Proc. 25th Int. Conf. on Multimedia Modeling (MMM 2019), Springer LNCS vol. 11296. Author's accepted version.
The final publication is available at Springer via https://doi.org/10.1007/978-3-030-05716-9_21

# Temporal Lecture Video Fragmentation using Word Embeddings

Damianos Galanopoulos and Vasileios Mezaris

Information Technologies Institute/CERTH
6th Km. Charilaou - Thermi Road, Thermi-Thessaloniki
{dgalanop,bmezaris}@iti.gr

**Abstract.** In this work the problem of temporal video lecture fragmentation in meaningful parts is addressed. The visual content of lecture video can not be effectively used for this task due to its extremely homogeneous content. A new method for lecture video fragmentation in which only automatically generated speech transcripts of a video are exploited, is proposed. Contrary to previously proposed works that employ visual, audio and textual features and use time-consuming supervised methods which require annotated training data, we present a method that analyses the transcripts' text with the help of word embeddings that are generated from pre-trained state-of-the-art neural networks. Furthermore, we address a major problem of video lecture fragmentation research, which is the lack of large-scale datasets for evaluation, by presenting a new artificially-generated dataset of synthetic video lecture transcripts that we make publicly available. Experimental comparisons document the merit of the proposed approach.

**Keywords:** Lecture Video Fragmentation, Word Embeddings, Video Segmentation

## 1 Introduction

As multimedia based e-learning systems and online video-lecture databases grow rapidly, accessing and searching lecture video content becomes an important and challenging task. A key challenge is enabling fine-grained access to the lectures, i.e. accessing the video fragments that satisfy the needs of the user, rather than entire lectures. This brings up the problem of lecture video fragmentation, i.e. how to segment video lectures in logical and meaningful parts in order to enable easy access. Video lecture fragmentation differs from the classic video segmentation approaches, since in lecture videos the changes in visual content are usually scarce and are not necessarily associated with semantic transitions in the videos.

A typical lecture video includes a speaker in front of a blackboard or a projector display, in which presentation slides are projected while the speaker comments on them. In most cases the camera is static and focuses only on the speaker and possibly also the slides. Camera movements are scarce and mainly smooth. In most cases the only noteworthy visual changes are the slide transitions. As a

Proc. 25th Int. Conf. on Multimedia Modeling (MMM 2019), Springer LNCS vol. 11296. Author's accepted version.
The final publication is available at Springer via https://doi.org/10.1007/978-3-030-05716-9_21

general observation, the main subject of a lecture video may often be relatively broad (e.g. “Neural Networks”, “Information Retrieval” etc.), but during the lecture a lot of different sub-topics are usually analyzed. It is vital to find an efficient and fast way to fragment lecture videos in parts, in a way that each part corresponds to a different sub-topic that can be indexed and searched efficiently.

Figure 1 illustrates an example of a typical lecture video. The visual content is quite unchanged during the entire video and cannot be associated with the sub-topics that are analyzed in this video. The corresponding slides, if available, could provide hints about the sub-topics; but, in many cases slides are not provided or are not used at all in a lecture. In contrast, speech transcripts can always be easily generated from an ASR system and they contain the key information for lecture video fragmentation, due to the detailed information that they convey.

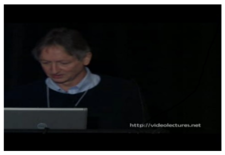
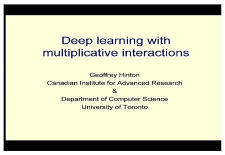
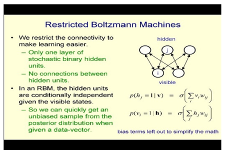
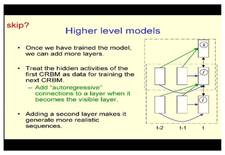
| Video Frames | Presentation slides | Speech transcripts |
|---|---|---|
|  | <br>Deep learning with multiplicative interactions<br>Geoffrey Hinton<br>Canadian Institute for Advanced Research<br>&<br>Department of Computer Science<br>University of Toronto | [...]<br>so today i'm going to spend the first ten minutes also give you background about the learning and for many of you know this stuff but i want to give the general background and then i'll show you one recent example where the ideology of deep learning worked perfectly [...] |
| | <br>Restricted Boltzmann Machines<br>• We restrict the connectivity to make learning easier.<br>– Only one layer of stochastic binary hidden units.<br>– No connections between hidden units.<br>• In an RBM, the hidden units are conditionally independent given the visible states.<br>– So we can quickly get an unbiased sample from the posterior distribution when given a data-vector.<br> <br>$p(h_j = 1 \mid \mathbf{v}) = \sigma\left(\sum_i v_i w_{ij}\right)$<br>$p(v_i = 1 \mid \mathbf{h}) = \sigma\left(\sum_j h_j w_{ij}\right)$<br>bias terms left out to simplify the math | [...]<br>so restricted boltzmann machines consists of two layers you're like units binary stochastic units and there's no connections between hidden units and also no connection between visible units<br>[...] |
| | <br>skip?<br>Higher level models<br>• Once we have trained the model, we can add more layers.<br>• Treat the hidden activities of the first CRBM as data for training the next CRBM.<br>– Add “autoregressive” connections to a layer when it becomes the visible layer.<br>• Adding a second layer makes it generate more realistic sequences.<br> | [...]<br>these models are so you can learn one model united states for the hidden units you treat those data once the data you can i put in autoregressive connections and at another hidden in the the and if you do that you'll get the best and graham taylor shown that if you're doing this for modeling metadata adding<br>[...] |

Fig. 1: Example of a typical lecture video where video frames, slides and the corresponding speech transcripts are illustrated.

The majority of state-of-the-art methods in video lecture fragmentation, e.g. [4] [14], utilize various video modalities, i.e. visual, audio, SRTs and information based on the presentation slides. However, previous research [10] [14] on lecture video fragmentation has shown that the performance of fragmentation methods that use the textual information extracted from lecture videos exceeds the corresponding performance of visual-based methods, when annotated lecture videos that could be used for training a supervised classifier are absent. In most cases, the visual part of a video segment is not associated with the semantic content that this segment deals with. For example, in the illustrated segments of Figure 1, it is impossible to determine subject transitions from the frame changes, while

Proc. 25th Int. Conf. on Multimedia Modeling (MMM 2019), Springer LNCS vol. 11296. Author's accepted version.
The final publication is available at Springer via https://doi.org/10.1007/978-3-030-05716-9_21

the information from transcripts and video slides could be used for video fragmentation. This is the reason why many previous works as well as the proposed approach exploit the spoken content of lecture video.

The major challenges in video lecture fragmentation are i) the consistency of the visual content (in most cases a static video scene with a speaker in front of a blackboard or a projector screen), which makes almost impossible to use the visual content for quality video fragmentation [14], ii) the lack of a proper evaluation dataset, due to manual annotation being time consuming, and the exact location of the fragment boundaries being often a matter of subjective assessment.

In this work a new video lecture fragmentation method, using only textual information of a video lecture derived from its transcripts, is proposed. State-of-the-art techniques from the ad-hoc video search and text analysis fields are utilized. Furthermore, neural networks with pre-trained word embeddings are utilized for textual representation. Our method is cost-effective, since it is only based on textual information. Furthermore, a large synthetically-generated dataset of video lectures, created by following an approach inspired from the document segmentation domain, is utilized for performance evaluation. The dataset is specifically designed for the temporal video lecture fragmentation problem and is made publicly available. The key contributions of the proposed work are:

- Inspired from state-of-the-art works on ad-hoc video search, new approaches to text analysis for cue extraction are examined.
- Word embeddings instead of traditional bag-of-words approaches are utilized. Taking into consideration the previous literature, this is the first work that exploits word embeddings on the lecture video fragmentation problem.
- A large-scale dataset of artificially-generated lectures, which is made available online, is created, and used for experimentation.

## 2 Related Work

Several works have been proposed in previous years dealing with lecture video related issues, such as lecture video indexing, retrieval, recommendation and segmentation. In [11] an automated lecture video indexing system is proposed. Boosted deep convolution neural networks are used to correlate lecture slide images with candidate video frames. In [18] slide-based video segmentation combined with OCR and ASR analysis are used for lecture video indexing and video search. Multi-modal language models are proposed in [5] for lecture video retrieval. The co-occurrence of words in the spoken content and the video slides are modeled by latent variable models for efficient multi-modal lecture video retrieval. Recommendation systems for educational videos have also been proposed. In [1] the *Videopedia* system is designed to recommend educational videos using topic extraction from video transcripts as well as from video metadata.

In earlier years a few approaches attempted to address the problem of temporal lecture video fragmentation. Those were based on audio-visual features combined with linguistic analysis methods [13]. [10] address the problem by exploiting textual features. The performance of different types of natural language processing methods are evaluated and their performance is compared. More recently, in [15] a method based on visual and textual analysis is presented. Lecture videos with known fragment boundaries are utilized in order to train SVMs using color histograms of video frames. Moreover, textual cues from the presentation slides and video transcripts are extracted. However, this approach in order to train SVMs requires already annotated (with ground-truth fragmentation information) lecture videos, which are scarce. In [4] a solution which segments lecture video by analyzing its supplementary synchronized slides using an OCR system is presented. Similarly, [17] uses slide transition recognition, text localization and OCR techniques in order to determine fragment boundaries. In [2] a supervised method using visual features along with transcripts is proposed. The authors of [2] trained SVMs in order to find events on a lecture video, e.g. “speaker writing on the blackboard” or “slide presentation”. Fragment boundaries were derived from these events. In [14] a method which utilizes Wikipedia articles is presented. Transcript blocks and Wikipedia text are matched w.r.t the topics that a lecture video examines. Additionally, similar to [15] color histograms of the visual content are used for training SVMs.

As the majority of the proposed methods for lecture video segmentation exploits textual information that is extracted from some modality (audio or visual), either exclusively or in combination with visual information, the text-based part of the task could also be approached as a text segmentation problem. Text segmentation is the problem of dividing documents in such way that each part is a self-contained piece of text dealing with a different sub-topic. In [7] *GRAPHSEG* is presented. This is an unsupervised graph based text segmentation method that exploits word embeddings and the semantic relatedness of text parts to construct a semantic relatedness graph, where each node represents a sentence and each edge is the semantic relatedness between two nodes. Then the maximal cliques of the graph determine the text segments. In contrast to previous text segmentation works, in [9] the task is addressed as a supervised learning problem. Using a large labeled dataset from Wikipedia corpus, an hierarchy model of two LSTN based sub-networks is trained. The first network calculates the text representation, while the second network estimates the segmentation boundaries.

The majority of previous works on lecture video fragmentation require the use of either supplementary materials of the lecture videos (e.g. sideshows), or annotated datasets for supervised training. Furthermore, the evaluation is performed in small datasets whose size ranges from 3 to 20 manually-annotated lecture videos. The method proposed in this work overcomes these problems by exploiting only one audio modality of the lecture video, and requiring no lecture-specific training data. Moreover, the introduction of a large-scale generated lecture video dataset enables the reliable evaluation of the proposed method and its comparison with previous approaches.

Proc. 25th Int. Conf. on Multimedia Modeling (MMM 2019), Springer LNCS vol. 11296. Author's accepted version.
The final publication is available at Springer via https://doi.org/10.1007/978-3-030-05716-9_21

## 3 Proposed Method

We propose a lecture fragmentation method that relies on the transcripts (SRT files) of a video lecture. Video transcripts can be easily generated by any off-the-shelve automatic speech recognition system (ASR); research on speech recognition is not the subject of this work, and therefore will not be further examined. Figure 2 illustrates the pipeline of the proposed method.

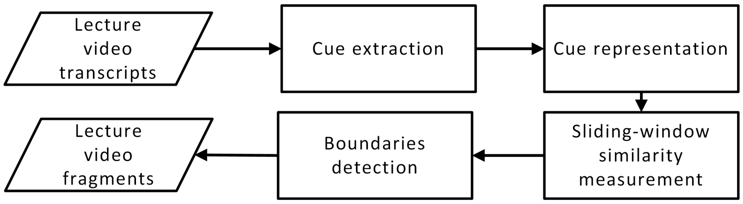


Fig. 2: Block diagram of the proposed temporal lecture video fragmentation method.

A transcript is a sequence of text parts, each one being followed by the start and end time of the corresponding spoken content in the video's audio track. To briefly explain our pipeline, the textual information derived from a transcript is utilized to extract meaningful textual cues, which are phrases or terms that the original text contains. These cues are characteristic of the original text; they capture very concisely the essence and the meaning of that text. The transcript text is used as input to our method, which outputs a set of time boundaries of the video fragments. Transcript textual parts are processed in order to extract meaningful textual cues. Two different methods for cue extraction are examined. The first method is a state-of-the-art work [14] in video lecture segmentation that uses Noun Phrases as cues. The second one is based on the textual analysis and textual decomposition component of an ad-hoc video search system [12]. Then, these cues are vectorized in a way that each textual part is represented as a single vector. Again, two different approaches are examined for transforming the extracted cues in a vector space. Finally, to fragment each lecture video, a sliding-window-based method is used in order to detect time boundaries. These boundaries define the final set of temporal video fragments.

### 3.1 Text processing and cue extraction

Standard Natural Language Processing (NLP) techniques are used in order to process the SRTs text. Text cleaning methods, such as stop-word removal, punctuation and tag cleaning are applied, followed by text lowercase conversion, in order to reduce vocabulary size. Consequently, the Stanford POS tagger [16] is used for part of speech tag extraction and the Stanford Named Entity Recognizer (NER) [6] for named entity extraction (e.g. names, organizations etc).

These tags are used to find cue phases and words that can encapsulate the information of a text part. Two different approaches are examined to this end. The first approach is the method of [14], based on which Noun Phrases (NP) are extracted from the available text. A "noun phrase" is basically a noun, plus all the words that surround and modify the noun, such as adjectives, relative clauses and prepositional phrases. The motivation behind choosing to examine this method is that in [10] the performance of several different textual features was examined, and it was shown that NP performance is better than that of other textual feature extraction methods. The second approach we examine is inspired from the query analysis and decomposition method of an ad-hoc video search (AVS) system [12]. Specifically, in [12] task-specific NLP rules are used in order to extract textual cues from a text part. For example, "Noun-Verb-Noun" sequences are searched for in the text. Such a triad can encapsulate more information than one word by itself. Both the obove approaches produce a set of words or phrases $C = [c_1, c_2, \ldots, c_t]$, where $t$ is the number of extracted cues in a textual part, which characterizes this particular part.

### 3.2 Cue representation

To represent the extracted cues in a vector space, two different representations are adopted.

First, a Bag-of-words approach with an N-gram language model, which uses the extracted cues as sequences of the model. For a specific part of text, the tf-idf weighting of the cues $C$ extracted form this part of text is calculated, to produce a vector $\mathbf{V}^C_{BoW} = [v_{c_1}, v_{c_2}, \ldots, v_{c_d}] \in \mathbb{R}^d$, where $d$ is the total number of distinct cues in the whole transcript, i.e. the dictionary of the language model.

As a cue representation alternative, Word2Vec is utilized, a state-of-the-art neural-network-based word embedding method that transforms words into a semantic vector space. Word2Vec represents every word $w_i$ of a phrase or other piece of text as a continuous vector $\mathbf{V}^{w_i}_{word2vec} = [v_1, v_2, \ldots, v_n]$ in a low dimensional space $\mathbb{R}^n$, which captures lexical and semantic properties of words. As global representation of a text part, $\mathbf{V}^C_{word2vec}$, the *average word vector* approach is followed, which averages the vectors of each word of each cue that has been extracted from this text part.

Each one of the aforementioned approaches results in a vector that represents a specific part of text, making the comparison of text parts easy.

### 3.3 Video fragmentation

To find meaningful fragments in a video transcript we follow a method similar to TextTiling [8], as it was described in [14], using textual sliding windows and measuring the similarity of neighbor windows.

A sliding window ($W_i$) of $N$ words moves across the entire text of a transcript with a certain step of $N/6$ words. On each step the similarity between two neighboring windows ($W_i$, $W_{i+1}$) is calculated. For each sliding window we follow the cue extraction process which is described above and each window is

Proc. 25th Int. Conf. on Multimedia Modeling (MMM 2019), Springer LNCS vol. 11296. Author's accepted version.
The final publication is available at Springer via https://doi.org/10.1007/978-3-030-05716-9_21

represented as a set of cues, $C_i$ and $C_{i+1}$ respectively. For each window a vector $\mathbf{V}^C$ is calculated using one of the two approaches described in the cue representation subsection above. Finally, the cosine similarity is utilized to calculate the similarity between two neighbor windows.

Following the similarity calculation between adjacent windows across the entire transcript, an one-dimensional signal $y = f(x)$, where $x$ represents time and the $y$ represents two neighboring windows similarity score, is produced. Subsequently, the valleys and peaks (local minima and maxima) of the signal are detected. The deepest valleys are assigned as candidates for segment boundaries. The depth of a valley is calculated based on the distances from the peaks on both sides of the valley. Let $val$ be the value of the signal in a local minimum, $peak_1$ the value of the signal in the closest peak on the left and $peak_2$ the value in the closest peak on the right. The depth of a valley $depth_{val}$ is calculated as: $depth_{val} = (peak_1 - val) + (peak_2 - val)$. $depth_{val}$ indicates how big the change in this particular time interval is. We make the assumption that when $depth_{val}$ is high, the semantic content of the windows on the two sides of the local minimum is highly dissimilar and therefore this time point is assigned as a fragment boundary.

Then, a fixed number of $k$ valleys with the largest $depth_{val}$ can be selected as the boundaries of the fragments. As an alternative, valleys with $depth_{val}$ larger than a threshold,

$$Thr = m \cdot (\mu - \sigma) \tag{1}$$

where $\mu$ is the mean of the signal's values in all local minima, $\sigma$ is the standard deviation and $m$ a multiplier, are selected as the actual fragment boundaries.

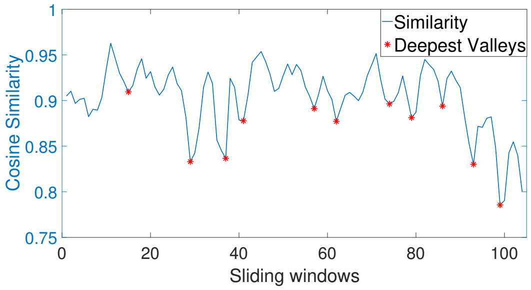


Fig. 3: An illustration of the similarities between neighboring windows of a synthetic video lecture, which is split in 12 fragments.

In Figure 3 a sample of the fragmentation procedure results are presented. The curve represents the similarity between two neighboring windows, while the dots indicate the selected valleys with the largest depth value, which are the extracted fragment boundaries.

# 4 Experimental Results

## 4.1 Dataset

An important problem in the development and evaluation of video lecture fragmentation methods is the lack of annotated datasets, due to the difficultly and the time-consuming nature of manual annotation. Moreover, constructing such datasets is a difficult task due to the subjectiveness of defining fragment boundaries. In most of the cases it is not clear where exactly a fragment boundary exists, even to lecturer. Thus, in a 1-2 hour lecture, where the transcripts of free continuous speech of a speaker are available, the fragmentation results will be quite arbitrary even if coming from a human expert.

To overcome this problem, we choose to follow an approach well-known in the document segmentation field. Following [3], in which document fragments of various lengths were concatenated and formed new documents, we have created a new dataset[1] of artificially-generated lectures. We used 1498 transcript files from the world's biggest academic online video repository, the VideoLectures.NET. These transcripts correspond to lectures from various fields of science, such as Computer science, Mathematics, Medicine, Politics etc. We split all transcripts in random fragments, the duration of which ranges between 4 and 8 minutes. A synthetic lecture is then created by combining exactly 20 randomly-selected parts. The first 300 such artificially-generated lectures were chosen for assembling our test dataset. Each such lecture file has a mean duration of about 120 minutes, and the overall dataset contains about 600 hours of artificially-generated lectures. Every pair of consecutive fragments in these lectures originally comes from different videos, consequently the point in time where such two fragments are joined is a known ground-truth fragment boundary. All these boundaries form the dataset's ground truth. We should stress that we do not generate the corresponding video files for the artificially-generated lectures (only the transcripts) and we do not use in any way the visual modality for finding the fragments.

## 4.2 Evaluation measures

To evaluate the performance of our video lecture fragmentation method, the Precision, Recall and $F-$Score measures were employed. In order to account for possible small differences between a ground truth fragment boundary $F^{GT}$ and a predicted one $F^{PR}$, we assign a score to every $(F^{GT}, F^{PR})_q$ pair, $(q = 1, \ldots, Q)$, where $Q$ is the total number of all pairs, which is calculated as follows:

$$S(F^{GT}, F^{PR})_q = \begin{cases} 1 & \text{if temporal distance between}(F^{GT}, F^{PR})_q < 30sec \\ 0 & otherwise \end{cases}$$

[1] Large-scale video lecture dataset and ground truth fragmentation available at `https://github.com/bmezaris/lecture_video_fragmentation`

In practice, this score introduces an error window to our calculations. We must mention that every $F^{GT}$ can be associated with just one $F^{PR}$. Precision ($P$) is defined as the fraction of the sum of all $(F^{GT}, F^{PR})_q$ pair scores over the number of the retrieved boundaries and Recall ($R$) is defined as the fraction of the sum of all $(F^{GT}, F^{PR})_q$ pair scores over the number of the ground-truth boundaries:

$$P = \frac{\sum_{q=1}^{Q} S(F^{GT}, F^{PR})_q}{L}, \quad R = \frac{\sum_{q=1}^{Q} S(F^{GT}, F^{PR})_q}{O}$$

where $L$ is the total number of predicted boundaries $F^{PR}$, and $O$ is the number of ground-truth boundaries $F^{GT}$. The F-Score is calculated by the standard formula: F-Score $= 2 \cdot P \cdot R/(P + R)$.

### 4.3 Results

In this subsection, our experimental results are presented. We evaluate the combination of the two cue extraction methods i) NP and ii) AVS, with the two different representations i) BoW and ii) Word2vec embeddings, as described in Section 3. First, we evaluate the performance of our method using a fixed number of 19 calculated fragment boundaries per video, which means we produce exactly 20 fragments for every artificially-generated video lecture. We also measure the performance of our system while the window size $N$ changes.

We compare our methods with two competitive works. The first one is the transcript based lecture video fragmentation of [14], which is actually identical to the BoW-NP combination of our experiments setup, with a fixed window size of 120 words. Moreover, we compare with the supervised text segmentation method of [9].

Table 1 reports the evaluation results of the three variations of the proposed method and the performance of [14] for a set of different window sizes, in Precision, Recall and F-Score. In Table 2 the proposed methods, using the best-performing window size from Table 1, are compared with [14] and [9]. As shown in Table 2, the best overall performance was achieved by the combination of the text analysis using Noun Phrases and Word2Vec representation. More specifically, from Tables 1 and 2 we conclude the following:

- Using the Word2Vec model consistently leads to better performance in terms of F-Score, regardless of the cue extraction method being used.
- Sliding window size matters. In contrast to previous works [10] [14], where a window was formed by a fixed number of 120 words, we show that performance can be significantly improved by varying the window size. When this is increased, the average number of the extracted cues that a window contains is also increased. Larger windows contain more semantically similar cues or multiple instances of the same cue, and are easier to distinguish from a neighboring window. However, there is an upper limit to the optimal window size, which possibly depends on the specifics of the lectures being fragmented (i.e., the size of the ground-truth fragments).

Proc. 25th Int. Conf. on Multimedia Modeling (MMM 2019), Springer LNCS vol. 11296. Author's accepted version.
The final publication is available at Springer via https://doi.org/10.1007/978-3-030-05716-9_21

Table 1: Experimental results (Precision, Recall and F-Score) of the three variations of the proposed approach, and comparison with [14] using different text window sizes.

| | Window size ($N$) | 120 | 240 | 360 | 480 | 600 | 720 | 840 | 960 | 1080 |
|---|---|---|---|---|---|---|---|---|---|---|
| BoW NP[14] | Precision | 0.287 | 0.228 | 0.204 | 0.262 | 0.349 | 0.415 | **0.426** | 0.408 | 0.391 |
| | Recall | 0.315 | 0.251 | 0.224 | 0.288 | 0.383 | 0.455 | **0.459** | 0.422 | 0.378 |
| | F-Score | 0.3 | 0.239 | 0.213 | 0.274 | 0.365 | 0.434 | **0.442** | 0.414 | 0.383 |
| | Avg Num_of_Cues | 20.78 | 42.03 | 63.15 | 84.28 | 105.30 | 126.18 | 146.98 | 167.68 | 188.25 |
| | Fragment duration mean | 330.5 | 330.5 | 330.5 | 330.5 | 330.7 | 332.0 | 338.6 | 354.0 | 380.1 |
| | Fragment duration std | 16.4 | 16.4 | 16.4 | 16.4 | 16.6 | 17.0 | 22.6 | 32.5 | 40.7 |
| BoW AVS | Precision | 0.27 | 0.281 | 0.287 | 0.315 | 0.365 | 0.398 | **0.415** | 0.386 | 0.377 |
| | Recall | 0.297 | 0.309 | 0.316 | 0.346 | 0.401 | 0.437 | **0.455** | 0.416 | 0.383 |
| | F-Score | 0.283 | 0.294 | 0.301 | 0.33 | 0.382 | 0.416 | **0.434** | 0.4 | 0.379 |
| | Avg Num_of_Cues | 17.31 | 34.93 | 52.49 | 70.13 | 87.54 | 104.95 | 122.22 | 139.42 | 156.57 |
| | Fragment duration mean | 330.5 | 330.5 | 330.5 | 330.5 | 330.5 | 330.6 | 332.1 | 338.6 | 361.0 |
| | Fragment duration std | 16.4 | 16.4 | 16.4 | 16.4 | 16.4 | 16.4 | 16.8 | 22.7 | 34.0 |
| Word2Vec NP | Precision | 0.335 | 0.248 | 0.252 | 0.29 | 0.373 | 0.427 | **0.465** | 0.448 | 0.427 |
| | Recall | 0.368 | 0.273 | 0.278 | 0.319 | 0.411 | 0.466 | **0.491** | 0.437 | 0.377 |
| | F-Score | 0.351 | 0.26 | 0.264 | 0.304 | 0.391 | 0.446 | **0.477** | 0.441 | 0.398 |
| | Avg Num_of_Cues | 20.77 | 42.07 | 63.20 | 84.28 | 105.30 | 126.18 | 146.98 | 167.68 | 188.25 |
| | Fragment duration mean | 330.5 | 330.5 | 330.5 | 330.5 | 330.6 | 333.0 | 345.3 | 375.3 | 417.1 |
| | Fragment duration std | 16.4 | 16.4 | 16.4 | 16.4 | 16.5 | 17.6 | 30.7 | 44.3 | 54.2 |
| Word2Vec AVS | Precision | 0.268 | 0.289 | 0.299 | 0.321 | 0.373 | 0.412 | **0.425** | 0.417 | 0.402 |
| | Recall | 0.295 | 0.318 | 0.329 | 0.353 | 0.41 | 0.453 | **0.46** | 0.423 | 0.377 |
| | F-Score | 0.281 | 0.303 | 0.313 | 0.336 | 0.391 | 0.431 | **0.441** | 0.419 | 0.388 |
| | Avg Num_of_Cues | 17.35 | 34.99 | 52.60 | 70.13 | 87.54 | 104.95 | 122.22 | 139.42 | 156.57 |
| | Fragment duration mean | 330.5 | 330.5 | 330.5 | 330.5 | 330.5 | 330.9 | 336.4 | 359.8 | 392.7 |
| | Fragment duration std | 16.4 | 16.4 | 16.4 | 16.4 | 16.4 | 17.0 | 22.1 | 36.8 | 52.2 |

- The NP cue extraction method consistently outperforms the AVS approach.
- The proposed method outperforms both [14] and [9]. Compared to the supervised segmentation method [9], our method is significantly better despite the fact that [9] is based on training data. However, the training corpus does not have similar structure with an annotated dataset consisting of lecture video transcripts, so we assume that the lack of a proper large-scale training dataset may be the reason for the non-competitive performance of [9].

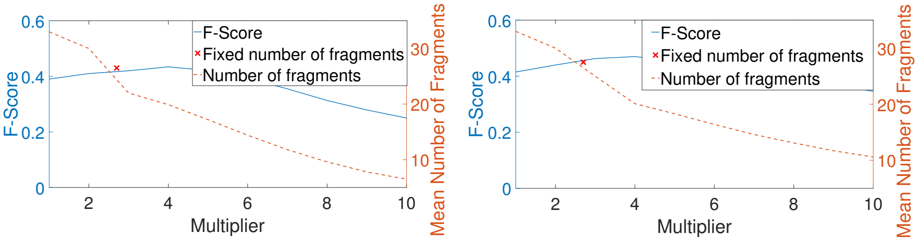
Figure 4 illustrates how the fragmentation performance is affected when we vary the number of fragments that are being generated. For this, in contrast to previous experiments, where the extracted number of fragments was fixed, in these experiments the number of fragments depends on a threshold $Thr$ (Eq. (1)). By varying the multiplier $m$ in (1) we can generate a variable number of fragments. The F-Score and corresponding number of fragments as a function of $m$ are shown in Fig. 4, while the red x indicates the F-Score achieved in the corresponding Table 2 experiment with a fixed number of 20 fragments. We observe that the F-Score is relatively insensitive to small variations in the value of $m$ and, correspondingly, in the number of fragments that are being generated.

## 5 Conclusions

In this work we proposed a new method to fragment lecture videos in meaningful parts. Our method takes advantage of the produced speech transcripts of a video,

Table 2: Experimental comparison of the three variations of the proposed approach, for the most appropriate window size (as shown in Table 1), with [14] and the supervised segmentation method of [9].

| | BoW AVS | Word2Vec NP | Word2Vec AVS | BoW AVS[14] | Supervised segmentation[9] |
|---|---|---|---|---|---|
| Precision | 0.415 | **0.465** | 0.425 | 0.426 | 0.237 |
| Recall | 0.455 | **0.491** | 0.46 | 0.459 | 0.393 |
| F-Score | 0.434 | **0.477** | 0.441 | 0.434 | 0.293 |



(a) BoW-NP

(b) Word2Vec-NP

Fig. 4: F-Score and mean number of generated fragments as a function of multiplier $m$ for (a) the BoW-NP, (b) the Word2Vec-NP variations of our method.

and analyzes them. We examined the performance of two different text analysis methods based on literature approaches in the video fragmentation and ad-hoc video search fields. A state-of-the-art word embedding was used for text representation, outperforming the classic N-gram approaches. Finally, we developed and provide online, a new large-scale dataset that consists of artificially-generated lectures and their corresponding ground truth fragmentation, which helps to overcome the lack of datasets for lecture video fragmentation evaluation.

## 6 Acknowledgements

This work was supported by the EUs Horizon 2020 research and innovation programme under grant agreement No 693092 MOVING. We are grateful to JSI/VideoLectures.NET for providing the lectures transcripts.

## References

1. Basu, S., Yu, Y., Singh, V.K., Zimmermann, R.: Videopedia: Lecture video recommendation for educational blogs using topic modeling. In: Int. Conf. on Multimedia Mod. pp. 238–250. Springer (2016)
2. Bhatt, C.A., Popescu-Belis, A., Habibi, M., Ingram, S., Masneri, S., McInnes, F., Pappas, N., Schreer, O.: Multi-factor segmentation for topic visualization and

recommendation: the must-vis system. In: Proc. of the 21st ACM Int. Conf. on Multimedia. pp. 365–368. ACM (2013)

3. Brants, T., Chen, F., Tsochantaridis, I.: Topic-based document segmentation with probabilistic latent semantic analysis. In: Proc. of the 11th Int. Conf. on Inf. and Knowl. Manag. pp. 211–218. CIKM '02, ACM, New York, NY, USA (2002)
4. Che, X., Yang, H., Meinel, C.: Lecture video segmentation by automatically analyzing the synchronized slides. In: Proc. of the 21st ACM Int. Conf. on Multimedia. pp. 345–348. ACM (2013)
5. Chen, H., Cooper, M., Joshi, D., Girod, B.: Multi-modal language models for lecture video retrieval. In: Proc. of the 22nd ACM Int. Conf. on Multimedia. pp. 1081–1084. ACM (2014)
6. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proc. of the 43rd Annual Meeting on Assoc. for Computational Linguistics. pp. 363–370. ACL '05 (2005)
7. Glavaš, G., Nanni, F., Ponzetto, S.P.: Unsupervised text segmentation using semantic relatedness graphs. Association for Computational Linguistics (2016)
8. Hearst, M.A.: Texttiling: Segmenting text into multi-paragraph subtopic passages. Computational linguistics **23**(1), 33–64 (1997)
9. Koshorek, O., Cohen, A., Mor, N., Rotman, M., Berant, J.: Text segmentation as a supervised learning task. In: Proc. of the 2018 Conf. of the North American Ch. of the Assoc. for Comp. Ling.: Human Language Technologies, Volume 2 (Short Papers). pp. 469–473 (2018)
10. Lin, M., Chau, M., Cao, J., Nunamaker Jr, J.F.: Automated video segmentation for lecture videos: A linguistics-based approach. Int. Jour. of Technology and Human Interaction (IJTHI) **1**(2), 27–45 (2005)
11. Ma, D., Zhang, X., Ouyang, X., Agam, G.: Lecture vdeo indexing using boosted margin maximizing neural networks. In: Machine Learning and Appl. (ICMLA), 2017 16th IEEE Int. Conf. on. pp. 221–227. IEEE (2017)
12. Markatopoulou, F., Galanopoulos, D., Mezaris, V., Patras, I.: Query and keyframe representations for ad-hoc video search. In: Proc. of the 2017 ACM on Int. Conf. on Multimedia Retrieval. pp. 407–411. ICMR '17, ACM (2017)
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Adv. in Neural Inf. Proc. Systems 26, pp. 3111–3119. Curran Associates, Inc. (2013)
14. Shah, R.R., Yu, Y., Shaikh, A.D., Zimmermann, R.: Trace: Linguistic-based approach for automatic lecture video segmentation leveraging wikipedia texts. In: 2015 IEEE Int. Symp. on Multimedia (ISM). pp. 217–220 (Dec 2015)
15. Shah, R.R., Yu, Y., Shaikh, A.D., Tang, S., Zimmermann, R.: Atlas: automatic temporal segmentation and annotation of lecture videos based on modelling transition time. In: Proc. of the 22nd ACM Int. Conf. on Multimedia. pp. 209–212 (2014)
16. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proc. of the 2003 Conf. of the North American Ch. of the Ass. for Comp. Ling. on Human Lang. Tech. - Volume 1. pp. 173–180. NAACL '03 (2003)
17. Yang, H., Siebert, M., Luhne, P., Sack, H., Meinel, C.: Automatic lecture video indexing using video ocr technology. In: 2011 IEEE Int. Symp. on Multimedia. pp. 111–116 (Dec 2011)
18. Yang, H., Meinel, C.: Content based lecture video retrieval using speech and video text information. IEEE Transactions on Learning Technologies **7**(2), 142–154 (April-June 2014)