# Reconstruction Network for Video Captioning

Bairui Wang[‡] Lin Ma[†*] Wei Zhang[‡*] Wei Liu[†]
[†]Tencent AI Lab [‡]School of Control Science and Engineering, Shandong University
{bairuiwong,forest.linma}@gmail.com davidzhang@sdu.edu.cn wliu@ee.columbia.edu

## Abstract

*In this paper, the problem of describing visual contents of a video sequence with natural language is addressed. Unlike previous video captioning work mainly exploiting the cues of video contents to make a language description, we propose a reconstruction network (RecNet) with a novel encoder-decoder-reconstructor architecture, which leverages both the forward (video to sentence) and backward (sentence to video) flows for video captioning. Specifically, the encoder-decoder makes use of the forward flow to produce the sentence description based on the encoded video semantic features. Two types of reconstructors are customized to employ the backward flow and reproduce the video features based on the hidden state sequence generated by the decoder. The generation loss yielded by the encoder-decoder and the reconstruction loss introduced by the reconstructor are jointly drawn into training the proposed RecNet in an end-to-end fashion. Experimental results on benchmark datasets demonstrate that the proposed reconstructor can boost the encoder-decoder models and leads to significant gains in video caption accuracy.*

## 1. Introduction

Describing visual contents with natural language automatically has received increasing attention in both the computer vision and natural language processing communities. It can be applied in various practical applications, such as image and video retrieval [33, 44, 22], answering questions from images [21], and assisting people who suffer from vision disorders [43].

Previous work predominantly focused on describing still images with natural language [15, 41, 42, 28, 13, 5]. Recently, researchers have strived to generate sentences to describe video contents [48, 8, 39, 40, 25]. Compared to image captioning, describing videos is more challenging as the amount of information (*e.g.*, objects, scenes, actions, *etc.*) contained in videos is much more sophisticated than that in still images. More importantly, the temporal dynamics within video sequences need to be adequately captured for captioning, besides the spatial content modeling.

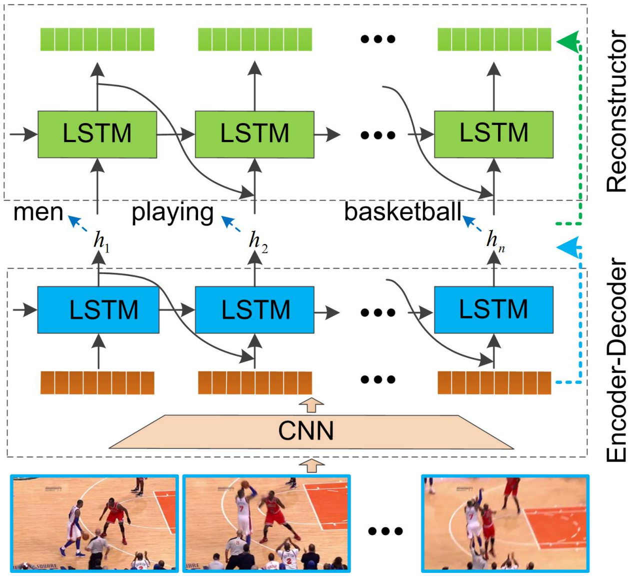


Figure 1. The proposed RecNet with an encoder-decoder-reconstructor architecture. The encoder-decoder relies on the forward flow from video to caption (blue dotted arrow), in which the decoder generates caption with the frame features yielded by the encoder. The reconstructor exploiting the backward flow from caption to video (green dotted arrow), takes the hidden state sequence of the decoder as input and reproduces the visual features of the video.

Recently, the encoder-decoder architecture, has been widely adopted for video captioning [8, 27, 14, 49, 9, 34, 24, 25, 19], as shown in Fig. 1. However, the encoder-decoder architecture only relies on the forward flow (video to sentence), but does not consider the information from sentence to video, named as backward flow. Usually the encoder is a convolutional neural network (CNN) capturing the image structure to yield its semantic representation. For a given video sequence, the yielded semantic representations by the CNN are further fused together to exploit the video temporal dynamics and generate the video representation. The

[*]Corresponding authors

decoder is usually a long short-term memory (LSTM) [12] or a gated recurrent unit (GRU) [7], which is popular in processing sequential data [53]. LSTM and GRU generate the sentence fragments one by one, and ensemble them to form one sentence. The semantic information from target sentences to source videos are never included. Actually, the backward flow can be yielded by the dual learning mechanism that has been introduced into neural machine translation (NMT) [37, 11] and image segmentation [20]. This mechanism reconstructs source from target when the target is achieved and demonstrates that backward flow from target to source improves performance.

To well exploit the backward flow, we refer to idea of dual learning and propose an encoder-decoder-reconstructor architecture shown in Fig. 1, dubbed as RecNet, to address video captioning. Specifically, the encoder-decoder yields the semantic representation of each video frame and subsequently generates a sentence description. Relying on the backward flow, the reconstructor, realized by LSTMs, aims at reproducing the original video feature sequence based on the hidden state sequence of the decoder. The reconstructor, targeting at minimizing the differences between original and reproduced video features, is expected to further bridge the semantic gap between the natural language captions and video contents.

To summarize, the contributions of this work lie in threefold.

- We propose a novel reconstruction network (RecNet) with an encoder-decoder-reconstructor architecture to exploit both the forward (video to sentence) and backward (sentence to video) flows for video captioning.
- Two types of reconstructors are customized to restore the video global and local structures, respectively.
- Extensive results on benchmark datasets indicate that the backward flow is well addressed by the proposed reconstructor and significant gains on video captioning are achieved.

## 2. Related Work

In this section, we first introduce two types of video captioning: template-based approaches [17, 10, 30, 29, 48] and sequence learning approaches [49, 39, 40, 8, 27, 14, 52, 24, 25, 32, 19], then introduce the application of dual learning.

### 2.1. Template-based Approaches

Template-based methods first define some specific rules for language grammar, and then parse the sentence into several components such as subject, verb, and object. The obtained sentence fragments are associated with words detected from the visual content to produce the final description about the input video with predefined templates. For example, a concept hierarchy of actions was introduced to describe human activities in [17], while a semantic hierarchy was defined in [10] to learn the semantic relationship between different sentence fragments. In [30], the conditional random field (CRF) was adopted to model the connections between objects and activities of the visual input and generate the semantic features for description. Besides, Xu *et al.* proposed a unified framework consisting of a semantic language model, a deep video model, and a joint embedding model to learn the association between videos and natural sentences [48]. However, as stated in [25], the aforementioned approaches highly depend on the predefined template and are thus limited by the fixed syntactical structure, which is inflexible for sentence generation.

### 2.2. Sequence Learning Approaches

Compared with the template-based methods, the sequence learning approaches aim to directly produce the sentence description about the visual input with more flexible syntactical structures. For example, in [40], the video representation was obtained by averaging each frame feature extracted by a CNN, and then fed to LSTMs for sentence generation. In [24], the relevance between video context and sentence semantics was considered as a regularizer in the LSTM. However, since simple mean pooling is used, the temporal dynamics of the video sequence are not adequately addressed. Yao *et al.* introduced an attention mechanism to assign weights to the features of each frame and then fused them based on the attentive weights [49]. Venugopalan *et al.* proposed S2VT [39], which included the temporal information with optical flow and employed LSTMs in both the encoder and decoder. To exploit both temporal and spatial information, Zhang and Tian proposed a two-stream encoder comprised of two 3D CNNs [36, 16] and one parallel fully connected layer to learn the features from the frames [52]. Besides, Pan *et al.* proposed a transfer unit to model the high-level semantic attributes from both images and videos, which are rendered as the complementary knowledge to video representations for boosting sentence generation [25].

In this paper, our proposed RecNet can be regarded as a sequence learning method. However, unlike the above conventional encoder-decoder models which only depend on the forward flow from video to sentence, RecNet can also benefit the backward flow from sentence to video. By fully considering the bidirectional flows between video and sentence, RecNet is capable of further boosting the video captioning.

### 2.3. Dual Learning Approaches

As far as we know, dual learning mechanism has not been employed in video captioning but widely used in NMT [37, 11, 45]. In [37], the source sentences are repro-

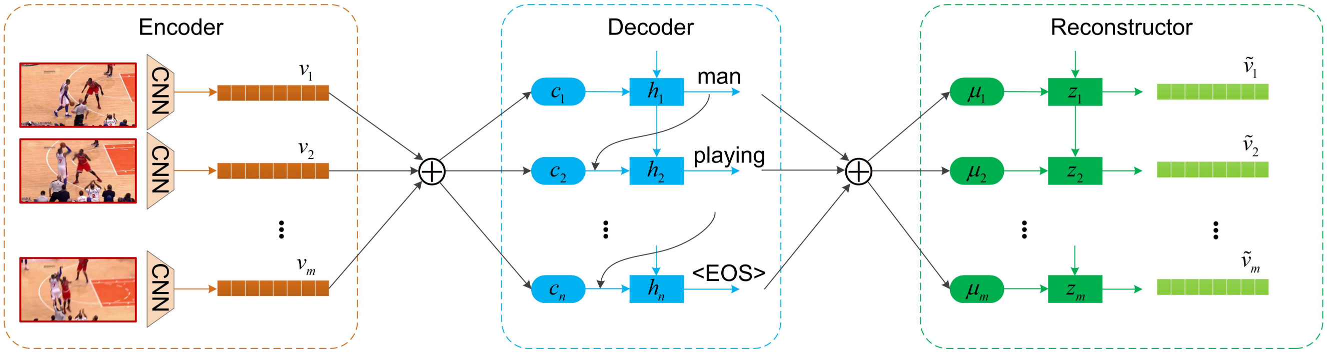


Figure 2. The proposed RecNet consists of three parts: the CNN-based encoder which extracts the semantic representations of the video frames, the LSTM-based decoder which generates natural language for visual content description, and the reconstructor which exploits the backward flow from caption to visual contents to reproduce the frame representations.

duced from the target side hidden states, and the accuracy of reconstructed source provides a constraint for decoder to embed more information of source language into target language. In [11], the dual learning is employed to train model of inter-translation of English-French, and get significantly improvement on tasks of English to French and French to English.

## 3. Architecture

We propose a novel RecNet with an encoder-decoder-reconstructor architecture for video captioning, which works in an end-to-end manner. The reconstructor imposes one constraint that the semantic information of one source video can be reconstructed from the hidden state sequence of the decoder. The encoder and decoder are thus encouraged to embed more semantic information about the source video. As illustrated in Fig. 2, the proposed RecNet consists of three components, specifically the encoder, the decoder, and the reconstructor. Moreover, our designed reconstructor can collaborate with different classical encoder-decoder architectures for video captioning. In this paper, we employ the attention-based video captioning [49] and S2VT [39]. We first briefly introduce the encoder-decoder model for video captioning. Afterwards, the proposed reconstructors with two different architectures are described.

### 3.1. Encoder-Decoder

The aim of video captioning is to generate one sentence $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n\}$ to describe the content of one given video $\mathbf{V}$. Classical encoder-decoder architectures directly model the captioning generation probability word by word:

$$P(\mathbf{S}|\mathbf{V}) = \prod_{i=1}^{n} P\left(\mathbf{s}_i|\mathbf{s}_{<i}, \mathbf{V}; \theta\right), \qquad (1)$$

where $\theta$ keeps the parameters of the encoder-decoder model. $n$ denotes the length of the sentence, and $\mathbf{s}_{<i}$ (*i.e.*, $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{i-1}\}$) denotes the generated partial caption.

**Encoder.** To generate reliable captions, visual features need to be extracted to capture the high-level semantic information about the video. Previous methods usually rely on CNNs, such as AlexNet [40], GoogleNet [49], and VGG19 [46] to encode each video frame into a fixed-length representation with the high-level semantic information. By contrast, in this work, considering a deeper network is more plausible for feature extraction, we advocate using Inception-V4 [35] as the encoder. In this way, the given video sequence is encoded as a sequential representation $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m\}$, where $m$ denotes the total number of the video frames.

**Decoder.** Decoder aims to generate the caption word by word based on the video representation. LSTM with the capabilities of modeling long-term temporal dependencies are used to decode video representation to video captions word by word. To further exploit the global temporal information of videos, a temporal attention mechanism [49] is employed to encourage the decoder to selecting the key frames/elements for captioning.

During the captioning process, the $i_{th}$ word prediction is generally made by LSTM:

$$P\left(\mathbf{s}_i|\mathbf{s}_{<i}, \mathbf{V}, \theta\right) \propto \mathbf{exp}\big(f(\mathbf{s}_{i-1}, h_i, c_i; \theta)\big), \qquad (2)$$

where $f$ represents the LSTM activation function, $h_i$ is the $i_{th}$ hidden state computed in the LSTM, and $c_i$ denotes the $i_{th}$ context vector computed with the temporal attention mechanism. The temporal attention mechanism is used to assign weight $\alpha_i^t$ to the representation of each frame $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m\}$ at the time step $t$ as follows:

$$c_t = \sum_{i=1}^{m} \alpha_i^t \mathbf{v}_i, \qquad (3)$$

where $m$ denotes the number of the video frames. With the $(i-1)_{th}$ hidden state $h_{i-1}$ summarizing all the current generated words, the attention weight $\alpha_i^t$ reflects the relevance of the $i_{th}$ temporal feature in the video sequence given all the previously generated words. As such, the temporal attention strategy allows the decoder to select a subset of key frames to generate the word at each time step, which can improve the video captioning performance as demonstrated in [49].

The encoder-decoder model can be jointly trained by minimizing the negative log likelihood to produce the correct description sentence given the video as follows:

$$\min_{\theta} \sum_{i=1}^{N} \left\{ -\log P\left(\mathbf{S}^i | \mathbf{V}^i ; \theta\right) \right\}. \tag{4}$$

## 3.2. Reconstructor

As shown in Fig. 2, the proposed reconstructor is built on the top of the encoder-decoder, which is expected to reproduce the video from the hidden state sequence of the decoder. However, due to the diversity and high dimension of the video frames, directly reconstructing the video frames seems to be intractable. Therefore, in this paper, the reconstructor aims at reproducing the sequential video frame representations generated by the encoder, with the hidden states $\mathbf{H} = \{h_1, h_2, ..., h_n\}$ of the decoder as input. The benefits of such a structure are two-fold. First, the proposed encoder-decoder-reconstructor architecture can be trained in an end-to-end fashion. Second, with such a reconstruction process, the decoder is encouraged to embed more information from the input video sequence. Therefore, the relationships between the video sequence and caption can be further enhanced, which is expected to improve the video captioning performance. In practice, the reconstructor is realized by LSTMs. Two different architectures are customized to summarize the hidden states of the decoder for video feature reproduction. More specifically, one focuses on reproducing the global structure of the provided video, while the other pays more attentions to the local structure by selectively attending to the hidden state sequence.

### 3.2.1 Reconstructing Global Structure

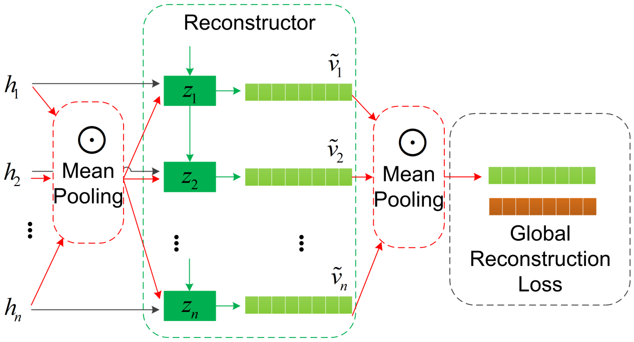


Figure 3. An illustration of the proposed reconstructor that reproduces the global structure of the video sequence. The left mean pooling is employed to summarize the hidden states of the decoder for the global representation of the caption. The reconstructor aims to reproduce the feature representation of the whole video by mean pooling (the right one) using the global representation of the caption as well as the hidden state sequence of the decoder.

The architecture for reconstructing the global structure of the video sequence is illustrated in Fig. 3. The whole sentence is fully considered to reconstruct the video global structure. Therefore, besides the hidden state $h_t$ at each time step, the global representation characterizing the semantics of the whole sentence is also taken as the input at each step. Several methods like LSTM and multiple-layer perception, can be employed to fuse the hidden sequential states of the decoder to generate the global representation. Inspired by [39], the mean pooling strategy is performed on the hidden states of the decoder to yield the global representation of the caption:

$$\phi\left(\mathbf{H}\right) = \frac{1}{n} \sum_{i=1}^{n} h_i, \tag{5}$$

where $\phi\left(\cdot\right)$ denotes the mean pooling process, which yields a vector representation $\phi\left(\mathbf{H}\right)$ with the same size as $h_i$. Thus, the LSTM unit of the reconstructor is further modified as:

$$\begin{gathered}\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \mathbf{T} \begin{pmatrix} h_t \\ z_{t-1} \\ \phi\left(\mathbf{H}\right) \end{pmatrix}, \\ m_t = f_t \odot m_{t-1} + i_t \odot g_t, \\ z_t = o_t \odot \tanh(m_t), \end{gathered} \tag{6}$$

where $i_t$, $f_t$, $m_t$, $o_t$, and $z_t$ denote the input, forget, memory, output, and hidden states of each LSTM unit, respectively. $\sigma$ and $\odot$ denote the logistic sigmoid activation and the element-wise multiplication, respectively.

To reconstruct the video global structure from the hidden state sequence produced by the encoder-decoder, the global reconstruction loss is defined as:

$$\mathcal{L}_{rec}^{g} = \psi\big(\phi(\mathbf{V}), \phi(\mathbf{Z})\big), \tag{7}$$

where $\phi(\mathbf{V})$ denotes the mean pooling process on the video frame features, yielding the ground-truth global structure of the input video sequence. $\phi(\mathbf{Z})$ works on the hidden states of the reconstructor, indicating the global structure recovered from the captions. The reconstruction loss is measured by $\psi(\cdot)$, which is simply chosen as the Euclidean distance.

#### 3.2.2 Reconstructing Local Structure

The aforementioned reconstructor aims to reproduce the global representation for the whole video sequence, while neglects the local structures in each frame. In this subsection, we propose to learn and preserve the temporal dynamics by reconstructing each video frame as shown in Fig. 4. Differing from the global structure estimation, we intend to reproduce the feature representation of each frame from the key hidden states of the decoder selected by the attention strategy [1, 49]:

$$\mu_t = \sum_{i=1}^{n} \beta_i^t h_i, \tag{8}$$

where $\sum_{i=1}^{n} \beta_i^t = 1$ and $\beta_i^t$ denotes the weight computed for the $i_{th}$ hidden state at time step $t$ by the attention mechanism. Similar to Eq. 3, $\beta_i^t$ measures the relevance of the $i_{th}$ hidden state in the caption given all the previously reconstructed frame representations $\{z_1, z_2, \ldots, z_{t-1}\}$. Such a strategy encourages the reconstructor to work on the hidden states selectively by adjusting the attention weight $\beta_i^t$ and yield the context information $\mu_t$ at each time step as in Eq. 8. As such, the proposed reconstructor can further exploit the temporal dynamics and the word compositions across the whole caption. The LSTM unit is thereby reformulated as:

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \mathbf{T} \begin{pmatrix} \mu_t \\ z_{t-1} \end{pmatrix}. \tag{9}$$

Differing from the global structure recovery step in Eq. 6, the dynamically generated context $\mu_t$ is taken as the input other than the hidden state $h_t$ and its mean pooling representation $\phi(\mathbf{H})$. Moreover, instead of directly generating the global mean representation of the whole video sequence, we propose to produce the feature representation frame by frame. The reconstruction loss is thereby defined as:

$$\mathcal{L}_{rec}^{l} = \frac{1}{m} \sum_{j=1}^{m} \psi(z_j, \mathbf{v}_j). \tag{10}$$

### 3.3. Training

Formally, we train the proposed encoder-decoder-reconstructor architecture by minimizing the whole loss defined in Eq. 11, which involves both the forward (video-to-sentence) likelihood and the backward (sentence-to-video) reconstruction loss:

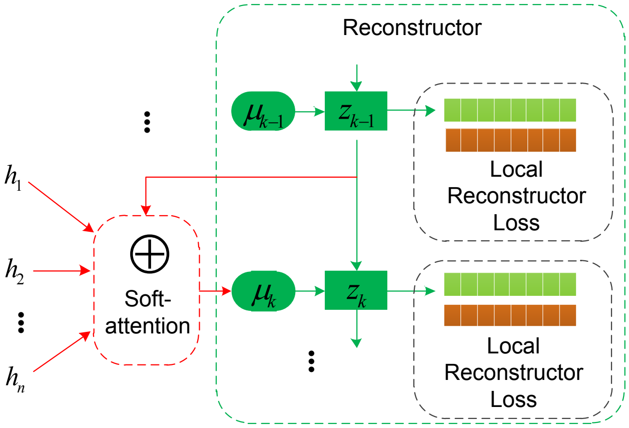


Figure 4. An illustration of the proposed reconstructor that reproduces the local structure of the video sequence. The reconstructor works on the hidden states of the decoder by selectively adjusting the attention weight, and reproduces the feature representation frame by frame.

$$\mathcal{L}(\theta, \theta_{rec}) = \sum_{i=1}^{N} \Big( \underbrace{-\log P\left(\mathbf{S}^i | \mathbf{V}^i; \theta\right)}_{\text{encoder-decoder}} + \lambda \underbrace{\mathcal{L}_{rec}(\mathbf{V}^i, \mathbf{Z}^i; \theta_{rec})}_{\text{reconstructor}} \Big). \tag{11}$$

The reconstruction loss $\mathcal{L}_{rec}(\mathbf{V}^i, \mathbf{Z}^i; \theta_{rec})$ can be realized by the global loss in Eq. 7 or the local loss in Eq. 10. The hyper-parameter $\lambda$ is introduced to seek a trade-off between the encoder-decoder and the reconstructor.

The training of our proposed RecNet model proceeds in two stages. First, we rely on the forward likelihood to train the encoder-decoder component of the RecNet, which is terminated by the early stopping strategy. Afterwards, the reconstructor and the backward reconstruction loss $\mathcal{L}_{rec}(\theta_{rec})$ are introduced. We use the whole loss defined in Eq. 11 to jointly train the reconstructor and fine-tune the encoder-decoder. For the reconstructor, the reconstruction loss is calculated using the hidden state sequence generated by the LSTM units in the reconstructor as well as the video frame feature sequence.

## 4. Experimental Results

In this section, we evaluate the proposed video captioning method on benchmark datasets such as Microsoft Research video to text (MSR-VTT) [46] dataset and Microsoft Research Video Description Corpus (MSVD) [4]. To compare with existing work, we compute the popular metrics including BLEU-4 [26], METEOR [3], ROUGE-L [18], and CIDEr [38] with the codes released on the Microsoft COCO evaluation server [6].

## 4.1. Datasets and Implementation Details

**MSR-VTT.** It is the largest dataset for video captioning so far in terms of the number of video-sentence pairs and the vocabulary size. In the experiments, we use the initial version of MSR-VTT, referred as MSR-VTT-10K, which contains 10K video clips from 20 categories. Each video clip is annotated with 20 sentences by 1327 workers from Amazon Mechanical Turk. Therefore, the dataset results in a total of 200K clip-sentence pairs and 29,316 unique words. We use the public splits for training and testing, *i.e.*, 6513 for training, 497 for validation, and 2990 for testing.

**MSVD.** It contains 1970 YouTube short video clips, and each one depicts a single activity in 10 seconds to 25 seconds. and each video clip has roughly 40 English descriptions. Similar to the prior work [24, 49], we take 1200 video clips for training, 100 clips for validation and 670 clips for testing.

For the sentences, we remove the punctuations, split them with blank space and convert all words into lowercase. The sentences longer than 30 are truncated, and the word embedding size for each word is set to 468.

For the encoder, we feed all frames of each video clip into Inception-V4 [35] which is pretrained on the ILSVRC-2012-CLS [31] classification dataset for feature extraction after resizing them to the standard size of 299 × 299, and extract the 1536 dimensional semantic feature of each frame from the last pooling layer. Inspired by [49], we choose the equally-spaced 28 features from one video, and pad them with zero vectors if the number of features is less than 28. The input dimension of the decoder is 468, the same to that of the word embedding, while the hidden layer contains 512 units. For the reconstructor, the inputs are the hidden states of the decoder and thus have the dimension of 512. To ease the reconstruction loss computation, the dimension of the hidden layer is set to 1536 same to that of the frame features produced by the encoder.

During the training, the AdaDelta [51] is employed for optimization. The training stops when the CIDEr value on the validation dataset stops increasing in the following 20 successive epochs. In the testing, beam search with size 5 is used for the final caption generation.

## 4.2. Study on the Encoder-Decoder

In this section, we first test the impacts of different encoder-decoder architectures in video captioning, such as SA-LSTM and MP-LSTM. Both are popular encoder-decoder models and share similar LSTM structure, except that SA-LSTM introduced an attention mechanism to aggregate frame features, while MP-LSTM relies on the mean pooling. As shown in Table 1, with the same encoder VGG19, SA-LSTM yielded 35.6 and 25.4 on the BLEU-4 and METEOR respectively, while MP-LSTM only produced 34.8 and 24.7, respectively. The same results can be obtained when using AlexNet and GoogleNet as the encoder. Hence, it is concluded that exploiting the temporal dynamics among frames with attention mechanism performed better in sentence generation than mean pooling on the whole video.

Table 1. Performance evaluation of different video captioning models on the testing set of the MSR-VTT dataset (%). The encoder-decoder framework is equipped with different CNN structures such as AlexNet, GoogleNet, VGG19 and Inception-V4. Except Inception-V4, the metric values of the other published models are referred from the work in [47], and the symbol "-" indicates that such metric is unreported.

| Model | BLEU-4 | METEOR | ROUGE-L | CIDEr |
|---|---|---|---|---|
| MP-LSTM (AlexNet) | 32.3 | 23.4 | - | - |
| MP-LSTM (GoogleNet) | 34.6 | 24.6 | - | - |
| MP-LSMT (VGG19) | 34.8 | 24.7 | - | - |
| SA-LSTM (AlexNet) | 34.8 | 23.8 | - | - |
| SA-LSTM (GoogleNet) | 35.2 | 25.2 | - | - |
| SA-LSTM (VGG19) | 35.6 | 25.4 | - | - |
| SA-LSTM (Inception-V4) | 36.3 | 25.5 | 58.3 | 39.9 |
| RecNet$_{global}$ | 38.3 | 26.2 | 59.1 | 41.7 |
| RecNet$_{local}$ | **39.1** | **26.6** | **59.3** | **42.7** |

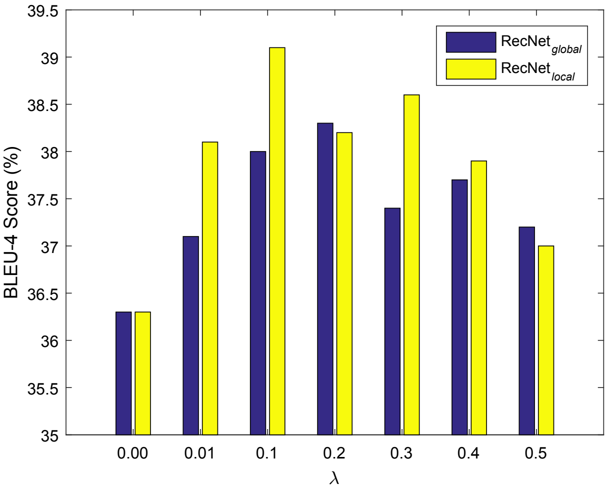
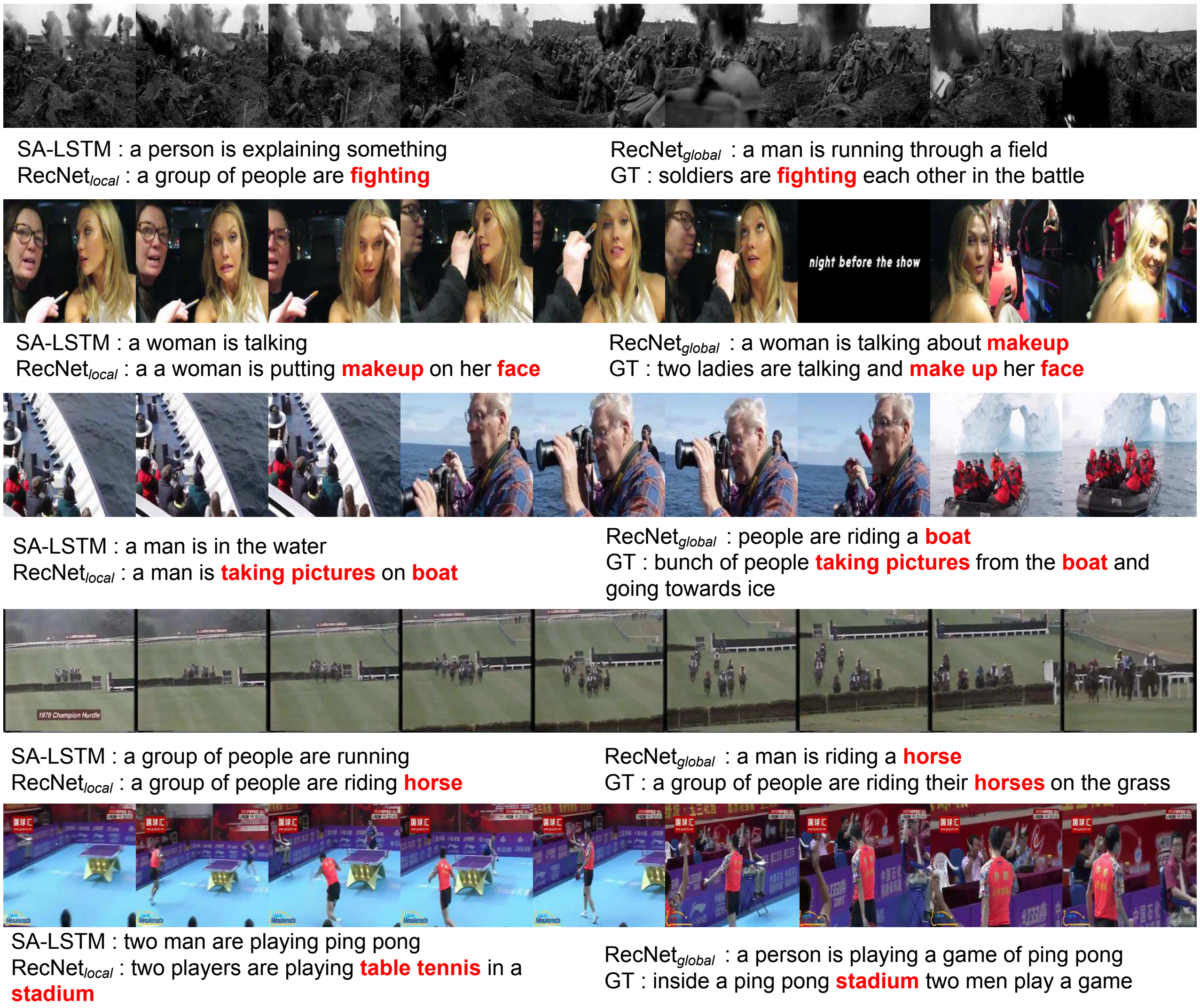


Figure 5. Effects of the trade-off parameter $\lambda$ for RecNet$_{global}$ and RecNet$_{local}$ in terms of BLEU-4 metric on MSR-VTT. It is noted that $\lambda = 0$ means the reconstructor is off, and the RecNet turns to be a conventional encoder-decoder model.

Besides, we also introduced Inception-V4 as an alternative CNN for feature extraction in the encoder. It is observed that with the same encoder-decoder structure SA-LSTM, Inception-V4 yielded the best captioning performance comparing to the other CNNs such as AlexNet, GoogleNet, and VGG19. This is probably because Inception-V4 is a deeper network and better at se-



SA-LSTM : a person is explaining something
RecNet$_{local}$ : a group of people are **fighting**
RecNet$_{global}$ : a man is running through a field
GT : soldiers are **fighting** each other in the battle

SA-LSTM : a woman is talking
RecNet$_{local}$ : a a woman is putting **makeup** on her **face**
RecNet$_{global}$ : a woman is talking about **makeup**
GT : two ladies are talking and **make up** her **face**

SA-LSTM : a man is in the water
RecNet$_{local}$ : a man is **taking pictures** on **boat**
RecNet$_{global}$ : people are riding a **boat**
GT : bunch of people **taking pictures** from the **boat** and going towards ice

SA-LSTM : a group of people are running
RecNet$_{local}$ : a group of people are riding **horse**
RecNet$_{global}$ : a man is riding a **horse**
GT : a group of people are riding their **horses** on the grass

SA-LSTM : two man are playing ping pong
RecNet$_{local}$ : two players are playing **table tennis** in a **stadium**
RecNet$_{global}$ : a person is playing a game of ping pong
GT : inside a ping pong **stadium** two men play a game

Figure 6. Visualization of some video captioning examples on the MSR-VTT dataset with different models. Due to the page limit, only one ground-truth sentence is given as reference. Compared to SA-LSTM, the proposed RecNet is able to yield more vivid and descriptive words highlighted in red boldface, such as "`fighting`", "`makeup`", "`face`", and "`horse`".

mantic feature extraction. Hence, SA-LSTM equipped with Inception-V4 is employed as the encoder-decoder model in the proposed RecNet.

By adding the global or local reconstructor to the encoder-decoder model SA-LSTM, we can have the proposed encoder-decoder-reconstructor architecture: RecNets. Apparently, such structure provided significant gains to the captioning performance in all metrics. This proved the backward flow information introduced by the proposed reconstructor could encourage the decoder to embed more semantic information and also regularize the generated caption to be more consistent with the video contents. More discussion about the proposed reconstrucutor will be given in Section 4.4.

### 4.3. Study on the Trade-off Parameter $\lambda$

In this section, we discuss the influence of the trade-off parameter $\lambda$ in Eq. 11. With different $\lambda$ values, the obtained BLEU-4 metric values are given in Figure 5. First, it can be concluded again that adding the reconstruction loss ($\lambda > 0$) did improve the performance of video captioning in terms of BLEU-4. Second, there is a trade-off between the forward likelihood loss and the backward reconstruction loss, as too large $\lambda$ may incur noticeable deterioration in caption performance. Thus, $\lambda$ needs to be more carefully selected to balance the contributions of the encoder-decoder and the reconstructor. As shown in Figure 5, we empirically set $\lambda$ to 0.2 and 0.1 for RecNet$_{global}$ and RecNet$_{local}$, respectively.

### 4.4. Study on the Reconstructors

The difference of the proposed two reconstructors is discussed in this section. The quantitative results of RecNet$_{local}$ and RecNet$_{global}$ on MSR-VTT are given on the bottom two rows of Table 1. It can be observed that RecNet$_{local}$ performs slightly better than RecNet$_{global}$. The reason mainly lies in the temporal dynamic modeling. RecNet$_{global}$ employs mean pooling to reproduce the video representation and misses the local temporal dynamics, while the attention mechanism is included in RecNet$_{local}$ to exploit the local temporal dynamics for each frame reconstruction.

However, the performance gap between RecNet$_{global}$ and RecNet$_{local}$ is not significant. One possible reason is that the visual information of frames is very similar. As the video clips of MSR-VTT are short, the visual representations of frames have few differences with each other, that is the global and local structure information is similar. Another possible reason is the complicated video-sentence relationship, which may lead to similar input information for RecNet$_{global}$ and RecNet$_{local}$.

### 4.5. Qualitative Analysis

Besides, some qualitative examples are shown in Fig. 6. Still, it can be observed that the proposed RecNets with local and global reconstructors generally produced more accurate captions than the typical encoder-decoder model SA-LSTM. For example, in the second example, SA-LSTM generated "`a woman is talking`", which missed the core subject of the video, *i.e.*, "`makeup`". By contrast, the captions produced by RecNet$_{global}$ and RecNet$_{local}$ are "`a woman is talking about makeup`" and "`a women is putting makeup on her face`", which apparently are more accurate. RecNet$_{local}$ even generated the word of "`face`" which results in a more descriptive caption. More results can be found in the supplementary file.

### 4.6. Evaluation on the MSVD Dataset

Finally, we tested the proposed RecNet on the MSVD dataset [4], and compared it to more benchmark encoder-decoder models, such as GRU-RCN[2], HRNE[23], h-RNN[50], LSTM-E[24], aLSTMs[9] and LSTM-LS[19]. The quantitative results are given in Table 2. It is observed that the RecNet$_{local}$ and RecNet$_{global}$ with SA-LSTM performed the best and second best in all metrics, respectively. Besides, we introduced the reconstructor to S2VT[39] to build another encoder-decoder-reconstructor model. The results show that both global and local reconstructors bring improvement to the original S2VT in all metrics, which again demonstrate the benefits of video captioning based on bidirectional cue modeling.

Table 2. Performance evaluation of different video captioning models on the MSVD dataset in terms of BLEU-4, METEOR, ROUGE-L, and CIDEr scores (%). The symbol "-" indicates such metric is unreported.

| Model | BLEU-4 | METEOR | ROUGE-L | CIDEr |
|---|---|---|---|---|
| MP-LSTM (AlexNet)[40] | 33.3 | 29.1 | - | - |
| GRU-RCN[2] | 43.3 | 31.6 | - | 68.0 |
| HRNE[23] | 43.8 | 33.1 | - | - |
| LSTM-E[24] | 45.3 | 31.0 | - | - |
| LSTM-LS (VGG19)[19] | 46.5 | 31.2 | - | - |
| h-RNN[50] | 49.9 | 32.6 | - | 65.8 |
| aLSTMs [9] | 50.8 | 33.3 | - | 74.8 |
| S2VT (Inception-V4) | 39.6 | 31.2 | 67.5 | 66.7 |
| SA-LSTM (Inception-V4) | 45.3 | 31.9 | 64.2 | 76.2 |
| RecNet$_{global}$ (S2VT) | 42.9 | 32.3 | 68.5 | 69.3 |
| RecNet$_{local}$ (S2VT) | 43.7 | 32.7 | 68.6 | 69.8 |
| RecNet$_{global}$ (SA-LSTM) | 51.1 | 34.0 | 69.4 | 79.7 |
| RecNet$_{local}$ (SA-LSTM) | **52.3** | **34.1** | **69.8** | **80.3** |

## 5. Conclusions

In this paper, we proposed a novel RecNet with the encoder-decoder-reconstructor architecture for video captioning, which exploits the bidirectional cues between natural language description and video content. Specifically, to address the backward information from description to video, two types of reconstructors were devised to reproduce the global and local structures of the input video, respectively. The forward likelihood and backward reconstruction losses were jointly modeled to train the proposed network. The experimental results on the benchmark datasets corroborate the superiority of the proposed RecNet over the existing encoder-decoder models in video caption accuracy.

## Acknowledgments

The authors would like to thank the anonymous reviewers for the constructive comments to improve the paper. This work was supported by the NSFC Grant no. 61573222, Shenzhen Future Industry Special Fund JCYJ20160331174228600, Major Research Program of Shandong Province 2015ZDXX0801A02, National Key Research and Development Plan of China under Grant 2017YFB1300205 and Fundamental Research Funds of Shandong University 2016JC014.

## References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[2] N. Ballas, L. Yao, C. Pal, and A. Courville. Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432*, 2015.

[3] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, volume 29, pages 65–72, 2005.

[4] D. L. Chen and W. B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 190–200. Association for Computational Linguistics, 2011.

[5] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*, 2017.

[6] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[7] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[8] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[9] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen. Video captioning with attention-based lstm and semantic consistency. *IEEE Transactions on Multimedia*, 2017.

[10] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, 2013.

[11] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T. Liu, and W.-Y. Ma. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pages 820–828, 2016.

[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[13] W. Jiang, L. Ma, X. Chen, H. Zhang, and W. Liu. Learning to guide decoding for image captioning. In *AAAI*, 2018.

[14] Q. Jin, J. Chen, S. Chen, Y. Xiong, and A. Hauptmann. Describing videos using multi-modal fusion. In *ACM MM*, 2016.

[15] A. Karpathy, A. Joulin, and F. F. F. Li. Deep fragment embeddings for bidirectional image sentence mapping. In *NIPS*, 2014.

[16] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. Li. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[17] A. Kojima, T. Tamura, and K. Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *IJCV*, 50(2):171–184, 2002.

[18] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004.

[19] Y. Liu, X. Li, and Z. Shi. Video captioning with listwise supervision. In *AAAI*, 2017.

[20] P. Luo, G. Wang, L. Lin, and X. Wang. Deep dual learning for semantic image segmentation. In *CVPR*, pages 2718–2726, 2017.

[21] L. Ma, Z. Lu, and H. Li. Learning to answer questions from image using convolutional neural network. In *AAAI*, volume 3, page 16, 2016.

[22] L. Ma, Z. Lu, L. Shang, and H. Li. Multimodal convolutional neural networks for matching image and sentence. In *Proceedings of the IEEE international conference on computer vision*, pages 2623–2631, 2015.

[23] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*, pages 1029–1038, 2016.

[24] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016.

[25] Y. Pan, T. Yao, H. Li, and T. Mei. Video captioning with transferred semantic attributes. *arXiv preprint arXiv:1611.07675*, 2016.

[26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[27] V. Ramanishka, A. Das, D. H. Park, S. Venugopalan, L. A. Hendricks, M. Rohrbach, and K. Saenko. Multimodal video description. In *ACM MM*, 2016.

[28] Z. Ren, X. Wang, N. Zhang, X. Lv, and L.-J. Li. Deep reinforcement learning-based image captioning with embedding reward. *arXiv preprint arXiv:1704.03899*, 2017.

[29] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele. Coherent multi-sentence video description with variable level of detail. In *GCPR*, 2014.

[30] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In *ICCV*, 2013.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[32] Z. Shen, J. Li, Z. Su, M. Li, Y. Chen, Y.-G. Jiang, and X. Xue. Weakly supervised dense video captioning. In *CVPR*, 2017.

[33] J. Song, L. Gao, L. Liu, X. Zhu, and N. Sebe. Quantization-based hashing: a general framework for scalable image and video retrieval. *Pattern Recognition*, 2017.

[34] J. Song, Z. Guo, L. Gao, W. Liu, D. Zhang, and H. T. Shen. Hierarchical lstm with adjusted temporal attention for video captioning. *arXiv preprint arXiv:1706.01231*, 2017.

[35] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.

[36] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3D: generic features for video analysis. *CoRR*, abs/1412.0767, 2014.

[37] Z. Tu, Y. Liu, L. Shang, X. Liu, and H. Li. Neural machine translation with reconstruction. In *AAAI*, pages 3097–3103, 2017.

[38] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.

[39] S. Venugopalan, M. Rohrbach, J. Donahue, R. J. Mooney, T. Darrell, and K. Saenko. Sequence to sequence - video to text. In *ICCV*, 2015.

[40] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL*, 2015.

[41] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, June 2015.

[42] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *TPAMI*, 39(4):652–663, 2017.

[43] V. Voykinska, S. Azenkot, S. Wu, and G. Leshed. How blind people interact with visual content on social networking services. pages 1584–1595, 2016.

[44] J. Wang, T. Zhang, N. Sebe, H. T. Shen, et al. A survey on learning to hash. *TPAMI*, 2017.

[45] Y. Xia, T. Qin, W. Chen, J. Bian, N. Yu, and T.-Y. Liu. Dual supervised learning. *arXiv preprint arXiv:1707.00415*, 2017.

[46] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016.

[47] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language [supplementary material]. CVPR, October 2016.

[48] R. Xu, C. Xiong, W. Chen, and J. J. Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, 2015.

[49] L. Yao, A. Torabi, K. Cho, N. Ballas, C. J. Pal, H. Larochelle, and A. C. Courville. Describing videos by exploiting temporal structure. In *ICCV*, 2015.

[50] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, pages 4584–4593, 2016.

[51] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[52] C. Zhang and Y. Tian. Automatic video description generation via lstm with joint two-stream encoding. In *ICPR*, 2016.

[53] W. Zhang, X. Yu, and X. He. Learning bidirectional temporal cues for video-based person re-identification. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.