Shot Retrieval and Assembly with Text Script for Video Montage Generation

Guoxing Yang
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
yangguoxing@ruc.edu.cn

Haoyu Lu
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
lhy1998@ruc.edu.cn

Zelong Sun
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
zelongsun@ruc.edu.cn

Zhiwu Lu
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
luzhiwu@ruc.edu.cn

ABSTRACT

With the development of video sharing websites, numerous users desire to create their own attractive video montages. However, it is difficult for inexperienced users to create well-edited video montages due to the lack of professional expertise. In the meantime, it is time-consuming even for experts to create video montages of high quality, which requires effectively selecting shots from abundant candidates and assembling them together. Instead of manual creation, various automatic methods have been proposed for video montage generation, which typically take a single sentence as input for text-to-shot retrieval, and ignore the semantic cross-sentence coherence given complicated text script of multiple sentences. To overcome this drawback, we propose a novel model for video montage generation by retrieving and assembling shots with arbitrary text scripts. To this end, a sequence consistency transformer is devised for cross-sentence coherence modeling. More importantly, with this transformer, two novel sequence-level tasks are defined for sentence-shot alignment in sequence-level: Cross-Modal Sequence Matching (CMSM) task, and Chaotic Sequence Recovering (CSR) task. To facilitate the research on video montage generation, we construct a new, highly-varied dataset which collects thousands of video-script pairs in documentary. Extensive experiments on the constructed dataset demonstrate the superior performance of the proposed model. The dataset and generated video demos are available at https://github.com/RATVDemo/RATV.

CCS CONCEPTS

• **Computing methodologies** → **Visual content-based indexing and retrieval**; • **Information systems** → **Multimedia content creation**.

ICMR '23, June 12–15, 2023, Thessaloniki, Greece

ACM ISBN 979-8-4007-0178-8/23/06…$15.00
https://doi.org/10.1145/3591106.3592247

KEYWORDS

Video montage generation, text-to-video retrieval, multimedia content creation, transformer

ACM Reference Format:
Guoxing Yang, Haoyu Lu, Zelong Sun, and Zhiwu Lu. 2023. Shot Retrieval and Assembly with Text Script for Video Montage Generation. In *International Conference on Multimedia Retrieval (ICMR '23), June 12–15, 2023, Thessaloniki, Greece.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3591106.3592247

1 INTRODUCTION

In recent years, with the rapid development of video sharing websites, users can conveniently share their own edited short videos (i.e., video montages), resulting in numerous video content creators that desire to create attractive video montages. However, editing video montage well is not easy for most of users due to the lack of professional expertise and aesthetic knowledge for video editing. In addition, for experts that master the video editing skills, it is time-consuming and cumbersome to create a video montage of high quality, because they have to carefully select shots from abundant candidates and then assemble the selected shots into a consecutive video montage that precisely expresses the desired content. Automatically generating video montages from descriptive sentences thus becomes topical, which aims to effectively retrieve shots from candidates and assemble them according to given text scripts.

Over the past few years, a number of previous methods have been proposed to address this task based on deep learning [31, 32, 34]. QuickCut [31] presents an interactive tool for narrated video editing, which can quickly create the story outline from raw video footage. It focuses on speeding up the process of establishing the script, but requires the users to manually select the shots from candidates. Write-A-Video [32] and Transcript-to-Video [34] propose automatic methods to retrieve shots with given texts from a huge shot gallery and then arrange them. However, they only take a single sentence as input for text-to-shot retrieval without modeling the cross-sentence coherence, which limits their performance given complicated text script of multiple sentences.

In this work, we propose a novel model to automatically **R**etrieve and **A**ssemble shots with arbitrary **T**ext scripts for **V**ideo montage generation (abbreviated RATV). Our proposed RATV consists of

Table 1: Brief comparison among our RATV and existing works. SN is the number of sentences of the input text. RS is whether the method uses the retrieved shots for retrieval. Seq. Con. Trans. is short for sequence consistency transformer.

Method	Retrieval		Assembly	
	SN	RS	Strategy	Text Used
QuickCut† [31]	Multiple	No	DP Algorithm	No
Write-A-Video† [32]	Single	No	Predefined Rules	No
Transcript-to-Video† [34]	Single	Yes	Score of Classifier	No
RATV (ours)	Multiple	Yes	Seq. Con. Trans.	Yes

three main components: a textual encoder, a visual encoder, and a sequence consistency transformer. Similar to Write-A-Video [32], the textual encoder and visual encoder aim to encode texts and shots into a joint feature space for matching video-script pairs. Differently, we directly employ a large-scale pre-training visual-language model (CLIP [25]) to obtain our textual encoder and visual encoder instead of training them with text-shot pairs with keywords from scratch. Note that CLIP is trained on text-image retrieval task, but we generalize it to text-video matching task by simply considering the average of the frame embeddings as the embedding of corresponding video. Based on the embeddings extracted by textual encoder and visual encoder, the novel sequence consistency transformer learns to match sentence sequence and shot sequence, which can better retrieve shots according to complicated texts of multiple sentences and assemble the retrieved shots. Specifically, with this transformer, we devise two novel training tasks for better sentence-shot alignment in sequence-level: Cross-Modal Sequence Matching (CMSM) task and Chaotic Sequence Recovering (CSR) task. The CMSM task is induced to explicitly encourage the model to learn sentence-shot alignment by distinguishing the positive and negative samples. The CSR task enforces the model to learn to recover the order of chaotic shot sequences according to paired texts, which benefits to learning both the sequence coherence and sentence-shot alignment. Overall, the difference between our proposed RATV and existing methods is shown in Table 1.

To our best knowledge, there is no publicly available dataset for the research on video montage generation. Furthermore, despite the significant progress in video understanding datasets [8, 24, 35], these datasets focus on human actions (mainly with a single shot per video), which can not meet the demand of video montage generation with text script. Therefore, to fill the gap in dataset construction for this task, we create a new dataset (Video-Script Pairs in Documentary, VSPD) to facilitate the community, which consists of diverse video-script pairs collected from publicly available documentaries.

Our main contributions are four-fold: (1) We propose a novel model to automatically generate video montages by retrieving and assembling shots with *arbitrary* text scripts (containing one sentence or *multiple* sentences). To our best knowledge, our RATV is the first model for video montage generation based on text-to-sequence retrieval, which can generate video montages more consistent with the input text scripts. (2) We devise novel Cross-Modal Sequence Matching (CMSM) and Chaotic Sequence Recovering (CSR) tasks, which are beneficial to learning both sentence-shot alignment in *sequence-level* and the coherence of shot sequence with text scripts. (3) To fill the gap in the dataset construction for video montage generation, we introduce the VSPD dataset that consists of diverse and highly varied video-script pairs from documentary videos. Meanwhile, we establish a benchmark for video montage generation task to facilitate the community. (4) Extensive experiments on the constructed VSPD dataset demonstrate the effectiveness and superior performance of our proposed method.

2 RELATED WORK

Video Montage Generation. Video montage generation with text script has been proposed for a long time [1, 5], but it was barely studied in the community. In recent years, with the rapid development of video sharing websites, the demand of automatic video montage generation becomes higher, and thus this task starts to draw more attention [15, 27, 31, 32, 34]. Particularly, QuickCut [31] presents an interactive tool for narrated video editing, which aims to help users to efficiently create the story outline of narrated videos. However, it only supplies the video segments corresponding to selected footage and requires users to manually select shots from them. In contrast, Write-A-Video [32] presents an interactive tool that enables users to automatically retrieve shots from a huge shot gallery and assembly them based on pre-defined rules (e.g., saturation, luminance and video content). Transcript-to-Video [34] proposes an automatic method to retrieve and assemble shots according to given texts. Although they can automatically generate video montages from texts without extra manual work, they only consider one sentence as input during retrieval and assembly, which limits their performance given complicated texts containing multiple sentences. In this work, we propose a novel automatic method to retrieve and assemble shots with texts for video montages generation, which expands the input from only one sentence to a sequence of sentences.

Vision-Language Representation Learning. Visual Semantic Embeddings (VSE) [7, 9] are commonly adopted in multi-modal tasks to learn vision-language joint representations [2, 10, 14, 17, 30]. Recently, large-scale pre-training has achieved great success in vision-language representation learning [4, 11, 12, 21, 25], and shown superior performance in various downstream tasks (e.g., action recognition in videos, and zero-shot classification), where the extracted embeddings are directly employed without further training. In this work, we similarly leverage the embeddings extracted by CLIP [25] in video montage generation. Differently, we generalize the image-text representation to video-text representation and learn video-text alignment in sequence-level.

Training Tasks for Transformer-Based Modeling. Masked Language Model (MLM) is firstly proposed by BERT [6] and then widely used for NLP with transformer. The MLM task randomly replaces a word token with the mask token (or another word token) with certain probability and enforces the model to predict the original word token, which has shown its superior power in representation learning. Inspired by BERT, ViLBERT [22] employs MLM, Masked Object Classification (MOC) and Visual-Linguistic Matching (VLM) tasks for vision-language representation learning. More recent works basically follow these training tasks [4, 13, 18, 20, 21, 28, 29]. Different from these training tasks, we devise novel Cross-Modal Sequence Matching (CMSM) and Chaotic Sequence Recovering (CSR) tasks to encourage our transformer-based model to better learn the video-text joint representation as well as the sequence coherence.

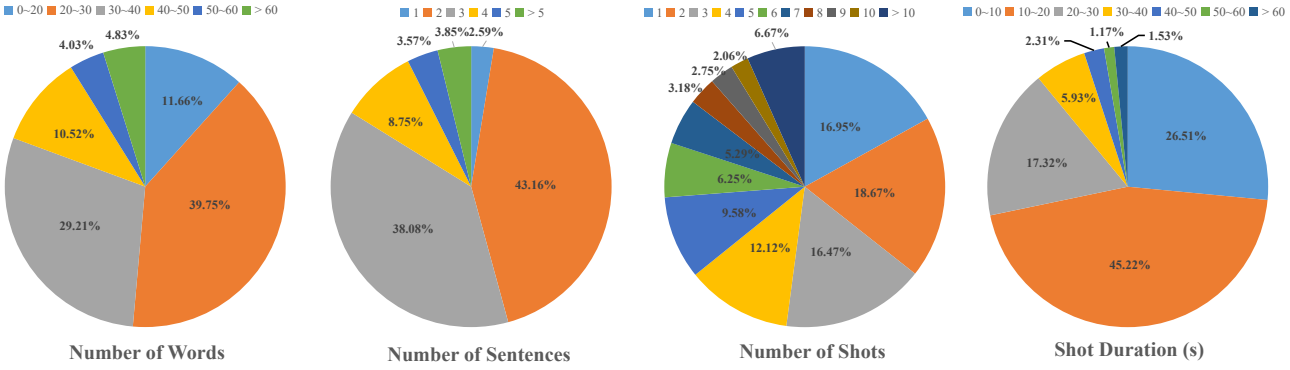

Figure 1: The statistics of text scripts and videos contained in our VSPD dataset.

3 DATASET CONSTRUCTION

A dataset for video montage generation should consist of thousands of video-script pairs, each having a narrated video with multiple shots and a paired script with descriptive sentences. To our best knowledge, there is no public dataset devoted to video montage generation. Similar to video montage generation, text-video retrieval aims to retrieve desired videos from candidates according to given texts, which has become topical in recent works [3, 16, 19, 33, 36, 37] along with many public datasets [8, 24, 26, 35]. However, these datasets are not suitable for video montage generation, because they mainly focus on human actions (with a single shot per video). Furthermore, the texts in these datasets either are too general or only cover a part of the video content. To fill the gap in dataset construction for video montage generation, we create a new, highly varied dataset, named VSPD (shorted for Video-Script Pairs in Documentary). Our VSPD dataset consists of 4,365 video-script pairs, with its statistics shown in Figure 1. We randomly select 200 video-script pairs for test and the other 4,165 pairs for training. Our considerations in collecting documentary videos are two-fold: (1) Most of documentary videos are narrated with well-aligned captions, and thus we can conveniently collect numerous video-script pairs that are highly consistent in semantics; (2) Documentary videos are commonly carefully edited, resulting in extensive consecutive video clips with multiple shots. Importantly, although our VSPD consists of videos from documentaries, our proposed RATV model can be adopted to generate videos with various themes, which mainly depends on the input texts instead of theme of videos in the gallery.

Note that video montage generation task is a highly subjective task, in which there can be multiple generated samples that could be almost equally good. Therefore, establishing a dataset to well evaluate this task is extreme challenging. To alleviate this issue, we collect video-script pairs from videos that have been well edited by the experts (e.g., movie, cartoon and MV). However, these thress types of videos are not good choice: (1) The captions of movie and cartoon are commonly not consistent with the shots because most of captions are dialogues. (2) The shots in MV are commonly not temporal coherence. As a result, we propose to collect the video-script pairs from documentaries, whose captions are descriptive and consistent with the temporally consecutive shots (as stated in the Section 3). More importantly, the documentaries are commonly shot and edited by the experts that have professional expertise. Therefore, the collected videos (i,e., the continuous clips of documentaries) in our VSPD dataset can indeed be considered as the perfect videos w.r.t. semantic consistency, temporal coherence and aesthetics, i.e., they can be used as the ground-truth for training and evaluation. With evaluation on such dataset, a model has to consider all of the factors (e.g., text-video alignment, temporal coherence, and aesthetics) for video montage generation. More details about our VSPD dataset are given in Appendix.

4 METHODOLOGY

4.1 Model Overview

Let $\mathcal{S}$ denote the set of candidate shots. Given an input text script t, the goal of video montage generation is to firstly retrieve a subset of shots $\tilde{S} = \{\tilde{s}_i | i = 1, \cdots, N\}$ from $\mathcal{S}$ according to t, where $\tilde{s}_i \in \mathcal{S}$ and N is the number of retrieved shots. The retrieved shots are then arranged in a certain order to create the final video montage $V = (v_1, v_2, \cdots, v_N)$, where $v_i \in \tilde{S}$ and $v_i \neq v_j$ $(\forall i, j = 1, 2, \cdots, N, i \neq j)$. Note that V should be temporally consecutive and consistent with the input text script t in semantics. To achieve this, we propose a novel model termed RATV. As illustrated in Figure 2, the main components of our RATV are textual encoder *TE*, visual encoder *VE* and sequence consistency transformer *ST*. In this work, the textual encoder and visual encoder can be formed with CLIP [25], both of which are frozen during training. Note that we employ image encoder of CLIP instead of other pre-trained video encoders due to its superior performance in vision-language semantic alignment (see Sec. 5.2). Given an input text script t, we first split it into sentences $t = (\tilde{t}_1, \tilde{t}_2, \cdots, \tilde{t}_m)$, where $\tilde{t}_i$ denotes the i-th sentence in text script t and m is the total number of sentences. We then encode each of them into a feature vector v_i^T $(i = 1, 2, \cdots, m)$ with the text encoder *TE*, and finally obtain a sequence of sentence embeddings $(v_1^T, v_2^T, \cdots, v_m^T)$. The process of text encoding is defined as:

$$\begin{aligned}(v_1^T, v_2^T, \cdots, v_m^T) &= TE(\tilde{t}_1, \tilde{t}_2, \cdots, \tilde{t}_m) \\ &= (TE(\tilde{t}_1), TE(\tilde{t}_2), \cdots, TE(\tilde{t}_m)).\end{aligned} \tag{1}$$

Similarly, we encode every shot in a shot sequence $(s_1, s_2, \cdots, s_n)$ into a feature vector v_i^S $(i = 1, 2, \cdots, n)$ with visual encoder *VE* and then concatenate them into a sequence of shot embeddings

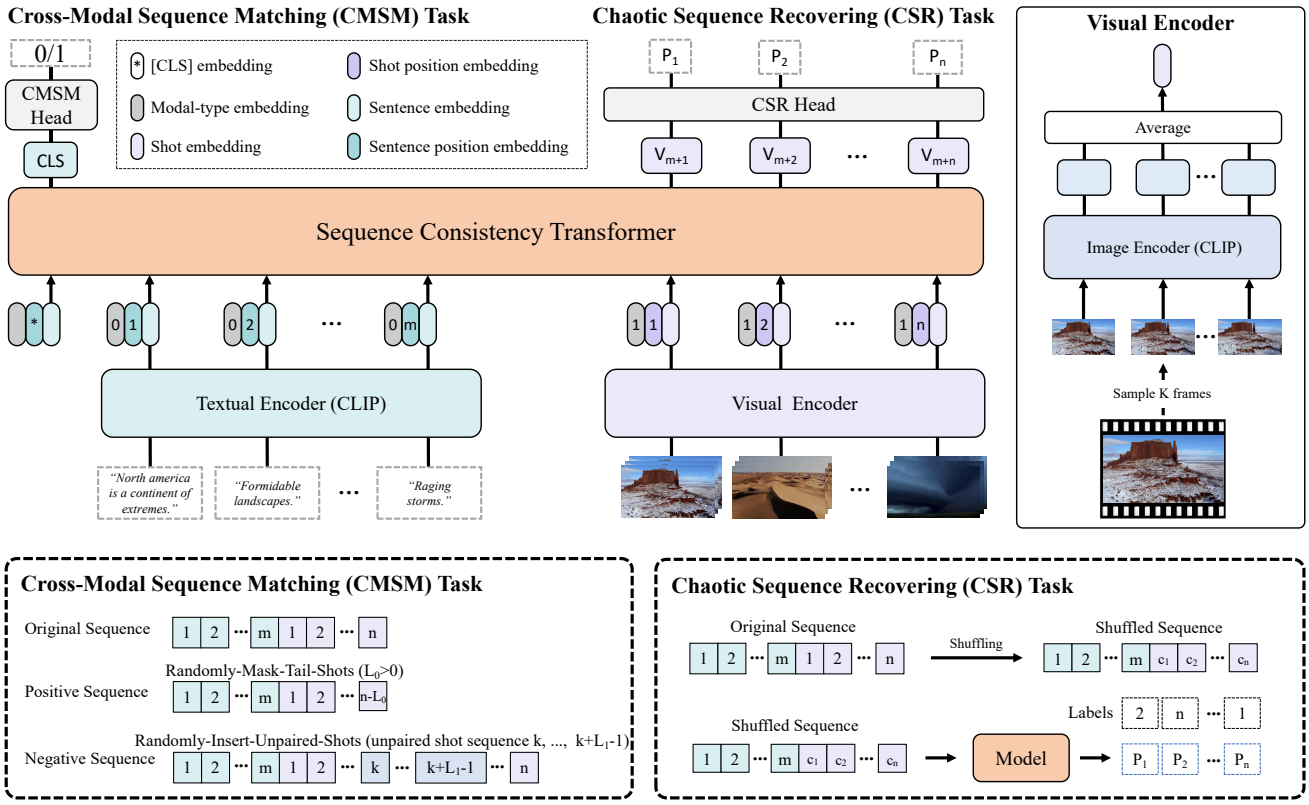

Figure 2: A schematic illustration of our proposed RATV model. RATV learns text-video alignment in sequence-level for video montage generation. Our novel Cross-Modal Sequence Matching (CMSM) and Chaotic Sequence Recovering (CSR) tasks are two key components for text-video alignment, which are only considered in the training phase.

$(v_1^S, v_2^S, \cdots, v_n^S)$. Specifically, for each shot in the sequence, we sample K frames from the original shot and encode every frame into feature vector $v_{i,j}^F$ $(i = 1, 2, \cdots, n, j = 1, 2, \cdots, K)$ with the image encoder VE. We simply average the feature vectors of K frames and take it as the feature vector of the corresponding shot. Formally, the process of extracting shot feature vectors is:

$$\begin{aligned} v_i^S &= Avg(v_{i,1}^F, v_{i,2}^F, \cdots, v_{i,K}^F) \\ &= Avg(VE(f_{i,1}), VE(f_{i,2}), \cdots, VE(f_{i,K}))), \end{aligned} \tag{2}$$

where Avg(·) is the average function and $f_{i,j}$ is the j-th frame of the i-th shot. On the top of the sequences of text embeddings and shot embeddings, we propose a novel sequence consistency transformer ST to learn the text-video joint representation in sequence-level. Note that our proposed sequence consistency transformer supports one or multiple shots as input for flexibility (i.e., $n \geq 0$ and n is an integer). Concretely, following previous works [4, 13], different token type embeddings are firstly added to each embedding in these two sequence respectively to discriminate text embedding and shot embedding. We then concatenate these two sequences into one:

$$\begin{aligned} (\tilde{v}_1, \tilde{v}_2, \cdots, \tilde{v}_{m+n}) = (&v_1^T + x, v_2^T + x, \cdots, v_m^T + x, \\ &v_1^S + \tilde{x}, v_2^S + \tilde{x}, \cdots, v_n^S + \tilde{x}), \end{aligned} \tag{3}$$

where x and $\tilde{x}$ denote the token type embedding of text and shot, respectively. Furthermore, the embedding of special **[CLS]** token is appended to the start of the concatenated sequence. The sequence consistency transformer ST is defined as:

$$(v_c, v_1, v_2, \cdots, v_{m+n}) = ST(\tilde{v}_c, \tilde{v}_1, \tilde{v}_2, \cdots, \tilde{v}_{m+n}), \tag{4}$$

where $\tilde{v}_c$ denotes the embedding of **[CLS]** token. To encourage the ST to learn the text-video joint representation in sequence-level well, we devise Cross-Modal Sequence Matching (CMSM) Task and Chaotic Sequence Recovering (CSR) tasks for model training.

4.2 Cross-Modal Sequence Matching (CMSM)

Similar to the Image-Text Matching (ITM) task [22], we devise a novel Cross-Modal Sequence Matching (CMSM) Task to encourage our RATV model to learn text-video alignment in sequence-level. Specifically, we extract the representation of **[CLS]** token as the joint representation of the input sequence, and then feed it into a FC layer with a sigmoid function to predict a score between 0 and 1, which indicates the probability of that the input text and shot sequences are matching. Differently, we generate the positive and negative samples with two novel mechanisms. To create the positive samples, we introduce a randomly-mask-tail-shots mechanism. Concretely, we sample a script-video pair at each step during training, where the script and paired video are denoted as sentence

sequence $(\tilde{t}_1, \tilde{t}_2, \cdots, \tilde{t}_m)$ and shot sequence $(s_1, s_2, \cdots, s_n)$, respectively. We then randomly decide the number of shots to mask and mask the desired number of shots from the tail of the shot sequence to generate a positive sample. Formally, the randomly-mask-tail-shots mechanism is:

$$(s_1, s_2, \cdots, s_{n-L_0}) = R((s_1, s_2, \cdots, s_n), L_0), \tag{5}$$

where R$(\cdot,\cdot)$ denotes the randomly-mask-tail-shots function and L_0 is the randomly decided number of shots to mask ($0 \le L_0 < n$). Moreover, we also introduce a randomly-insert-unpaired-shots mechanism to create the negative samples. For a text-video pair, we randomly select another shot sequence $(\hat{s}_1, \hat{s}_2, \cdots, \hat{s}_{\hat{n}})$ from the training set, which is unpaired with the script. We randomly select one of the sub-sequences of unpaired shot sequence and insert it into the paired shot sequence in a random position. Formally, we randomly decide the length L_1 and start position k ($1 \le L_1 \le \hat{n}, 1 \le k \le \hat{n}, k + L_1 \le \hat{n} + 1$), and obtain the sub-sequence with them:

$$(\hat{s}_k, \hat{s}_{k+1}, \cdots, \hat{s}_{k+L_1-1}) = G((\hat{s}_1, \hat{s}_2, \cdots, \hat{s}_{\hat{n}}), k, L_1), \tag{6}$$

where $G(\cdot,\cdot,\cdot)$ denotes the function of obtaining sub-sequence from the input sequence. The randomly-insert-unpaired-shots mechanism is formally defined as:

$$\begin{aligned}&(s_1, s_2, \cdots, s_{n-L_2}, \hat{s}_k, \hat{s}_{k+1}, \cdots, \hat{s}_{k+L_1-1}, s_{n-L_2+1}, \cdots, s_n)\\ &= I((s_1, s_2, \cdots, s_n), (\hat{s}_k, \hat{s}_{k+1}, \cdots, \hat{s}_{k+L_1-1}), L_2),\end{aligned} \tag{7}$$

where $I(\cdot,\cdot,\cdot)$ denotes the random insert shots function and L_2 is the randomly decided position to insert the sub-sequence ($0 \le L_2 \le n$). Since we create positive and negative samples by manipulating the shot sequence, the CMSM task encourages the model to learn a well-aligned text-video joint representation space in sequence level.

During training, we randomly decide the input sample is positive or negative with probability 0.5, and give the corresponding binary label $y \in \{0, 1\}$ (i.e., 0 is negative and 1 is positive). We then apply the randomly-mask-tail-shots or randomly-insert-unpaired-shots mechanism on the sample and feed it into the sequence consistency transformer *ST* to obtain the matching score *M*. The binary cross-entropy loss is taken on board for optimization:

$$\mathcal{L}_V = \mathbb{E}\big[-(y \log M + (1-y)\log(1-M))\big]. \tag{8}$$

4.3 Chaotic Sequence Recovering (CSR)

We further devise another novel pretext task termed Chaotic Sequence Recovering (CSR) for text-video joint representation learning. The CSR task aims to recover the original order of the shots in the chaotic shot sequence with the paired text. Formally, given the input sentence sequence $(\tilde{t}_1, \tilde{t}_1, \cdots, \tilde{t}_m)$ and its paired shot sequence $(s_1, s_2, \cdots, s_n)$, we first shuffle the shot sequence:

$$(c_1, c_2, \cdots, c_n) = S(s_1, s_2, \cdots, s_n), \tag{9}$$

where $S(\cdot)$ denotes the function of shuffling the input sequence, $c_i \in \{s_1, s_2, \cdots, s_n\}$ and $c_i \ne c_j$ ($\forall i, j = 1, 2, \cdots, n, i \ne j$). Meanwhile, we can obtain the original positions of all shots in the shuffled sequence, which are defined as $(\tilde{p}_1, \tilde{p}_2, \cdots, \tilde{p}_n)$. After feature extraction, the shuffled sequence is transformed to a sequence of shot embeddings, which is then concatenated with the sequence of sentence embeddings and embedding of **[CLS]** token. According to Eq. (3), we have the concatenated sequence of embeddings $(\tilde{v}_c, \tilde{v}_1^T, \tilde{v}_2^T, \cdots, \tilde{v}_m^T, \tilde{v}_1^S, \tilde{v}_2^S, \cdots, \tilde{v}_n^S)$. Note that we use different symbols to distinguish the sentence and shot embeddings here for easier understanding. We finally feed the sequence into *ST* and employ a FC layer with a softmax function on every outputted shot embedding to predict the score of i-th position:

$$p_i = SM(FC(v_i^S)), i = 1, 2, \cdots, n, \tag{10}$$

where $SM(\cdot)$ is the softmax function, $FC(\cdot)$ is the FC layer. Note that the length of shot sequence n can vary for different input and the output dimension of FC thus can not be defined. To address this, we follow previous transformer based methods to set the max length L_{max}. If the length of input shot sequence is less than L_{max}, we pad the sequence so that its length is equal to L_{max}, and mask the padding tokens when the sequence is fed into the ST. If the length of input shot sequence is greater than L_{max}, we truncate the sequence so that its length is equal to L_{max}. With this setting, the FC layer is defined for L_{max}-class classification task (i.e., $p_i \in \mathbb{R}^{L_{max}}$). The CSR task minimizes the cross-entropy loss with $(\tilde{p}_1, \tilde{p}_2, \cdots, \tilde{p}_n)$ as ground-truth labels:

$$\mathcal{L}_R = \mathbb{E}\big[\sum_{i=1}^{n} CE(p_i, \tilde{p}_i)\big], \tag{11}$$

where $CE(\cdot,\cdot)$ denotes the cross-entropy function. Note that the CSR task is parallel with the CMSM task, i.e., our model is trained with these two tasks together at each step during training. The overall loss can be defined as:

$$\mathcal{L}_{RATV} = \mathcal{L}_V + \lambda \mathcal{L}_R, \tag{12}$$

where λ is the hyperparameter to balance the two losses.

4.4 Inference

In the inference phase, our proposed RATV generates video montages from given text scripts by retrieving shots from the set of candidates $\mathcal{S}$ iteratively. Formally, given a query text t, we encode it into sentence embedding sequence $(v_1^T, v_2^T, \cdots, v_m^T)$ according to Eq. (1). We then retrieve a shot from candidates at a time, resulting in retrieved shot sequence $(\tilde{s}_1, \tilde{s}_2, \cdots, \tilde{s}_{t-1})$ at step t (the sequence is empty when $t = 1$). For shot s_c in leftover candidates (i.e., $s_c \in \mathcal{S}$ and $s_c \notin \{\tilde{s}_i | i = 1, 2, \cdots, t-1\}$), we calculate the scores in both instance-level and sequence-level, and add them as the ensemble score for shot s_c. Concretely, we get the shot embedding v_c^S with Eq. (2), and calculate the score in instance-level to measure whether the shot s_c and input text t are matching in semantics:

$$I_c = cos(Avg((v_1^T, v_2^T, \cdots, v_m^T)), v_c^S), \tag{13}$$

where $\cos(u, v) = u^T v / \|u\| \|v\|$ denotes the cosine similarity between the two vectors u and v. In addition, we append the shot s_c to the end of retrieved shot sequence and feed it with input text to sequence consistency transformer *ST* to get the output matching score M_c as the score in sequence-level. The ensemble score is $E_c = M_c + \lambda_s I_c$, where λ_s is the weight hyperparameter. The shot with highest ensemble score is taken as the retrieved shot $\tilde{s}_t$ and appended to the end of retrieved shot sequence. We set a threshold hyperparameter ϵ for automatically terminating the inference process. That is, when the highest ensemble score $E_h \le \epsilon$, our RATV terminates the process and returns the retrieved shot sequence as the generated video montage. Crucially, our proposed RATV learns

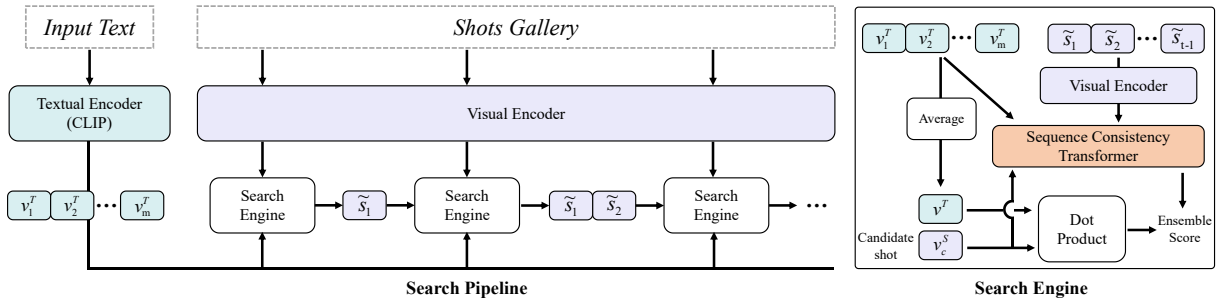

Figure 3: Th inference pipeline of our proposed RATV. At each step, RATV retrieves one shot from shot gallery with search engine. Note that we adopt the ensemble technique to exploit the similarity between input text and candidates in both instance-level and sequence-level for text-to-shot retrieval.

Table 2: Quantitative results on the VSPD dataset. † denotes directly using the pre-trained model for retrieval without fine-tuning. ↓ means that lower is better while ↑ means the opposite. CLIP-A denotes that we split the input text into sentences and encode each sentence with CLIP, and then use the average of the sentence embeddings for retrieval.

Method	Automated Metrics				User Study	
	IoU ↑	UMS ↓	SMS ↑	CS ↑	Semantic ↑	Coherence ↑
VSE [9]	0.017	4.925	0.013	0.194	2.47	3.34
VSE++ [7]	0.020	6.141	0.007	0.040	2.14	3.08
MIL-NCE† [23]	0.011	5.431	0.002	0.116	–	–
MIL-NCE	0.023	6.111	0.009	0.035	2.03	2.74
Frozen-in-Time† [3]	0.077	4.723	0.054	0.143	–	–
Frozen-in-Time	0.085	5.213	0.066	0.178	2.95	3.29
CLIP† [25]	0.072	5.026	0.034	0.073	–	–
CLIP-A†	0.104	4.669	0.072	0.095	3.09	3.24
Write-A-Video† [32]	0.104	4.669	0.079	0.097	3.09	3.24
Transcript-to-Video† [34]	0.096	4.621	0.064	0.124	2.69	3.16
RATV (ours)	**0.144**	**3.393**	**0.090**	**0.685**	**3.40**	**3.76**

video-text joint representation in sequence-level with carefully designed two mechanisms in the CMSM task. Therefore, the matching score measures whether the candidate shot s_c is matching with the input text and retrieved shot sequence in semantics and coherence at the same time. As a result, our RATV can directly return the retrieved shots in sequence of retrieval order without rearranging them. The illustration of inference process is shown in Figure 3.

5 EXPERIMENTS

5.1 Experimental Setup

We adopt four metrics for quantitative evaluation. **(1) Intersection over Union (IoU).** Recall is considered as an important metric for retrieval task in previous works, which measures whether the target in the top-K retrieved results (resulting in Recall@K). Note that the target in retrieval task commonly contains a single instance, but the target in video montage generation contains multiple shots. Therefore, we adopt IoU instead of recall, which is applied on the retrieved shots and the (ground truth) shots paired with each query text. **(2) Unmatching Score (UMS).** This metric is defined as follows: for all shots in a generated video, we find those shots not in the (ground truth) video paired with the input/query text, then calculate the dissimilarity $dsim = 1 - sim$ for each of the found shots, where sim is the similarity between shot and the input text calculated by CLIP. We finally add all dissimilarities together as the UMS. Note that ResNet50×4 (not used by any competitors) from CLIP is used to calculate the similarity. **(3) Sequence Matching Score (SMS).** This metric is used to evaluate the overall quality of the generated video montage. Specifically, given a query text t, we define the paired shot sequence and retrieved shot sequence as $\{s_1, s_2, \cdots, s_n\}$ and $\{\tilde{s}_1, \tilde{s}_2, \cdots, \tilde{s}_{\tilde{n}}\}$, respectively. The SMS is:

$$SMS = \frac{1}{n} \sum_{i=1}^{min(n,\tilde{n})} \mathbb{B}(s_i = \tilde{s}_i), \tag{14}$$

where $\mathbb{B}(\cdot)$ is the indicator function (indicating whether two shots are the same). **(4) Consistency Score (CS).** This metric is used to evaluate the consistency of the generated video given the input text. To obtain this score for arbitrary video-text pair, a binary classifier is needed to distinguish whether all shots in the video are both temporally consecutive and consistent with the context of the text. In this work, we train the binary classifier over a large set of video-text pairs, where the positive pairs are directly obtained from the training set but the negative pairs are generated by shuffling the shots in the video from each positive pair.

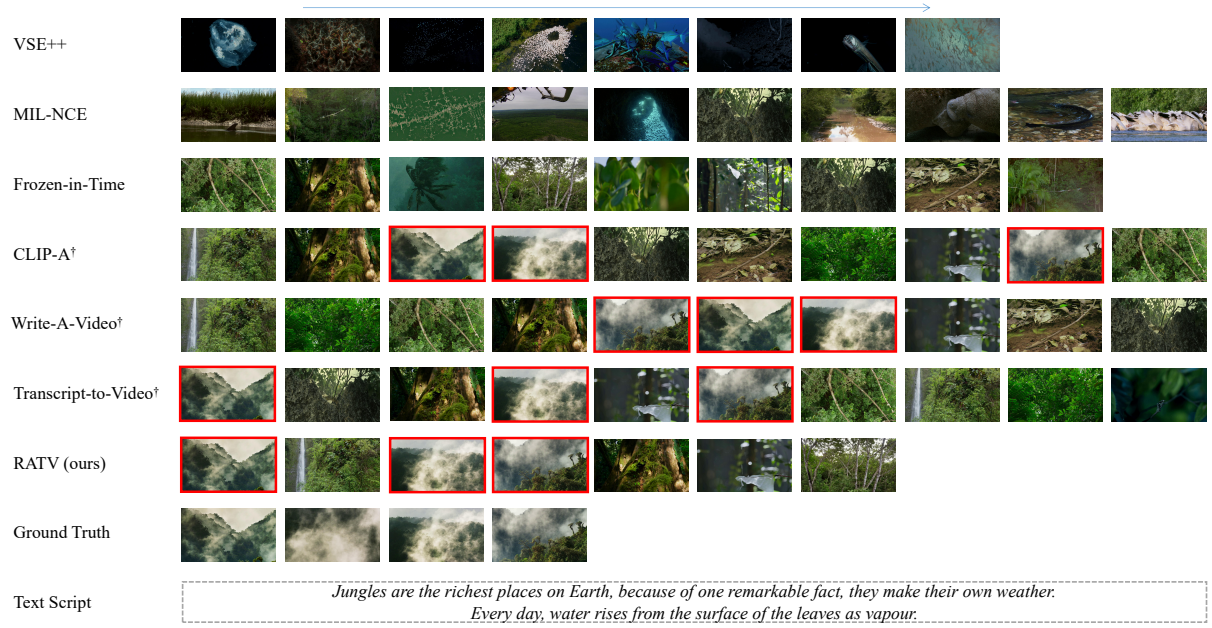

Figure 4: Qualitative results for video montage generation on the VSPD dataset. Every image here denotes a shot, and the image in red box means that the shot falls in the ground-truth video. Each row (except the last two rows) shows the shots in the video generated by a compared method. All the videos are generated according to the text script in the last row. Note that the number of shots of some generated videos is less than that of the other videos due to the earlier termination by threshold.

5.2 Quantitative Results

The quantitative results are shown in Table 2. It can be seen that: **(1)** Our RATV outperforms all competitors with large margins on all metrics, indicating that our method can retrieve shots from candidates more precisely and assemble them into final video montage with better coherence. **(2)** CLIP-A leads to significant improvements over CLIP [25], which shows that CLIP still suffers from large information loss when it directly encodes complicated texts of multiple sentences. This is mainly due to the fact that only short/brief texts are used for pre-training CLIP. **(3)** Write-A-Video and Transcript-to-Video improve the results over CLIP-A. However, they lead to limited gains over CLIP-A because they ignore the context of input text when arranging the retrieved shots. **(4)** Among the three methods (i.e., Write-A-Video, Transcript-to-Video, and our RATV) that consider the coherence during video montage generation, our RATV performs the best because of expanding from instance-level modeling to sequence-level modeling.

We further conduct user study to evaluate the quality of generated videos under human perception. Specifically, with each method, we randomly select 50 text scripts from the test set and generate videos according to these scripts. We then invite volunteers to score all generated videos (i.e., 400 videos in total for all methods) according to the semantic consistency and temporal coherence, which are shortened as 'Semantic' and 'Coherence', respectively. In this paper, ten independent volunteers are asked to score the videos from 1 to 5 (higher is better), and the chosen text scripts for video montage generation are different for different volunteers. Table 2 shows the user study results averaged over all ten volunteers. As expected, our RATV outperforms all competitors on both semantic consistency and temporal coherence. Interestingly, Write-A-Video and Transcript-to-Video are even inferior to CLIP-A and Frozen-in-Time under human perception, showing that their assembly techniques (using pre-defined rules or coherence classifier) are not that effective without considering the context of input text.

5.3 Qualitative Results

The qualitative results on the VSPD dataset are shown in Figure 4. Among the competitors, VSE++ [7], MIL-NCE [23], and Frozen-in-Time [3] are trained on the VSPD dataset. We can observe that: **(1)** Compared with the other competitors, the shots retrieved by CLIP-based methods can express the text script more precisely, which also contain the 3 shots from the ground-truth video. **(2)** Based on CLIP, Write-A-Video and Transcript-to-Video can both retrieve the shots that are well aligned with the text script. Importantly, the shots in the generated videos are more consecutive due to the pre-defined rules and the coherence classifier (as compared with CLIP-A). However, some shots in the generated videos are still in an unreasonable order or not so relevant to the text script, because the context of the text script is ignored when arranging the retrieved shots. **(3)** Our RATV can generate the video that precisely expresses the whole text script. Importantly, our model can consider the semantic alignment and temporal coherence in the meantime during generation. The generated video thus does not contain any shots irrelevant to the text script. Although our model happens to miss the second shot of the ground-truth video, this shot is not found by any competitors. More results are given in Appendix.

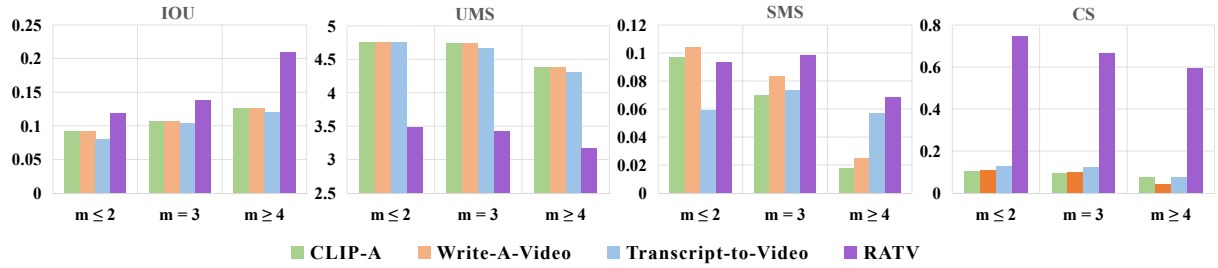

Figure 5: Comparative results on three subsets of the test set. These test subsets are obtained by splitting the test set according to the number of sentences m in each input text script.

Table 3: Ablation study results for our full RATV model. Base denotes the RATV model trained with the ITM task. RM and RI denote the randomly-mask-tail-shots and randomly-insert-unpaired-shots mechanisms, respectively. The full CMSM task is equal to RM+RI.

Method	IoU ↑	UMS ↓	SMS ↑	CS ↑
Base	0.093	4.741	0.051	0.079
Base+RM	0.109	4.513	0.056	0.131
Base+RI	0.096	4.373	0.078	0.256
Base+CMSM	0.142	3.535	0.083	0.645
Base+CSR	0.099	4.608	0.065	0.136
Base+RM+CSR	0.116	4.235	0.070	0.164
Base+RI+CSR	0.099	4.282	0.087	0.337
RATV (ours)	**0.144**	**3.393**	**0.090**	**0.685**

5.4 Ablation Study

We conduct ablation study to show the contribution of our proposed CMSM and CSR tasks. We firstly adopt simple ITM task [4] to train our proposed RATV model, which is denoted as Base. To further explore the contribution of the two novel tasks, we then gradually add randomly-mask-tail-shots mechanism (RM), randomly-insert-unpaired-shots mechanism (RI) and CSR task on the top of Base. When we adopt RM (RI) along, we follow the strategy of ITM task to create negative (positive) samples. Our full RATV is actually Base+RI+RM+CSR, which is trained with full CMSM (i.e., RI+RM) and CSR tasks. The results of ablation study are shown in Table 3.

We have the following observations: **(1)** The RM and RI lead to improvements on all metrics, indicating that both mechanisms are beneficial to text-video joint representation learning for video montage generation **(2)** The combination of RM and RI in the CMSM task yields significant improvements on all metrics over adopting them alone, which shows the complementarity of the two mechanisms. Crucially, our RATV trained with the CMSM task has now outperformed CLIP-A (see Base+CMSM in in Table 3 vs. CLIP-A† in Table 2), which directly verifies the effectiveness of our proposed RATV framework for video montage generation. **(3)** When combined with either of the two mechanisms, the CSR task leads to further improvements on all metrics, especially on the SMS and CS. This still holds for CMSM+CSR (i.e., our full RATV).

5.5 Further Evaluation

Number of Input Sentences. We make further comparison under different test conditions. Concretely, we split the test set into three subsets according to the number of sentences $\mathbf{m}$ ($\mathbf{m} \leq 2$, $\mathbf{m} = 3$, $\mathbf{m} \geq 4$) in each input text script, and report comparative results on each test subset in Figure 5. We can observe that our RATV consistently outperforms all competitors on three (i.e., IOU, UMS, and CS) out of four metrics over all test subsets. When it comes to SMS, our RATV is slightly inferior to CLIP-A and Write-A-Video when $\mathbf{m} \leq 2$. However, as $\mathbf{m}$ increases, the performance of CLIP-A and Write-A-Video in terms of SMS decreases sharply, thus becoming worse than that of our RATV. Overall, the superior performance of RATV with $\mathbf{m} > 2$ shows that our RATV is indeed effective in video montage generation given complicated text script of multiple sentences.

Wild Text Scripts. Although the VSPD dataset contains the videos from documentaries, the goal of our RATV is not to generate documentaries only. In contrast, it devotes to generating videos on various themes which mainly depend on the input text scripts. To demonstrate this, we deploy our RATV to generate travel vlog with narrated text scripts, MV for music with lyrics, and background video with poem, whose results are given in Appendix due to the space constraint. The results suggest that our RATV has a good ability of generating videos on various themes with wild text scripts.

6 CONCLUSION

In this work, we have proposed a novel framework termed RATV to automatically generate video montages by retrieving and assembling shots with arbitrary text scripts. Due to the novel Cross-Modal Sequence Matching (CMSM) and Chaotic Sequence Recovering (CSR) tasks, our proposed RATV can effectively learn the text-video joint representation in sequence-level and also the coherence of shot sequence. To our best knowledge, our RATV is the first model for video montage generation based on text-to-sequence retrieval, which can generate video montages more consistent with the input text scripts. To fill the gap in dataset construction for video montage generation, we create a new dataset called VSPD, which contains thousands of diverse video-script pairs. Extensive experiments on the VSPD dataset demonstrate the effectiveness of our RATV.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (61976220). Zhiwu Lu is the corresponding author.

REFERENCES

[1] Gulrukh Ahanger and Thomas D. C. Little. 1998. Automatic Composition Techniques for Video Production. *IEEE Transactions on Knowledge and Data Engineering* 10, 6 (1998), 967–987.

[2] Alex Andonian, Camilo Fosco, Mathew Monfort, Allen Lee, Rogério Feris, Carl Vondrick, and Aude Oliva. 2020. We Have So Much in Common: Modeling Semantic Relational Set Abstractions in Videos. In *ECCV*, Vol. 12363. 18–34.

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. 2021. Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval. In *ICCV*. 1708–1718.

[4] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: UNiversal Image-TExt Representation Learning. In *ECCV*, Vol. 12375. 104–120.

[5] Tat-Seng Chua and Li-Qun Ruan. 1995. A Video Retrieval and Sequencing System. *ACM Transactions on Information Systems* 13, 4 (1995), 373–407.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. 4171–4186.

[7] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. VSE++: Improving Visual-Semantic Embeddings with Hard Negatives. In *BMVC*. 12.

[8] David F. Fouhey, Weicheng Kuo, Alexei A. Efros, and Jitendra Malik. 2018. From Lifestyle Vlogs to Everyday Interactions. In *CVPR*. 4991–5000.

[9] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomás Mikolov. 2013. DeViSE: A Deep Visual-Semantic Embedding Model. In *NeurIPS*. 2121–2129.

[10] De-An Huang, Shyamal Buch, Lucio M. Dery, Animesh Garg, Li Fei-Fei, and Juan Carlos Niebles. 2018. Finding "It": Weakly-Supervised Reference-Aware Visual Grounding in Instructional Videos. In *CVPR*. 5948–5957.

[11] Yuqi Huo, Manli Zhang, Guangzhen Liu, Haoyu Lu, Yizhao Gao, Guoxing Yang, Jingyuan Wen, Heng Zhang, Baogui Xu, Weihao Zheng, et al. 2021. WenLan: Bridging vision and language by large-scale multi-modal pre-training. *arXiv preprint arXiv:2103.06561* (2021). https://arxiv.org/abs/2103.06561

[12] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *ICML*. 4904–4916.

[13] Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. In *ICML*, Vol. 139. 5583–5594.

[14] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-Captioning Events in Videos. In *ICCV*. 706–715.

[15] Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. 2017. Computational video editing for dialogue-driven scenes. *ACM Transactions on Graphics* 36, 4 (2017), 130:1–130:14.

[16] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. 2021. Less is More: ClipBERT for Video-and-Language Learning via Sparse Sampling. *CVPR* (2021), 7331–7341.

[17] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L. Berg. 2018. TVQA: Localized, Compositional Video Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1369–1379.

[18] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 2020. Unicoder-VL: A Universal Encoder for Vision and Language by Cross-Modal Pre-Training. In *AAAI*. 11336–11344.

[19] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. 2020. HERO: Hierarchical encoder for video+ language omni-representation pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2046–2065.

[20] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. VisualBERT: A Simple and Performant Baseline for Vision and Language. *arXiv preprint arXiv:1908.03557* (2019). https://arxiv.org/abs/1908.03557

[21] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. In *ECCV*, Vol. 12375. 121–137.

[22] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NeurIPS*. 13–23.

[23] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2020. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*.

[24] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*. 2630–2640.

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*. 8748–8763.

[26] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. 2012. A database for fine grained activity detection of cooking activities. In *CVPR*.

[27] Edward Yu-Te Shen, Henry Lieberman, and Glorianna Davenport. 2009. What's next?: emergent storytelling from video collection. In *SIGCHI Conference on Human Factors in Computing Systems*. 809–818.

[28] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. In *ICLR*. https://openreview.net/forum?id=SygXPaEYvH

[29] Hao Tan and Mohit Bansal. 2019. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 5099–5110.

[30] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. MovieQA: Understanding Stories in Movies through Question-Answering. In *CVPR*. 4631–4640.

[31] Anh Truong, Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 2016. QuickCut: An Interactive Tool for Editing Narrated Video. In *Annual Symposium on User Interface Software and Technology (UIST)*. 497–507.

[32] Miao Wang, Guo-Wei Yang, Shi-Min Hu, Shing-Tung Yau, and Ariel Shamir. 2019. Write-a-video: computational video montage from themed text. *ACM Transactions on Graphics* 38, 6 (2019), 177:1–177:13.

[33] Xiaohan Wang, Linchao Zhu, and Yi Yang. 2021. T2VLAD: Global-Local Sequence Alignment for Text-Video Retrieval. In *CVPR*. 5079–5088.

[34] Yu Xiong, Fabian Caba Heilbron, and Dahua Lin. 2021. Transcript to Video: Efficient Clip Sequencing from Texts. *arXiv preprint arXiv:2107.11851* (2021). https://arxiv.org/abs/2107.11851

[35] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*. 5288–5296.

[36] Youngjae Yu, Jongseok Kim, and Gunhee Kim. 2018. A joint sequence fusion model for video question answering and retrieval. In *ECCV*. 487–503.

[37] Linchao Zhu and Yi Yang. 2020. ActBERT: Learning global-local video-text representations. In *CVPR*. 8743–8752.