Unsupervised Audio-Visual Lecture Segmentation

Darshan Singh S* Anchit Gupta* C. V. Jawahar Makarand Tapaswi
CVIT, IIIT Hyderabad
projectpageurl

Abstract

Over the last decade, online lecture videos have become increasingly popular and have experienced a meteoric rise during the pandemic. However, video-language research has primarily focused on instructional videos or movies, and tools to help students navigate the growing online lectures are lacking. Our first contribution is to facilitate research in the educational domain by introducing AVLectures*, a large-scale dataset consisting of 86 courses with over 2,350 lectures covering various STEM subjects. Each course contains video lectures, transcripts, OCR outputs for lecture frames, and optionally lecture notes, slides, assignments, and related educational content that can inspire a variety of tasks. Our second contribution is introducing* video lecture segmentation *that splits lectures into bite-sized topics. Lecture clip representations leverage visual, textual, and OCR cues and are trained on a pretext self-supervised task of matching the narration with the temporally aligned visual content. We formulate lecture segmentation as an unsupervised task and use these representations to generate segments using a temporally consistent 1-nearest neighbor algorithm, TW-FINCH [44]. We evaluate our method on 15 courses and compare it against various visual and textual baselines, outperforming all of them. Our comprehensive ablation studies also identify the key factors driving the success of our approach.*

1. Introduction

The last decade has seen a significant increase in online lectures in the form of Massive Open Online Courses (MOOCs) through platforms such as Coursera or EdX. Many high-quality recorded lectures are also published online, *e.g.*, MIT through MIT OpenCourseWare (OCW)[1], top Indian universities through NPTEL[2], and several professors that make their lectures publicly available[3]. This increase in online content is considered one of the biggest turning points in the history of education as anybody can learn any topic from the world's leading teachers from the comfort of their home [3, 22]. As the world moved to an online mode during the pandemic, there is absolutely no doubt that such online lecture content creation will only increase.

Creating an online course requires tremendous effort from the instructor and teaching assistants. Apart from designing and preparing the content itself, the mode of presentation poses challenges include segmenting the large videos into smaller topics to enhance the learning experience, adding quiz-like questions during the lecture to retain the student's engagement, summarizing the lecture at the end, *etc*. These tasks require carefully combing through the lecture several times, a time-consuming and error-prone process. Our goal is to encourage the community to address these tasks automatically or at least provide automatic recommendations for a human-in-the-loop system as they have the potential to reduce instructor's efforts, giving them more time and energy to improve the lecture content.

To build such solutions, machine understanding of audio-visual (AV) lectures is crucial. However, currently, there are no large-scale datasets of audio-visual lectures[4]. Our *first contribution* is *AVLectures*, a large-scale dataset to facilitate research in automatic understanding of lecture videos (see Sec. 3 for details and statistics). By releasing *AVLectures*, we wish to ignite research in the largely overlooked applications in education to help manage the fast-growing online lecture content.

Our *second contribution* is the formulation and benchmarking of the *lecture segmentation* task, where, given a long video lecture, our goal is to temporally segment it into smaller bite-sized topics. Lecture segmentation can be more challenging than scene segmentation in movies [41] or cooking videos [28] as the differences across segments are subtle, in both the visual and transcribed narrations. For example, Fig. 1 shows a professor teaching on the blackboard and walking along the podium. A model trained on movies or instructional videos may find it hard to segment the lecture as the objects or actions in the video do not change

[1]MIT-OCW - https://ocw.mit.edu/

[2]NPTEL - https://nptel.ac.in/

[3]*e.g.* Statistics 110 or Stanford's CS231n.

[4]Despite educational videos being the fourth most consumed content on the Internet according to this survey, just behind "How-to" videos.

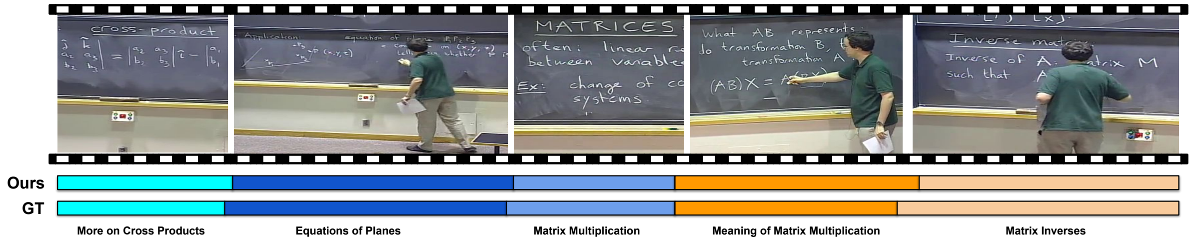

Figure 1: We address the task of lecture segmentation in an unsupervised manner. We show an example of a lecture segmented using our method. Our method predicts segments close to the ground-truth. Note that our method *does not predict the segment labels*, they are only shown so that the reader can appreciate the different topics.

significantly. Across segments, the visual boundaries are subtle changes such as clearing the board, while the narration may see a shift in the overall topic of discussion.

We propose lecture segmentation as an unsupervised task that leverages visual, textual, and OCR cues from the audio-visual lecture. We first split the lecture into small clips and extract each clip's visual and textual features using pre-trained models. To make our representations lecture-aware, we learn a joint text-video embedding in a self-supervised manner by matching the narration with the aligned visual content. Finally, we obtain clusters using a temporally consistent 1-nearest neighbor algorithm, TW-FINCH [44][5].

We pick lecture segmentation as our first use case based on an insightful large-scale study conducted on the EdX platform [23]. They find that students who successfully complete an online course typically spend 4.4 minutes on a 12-15 minute long lecture clip, clearly demonstrating the need for simplified navigation of long clips. Lecture segmentation is also a first step towards creating a multi-modal table of contents to summarize a lecture [32]. Finally, there is evidence for segmentation to assist in enabling non-linear video consumption [50] and efficient previewing [12, 16, 40]. While segmentation is our first task, we emphasize that *AVLectures* can be used for various other tasks in the future such as generating automatic quizzes for the lecture, aligning lecture videos with the notes enabling generation of lecture notes, retrieving relevant clips of the lecture using text queries, summarizing long lecture videos, retrieving and aligning similar courses/lectures from different learning platforms, and many more.

Our key contributions are summarized below. (i) We introduce a novel educational audio-visual lectures dataset, *AVLectures*, that can facilitate several applications in the education domain. (ii) We formulate and benchmark the problem of *unsupervised lecture segmentation*. We show that self-supervised multimodal representations learned by matching the narration with temporally aligned video clips greatly helps the task of segmentation. (iii) Our method outperforms several baselines. We also provide extensive ablation studies to understand prominent factors leading to the success of our approach. We will release code and data.

[5]Temporally *consistent* here refers to temporally *contiguous*, *i.e.* the segment membership of clips looks like [0, 0, 1, 1, 1, 2, 2] rather than [0, 1, 0, 2, 2, 1, 1]. TW-FINCH [44] allows this over base FINCH [45].

2. Related Work

Applications in educational videos. Research in video-language domain has focused primarily on movies [39, 42, 48], and instructional videos [7, 36, 43], especially cooking videos [17, 55]. However, there are a few isolated works [13, 14, 20, 22, 31, 32] that attempt to solve various problems in the education domain that we highlight below. Mahapatra *et al.* [31] propose an approach to generate a hierarchical table of contents for a lecture video using multimodal information such as transcripts and associated metadata from video key frames. In the direction of localizing and recognizing text on a blackboard, Dutta *et al.* [20] introduce LectureVideoDB, a dataset consisting of frames from multiple lecture videos (including blackboard). Bulathwela *et al.* [13, 14] introduce datasets to understand learner engagement with educational videos.

Related to our work, lecture video segmentation was first proposed by Gandhi *et al.* [22]. A visual saliency algorithm is adopted to find the topic transition points in the lecture automatically, however, this works primarily for slide-based lectures. In contrast, our method shows promising results across all lecture types: blackboard, slide-based, and digital board. Additionally, the dataset of [22] is orders of magnitude smaller, 10 vs. 2,350 lectures. Finally, AVLectures is not only video material but is augmented by rich metadata, including transcripts, OCR outputs for slides/blackboard frames, lecture notes, lecture slides, and assignments.

Joint representation learning of video and language. Our proposed model learns meaningful representations of lectures and aligned transcripts, which we use to perform the lecture segmentation task. In this section, we review popular works that address joint representation learning in video and language. A common self-supervised objective used to learn good representations is aligning video with it's corresponding narrations [34, 36], which can then be used for a number of downstream tasks such as text-to-video retrieval [21, 29, 36], visual question answering [9, 48, 53], video captioning [26, 38, 54], natural lan-

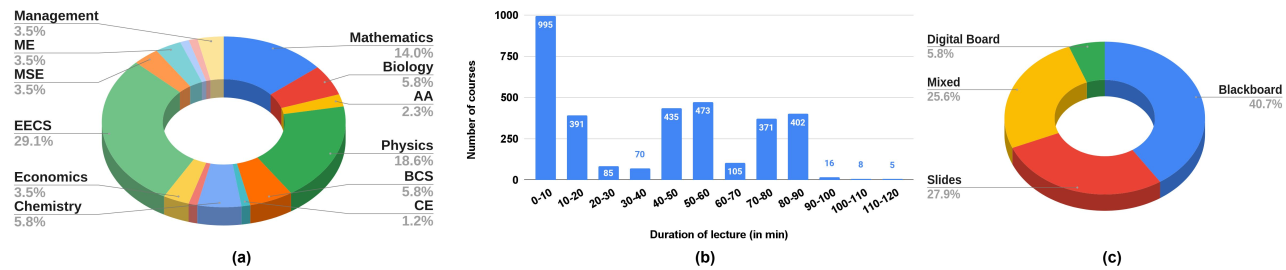

Figure 2: AVLectures statistics. (a) **Subject areas.** ME: Mechanical Eng., MSE: Materials Science and Eng., EECS: Electrical Eng. and Computer Science, AA: Aeronautics and Astronautics, BCS: Brain and Cognitive Sciences, CE: Chemical Eng. (b) **Lecture duration** distribution. (c) **Presentation modes** distribution.

guage guided video summarization [37] among others. Typically, representations from off-the-shelf pre-trained visual and language models are improved via a joint video-text embedding trained on the alignment task [36]. Recent approaches [18, 21, 29] also adopt Transformer-based models that learn in an end-to-end manner from raw video pixels. Our work explores the first direction. We extract video features using off-the-shelf models and combine them with OCR features. Then joint embeddings are learned using a pretext self-supervised task of matching the embeddings from narrations with temporally aligned video clips.

Temporal video segmentation. While fully supervised [19], weakly supervised [30, 47], and unsupervised [6, 7, 28, 44] approaches have been explored, we adopt the unsupervised path as collecting ground-truth segmentation labels is challenging, and we would like our method to generalize to diverse courses from novel educational platforms. In the unsupervised space, instructional videos are segmented by finding and grouping direct object relations in the narrations [7] or through the use of frame-level features that incorporate relative temporal information followed by K-means clustering (CTE) [28]. Proxy tasks such as future frame prediction are also used to perform temporal segmentation [6]. Recently, a temporally weighted version of a 1-nearest neighbor clustering algorithm is proposed to produce temporally consistent clusters (TW-FINCH) [44]. We will show that self-supervised joint text-video representation learning together with TW-FINCH leads to good segmentation performance on AVLectures.

3. The AVLectures Dataset

We introduce *AVLectures*, a large-scale educational audio-visual lectures dataset to facilitate research in the domain of lecture video understanding. The dataset comprises of 86 courses with over 2,350 lectures for a total duration of 2,200 hours. Each course in our dataset consists of video lectures, corresponding transcripts, OCR outputs for frames, and optionally lecture notes, slides, and other metadata, making our dataset a rich multi-modality resource.

Courses span a broad range of subjects, including Mathematics, Physics, EECS, and Economics (see Fig. 2a). While the average duration of a lecture in the dataset is about 55 minutes, Fig. 2b shows a significant variation in the duration. We broadly categorize lectures based on their presentation modes into four types: (i) Blackboard, (ii) Slides, (iii) Digital Board, and (iv) Mixed - a combination of blackboard and slides. Fig. 2c depicts a healthy distribution of presentation modes in our dataset. Additional statistics about AVLectures are discussed in supplementary material.

Courses with Segmentation. Among the 86 courses in AVLectures, a significant subset of 15 courses also have temporal segmentation boundaries. We refer to this subset as the *Courses with Segmentation* (CwS) and the remainder 71 courses as the *Courses without Segmentation* (CwoS).

3.1. Dataset Collection Procedure

Our dataset is primarily sourced from MIT-OCW [4]. We curated a list of courses by browsing the OCW website and used web scraping tools to download the video lectures and accompanying metadata such as narration transcripts, assignments, lecture notes/slides, *etc*. Non-lecture videos (*e.g*. instructor interviews) that were found in some courses are manually discarded. We process and store the OCR outputs of video frames in each lecture using Google Cloud Vision API. As sudden changes in the visual content of a lecture are rare, we process one frame at every 10 seconds.

3.2. Curating the Lecture Segmentation Dataset

It is shown that partitioning a long duration lecture into shorter topic-based clips helps in capturing students' attention and improves the overall learning experience [23, 50]. However, manually segmenting lecture recordings is a time-consuming and costly task. To evaluate automatic methods for lecture segmentation, we create a subset of our dataset, called *Courses with Segmentation* (CwS), that includes courses in which long lecture videos are segmented into multiple smaller clips. We curate 15 such courses with 350 lectures in total, where temporal segmentation *ground-truth* (for each lecture) is obtained in one of two ways. (i)

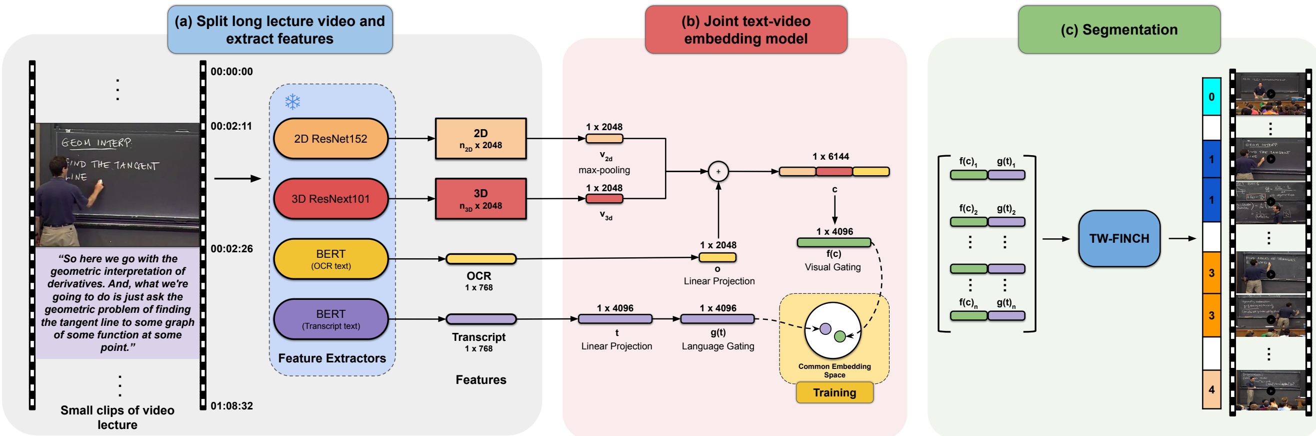

Figure 3: **Segmentation pipeline.** (a) *Video clip and feature extraction pipeline* used to extract visual and textual features from small clips of 10s-15s duration. The feature extractors are frozen and are not fine-tuned during the training process. (b) *Joint text-video embedding model* learns lecture-aware representations. (c) *Lecture segmentation process*, where we apply TW-FINCH at a clip-level to the learned (concatenated) visual and textual embeddings obtained from (b).

Out of the 15 courses, 5 courses[6] have topics in the table of contents refer to various temporal segments in a long lecture video. We obtain the segmentation timestamps for such courses directly by web scraping. (ii) The rest of the 10 courses[7] have concepts that are presented as pre-segmented short videos. Here, we re-assemble the small segments to build the original complete lecture. We trim the intro and outro from short video clips to avoid biasing the models to identify the segments easily.

4. Lecture Segmentation

Our lecture segmentation approach involves three stages (Fig. 3). In the first stage we extract features from diverse modalities of the lecture (Sec. 4.1 and Fig. 3a). In the second stage, we learn lecture-aware representations by aligning the visual content with the corresponding narration using self-supervision (Sec. 4.2 and Fig. 3b). Finally, we perform segmentation using TW-FINCH [44] on the learned representations (Sec. 4.3 and Fig. 3c).

4.1. Video clip feature extraction

We divide a lecture into small clips of 10-15 seconds while ensuring that subtitles are not split. This clip is a basic unit for segmentation, *i.e.* segmentation boundaries can be placed before or after, not in between. The chosen duration is small enough to not introduce boundary errors for segmentation but big enough to contain meaningful information about the lecture, as will also be shown empirically.

Video feature extraction. The visual clip representation consists of three feature types: OCR, 2D, and 3D. The *OCR feature* encodes the output text from an OCR API using the BERT sentence transformer model. Specifically, we use MPNet [46] (`all-mpnet-base-v2`) [52] from HuggingFace to obtain a 768-dimensional vector that captures the semantic information of the recognized text. The *2D and 3D features* are extracted using a video feature extraction pipeline [36]. An ImageNet pre-trained Resnet-152 [25] model produces 2D features at 1 fps while the 3D features are extracted using the Kinetics [15] pre-trained ResNeXt-101 [24] to obtain 1.5 features per second. We apply max-pooling across the temporal dimension to obtain 2048-dimensional vectors, $\mathbf{v_{2d}}$ and $\mathbf{v_{3d}}$ respectively.

Text feature extraction uses the same model as used for OCR. The text feature encodes the instructor's spoken words or subtitles corresponding to each video clip.

4.2. Learning joint text-video embeddings

Our approach transforms features from off-the-shelf models into lecture-aware embeddings and is inspired by popular works on instructional videos [36, 43].

Model architecture. Fig. 3b depicts our model used to learn lecture-aware embeddings by matching the visual feature of a clip with its corresponding text pair. We first extract the visual and textual features for a video clip C and transcript (text) T using the feature extraction pipelines described above. We pass the OCR feature through a fully-connected layer to obtain a 2048-dimensional vector $\mathbf{o}$, and concatenate it with $\mathbf{v_{2d}}$ and $\mathbf{v_{3d}}$ to form a 6144-dimensional vector $\mathbf{c}$ describing the clip C. Similarly, the text feature vector (output of the transformer) is passed through a fully connected layer to obtain a 4096-dimensional vector $\mathbf{t}$, representing text T. Next, we learn a projection using the non-linear context gating [35, 36] de-

[6](i) *e.g.* Single Variable Calculus
[7](ii) *e.g.* Classical Mechanics

fined as follows:

$$f(\mathbf{c}) = (W_1^c\mathbf{c} + b_1^c) \odot \sigma(W_2^c(W_1^c\mathbf{c} + b_1^c) + b_2^c), \quad (1)$$

$$g(\mathbf{t}) = (W_1^t\mathbf{t} + b_1^t) \odot \sigma(W_2^t(W_1^t\mathbf{t} + b_1^t) + b_2^t), \quad (2)$$

where $W_1^c, W_2^c, W_1^t, W_2^t$ and $b_1^c, b_2^c, b_1^t, b_2^t$ are learnable parameters, $\odot$ is element-wise multiplication and σ is an element-wise sigmoid. $f(\mathbf{c})$ and $g(\mathbf{t})$ are 4096-dimensional embeddings, which are used later for the segmentation task.

Loss function. We train our embedding model's parameters with the max-margin ranking loss [27, 51]. Specifically, we consider the (cosine) similarity score between a clip C_i and transcript T_j as $s_{ij} = \langle f(\mathbf{c}_i), g(\mathbf{t}_j)\rangle$. We loop over paired samples of a mini-batch $\mathcal{B}$ and compute the loss as

$$\sum_{i\in\mathcal{B}} \sum_{j\in\mathcal{N}(i)} \max(0, \delta+s_{ij}-s_{ii})+\max(0, \delta+s_{ji}-s_{ii}), \quad (3)$$

where s_{ii} corresponds to a positive (aligned) clip-transcript pair (C_i, T_i) and should score high, while $\mathcal{N}(i)$ is the set of negative pairs such that half the negative pairs are from the same lecture and act as hard negatives, while the others stem from other lectures [8, 36]. Our mini-batch size is $|\mathcal{B}| = 32$ and the margin is set at $\delta = 0.1$.

4.3. Lecture segmentation with learned embeddings

We extract clip and transcript embeddings from our joint text-video model and concatenate them to obtain an overall representation $\phi_i = [f(\mathbf{c}_i), g(\mathbf{t}_i)]$. All such representations of a lecture with N clips, $\{\phi_1, \dots, \phi_N\}$, are passed to the TW-FINCH algorithm [44] that encodes feature similarity and temporal proximity as a 1-nearest-neighbor graph and produces a clustering as shown in Fig. 3c. Specifically, we denote the feature similarity between clips as E_s and temporal proximity as E_τ.

$$E_s(m, n) = \begin{cases} 1 - \langle\phi_m, \phi_n\rangle & \text{if } m \neq n\,, \\ 1 & \text{otherwise}\,. \end{cases} \quad (4)$$

$$E_\tau(m, n) = \begin{cases} |\tau_m - \tau_n|/T & \text{if } m \neq n\,, \\ 1 & \text{otherwise}\,, \end{cases} \quad (5)$$

where $m, n \in [1, \dots, N]$, τ_m and τ_n are timestamps for the clips m and n and T is the total lecture duration.

We construct a fully-connected graph $\mathcal{G}$ with N nodes that have edge distances obtained as a combination of feature-space distances and temporal proximity

$$E(m, n) = E_s(m, n) \cdot E_\tau^\alpha(m, n)\,, \quad (6)$$

where α acts as a further modulating factor. The graph $\mathcal{G}$ is converted to a 1-nearest-neighbor graph by keeping only one edge to the *nearest* node for each node based on the edge distances defined in E, resulting in the first clustering partition. TW-FINCH [44] operates recursively and merges clusters (nodes) by averaging their representations and timestamps until the desired number of clusters (connected components) is obtained. For more details, we request the reader to refer to Algorithm 1 and 2 in [44].

Note that the original algorithm [44] does not include an α scaling factor, or considers it to be 1 (*c.f.* Eq. 6). However, we observed a few cases where this is unable to produce temporally consistent segments using our learned embeddings. As higher values of alpha amplify the strength of the temporal proximity factor, incrementing it progressively (*e.g.* by 0.1 steps) yields temporally consistent clusters.

5. Experiments

We evaluate our proposed approach for lecture segmentation and present extensive ablation studies.

5.1. Experiment setup

Training procedure involves two stages. In the first stage, we pre-train the embedding model (Sec. 4.2) on the Courses without Segmentation (CwoS). In the second stage, we fine-tune our embedding model on the Courses with Segmentation (CwS) in an unsupervised manner. Note that we do not update the feature extraction backbones (BERT, ResNet, *etc.*). Next, we extract the visual and textual embeddings from the trained model, which are used to perform segmentation using the TW-FINCH algorithm. We evaluate the segments obtained from TW-FINCH using five different metrics described below. Additional training details can be found in the supplementary material (Sec. E).

Evaluation dataset. We evaluate on all 15 courses of CwS to report performance. Our self-supervised fine-tuning process can be easily extended to a new course that needs segmentation. Further impact of pre-training and fine-tuning strategies is evaluated in Sec. 5.3, Ablation 2.

Evaluation metrics. Normalized Mutual Information (NMI) is a standard clustering metric [33]; Mean over Frames (MoF), F1-score, and Intersection over union (IoU) or the Jaccard index are standard metrics used in segmentation (*e.g.* [44]); and Boundary Score @ k (BS@k), is the average number of predicted boundaries matching with the ground truth boundaries within a k second interval. Different from above metrics, BS@k measures localization of boundaries rather than the overlap of segments.

5.2. Comparison against Segmentation Baselines

We briefly describe the baselines below:

1. Naïve. The video lecture is split into equal parts based on the number of ground-truth (GT) segments.

2. Content-Aware Detector [2] is a scene detection algorithm that detects jump cuts in a video by finding areas of

	Method	Feature modality visual	textual	learned	NMI ↑	MOF ↑	IOU ↑	F1 ↑	BS@30 ↑
1	Naïve (Equal Splits)	-	-	-	71.8	75.5	62.7	74.0	32.5
2	Content-Aware Detector [2]	✓	-	-	72.9	73.3	59.4	65.9	57.0
3	Text Tiling [5]	-	✓	-	67.9	64.7	46.3	50.9	33.7
4	LDA [11]	-	✓	-	70.0	72.4	57.6	68.2	38.8
5	K-Means	-	-	✓	63.9	66.8	48.2	55.7	44.9
6	CTE [28]	-	-	✓	67.2	67.3	48.1	57.3	41.5
7		✓	-	-	71.6	71.3	56.5	66.4	46.9
8	Vanilla TW-FINCH [44]	-	✓	-	74.6	75.4	62.0	71.2	48.9
9		✓	✓	-	74.9	75.1	61.7	70.9	52.1
10	**Ours**	-	-	✓	**79.8**	**80.3**	**69.2**	**76.9**	**58.7**

Table 1: Segmentation performance on all 350 lectures from 15 courses. Our approach outperforms all baselines. Here, *learned feature modality* refers to the features extracted from our joint text-video embedding model (Sec. 4.2). For rows 2-4, the *visual and textual feature modalities* refer to the unprocessed lecture video or transcripts respectively. For rows 7-9, *visual and textual feature modalities* refer to the features obtained from pre-trained backbones (ResNet or BERT, Sec. 4.1).

high difference between two adjacent frames. While there is no direct way to set the number of segments, we search across several thresholds to generate the GT number of segments to ensure a fair comparison.

3. Text Tiling utilizes only the transcripts to predict the segments. We implement text tiling using the NLTK [5] library. As there is no way to set the number of clusters, we let the algorithm decide the appropriate number of clusters.

4. Latent Dirichlet Allocation (LDA) [1, 11] is a generative probabilistic model that automatically discovers hidden topics based on a text corpora. LDA is used as a baseline in identifying topic transitions in educational videos [22] and many other topic modeling works [10, 49]. We train the LDA model on the transcripts of AVLectures and represent each clip as a distribution over topics. Finally, we use TW-FINCH to perform lecture segmentation using these vectors.

5. K-Means clustering algorithm is applied to the learned embeddings from our joint text-video embedding model.

6. CTE [28] is a *strong unsupervised approach* that infuses features with relative temporal information and clusters them using K-Means. We report CTE scores using learned embeddings from our joint model.

7. Vanilla TW-FINCH [44]. Visual and textual features from the feature extraction pipeline described in Sec. 4.1 are adopted here. We apply the TW-FINCH segmentation algorithm directly on these features.

We compare all baselines against our approach and report performance in Table 1. For K-Means (row 5) and CTE (row 6), we report the best performance with learned features, while detailed ablations are presented in the Sec. D of the supp. mat. We observe that the Naïve baseline (row 1) performs quite well, and in fact outperforms strong baselines with learned features such as K-Means (row 5) and CTE (row 6). This may be due to an inherent bias of the instructor spending close to equal amounts of time on various sub-topics of the lecture (supp. mat. Sec. D digs deeper into this). The text-only approach, Text Tiling (row 3) lags behind the visual-only approach Content-Aware Detector (row 2) as the latter performs specially well on non-blackboard courses (see Fig. 5). An additional factor is that we are unable to select the ground-truth number of clusters for Text Tiling. Our approach (row 10) outperforms all baselines. In fact, the gap between our approach and Vanilla TW-FINCH baselines (rows 7-9) highlights the importance of training lecture-aware representations using the joint text-video embedding model, as even a combination of both modalities (row 9) falls short of our approach by almost 5% on NMI. This emphasizes the importance of learning lecture-aware embeddings in a self-supervised manner.

We further analyze the results by slicing lectures based on the number of GT segments in Fig. 4. Our method outperforms all the other baselines irrespective of the number of segments in the ground truth, indicating the robustness of our approach. Another way is to slice the data based on presentation mode, specifically blackboard and non-blackboard. Fig. 5 shows a similar trend, our approach outperforms all baselines in both scenarios. Interestingly, the Naïve baseline works well for blackboard lectures (perhaps indicative of relatively equal time allocation across sub-topics), while slide-based lectures with clear transitions are segmented well by the visual Content-Aware Detector.

5.3. Ablation Studies

We present various ablation studies to understand the contributing factors to our approach's performance.

1. How important is each visual feature? To understand

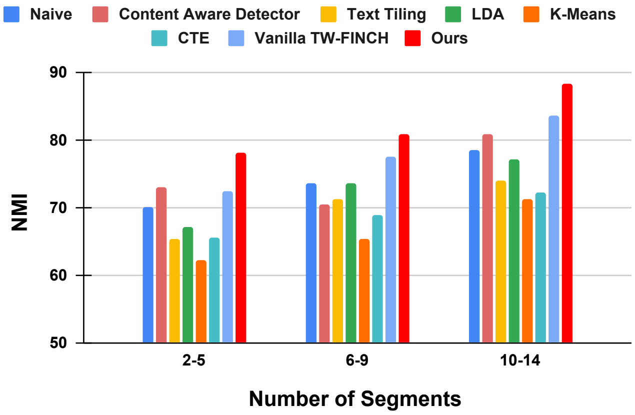

Figure 4: Comparing NMI across all methods grouped by the number of ground-truth segments.

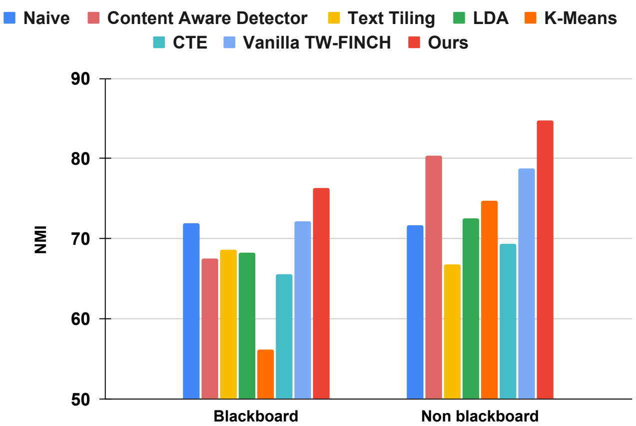

Figure 5: Comparing NMI across all methods grouped by presentation mode: blackboard and non-blackboard.

Features			Metrics				
2D	**3D**	**OCR**	**NMI ↑**	**MOF ↑**	**IOU ↑**	**F1 ↑**	**BS@30 ↑**
✓	-	-	76.6	76.8	64.4	73.0	54.4
-	✓	-	75.1	76.0	62.9	72.2	50.7
-	-	✓	78.9	79.7	68.2	76.2	57.7
✓	✓	-	76.6	77.0	64.7	73.5	53.9
✓	-	✓	79.5	80.3	69.1	76.9	58.6
-	✓	✓	78.4	79.5	68.3	76.4	57.9
✓	✓	✓	**79.8**	**80.3**	**69.2**	**76.9**	**58.7**

Table 2: Impact of visual features.

the impact of each individual visual feature, we train separate models on all combinations of visual features and report performance in Table 2. We observe that although the individual features perform reasonably well, with OCR outperforming 2D and 3D representations, it is the combination of all features that outperforms all other variations.

2. Impact of training datasets. Educational lecture videos are very different compared to instructional videos or movies. Lecture videos typically have much less dynamic visual content and compensate for this through substantial amounts of textual information, both accompanying (narrated speech/transcripts) and even inside the video (which we extract using OCR). As a result, the representations learned from instructional videos may not transfer well to the tasks in the education domain, necessitating a collection of lecture videos for learning representations.

	PT	FT	NMI ↑	MOF ↑	IOU ↑	F1 ↑	BS@30 ↑
1	HowTo100M	-	73.0	58.8	68.3	73.0	48.5
2	HowTo100M	CwS	74.5	75.1	61.5	71.0	49.7
3	-	CwS	78.5	79.0	67.2	75.3	57.2
4	CwoS	-	77.7	78.0	66.0	74.2	57.1
5	**CwoS**	**CwS**	**79.8**	**80.3**	**69.2**	**76.9**	**58.7**

Table 3: Impact of pre-training (PT) on HowTo100M or CwoS. The second column indicates whether unsupervised fine-tuning (FT) is performed on CwS.

Embed. type	NMI ↑	MOF ↑	IOU ↑	F1 ↑	BS@30 ↑
Visual	78.6	79.1	67.7	75.7	57.9
Textual	75.6	77.0	64.4	73.5	50.3
Visual + Textual	**79.8**	**80.3**	**69.2**	**76.9**	**58.7**

Table 4: Impact of different embedding modalities.

We validate the above claim by showing that pre-training on AVLectures is more effective than pre-training on the general instructional videos (*e.g*. HowTo100M) for the lecture segmentation task, see Table 3. While using a model to improve representations is clearly better than the naïve baseline (NMI 73.0 vs. 71.8), we can see that a model pre-trained on AVLectures (rows 3-5) outperforms a model pre-trained on HowTo100M (rows 1-2) consistently. This strengthens our dataset contribution and highlights the importance of pre-training on AVLectures for tasks in the education domain. In row 4, though the model is trained only on CwoS, it is able to generalize well to unseen courses and predict reasonable segmentation boundaries. After fine-tuning the model on CwS we get a slight boost in performance (row 5). Row 5 outperforms row 3 that is trained only on CwS, justifying our adoption of pre-training on CwoS followed by fine-tuning on CwS. Note that all the training is performed in an unsupervised manner and only applies to the text-video embedding model.

3. Impact of modalities. From the joint text-video embedding model we can extract visual and textual embeddings. We compare visual-only, textual-only, and a concatenation of visual and textual learned embeddings in Table 4. A combination of both modalities shows best results.

4. Impact of lecture clip duration. Works on instructional videos such as [34, 36] typically split videos into short clips of 4s. We perform an experiment to determine an appropriate clip duration for lecture videos: 4-8s, 10-15s, or 20-25s. The results reported in Table 5 coincide with our expec-

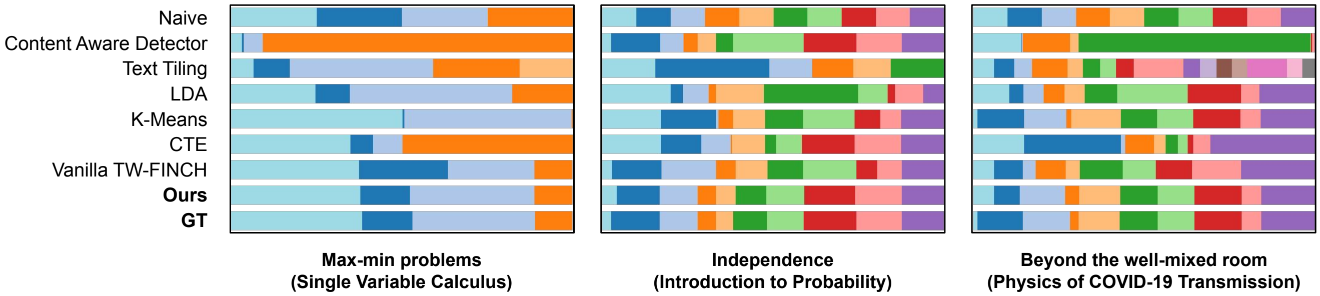

Figure 6: Segmentation examples for three lectures. Our approach closely resembles the ground-truth. Best viewed in color.

PT	FT	Duration	NMI ↑	MOF ↑	IOU ↑	F1 ↑	BS@30 ↑
		4-8	53.2	58.7	53.0	40.9	26.4
✓	-	**10-15**	**77.7**	**78.0**	**66.0**	**74.2**	**57.1**
		20-25	73.9	77.0	64.6	74.8	36.7
		4-8	54.6	60.0	54.1	42.2	26.6
✓	✓	**10-15**	**79.8**	**80.3**	**69.2**	**76.9**	**58.7**
		20-25	74.5	77.7	65.6	75.6	36.8

Table 5: Performance for different clip durations (in seconds). PT: Pre-training on CwoS, FT: Fine-tuning on CwS.

tations that 4-8s clips are too short to capture meaningful information while 20-25s clips are harder to represent due to the pooling operation and also cause a significant drop in BS@30 due to their longer duration. Clips of 10-15s are a good compromise and span meaningful lecture content while not losing information to pooling.

Additional ablations are presented in Sec. D of the supplementary material due to lack of space. (i) We analyze the impact of not knowing the GT number of segments; (ii) compare different language embedding models – two variations of MPNet and Word2Vec; (iii) compare embedding dimensionality; (iv) evaluate approaches at several values of k for the BS@k metric; and (v) observe that max-margin loss is better suited to our task and scale than NCE [34].

5.4. Qualitative results

We visualize segmentation outputs for three video lectures from different courses in Fig. 6 and compare our method with all other baselines. It is clear that our method yields better segments (overlap) as well as boundaries as opposed to other methods that produce noisy segments. In the third lecture, the first and second predicted segments of our approach are different from the GT while the other boundaries are detected correctly. We explain failure cases in Sec. B and show more results in Sec. F of the supp. mat.

An additional problem that can be addressed using the embeddings learned from our joint text-video model is the text-to-video retrieval task. Given a text query, we retrieve a list of lecture clips for which the similarity scores with the text query are the highest. While we do not perform a quantitative evaluation, Fig. 7 shows some of the retrieved clips for various text queries. We can see that our model is able to relate the visual notion of graphs with the word. Similar results are observed for the other queries. Supplementary material Sec. F shows many more examples.

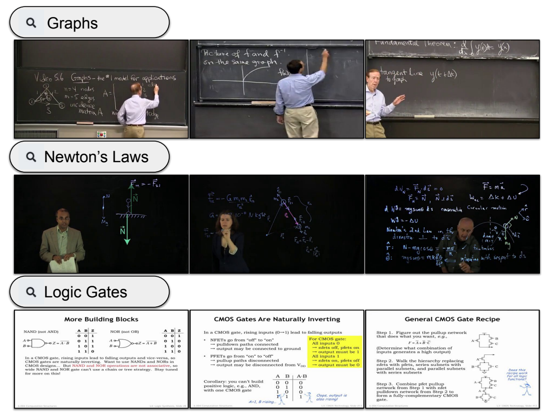

Figure 7: Examples of text-to-video retrieval for different queries using our learned joint embeddings. Our model is able to retrieve relevant lecture clips based on the query.

6. Conclusion

We made two significant contributions. We introduced *AVLectures*, a large-scale audio-visual lectures dataset sourced from MIT OpenCourseWare, with 86 courses and over 2,350 lectures from various STEM subjects and showed it's efficacy for pre-training on tasks in the educational domain. We also formulated *unsupervised lecture segmentation* and proposed an approach that learns multimodal representations by matching the narration with temporally aligned visual content. When used with TW-FINCH, the learned embeddings resulted in significant performance improvements and highlighted the importance of both the visual and the textual modalities. Thorough experiments demonstrated that our approach outperforms multiple baselines while comprehensive ablation studies identified the key factors that lead to the success of our approach: textual and visual representations with all 3 features (2D, 3D, OCR) and the pre-training and fine-tuning strategy.

Acknowledgement. This material is based upon work supported by the Google Cloud Research Credits program with the award GCP19980904. We also thank MIT-OCW for making their content publicly available which made this work possible.

References

[1] MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu, 2002.

[2] PySceneDetect. http://scenedetect.com/en/latest/reference/detection-methods/, 2021.

[3] Benefits of Using OER. https://oer.psu.edu/benefits-of-using-oer/, 2022.

[4] Mit opencourseware. https://ocw.mit.edu/, 2022.

[5] Natural Language Toolkit: TextTiling. https://www.nltk.org/_modules/nltk/tokenize/texttiling.html, 2022.

[6] Sathyanarayanan N Aakur and Sudeep Sarkar. A perceptual prediction framework for self supervised event segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *International Conference on Computer Vision (ICCV)*, 2017.

[9] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015.

[10] Federico Bianchi, Silvia Terragni, and Dirk Hovy. Pretraining is a hot topic: Contextualized document embeddings improve topic coherence. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 759–766, 2021.

[11] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning research*, 3(Jan), 2003.

[12] Sahan Bulathwela, Stefan Kreitmayer, and María Pérez-Ortiz. What's in it for me? augmenting recommended learning resources with navigable annotations. In *International Conference on Intelligent User Interfaces (IUI)*, 2020.

[13] Sahan Bulathwela, Maria Perez-Ortiz, Erik Novak, Emine Yilmaz, and John Shawe-Taylor. Peek: A large dataset of learner engagement with educational videos. *arXiv preprint arXiv:2109.03154*, 2021.

[14] Sahan Bulathwela, Maria Perez-Ortiz, Emine Yilmaz, and John Shawe-Taylor. VLEngagement: A Dataset of Scientific Video Lectures for Evaluating Population-based Engagement. *arXiv preprint arXiv:2011.02273*, 2020.

[15] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[16] Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. Temporally grounding natural sentence in video. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[17] P. Das, C. Xu, R. F. Doell, and J. J. Corso. A Thousand Frames in Just a Few Words: Lingual Description of Videos through Latent Topics and Sparse Object Stitching. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[18] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[19] Li Ding and Chenliang Xu. Tricornet: A hybrid temporal convolutional and recurrent network for video action segmentation. *arXiv preprint arXiv:1705.07818*, 2017.

[20] Kartik Dutta, Minesh Mathew, Praveen Krishnan, and CV Jawahar. Localizing and recognizing text in lecture videos. In *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2018.

[21] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*, 2021.

[22] Ankit Gandhi, Arijit Biswas, and Om Deshmukh. Topic transition in educational videos using visually salient words. *International Educational Data Mining Society*, 2015.

[23] Philip J Guo and Katharina Reinecke. Demographic differences in how students navigate through MOOCs. In *Proceedings of the ACM Conference on Learning @ Scale Conference*, 2014.

[24] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] Vladimir Iashin and Esa Rahtu. Multi-modal dense video captioning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[27] Andrej Karpathy, Armand Joulin, and Li F Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[28] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[29] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[30] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *International Conference on Computer Vision (ICCV)*, 2019.

[31] Debabrata Mahapatra, Ragunathan Mariappan, and Vaibhav Rajan. Automatic hierarchical table of contents generation for educational videos. In *Companion Proceedings of The Web Conference*, 2018.

[32] Debabrata Mahapatra, Ragunathan Mariappan, Vaibhav Rajan, Kuldeep Yadav, and Sudeshna Roy. Videoken: Automatic video summarization and course curation to support learning. In *Companion Proceedings of The Web Conference*, 2018.

[33] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval (chapter 16)*. Cambridge University Press, 2008.

[34] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[35] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *arXiv preprint arXiv:1804.02516*, 2018.

[36] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *International Conference on Computer Vision (ICCV)*, 2019.

[37] Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. Clip-it! language-guided video summarization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[38] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[39] Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. Movie summarization via sparse graph construction. *arXiv preprint arXiv:2012.07536*, 2020.

[40] Maria Perez-Ortiz, Claire Dormann, Yvonne Rogers, Sahan Bulathwela, Stefan Kreitmayer, Emine Yilmaz, Richard Noss, and John Shawe-Taylor. X5learn: A personalised learning companion at the intersection of AI and HCI. In *International Conference on Intelligent User Interfaces (IUI)*, 2021.

[41] Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A Local-to-Global Approach to Multi-modal Movie Scene Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[42] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *International Conference on Computer Vision (ICCV)*, 2017.

[43] Andrew Rouditchenko, Angie Boggust, David Harwath, Brian Chen, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Hilde Kuehne, Rameswar Panda, Rogerio Feris, et al. AVLnet: Learning audio-visual language representations from instructional videos. *arXiv preprint arXiv:2006.09199*, 2020.

[44] Saquib Sarfraz, Naila Murray, Vivek Sharma, Ali Diba, Luc Van Gool, and Rainer Stiefelhagen. Temporally-weighted hierarchical clustering for unsupervised action segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[45] Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[46] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[47] Yaser Souri, Mohsen Fayyaz, and Juergen Gall. Weakly supervised action segmentation using mutual consistency. *arXiv preprint arXiv:1904.03116*, 2019.

[48] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[49] Silvia Terragni, Elisabetta Fersini, Bruno Giovanni Galuzzi, Pietro Tropeano, and Antonio Candelieri. Octis: comparing and optimizing topic models is simple! In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 263–270, 2021.

[50] Gaurav Verma, Trikay Nalamada, Keerti Harpavat, Pranav Goel, Aman Mishra, and Balaji Vasan Srinivasan. Non-linear consumption of videos using a sequence of personalized multimodal fragments. In *International Conference on Intelligent User Interfaces (IUI)*, 2021.

[51] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[52] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[53] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just Ask: Learning to Answer Questions from Millions of Narrated Videos. In *International Conference on Computer Vision (ICCV)*, 2021.

[54] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[55] Luowei Zhou, Nathan Louis, and Jason J Corso. Weakly-supervised video object grounding from text by loss weighting and object interaction. *arXiv preprint arXiv:1805.02834*, 2018.