International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:11, No:9, 2017

Lecture Video Indexing and Retrieval Using Topic Keywords

B. J. Sandesh, Saurabha Jirgi, S. Vidya, Prakash Eljer, Gowri Srinivasa

Abstract**—**In this paper, we propose a framework to help users to search and retrieve the portions in the lecture video of their interest. This is achieved by temporally segmenting and indexing the lecture video using the topic keywords. We use transcribed text from the video and documents relevant to the video topic extracted from the web for this purpose. The keywords for indexing are found by applying the non-negative matrix factorization (NMF) topic modeling techniques on the web documents. Our proposed technique first creates indices on the transcribed documents using the topic keywords, and these are mapped to the video to find the start and end time of the portions of the video for a particular topic. This time information is stored in the index table along with the topic keyword which is used to retrieve the specific portions of the video for the query provided by the users.

Keywords**—**Video indexing and retrieval, lecture videos, content based video search, multimodal indexing.

I. Introduction

DUE to the advancement in video recording technologies and availability of these recording devices in cheaper rates, a lot of universities today record the courses and publish these videos on the web to facilitate students to access the lecture videos on their convenience and requirements. With the advancement of internet technologies, today a lot of universities started providing online courses using the videos lectures. After retrieving the videos of user interest, the user may want to watch the specific topic in the hour-long video. In this scenario, the user has to manually watch the complete video and needs to find the specific portion on topics of his/her interest. Manually searching the portions of a specific topic by users in the video is tedious and time-consuming. To retrieve such portions from the video of user interest efficiently and quickly based on the specific topic requires videos to be indexed by using the keywords. Manually annotating these lecture videos is time-consuming and biased by the keywords provided by the administrators. In order to efficiently retrieve the videos or clips, these video databases or archives need to be indexed.

Much research had been done in order to index the lecture video for efficient retrieval. Researchers have used structural analysis to segment the slide frames from the other frames by applying SVM classifier on the image histogram. Edge and discrete cosine transformation based text detector is used to retrieve the text and the false alarm are removed using stroke width transformations (SWT) based analysis. This text extracted from slides is used to index the video [2], [8]. Search engine for a lecture web cast to locate the specific topic within the lecture videos is developed by creating a metadata to search, which includes video frames containing slides and their time codes. The text in the frame is used to create the indices which are used to play the required clips directly from the web link to avoid storage on the local media [1]. A traditional content-based video retrieval technique that uses visual content in the video cannot give satisfactory results to segment video content because of the content in lecture videos is homogeneous. Further, the quality of a slide could differ withina video because of the movement of the speaker and camera. To overcome such issues and in order to enhance the user's experience of understanding the content presented in the video, researchers have used the videos showing the speaker and slides separately. The slides of the lecture could be grabbed separately using the frame grabber facility and played separately from the speaker. From such videos, textual data from the slide extracted using OCR and Automatic Speech Recognition (ASR) applied on audio tracks are used to create the metadata about the lecture in the video. Keywords are extracted using manual annotation data, and OCR and ASR are used to create the indices for efficient retrieval [6]. In order to enhance the ASR quality applied on the audio tracks in the previous work, speech corpus is created by using the lecture videos as training data and is used to generate the transcripts of the audio tracks [7].

Extracting the text from the video poses challenges and depends on the quality of the characters on the slide in the video. In order to avoid processing of slide and text extraction, indexing technique is devised by using only the transcribed text in the video. In this work, transcripts of the lecture video are clustered based on the topic word, and later in phrasing phase, these clustered topics are analysed for metaphases like 'example', 'definition', 'overview', etc. for indexing the lecture videos [5]. The quality of the indexing, using only transcripts of the audio in the video depends on the performance of automatic speech recognition software and will be confined to a single modality. In order to overcome the deficiencies of different modalities of the data in the video and to improve the quality of retrieval, multi-modal data about the lecture videos from different sources are combined for indexing purpose. In such a work, semantic annotation of the lecture videos is done by extracting the keywords from text on the video, associated lecture notes, and transcripts from the

Sandesh B. J and Gowri Srinivasa are with the Department of Computer Science and Engineering, PES Center for Pattern Recognition, PESIT Bangalore South Campus, Bangalore, India (e-mail: sandesh_bj@pes.edu, gsrinivasa@pes.edu).

Saurabha Jirgi, S. Vidya, Prakash Eljer and Gowri Srinivasa is with the Department of Computer Science and Engineering, PES Center for Pattern Recognition, PESIT Bangalore South Campus, Bangalore, India.

audio of the lecture video. A semantic network of concepts is created by correlating the words extracted to create the enhanced set of words. The lecture videos are indexed by first segmenting the video temporally and using the keywords extracted for efficient retrieval [3].

A lecture includes several topics, but it is difficult to judge their boundaries. A system is required which allows users to view an interesting part of lecture videos, by selecting identified topics from the video. If these videos are segmented into topics, the user can easily access and repeatedly view an interesting or inversely incomprehensive topic in the video. In this work, we propose an indexing system which segments the videos of continuous lecture speech based on the topics taught in the video. This requires our system to find the boundaries in the portions of the video based on the topic. Finding such boundaries using the text data available on the slide depends on the quality of the frame containing these slides in the video. Lecture videos proceed according to the topic and sub topics contained in the specific area. This topic and the sub-topics in lecture videos match with most of the documents available on the web on specific search given by the user. The person giving a lecture in the video, uses specific keywords related to topics and sub-topics while delivering the lecture. So, we propose to use the audio transcripts of the lecture extracted from the video and this is associated with the textual content available on the web to segment the transcripts into different topics. These segmented transcript documents are later temporally matched with the video to find the boundaries in the video. Using the topic keyword extracted by matching the audio transcripts and the web data, we index the lecture videos.

The proposed framework fulfils the following objectives:

1) Separates the audio channel from video and the audio is transcribed into a text file.
2) Extracts the corresponding text documents from the web for the given topic in the query and finds the topic keywords.
3) Finds the frequency and position of the topic keywords in the transcribed text file to divide the file into segments and indexes these segments using topic keywords.
4) Maps the indexed transcribed file to video to temporally segment the video and indexes the segments with the topic keywords used in indexing the transcribed file.
5) Creates the index file using the temporal information of the segmented video with corresponding indexed keywords.
6) Provides the user with querying system based on the topic of user interest and plays the relevant clips by retrieving the clips from the video using the index table.

A. Proposed System

The framework that we proposed for automatic indexing of lecture videos consists of four phases: video to text conversion, web document topic modelling, transcribed text processing and indexing and video mapping and playback system. The schematic diagram of the proposed system is shown in Fig. 1. In the first phase, videos uploaded by the administrator into lecture video archival are retrieved and transcripts from this video are extracted. In the web document topic modelling phase, text documents corresponding to the topic in the video lecture are extracted from the web, and from these text documents, topic keywords are found by applying the topic modelling using non-negative matrix factorization (NMF). In the transcribed text processing and indexing phase, the transcribed text is processed and is indexed using the topic keywords found in the previous phase. In the final phase, the indexed transcribed files from the previous phase are mapped to the video for temporally segmenting and indexing the video using topic keywords.

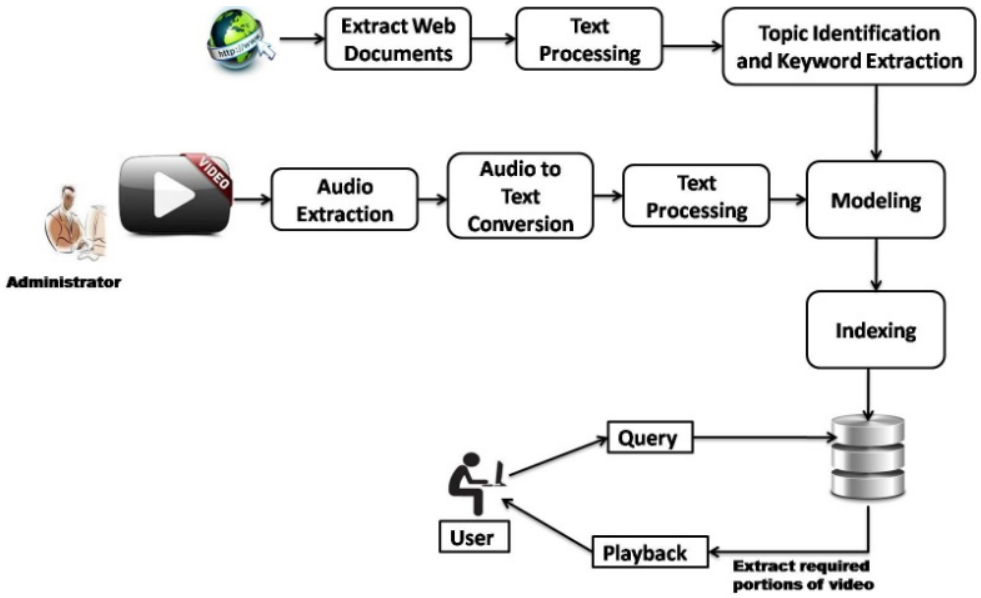

Fig. 1 Architecture of proposed lecture video indexing system

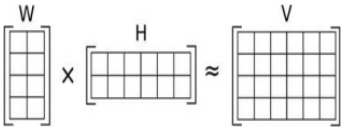

Fig. 2 Matrix V and its factors W and H

1) Video to Text Conversion

Our indexing system uses the transcripts of the speaker in the lecture video for indexing. So, to transcribe the audio in the video, we first separated audio channel from the video using audio extractor facility in VLC. Later this audio is transcribed using Dragon naturally speaking tool into a text file and is stored.

2) Web Document Topic Modelling

Statistical analysis of multivariate data could be done using non-negative Matrix Factorization (NMF). These techniques are broadly used in areas like bio informatics, image processing, audio processing and text analysis. Given a non-negative matrix V, NMF finds non-negative matrix factors W and H [4] such that

$$V \approx WH$$

The non-negative matrix V and its factors matrices are shown in Fig. 2.

We have used this algorithm to find out related topics to a given topic from a set of web documents. In this module, the keyword for searching the topic provided by the user is taken, and the related web documents from the web are extracted. These web documents are processed using NMF method to find the keywords related to the topic provided by the user. The schematic diagram for topic modeling is shown in Fig. 3. The text in the documents is tokenized as individual tokens which represent the terms in our data. A filtering is applied to remove the non-content bearing terms using stop word removal and infrequent word occurrences by considering the presence of these terms in different documents. Document term-matrix is created using n-dimensional data vectors and these vectors are placed in the columns of n x m matrix V where m is the number of documents in the dataset. Later, TF-IDF term weighing is applied to find the frequency of the terms. This is fed as an input to the NMF module to find the topic keywords. The document term matrix of 6 x 8 is shown for a set of six documents with eight terms. This matrix is factorized into two matrices by choosing the value of k as 3. Matrix and factorized matrices for this example are shown in Fig. 4. To find the topic keywords, initial factors are generated using non-negative double singular value decomposition (NNDSVD). Using these initial factors, final factors W and H are found. Using the top ranked terms in the column of the W and document membership weights in the rows of H, K topics are extracted for particular document. The value of k is chosen as smaller than n or m, so matrix W and H are smaller than V.

Fig. 3 Topic Modelling

Transcribed Text Processing and Indexing: Keywords found in the topic modeling phase is used to process the transcribed text extracted from the video. These keywords are searched by tokenizing the transcribed text and filtering it using stop words. The frequency of these keywords and the positions of their occurrence in the transcribed document are found. Using this information and these topic keywords, transcribed textual document is divided into segments, and these segments are indexed using the relevant topic keywords. Later, the adjacent segments are checked, and if two adjacent slots are tagged with the same topic, then such segments are merged. After delineating the transcribed file, the range of delineated portions is marked, which provides the percentage values for start and end time of the portions in the transcribed file. These percentage values are calculated by considering the total length of transcribed document which is approximated to the total length of the lecture videos. The length of the transcribed document is given by the total number of characters in the document and is represented as W_l. The start position of any topic p in the transcribed file is represented using the word position of the topic word in the document as W_p. Using the information about the length of the transcribed text and start position of the topic, the indexed start and end time of topic p in transcribed text file is calculated using (1). Using this equation, the percentage of a particular segment in the transcribed text file could be calculated. The representation of the percentage coverage of segments is shown in Fig. 5. Transcribed index table (TIT) for transcribed file is created using indexed start and end time along with the topic keyword.

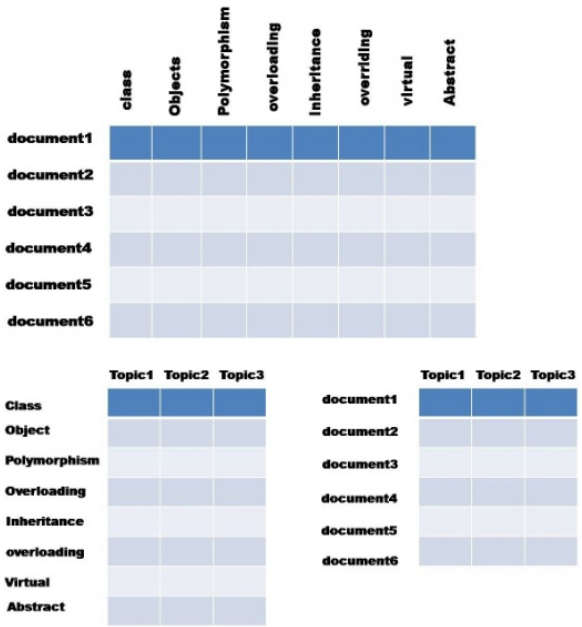

Fig. 4 Document term matrix

$$I_{st} = \frac{W_p}{W_l}, I_{et} = \frac{W_{p+1}-1}{W_l} \quad (1)$$

3) Video Mapping and Playback System

The indexed transcribed text file TIT in the previous phase is used to index the portions of videos with different topics and to create the video index table (VIT). The mapping of TIT to VIT is done using the start and end indices of topics in TIT. The video start time (V_{st}) and video end time (V_{et}) is calculated using (2). These video topic start and end times along with the topic keywords for the portions of the video are stored as the index table. In order to play the portions of the video requested by the user on his/her topic of interest in the query, we use VIT. The topic start and end times for the topic keyword in the query are searched, and using this information, the corresponding portions of the video clips are extracted and played back to the user.

$$V_{st} = I_{st} * T_v \, , \; V_{et} = I_{et} * T_v \quad (2)$$

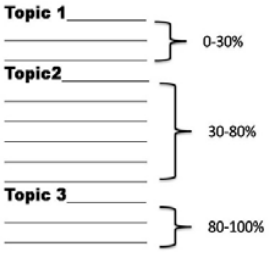

Fig. 5 Representation of transcribed text files with percentage of topics

II. Experimental Results

The proposed system for lecture video indexing is tested on the four lecture videos of different topics. These videos on topic object-oriented programming (OOP), database system management concepts (DBMS), test search with lucent (TST) and simple programming languages (SPL) are used to test the correctness of our indexing method. The details of these videos with the duration are given in the Table I. The relevant text documents from the web for these four videos are extracted from the web and topic keyword are found using NMF topic modelling. The transcripts of these videos are used to create the text files and this text file is analysed using the topic extracted. The relative position and percentage are calculated as indicated in the Section I.A.3 and index table TIT is created. This TIT is mapped to temporal information of the video and VIT is created using the equations shown in the Section I.A.4.

Using VIT the specific topic for each video is played, and the start time is noted. This start time of the portions of the clip using our algorithm is compared with the manually annotated topic clip timings, and the results are tabulated in Table II.

A. Discussion

The results in the table show the indexed time of particular topic by our algorithm and the actual start time in the video. For some topics, the indexed time is ahead of the actual start time and for other topics, the indexed time is lagging. The clips played after the actual start time of video by our algorithm is represented by positive and clips played earlier than the actual start time of the particular topic is represented as negative sign. The average differences for both the cases are calculated and found that total of six clips are played after the actual time and 11 clips are played before the actual time. The average lag time is 31.3, and the average ahead time is 25.1. The general average of both the lag and ahead time is 28.23. So, to make the correction by considering these values, we can observe that the most number of samples in our experiment has played the clips ahead of the actual time. So, we can add 25 units of time to our indexed time to play the clip at an almost exact time.

TABLE I
Details of the Dataset

Video	Duration
Object Oriented Programming (OOP)	29:01
Simple Programming Language (SPL)	35:58
Text Search Using Lucent	19:39
Database Management System Concepts	21:48

TABLE II
The Topic Segments Start Time and Hand Annotated Start Time of Video Clips

Video Name	Identified Topics	Indexed start time	actual time	start	difference
	Encapsulation	4:52			+31
	Inheritance	7:43	7:15		+28
OOP	Constructor	21:59	21:45		-14
	Interface	22:52	23:12		-20
	Abstract	25:01	24:39		+22
	Database	3:16	3:23		-7
	Model	11:59	12:38		-39
DBMS	Schema	13:54	14:10		-16
	Language	15:45	15:18		+27
	Administrator	18:31	19:20		-49
	Lucene	0:00	0:00		00
TSL	Document	1:45	1:58		-13
	Segment	6:02	6:52		-50
	Writer	9:49	9:33		+16
	Function	0:00	0:00		0
	Scope	4:56	6:08		-72
SPL	Mutable	21:25	22:10		-45
	Reference	24:43	24:16		+27
	Key	28:01	28:21		-20

III. Conclusion

In this research, we have used multi modal data from different sources to index the lecture videos. The indexing is done using the topic keywords extracted from the web by topic modelling. In order to index the sections of the video, we first indexed the transcribed video text file and mapped the same to video. Using the indexed information, we have generated the index table which is used for retrieval. We have tested our indexing methods on videos of different topics and

found the average error rate.

REFERENCES

[1] J. Adcock, M. Cooper, L. Denoue, H. Pirsiavash, and L. A. Rowe, "Talkminer: a lecture webcast search engine," in *Proceedings of the 18th ACM international conference on Multimedia*. ACM, 2010, pp. 241–250.

[2] H. S. Haojin Yang and C. Meinel, "Lecture video indexing and analysis using video ocr technology," in *Signal-Image Technology and InternetBased Systems (SITIS), 2011 Seventh International Conference on*. IEEE, 2011, pp. 54–61.

[3] A. S. Imran, L. Rahadianti, F. A. Cheikh, and S. Y. Yayilgan, "Semantic tags for lecture videos," in *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*. IEEE, 2012, pp. 117–120.

[4] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

[5] S. Repp and M. Meinel, "Semantic indexing for recorded educational lecture videos," in *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*. IEEE, 2006.

[6] H. Yang and C. Meinel, "Content based lecture video retrieval using speech and video text information," *IEEE Transactions On Learning Technologies*, vol. 7, no. 2, pp. 142–154, 2014.

[7] H. Yang, C. Oehlke, and C. Meinel, "An automated analysis and indexing framework for lecture video portal," in *International Conference on WebBased Learning*. Springer, 2012, pp. 285–294.

[8] H. Yang, M. Siebert, P. Luhne, H. Sack, and C. Meinel, "Automatic lecture video indexing using video ocr technology," in *Multimedia (ISM), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 111–116.