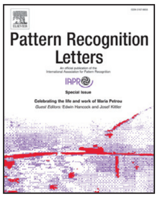
Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec



Sequence in sequence for video captioning

Huiyun Wang, Chongyang Gao, Yahong Han*

School of Computer Science and Technology, Tianjin University, Tianjin 300350, China

ARTICLE INFO

Article history:
Available online xxx

Keywords:
Video captioning
Encoding
Decoding
Spatio-temporal representation

ABSTRACT

For video captioning, the words in the caption are closely related to an overall understanding of the video. Thus, a suitable representation for the video is rather important for the description. For more precise words in the task of video captioning, we aim to encode the video feature for current word at each time-stamp of the generation process. This paper proposes a new framework of 'Sequence in Sequence' to encode the sequential frames into a spatio-temporal representation at each time-stamp to utter a word and further distill most related visual content by an extra semantic loss. First, we aggregate the sequential frames to extract related visual content guided by last word, and get a representation with rich spatio-temporal information. Then, to decode the aggregated representation for a precise word, we leverage two layers of GRU structure, where the first layer further distills useful visual content based on an extra semantic loss and the second layer selects the correct word according to the distilled features. Experiments on two benchmark datasets demonstrate that our method outperforms the current state-of-the-art methods on Bleu@4, METEOR and CIDEr metrics.

1. Introduction

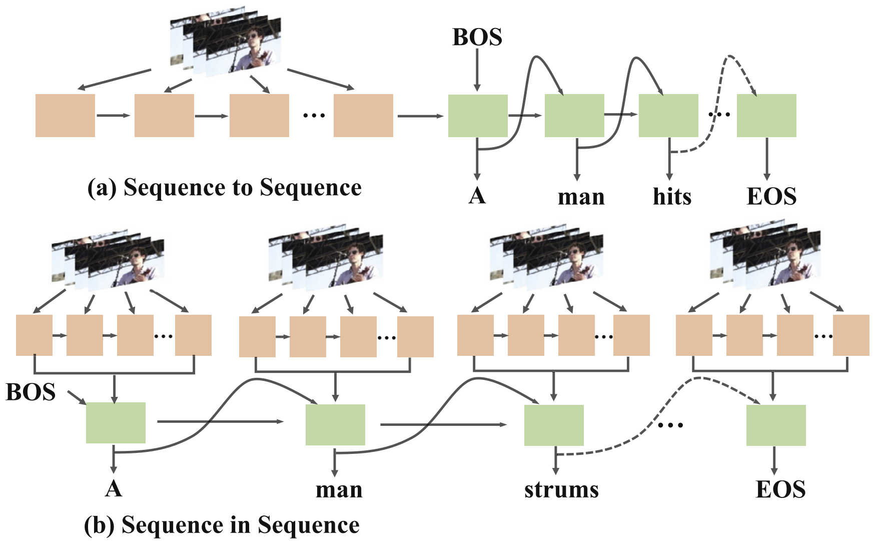
Automatically generating a natural language description for the video, called video captioning, refers to a summarization of the input video based on visual content understanding. Widespread applications, e.g., video indexing, human-robot interaction, and video descriptions for the visually impaired, may benefit much from good video descriptions. Thus it attracts much attention in computer vision community [14,31,34,35,37,41]. Due to rich and open-domain activities in visual content, video captioning remains a challenging task. Inspired by the successful use of the encoder-decoder framework in machine translation [2,11,30] and the development of deep learning, most video captioning methods [13,26,27,38,39] are sequence-to-sequence models based on the encoder-decoder framework. Particularly, the encoder first utilizes Convolutional Neural Networks (CNN) to extract representations for static frames and the representations of all frames are stacked with a Recurrent Neural Networks (RNN) to form the video representation, and then the decoder utilizes another RNN to generate natural language descriptions.

To summarize the visual content into a meaningful natural language sentence, the captioning model must be able to represent the sequential frames of the video into a spatio-temporal feature with rich visual information to express the objects, actions and scenes for each word in the generated sentence. To model dynamic temporal structure of the video, several works [3,34,39,41] first encode the representations of the frames from CNN one by one in sequence before the decoder. Although the method in [34] represents the global temporal interaction of actors and objects that evolve over time, it ignores the local temporal structure of the video. To solve the problem, the method [39] leverages a 3-D CNN [19,21] to encode the local temporal structure and a temporal attention mechanism to exploit global temporal structure. And the model [3] applies the convolution operation to the GRU-RNN model [2] for preserving the frame spatial topology, and meantime catch the temporal information. However, due to the different speeds of motions in the video, the methods above can only model temporal information in short videos. The method in [41] proposes a Multirate Visual Recurrent Model which adopts multiple encoding rates, and thus obtain a multirate representation which is robust to motion speed variance in videos. To express the static and dynamic information for sequential words, all the above methods attempt to obtain a representation of the video which expresses the spatial and temporal information.

However, in the generation process of the sentence, the models first encode the video features before the encoder generates the description of video. Thus, as the input of the decoder, the visual representation remains the same at each time-stamp, which is unreasonable for the generation of the words with different meaning. For example, in Fig. 1 (a), at the time of uttering the word 'strums', 'Sequence to Sequence' models generate the wrong word

* Corresponding author.
E-mail address: yahong@tju.edu.cn (Y. Han).

https://doi.org/10.1016/j.patrec.2018.07.024
0167-8655/© 2018 Elsevier B.V. All rights reserved.



(a) Sequence to Sequence

(b) Sequence in Sequence

Fig. 1. Comparison of 'Sequence to Sequence' and 'Sequence in Sequence'. (a) 'Sequence to Sequence' first reads the sequence of frames before the process of caption generation. (b) 'Sequence in Sequence' encodes the features of all the sequential frames to utter a word. The ground truth sentence of the video: a man strums a violin on a stage.

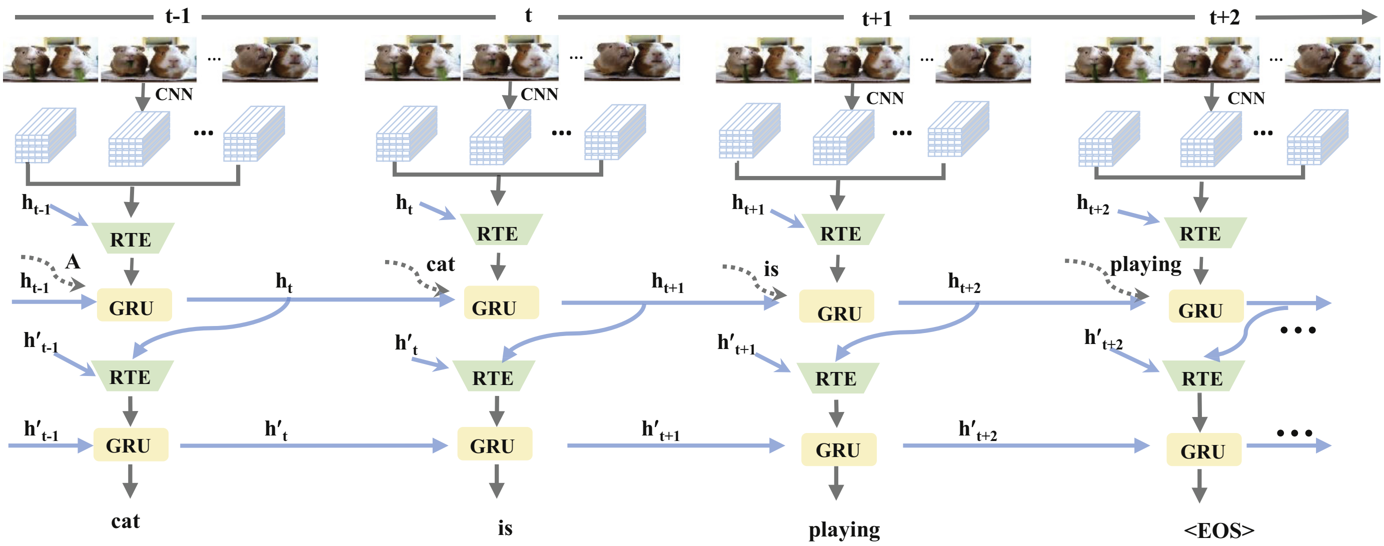


Fig. 2. The framework of our method. 'RTE' represents our proposed encoder: 'Real-time Encoder'. h_{t-1}, h_t, h_{t+1} and h_{t+2} are the hidden states of the GRU in the first layer. h'_{t-1}, h'_t, h'_{t+1} and h_{t+2} are the hidden states of the GRU in the second layer.

"hits" with improper understanding of related visual content in all the frames.

In order to understand the related visual content accurately for uttering a word, we propose a novel framework of 'Sequence in Sequence' (SeqInSeq) by encoding the sequential frames at each time-stamp to generate the word in the sentence and aggregating video feature into a spatio-temporal representation by our proposed 'Real-time Encoder' (RTE). The RTE learns the spatio-temporal representation by a trainable process and takes the history information by the current time-stamp to ensure more related visual information for the word in the caption. The architecture of SeqInSeq is outlined in Fig. 2. First, we extract features of convolutional layer using CNN for the sequential frames of video with temporal dependency. Next, at each time-stamp, according to the generated word at last time-stamp and visual content, we aggregate the video feature into a spatio-temporal representation to focus on the visual information which is most related to the current word. Then, to get more related visual information for a precise word, we leverage two layers of GRU structure to decode the aggregated representation, where the first layer further distills useful visual content based on an extra semantic loss and the second layer selects the correct word according to the distilled features from the first layer.

The contributions of our proposed model are summarized below: (1) We propose a novel framework to utter more precise words by encoding the video feature at each time-stamp and adding an extra semantic loss to focus on the most related visual content; (2) A new encoder of SeqInSeq is proposed to aggregate the video into a spatio-temporal representation according to the generated word at last time-stamp and visual content, which is able to incorporate more related visual information for the current generated word; (3) We conduct extensive experiments on two standard benchmark datasets for video captioning, and experimental results achieve state-of-the-art performance on both datasets.

Please cite this article as: H. Wang et al., Sequence in sequence for video captioning, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.07.024

2. Related work

The task of describing video content with natural language poses an important challenge due to large amounts of related applications. But it remains a challenging task due to complex interactions of actors and objects in the video. In previous works [14,23,31] identified $<$subject, verb, object$>$ triplets with visual classifiers and then generated caption with predefined sentences model with fixed length, which could not satisfy the richness of natural language.

Due to the remarkable development of CNN, Venugopalan et al. [35] first utilized a mean pooling over the extracted CNN features for the video representation, and then fed the video representation into the decoder for caption generation. A video consists of sequential frames with much temporal dependency. However, due to this indiscriminate averaging of all the frames, this approach risks ignoring much of the temporal structure underlying the video. To solve this problem, Yao et al. [39] proposed a temporal attention mechanism to automatically select the most relevant temporal segments for word prediction, which used a weighted sum of frame-level CNN features. And it indicates that exploring the temporal structure in the video is helpful for a better caption. Therefore, the following methods always attempted to model the temporal structure in the video.

The progresses [2,11,30] in machine translation prompt the study in video captioning to the use of (RNN), which succeed in outputting a variable-length sequence of words based on the input of a vector representation of visual content [20,36]. To directly map a sequence of frames to a sequence of words, in model [34] an encoder (a stacked LSTM) first encoded the representation of the frames from CNN one by one to catch the temporal information in the video, and then generated a sequence of words. Videos are spatio-temporal extension of images, how to model the video into a discriminative representation is rather challenging. For example, event detection [5,7,24,25] as an important part of video analysis tried to describe the video content with a discriminative representation. Besides, saliency detection [10,15,16] also focused on distilling important visual content to represent the video.

To model the temporal aspects of activities typically shown in the video, the method [26] utilized 3D ConvNets for a video representation to catch action information, and proposed a hierarchical recurrent encoder to model temporal dependencies among the actions in the video. By modeling different speed of motions in video, the method in [41] proposed a Multirate Visual Recurrent Model which adopted multiple encoding rates to obtain a multirate representation. To leverage a higher spatial resolution in low-level representations, the model [3] applied the convolution operation to GRU-RNN model [11] for preserving the frame spatial topology, and meantime catch the temporal information. The method in [6] tried to generate semantic representation of the video by using external image/video archives and applying the concept detectors trained on them to the event videos. Above attempts just obtain a static representation of the video and input it to the decoder, however, the same video representation retains the same at all time-stamps of the generation process. Due to the different semantic meaning in the words, it is reasonable to encode the video feature according to the word to utter. In the process of the generation of the whole sentence, the models encode the video features once and remains the same representation as the input of decoder at each time-stamp, which is unreasonable for the generation of the words with different meaning.

3. The proposed approach

For a precise description of the video, We first design a ‘Real-time Encoder’ which aggregates the video feature into a specific visual descriptor for the current generated word. Then we utilize a two-layer GRU where the first layer operates at a lower semantic level to narrow the range of related visual areas and the second layer focuses on the correct word incorporating the narrowed range, visual content from the ‘Real-time Encoder’ and the guidance of last word.

3.1. The real-time encoder

Given an input video with N frames, we denote feature maps extracted from convolutional layer of CNN as $X = \{x_1, x_2, \ldots, x_N\}$, $x_n \in \mathbb{R}^{H\times W\times D}$ and $n \in \{1, \ldots, N\}$, where H, W, and D denote the height, the width, and the number of channels in the convolutional feature map, respectively.

Video captioning usually utilizes the temporal information of successive frames and the spatial information of each frame to model the temporal and spatial structure of the video. Thus, we need to encode the video feature into a discriminative spatio-temporal representation containing rich visual content. Inspired by [18] which represents the static image by a locally aggregated descriptors (VLAD), we encode the sequential frames into a sequence of VLAD representations. The VLAD encoding learns a codebook $C \in \mathbb{R}^{D\times K} = \{c_1, c_2, \ldots, c_K\}$ of visual cluster centers. The key point of the VLAD encoding is to map $H \times W \times D$-dimensional local image descriptors x_n to the nearest cluster center c_k, thus the output is $K \times D$-dimension which stores the sum of residuals (difference vector between each video descriptor and its nearest cluster center).

The assignment of the descriptors to cluster centers is vital to map visual content into a rich representation. Inspired by NetVLAD [1] which utilizes a 1×1 convolution layer to compute the assignment between the local descriptors and corresponding centers, we utilize the Convolutional GRU to compute the assignment to investigate the spatio-temporal correlation between the successive frames at each time-stamp. The inputs of Convolutional GRU are $x_n \in \mathbb{R}^{H\times K\times D}$ and α_{n-1}, which denote the nth video feature and the assignment at time step $n-1$, respectively. The Convolutional GRU is computed as follows:

$$z_n = \sigma(\boldsymbol{W_z} * x_n + \boldsymbol{U_z} * \alpha_{n-1}), \tag{1}$$

$$r_n = \sigma(\boldsymbol{W_r} * x_n + \boldsymbol{U_r} * \alpha_{n-1}), \tag{2}$$

$$\tilde{\alpha}_n = tanh(\boldsymbol{W} * x_n + \boldsymbol{U} * (r_n \odot \alpha_{n-1})), \tag{3}$$

$$\alpha_n = (1 - z_n) \odot \alpha_{n-1} + z_n \odot \tilde{\alpha}_n, \tag{4}$$

where $\boldsymbol{W_z}$, $\boldsymbol{W_r}$, $\boldsymbol{W}$, $\boldsymbol{U_z}$, $\boldsymbol{U_r}$ and $\boldsymbol{U}$ refer to 2D-convolutional kernels, $*$ denotes a convolution operation, σ denotes Sigmoid function, $\odot$ is element-wise multiplication, z_t and r_t denote the update gate and reset gate, respectively. In this paper, we share the parameters of $\boldsymbol{W_z}$, $\boldsymbol{W_r}$, $\boldsymbol{W}$.

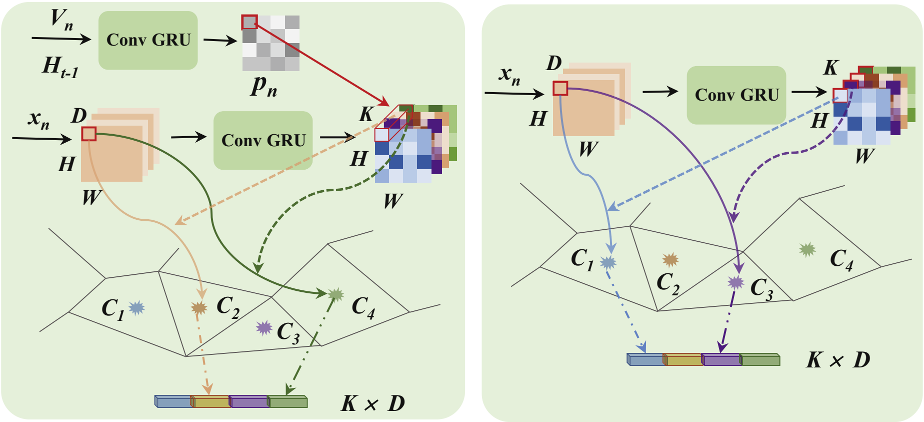
To get a video representation suitable for the word generated at current time-stamp, we use the hidden state of last time-stamp H_{t-1} in the decoder and visual content V_n to guide the process of video encoding. Thus, we calculate the weights of the assignment on cluster centers, that is, the contribution of the same visual area changes for different words. The weight is calculated as follows:

$$e_n = \mathcal{F}(V_n, H_{t-1}), \tag{5}$$

$$z'_n = \sigma(\boldsymbol{W_{az}} * e_n + \boldsymbol{U_{az}} * \beta'_{n-1}), \tag{6}$$

$$r'_n = \sigma(\boldsymbol{W_{ar}} * e_n + \boldsymbol{U_{ar}} * \beta'_{n-1}), \tag{7}$$

Please cite this article as: H. Wang et al., Sequence in sequence for video captioning, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.07.024



(a) Real-time Encoder (RTE)

(b) Seq Encoder

Fig. 3. Comparison of ‘Seq Encoder’ and ‘Real-time Encoder’. (a) ‘Real-time Encoder’ aggregates the features of all the sequential frames based on last word and visual content. (b) ‘Seq Encoder’ denotes the encoding which encodes video features only once before the decoder.

$$\widetilde{\beta}'_t = tanh(\boldsymbol{W_a} * e_n + \boldsymbol{U_a} * (r'_n \odot \beta'_{n-1})), \tag{8}$$

$$\beta'_n = (1 - z'_n) \odot \beta'_{n-1} + z'_n \odot \widetilde{\beta}'_n, \tag{9}$$

$$p_n = W_e \beta'_n \tag{10}$$

where $\mathcal{F}$ denotes the concat operation, $\boldsymbol{W_{az}}$, $\boldsymbol{W_{ar}}$, $\boldsymbol{W_a}$, $\boldsymbol{U_{az}}$, $\boldsymbol{U_{ar}}$, $\boldsymbol{U_a}$ refer to 2D-convolutional kernels, $\widetilde{\beta}'_t$ and β'_n denote the update gate and reset gate, respectively. And we share the parameters of $\boldsymbol{W_{az}}$, $\boldsymbol{W_{ar}}$, $\boldsymbol{W_a}$. For the first layer of the decoder in Section 3.2.1, V_n and H_t denote x_n and h_{t-1}, respectively. For the second layer of the decoder in Section 3.2.2, V_n and H_t denote h_t and h'_{t-1}, respectively.

Once the assignment is computed, we define $C \in \mathbb{R}^{D \times K}$ as a trainable parameter. Next, for the n−th frame, we aggregate x_n into a spatio-temporal descriptor as $\mu_n(k)$ using VLAD encoding procedure:

$$\mu_n^k(t) = \sum_{i=1}^{H}\sum_{j=1}^{W} p_n(i,j)\alpha_n(i,j,k)(x_n(i,j) - c_k) \tag{11}$$

where c_k denotes the kth cluster center, $p_n(i, j)$ denotes the weight of location (i, j) on the kth cluster center, $\alpha_n(i, j, k) \in \mathbb{R}$ denotes the assignment value between the local descriptor at image location (i, j) of nth frame and kth cluster center c_k. $x_n(i, j) \in \mathbb{R}^D$ refers to local descriptor at image location (i, j). The output is $\mu(t) \in \mathbb{R}^{N \times D \times K}$, which indicates spatio-temporal representations.

As shown in Fig. 3, we visualize the aggregation methods in ‘Sequence to Sequence’ (Seq Encoder) and ‘Sequence in Sequence’ (Real-time Encoder). The ‘Seq Encoder’ aggregate the video features using Eqs. (1)–(4) and (11). And it feeds the output into the decoder directly. The ‘Real-time Encoder’ aggregates the features of all the sequential frames based on last word and visual content at each time-stamp to utter a word.

3.2. The decoder

In the decoding phase, our model has two layers and each layer has one GRU, where the first layer further distills useful visual content based on a low-level loss and the second layer select the correct word according to the guidance of visual information from the first layer.

3.2.1. The first layer

we first encode the video feature x_n through the RTE. For the encoding of first layer, the inputs of V_n and H_{t-1} are x_n and h_{t-1}, respectively. And the output of RTE is $\mu(t)$. Thus, the inputs of first layer are $\mu(t)$, w_t and h_{t-1}. The calculation of second layer is as follows:

$$z_t = \sigma(\boldsymbol{w_{vz}^{(1)}}\mu(t) + \boldsymbol{w_{dz}^{(1)}}w_t) + \boldsymbol{u_{dz}^{(1)}}h_{t-1}), \tag{12}$$

$$r_t = \sigma(\boldsymbol{w_{vr}^{(1)}}\mu(t) + \boldsymbol{w_{dr}^{(1)}}w_t + \boldsymbol{u_{dr}^{(1)}}h_{t-1}), \tag{13}$$

$$\widetilde{h}_t = tanh(\boldsymbol{w_v^{(1)}}\mu(t) + \boldsymbol{u_d^{(1)}}(r_t \odot h_{t-1})), \tag{14}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h}_t, \tag{15}$$

where $\boldsymbol{w_{vz}^{(1)}}$, $\boldsymbol{w_{vr}^{(1)}}$, $\boldsymbol{w_v^{(1)}}$, $\boldsymbol{w_{dz}^{(1)}}$, $\boldsymbol{w_{dr}^{(1)}}$, $\boldsymbol{u_{dz}^{(1)}}$, $\boldsymbol{u_{dr}^{(1)}}$ and $\boldsymbol{u_d^{(1)}}$ are trainable parameters.

We denote S_1 as a sequence of embedding words of the caption:

$$S_1 = \{w_1, w_2, \ldots, w_T\}, w_t \in \mathbb{R}^E \tag{16}$$

where E and T refer to the embedding size of the word and the length of the caption, respectively. And w_t represents the word generated at time step t.

Then a linear transformation layer is used to map h_t to the word space, and a softmax layer to produce a probability distribution over all the words in the vocabulary:

$$Pr(y_t^{(1)}|y_0^{(1)}, y_1^{(1)}, \ldots, y_{t-1}^{(1}, \mu(t)) \tag{17}$$

where $y_t^{(1)}$ denotes the tth word along the time. We use an extra semantic loss to distill the most related visual content. And the objective function is the sum of the negative log-likelihood of the generated words:

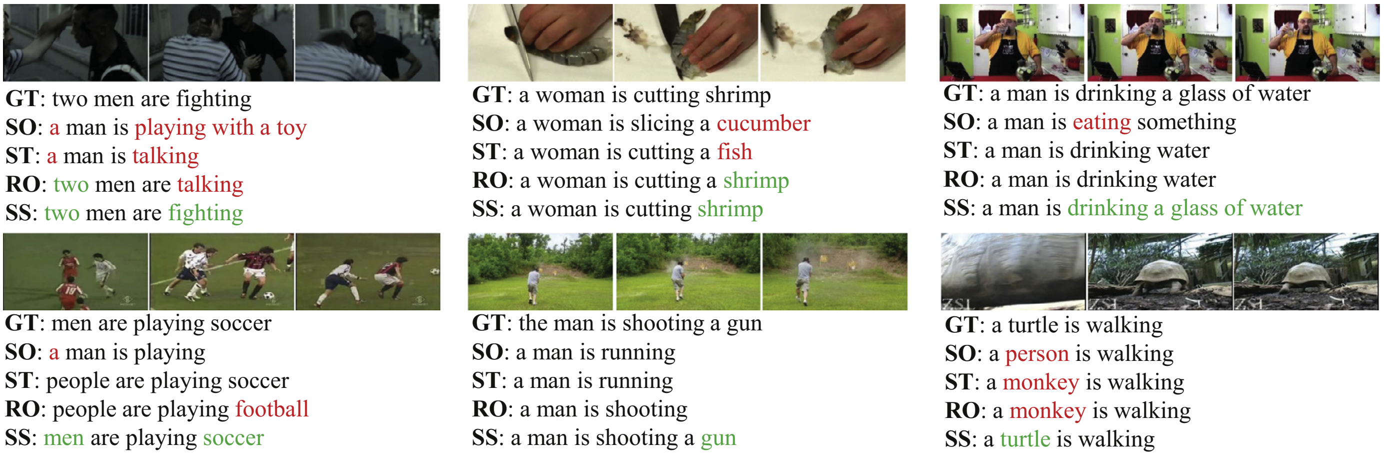
$$Loss1 = -\sum_{t=1}^{T} logPr(y_t^{(1)}|y_0^{(1)}, y_1^{(1)}, \ldots, y_{t-1}^{(1)}, \mu(t); \theta) \tag{18}$$

where θ represents all the parameters in the model.

3.2.2. The second layer

For the encoding of second layer, the inputs of V_n and H_{t-1} are h_t and h'_{t-1}, respectively. And the output of RTE is $\mu'(t)$. Thus, the

Please cite this article as: H. Wang et al., Sequence in sequence for video captioning, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.07.024



GT: two men are fighting
SO: a man is playing with a toy
ST: a man is talking
RO: two men are talking
SS: two men are fighting

GT: a woman is cutting shrimp
SO: a woman is slicing a cucumber
ST: a woman is cutting a fish
RO: a woman is cutting a shrimp
SS: a woman is cutting shrimp

GT: a man is drinking a glass of water
SO: a man is eating something
ST: a man is drinking water
RO: a man is drinking water
SS: a man is drinking a glass of water

GT: men are playing soccer
SO: a man is playing
ST: people are playing soccer
RO: people are playing football
SS: men are playing soccer

GT: the man is shooting a gun
SO: a man is running
ST: a man is running
RO: a man is shooting
SS: a man is shooting a gun

GT: a turtle is walking
SO: a person is walking
ST: a monkey is walking
RO: a monkey is walking
SS: a turtle is walking

Fig. 4. Caption generation results on the MSVD dataset. 'GT' refers to the ground truth. 'SO' denotes the method of Seq Encoder+1Layer Loss, 'ST' denotes the method of Seq Encoder+2Layer Loss, 'RO' denotes the method of RTE+1Layer Loss, and 'SS' denotes the method of SeqInSeq.

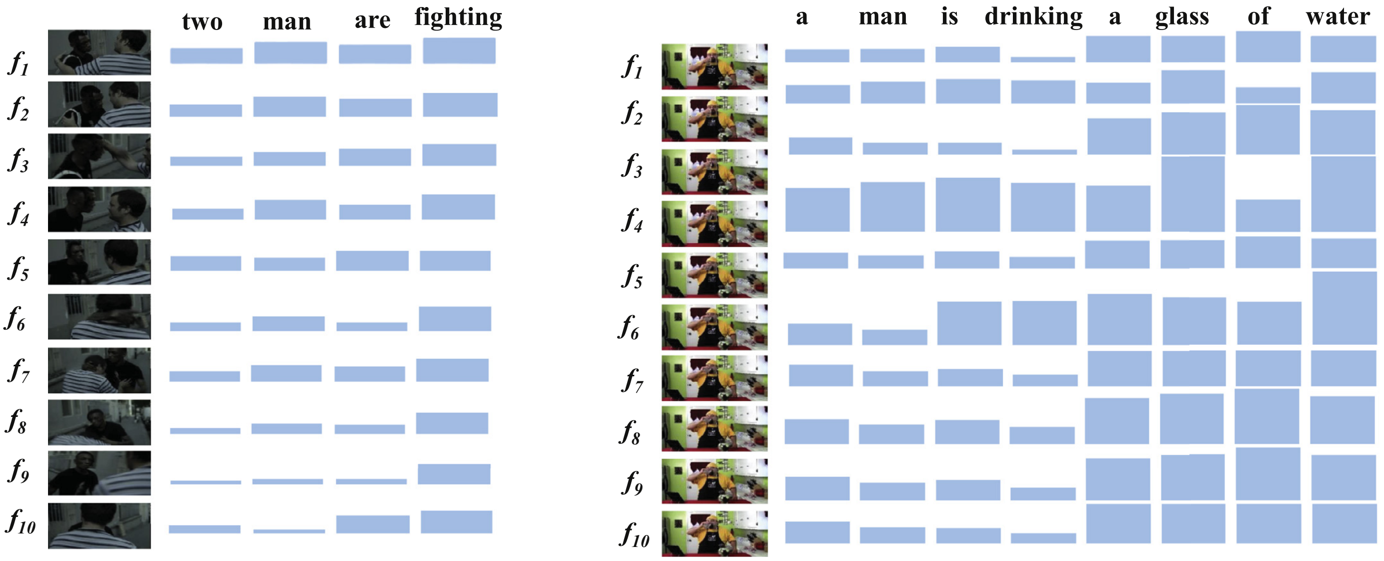


Fig. 5. The mean value of p_n in the generation of each word based on the model of SeqInSeq. The bar denotes the mean value of p_n. The higher the bar is, the nearer the descriptor is to the cluster center in the VLAD encoding.

inputs of second layer are $\mu'(t), w'_t$ and h'_{t-1}. The calculation of second layer is as follows: The input of second layer

$$z'_t = \sigma(\mathbf{w}^{(2)}_{vz}\mu'(t) + \mathbf{w}^{(2)}_{dz}w'_t) + \mathbf{u}^{(2)}_{dz}h'_{t-1}), \tag{19}$$

$$r'_t = \sigma(\mathbf{w}^{(2)}_{vr}\mu'(t) + \mathbf{w}^{(2)}_{dr}w'_t + \mathbf{u}^{(2)}_{dr}h'_{t-1}), \tag{20}$$

$$\widetilde{h'}_t = tanh(\mathbf{w}^{(2)}_{v}\mu'(t) + \mathbf{u}^{(2)}_{d}(r_t \odot h'_{t-1})), \tag{21}$$

$$h'_t = (1 - z_t) \odot h'_{t-1} + z_t \odot \widetilde{h'}_t, \tag{22}$$

where $\mathbf{w}^{(2)}_{vz}, \mathbf{w}^{(2)}_{vr}, \mathbf{w}^{(2)}_{v}, \mathbf{w}^{(2)}_{dz}, \mathbf{w}^{(2)}_{dr}, \mathbf{u}^{(2)}_{dz}, \mathbf{u}^{(2)}_{dr}$ and $\mathbf{u}^{(2)}_{d}$ are trainable parameters.

We denote S_2 as a sequence of embedding words of a caption:

$$S_2 = \left\{w'_1, w'_2, \dots, w'_T\right\}, w'_t \in \mathbb{R}^E \tag{23}$$

where E and T refer to the embedding size of the word and the length of the caption, respectively. And w'_t represents the word generated at time step t.

Then a linear transformation layer is applied after the GRU layer to map h'_t to the word space and a softmax layer to produce a probability distribution over all the words in the vocabulary. Thus the objective of generating the words is the sum of the negative log-likelihood of the captions:

$$Loss2 = -\sum_{t=1}^{T}(y^{(2)}_t | y^{(2)}_0, y^{(2)}_1, \dots, y^{(2)}_{t-1}, \mu'(t); \theta) \tag{24}$$

where $y^{(2)}_t$ denotes the tth word along the time and θ represents all the parameters in the model.

When training the model, $Loss1$ guides the process of sentence generation to distill related visual information as a low-level loss. Thus the whole loss function will be formulated as:

$$Loss = \lambda Loss1 + (1 - \lambda)Loss2 \tag{25}$$

where λ is the weight factor and trades off $Loss1$ and $Loss2$.

Please cite this article as: H. Wang et al., Sequence in sequence for video captioning, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.07.024

Table 1
Comparison of model variants on the MSVD dataset. All values are reported as percentage (%). '1Layer Loss' denotes the decoder with one layer of GRU structure. '2Layer Loss' denotes the decoder with two layers of GRU structure with one loss function in each layer.

Method	BLEU@4	METEOR	CIDEr
Seq Encoder+1Layer Loss	54.2	34.4	84.2
Seq Encoder+2Layer Loss	54.1	34.2	85.4
RTE+1Layer Loss	54.7	34.5	86.9
RTE+2Layer Loss(SeqInSeq)	**56.1**	**34.9**	**88.2**

Table 2
Experiment results on the MSVD dataset compared to the state-of-the-art methods. All values are reported as percentage (%). V, A, C, R, and G denote VGG, AlexNet, C3D, Resnet and GoogLeNet, respectively.

Method	BLEU@4	METEOR	CIDEr
SA [39] (C)	38.7	28.7	44.8
GRU-RCN [3] (G)	43.3	31.6	68.0
mGRU+pre-train [41] (R)	53.8	34.5	81.2
SA [39] (GC)	41.9	29.6	51.7
S2VT [34] (VA)	–	29.8	–
LSTM-E [27] (VC)	45.3	31.0	–
h-RNN [40] (VC)	49.9	32.6	65.8
HRNE [26] (GC)	46.7	33.9	–
DMRM [38] (G)	51.1	33.6	74.8
Boundary-aware [4] (RC)	42.5	32.4	63.5
SCN-LSTM [13] (RC)	51.1	33.5	77.7
LSTM-TSA [28] (VC)	52.8	33.5	74.8
SeqInSeq	**56.1**	**34.9**	**88.2**

Table 3
Runtime of our method and the state-of-the-art methods on MSVD dataset. All values are reported as second (s). V, C, R, and G denote VGG, C3D, Resnet and GoogLeNet, respectively.

Method	Runtime
SA [39] (GC)	23
h-RNN [40] (VC)	38
S2VT [34] (V)	18
LSTM-E [27] (VC)	29
DMRM [38] (G)	43
Boundary-aware [4] (RC)	42
SCN-LSTM [13] (RC)	45
SeqInSeq	196

4. Experiments

4.1. Datasets

We conduct extensive experiments on two benchmark video captioning datasets: Microsoft Video Description Dataset (MSVD) [8] and Montreal Video Annotation Dataset (M-VAD) [32]. For MSVD, it consists of 1970 videos which range from 10 s to 25 s, with the average length of about 9s. Each video has multi-lingual descriptions which are labelled by Amazon Mechanical Turkers (AMT). For each video, the descriptions depict a single activity scene with about 40 sentences. So there are about 80,000 clip-description pairs. Following the standard split [26,27,39,40], we divide this dataset into a training set of 1200 videos, a validation set of 100 videos, and a test set of 670 videos, respectively. For M-VAD, it is a large-scale movie description dataset, which is composed of 46,523 movie snippets from 92 popular DVD moives annotated with 54,997 sentences, around only 1 or 2 captions per video and with an average length of 6.2 s per snippet. We follow the setting in [32], taking 36,921 videos for training, 4651 videos for validation, and 4951 videos for testing.

4.2. Experimental settings

We uniformly sample 20 frames for each video and then extract convolutional layer features from ResNet-200 [17]. We convert all the sentences to lower cases, remove punctuation characters and tokenize the sentences. We retain all the words in the dataset and obtain a vocabulary of 12,593 words for MSVD, 16,000 words for M-VAD. In the training phase, we add $<$BOS$>$ tag at the beginning of each sentence and $<$EOS$>$ tag at its end. For the unseen words in the vocabulary, we set them to $<$UNK$>$ flag. In the testing phase, we input $<$BOS$>$ tag to begin caption generation process.

The size of hidden units and word embedding is set to 512 and 512 respectively. The model is trained using mini-batch 64, and Adam [22] algorithm is adopted to optimize our loss function with learning rate 1×10^{-4} on MSVD, 2×10^{-4} on M-VAD. To reduce the overfitting during training phase, we apply dropout [24] with rate of 0.5 on the output of decoder GRU. To further prevent gradient explosion, we clip the gradients to [-10,10] to prevent gradient explosion. Empirically, we set λ to 0.4. In testing phase, we adopt beam search strategy for caption generation and set the beam size to $k = 5$.

Three common metrics in image/video captioning tasks are adopted to evaluate the captioning results: BLEU [29], CIDEr [33], and METEOR [12]. All the metrics are computed by using the codes released by the Microsoft COCO evaluation server [9].

4.3. Performance on MSVD

In Table 1, we compare the performance of different model variants. '1Layer Loss' denotes the decoder with one layer of GRU structure. '2Layer Loss' denotes our proposed decoder: two layers of GRU structure with one loss function in each layer. It demonstrates that a suitable representation of video features is helpful to generate the corresponding caption and the Real-time Encoder contributes to video description for a specific word at each time-stamp. From Table 1, the performance of 'SeqInSeq' outperforms 'RTE+1Layer Loss'. The performance of 'Seq Encoder+1Layer Loss' outperforms 'Seq Encoder+2Layer Loss'. It demonstrates that the decoder of '2Layer Loss' with 'Seq Encoder' could not guide the model to focus on effective visual content for the current word compared with '2Layer Loss' of RTE. This result indicates that the optimization process deviates from the correct direction without a suitable representation for the video. Moveover, it validates that our proposed SeqInSeq contributes to a suitable presentation for a specific word.

In Table 2, we compare the performance of SeqInSeq with other state-of-the-art methods. The first block lists the performance of the methods with video features extracted from a single deep network, and the second block lists the performance of the methods with video features extracted from multiple networks. Our method outperforms the top performance of these representative methods by 2.3 on BLEU@4, 0.4 on METEOR and 7.0 on CIDEr.

In Table 3, we compare the runtime of our model and some representation methods. As most of state-of-the-art methods don't release the source codes of their methods, so the accurate performance may not be available. Thus, we list the results of the methods which released the source codes. We estimate the runtime based on GTX 1080, and other experimental settings are the same as the best running in their papers. From Table 3, the runtime of SeqInSeq is 196 s, which is more than the runtime of compared methods. As we encode the video at each time-stamp and other methods encode the video just before the decoder, much time is

Table 4
Experiment results on the M-VAD dataset compared to the state-of-the-art methods. All values are reported as percentage (%). V, C, R, and G denote VGG, C3D, Resnet and GoogLeNet, respectively.

Method	METEOR
SA [39] (GC)	5.7
S2VT [34] (V)	6.7
LSTM-E [27] (VC)	6.7
HRNE [26]	5.8
HRNE (with attention) [26]	6.8
DMRM [38] (G)	6.9
Boundary-aware [4] (RC)	**7.3**
LSTM-TSA$_I$ [28] (VC)	6.4
LSTM-TSA$_V$ [28] (VC)	6.9
LSTM-TSA$_{IV}$ [28] (VC)	7.2
SeqInSeq	**7.3**

spent in the process to encode the video. Thus, the result in the Table 3 is reasonable.

Fig. 4 shows a few representative examples of human-annotated ground truth sentences and sentences generated by the models: ‘Seq Encoder+1Layer Loss’, ‘Seq Encoder+2Layer Loss’, ‘RTE+1Layer Loss’ and SeqInSeq. From the results, our method contributes to focus on the correct concepts in the video. For instance, compared with ‘a man’ and ‘talking’ in the sentence generated by ‘SeqEncoder+1Layer Loss’ for the first video, ‘two men’ and ‘fighting’ in SeqInSeq are more relevant to the video content. The examples also show that our proposed model helps to focus on more precise word.

Fig. 5 demonstates some examples that p_n of each frame has different values in the process of generating the word at different time-stamp. Each bar for the frame is the mean value of its p_n. For instance, in the first example of Fig. 5, the bar of f_1 is highest when generating the word ‘two’. However, for the word ‘man’, the bars of f_1, f_2, and f_4 almost have the same height, which shows roughly the same contribution to the generation of ‘man’. And for the words ‘are’ and ‘fighting’, the highest bars vary. The results validate that the model of SeqInSeq extracts different visual information of each frame for different word. Combined with the corresponding captions in the Fig. 4, our model contributes to focus on more precise word describing the video.

4.4. Performance on M-VAD

We also validate the performance of SeqInSeq on M-VAD dataset. As a movie description dataset, M-VAD contains more visual concepts and complex sentence structures. So it is a more challenging dataset compared with MSVD. The authers of CIDEr [33] showed that METEOR is always better than BLEU and outperforms CIDEr when the number of captions are small (CIDEr is comparable to METEOR when the number of captions are large). Since M-VAD has only a single caption, we decide to use METEOR on the evaluation of M-VAD dataset, which is also suggested by the work [26]. Table 4 lists the performance of our model and state-of-the-art methods on M-VAD dataset, and the results show that the performance of SeqInSeq on METEOR outperforms most of state-of-the-art methods. The performance of ‘Boundary-aware’ is also 7.3 on METEOR, while it detects boundaries of discontinuous segments, which is suitable for M-VAD collected from realistic movies. Moreover our method outperforms it on MSVD dataset.

5. Conclusions

In this paper, we propose a novel model of SeqInSeq for the precise description of the video, which aggregates the sequential frames of the video into a spatio-temporal representation at each time-stamp to utter a word in the decoder and adopts the decoder of two-layer GRU to focus on more correct concepts in visual content. We evaluate our model on popular MSVD dataset. And the performance exceeds the representative state-of-the-art methods. The visualization also validates the effectiveness of our model. Moreover, our model also achieves the best performance on another standard video captioning dataset.

Acknowledgments

This work is supported by the NSFC (under Grant U1509206,61472276) and Tianjin Natural Science Foundation (no. 15JCYBJC15400).

References

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, J. Sivic, Netvlad: Cnn architecture for weakly supervised place recognition, in: CVPR, 2016, pp. 5297–5307.
[2] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv:1409.0473, (2014).
[3] N. Ballas, L. Yao, C. Pal, A. Courville, Delving deeper into convolutional networks for learning video representations, ICLR, 2016.
[4] L. Baraldi, C. Grana, R. Cucchiara, Hierarchical boundary-aware neural encoder for video captioning, CVPR, 2017.
[5] X. Chang, Z. Ma, M. Lin, Y. Yang, A.G. Hauptmann, Feature interaction augmented sparse learning for fast kinect motion detection, IEEE Trans. Image Process. 26 (8) (2017) 3911–3920.
[6] X. Chang, Z. Ma, Y. Yang, Z. Zeng, A.G. Hauptmann, Bi-level semantic representation analysis for multimedia event detection, IEEE Trans. Cybern. 47 (5) (2017) 1180–1197.
[7] X. Chang, Y. Yang, Semisupervised feature analysis by mining correlations among multiple tasks, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2017) 2294–2305.
[8] D.L. Chen, W.B. Dolan, Collecting highly parallel data for paraphrase evaluation, in: ACL, Association for Computational Linguistics, 2011, pp. 190–200.
[9] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, C.L. Zitnick, Microsoft coco captions: data collection and evaluation server, arXiv:1504.00325 (2015).
[10] G. Cheng, C. Yang, X. Yao, L. Guo, J. Han, When deep learning meets metric learning: remote sensing image scene classification via learning discriminative cnns, IEEE Trans. Geosci. Remote Sens. 56 (5) (2018) 2811–2821, doi:10.1109/TGRS.2017.2783902.
[11] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv:1406.1078 (2014).
[12] M. Denkowski, A. Lavie, Meteor universal: Language specific translation evaluation for any target language, in: Proceedings of the Ninth Workshop on Statistical Machine Translation, 2014, pp. 376–380.
[13] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, L. Deng, Semantic compositional networks for visual captioning, CoRR abs/1611.08002 (2016).
[14] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, K. Saenko, Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition, in: ICCV, 2013, pp. 2712–2719, doi:10.1109/ICCV.2013.337.
[15] J. Han, G. Cheng, Z. Li, D. Zhang, A unified metric learning-based framework for co-saliency detection, IEEE Trans. Circuits Syst. Video Technol. (2017).
[16] J. Han, D. Zhang, G. Cheng, N. Liu, D. Xu, Advanced deep-learning techniques for salient and category-specific object detection: a survey, IEEE Signal Process. Mag. 35 (1) (2018) 84–100.
[17] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: ECCV, Springer, 2016, pp. 630–645.
[18] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: CVPR, IEEE, 2010, pp. 3304–3311.
[19] S. Ji, W. Xu, M. Yang, K. Yu, 3D convolutional neural networks for human action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 221–231.
[20] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: CVPR, 2015, pp. 3128–3137.
[21] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: CVPR, 2014, pp. 1725–1732.
[22] D. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv:1412.6980 (2014).
[23] N. Krishnamoorthy, G. Malkarnenkar, R. Mooney, K. Saenko, S. Guadarrama, Generating natural-language video descriptions using text-mined knowledge, in: Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, in: AAAI'13, 2013, pp. 541–547.
[24] Z. Ma, X. Chang, Z. Xu, N. Sebe, A.G. Hauptmann, Joint attributes and event analysis for multimedia event detection, IEEE Trans. Neural Netw. Learn. Syst. (2017).

Please cite this article as: H. Wang et al., Sequence in sequence for video captioning, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.07.024

[25] Z. Ma, X. Chang, Y. Yang, N. Sebe, A.G. Hauptmann, The many shades of negativity, IEEE Trans. Multimedia 19 (7) (2017) 1558–1568, doi:10.1109/TMM.2017.2659221.

[26] P. Pan, Z. Xu, Y. Yang, F. Wu, Y. Zhuang, Hierarchical recurrent neural encoder for video representation with application to captioning, in: CVPR, 2016, pp. 1029–1038.

[27] Y. Pan, T. Mei, T. Yao, H. Li, Y. Rui, Jointly modeling embedding and translation to bridge video and language, in: CVPR, 2016, pp. 4594–4602.

[28] Y. Pan, T. Yao, H. Li, T. Mei, Video captioning with transferred semantic attributes, CVPR, 2017.

[29] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2002, pp. 311–318.

[30] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.

[31] J. Thomason, S. Venugopalan, S. Guadarrama, K. Saenko, R.J. Mooney, Integrating language and vision to generate natural language descriptions of videos in the wild., in: Coling, vol. 2, 2014, p. 9.

[32] A. Torabi, C.J. Pal, H. Larochelle, A.C. Courville, Using descriptive video services to create a large data source for video annotation research, CoRR abs/1503.01070 (2015).

[33] R. Vedantam, C. Lawrence Zitnick, D. Parikh, Cider: Consensus-based image description evaluation, in: CVPR, 2015, pp. 4566–4575.

[34] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, K. Saenko, Sequence to sequence-video to text, in: CVPR, 2015, pp. 4534–4542.

[35] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, K. Saenko, Translating videos to natural language using deep recurrent neural networks, arXiv:1412.4729 (2014).

[36] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, in: CVPR, 2015, pp. 3156–3164.

[37] Y. Xu, Y. Han, R. Hong, Q. Tian, Sequential video vlad: Training the aggregation locally and temporally, IEEE TIP, 2018, doi:10.1109/TIP.2018.2846664.

[38] Z. Yang, Y. Han, Z. Wang, Catching the temporal regions-of-interest for video captioning, ACM MM, 2017.

[39] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, A. Courville, Describing videos by exploiting temporal structure, in: ICCV, 2015, pp. 4507–4515.

[40] H. Yu, J. Wang, Z. Huang, Y. Yang, W. Xu, Video paragraph captioning using hierarchical recurrent neural networks, in: CVPR, 2016, pp. 4584–4593.

[41] L. Zhu, Z. Xu, Y. Yang, Bidirectional multirate reconstruction for temporal modeling in videos, CVPR, 2017.

Please cite this article as: H. Wang et al., Sequence in sequence for video captioning, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.07.024