

ADAPTIVE LEARNING ENVIRONMENTS BASED ON INTELLIGENT MANIPULATION FOR VIDEO LEARNING OBJECTS

Mohammed Atef, Shehab Gamalel-Din, Gamal Tharwat
Systems & Computers Dept, Faculty of Engineering, Al-Azhar University, Cairo, Egypt.

*Corresponding Author E-mail: mhmbw@gmail.com

Received :3 Oct. 2021 Accepted:23 Dec. 2021

ABSTRACT

e-Learning is become the education method of the future. Lecture videos are powerful and expressive learning resources that are commonly and extensively used in e-learning. However, one lecture video usually covers many topics/subtopics or even many conceptual instructional roles for a single topic. In adaptive e-Learning, the e-Learning of the future, requires small granular objects (Micro Learning Objects MLO) for more flexible and adaptive presentation of the lecture, hence, arose is the need for splitting this lecture video into many MLOs each playing a specific instructional role in the lecture.

This research is concerned with the automatic extraction of MLOs out of existing lecture videos also with automatic annotation with the appropriate metadata attributes needed for the appropriate selection of the MLO by the adaptive process. This article, however, assumes that the lecture video is based on PowerPoint (or presentation) slides that are also available to the proposed MLO extraction process.

KEYWORDS: Applied computing → Education → e-Learning → Intelligent e-Learning.

بيانات التعلم التكيفية القائمة على المعالجة الذكية لكائنات التعلم بالفيديو

محمد عاطف النبوي، شهاب جمال الدين ، جمال ثروت

قسم هندسة النظم والحاسبات، كلية الهندسة، جامعة الأزهر، القاهرة، مصر
البريد الإلكتروني للباحث الرئيسي: *[E-mail mhmbw@gmail.com](mailto:mhmbw@gmail.com)

المخلص

أصبح التعلم الإلكتروني طريقة تعليم المستقبل. تعد مقاطع فيديو المحاضرات مصادر تعليمية قوية ومعبرة والتي يتم استخدامها بشكل شائع وعلى نطاق واسع في التعلم الإلكتروني. ومع ذلك، عادةً ما يغطي مقطع فيديو محاضرة واحد العديد من الموضوعات / الموضوعات الفرعية أو حتى العديد من الأدوار التعليمية المفاهيمية لموضوع واحد. في التعلم الإلكتروني التكيفي، يتطلب التعلم الإلكتروني للمستقبل كائنات صغيرة (Micro Learning Objects MLO) لتقديم عرض أكثر مرونة وتكيفاً للمحاضرة، ومن ثم نشأت الحاجة إلى تقسيم فيديو المحاضرة إلى العديد من MLOs. كل MLO له دور تعليمي محدد في المحاضرة.

يهتم هذا البحث بالاستخراج التلقائي لـ MLOs من مقاطع فيديو المحاضرات الحالية أيضاً مع التعليق التوضيحي التلقائي بسمات البيانات الوصفية المناسبة اللازمة للاختيار المناسب لـ MLO من خلال العملية التكيفية. ومع ذلك، تفترض هذه المقالة أن فيديو المحاضرة يستند إلى شرائح PowerPoint (أو العرض التقديمي) المتوفرة أيضاً لعملية استخراج MLO المقترحة.

1. INTRODUCTION

The main purpose of adaptive e-Learning systems is to enable learners to acquire knowledge and skills more efficiently (less time), more effectively (high learning outcomes), and more pleasantly. Hence, the adaptation process involves several constituents without them no adaptation can take place, among them are the student's model, the small granular Learning Objects (LOs), and the concept ontology [1]. In this research, the focus is on the Los. Fortunately, there exist several open source learning contents and are made available for public use, such as those of Harvard University and MIT. Unfortunately, however, adaptive learning requires small granularity LOs rather than a full lecture, which are called Micro LOs (MLOs). A monolithic LO for a full lecture is not suitable for the personalization process. Small granular MLOs, on the other hand, facilitate the reassembling process into personalized lessons—a process that reorganizes these MLOs differently for the different individual students [2] according to his/her own personal learning profile (student model), which explains the need for MLOs. Clearly, the manual splitting of a large full lecture into several smaller MLOs is not an easy process and requires special skills and intensive training that an instructor is not guaranteed ready for it. Hence, comes the idea of this research—the automatic identification and splitting of those MLOs out of a monolithic full lecture. Moreover, for the automatic lesson personalization process to work properly, there must exist rich Learning Object Repositories (LORs) from which the MLOs are drawn according to specific criteria usually specified as descriptive metadata annotating the MLO. The most important of such metadata attributes are the LO's concerned concept, and the Instructional Role it serves (e.g., introduction, theory proving, example, or experiment, etc.) among others.

In summary, the objective of this research is to automate the process of identifying, extracting, and annotating those MLOs from a large full lecture, hence, splitting the lecture into its constituent MLOs covering a specific concept or a certain instructional role in a lesson. The research considers the two most commonly used media for lecturing, namely PowerPoint presentations (PPT) and Video Lectures. This article focusses on the handling of the video lectures, while the handling of the PPT lectures was thoroughly discussed in a previous article [17] and is summarized in Section 5.1 for easy follow up of the remaining of the article. More this article considered a special type of video Lectures those that are based on PPT presentation slides that are also available to the proposed MLO extraction process.

To achieve this goal, the proposed framework and algorithms are implemented using several NLP techniques, especially text and video processing techniques, such as (text cleaning, tokenization, stemming, text similarity, OpenCV and OCR), and a prototype was implemented. The implementation was proven through iterative tested implementations.

The remaining parts of this article are organized as follows. Section 2 reviews some related work, while Section 3 sheds light on the concept of learning Objects. Section 4 gives an overview on the proposed model and framework of extracting MLOs from Lecture Videos, while the implementation details of this model are presented in Section 5. Finally, Section 6 concludes.

2. Related Work

Several works have been proposed on previous years dealing with Multimedia Learning Object (MLO) Segmentation, Extraction and Annotation. These works can be categorized based on their segmentation strategy that can be classified into text-based segmentation or

motion- based segmentation. For instance, in [3], Lin used text-based segmentation, where they segmented the instructional role on the basis of associated transcribed text. They used a sliding window algorithm to detect topic boundaries by moving a sliding window of a certain size across the entire transcript by certain interval then compare the similarities between two neighboring windows of text, the places where similarities have a large variation are identified as potential topic boundaries. In [4], Shah, Shaikh, and Zimmermann employed a linguistic-based approach for automatic video segmentation using Wikipedia texts. They proposed an approach to determine segment boundaries by matching transcript file blocks and Wikipedia texts with lecture video topics. Galanopoulos et al. [5] proposed a new method for lecture video fragmentation in which only automatically generated speech transcripts of a video are exploited .they used a state-of-the-art word embedding text representation into their proposed method to segment videos in meaningful parts. In the above works on video lectures fragmentation required a transcript file of the video, some other works used different techniques like OCR (Optical Character Recognition) and ASR (Automatic Speech Recognition) to extract text from a video lecture, then use the extracted text in the segmentation process. For instance, Che, Yang, and Meinel [6] proposed a technique for segmenting Lecture videos by analyzing its supplementary synchronized PowerPoint slides using an OCR. Similarly, Yang et al. [7] used slide transition recognition, text localization, and OCR techniques in order to determine fragment boundaries. Also in [8] Yang et al, proposed an approach for automated lecture video indexing based on video OCR technology, they have implemented an algorithm for extracting lecture structure using OCR by using geometrical information and text stroke width of detected text lines. They use both segmented key frames and extracted lecture outlines for the further video indexing. Noteworthy, the above-mentioned researches used a text-based segmentation strategy for the segmentation process; however, another category of researches used a motion-based strategy for segmenting video lectures. For instance, in [9], Ma and Agam proposed an approach to segment video into various scenes by identifying the transition of frames by the analysis of color histogram of lecture video's frames. In [10], Ma et al, proposed an automatic lecture video indexing framework that compares lecture slide images with candidate video frames using Boosted deep neural networks. In [11], a supervised technique is proposed that uses visual features and transcripts to detect changes in occurrences, such as "speaker writing on the blackboard" or "slide presentation." These occurrences served as a source of fragment boundaries. On the other hand, few researches work on segmentation, extraction, and annotation for adaptive learning purposes. For instance, [12] Lucia et al, presented an approach for migrating video lectures into digital learning objects. The method detects slide transitions and extracts information from a presentation document such as (author name, title, date of creation) to obtain slide images, fill in the metadata of the learning object, and extract the table of contents from the presentation. Imran and Cheikh [13] proposed a framework for the development of multi-media learning objects. They presented this framework to integrate with LMSs (Learning Management System) for the creation, storage, distribution, and evaluation of automatically extracted learning objects from digital media. They also extracted relevant basic information from the media such as (keywords, title, and type).

3. LEARNING BJECTS

In e-Learning, learning materials consist of multimedia learning objects referred to as learning objects (LOs). A Learning Object (LO) is a self-contained piece of learning contents. Those Learning objects are drawn from repositories (LOR) that are specified using standard metadata formats, such as SCORM [14] or IEEE LOM [15]. Reusability is a major advantage of using learning objects, especially MLOs that are designed to present a small piece of information regarding certain pedagogical aspects of a concept, e.g., an experiment, a theory proofing, an example, an application, etc. MLOs must be designed independently of a

specific lesson, although can be extracted from a full lecture, therefore, the same MLO can be shared among different lessons and contexts as long as it fits the required aspect in the new context. However, for automatic selection by the adaptive processes, it is required that the MLOs be properly and accurate. **Table 1.** MLO's Metadata. its attributes [16]. Examples of Some of those attributes that are adopted in this research are instructional role, format, and concept, as shown in Table 1.

Attributes	Domain of Values
Instructional Role	{Introduction, Summary, Conclusion, Explanation, Experiment, Example- Exercise, etc.}
Format	{Figure, Table, Text, Graph, Image, Video Clip, Audio Clip}.
Content Type	{Concrete, Abstract}
Depth type	{General overview, In-depth}
Concept	The Concept discussed by this LO.

4. THE PROPOSED MODEL

The objective of this research is to provide aids for the instructors in building the adaptive knowledgebase. Instructors are used to design full lectures explaining a topic, which is usually done in one of the two forms, namely, videos and/or PowerPoint representations. The instructor does his/her lecture design according to his/her own preferences. Of course, it would be much unusual task to him/her to think in pieces instead of thinking holistically. Therefore, this research takes a full video lecture and tries to split it automatically into its constituent components of MLOs each of which is concerned about a single instructional role of the topic, e.g., explanation, example, experiment, ...etc. Identifying an MLO should also be complemented by annotating it with its descriptive metadata attributes to ease the adaptive selection process. The annotated MLOs are saved in the appropriate LOR. Fig. 1 shows an example of extracted video MLOs from a given video lecture, the input is a lecture video explaining a whole lesson about Data structure and Algorithms as described by the lecture Agenda as shown in Fig. 2. The Agenda slide is one of the slides of the associated PPT lecture (and of course is one of the frames of the lecture video. Each extracted MLO represents an atomic part of the video lecture (few frames) that represents a specific instructional role of the lecture. These extracted annotated MLOs can be later used to create other video lectures that conform each student's learning model, e.g., MLOs can be added or removed from the newly composed video lecture according to the background knowledge of each individual student.

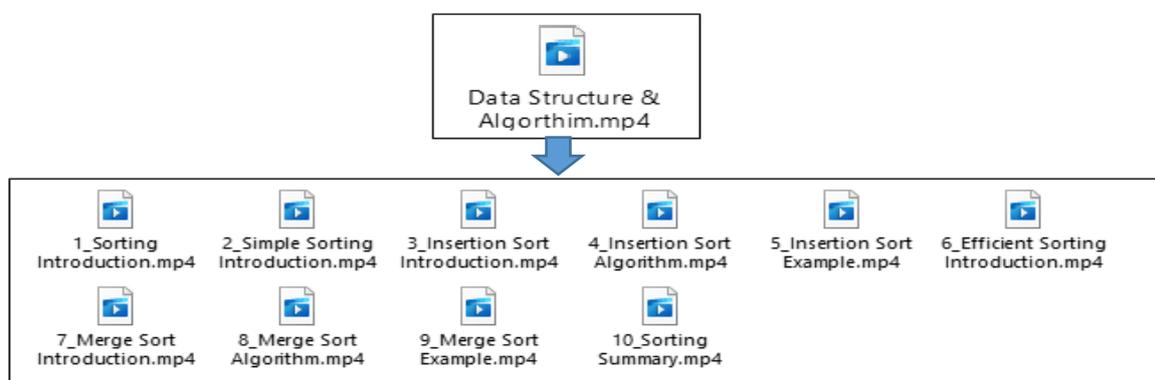


Figure 1. An example of extracted MLOs by Proposed Micro LO Extractor.

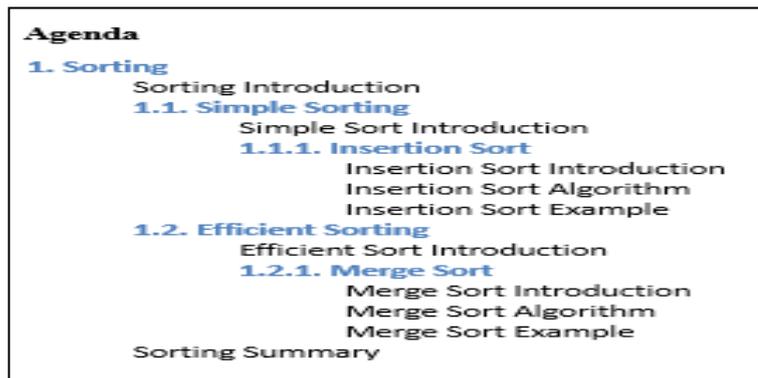


Figure 2. An example Agenda

Fig. 3 draws the proposed framework of the MLO extractor demonstrating its different components. The framework consists of four main components. The first component is the PowerPoint Learning Object (PPT LO) Extractor (1). It takes a PowerPoint Presentation as an input to extract its different PPT MLOs constituents [17]. The second component is the Video Preprocessing component (2). This component prepares the video lecture into a suitable format for matching with the PPT slides by eliminating all consecutive duplicate frames. It then applies OCR to those frames to extract their key contents. Both extracted PPT MLOs and extracted video key frames are then fed to the third component, the Video Learning Object Extractor (3), to extract the different video lecture MLOs.

The Video Learning Object Extractor matches the PPT MLOs produced by the first component with the key frames (annotated with their timestamps) to identify those frames handling the slides of the PPT MLO and, hence, composes the Video MLO.

The fourth and final component is the Metadata Annotator (4). This component is responsible for annotating the extracted LOs with the appropriate metadata, such as the Concept, the Instructional Role, the Media type, and the Language. Those extracted MLOs are then stored in the MLO repository along with their metadata information represented in XML format (Extensible Markup Language). The details of the implementation of this framework and the used algorithms are described in more details in the next section.

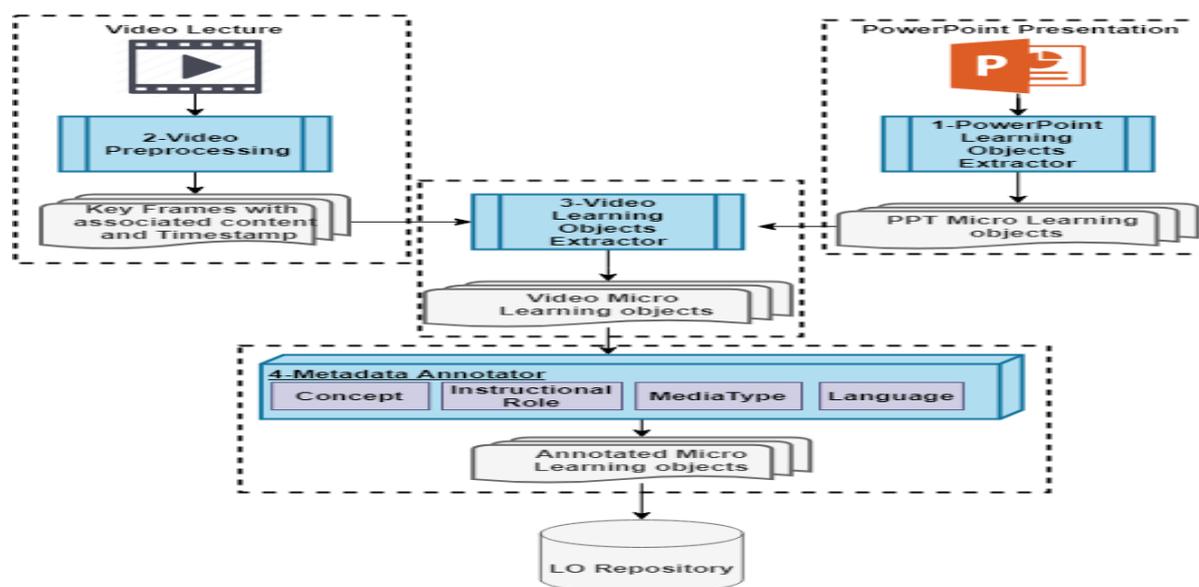


Figure 3. High-level Framework Architecture of the Proposed Micro LO Extractor out of a Video Lecture based on Presentation slides.

5. IMPLEMENTATION

This section presents in more details the proposed process of extracting and annotating MLOs from video lectures given an associated PowerPoint presentation. In this research, a prototype was implemented by python using several libraries as indicated in the description below. The following sections describe in more details the four processes of extracting the Video MLOs as depicted by the proposed framework of Fig. 3 above.

5.1. Extracting LOs from PowerPoint Presentations

It is observed that most presentations tend to follow a basic structure in spite of the contents and formats. A typical presentation, especially a lecture presentation, starts with a title slide, then an outline/overview slide, followed by a number of content explanation and separator slides. The Agenda slide of a presentation (as shown in Figure 2 above) usually summarizes different concepts and LOs that are covered by the other content slides. Figure 4 depicts the proposed PPT MLO extraction process. The first step is to identify the Agenda slide (1). All the consecutive steps are guided by the Agenda. The title of the first few slides of the presentation are parsed for a synonym of the word “Agenda”, such as "Topics", "outlines", "Road Map", "menu", ...etc. If not found in the title, the top part of the body of the slide is parsed for the same synonyms. Once identified, the agenda is parsed (2) and the discussed concepts and LOs are then identified by applying Regular expression rules as shown in Figure 5. Items matching these rules are identified as "Concepts", while the other items are identified as "Learning Objects". These rules are applied recursively on each identified “Concept” to determine its level (Concept or sub-concept or sub-sub-concept). Figure 6 depicts a parse tree for a sample agenda of Figure 2, where the parsed tree nodes represent the identified concepts and the leaves represent Learning Objects. The Identified concepts and LOs are then mapped to the other remaining slides (3) as shown in Figure 7. All Learning objects with matched slides are then fed to the PPT LO Extractor (4) to extract the micro LOs as shown in figure 8.

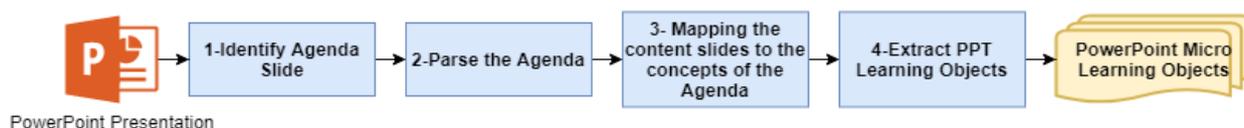


Figure 4. Process of Extracting MLOs from a PowerPoint Presentation.

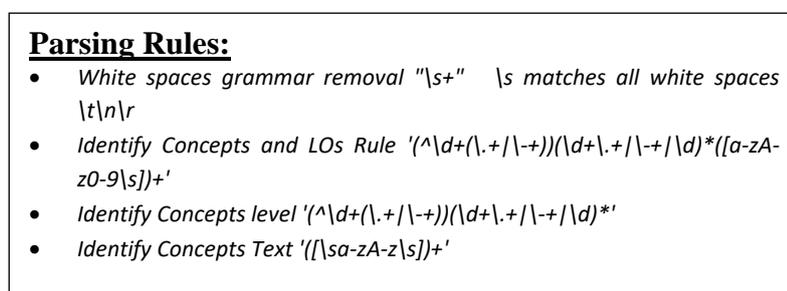


Figure 5. Regular Expressions Used for Parsing the Agenda.

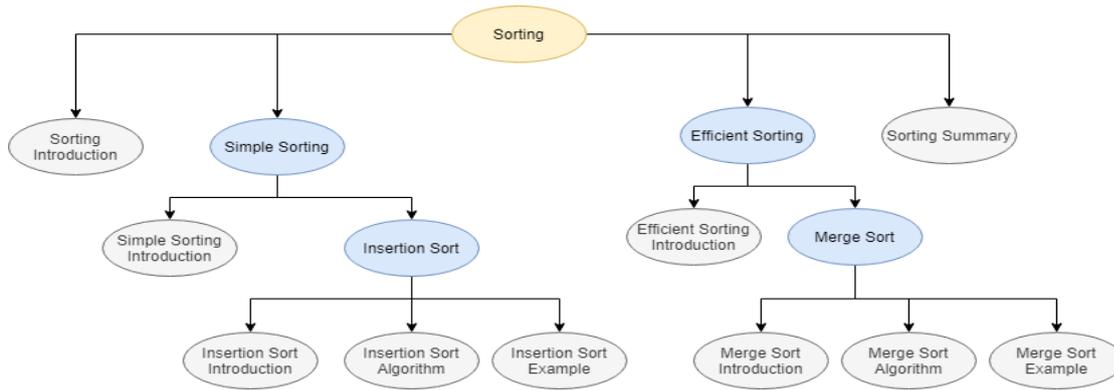


Figure 6. The Parse Tree for a Sample Agenda.

```
#iterate over all identified agenda concepts to find its associated slide
FOR concept IN agendaConcepts:
    #loop over all presentationSlides to find best slide match
    FOR currentSlide IN presentationSlides:
        IF currentSlide['Title'] EQUALS concept['Title']:
            concept['matchedSlideNumber']=currentSlide['Number']
            break
#iterate over all identified agenda LOs to find its associated slide
FOR LO IN agendaLOs:
    #loop ove presentationSlides starting from LO's Concept slide number to find best slide match
    FOR slideIndex IN range(LO['concept']['matchedSlideNumber'],len(presentationSlides)):
        IF presentationSlides[slideIndex]['Title'] EQUALS LO['Title']:
            LO['matchedSlideNumber']=presentationSlides[slideIndex]['Number']
            break
```

Figure 7. The Algorithm of Searching for Agenda concepts and LOs slides.

```
#iterate over all LOs to extract #each LO slides from presentation Slides
FOR i IN range(0,len(agendaLOs)):
    CurrentLO=agendaLOs[i]
    nextLO=agendaLOs[i+1]
    CurrenLOSlides=[]
    startSlideNumber=CurrentLO['matchedSlideNumber']
    endSlideNumber=nextLO['matchedSlideNumber']
    #extract LOs slides by iterate over presentation Slides
    #starting from current LO Start slide and next LO start slide
    FOR j IN range(startSlideNumber,endSlideNumber):
        CurrenLOSlides.append(presentationSlides[j])
    #assign extracted Slides to Current LO
    CurrentLO['Slides']=CurrenLOSlides
```

Figure 8. The Algorithm for Extracting Micro LOs.

5.2. Extracting Learning Objects from video lectures

This section presents an overview of the approach used for extracting Micro LOs from a lecture video with an associated PowerPoint file. This process covers the second and third steps of Figure 3 above—namely, the video preprocessing and the video MLO extractor.

5.2.1. Lecture video frame preprocessing

The lecture video preprocessing phase consists of three components as shown in Figure 9. The first component, the Video Frames Extractor (1) extracts the video frames using Open CV library [18].

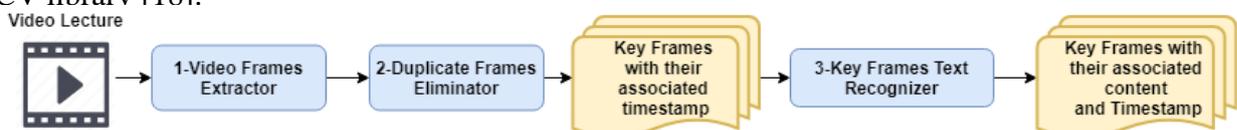


Figure 9. Preprocessing Video Frames.

Similar slides in lecture video usually last for few seconds to minutes, so, only one frame per second (fps) is considered to minimize the count of extracted video frames. Accordingly, duplicate consecutive frames are eliminated by the second component, the Duplicate Frames Eliminator (2), by computing frame-to-frame Similarity [19]; if the Similarity value is greater than a predefined threshold value, then the second frame is eliminated. The similarity is calculated using Structural Similarity Index [19] as shown in equation (1). The parameters include the (x, y) location of the N * N window in each image, the mean of the pixel intensities in the x and y direction, the variance of intensities in the x and y direction, along with the covariance. The output value can vary between -1 and 1, where 1 indicates perfect similarity. For more detailed explanation, please refer to [19].

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1) + (\sigma_x^2 + \sigma_y^2 + c_2)} \quad (1)$$

Where μ_x : the average of x, μ_y : the average of y
 σ_x^2 : the variance of x, σ_y^2 : the variance of y
 $2\sigma_{xy}$: the covariance of x and y
 $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$
 L = the dynamic range of pixel values
 $k_1 = 0.01$ and $k_2 = 0.03$

In the third component, Key Frames Text Recognizer (3), a Google Tesseract OCR [20] is applied to all key video frames to extract the text of each frame. In this implementation, we have used the python Google Tesseract OCR library. The output of this step will be used as input to the next phase of mapping and identifying the video MLOs, as discussed in the following section.

5.2.2. The Mapping and Segmentation Process

To associate each PPT MLO slides with its corresponding video segment, the temporal relationship between a slide video frame and PPTLOs slides must be established. This is accomplished by matching the slide contents from PowerPoint slides with the content of the extracted key frames from the lecture video. Based on timing information of each slide, slide LOs can be mapped with video segments, thus generating LOs from the lecture video. Figure 10 shows the process of mapping PPTLOs to lecture video key frames. Each of the components of this process can be described as follows:

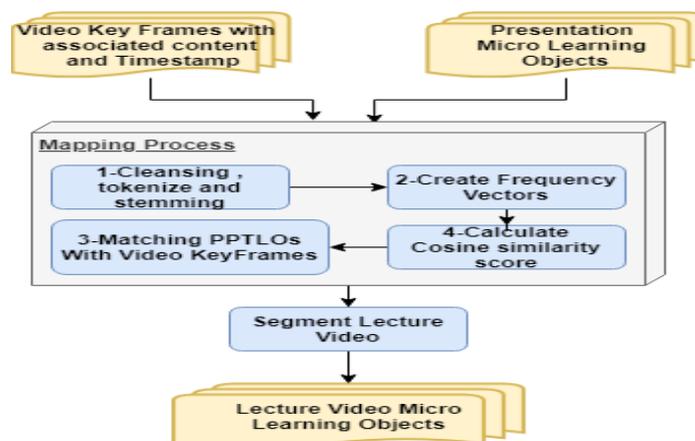


Figure 10. The Mapping and Segmentation Process.

1- Text cleansing, tokenizing, and stemming

All the textual content obtained from the video preprocessing step (OCR) and PPTLOs slides content are fed into the NLP Process for linguistic processing. After tokenizing the text, extraneous text cleansing takes place for removing all tokens consisting of single characters, numbers alone, or special characters only. A spellchecker is then applied on each valid token to validate and correct any spelling mistake via accepting the top suggestion of the spellchecker. The term stemmer is then applied. The resulted output of this step is a set of tokens representing the stemming of all words of the slide text.

2- Creating frequency vectors

In order to relate the PPT MLOs slides to their corresponding frames in the video lecture, Similarities measure is applied. Measured are the similarities between the frequency of occurrence of each term in the text of a slide and those that are extracted from the video frames content. Therefore, a dictionary of word stem frequencies is constructed for each line of text for each key frame and is represented by a vector of frequency counts. These vectors are called Key frames vectors. See table 2 Vectors representation of video key frames.

Table 2. Vectors representation of video key frames

	Term#1	Term #2	Term #m
Key Frame # 1	F_{ft11}	F_{ft12}	F_{ft1m}
Key Frame # 2	F_{ft21}	F_{ft22}		F_{ft2m}
Key Frame # n	F_{ftn1}	F_{ftn2}		F_{ftnm}

Where F: represent frequency value, f: represent Key Frame and t: represent term.

Similarly, a dictionary of word stem frequencies is constructed for each PPTLOs slide content. This is again represented as a vector of frequency counts, as shown in table 3. These vectors are called PPTLOs content vectors.

Table 3. Vectors representation of PPTLO slides content.

	Term#1	Term #2	Term #m
Slide# 1	F_{st11}	F_{st12}		F_{st1m}
Slide # 2	F_{st21}	F_{st22}		F_{st2m}
Slide # n	F_{stn1}	F_{stn2}		F_{stnm}

Where F: represent frequency value, s: represent slide and t: represent term

3- Calculating the Cosine Similarity Indexes

The Cosine Similarity index, as shown in equation 2, is often used to measure document similarity in text analysis [21, 22, and 23]. The Cosine Similarity value is bound by a constrained range between 0 and 1, where the closer the value to 1, the higher the similarity is, and vice versa. Therefore, the Cosine Similarity vector spaces for both the content of the remaining key video frames and those of the content in the PPTLO’s slides are then calculated.

$$Cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}} \quad (2)$$

Where a_i and b_i are components of vector a and b respectively

4- Matching PPTLOs with video key frames

After calculating Cosine similarity, PPTLOs Slides are matched with video frames using highest similarity value between each LO slides and video Frames. Finally, video LOs are generated by applying segmentation on input video using matched frames time stamps. Figure 11 show the proposed algorithm for matching PPTLOs with video frames and segment video LOs.

```
listExtractedVideoLOs=[]
#iterate over all extarcted PPTLOs to find Video LO which explains it
FOR PPTTLO IN listPPTLOs:
    #prepare PPTLO first slide Content
    PPTLO['FirstSlide']['Content']=PreProcess(PPTLO['FirstSlide']['Content'])
    #search for best match video frame
    maxSimilarity=0
    selectedvideoframe=None
    FOR videoframe IN listVideoFrames:
        #prepare videoframe Content
        videoframe['Content']=PreProcess(videoframe['Content'])
        #calculate Cosine Similarity between PPTLO first slide Content and videoframe content
        similarity=calculateCosineSimilarity(PPTLO['FirstSlide']['Content'],videoframe['Content'])
        if(similarity>maxSimilarity):
            maxSimilarity=similarity
            selectedvideoframe=videoframe
    #create Video LO associate to current PPTLO
    videoLO['Title']= PPTTLO['title']
    videoLO['startTimeStamp']= selectedvideoframe['TimeStamp']
    #append video LO to
    listExtractedVideoLOs.append(videoLO)
#iterate over lstExtractedVideoLOs to segment video lecture
start_time=0
end_time=0
FOR i IN range (0,len(listExtractedVideoLOs)):
    currentVideoLO=listExtractedVideoLOs[i]
    nextVideoLO=listExtractedVideoLOs[i+1]
    start_time=(currentVideoLO['startTimeStamp'] -1000) /1000
    end_time=(nextVideoLO['startTimeStamp'] - 1000) /1000
    VideoLOFilename= currentVideoLO['title'] +".mp4"
    #segment video using start_time and end_time
    ffmpeg_extract_subclip(VideoLOFilename, start_time, end_time)
```

Figure 11. An Algorithm for matching PPTLOs with video frames and segment video LOs.

5.3. Meta-data Annotation

After extracting the LOs, they are then annotated with the appropriate metadata, especially the concerned concept and the instructional role. The parent-concept relationship of the parsed agenda is used for identifying the concerned concept. On the other hand, instructional role is identified via searching the slide titles for keywords indicating the known instructional roles, like “example”, “experiment,” “application”, “theory proof,” etc. General titles are considered having “explanation” role. Figure 12 shows a sample LO metadata as extracted by the Annotator.

```
<?xml version="1.0"?>
<LO>
  <Concept>Sorting</Concept>
  <Language>English</Language>
  <InstructionalRole>Introduction</InstructionalRole>
  <MediaType>Video</MediaType>
</LO>
```

Figure 12. A Sample LO Metadata Extracted by the Annotator.

6. CONCLUSION

Lecture Videos are the most commonly used multimedia in higher education e- Learning. Extracting Micro LOs from these learning contents is becoming necessary for personalized/adaptive learning. This article proposed a novel model and a framework for identifying micro LOs from video lectures using NLP, text and video processing techniques.

More work is planned for future research. In this stage of research, video lectures are manipulated assuming that they are centered on associated PPT slides that the instructor uses all over the lecture in his/her explanation. For future research, we plan to relax this condition and investigate how to manipulate a video lecture independently of an associated PPT.

REFERENCES

1. S. A. Gamalel-Din, "An Intelligent Etutor-Student Adaptive Interaction Framework," in Proceedings of the 13th International Conference on Interacción Persona-Ordenador, 2012, doi: 10.1145/2379636.2379648.
2. R. Al-Otaibi and S. Gamalel-Din, "Intelligent querying for adaptive course preparation and delivery in e-learning," Proc. 8th IASTED Int. Conf. Web-Based Educ. WBE 2010, pp. 12–21, 2010, doi: 10.2316/p.2010.688-026.
3. M. Lin, "for Lecture Videos : A Linguistics-Based Approach," Int. J., vol. 1, no. June, pp. 27–45, 2005.
4. R. R. Shah, Y. Yu, A. D. Shaikh, and R. Zimmermann, "TRACE: Linguistic-Based Approach for Automatic Lecture Video Segmentation Leveraging Wikipedia Texts," Proc. - 2015 IEEE Int. Symp. Multimedia, ISM 2015, pp. 217–220, 2016, doi: 10.1109/ISM.2015.18.
5. Galanopoulos, Damianos & Mezaris, Vasileios. "Temporal Lecture Video Fragmentation Using Word Embeddings:" 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8–11, 2019, Proceedings, Part II. 10.1007/978-3-030-05716-9_21.
6. X. Che, H. Yang, and C. Meinel, "Lecture video segmentation by automatically analyzing the synchronized slides," MM 2013 - Proc. 2013 ACM Multimed. Conf., pp. 345–348, 2013, doi: 10.1145/2502081.2508115.
7. H. Yang, M. Siebert, P. Lühne, H. Sack, and C. Meinel, "Automatic lecture video indexing using video OCR technology," Proc. - 2011 IEEE Int. Multimedia, ISM 2011, no. December 2011, pp. 111–116, 2011, doi: 10.1109/ISM.2011.26.
8. H. Yang, M. Siebert, P. Luhne, H. Sack and C. Meinel, "Lecture Video Indexing and Analysis Using Video OCR Technology," 2011 Seventh International Conference on Signal Image Technology & Internet-Based Systems, 2011, pp. 54–61, doi: 10.1109/SITIS.2011.20.
9. D. Ma and G. Agam, "Lecture video segmentation and indexing," Doc. Recognit. Retr. XIX, vol. 8297, p. 82970V, 2012, doi: 10.1117/12.910594.
10. D. Ma, X. Zhang, X. Ouyang, and G. Agam, "Lecture video indexing using boosted margin maximizing neural networks," Proc. - 16th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2017, vol. 2017-Decem, pp. 221–227, 2017, doi: 10.1109/ICMLA.2017.0-155.
11. C. Bhatt et al., "Multi-factor segmentation for topic visualization and recommendation: The MUST-VIS system," MM 2013 - Proc. 2013 ACM Multimed. Conf., pp. 365–368, 2013, doi: 10.1145/2502081.2508120.
12. A. De Lucia, R. Francese, M. Giordano, I. Passero, and G. Tortora, "Migrating legacy video lectures to multimedia learning objects," ICEIS 2006 - 8th Int. Conf. Enterp. Inf. Syst. Proc., vol. SAIC, pp. 51–58, 2006.
13. A. S. Imran and F. A. Cheikh, "Multimedia learning objects framework for e-learning," 2012 Int. Conf. E-Learning E-Technologies Educ. ICEEE 2012, pp. 105–109, 2012, doi: 10.1109/ICeLeTE.2012.6333417.

14. Bohl, J. Scheuhase, R. Sengler and U. Winand, "The sharable content object reference model (SCORM) - a critical review," International Conference on Computers in Education, 2002. Proceedings., 2002, pp. 950-951 vol.2, doi: 10.1109/CIE.2002.1186122
15. IEEE Standard for Learning Object Metadata," in IEEE Std 1484.12.1-2002 , vol., no., pp.1-40, 6 Sept. 2002, doi: 10.1109/IEEESTD.2002.94128
16. H. S. Al-Khalifa and H. C. Davis, "The evolution of metadata from standards to semantics in e-learning applications," Proc. Seventeenth ACM Conf. Hypertext Hypermedia, HT'06, vol. 2006, no. February 2006, pp. 69–72, 2006, doi: 10.1145/1149941.1149956.
17. Mohammed Atef, Shehab Gamalel-Din, Gamal Tharwat, "Automatic Identification and Extraction of Adaptive Learning Objects from Traditional Courseware" AL-AZHAR ENGINEERING FIFTEENTH INTERNATIONAL CONFERENCE 2021.
18. G. Bradski, "The openCV library," Dr. Dobb's J. Softw. Tools, vol. 25, Jan. 2000.
19. Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image Quality Assessment: from Error Visibility to Structural Similarity. IEEE Transactions on Image Processing 13, 4 (2004), 600–612.
20. R. Smith, An Overview of the Tesseract OCR Engine, vol. 2. 2007.
21. D. Gunawan, C. A. Sembiring, and M. A. Budiman, "The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents," J. Phys. Conf. Ser., vol. 978, no. 1, 2018
22. L. Zahrotun, "Comparison Jaccard similarity, Cosine Similarity and Combined Both of the Data Clustering With Shared Nearest Neighbor Method," Comput. Eng. Appl. J., vol. 5, no. 1, pp. 11–18, 2016
23. P. Sitikhu, K. Pahi, P. Thapa, and S. Shakya, "A comparison of semantic similarity methods for maximum human interpretability," arXiv, 2019.