0:8:32	okay', 'element', 'portion', 'array', 'unsorted'
		0:8:32,0:11:35	element', '17', 'portion', 'array', 'unsorted'
0:11:17 - 0:21:16	Bubble, sort, algorithm, array, example,code	0:11:35,0:14:21	'element', 'portion', 'array', 'unsorted', 'end'
		0:14:23,0:17:32	compare', 'inner', 'greater', '10', '14'
		0:17:32,0:20:20	'element', 'bubble', 'portion', 'array', 'unsorted'}
0:21:17 - 0:33:22	Quick, sort, algorithm, array, example,pseudo , code	0:20:23 -0:23:35	sort', 'quicksort', 'smaller', 'algorithm', 'problem'
		0:23:35 - 0:26:29	left', 'solution', 'solve', 'array', 'problem'
		0:26:29,0:29:50	partition', 'quicksort', 'method', 'portion', 'value'

Figure 2: Results on youtube video ”Introduction to Sorting”

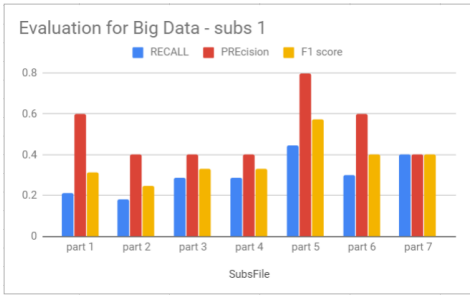

Figure 3: Precision, Recall F1 score on sub1.srt

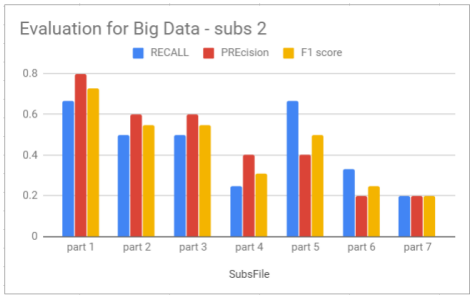

Figure 4: Precision, Recall F1 score on sub2.srt

5 Evaluation

We tried to come up many automatic evaluation schemes for calculating the recall and RMSE error.These automatic evaluations were unsuccessful as the final outcome of the video was independent of the domain. The number of segments that the model gave was not always equal to the number of topics in the ground truth. So, we had to settle down for manual recall and precision calculation. A judge was assigned for the same and hence recall for multiple files was calculated.

$$Precision = \frac{\text{number of correctly retrieved topics}}{\text{total number of retrieved topics}} \quad (1)$$

$$Recall = \frac{\text{number of correctly retrieved topics}}{\text{total number of topics in ground truth}} \quad (2)$$

$$F1score = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

The judge calculated Precision, Recall and F1 score.

6 Results

We ran our indexer on two data sets. One was course era course on Big data. For this we treated the video titles as the ground truth. sub1.srt, sub2.srt, sub3.srt and sub4.srt are

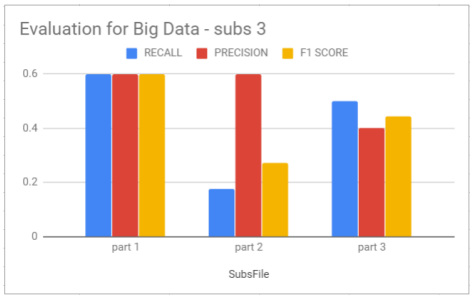

Figure 5: Precision, Recall F1 score on sub3.srt

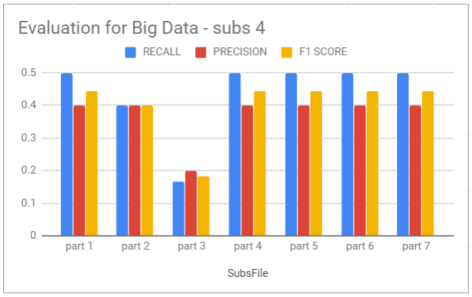

Figure 6: Precision, Recall F1 score on sub4.srt

videos consisting of 5 parts each in the ground truth. We computed precision, recall and F1 score for each 3 , 4, 5, 6. Another data set was a video we chose from you tube titled ”Introduction to Sorting” for which we did not have any automatic ground truth. A human judge saw and indexed the videos and we have compared his observations with the results of our indexer and is shown in 2. We can see that although the results have been split into multiple sections, but terms like ”bubble sort” have appeared in the indexer output when the human judge has titled the video section to be about bubble sort.

7 Conclusion and Future Work

We have attempted at building a fully automatic, domain independent indexer. The approach we suggest is novel and works quite well. It mines the required topic from videos and timing information provided is also quite accurate. But the output of the indexer needs to be refined further. A lot of extra topics are part of the result which are related to the topic discussed in the video but may not be very apt for indexing.