Vol.2018-CVIM-211 No.7
2018/3/1

Finding Video Parts with Natural Language

Mayu Otani[1,a)] Yuta Nakashima[2,b)] Esa Rahtu[3,c)] Janne Heikkilä[4,d)]

Abstract: The increasing number of videos have motivated the development of content-based video retrieval (CBVR) methods, which search for videos whose content is relevant to a query. Since most existing datasets for this task provide short video clips capturing a single activity, previous methods have focused on short video clips. However, the majority of real-world videos are more lengthy and edited. Such videos may consist of multiple video clips and may include various content within a video, thus previous methods may fail with real-world videos. In this paper, we propose a new video retrieval task which aims to handle such multi-clip videos. The task is to find query-relevant parts from a video consisting of multiple clips, which we call fine-grained video retrieval (FGVR). For this new task, we build datasets from existing video-description datasets. We synthesize multi-clip video and query pairs by augmenting video-description datasets, which results in large-scale training and evaluation data. We introduce several deep neural network-based approaches as baselines and a training scheme using the synthesized video and query pairs. We investigate the baselines on two datasets built from YouTube and movie datasets, respectively, and present preliminary results.

1. Introduction

The tremendous growth of online videos has increased demands for content-based video retrieval (CBVR) that takes a natural language query as input and retrieves videos relevant to the query from a huge database. To retrieve videos, the relevance between a query and a video clip is required. A major approach is to develop a deep neural network that models relevance between a query and a video clip as in [12], [26], [28], [30]. Most of these existing methods assume that video clips in a database are trimmed in such a way that the resulting video has consistent content, such as a single activity or an event. However, this problem setting is not always valid in practice. Many real-world videos including, YouTube videos, TV shows, or movies, are lengthy and consist of multiple video clips which are not limited to a single action or an event. As most existing methods are designed to produce a relevance score aggregated over the whole content of a video clip, this may exhibit drawbacks when they are applied to such videos.

Observing the limitations, we propose a new video retrieval task to find video parts which match a natural language query (Figure 1). In this paper, we call this task as fine-grained video retrieval (FGVR). In contrast to existing CBVR tasks, FGVR aims to handle more complex videos which may have multiple clips and varying content within a video. This problem setting is more alike to videos in the wild. Thus, we expect developing methods for this task contributes to a wide range of applications.

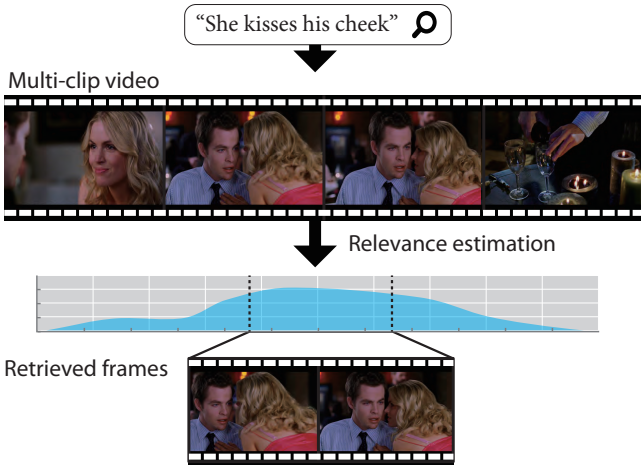

Fig. 1 Given a natural language query, fine-grained video retrieval finds video frames which the query describes (yellow borders). An input video consists of multiple video clips.

Since this is a new task of video retrieval, there is no dataset for training and testing FGVR methods. Making a FGVR dataset that is large enough to develop recent deep network models will require immense amount of human intervention, thus we exploit existing datasets. Previous work on CBVR utilizes large-scale video-description datasets as benchmarks for their task [2], [14], [16], [27]. As videos in video-description datasets are trimmed to exclude scenes irrelevant to their descriptions, these datasets cannot be used for our FGVR task. Instead of using videos in these datasets as is, we make video and query pairs for FGVR from the existing datasets. We concatenate several videos and use one of the descriptions annotated to these videos as a query sentence. By this data generation, we can obtain a number of videos, which have a query sentence and corresponding

1 Nara Institute of Science and Technology
2 Osaka University
3 Tampere University of Technology
4 University of Oulu
a) otani.mayu.ob9@is.naist.jp
b) n-yuta@ids.osaka-u.ac.jp
c) esa.rahtu@tut.fi
d) janne.heikkila@ee.oulu.fi

frame-level annotation of ground truth labels. As our data generation scheme can be applied to any video-description datasets, large-scale benchmarks can be built. In this paper, we present two FGVR benchmarks built from two video-description datasets: one with YouTube videos, and the other with movies. In order to promote the FGVR research, the compiled datasets will be released in public.

One possible approach for the FGVR task is to divide an input video into shorter video clips and rank the video clips based on query-relevance which can be computed by existing video retrieval methods, such as [17], [26]. Another way is to compute query-relevance for every frame. In this approach, FGVR can be done without temporal video segmentation. In this paper, we explore both of these approaches. We implement deep neural network-based methods and present preliminary results. We will also provide implementations of the FGVR methods as baselines on this task.

The contributions of this paper are as follows:

- We propose a new task of video retrieval, *i.e.*, FGVR. This task assumes that a video consists of multiple video clips, which may contain different object, actions, or scenes. This assumption is more practical because most videos (online videos, broadcast programs, and movies) are edited and consist of multiple video clips.
- We present several neural network-based baseline methods for FGVR. We also propose a training scheme of models of the baseline methods. In the experiments, we demonstrate performance of the baseline methods on two datasets, which are built from YouTube videos and movies, respectively. The comparison of their results offers insights into developing FGVR methods.
- We propose to synthesize video and query pairs from existing video-description datasets. Our data generation scheme can build FGVR samples from any video description datasets. This enables large-scale benchmarks of this task, which are essential for developing deep neural network-based methods.

2. Related work

Video retrieval.

Early work addressed content-based video retrieval by detecting predefined concepts in videos, such as objects, actions, and events [18], [25]. A single visual concept may not be enough to spot the desired video, so users are more likely to query with their combinations. Video retrieval by natural language queries provides an intuitive way to make a combination of concepts in a specific context represented in a query. One possible approach is to detect visual concepts and match them to extracted keywords in a natural language query [7], [9], [24], [26], but as they require pre-trained concept detectors, such as [4], [23], [32], types of concepts are limited.

In order to overcome such limitations, Socher *et al.* [19] proposed to train embeddings of images and concept labels into a common space, which can handle unseen concepts. Several approaches in this direction have been proposed on both image retrieval [3], [6] and video retrieval [12], [28], [33]. Xu *et al.* [28] proposed a deep neural network for video retrieval by sentence queries and vise versa. They embed a video clip and a sentence into a common space to compute the similarity between them. Yu *et al.*'s approach [30] learns a similarity metric between a whole video content and a query sentence. In contrast to these methods, FGVR requires to estimate the relevance that varies within a video.

FGVR is closely related to the works by Tapaswi *et al.* [21] and Zhu *et al.* [33], which aim to align book text and movie scenes, as well as query-focused video summarization by Sharghi *et al.* [17]. Both methods search for a part of a long video using a natural language query. The main difference between ours and this task is that ours have less assumptions about target videos and queries. In order to align book chapters or sentences to movie scenes, these approaches [21], [33] assume that the movie comes with closed captions and that the book text and the movie follow a similar timeline. Sharghi *et al.*'s approach [17] only uses a limited set of nouns as queries and does not accept more generic queries, *e.g.*, by natural language. Our task has neither rich metadata of videos nor rough temporal locations.

Video and language datasets.

The research community has provided various datasets involving video and language, such as descriptions [2], [14], [16], [27], [31], titles [20], [31], and concept labels [1], [8], [13]. Chen *et al.* [2] provide 1,967 short YouTube video clips capturing a single activity. Each video in this dataset is annotated with descriptions. Xu *et al.* [27] released a larger-scale dataset, which contains 10K video clips collected with a video search engine and natural language descriptions annotated by crowdsource workers.

There are several datasets for specific domains. The movies are one of such domains, and datasets of movies aligned with descriptions are introduced in [10], [14], [15], [22]. Senina *et al.* [16] collected cooking video clips and their descriptions. Zeng *et al.* collected 18K user-generated videos and their titles. The averaged length of the videos in this dataset are approximately 1.5 minutes (longer than in most other datasets), and they are not edited. The movie datasets in [14], [16] have alignment of description and video frames; therefore, they might be suitable for our task. However, their vocabulary and content in video clips are fairly different from other videos, such as online videos or broadcast programs, since the movies includes fantasy, sci-fi, etc.

3. Fine-grained video retrieval by sentence queries

3.1 Problem statement

In the FGVR task, the input is a video consisting of multiple clips and a natural language query. The goal is to retrieve a subset of frames whose content is semantically relevant to the query (Figure1). Specifically, given a sentence and video frames $V = \{v_1, \dots v_T\}$, where v_t is a visual feature extracted from the t-th frame, FGVR estimate relevance scores $R = \{r_1, \dots, r_T\}$ at each time step to retrieve frames. This task is similar to the video retrieval task for finding videos in a dataset which are relevant to a query. However, video retrieval tasks often implicitly assume that each video in the dataset is short and can be represented by a single query sentence. This assumption is not valid for most videos,

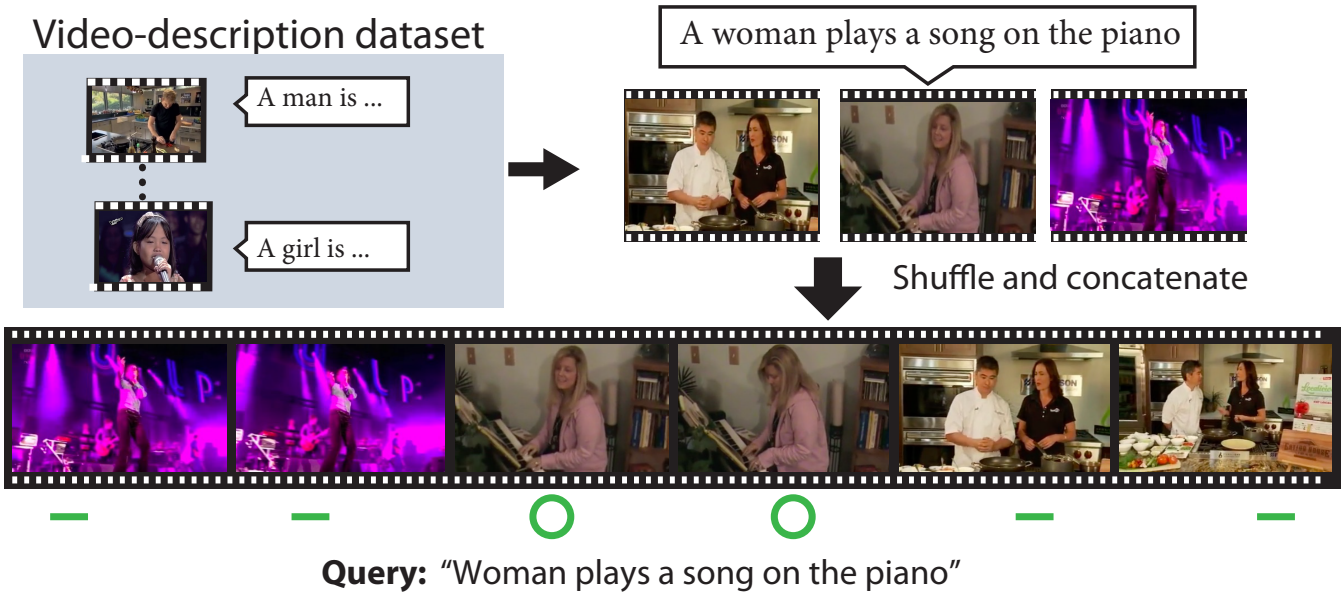

Fig. 2 FGVR samples are generated from a video-description dataset. A video clip associated with a description is combined with randomly sampled videos. This results in a multi-clip video and a sentence which describes only a part of the video.

e.g., broadcast programs, movies, and even YouTube videos. A majority of these videos are lengthy and come with multiple concepts or scenes. The FGVR task relaxes this assumption: only a small part of the target video is relevant to a sentence query.

3.2 Data generation

Since there are no existing datasets for FGVR, we build such by re-utilizing the existing CBVR datasets. To build datasets, we need a number of videos and corresponding query sentences. For FGVR benchmarks, videos must 1) consist of multiple clips, 2) have corresponding query sentence related to only a part of the video, and 3) be annotated with frame-level relevance labels. Since there is no dataset tailored for this task, we make video and query pairs from a large-scale video-description dataset, such as [14], [27].

The data generation using a video-description dataset is illustrated in Figure 2. To get a video consisting of multiple clips, we sample several video clips and their corresponding descriptions. We then choose one of the descriptions as a query sentence and concatenate the video clips in random order. Concatenation of multiple videos results in shot boundaries like most edited videos. The frames in a video clip corresponding to the selected query sentence are labeled as relevant frames, and other frames as irrelevant ones. By doing this, we can generate a number of videos where only a small part of it is relevant to a query sentence. Our data generation scheme can be applied to any dataset which provides videos and descriptions. This enables us to evaluate FGVR methods on diverse videos provided by existing datasets.

3.3 FGVR baselines

For this task, we introduce several baseline methods which utilize deep neural network models that read video frames V and produce relevance scores R. We employed the `pool5` layer of ResNet-50 [4] for feature extraction from video frames.

3.3.1 Clip-level relevance prediction

One possible approach is to divide input video into short video clips and compute relevance scores for each video clip as illustrated in Figure 3 (left). We call this approach a clip-level approach. We test two temporal video segmentation for this approach: Ground truth video segmentation uses clip boundaries in a synthesized videos, and uniform segmentation divides videos with a uniform interval. Similarly to [22], we implement two neural network models that take a sequence of frames $\{v_{t_s}, \ldots, v_{t_e}\}$ in a video clip as input and produce a vector representation x that summarizes the frames.

Frame pooling (F-Pool)

summarizes the frames $\{v_{t_s}, \ldots, v_{t_e}\}$ in a video clip by average pooling. The averaged feature vectors are fed to a fully-connected layer. Therefore, the F-Pool model maps a video clip into the common feature space by

$$\tilde{v} = \sum_{i=t_s}^{t_e} v_i, \tag{1}$$

$$x = \tanh(W_{\mathrm{fp}}\tilde{v} + b_{\mathrm{fp}}), \tag{2}$$

where W_{fp} and b_{fp} are parameters of the fully-connected layer.

Weighted average (WA)

incorporates the soft-attention mechanism [29] in frame pooling. The weights a_i of the frame v_i is computed based on the frame feature and a query sentence y by

$$e_i = w_{\mathrm{a}}^{\mathrm{T}} \tanh(W_{\mathrm{a}}[y, v_i] + b_{\mathrm{a}}), \tag{3}$$

$$a_i = \exp(e_i) / \sum_{j=t_s}^{t_n} \exp(e_j), \tag{4}$$

where w_{a}, W_{a}, and b_{a} are learnable parameters, and $[\cdot, \cdot]$ denotes the concatenation of vectors. The vector t is a text embedding computed with a text encoding model described in Sec. 3.3.3. Using the weights, we obtain a weighted sum of frames and feed it to a fully-connected layer to get a clip representation x as:

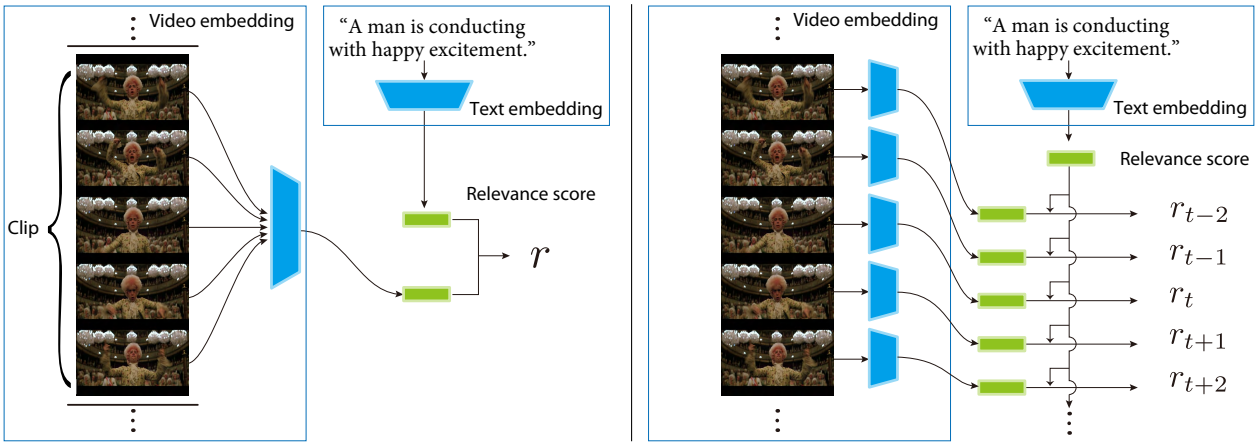

Fig. 3 Illustration of clip-level (left) and frame-level (right) approaches. Green bars for video frames are feature vectors extracted using the ResNet model. For text, the sentence representation is computed by the text encoding model in Section 3.3.3.

$$\tilde{v}_{\text{wa}} = \sum_{i=t_s}^{t_e} a_i v_i, \tag{5}$$

$$x = \tanh(W_{\text{wa}}\tilde{v}_{\text{wa}} + b_{\text{wa}}), \tag{6}$$

where W_{wa} and b_{wa} are parameters of the fully-connected layer.

3.3.2 Frame-level relevance prediction

In the clip-level approach, an input video needs to be segmented beforehand; however, segment boundaries are not always available, and temporal video segmentation itself is still a challenging task. Another direction for this task is to read frames and produce a relevance score at each time step as in Figure 3 (right). For this approach, we implemented three models that encodes video frames to a sequence of vector representations $\{x_1, \ldots, x_T\}$.

Sliding window (SW)

model reads an input frame sequence in the sliding window fashion. At each time step, we perform average pooling over frames within a temporal window and feed its output to a fully-connected layer in the same way as the F-Pool model.

Bidirectional-LSTM (biLSTM)

model utilizes a two-layer LSTM network that reads frames in forward and backward directions as in Figure 3 (right). Hidden states at each time step are concatenated and transformed with a fully-connected layer as:

$$x_t = \tanh(W[h_t^{\text{forward}}, h_t^{\text{backward}}] + b), \tag{7}$$

where h_t^{forward} and h_t^{backward} are hidden states of the forward-LSTM and the backward-LSTM layers for the input frame v_t.

Fully-connected (FC)

model is a variation of the biLSTM model. We remove the temporal connection by replacing the bidirectional LSTM layers with a fully-connected layer. Therefore, the input frame v_t is transformed by

$$h_t = \tanh(W_1 v_t + b_1), \tag{8}$$

$$x_t = \tanh(W_2 h_t + b_2), \tag{9}$$

where W_1, W_2, b_1, and b_2 are parameters of the fully-connected layers. This model estimates relevance scores in a frame-by-frame fashion. Therefore, this model is equivalent to frame-level CBVR.

In the frame-level approaches, relevance score between a query text and frame embedding at each time step is computed.

3.3.3 Text encoding

For text encoding, we employ two models that encode a sequence of words $\{w_1, \ldots, w_N\}$ into a vector representation y, where w_n is a word vector. One is the word pooling-based model (**W-Pool**). Input word vectors are averaged to be transformed with a fully-connected layer as:

$$\tilde{w} = \sum_{n=1}^{N} w_n, \tag{10}$$

$$y = \tanh(W_{\text{wp}}\tilde{w} + b_{\text{wp}}), \tag{11}$$

where W_{wp} and b_{wp} are parameters of the fully-connected layer and y is a sentence representation.

The other is the word LSTM model (**W-LSTM**) that encodes a sequence of word vectors with an LSTM layer, *i.e.*,

$$h_n, c_n = \text{LSTM}(w_n, h_{n-1}, c_{n-1}), \tag{12}$$

where h_n and c_n are a hidden state and a memory cell of the LSTM layer, respectively. We employ the last hidden state as a representation of the sentence in the common feature space.

3.4 Training

The models for videos and sentences are jointly trained so that the query relevance scores of relevant frames are larger than those of others. We compute an averaged score of relevant and irrelevant frames, and update the model to make the difference between the scores larger. During the training, a model is trained by minimizing the loss computed from predicted relevance score R and ground truth label $L = \{l_1, \ldots, l_T\}$ as:

$$\text{Loss}(R, L) = \max(-Score_{\text{pos}} + Score_{\text{neg}} + \mu, 0), \tag{13}$$

$$Score_{\text{pos}} = \frac{1}{N_{\text{pos}}} \sum_{t=1}^{T} l_t r_t, \tag{14}$$

$$Score_{\text{neg}} = \frac{1}{N_{\text{neg}}} \sum_{t=1}^{T} (1 - l_t) r_t, \tag{15}$$

where N_{pos} and N_{neg} are the number of relevant and irrelevant frames in a video, respectively. l_t is a label representing if the

Fig. 4 Query sentence and video pairs, where only keyframes are displayed for videos. The videos are composed by combining multiple video clips from an existing video-description dataset. The examples in the left column are built from MSR-VTT, and in the right from MPII-MD. The red boxes indicate the frames corresponding to the query sentence.

frame is relevant to a query sentence. We set $l_t = 1$ if the frame is relevant, and otherwise 0. The parameter μ is a predefined margin to penalizes the smaller difference between the averaged score of relevant and irrelevant frames than the margin. Models of the clip-level approach do not produce frame-level scores, thus we spread a clip-level score to all frames in the clip.

4. Experiments

We investigated the performance of the baselines described in Section 3.3 on two benchmarks built from MSR Video to Text (MSR-VTT) dataset [27] and the MPII Movie Description dataset (MPII-MD) [15].

4.1 Implementation detail

The model was trained in an end-to-end manner with stochastic gradient decent with the mini-batch size of 100. We used Adam [5] for optimization with the initial learning rate 10^{-3} for MSR-VTT and 10^{-4} for MPII-MD. In all experiments, models were trained for 15 epochs, and we employed a model at the minimum loss on the validation split. During training, we halved the learning rate at the 10th epoch. We adopted gradient clipping with threshold 10.0 and weight decay with weight 0.0005 for MPII-MD. We set the parameter μ for the loss function to 1.0. To extract video frame features, we employed the output of the `pool5` layer of ResNet-50 pretrained on ImageNet [4]. The word embeddings were initialized with word vectors by [11], which we found helpful for training. We set the output size of video and text encoding models to 256. The window size of SW was 5, and input videos were padded with zeros to keep the output length the same as the number of input video frames. Both of the bidirectional LSTM layers in the biLSTM model have 256 units, and the output vectors were fed to the fully connected layers whose output size was also 256.

4.2 Datasets

We tested baselines on the MSR-VTT and the MPII-MD datasets. Examples of generated video and query pairs are displayed in Figure 4. The MSR-VTT dataset includes 6,513 YouTube video clips, and 20 descriptions were annotated for each video clips. MPII Movie Description dataset has 101,046 video clips from movies, and each video clip was annotated with one description. For the MSR-VTT dataset, we used training and test splits provided by the MSR-VTT official web page. For the MPII-MD dataset, we used splits for the LSMDC'16 movie annotation and retrieval task [22]. Word vocabulary is collected from descriptions in the training split. The descriptions were normalized by punctuation removal and lowercasing, then we compiled a vocabulary dictionary by sampling words occurring more than three times in training queries, which results in 8,935 words for the YouTube dataset and 10,066 words for the movie dataset. The videos were down-sampled at 5 fps and rescaled to 244×244. During building the datasets as in Section 3.2, we sampled three video clips and description to get a video query pair as in Figure 2, which were trimmed into 20-100% of its original length.

4.3 Qualitative evaluation

We show some examples of relevance prediction by a model trained for the FGVR task. We show relevance prediction examples by a model trained for the FGVR task. Figure 5 shows an example of frame-level scores for different queries by the biLSTM model. The video shown in Figure 5 was generated from the MSR-VTT dataset. For query sentence (1) and (2), the model predicted high relevance scores for corresponding frames. Interestingly, for query (3), frames of a girl with a microphone got high score as well as the ground truth frames of a crowds. This might be caused by the crowd behind the girl. Within a video clip, we can observe that relevance scores varied according to the content of the frame, *e.g.*, frame without the cooking tools are less relevant than other frames for query (2).

4.4 Quantitative evaluation

We conducted a quantitative evaluation of predicting relevant frames from multi-clipped videos on MSR-VTT and MPII-MD datasets. We generated test videos in the same way as in Section 3.2 from test splits of the datasets. For each test samples, we computed frame-level relevance scores of a video to a query sentence, then evaluated the performance with average precision (AP). We report the mean and the standard deviation (the values in parenthesis) of the AP scores over all test samples in Table 1. To compute AP, the clip-level scores were transformed to frame-level scores by simply spreading the clip-level score to all frames in the clip. The scores obtained by random score prediction are

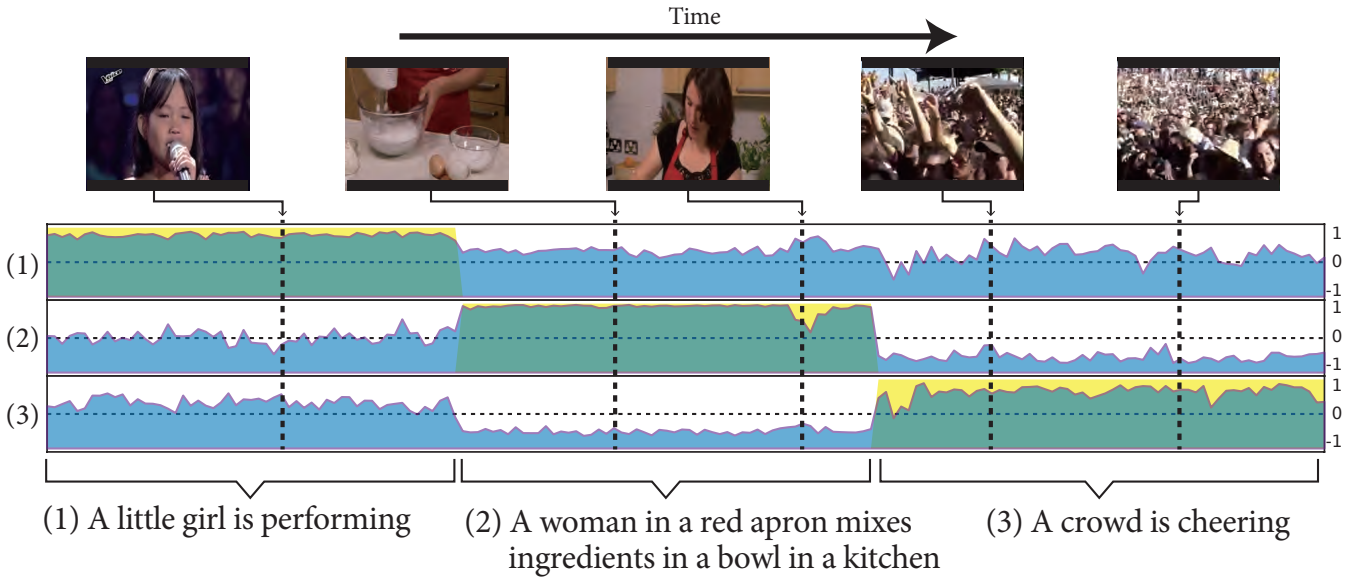

Fig. 5 Relevance scores of a multi-clipped video for different queries. The horizontal axes represents time. From top to bottom: scores for query (1), (2), and (3). Blue represents relevance scores and yellow ground truth relevance labels. Overlapping areas are thus green.

reported in the bottom row.

Overall, cosine similarity performs better than partial order similarity in this task. For clip-level approaches, there are no significant differences between models. Note that these scores with ground truth clip boundaries can be regarded as a sort of upper bounds of the clip-level approaches. We also report scores obtained by uniformly dividing input video into three clips (UNI). These results suggest that the performance of clip-level FGVR methods highly rely on temporal video segmentation.

On the other hand, we can also observe the frame-level approach, which do not require temporal video segmentation, achieving good retrieval performance on the MSR-VTT dataset. This suggests that video segmentation is not a necessity for FGVR. From the comparison between models for the frame-level approach, we can see that incorporating nearby frames improves the performance. This might be because context obtained from other frames is helpful to understand a video content.

For the MPII-MD benchmark, all baselines resulted in lower scores. As this benchmark is more challenging as shown in Figure 4. Many of the sentences often describe complex scenes, for which LSTM may have difficulties in encoding the semantics. Moreover, movies often have dark and low-contrast scenes, which may cause failures in understanding video content.

5. Conclusion

In this paper, we propose a new video retrieval task to find relevant frames to a natural language query. This task is based on the idea that developing video retrieval methods that can handle untrimmed videos consisting of multiple clips is important for real-world applications. For this task, we present a data generation scheme to build large-scale datasets. We also introduce two lines of approaches and implemented baseline models for this task. In our experiments, we present preliminary results on two benchmarks, which are built from a YouTube video dataset and a movie dataset. The benchmarks and codes will be available in public. The experimental results suggest that the clip-level approach can be improved by leveraging sophisticated video segmentation methods. We also observed that considering context by the sliding window fashion or temporal connections between frames helps to encode video frames. We also expect that text encoding methods that can handle complicated sentence will be a key component for further improvement. An FGVR task on manually edited videos, *e.g.*, retrieving a scene from a movie, is a challenging and important topic. We will explore FGVR on manually created videos by modifying video and text alignment datasets, such as [14], [16] .

References

[1] Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B. and Vijayanarasimhan, S.: YouTube-8M: A Large-Scale Video Classification Benchmark, *arXiv preprint, arXiv:1609.08675* (10 pages, 2016).

[2] Chen, D. L. and Dolan, W. B.: Collecting highly parallel data for paraphrase evaluation, *Association for Computational Linguistics*, pp. 190–200 (2011).

[3] Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M. and Mikolov, T.: DeViSE: A Deep Visual-Semantic Embedding Model, *Advances in Neural Information Processing Systems (NIPS)*, pp. 2121–2129 (2013).

[4] He, K., Zhang, X., Ren, S. and Sun, J.: Deep Residual Learning for Image Recognition, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778 (2016).

[5] Kingma, D. and Ba, J.: Adam: A Method for Stochastic Optimization, *International Conference on Learning Representations (ICLR)* (15 pages, 2015).

[6] Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A. and Fidler, S.: Skip-Thought Vectors, *Advances in Neural Information Processing Systems (NIPS)*, pp. 3276–3284 (2015).

[7] Le, D.-D., Phan, S., Vinh-Tiep, N., Benjamin, R., Nguyen, T. A., Hoang, V.-N., Ngo, T. D., Tran, M.-T., Watanabe, Y., Klinkigt, M., Hiroike, A., Duong, D. A., Miyao, Y. and Satoh, S.: NII-HITACHI-UIT at TRECVID 2016, *TRECVID Workshops* (25 pages, 2016).

[8] Lee, Y. J., Ghosh, J. and Grauman, K.: Discovering important people and objects for egocentric video summarization, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1346–1353 (2012).

[9] Lin, D., Fidler, S., Kong, C. and Urtasun, R.: Visual Semantic Search: Retrieving Videos via Complex Textual Queries, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*

Table 1 Mean AP scores (%) of FGVR. GT denotes ground truth clip boundaries, and UNI denotes uniform segmentation.

video model / sentence model	clip boundaries	MSR-VTT		MPII-MD	
		cosine	p-order	cosine	p-order
F-Pool / W-Pool	GT	**86.5 (27.9)**	80.9 (31.5)	**77.7 (33.2)**	73.6 (34.8)
	UNI	81.1 (22.5)	76.0 (25.2)	74.4 (26.6)	70.7 (27.4)
F-Pool / W-LSTM	GT	85.4 (28.7)	79.2 (32.3)	74.8 (34.3)	69.0 (35.8)
	UNI	80.1 (23.1)	75.9 (25.3)	72.5 (27.6)	68.2 (28.4)
WA / W-LSTM	GT	86.4 (28.0)	75.9 (33.7)	75.8 (34.0)	69.0 (35.9)
	UNI	79.7 (23.2)	71.0 (26.7)	72.6 (27.4)	67.4 (28.5)
FC / W-LSTM	—	80.9 (23.7)	75.7 (25.2)	73.1 (27.7)	63.3 (27.6)
SW / W-LSTM	—	83.3 (22.9)	76.3 (25.7)	73.5 (27.9)	69.8 (28.8)
biLSTM / W-LSTM	—	**83.8 (22.7)**	72.5 (25.7)	**76.1 (28.9)**	61.7 (26.5)
by chance	—	47.0 (12.2)		49.4 (17.6)	

(CVPR), pp. 2657–2664 (2014).

[10] Maharaj, T., Ballas, N., Rohrbach, A., Courville, A. and Pal, C.: A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering, *arXiv preprint, arXiv:1611.07810* (9 pages, 2016).

[11] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, *Advances in Neural Information Processing Systems (NIPS)*, pp. 3111–3119 (2013).

[12] Otani, M., Nakashima, Y., Rahtu, E., Heikkilä, J. and Yokoya, N.: Learning Joint Representations of Videos and Sentences with Web Image Search, *European Conference on Computer Vision (ECCV) Workshops*, pp. 651–667 (2016).

[13] Real, E., Shlens, J., Mazzocchi, S., Pan, X. and Vanhoucke, V.: YouTube-BoundingBoxes: A Large High-Precision Human-Annotated Data Set for Object Detection in Video, *arXiv preprint, arXiv:1702.00824* (11 pages, 2017).

[14] Rohrbach, A., Rohrbach, M., Tandon, N. and Schiele, B.: A Dataset for Movie Description, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3202–3212 (2015).

[15] Rohrbach, A., Torabi, A., Rohrbach, M., Tandon, N., Pal, C., Larochelle, H., Courville, A. and Schiele, B.: Movie Description, *International Journal of Computer Vision*, Vol. 123, No. 1, pp. 94–120 (2017).

[16] Senina, A., Rohrbach, M., Qiu, W., Friedrich, A., Amin, S., Andriluka, M., Pinkal, M. and Schiele, B.: Coherent Multi-Sentence Video Description with Variable Level of Detail, *German Conference on Pattern Recognition (GCPR)*, pp. 184–195 (2014).

[17] Sharghi, A., Gong, B. and Shah, M.: Query-Focused Extractive Video Summarization, *European Conference on Computer Vision (ECCV)*, pp. 3–19 (2016).

[18] Snoek, C. G. M. and Worring, M.: Concept-Based Video Retrieval, *Foundations and Trends in Information Retrieval*, Vol. 2, No. 4, pp. 215–322 (2009).

[19] Socher, R., Ganjoo, M., Manning, C. D. and Ng, A. Y.: Zero-Shot Learning Through Cross-Modal Transfer, *Advances in Neural Information Processing Systems (NIPS)*, pp. 935–943 (2013).

[20] Song, Y., Vallmitjana, J., Stent, A. and Jaimes, A.: TVSum: Summarizing web videos using titles, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5179–5187 (2015).

[21] Tapaswi, M., Bäuml, M. and Stiefelhagen, R.: Book2movie: Aligning video scenes with book chapters, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1827–1835 (2015).

[22] Torabi, A., Tandon, N. and Sigal, L.: Learning Language-Visual Embedding for Movie Understanding with Natural-Language (13 pages, 2016).

[23] Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M.: Learning Spatiotemporal Features With 3D Convolutional Networks, *IEEE International Conference on Computer Vision (ICCV)*, pp. 4489–4497 (2015).

[24] Ueki, K., Kikuchi, K., Saito, S. and Kobayashi, T.: Waseda at TRECVID 2016: Ad-hoc Video Search, *TRECVID Workshops* (5 pages, 2016).

[25] Wang, M., Hong, R., Li, G., Zha, Z. J., Yan, S. and Chua, T. S.: Event driven web video summarization by tag localization and key-shot identification, *IEEE Trans. Multimedia*, Vol. 14, No. 4, pp. 975–985 (2012).

[26] Xiong, B., Kim, G. and Sigal, L.: Storyline Representation of Egocentric Videos With an Applications to Story-Based Search, *IEEE International Conference on Computer Vision (ICCV)*, pp. 4525–4533 (2015).

[27] Xu, J., Mei, T., Yao, T. and Rui, Y.: MSR-VTT: A Large Video Description Dataset for Bridging Video and Language, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5288–5296 (2016).

[28] Xu, R., Xiong, C., Chen, W. and Corso, J.: Jointly modeling deep video and compositional text to bridge vision and language in a unified framework, *AAAI Conference on Artificial Intelligence*, pp. 2346–2352 (2015).

[29] Yao, L., Ballas, N., Larochelle, H. and Courville, A.: Describing Videos by Exploiting Temporal Structure, *IEEE International Conference on Computer Vision (ICCV)*, pp. 4507–4515 (2015).

[30] Yu, Y., Ko, H., Choi, J. and Kim, G.: End-to-end Concept Word Detection for Video Captioning, Retrieval, and Question Answering, *arXiv preprint, arXiv:1610.02947* (20 pages, 2016).

[31] Zeng, K.-H., Chen, T.-H., Niebles, J. C. and Sun, M.: Title Generation for User Generated Videos, *European Conference on Computer Vision (ECCV)*, pp. 609–625 (2016).

[32] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A. and Oliva, A.: Learning Deep Features for Scene Recognition using Places Database, *Advances in Neural Information Processing Systems (NIPS)*, pp. 487–495 (2014).

[33] Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A. and Fidler, S.: Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books, *IEEE International Conference on Computer Vision (ICCV)*, pp. 19–27 (2015).