An Optimized e-Lecture Video Search and Indexing framework

Lakshmi Haritha Medida[1†] and Kasarapu Ramani[2††],

Research Scholar, CSE, JNTUA, Ananthapuramu, AP, India[1†]
Professor & Head, IT, Sree Vidyanikethan Engg. College, Tirupati, AP, India[2††].

Summary
The demand for e-learning through video lectures is rapidly increasing due to its diverse advantages over the traditional learning methods. This led to massive volumes of web-based lecture videos. Indexing and retrieval of a lecture video or a lecture video topic has thus proved to be an exceptionally challenging problem. Many techniques listed by literature were either visual or audio based, but not both. Since the effects of both the visual and audio components are equally important for the content-based indexing and retrieval, the current work is focused on both these components. A framework for automatic topic-based indexing and search depending on the innate content of the lecture videos is presented. The text from the slides is extracted using the proposed Merged Bounding Box (MBB) text detector. The audio component text extraction is done using Google Speech Recognition (GSR) technology. This hybrid approach generates the indexing keywords from the merged transcripts of both the video and audio component extractors. The search within the indexed documents is optimized based on the Naïve Bayes (NB) Classification and K-Means Clustering models. This optimized search retrieves results by searching only the relevant document cluster in the predefined categories and not the whole lecture video corpus. The work is carried out on the dataset generated by assigning categories to the lecture video transcripts gathered from e-learning portals. The performance of search is assessed based on the accuracy and time taken. Further the improved accuracy of the proposed indexing technique is compared with the accepted chain indexing technique.

Key words
Classification, Clustering, CNN, Indexing, Machine Learning, Text Detector, Video retrieval.

1. Introduction

The pursuit for online education is steadily increasing because to its diverse advantages over the conventional learning methods in classrooms. The students use recorded video lectures more generally for the purpose of online education. The demand for e-learning by video lectures is increasing rapidly as they lead to a very effective and versatile learning process. This demand has led to the enormous volumes of lecture video files on the web. So, it is a tedious task for a user/student to look for an exact topic or a specific video in these abundant repositories. Therefore, the urge for the most capable technique for indexing and retrieving videos within huge collections of lecture video emerged. The text or content can thus be easily searched and browsed within the video lecture repositories [1]. The learning efficiency may accordingly be improved. Using suitable techniques, the content can automatically be obtained from the lecture videos.

Text, a remarkable semantic component has frequently been studied for the retrieval of content-based information. Text present in the lecture videos, serves as a skeleton of the lecture and is significant in retrieving the contents. The textual knowledge can consequently be obtained by applying the MBB Text Detector and GSR techniques, which unpacks the video lecture contents. The present work is implemented on the textual data gathered from the lecture slides as well as the audio track [2]

The textual data is used for the goal of indexing apart from the text classification and clustering tasks. Indexing refers to the task of binding metadata with the video content to facilitate easy search and browsing. The indexing technology in particular is constructive in mining data in a large video corpus. In addition, classification enables users/students to access more appropriate information easily by performing search only in the relevant predefined categories and not in the whole corpus of videos [3]. Further clustering the categories into topics significantly reduced the search time. Text document clustering is grouping of the similar documents in a group of unstructured documents. It is the most widely used technique for generalizing huge amount of information [4]. In this paper, we proposed a system for indexing and retrieving the lecture videos based on the text classification and clustering models built on the audio and visual content.

The paper is structured as: Section 2 analyzes the efforts pertaining to the indexing and retrieval of lecture videos. Section 3 signifies the proposed Modified Topic Based Indexing in conjunction with the NB text classifier and clustering models included for the increase in effectiveness and diminishing the time taken for the retrieval of lecture videos. Section 4, equips the technicalities of the implemented search system. Section 5 assesses the projected search and indexing techniques. As a final point, Section 6 winds up the paper with a perspective on future work.

Manuscript received August 5, 2021
Manuscript revised August 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.8.12

2. Literature Review

Several techniques have been introduced to address the problem of lecture video indexing. Yang et al. [5] introduced an automatic indexing technique for lecture videos by applying OCR technology. In [6], Shuangbao Paul Wang et al. presented a novel system named "InVideo" for automatic indexing and search based on the keywords spoken in the videos and the content in the video frames. InVideo implemented the time stamped commenting and tagging features based on ML and pattern recognition in order to refine the search result accuracy.

Haojin Yang et al., [7] adopted OCR technology to develop a novel video segmenter for extracting the unique slide frames from the lecture video. The video indexer developed is based on the automated video segmentation and extracted lecture outlines. Manish Kanadje et al., proposed a system for indexing in order to assist low-vision students to more easily locate topics in the videos [8]. The lecture video indexing was assisted by instructors using within-speaker keyword spotting system. Yang H et al., in [9] presented a framework consisting of video segmentation, OCR and Automatic Speech Recognition (ASR) for the analysis and indexing of lecture videos. An architecture was designed for efficient analysis and integration of the analysis engine into the lecture video portal.

A web-based implementation for browsing the lecture video system [10] was developed by Vijaya Kumar Kamabathula et al. A fine level granularity of the video content search was provided based on the system developed in composition of ASR and search engine. Stephan Repp et al., in [11] introduced an indexing technique named as chain indexing for the easy search and navigation of content with in lecture videos. The chain indexing was based on the transcriptions generated by ASR. The user interface and the index structure were presented.

Most of the techniques mentioned in the literature focused on either of the visual or audio contents but not the both. Since the effect of both the visual and audio components are equally important for the content based indexing and retrieval, the current work is designed distributing weightage to the content from visual and audio features. This resulted in achieving enhanced results both in terms of accuracy and the time taken.

3. Modified Topic Based Indexing

This work, presented a methodology for indexing the topics and retrieving the lecture videos based on NB text classification and clustering models trained on the keywords and summary obtained from the text transcripts. These transcripts are developed from the audio and visual components of the video lectures using GSR and MBB Text Detector respectively. In this context, text transcript generation is a vital pace as the efficacy of the video indexing and retrieval depends totally on the accuracy achieved by GSR and MBB Text Detector. Further, training the text classification and clustering models in order to cluster the documents into topics within the categories defined minimizes the search time while enhancing the relevancy of the retrieved results. Fig. 1 provides the framework for the proposed topic based indexing technique.

3.1 Database Description

Initially a database is created from course videos across eight different domains. These domains include 'Aerospace', 'Computers', 'Civil', 'Data science' 'Electronics', 'Health and Safety', 'Irrigation' and 'Mechanics'. A total of 101 lecture videos of MP4 format are considered. These video lectures are taken from the courses offered by e-learning initiatives such as MIT OCW [12], Khan Academy [13] and NPTEL [14]. The camera angle and the recording style varies for each of these videos. Each lecture video has a variable length and distinctive internal structure. The basic features such as name, size, storage location, category of lecture video file etc. are initially considered for each of these lecture video files. In the later stages, the details of the extracted audio track, OCR and ASR transcripts, keywords and summary are also stored in the database. The duration of each of these lecture videos varies between 7 to 90 minutes.

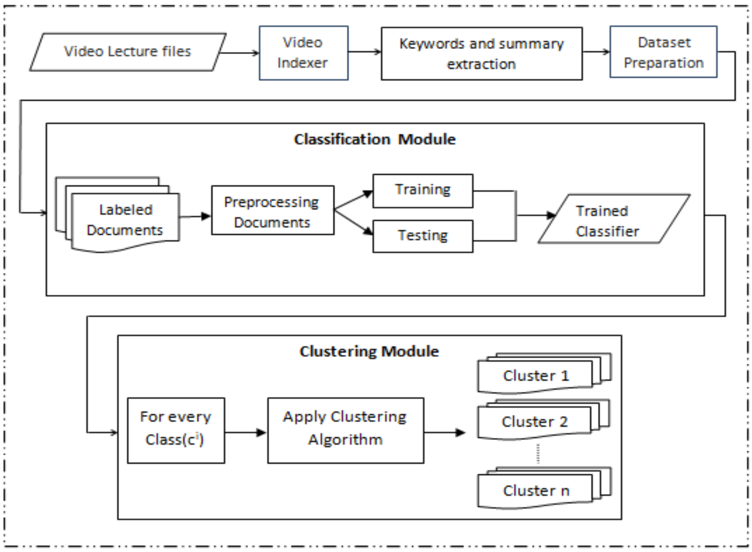

Fig. 1 Framework for topic-based indexing

Table 1: Description of the lecture video data

Domain	***No. of Videos***
Computers	44
Electronics	8
Aerospace	10
Civil	8
Mechanics	3
Irrigation	4
Data science	22
Health and Safety	2

Table 2: Partition details of lecture video data w.r.to duration

Domain	***# duration < 30 mins***	***# duration > 30 mins &< 60 mins***	***# duration > 60 mins***
Computers	26	13	5
Electronics	4	4	0
Aerospace	8	2	0
Civil	5	2	1
Mechanics	1	1	1
Irrigation	4	0	0
Data science	13	7	2
Health and Safety	1	1	0

The detailed breakdown of the database across the eight domains mentioned above is given in Table 1. Table 2 shows the partition of the database w.r.to duration of the lecture video. This variation in duration will have a significant effect on different stages of the retrieval process.

3.2 Video Indexer

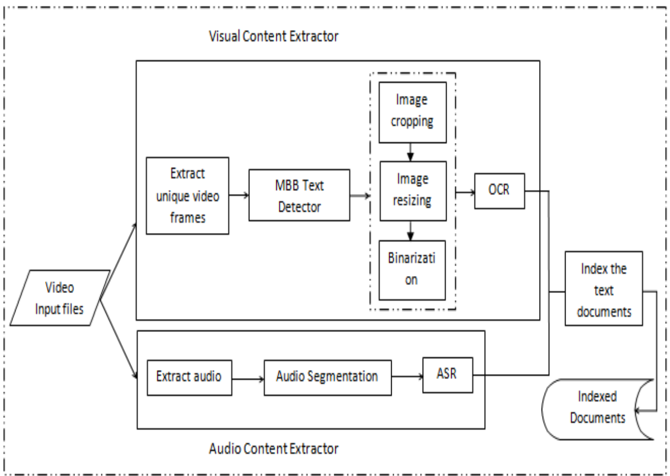

Fig. 2 Video Indexer Module

These lecture video files are fed as input to the video indexer module as shown in Fig. 2. The proposed video indexer module is composed of the Visual and Audio Content Extractor. Algorithms 1 and 2 describe the detailed functionality of each of the Visual and Audio Content Extractors respectively.

Manuscript received August 5, 2021
Manuscript revised August 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.8.12

Algorithm 1

Input

{ V } : Set of video lecture files

Output

{ T_V } : Visual Text Transcripts

Procedure

for each v_{in} ϵ { V } *do*

1. Derive each frame of the lecture video

   f_{cur} : current frame

2. Calculate the hash difference between the current frame and the previous frame if exists

   diff ← f_{cur} - f_{prev}

   diff : hash difference between the two frames

   f_{cur} : current frame

   f_{prev} : previous frame

3. *If* diff < threshold *then*

   Discard the current frame f_{cur}

end if

else

a. Extract the merged bounding boxes { bb } by means of MBB Text Detector for the text regions of the previous frame f_{prev}

b. *for each* box ϵ { bb } *do*

i. Crop the image

cropped image ← box[startY:endY, startX:endX]

startY, endY, startX, endX : coordinates of the bounding box

ii. Resize the image

resized image ← cropped image[width*3, height*3]

iii. Convert the image to binary

binary image ←COLOR_BGR2GRAY(resized image)

iv. Apply OCR on the binary image

text ← OCR (binary image)

T_V ← T_V + text

end for

end for

As discussed in algorithm 1, the bounding boxes generated for the text regions of each frame were as shown in the Fig. 3. MBB Text Detection was performed on each frame where the text detection is carried out by means of Convolutional Neural Network (CNN) and the generated text bounding boxes are merged in order to remove the overlap as in Fig. 4. This is done in order to prevent the loss of text identification

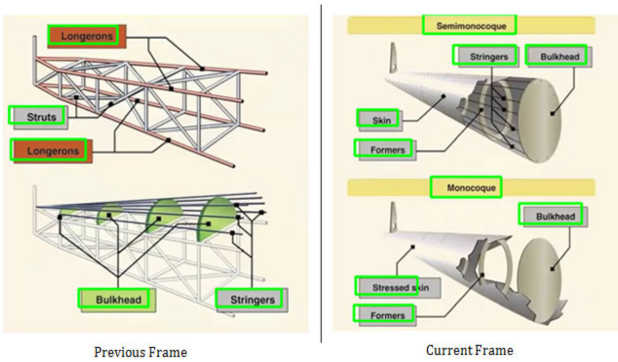

Fig. 3 Bounding boxes for different consecutive frames in a video

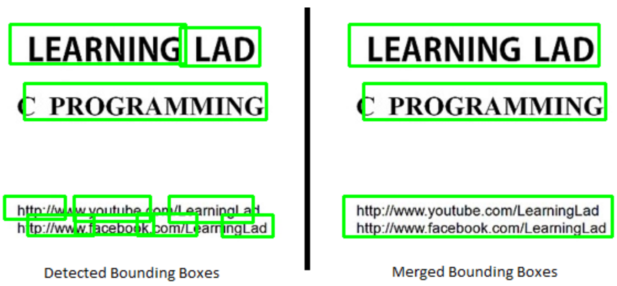

Fig. 4 Merged Bounding Boxes

Fig. 5 Workflow for the text detection

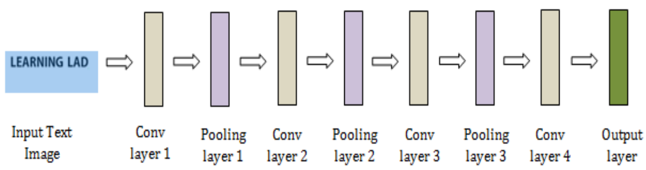

Fig. 6 Architecture of the MBB Text Detector

due to overlapped bounding boxes. The basic architecture of the MBB Text Detector is given by Fig. 6. To further achieve high detection rates of the text in the bounding boxes, before feeding the MBB Text Detected image to OCR we apply a refinement procedure on the cropped bounding box as given by Fig. 5. This binarized image is fed to OCR for character recognition.

Algorithm 2

Input

{ V } : Set of video lecture files

Output

{ T_A } : Audio Text Transcripts

Procedure

for each v_{in} ϵ { V } *do*

1. Obtain the audio track of the lecture video

   aud ← ffmpeg (v_{in})

   aud : a 16- bit 16KHz mono little endian file

2. Split the audio track into segments of 20 secs duration each

   aud ← { aud_i } where i ϵ [1, n] and n: total no. of segments

3. Generate audio text transcript using GSR

   for each i ϵ { aud_i } *do*

   trans ← transcript (i)

   T_A ← T_A + trans

   T_A :audio text transcript

   end for

end for

The indexed text documents were generated from the transcripts generated from both the Visual and Audio Component Extractors. These indexed documents are stored for a well-organized search and retrieval of the video lecture files.

3.3 Summary and Keyword Generation

Summarization is a technique that outlines the given text [15] automatically. It is done by extracting the most significant sentences present in the text. Keyword extraction is a procedure that determines the terms which carry the most significant information of the text. Summary and keyword extraction techniques therefore decrease the document's length and complexity, while maintaining the crucial characteristics of the primary textual content. In place of training the classification model on the text transcripts, we directly train the model on the extracted summaries and keywords. This is a crucial step as it slashes the training time of the classification model while maintaining the model efficiency.

The gensim library of python is made use of to accomplish the task of summary and keywords generation. The performance of the gensim library is based on the conventional "TextRank" algorithm. The generated summary and keyword attributes added to the dataset are used for training the text classification model. A category stored in the database is assigned to each instance for the classification to be carried out. This organized dataset is used to train the model which classifies the lecture video files according to the assigned categories. The dataset description is as follows:

Dataset Characteristics	Text
No. of Instances	101
No. of Attributes	3
Missing Values	0
Associated Task	Classification

Attribute Description

V_id	This number uniquely identifies each individual lecture video from the database of 101 different videos.
Category	Defines the domain classification to which each lecture belong to
Ke_summ	Represents the output summary and keywords extracted from each lecture video

3.4 Classification Module

Classification is a supervised learning method that aims to construct a predictive model by training on a dataset with the instances assigned to a class, to predict the class labels of test instances whose classes are not known. Text classification is a definite process of classifying textual files into certain categories. ML further makes the text classification job more fast and proficient [16].

Text classification in the current scenario of lecture video retrieval supports in obtaining the most appropriate results in a highly reduced time [17]. The NB text classifier is modeled on the prepared dataset. As NB is a simple classifier model based on probability, it makes prompt and independent predictions making an assumption of feature independence. NB generates better results when there is less overlap among the class variables for the reason of its assumption of the independent class variables [18]. In view of these factors NB is better suited for the dataset considered. The output of this module is a trained text classification model that is saved for the category prediction of the search query. The algorithm for the classification module is described by algorithm 3.

Manuscript received August 5, 2021
Manuscript revised August 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.8.12

Algorithm 3

Input

sum_key.csv : input data file

Output

nbc_model.sav : NB text classification model

Procedure

1. Set S to summary and C to category of the dataset
   S ← summary
   C ← category
2. Oversample the dataset in order to balance it
3. Text preprocessing by symbol and stop words removal
   S ← text_clean(S)
4. Dataset split into training and testing sets
   S_train, S_test, C_train, C_test ← traintest_split(S, C)
5. Generate a classifier pipeline
   nbm : classifier pipeline
6. Train the classifier model
   nbm.fit(S_train, C_train)
7. Calculate the accuracy of final classifier model
   accuracy(nbm.predict(S_test), C_test)
8. Save the model to a file as nbc_model.sav

3.5 Clustering Module

Document clustering achieved prominent attention as a research problem in the information retrieval field. The framework involving clustering model the corpus with a pre-determined number of topics, where each topic is taken to be a cluster. The output suggests the probability of any given document belonging to any one of the computed topics. K-Means, a divisive form of clustering is most frequently used because of its simplicity and flexibility [19]. Clustering the documents with K-Means continues until a stopping condition is met. Once the clustering model is trained, relevant information can be pulled out of the collection based on the user query. The implementation of the clustering technique is given by algorithm 4. With the clustering module trained on top of the classification module, the relevancy of the search is drastically improved along with the reduced search time.

Algorithm 4

Input

Set of text documents of a specific category

Output

Set of n-clusters

Procedure

1. Choose the number of clusters n
   n ← no. of clusters
2. Choose n objects randomly as the initial cluster centres
3. *Repeat*
   i. Assign each object to their nearest cluster by calculating the Euclidean distance measure
   ii. Compute the new mean points as the cluster centres
4. *Until*
   i. No change in cluster centres
   ii. No object changes its cluster
5. Save the cluster model as category_model.sav

4. Search System

Search process finds the documents match in a corpus. On the retrieval of matched documents, the documents are indexed and ranked in accordance to the significance and presented in that order. Fig. 7 presents the elements of the search scheme. When a query is made, the category and query of search are retrieved from the search page. The categories provided in the search form includes 'All', 'Aerospace', 'Civil', 'Computers', 'Datascience', 'Electronics', 'Health and Safety', 'Irrigation', 'Mechanics'. If the user didn’t specify any category, the trained NB text classification model is called to predict the category of search query. Based on the trained cluster model, the search query is then assigned to the nearest cluster of the category.

The text transcripts within the cluster are then checked in order to obtain the documents that match the query. The matched documents are indexed with the information stored in the database. The indexed videos are then ranked in order of their relevance, by calculating the Term Frequency Inverse Document Frequency (TFIDF) [20] and Cosine Similarity scores. The indexed videos corresponding to the search query are shown in accordance with the relevance. Algorithm 5 describes each step of the search process performed.

Algorithm 5

Input

term : query of search

cat : category of search

Output

Indexed Lecture videos that match the search query

Procedure

1. *If* cat = ‘All’ *then*
   Perform category prediction on NB classification model nbc_model.sav
   category ← nbm.predict([term])

end if
else
    Assign category to cat
        category ← cat
2. Assign the search query to the nearest cluster in the category
3. Execute a search in the cluster
4. *If* match is found *then*
    i. Retrieve the matched documents
        {docs} : matched documents list
    ii. Index the documents
    iii. *for each* f ϵ {docs}*do*
        Determine the cosine similarity score
        tfidf ← Tfidfvectorizer().fit_transform(f)
        cos ← cosine_similarity(tfidf[0:1], tfidf)
        cos_val ← cos_val + [cos]
    end for
    Display the indexed videos in the order of relevancy
end if
else
    Display "No results found"

Fig. 8 and Fig. 9 gives the snapshots of the search page developed and the indexed results obtained. The lecture video results are indexed by the indexing bar provided at the bottom of each video lecture. Each portion highlighted by the black color corresponds to the portion of the lecture video the matches the search query. The user is provided with the feasibility to directly navigate to the interested part of the video relevant to the search query.

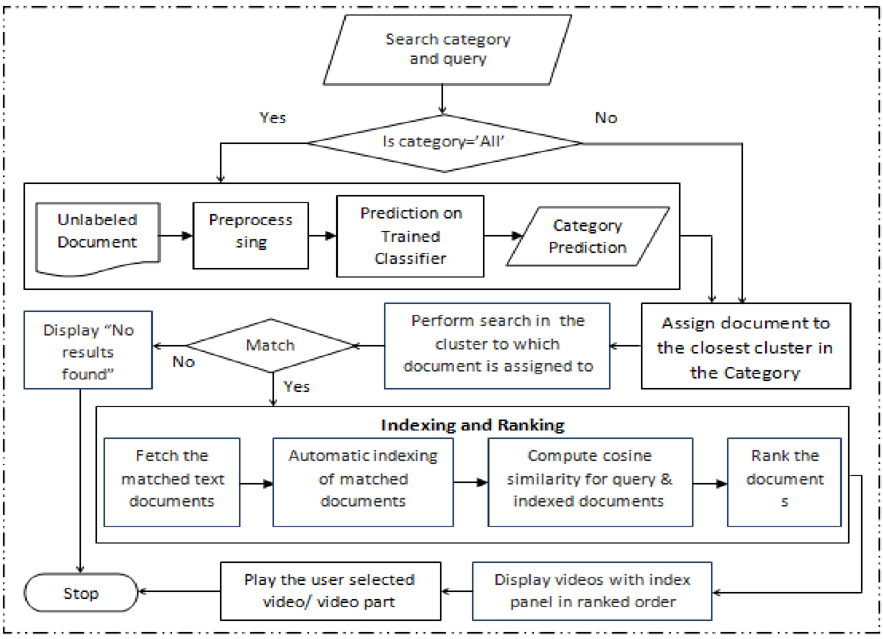

Fig. 7 Architecture of the search system

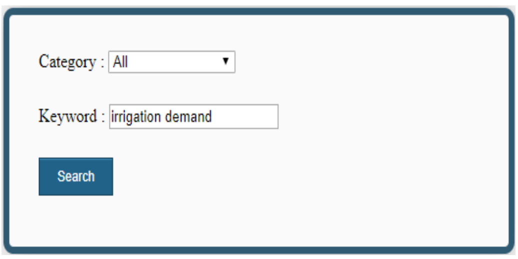

Fig. 8 Sample Search pages

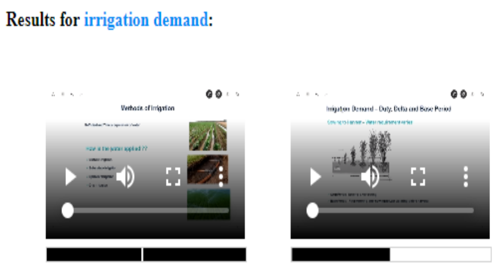

Fig. 9 Result pages with indexed videos for the search queries of fig. 8

5. Result Evaluation

5.1 Evaluation of Indexing Technique

Video indexing is provided to generate access points to the lecture videos to facilitate retrieval of the relevant information in order to improve the accessibility to the system. Accuracy metric is used for evaluating the proposed indexing technique. The current technique is compared with the accuracy achieved by the chain indexing technique proposed in [11].

	Relevant	Nonrelevant
Retrieved	Tp	Fp
Not Retrieved	Fn	Tn

The accuracy metric is given by

$$Accuracy = \frac{(Tp+Tn)}{(Tp+ Fp + Fn+Tn)} \quad (1)$$

Table 3: Comparision of proposed indexing technique with chain indexing technique w.r.to accuracy

Search Term	***Accuracy of Indexing***	
	Chain Indexing	***Proposed Indexing technique***
Breadth First Search	45.19	83.57
Pre order	12.87	78.54
Software Production	53.5	92.32
Relational Database	59.54	95.4
Java Development Tool	29.78	72

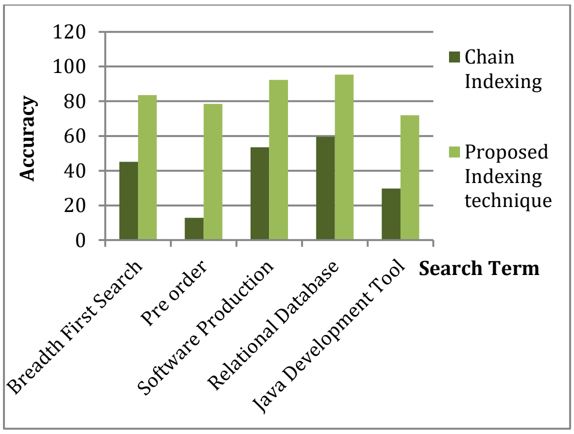

Fig. 10 Chain Indexing Vs Proposed indexing w.r.to Accuracy

Table 3 gives the improved accuracy achieved by the proposed indexing technique in comparison with the chain indexing. Fig. 10 gives the graphical representation for the same. The proposed indexing technique is based on the visual and audio contents present in the video. As both these components form a crucial part of the lecture video, understanding the topics can be made easy. However the effectiveness of the indexing technique is highly influenced by the results achieved by the proposed MBB Text Detector and GSR. As the audio component composes majority of the lecture video content, better indexing can be performed by taking into consideration the results of GSR. A limitation was observed with the proposed framework for MBB Text Detector. The framework achieved superior results in case with the visual contents based on slide presentations and other form of presentations where as minimal recognition rates were observed when the visual contents are based on blackboard teaching. The current framework needs to be modified in this direction so that the accuracy of indexing can further be improved.

5.2 Evaluation of Search Technique

The performance of the implemented search method is analyzed based on the measures of accuracy and time taken. Accuracy is distinct measure to review the efficiency of search given by Eq. 1.

The values of accuracy and time taken for diverse search terms on the Regular search, Search based on Machine Learning Classification (MLC) and Search based on the topic clustering on top of MLC model (Topics on MLC) are tabulated below in Table 4 and Table 5. Fig. 11 and Fig. 12 gives the graphical representation of the obtained results.

Table 4: Comparision of proposed topics on MLC search model with regular and MLC models w.r.to accuracy

Search Term	***Accuracy of Search Model***		
	Regular	***MLC***	***Topics on MLC***
Breadth First Search	42.53	69.90	91.26
Pre order	57.98	82.52	98.06
Software Production	83.20	95.37	97.25
Relational Database	87.88	95.19	98.09
Java Development Tool	72.69	91.26	91.26

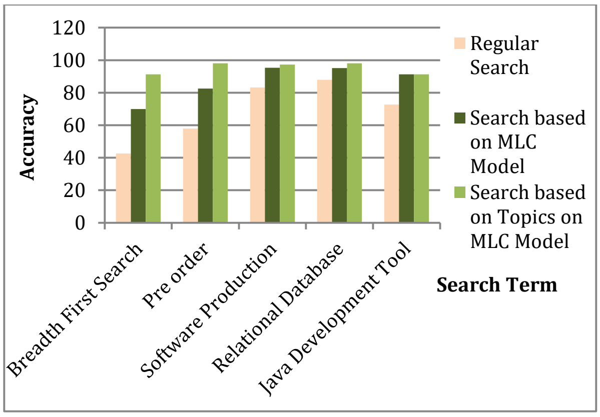

Fig. 11 Regular Vs MLC Vs Topics on MLC search models w.r.to Accuracy

Table 5: Comparision of proposed topics on MLC search model with regular and MLC models w.r.to time taken

Search Term	***Time taken (msec) by Search Model***		
	Regular	***MLC***	***Topics on MLC***
Breadth First Search	19.0007	7.49508	3.8713
Pre order	16.7625	1.67981	0.5995
Software Production	17.05768	1.94227	1.0698
Relational Database	1.5	0.57597	0.47225
Java Development Tool	2.7583	1.1291	1.05784

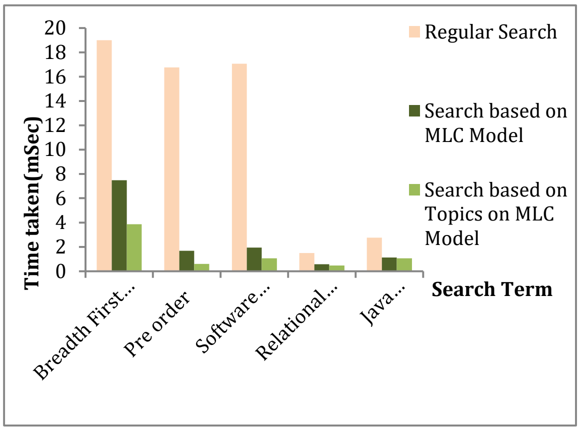

Fig. 12 Regular Vs MLC Vs Topics on MLC search models w.r.to time taken

The results illustrate that the proposed algorithm for the search based on topic clustering on the classification model achieved state of art results both in terms of accuracy and the time taken. By the implementation of ML classification model, the search is restricted to the specific category. By clustering the documents of a specific category into topics, the search is further restricted to the cluster that is closest to the search query. Therefore, with the topic clustering on the MLC model more relevant and refined results are achieved. From the figures shown above we can observe the improved relevancy of 26.324% is achieved with the proposed model in comparison with the regular search model. However the efficiency of the entire model depends wholly on the accuracy achieved by the trained classification and clustering models. With the proposed model, the time taken for search is drastically reduced to 1.42 seconds on an average as figured in Fig. 12 based on the fact that the no. of documents to search are reduced.

6. Conclusion and Future Work

In this paper, topic-based indexing and retrieval based on ML text classification and clustering of lecture videos is presented. The work was implemented on the lecture video corpus of 101 video lectures downloaded from different e-learning repositories. The visual and audio contents are extracted using the proposed MBB Text Detector and GSR. Extraction of summary and keywords was carried out on the merged text transcripts generated from both the visual and audio content extractors. Based on these transcripts the dataset for training the NB text classification model is generated. The NB classification model implemented achieved immense accuracy on prediction of search queries. The classification constrained the search to one of the eight classes predicted by the classification model and hence the search is reduced to less than 50% of the entire video corpus. A topic clustering model was trained on each category of the documents in order to further confine the search to more relevant cluster within the predicted class and therefore the search is further reduced to 15% of the video corpus. The proposed search carried out based on topic clustering on NB classification model accomplished results with high relevancy of 95.19% in an average search time of 1.42 seconds. The results of the search are presented as indexed lecture video files which specify the identified search topics within the lecture video that are more relevant to the search query. Therefore, the proposed method is vastly suitable for large corpus of video lectures. The indexed results are ranked according to the scores calculated by TFIDF and Cosine Similarity to display the videos based on the relevancy level. The extensions of the work may incorporate the implementation techniques of further improving the accuracy of indexing and to further reduce search time.

Acknowledgements

This work is supported by University Grants Commission (UGC) under Minor Research Project titled "Fast Content Based Search, Navigation and Retrieval system for E-Learning". Project Id: F.No:4-4/2015(MRP/UGC-SERO).

References

[1] Aasif Ansari and Muzammil H Mohammed, “Content based Video Retrieval Systems - Methods, Techniques, Trends and Challenges”, International Journal of Computer Applications, Volume 112, No. 7, Feb 2015.

[2] Jose M. Perea-Ortega, M. Teresa Martin-Valdivia, Arturo Montejo-Raez and L. Alfonso Urena-Lopez, “A Content-based Information Retrieval System for Video Searching”, IEEE Eighth International Symposium on Natural Language Processing, 2009.

[3] M. Ikonomakis, S. Kotsiantis and V. Tampakas, “Text Classification Using Machine Learning Techniques”, WSEAS Transactions on Computers, Volume 4, Issue 8, August 2005, pp. 966-974.

[4] V. S. Reddy, P. Kinnicutt and R. Lee, “Text Document Clustering: The Application of Cluster Analysis to Textual Document”, International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, 2016, pp. 1174-1179. doi: 10.1109/CSCI.2016.0222

[5] 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.

[6] Shuangbao Paul Wang, Xiaolong Cheng, Carolyn Maher and William Kelly,“InVideo: An Automatic Video Index and Search Engine for Large Video Collections”, in The Second International Conference on Advances in Signal, Image and Video Processing (SIGNAL), 2017, pp. 27-32.

[7] Yang Haojin, Franka Grünewald and Christoph Meinel, “Automated Extraction of Lecture Outlines from Lecture Videos - A Hybrid Solution for Lecture Video Indexing”, in 4th International Conference on Computer Supported Education (*CSEDU*), 2012, pp. 13-22.

[8] Manish Kanadje, Zachary Miller, Anurag Agarwal, Roger Gaborski, Richard Zanibbi and Stephanie Ludi, “Assisted keyword indexing for lecture videos using unsupervised keyword spotting”, Pattern Recognition Letters, Volume 71, Feb 2016, pp. 8-15.

[9] Yang H., Oehlke C. and Meinel C, “An Automated Analysis and Indexing Framework for Lecture Video Portal”, In: Popescu E., Li Q., Klamma R., Leung H., Specht M. (eds) Advances in Web-Based Learning - ICWL 2012. ICWL 2012. Lecture Notes in Computer Science, vol 7558. Springer, Berlin, Heidelberg.

[10] V.K. Kamabathula and S. Iyer, “Automated tagging to enable fine-grained browsing of lecture videos”, in IEEE International Conference on Technology of Education, 2011.

[11] Stephan Repp, Andreas Groß and Christoph Meinel, “Browsing within Lecture Videos Based on the Chain Index of Speech Transcription”, IEEE Transactions On Learning Technologies, Vol. 1, No. 3, July-September 2008, pp. 145-156.

[12] MIT opencourseware. [Online]. Available: https://ocw.mit.edu/index.htm

[13] Khan Academy. [Online]. Available: https://www.khanacademy.org/

[14] NPTEL. [Online]. Available: http://nptel.ac.in/

[15] I. Mani, M. T. Maybury, “Advances in Automatic Text Summarization,” Volume 293, Mit press, 1999.

[16] I. Rish, “An empirical study of the naive bayes classifier,” in IJCAI 2001 workshop on empirical methods in artificial intelligence, pp. 41–46, 2001.

[17] M. L. Haritha, K. Ramani,“An Optimized E-Lecture Video Retrieval based on Machine Learning Classification”, International Journal of Engineering and Advanced Technology,Volume8, Issue6,August 2019, pp.4820-4827. doi: 10.35940/ijeat.F9114.088619

[18] Zhang, Harry. “The Optimality of Naive Bayes.” FLAIRS Conference, 2004.

[19] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman and A. Y. Wu, “An efficient k-means clustering algorithm: analysis and implementation”, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, July 2002,pp. 881-892.doi: 10.1109/TPAMI.2002.1017616

[20] Bing Zhou, Yiyu Yao, “Evaluating information retrieval system performance based on user preference”, Journal Of Intelligent Information Systems, Volume 34, Issue 3, pp 227-248, June 2010.

Ms. Lakshmi Haritha Medida is currently pursuing Ph D from JNTUA, Ananthapuramu, Andhra Pradesh in the Department of Computer Science and Engineering and received her M. Tech Degree in Computer Science and Engineering from JNTUK, Andhra Pradesh in the year 2016 and pursued her B. Tech in Electronics and Communication Engineering from JNTUK, Andhra Pradesh in the year 2011. Her research interests include Information Retrieval, Data Mining and Image Processing.

Dr. K Ramani is working is as a Professor and Head in the Department of Information Technology, SVEC, A. Rangampet and received her Ph.D Degree from JNTUK, Andhra Pradesh in the year 2010. She has more than 16 years of teaching experience. Her areas of interests include Image Processing, Data mining, Computer Networks and Software Engineering.