

# Abstractive Multi-Video Captioning: Benchmark Dataset Construction and Extensive Evaluation

Rikito Takahashi, Hirokazu Kiyomaru, Chenhui Chu\* and Sadao Kurohashi

Kyoto University

{r-takahashi, kiyomaru, chu, kuro}@nlp.ist.i.kyoto-u.ac.jp

## Abstract

This paper introduces a new task, abstractive multi-video captioning, which focuses on abstracting multiple videos with natural language. Unlike conventional video captioning tasks generating a specific caption for a video, our task generates an abstract caption of the shared content in a video group containing multiple videos. To address our task, models must learn to understand each video in detail and have strong abstraction abilities to find commonalities among videos. We construct a benchmark dataset for abstractive multi-video captioning named AbstrActs. AbstrActs contains 13.5k video groups and corresponding abstract captions. AbstrActs is available at <https://github.com/ku-nlp/AbstrActs>. As abstractive multi-video captioning models, we explore two approaches: end-to-end and cascade. For evaluation, we proposed a new metric, CocoA, which can evaluate the model performance based on the abstractness of the generated captions. In experiments, we report the impact of the way of combining multiple video features, the overall model architecture, and the number of input videos.

**Keywords:** video captioning, multi-video, abstraction

## 1. Introduction

Video captioning has attracted much attention as a fundamental task in vision-and-language research (Li et al., 2019; Aafaq et al., 2019b; Krishna et al., 2017). In the standard setting, models are presented with a video and generate a sentence that describes its content (Venugopalan et al., 2015b). In another popular setting, called dense video captioning (Krishna et al., 2017), models extract clips from the video, each representing a distinct event, and generate a caption for each clip. In both settings, the focus is on describing a given single video in detail.

In this paper, we shed light on another important aspect of video comprehension: abstractive video understanding. To elaborate, we examine the two videos in Fig. 1. We can, for example, describe the left video as “adults in sportswear are dancing in front of a mirror” and the right video as “elementary school girls are dancing in a gym.” However, we can also abstractly comprehend both videos and collectively describe them as “a group of people is dancing in a gym.” Such abstractive understanding is the key to identifying commonalities among videos.

Abstractive video understanding can help analyze large amounts of video data. One effective way to analyze large amounts of video data is video clustering (Jain, 2010). Auto-labeling to video clusters is a direct application of abstractive video understanding. Auto-labeling is necessary because it is difficult to know the common content of videos in each cluster and the tendency of a large num-

A group of people is dancing in a gym.



Figure 1: Example of abstractive multi-video captioning. The inputs are multiple videos. The output is the caption that describes the shared information in the videos.

ber of clusters. Abstractive video understanding can identify and describe the shared information in videos, i.e., it can generate a label from a video cluster, which can promote video data analysis.<sup>1</sup> Unfortunately, abstractive video understanding is difficult to learn through conventional video captioning. This is because conventional video captioning focuses on providing diverse and concrete descriptions of a single video (Aafaq et al., 2019b; Li et al., 2019). Therefore, in order to learn to abstract video content appropriately, we need to consider a task focusing on information abstraction. In this paper, we propose a new task, *abstractive multi-video captioning*, for abstractive video understanding. This task requires that models describe information shared by multiple videos as much as

<sup>1</sup>Although generating captions for each video with conventional models and then summarizing them can be another way, we show that abstractive video captioning performs significantly better.

\*Corresponding author

possible. In order to solve this task, models need to not only understand each video in detail but also have strong abstraction abilities to find commonalities among videos.

We construct *AbstrActs*, a dataset for abstractive multi-video captioning. *AbstrActs* consists of video groups, their corresponding abstract captions, abstractness scores of the abstract captions, and scores representing the degree of agreement between the video and the caption.

We explore model variants for the task and evaluate their performance on *AbstrActs*. Specifically, we investigate the impact of the way of combining multiple video features, the overall model architecture (end-to-end and cascade), and the number of input videos.<sup>2</sup> For evaluation, we propose a new metric, *CocoA*, which evaluates the model performance based on the correlation coefficient of abstractness scores between abstract captions and generated captions.

## 2. Related Work

We discuss video captioning datasets, video captioning models, and multi-image vision-and-language tasks.

### 2.1. Video Captioning Datasets

Over the past few years, several datasets have been constructed for video captioning. The most widely-used datasets include HowTo100M (Miech et al., 2019), VATEX (Wang et al., 2019), MSR-VTT (Xu et al., 2016), and MSVD (Chen and Dolan, 2011). HowTo100M is by far the largest video-caption dataset, containing  $136M$  video-caption pairs and  $23k$  types of actions. HowTo100M is used for pretraining vision-and-language models. VATEX, which is the source data for *AbstrActs* presented in this paper, is characterized by its large number of captions. VATEX has ten English and ten Chinese captions for each video. VATEX has 41,250 videos and 412,500 English captions in total.

ActivityNet Captions (Krishna et al., 2017) and YouCook2 (Zhou et al., 2018) are datasets for dense video captioning with multiple captions corresponding to video clips, each of which corresponds to a single event happening in the video. YouCook2 is a dataset with captions for long, unsegmented videos restricted to the cooking domain. YouCook2 contains 2,000 videos describing 89 different cooking recipes for 176 hours.

### 2.2. Video Captioning Models

Video captioning models are given a video and generate the caption. First, video captioning mod-

els extract video features using pretrained video encoders, typically constructed as either a CNN-based model (Tran et al., 2015; Carreira and Zisserman, 2017; Xie et al., 2018) or Transformer-based model (Dosovitskiy et al., 2021; Arnab et al., 2021; Luo et al., 2021; Liu et al., 2022; Bertasius et al., 2021). Most previous studies rely on CNN-based video encoders (Wang et al., 2018; Aafaq et al., 2019a; Zhang and Peng, 2019), but Transformer-based models are becoming popular in recent studies (Luo et al., 2020; Tang et al., 2021; Lin et al., 2022). Then, extracted video features are processed to generate captions. To this end, while early studies rely on LSTM-based models (Gao et al., 2017; Pan et al., 2017; Yan et al., 2019), recent studies increasingly use Transformer-based models (Zhou et al., 2018; Wang et al., 2018).

Note that the main focus of existing video captioning models, as well as datasets, is to make the specific content of a single video recognizable and to generate accurate captions. In this paper, we rather focus on understanding video information abstractly.

### 2.3. Multi-image Vision-and-Language Tasks

There are a handful of studies on vision-and-language tasks considering multiple images. ISVQA (Bansal et al., 2020) is a task of visual question answering from multiple images. The model answers natural language questions given multiple images showing the same location from different views. Context-aware group captioning is a task of image group captioning (Li et al., 2020b). The model generates a caption that summarizes multiple target images in the context of another group of related reference images.

Our task focuses on generating a caption from multiple videos and thus can be viewed as a temporal extension of previous studies on multi-image language generation tasks.

## 3. Task Definition

Abstractive multi-video captioning is a task to generate an abstract caption for multiple videos. The input is a video group  $G$  with  $n$  videos. The output is an abstract caption  $y$  that describes the shared content of the video group  $G$  as much as possible. Our goal is to learn an abstractive multi-video captioning model  $p_{\theta}(y|G)$  from training data, where  $\theta$  is the set of model parameters.

Fig. 1 shows an example of abstractive video captioning. The caption “A group of people is dancing in a gym.” is a good abstract caption for the video group; the phrase “a group of people” appropriately abstracts “adults in sportswear are dancing” shown in the left video and “elementary school

---

<sup>2</sup>Codes for our models are available at <https://github.com/ku-nlp/Abstractive-Multi-Video-Captioning>


Video Group	
Abstract Caption	a person is doing an exercise in a gym
Abstractness Score	0.73
TER Scores	10    ...    8

Figure 2: Sample of AbstrActs. AbstrActs is built from VATEX (Wang et al., 2019).

girls” shown in the right video, describing the commonalities between them.

We define a well-abstracted caption as a sentence that explains the shared information of multiple videos as much as possible. Note that we do not allow over-abstractation. This is because highly abstract captions such as “People are doing something.” are not helpful. However, if we consider abstractive *single-video* captioning, it is difficult to determine if a caption is overly abstract because the clear criterion is hard to establish. This is why we consider abstractive *multi-video* captioning; we consider the minimum level of abstraction at which commonalities among given videos can be found to be the criterion for over-abstractation. Thus, in our task, the caption “People are doing something.” for the video group in Fig. 1 is regarded as an overly abstract caption as it violates the requirement that the shared content must be described as much as possible.

## 4. Dataset Construction

We construct a new dataset for abstractive video captioning named *AbstrActs*. Fig. 2 shows an example in AbstrActs. An example consists of a video group, a human caption, abstractness, and TER scores. A video group is a collection of videos that have commonalities. A human caption is a manually assigned abstract caption for the video group. We use human captions as gold labels of abstractive multi-video captioning. Abstractness is a score that indicates the degree of abstraction of an abstract caption. A TER score indicates the content agreement between each video in the video group and the abstract caption, calculated using an off-the-shelf textual entailment recognition (TER) model. We use TER scores to evaluate the quality of abstract captions and filter out noisy data.

AbstrActs is built from VATEX (Wang et al., 2019). First, we construct video groups by performing a similarity search on videos in VATEX. Then, we use crowdsourcing to assign human captions to the video groups. Next, we define and calculate the abstractness of each abstract caption. Finally, we calculate TER scores using a TER model.

### 4.1. Video Group

We collect video groups, each of which contains videos with shared information, by performing video retrieval on the videos in VATEX. First, we extract video features of all videos using a pre-trained video encoder. In this study, we employed the CLIP4Clip (Luo et al., 2021) model. Then, we group videos by performing a k-nearest neighbor search on video features. We retrieve the top six similar videos for every video to form a group. Considering that the number of videos should be greater than the minimum problem set at two, we decided on six as the number of videos in the video group. We used Faiss (Johnson et al., 2019) to perform a k-nearest neighbor search.

### 4.2. Abstract Caption

We use crowdsourcing to annotate video groups with abstract captions. In order to increase the reliability and uniformity of abstract captions, we instruct crowdworkers to keep the following rules: (1) Do not write captions that describe the content that does not appear in the videos. (2) Do not write captions that simply list the events in each video. (3) Do not write captions that sum up the number of people or objects for each video. The instruction part of the crowdsourcing interface is shown in Appendix A.

To prevent over-abstractation in abstract captions, we also instructed crowdworkers to write a well-abstracted caption that explains the shared information of multiple videos as much as possible. We assigned one annotator to each video group, and the annotators varied by video group. After repeating annotations with 50 video groups and instructions improvements, we confirmed that the annotated captions correctly describe the shared video content. Then, we started the annotation of the remaining video groups.

### 4.3. Abstractness Score

We define the abstractness score for investigating the abstractness of abstract captions in AbstrActs. Each video in VATEX has ten captions annotated. The abstractness score is calculated between the abstract caption annotated by crowdworkers in our work and the original video captions in VATEX in a video group. Each video group only has one abstractness score.

We use WordNet (Miller, 1995) to calculate an abstractness score. WordNet is a lexical database of English. Words are grouped into sets of cognitive synonyms (synsets). Super-subordinate relations among synsets are organized as a tree structure. We use the distance on the tree for calculating an abstractness score.

We define the similarity between word  $w_1$  and

word  $w_2$  as follows:

$$\text{similarity}(w_1, w_2) = \frac{1}{1 + \text{path}(\text{syn}(w_1), \text{syn}(w_2))} \quad (1)$$

$\text{path}(\text{syn}(w_1), \text{syn}(w_2))$  is the number of edges of the shortest path between two synsets on the tree.

We calculate an abstractness score  $a$  by following equations:

$$a = 1 - f(W_a, V) \quad (2)$$

$$f(W_a, V) = \text{mean}_{W_v \in V} g(W_a, W_v) \quad (3)$$

$$g(W_a, W_v) = \text{mean}_{w_v \in W_v} h(W_a, w_v) \quad (4)$$

$$h(W_a, w_v) = \max_{w_a \in W_a} \text{similarity}(w_a, w_v) \quad (5)$$

$W_a$  is a set of nouns and verbs that appears in an abstract caption.  $V$  is a set of captions for a video in a video group.  $W_v$  is a set of nouns and verbs that appears in a caption of a video.  $w_v$  is a word in  $W_v$ .  $w_a$  is a word in  $W_a$ .

The abstractness score is a real number that ranges from 0 to 1. A higher abstractness score means a greater semantic difference between an abstract caption and a video caption.

#### 4.4. TER Score

Textual entailment recognition (TER) is the task of determining the entailment relationship between two sentences. Given a premise sentence and a hypothesis sentence, models determine whether the hypothesis sentence is true (entailment), false (contradiction), or undetermined (neutral), supposing that the premise sentence is true (Storks et al., 2019).

We use TER to evaluate the quality of abstract captions. We suppose that if many of the original captions of a video are entailed by the abstract caption, the abstract caption should appropriately abstract the shared content and thus be of high quality. Such an automatic quality evaluation is helpful in filtering out noisy examples, which are inevitably included as we rely on crowdsourcing or heuristics for data collection.

We assign a TER score to each pair of a video and an abstract caption. We perform TER regarding each of the original captions as a premise and the abstract caption as a hypothesis. We regard the number of original captions that entail the abstract caption as the TER score and assign it to the pair. We performed TER with SemBERT (Zhang et al., 2020a), a state-of-the-art TER model. We first excluded videos with a TER score of zero from a video group (making the number of videos of a video group possibly fewer than two), and then excluded video groups with fewer than two videos.

	Training	Validation	Test
Video Groups	10,983	830	1,674
Videos	38,514	2,452	5,157
Unique Videos	16,732	1,475	2,848
Videos per Group (avg.)	3.5	3.0	3.1
Abstractive Captions	10,983	830	1,674

Table 1: The statistics of AbstrActs. Videos indicate the total number of videos in each video group, including duplicates. Unique videos indicate the number of unique videos in each split.

#### 4.5. Analysis of AbstrActs

Tab. 1 shows the statistics of AbstrActs. We followed the data splitting of VATEX. The training, validation, and test videos and captions were obtained by processing VATEX’s training, validation, and test data, respectively. Video groups in every split had three or more videos on average. Videos in AbstrActs contained 600 types of human actions, listed in Kinetics-600 (Kay et al., 2017; Carreira et al., 2018).

Fig. 3 shows the distribution of abstractness scores in the training set. Abstractness scores were concentrated around 0.7, with a minimum value of 0.4 and a maximum value of 1.0. The result indicates that abstract captions are different in content from existing single-video captions.

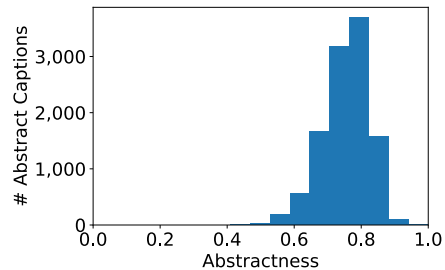


Figure 3: Abstractiveness score distribution of training data.

Fig. 4 shows the distribution of TER scores in the training set that indicate how well a video and an abstract caption match. We found that human captions have high TER scores, indicating that human captions properly abstract the video content in video groups. We observed the same tendency in the validation and test splits as well.

### 5. Abstractive Captioning Models

We construct abstractive captioning models as Transformer-based models. We explore various model variants to see what sort of improvements can be effective for abstractive multi-video caption generation.



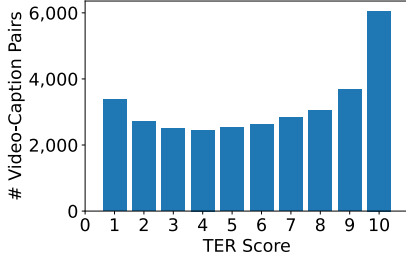


Figure 4: TER score distribution of training data.

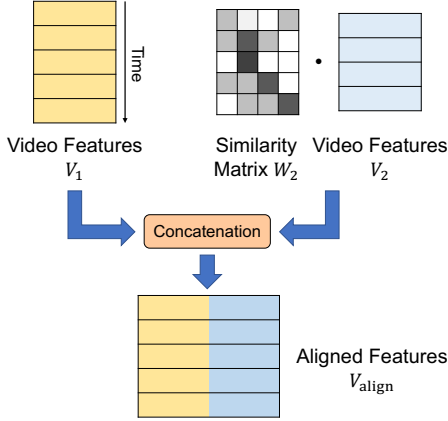


Figure 5: Example of soft alignment with two video features  $V_1$  and  $V_2$ .

### 5.1. Combination Methods for Multiple Features

As Transformer-based models take a single sequence of features as input, multiple video features must be processed into the form to feed them into the Transformer.

We explore two methods for inputting multiple video features: *concatenation* and *soft alignment*. Concatenation is a naive implementation to combine multiple video features, which performs a frame-wise concatenation of video features. This method does not consider differences between the contents of different frames in each video.

Soft alignment combines multiple video features by focusing on one video and collecting similar frames from the other videos, which shares the same idea as the attention mechanism (Bahdanau et al., 2014). Fig. 5 shows an overview of soft alignment. The frame-wise similarity between multiple video features is calculated first, and then the similarity weights the video features before being combined. By applying frame-wise soft alignment to  $n$  types of video features  $V_1, V_2, \dots, V_n$ , one video features  $V_{\text{align}}$  is obtained by the following equation:

$$V_{\text{align}} = \text{concat}(V_1, V_2', \dots, V_n'), \quad (6)$$

where

$$V_i' = W_i \cdot V_i. \quad (7)$$

Here,  $W_i$  is the similarity matrix for the video features  $V_1$  and the video features  $V_i$ . Let  $V_1 \in \mathbb{R}^{T_1 \cdot M}$  and  $V_i \in \mathbb{R}^{T_i \cdot M}$ , where  $T_1$  and  $T_i$  denote the number of frames in  $V_1$  and  $V_i$ , respectively, and  $M$  is feature dimension for each frame. Then  $W_i$  is a matrix  $\in \mathbb{R}^{T_1 \cdot T_i}$ . The similarity between each frame of the two videos is computed as follows:

$$W_i(t_1, t_i) = \frac{V_1(t_1) \cdot V_i^T(t_i)}{|V_1(t_1)| |V_i(t_i)|} \quad (8)$$

The feature  $V_{\text{align}}(t)$  at a frame  $t$  is the result of collecting features similar to the feature  $V_1(t)$  from other video features  $V_i$  and then combining them by weighting by similarity. The sequence length  $l$  of the features  $V_{\text{align}}$  is equal to that of the video features  $V_1$ .

The dimensions of the soft alignment feature increase in proportion to the number of input videos. For instance, the dimensionality of video features when using six videos as input is three times larger than when using two videos as input. When using more than three videos as input, we change the dimension of the input layer of the model corresponding to the dimensionality of the input video features, and we padded zeros to the portion of the features corresponding to the missing videos during soft alignment for video groups including fewer videos.

### 5.2. End-to-End Model

The end-to-end model learns to directly generate an abstract caption from an input video group. The end-to-end model has two advantages compared to the cascade model described next. First, it is free from error propagation. Second, the model can take full advantage of the information in the given videos.

Fig. 6a shows an overview of the end-to-end model. First,  $n$  input video features are obtained by a pretrained video encode. Then, the sequence of features  $V_{\text{multi}}$  is obtained using either the concatenation method or the soft alignment method described in Sec. 5.1. The transformed features  $V_{\text{multi}}$  is input to the Transformer-based video encoder to obtain the sequence of features  $\mathbf{z} = f_{\text{enc}}(V_{\text{multi}}) = (z_1, z_2, \dots, z_l)$ . The  $l$  is the length of the combined features  $V_{\text{multi}}$ .

Finally, the abstract caption  $y$  is generated using a Transformer-based language decoder. The word  $y_t = f_{\text{dec}}(\mathbf{y}, \mathbf{z})$  at decoding step  $t$  is generated based on the previously generated word sequence  $\mathbf{y} = (y_1, y_2, \dots, y_{t-1})$  and the feature sequence  $\mathbf{z}$ .

### 5.3. Cascade Model

The cascade model combines a single-video captioning module and a multi-sentence abstraction module. The single-video captioning module takes a single video as input and generates the

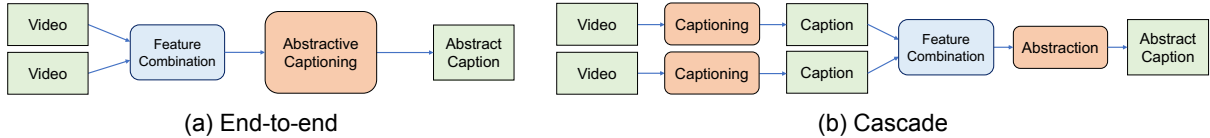


Figure 6: Two models for abstractive captioning. The end-to-end model directly generates an abstract caption from multiple videos. In the cascade model, the single-video captioning module first generates concrete captions for each video, and then the multi-sentence abstraction module generates an abstractive caption from them.

caption. The multi-sentence abstraction module takes multiple captions as input and generates the abstract caption.

Fig. 6b shows an overview of the cascade model. First, the single-video captioning module generates a caption for each video using the Transformer model. The resulting video captions are then encoded into sequences of word embeddings with a pretrained word embedding model. The resulting multiple caption features are converted into a single sequence of features using a soft alignment method similar to the one presented in Sec. 5.1. The difference is that instead of conducting soft alignment on video features, we do it on word embeddings of multiple captions. Finally, the transformed features are input to the Transformer to generate the abstract caption.

The advantage of the cascade model is its reusability. The performance of the cascade model can be improved by replacing the single-video captioning module with pretrained models. The disadvantage is that there is a possibility of error propagation due to the nature of solving each subtask independently and sequentially. If single-video captioning produces poor captions, the abstract caption produced by the multi-sentence abstraction module will also be of poor quality. Even if the performance of the single-video captioning module is adequate, there is a problem with missing information due to the conversion of video information into text. Spatio-temporal information in a video is difficult to explain entirely with text. Therefore, the caption generated by the single-video captioning module can lose some shared information in multiple videos, restricting the multi-sentence abstraction module from generating an abstract caption that describes the shared information as much as possible.

#### 5.4. Cascade (Gold) Model

We consider the cascade (gold) model as the upper-bound setting of the cascade model. This model does not use the single-video captioning module in the cascade model; instead, this model feeds the gold captions of each video to the multi-sentence abstraction module. As the gold caption should fully describe the content of the video, the gold caption can be regarded as a caption generated by a single-video captioning module with per-

fect performance. In both training and inference of the cascade (gold) model, the input to the model is the gold captions of multiple videos.

## 6. Experiments

We conduct experiments on abstractive captioning using the models described in Sec. 5.

### 6.1. CocoA: Correlation Coefficient of Abstractness Scores

BLEU-4 (Papineni et al., 2002), CIDEr (Vedantam et al., 2015), METEOR (Banerjee and Lavie, 2005), and ROUGE-L (Lin, 2004) are de facto standards in existing video captioning tasks. However, abstractive captioning is a task to generate abstract captions, whose purpose is different from existing video captioning tasks. Therefore, the criteria for a good caption differs between the abstractive multi-video captioning task and the existing video captioning task. For example, in the case of abstractive multi-video captioning, not only the word agreement between the correct label and the generated caption but also the abstractness of each word can be a criterion for evaluation. In addition, abstract captions generated by abstractive multi-video captioning tend to have fewer words than single-video captioning. This short caption is incompatible with the evaluation criteria that use n-grams in the sentence.

Motivated by the above, we propose a new metric, CocoA, which can evaluate model performance based on the abstractness score mentioned in Sec. 4.3. The value of CocoA is the Pearson correlation coefficient among the abstractness scores of generated captions  $A_Y$  and that of gold captions  $A_{Y'}$ .

$$\text{CocoA}(A_Y, A_{Y'}) = \frac{\text{cov}(A_Y, A_{Y'})}{\sigma_{A_Y} \sigma_{A_{Y'}}} \quad (9)$$

cov is the covariance.  $\sigma_{A_Y}$  and  $\sigma_{A_{Y'}}$  are the standard deviations of  $A_Y$  and  $A_{Y'}$ . We hypothesize that a strong correlation exists between abstractness scores when a model correctly generates an abstract caption.

### 6.2. Experimental Settings

We conducted experiments on AbstrActs and VA-TEX (Wang et al., 2019). Although each video

Method	CocoA	BLEU	CIDEr	METEOR	ROUGE-L
Concat	0.27	16.0	84.5	18.2	43.6
Soft	<b>0.34</b>	<b>18.6</b>	<b>130.1</b>	<b>21.5</b>	<b>47.3</b>

Table 2: Performance comparison of different feature combination methods with the end-to-end model. “Concat” and “Soft” denote for “Concatenation” and “Soft Alignment,” respectively.

Model	CocoA	BLEU	CIDEr	METEOR	ROUGE-L
E2E	<b>0.34</b>	<b>18.6</b>	<b>130.1</b>	<b>21.5</b>	<b>47.3</b>
Cas	0.25	14.8	65.4	17.4	40.9
Cas (G)	0.33	17.5	103.5	19.8	44.1
T5	0.50	20.3	154.6	23.2	49.1

Table 3: Performance comparison of different model structures. The T5 model is for reference. “E2E,” “Cas,” and “Cas (G)” denote for “End-to-End,” “Cascade,” and “Cascade (Gold),” respectively.

group in AbstrActs contains up to six videos, we fixed the number of videos to use to two except in Sec. 6.5 to simplify the experiment settings. We used two videos in each video group with the highest similarity scores in video retrieval when composing the video group. We used VATEX for training the cascade models and for inference in the cascade (gold) models. For evaluation metrics, we used CocoA together with BLEU-4, CIDEr, METEOR, and ROUGE-L.

### 6.3. Feature Combination Methods

We compared concatenation and soft alignment, the two methods for combining multiple features into a single feature described in Sec. 5.1. We conducted experiments with the end-to-end model described in Sec. 5.2 comparing two types of multi-input methods. For extracting video features, we used the CLIP4Clip model. Tab. 2 shows the results. The soft-aligned method consistently outperformed the naive concatenation method in all the metrics regardless of the video encoder. Soft alignment is different from the naive concatenation method in that soft alignment considers the similarity of the contents between the different frames of the videos. This difference helps the models find the shared content in multiple videos. We conclude from the experiments that it is better to use the soft-aligned video features as input to the end-to-end model, and thus, we adopt this setting for the subsequent experiments.

### 6.4. Model Architectures

We compared abstractive multi-video captioning with the end-to-end and cascade models. For extracting video features, we used the CLIP4Clip model. We used a pretrained word embedding

model from fastText (Bojanowski et al., 2017) for feature extraction of the caption obtained from the single-video captioning module.

Training of the cascade model was performed independently for each module. For training the single-video captioning module, we used VATEX. For training the multi-sentence abstraction module, we used both AbstrActs and VATEX. Each video in AbstrActs has gold captions in VATEX. We trained the abstractive captioning module to generate the abstract caption from each video’s gold captions in VATEX.

Tab. 3 shows the result. The end-to-end model outperformed the cascade and cascade (gold) models in all evaluation metrics. For reference, we also provide the score of the T5 (Raffel et al., 2020) model trained to abstract the gold captions of each video in a video group. Note that we concatenate gold captions and input them into the T5 model.

One reason for the better performance of the end-to-end model is that it does not have the error propagation problem that can occur in the cascade model. We manually investigated 50 inference results on the test set. We found that in 7 cases, the cascade model generated a poor abstract caption due to the generation error in the single captioning module. Fig. 7a shows an example that the end-to-end model describes the children in the two videos, while the cascade model describes it as a “person.” Fig. 7b shows an example of the generation of the cascade model. A child is in the right video, but the single-video captioning module describes him in a more abstract word “man.” This generation error propagates to the multi-sentence abstraction module, which generates the overly abstract word “person” instead of the expected words such as “child” or “kid.” The end-to-end model uses the video features directly, eliminating the risk of error propagation and producing the preferred word “kid.”

We investigated the generated captions related to their abstractness scores. Samples with different abstractness scores may also have different difficulties in abstraction. Fig. 8a shows the example with a low abstractness score. The two videos have almost the same content: a person is swimming in a pool. In this example, models correctly generated an abstract caption from the videos. Fig. 8b shows the example with a high abstractness score. The left video shows a male hairstylist combing and cutting a woman’s hair, and the right video shows a female hairstylist washing and cutting a man’s hair. The shared information in the two videos is that a hairstylist works on a customer. Models generated captions with improper abstraction from these videos. It would be interesting to evaluate and investigate the generated

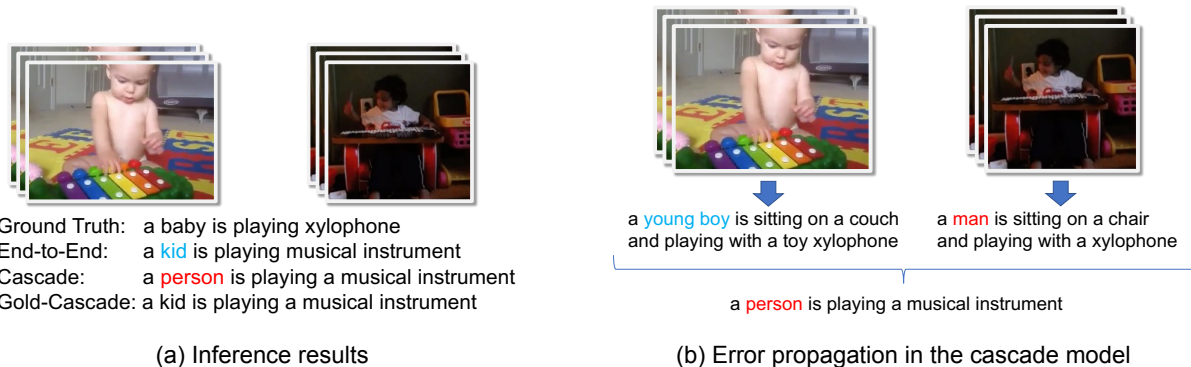


Figure 7: Examples of the inference results with various types of model structures. Words highlighted in blue represent a preferred generation, while words highlighted in red represent a poor generation.

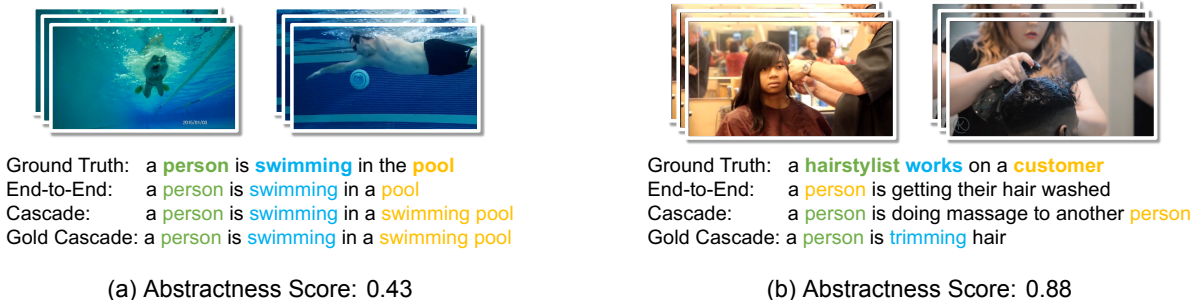


Figure 8: Inference examples with different abstractness scores. Highlighted words represent an abstraction. Words highlighted in the same color describe the same object in the videos.

#Videos	CocoA	BLEU	CIDEr	METEOR	ROUGE-L
One	0.32	18.1	119.2	21.1	46.6
Two	<b>0.34</b>	<b>18.6</b>	<b>130.1</b>	<b>21.5</b>	<b>47.3</b>
Six	0.32	17.6	120.8	20.9	46.6

Table 4: Performance comparisons of different numbers of video inputs for the end-to-end model.

caption more deeply regarding the abstractness, which we leave as future work.

### 6.5. Number of Videos

We further investigated the impact of the number of input videos. In previous experiments, two videos were fed to the models. We compared the number of input videos with one, two, and six, using the end-to-end model as it consistently outperformed the cascade model. For the one video setting, we randomly selected one video from a video group.

Tab. 4 shows the result. We can see that using one video performs worse than using two videos, indicating the importance of understanding multiple videos for the proposed task. Using the input features from six videos did not have a positive impact; in all evaluation metrics, the two-video setting outperformed the six-video setting. One possible reason is that inputting many videos into the model degrades each video’s information. We

fixed the number of parameters in the Transformer of the end-to-end model. Even if the number of input videos increases to six, the dimension of features processed by the Transformer’s attention mechanism remains the same. As the number of input videos increases, the dimensions of the feature values used per video decrease. In the model where six videos are input, the information for each video is compressed, leading to performance degradation. Another possible reason for performance degradation is zero-padding in the soft alignment. We used zero-padding to shape the features when performing soft alignment in a video group with fewer than six videos. For example, if only two videos are in a video group, all the values in the matrix for the remaining four videos are padded with zeros. This zero-padding may become noise and prevent the model from learning. Inference examples with different number of input videos can be found in Appendix B.

## 7. Conclusion

We introduced a new task, abstractive multi-video captioning, aiming at training models to find and describe commonalities among videos. We constructed the benchmark dataset AbstrActs, containing 13.5k video groups, 27k abstract captions, abstractness scores, and scores representing the degree of agreement between the video and the



abstract caption. We proposed a new metric named CocoA, which evaluates the model performance in terms of the abstractness of captions. We extensively explored model variants to see what improvements can be effective for abstract caption generation on AbstrActs. We hope the dataset, the metric, and insights into models presented in this study facilitate future research on abstractive multi-video captioning. In the future, we plan to investigate the applicability of the models for auto-labeling to video clusters.

## 8. Limitations

**Actions in AbstrActs.** In order to simplify the problem setup, We built AbstrActs from VATEX (Wang et al., 2019), in which each video includes just one representative action. However, a video usually includes multiple actions and events. To extend the present work, we need to expand the dataset domain with videos that contain multiple actions.

**Constructing video groups.** When collecting video groups by video retrieval, we used the video features extracted by CLIP4Clip (Luo et al., 2021). This may lead to some bias if we use other video encoders such as (Bain et al., 2021; Li et al., 2020a) in experiments because the similarity of videos depends on the video encoders being used for extracting video features. Our future work is evaluating the effect of the biases and considering another way of collecting video groups.

**Applicability of the models.** In this paper, we did not address the applicability of the models for auto-labeling to video clusters. To simplify the experiment settings, we fixed the number of videos to two except in Sec. 6.5. Besides, the result in Sec. 6.5 indicates that using six videos as input did not positively impact. Given the above, it is worth considering if we can apply models to generate the label of a large video cluster. Applying our model to each of the two video pairs in a cluster to generate abstract captions and then further generate more abstractive captions from them may address the problem of many videos, which we leave as future work.

## 9. Ethics Statement

We used Amazon Mechanical Turk to recruit the crowdworkers at a price of 10 US dollars per hour on average. By agreeing to work on the annotation, crowdworkers have agreed to give the right to use the annotation for research purposes. Videos were shown to crowdworkers for annotation. The videos were from VATEX, which does not contain any harmful ones.

## 10. Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP23H03454 and Fujitsu.

## 11. Bibliographical References

- Nayyer Aafaq, Naveed Akhtar, Wei Liu, Syed Zulqarnain Gilani, and Ajmal Mian. 2019a. Spatio-temporal dynamics and semantic attribute enriched visual encoding for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12487–12496.
- Nayyer Aafaq, Ajmal Mian, Wei Liu, Syed Zulqarnain Gilani, and Mubarak Shah. 2019b. Video description: A survey of methods, datasets, and evaluation metrics. *ACM Computing Surveys (CSUR)*, 52(6):1–37.
- Naomi S Altman. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185.
- Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. 2021. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738.
- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.
- Ankan Bansal, Yuting Zhang, and Rama Chellappa. 2020. Visual question answering on image sets. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
- Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*.

- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.
- Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308.
- Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.
- Lianli Gao, Zhao Guo, Hanwang Zhang, Xing Xu, and Heng Tao Shen. 2017. Video captioning with attention-based lstm and semantic consistency. *IEEE Transactions on Multimedia*, 19(9):2045–2055.
- Andrew Gilbert and Richard Bowden. 2011. igroup: Weakly supervised image and video grouping. In *2011 International Conference on Computer Vision*, pages 2166–2173. IEEE.
- Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. 2020. Coot: Cooperative hierarchical transformer for video-text representation learning. *arXiv preprint arXiv:2011.00597*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Vladimir Iashin and Esa Rahtu. 2020. Multi-modal dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 958–959.
- Anil K Jain. 2010. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.
- Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. 2020a. Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *EMNLP*.
- Sheng Li, Zhiqiang Tao, Kang Li, and Yun Fu. 2019. Visual to text: Survey of image and video captioning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 3(4):297–312.
- Zhuowan Li, Quan Tran, Long Mai, Zhe Lin, and Alan L Yuille. 2020b. Context-aware group captioning via self-attention and contrastive features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3440–3450.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. 2022. Swinbert: End-to-end transformers with sparse attention for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17949–17958.
- Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. 2022. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211.
- Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. 2020. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*.
- Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2021. Clip4clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860*.
- 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*.
- Yingwei Pan, Ting Yao, Houqiang Li, and Tao Mei. 2017. Video captioning with transferred semantic attributes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6504–6512.

- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.
- Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. [Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation](#). *CoRR*, abs/1706.09799.
- Alok Singh, Thoudam Doren Singh, and Sivaji Bandyopadhyay. 2020. Nits-vc system for vatex video captioning challenge 2020. *arXiv preprint arXiv:2006.04058*.
- Shane Storks, Qiaozi Gao, and Joyce Y Chai. 2019. Recent advances in natural language inference: A survey of benchmarks, resources, and approaches. *arXiv preprint arXiv:1904.01172*.
- Mingkang Tang, Zhanyu Wang, Zhenhua Liu, Fengyun Rao, Dian Li, and Xiu Li. 2021. Clip4caption: Clip for video caption. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4858–4862.
- Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497.
- Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.
- Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015a. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542.
- Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. 2014. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*.
- Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. 2015b. [Translating videos to natural language using deep recurrent neural networks](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1494–1504, Denver, Colorado. Association for Computational Linguistics.
- Bairui Wang, Lin Ma, Wei Zhang, and Wei Liu. 2018. Reconstruction network for video captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7622–7631.
- Zheshen Wang, Ming Zhao, Yang Song, Sanjiv Kumar, and Baoxin Li. 2010. Youtubecat: Learning to categorize wild web videos. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 879–886. IEEE.
- Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321.
- Chenggang Yan, Yunbin Tu, Xingzheng Wang, Yongbing Zhang, Xinhong Hao, Yongdong Zhang, and Qionghai Dai. 2019. Stat: Spatial-temporal attention mechanism for video captioning. *IEEE transactions on multimedia*, 22(1):229–241.
- Junchao Zhang and Yuxin Peng. 2019. Object-aware aggregation with bidirectional temporal graph for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8327–8336.
- Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou.

2020a. Semantics-aware bert for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9628–9635.

Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. 2020b. Object relational graph with teacher-recommended learning for video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13278–13288.

Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. 2018. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8739–8748.

## 12. Language Resource References

Joao Carreira and Eric Noland and Andras Banki-Horvath and Chloe Hillier and Andrew Zisserman. 2018. *A short note about kinetics-600*.

David Chen and William B Dolan. 2011. *Collecting highly parallel data for paraphrase evaluation*.

Will Kay and Joao Carreira and Karen Simonyan and Brian Zhang and Chloe Hillier and Sudheendra Vijayanarasimhan and Fabio Viola and Tim Green and Trevor Back and Paul Natsev and others. 2017. *The kinetics human action video dataset*.

Ranjay Krishna and Kenji Hata and Frederic Ren and Fei-Fei Li and Juan Carlos Niebles. 2017. *Dense-captioning events in videos*.

Antoine Miech and Dimitri Zhukov and Jean-Baptiste Alayrac and Makarand Tapaswi and Ivan Laptev and Josef Sivic. 2019. *Howto100m: Learning a text-video embedding by watching hundred million narrated video clips*.

George A Miller. 1995. *WordNet: a lexical database for English*. ACM New York, NY, USA.

Xin Wang and Jiawei Wu and Junkun Chen and Lei Li and Yuan-Fang Wang and William Yang Wang. 2019. *Vatex: A large-scale, high-quality multilingual dataset for video-and-language research*.

Jun Xu and Tao Mei and Ting Yao and Yong Rui. 2016. *Msr-vtt: A large video description dataset for bridging video and language*.

Luowei Zhou and Chenliang Xu and Jason J Corso. 2018. *Towards automatic learning of procedures from web instructional videos*.

### A. Crowdsourcing Interface

Fig. 9 shows the instruction part of the crowdsourcing interface that we used to collect human captions. We presented the task instructions with an annotation example to the crowdworker. We showed multiple videos in a video group to crowdworkers and asked them to write an abstract caption describing the shared content in the videos.

### B. Inference Examples With Different Number of Input Videos

Fig. 10 shows the generated captions in the experiments with the different number of input videos. In the six-video setting, the model generated an over-abstract caption, while the model generated an appropriate caption in the two-video setting.




**Instruction:**  
Given multiple videos, write an abstract caption that describes what they show in common.

**Notes:**

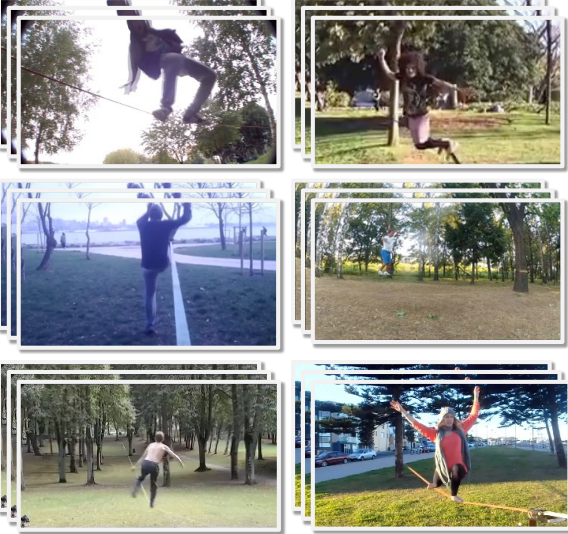
- Write a sentence that is grammatically correct (e.g. "A person is jumping.")
- Listen to the audio of each video.
- Don't add information that does not appear in videos.
- Don't submit a sequence of descriptions for each video (e.g. "A person is running in a gym, and another is climbing up a rope, and...")
- Don't mix the number of objects (e.g. "people are walking" when a person is walking in each video.)

**Example:**



A person is trying to climb a rope.

Figure 9: Instruction part of the crowdsourcing interface for collecting human captions.



Ground Truth: a person is trying to walk on a rope  
 Two Videos: a person is walking on a rope  
 Six Videos: a person is doing gymnastics

Figure 10: Inference examples with different numbers of input videos. We used the top two videos in this figure in the two-video setting.