Fully Convolutional Video Captioning with Coarse-to-Fine and Inherited Attention

Kuncheng Fang,[1] Lian Zhou,[1] Cheng Jin,[1] Yuejie Zhang,[1]
Kangnian Weng,[2] Tao Zhang,[2] Weiguo Fan[3]
[1]School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing,
Shanghai Institute of Intelligent Electronics & Systems, Fudan University, Shanghai, China
{17210240084, 16110240019, jc, yjzhang}@fudan.edu.cn
[2]School of Information Management and Engineering, Shanghai Key Laboratory of Financial Information Technology,
Shanghai University of Finance and Economics, Shanghai, China
weng.kangnian@163.sufe.edu.cn, taozhang@mail.shufe.edu.cn
[3]Department of Management Sciences, Tippie College of Business, University of Iowa, Iowa City, Iowa, USA, 52242
weiguo-fan@uiowa.edu

Abstract

Automatically generating natural language description for video is an extremely complicated and challenging task. To tackle the obstacles of traditional LSTM-based model for video captioning, we propose a novel architecture to generate the optimal descriptions for videos, which focuses on constructing a new network structure that can generate sentences superior to the basic model with LSTM, and establishing special attention mechanisms that can provide more useful visual information for caption generation. This scheme discards the traditional LSTM, and exploits the fully convolutional network with coarse-to-fine and inherited attention designed according to the characteristics of fully convolutional structure. Our model cannot only outperform the basic LSTM-based model, but also achieve the comparable performance with those of state-of-the-art methods.

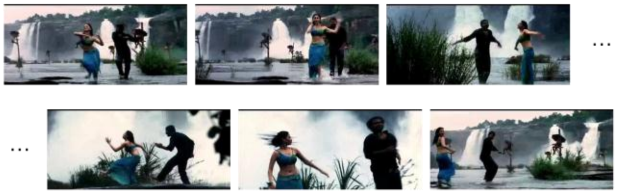
Introduction

With the explosive growth of video data on the Web, how to seamlessly handle the complex structures of various videos to achieve effective description generation becomes a research focus (Venugopalan et al. 2015b; Yao et al. 2015). Although it's easy for human to describe a video with a quick glance, it needs more complicated designs for computers to do the same task (Venugopalan et al. 2015a; Pan et al. 2016). The caption generation model must identify what objects appear in a video, the attributes of these objects, and the relationships among objects. All of these tasks are the fundamental challenges in computer vision. Thus video captioning, which aims at constructing a language generation model that can express the semantic understanding with accurate and meaningful descriptions for videos, has received considerable attentions.

Some pioneering methods try to address video captioning through hard-coded visual concepts and sentence templates (Kojima et al. 2002). However, these methods are highly human-crafted and the generated sentences are less natural. Recently, inspired by the advances of neural machine translation model (Cho et al. 2014; Sutskever et al. 2014), the combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) is widely adopted, which has significantly improved the quality of video textual description. Generally, the CNN-RNN models first encode the video into a fixed-length feature vector using CNN, and then feed the feature vector into a RNN decoder to generate captions.

Despite the existing progress, the fundamental CNN-RNN model is still a "*rough*" one. For video captioning, its input and output are sequential structures which contain temporal information. Due to the outstanding performance of RNN especially Long Short Term Memory network (LSTM), RNN naturally becomes an essential part in the sequence generation tasks (Venugopalan et al. 2015a; 2015b). However, its shortcomings in captioning tasks have been discovered. LSTM requires the previous output as the input at each moment, which makes the training extremely slow. The memory unit of LSTM is complex and requires the significant storage because of the fairly long path of back-propagation. All of these make the training of LSTM difficult. Thus some researchers have tried to solve the sequence generation via new model architecture without LSTM, and made some breakthroughs in neural machine translation (Gehring et al. 2017; Vaswani et al. 2017).

Input Video:



Output Captions:

- **LSTM:** a man is running.
- **Ours :** a man and woman are dancing.
- **Ground Truth:** a man and a woman are dancing near a waterfall. / a man and a woman is dancing in the water. / a couple are dancing near a waterfall.

Figure 1: The example sentences generated by LSTM, our method, and the ground truth captions.

Motivated by the above observations, we propose a novel framework to generate the optimal descriptions for videos. Our framework discards the traditional LSTM and exploits the fully convolutional network as the basic architecture, and meanwhile leverages the coarse-to-fine and inherited attention which are designed based on the characteristics of the fully convolutional structure. This framework aims at addressing two key issues to tackle the obstacles of LSTM, i.e., constructing a new network structure that can generate sentences superior to the LSTM-based models, and establishing special attention mechanisms that can provide more useful visual information for caption generation. The example captions generated by the LSTM-based model and ours are given in Figure 1. To the best of our knowledge, this is a new attempt to apply the pure fully convolutional network with attention for video captioning.

Our main contributions involve the following aspects: 1) Different from the traditional LSTM-based model, a pure fully convolutional architecture with coarse-to-fine and inherited attention is introduced for video captioning; and 2) The coarse-to-fine attention is designed to make full use of different levels of visual information involved in videos, and the inherited attention is developed to better concentrate on the region-level information that needs to be focused on at different moments. Our model can not only outperform the basic LSTM-based model, but also achieve the comparable performance with those of the state-of-the-art models on MSVD, which verifies the effectiveness and feasibility of our proposed framework.

Related Work

Describing the content of videos with natural language has made great progress in recent years. The existing methods for video captioning can be divided into two categories, i.e., template-based and sequence learning-based (Kojima et al. 2002; Venugopalan et al. 2015a; Gan et al. 2017).

The template-based method predefines some special rules for the generated description and subdivides the caption into several parts such as subject, verb and object (Kojima et al. 2002). With the predefined templates, each part of the sentence is associated with the words detected from the visual information of video, and finally the description of video can be generated. For example, to describe human activities with natural language, Kojima et al. (2002) proposed a concept architecture of actions, and a semantic hierarchy was designed to learn the semantic relationships among different sentence fragments. Recently, Xu et al. (2015) proposed a unified framework which consisted of a compositional semantic language model, a deep video model, and a joint embedding model for video captioning. These methods could generate fluent sentences, but had obvious problems. They highly relied on the predefined templates and rules, which made the generated sentences very rigid.

The sequence learning-based method is different from the template-based. Instead of utilizing the predefined rules, it directly generates the final caption with a much more flexible syntactical structure based on the input video. Donahue et al. (2015) proposed Long-term Recurrent Convolutional Networks (LRCNs), which leveraged the strengths of CNNs for visual recognition and utilized LSTM as the language model. To get better visual representation, Venugopalan et al. (2015a) considered the temporal information with optical flow and used LSTM in both encoder and decoder. Pan et al. (2017) proposed a structure to consider the semantic attributes from both image and video, which could provide additional semantic information. To get more beneficial information, Xu et al. (2017) took multimodal features into consideration, which contained the features of frame, motion and audio. Li et al. (2017) jointly applied the region-level and frame-level attention to the task of video captioning to catch more useful and precise visual representation. Wang et al. (2018a) proposed a hierarchical reinforcement learning framework to learn the semantic dynamics when captioning a video. To get a better caption than that generated by the basic encoder-decoder structure, some researchers added an additional module called "*Reconstructor*" (Wang et al. 2018b) or "*ARNet*" (Chen et al. 2018). Although these modules have different names, the core idea is to solve some inherent problems of the basic encoder-decoder model by stacking the LSTM structure. Such works have made great contributions to video captioning and give us much inspiration. The advantage is that the generated captions are all well-formed sentences, which are much more natural than the sentences generated by the template-based ones. Though great progress has been made in the sequence learning method, many problems still exist in the LSTM-based models, which limits the further improvement for video captioning.

To mitigate the deficiencies in template-based and sequence learning-based methods, both *Facebook* (Gehring et al. 2017) and *Google* (Vaswani et al. 2017) proposed novel model architectures to solve sequence generation tasks without RNN. The model proposed by *Facebook* was based on CNN, while that proposed by *Google* was based on attention. Both models have achieved significant results on neural machine translation, which demonstrates the potentials of these models in other sequence modeling tasks like image or video captioning. Aneja et al. (2018) proposed to use CNN for image captioning, and the scores of sentences generated by the CNN-based model were comparable with the basic RNN-based model. Inspired by such progress, we particularly propose a video captioning model with a fully convolutional network, and establish new attention mechanisms for stacked structure and region-level attention calculation to generate more accurate descriptions.

Methodology

Revisit of RNN-based Model

Encoder A 2-D/3-D CNN is used as the video encoder. Each frame sampled from a video is encoded as a D_v-dimensional feature, i.e., $V = \{v_{1,}, v_2, \dots, v_n, \dots, v_N\}$, where N denotes the number of frames and v_n is the *n*-th frame of video. The common way to obtain the video representation is to take the average of the frame feature vectors in V. However, the mean pooling strategy discards the temporal information among the frames. More effectively, LSTM can be adopted as the video encoder. The current hidden state h_t can be updated as:

$$h_t = LSTM(h_{t-1}, v_t) \tag{1}$$

where h_{t-1} is the previous hidden state, and v_t is the feature of the input frame at the current time step. The last hidden state of the LSTM encoder can be taken as the global feature representation, and then fed into the decoder.

Decoder Another LSTM is used as the decoder, which is initialized by the video representation. Each word in a caption is mapped to a D_w-dimensional word embedding. The whole sentence can then be represented as a sequence $S = \{s_{1,}, s_2, \dots, s_n, \dots, s_N\}$. Finally, the output caption can be generated based on the following Eq. (2).

$$h_t = LSTM(h_{t-1}, s_t) \tag{2}$$

where s_t is the input word embedding at the current time step. When the hidden state at each time step is obtained, the corresponding word of the caption can be generated.

Loss The negative log probability of sentence is given by summing the log probabilities over words in the sentence, which is defined as:

$$L_{NL} = -\sum_{t=0}^{T} \log \Pr(s_t | v, s_0, s_1, \dots, s_t) \tag{3}$$

where T is the length of sentence; v is the video representation; and s_t is the generated word at the current time step.

Our Fully Convolutional Model with Attention

Although the LSTM-based model can generate natural language descriptions for videos, the problems of LSTM still exist. To solve the problems, we propose a novel framework which integrates Fully Convolutional Network (a novel generation model for video captioning without the assistance of LSTM), Coarse-to-Fine Attention (a new attention mechanism used for a stacked structure) and Inherited Attention (a novel calculation strategy for frame weights at the region-level). An overview of our fully convolutional model with attention is shown in Figure 2.

Fully Convolutional Network

As described previously, the features of a video can be represented as $V = \{v_{1,}, v_2, \dots, v_t, \dots, v_N\}$, and the word embeddings of a corresponding caption can be represented as $S = \{s_{1,}, s_2, \dots, s_t, \dots, s_N\}$. The features of all the frames will be fed into the initial input part of the fully convolutional network and the attention module.

To make the input of the initial moment contain both semantic and visual information, the initial input can be obtained based on the following Eq. (4).

$$I = Concat(s_t, G) \tag{4}$$

where s_t denotes the word embedding of the word in the sentence at the current time step; and G denotes the global feature that is obtained by taking the average of the features of the sampled frames. Therefore, I is the concatenation of word embeddings and global features of a video.

As shown in Figure 2, the main structure of our model is the stacked N layers of masked 1-*D* CNN. The kernel size of each convolutional kernel is K. The latter part of the kernel is masked with zero, because the word embeddings corresponding to next time steps are not available at the current time step. A CNN kernel can receive *k* input features, that is, the current input feature is in the middle of the kernel, the left part receives the word embeddings of past time steps, and the right part is masked with zero. Due to the characteristics of a stacked CNN, each kernel of the high-level layer can process more information as the number of the stacked layer increases. Thus the output features of the last layer for the stacked structure can be considered to contain all information of the word embeddings provided to the first layer. The final caption of the video can then be generated by the fully convolutional network.

Coarse-to-Fine and Inherited Attention

Because the structure of our model is greatly different from the RNN-based model, the new attention mechanisms are designed based on the characteristics of our structure. In this section, we will introduce two attention mechanisms. The first one is the coarse-to-fine attention, which is used to provide different information for different layers. The second one is the inherited attention, which is used to achieve a more accurate visual representation.

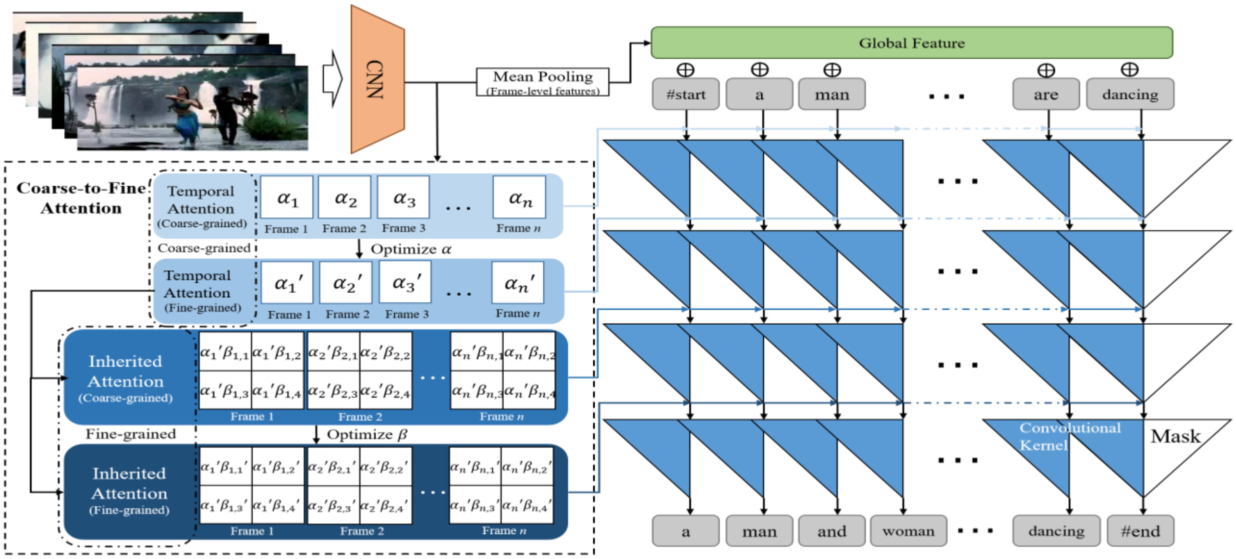


Figure 2: An illustration of our proposed fully convolutional model with coarse-to-fine and inherited attention. The video representation is constructed by CNN. GF means the global feature of video after mean pooling. The initial input is the concatenation of word embeddings and global features of a video. The coarse-to-fine attention consists of temporal attention and inherited attention. The different colors in coarse-to-fine attention mean the different fineness degrees of information. The darker the color, the more accurate the found information is. α_i means the weight of the i-th frame, and $\beta_{i,j}$ means the weight of the j-th region in the i-th frame. As the number of a stacked layer increases, the obtained information becomes more and more accurate.

Coarse-to-Fine Attention Because of the stacked structure of a fully convolutional network, where the input of each layer comes from the output of the previous layer, we can regard the entire process as a continuous optimization of the sentence generated by the previous layer. To make full use of the knowledge learned in each layer, a coarse-to-fine attention that is adaptive to the hierarchical architecture of our model is developed to help the model focus on the salient frames and regions, so as to make optimization for optimal caption generation.

Our coarse-to-fine attention aims to provide the most necessary information for different layers. It consists of temporal attention and inherited attention. Temporal attention can provide the frame-level visual information, and inherited attention provides the region-level visual information. As the number of a stacked layer increases, the coarse-to-fine attention can find much more precise visual information from a video. Each layer of our stacked model can utilize more precise information than before, in order to optimize the outputs generated by the previous layer with coarse-to-fine attention.

Temporal attention aims at calculating the weights of all frames sampled from a video based on the importance degrees of different frames at different time steps and providing the final result to the corresponding layer. It is used in the previous N layers of the model. Here, we adopt Multi-head Attention (Vaswani et al., 2017) to get the required weights. The result of temporal attention can be computed based on the following Eq. (5):

$$TA_t(i_t, F, F) = Concat(head_1, head_2, \dots, head_h)W^o$$
$$head_i(i_t, F, F) = Tatt_t\left(i_tW_i^Q, FW_i^K\right) * FW_i^V \qquad (5)$$
$$Tatt_t(i_t', F') = softmax\left(\frac{i_t'F'^T}{\sqrt{d_{F'}}}\right)$$

where i_t is the input at the time step t; $F = \{f_1, f_2, \dots, f_n\}$ is the features of frames; i_t' and $F' = \{f_1', f_2', \dots, f_n'\}$ are the representations of input and frames in the same feature space respectively; and n is the number of sample frames. TA_t is the final representation of visual information after using attention, and $Tatt_t$ denotes the weights of all frames of a video at the current time step. Multi-head attention allows the model to jointly attend to the information from different representation subspaces at different positions. The concatenation of all results of heads can be regarded as the collection of different representations of a video. As shown in Eq. (5), the representation is the concatenation of features generated by many different heads. Our model can then learn all the features, and get the final representation from the concatenation for a video. Multi-head attention can obtain multiple representations that focus on different features, because they use full connections that are independent of each other in every head. As the number of a stacked layer increases, the weights of different frames can be continuously optimized. Thus the final weights of frames are better than those with a single-layer structure.

Temporal attention can provide the frame-level visual information in the previous N layers, but such a frame-level attention may ignore region details. To put emphasis on more accurate local visual regions and get meaningful visual information for upper layers, it is important to explore inherited attention to focus on the region-level visual information in the next few layers.

Inherited Attention Some researchers have considered both frame-level and region-level attention in video captioning (Li et al. 2017). However, they do not fully utilize the relationship between the frame-level and region-level information, and cannot exploit the knowledge learned in the previous moment because of their model constraints.

To solve the above problems, we propose the inherited attention to calculate the region-level weights of visual information. The visual representations at the frame-level and region-level are quite different. The region-level representation is the more precise description of frame, and contains more detailed information than the frame-level representation. This means that the region-level representation at different times should first conform to the representation weight at the frame-level. Inherited attention is used in the next N layers after the layers with temporal attention, so that we can use the learned knowledge about frames to calculate the weights of regions. The representations and weights of regions at different times can be calculated as:

$$IA_t = \sum_{n=0}^{N} Tatt_{t,n} * \ RA_t(i_t, R_n, R_n)$$

$$RA_t(i_t, R, R) = Concat(head_1, head_2, \dots, head_h)W^o \quad (6)$$

$$head_i(i_t, R, R) = Ratt_t\left(i_t W_i^Q, RW_i^K\right) * RW_i^V$$

$$Ratt_t(i_t', R') = softmax\left(\frac{i_{t'} {R'}^T}{\sqrt{d_{R'}}}\right)$$

where i_t is the input at the time step t; $R = \{r_1, r_2, \dots, r_x\}$ denotes the features of regions; N is the number of sampled frames for video; and $Tatt_{t,n}$ is the weight of the n-th frame at the time step t. The final representation IA_t considers both the weights learned at the frame-level and those calculated at the region-level. Thus the final attention areas are more accurate, which can provide more beneficial information for the upper layers and then generate sentences that describe the video contents more precisely.

Caption Generation

The commonly used loss function for video captioning has been introduced in the part of RNN-based model. However, our fully convolutional model takes much more information into consideration. Thus we need a novel loss function that cannot only guide the model to generate reasonable captions, but also make the model pay attention to the most valuable areas. Thus our model is trained by minimizing the following loss function:

$$L = \ L_{NL} + \alpha \sum_i^F (1 - \sum_t^C w_{ti})^2 + \beta \sum_k^R (1 - \sum_t^C v_{tk})^2 \quad (7)$$

where the first part is the negative log-likelihood mentioned in Eq. (3); w_{ti} and v_{tk} are the weights of the i-th frame and k-th region at the time step t respectively, which can encourage the model not to focus on the same frame or the same region of the video at different time steps; α and β are two pre-set hyper-parameters to ensure that the negative log-likelihood loss contributes to the majority part of the final loss while the other parts are functioning.

Experiments

Dataset and Evaluation Metrics

The Microsoft Video Description Corpus (MSVD) is the most popular benchmark dataset for video captioning (Guadarrama et al. 2013). It contains 1,970 video clips downloaded from *YouTube* with roughly 40 English descriptions for each video. Typically, each video clip is about a single activity in open domains. For fair comparison, we follow the commonly utilized setting in our experiments, i.e., 1,200 videos for training, 100 videos for validation and 670 videos for testing.

For performance evaluation, we consider all the public evaluation metrics, including BLEU (Papineni et al. 2002), CIDEr (Vedantam et al. 2015), METEOR (Denkowski and Lavie, 2014) and ROUGE (Lin 2004).

Experimental Settings

We sample 15 frames for each video. *Inception*-V3 (Szegedy et al. 2016) and *C3D* (Ji et al. 2013) are used to extract features for video representation. The input images are resized to 299×299, and thus the dimension of frame features is 2,048. The region features of frames are extracted from a lower layer of *Inception*-V3. Here, the inception net is pre-trained on *ImageNet* (Deng et al. 2009), and *C3D* is on *Sports*-1M (Karpathy et al. 2014). The layer of *Inception*-V3 before a full connection (8×8×2,048) is utilized to extract the region features. Each frame can be represented with 8×8 grid regions. The kernel size of 1-*D* CNN is 5, and the number of stacked layers is 4. We use temporal attention in the first two layers and inherited attention in the last two layers. The head of multi-head attention is 8. The dimension of word-embedding, global feature, frame feature, and region feature are all 512.

Experiment Results and Analyses

Comparison with the State-of-the-Art Approaches

As shown in Table 1, we compare our results with those of the state-of-the-art approaches for video captioning. All the selected methods are LSTM-based. It can be observed that our method achieves the better results superior to the state-of-the-art approaches on all metrics. This indicates that our CNN-based model and attention mechanisms can significantly improve the whole generation performance.

Approach	BLEU-1	BLEU-2	BLEU-3	BLEU-4	CIDEr	METEOR	ROUGE
LSTM-E (Pan et al. 2016)	78.8	66.0	55.4	45.3	-	31.0	-
h-RNN (Yu et al. 2016)	81.5	70.4	60.4	49.9	65.8	32.6	-
MAM-RNN (Li et al. 2017)	80.1	66.1	54.7	41.3	53.9	32.2	-
LSTM-LS (Liu et al. 2017)	80.2	69.0	60.1	51.1	-	32.6	-
MA-LSTM (Xu et al. 2017)	82.3	71.1	61.8	52.3	70.4	33.6	-
MFATT (Long et al. 2018)	83.0	71.9	63.0	52.0	72.1	33.5	-
TSL (Wu et al. 2018)	-	-	-	51.7	74.9	34.0	-
M3 (Wang et al. 2018c)	81.6	71.4	62.3	52.0	-	32.2	-
FCVC-*CF&IA* (Ours)	**83.5**	**72.8**	**63.3**	**53.1**	**79.8**	**34.8**	**71.8**

Table 1: The comparison results between state-of-the-art approaches and ours on MSVD.

Comparison with the Basic LSTM-based Model

Although Table 1 shows the effectiveness of our approach, it cannot completely exhibit the advantages of our fully convolutional model relative to a basic LSTM-based model, and whether the performance gains obtained from our attention mechanisms is remarkable or not. Thus we further carry out more experiments to validate these aspects. Meanwhile, the state-of-the-art approaches in Table 1 adopt different networks to get video features. To make a fair comparison, we particularly summarize the results of the basic LSTM-based model (LSTM), our basic CNN-based model (FCVC), and our fully convolutional model with attention (FCVC-*CF&IA*), as shown in Table 2. All these approaches use the features of frames extracted from *Inception*-V3 and *C3D*. The results indicate that our CNN-based model obviously outperforms the basic LSTM-based model. It's worth noting that the coarse-to-fine and inherited attention can also greatly improve the captioning performance, which further verifies their important roles.

Model	METEOR	BLEU-4	CIDEr	ROUGE
LSTM	31.9	43.4	44.1	68.6
FCVC	32.1	48.7	64.8	69.1
FCVC-*CF&IA*	**34.8**	**53.1**	**79.8**	**71.8**

Table 2: The results of the basic LSTM-based model, our basic CNN-based model and our CNN-based model with attention.

Why CNN but not LSTM

As mentioned previously, the LSTM-based model has some drawbacks, thus we exploit CNN to overcome such obstacles. To further exhibit the power of our model, we have done some experiments for better comparison between the CNN-based and LSTM-based model.

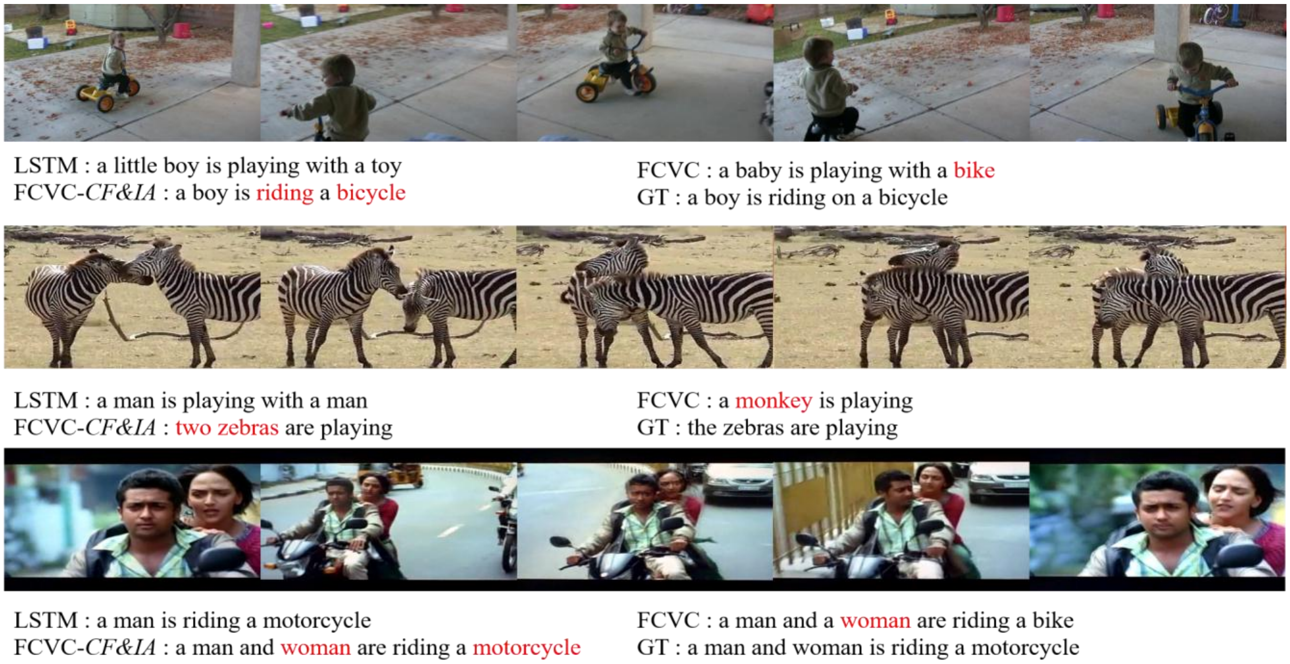
The results in Tables 1&2 clearly show the superiority of CNN. It can be observed that the results of a CNN-based model are much better than those of the LSTM-based models, and our model can generate more accurate sentences. According to the statistics on the reference sentences, it can be found that the number of "#*unk*" in the sentences generated by our model is much less than that of the basic LSTM-based model. The word "#*unk*" indicates that the model does not know which word to generate at the current time step, and thus makes the sentence unsmooth and lowers the evaluation scores. Because LSTM is hard to train, the training of the structure with a stacked LSTM is difficult and needs a large amount of time. Our CNN-based model, on the other hand, can optimize the sentence generation layer by layer with the stacked structure, and does not need more time than the LSTM-based models. The output of a CNN is all dependent on the input, which increases the power of the model to optimize the sentences, while a LSTM-based model has too many parameters to optimize.

LSTM needs the hidden state of a previous time step as the input of the current time step, which makes the model training time-consuming. CNN can be trained faster because it can run in parallel. Meanwhile, it is faster than a RNN-based model even with more extra actions due to the stacked model. Since the extraction process of frame representation is the same for all models, we use the frame features that have been extracted from *Inception*-V3 pre-trained on *ImageNet* and *C3D* pre-trained on *Sports*-1M, and then compare the cost of training time. The related results of these three methods are shown in Table 3.

Model	#*unk*	*Time*/10M parameters
LSTM	116	203sec
FCVC	11	**186sec**
FCVC-*CF&IA*	**0**	253sec

Table 3: The comparison of training time among the basic LSTM-based model, our basic CNN-based model and our CNN-based model with attention. "#unk" means the number of "#unk" in testing captions, and Time refers to the time cost in the training stage for every 10 million parameters.

As we expected, the training speed of a CNN-based model is faster than that of the basic LSTM-based model. Our fully convolutional model with attention needs more time for training because the model needs to consider both the frame-level and region-level features for video. This may introduce more operations than those of a basic LSTM-based model and our basic CNN-based model. All the operations used here is very concise and do not in-



LSTM : a little boy is playing with a toy
FCVC-*CF&IA* : a boy is riding a bicycle

FCVC : a baby is playing with a bike
GT : a boy is riding on a bicycle

LSTM : a man is playing with a man
FCVC-*CF&IA* : two zebras are playing

FCVC : a monkey is playing
GT : the zebras are playing

LSTM : a man is riding a motorcycle
FCVC-*CF&IA* : a man and woman are riding a motorcycle

FCVC : a man and a woman are riding a bike
GT : a man and woman is riding a motorcycle

Figure 3: Some captioning examples on MSVD with different models. The sentences generated by our fully convolutional model with attention can better describe the semantic contents of the video. GT means the ground truth sentences of MSVD.

crease the training difficulty. As shown in Figure 3, we give some captioning examples by the basic LSTM-based model, our basic CNN-based model, and our CNN-based model with coarse-to-fine and inherited attention.

How to use Attention

As described before, we put emphases on coarse-to-fine and inherited attention, and have obtained promising performance gains. To further verify the superiority of our attention mechanisms, we propose several different ways to exploit the attention mechanisms, and compare them with the basic one. The related results are shown in Table 4. FCVC is the baseline model, that is, the CNN-based model without attention. FCVC-*TA* is the CNN-based model with temporal attention in every layer of the model. FCVC-*IA* means we only use inherited attention. In every layer, we use temporal attention to compute the weights of different frames, and then inherited attention to get region-level weights. FCVC-*all* and FCVC-*final* use both coarse-to-fine attention and inherited attention. FCVC-*all* uses temporal attention in the first two layers, and both temporal attention and inherited attention in the last two layers. FCVC-*final* uses temporal attention in the first two layers and only inherited attention in the last two layers. It can be seen that with both coarse-to-fine and inherited attention, we can obtain the best result. The result of FCVC-*final* is better than that of FCVC-*all* because the latter can only focus on the region-level attention in the last two layers. To reduce the difficulty of model learning, we specially distribute the learning of frame-level and region-level attention weights in different layers. This manner can mitigate the conflict among different modules, and help achieve better results. It will be better if the modules of the model only need to care about a small part of actual calculation. The features used here are also extracted from *Inception*-V3 and *C*3*D*.

Model	METEOR	BLEU-4	CIDEr	ROUGE
FCVC	32.1	48.7	64.8	69.1
FCVC-*TA*	33.1	49.8	70.7	69.7
FCVC-*IA*	33.3	50.5	73.7	70.2
FCVC-*all*	34.2	51.9	77.2	71.1
FCVC-*final*	**34.8**	**53.1**	**79.8**	**71.8**

Table 4: The comparison between the baseline CNN-based model without attention and the CNN-based models with attention.

Conclusion

In this paper, we propose a novel fully convolutional network with coarse-to-fine and inherited attention to generate video captions. We build a CNN-based model to replace LSTM, which can achieve faster training and better descriptions of videos. We develop coarse-to-fine and inherited attention, and prove their feasibility with extensive experiments. The promising results on the MSVD dataset verify the effectiveness of our model. Since simply stacking CNN cannot fully realize the potentials of a CNN-based model in sequence generation tasks, it may not be adaptive enough for more complex dataset. We will consider exploring a more effective CNN-based model, and further enhance its generalization ability and robustness for video caption generation in our future work.

Acknowledgement

This work was supported by National Natural Science Foundation of China (No. 61572140 and No. 61672165), Shanghai Municipal R&D Foundation (No. 17DZ1100504, No. 16JC1420401, and No. 16511104704). Weiguo Fan is supported by the Henry Tippie Endowed Chair Fund from the University of Iowa. Yuejie Zhang from Fudan University is the corresponding author.

References

Aneja, J.; Deshpande, A.; and Schwing, A. G. 2018. Convolutional Image Captioning. In *Proceedings of CVPR 2018*, 5561-5570.

Chen, X.; Ma, L.; Jiang, W.; Yao, J.; and Liu, W. 2018. Regularizing RNNs for Caption Generation by Reconstructing The Past with The Present. In *Proceedings of CVPR 2018*, 7995-8003.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of EMNLP 2014*, 1724-1734.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Li, F. 2009. ImageNet: A Large-scale Hierarchical Image Database. In *Proceedings of CVPR 2009*, 248-255.

Denkowski, M.; and Lavie, A. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proceedings of ACL 2014 Workshop*, 376-380.

Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T. 2015. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. In *Proceedings of CVPR 2015*, 2625-2634.

Gan, Z.; Gan, C.; He, X.; Pu, Y.; Tran, K.; Gao, J.; Carin, L.; and Deng, L. 2017. Semantic Compositional Networks for Visual Captioning. In *Proceedings of CVPR 2017*, 955-964.

Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional Sequence to Sequence Learning. In *Proceedings of ICML 2017*, 1243-1252.

Ji, S.; Xu, W.; Yang, M.; and Yu, K. 2013. 3D Convolutional Neural Networks for Human Action Recognition. *TPAMI* 35(1): 221-231.

Li, X.; Zhao, B.; and Lu, X. 2017. MAM-RNN: Multi-level Attention Model based RNN for Video Captioning. In *Proceedings of IJCAI 2017*, 2208-2214.

Lin, C. 2004. Rouge: A Package for Automatic Evaluation of Summaries. In *Proceedings of ACL 2004 Workshop*, 74-81.

Liu, Y.; Li, X.; and Shi, Z. 2017. Video Captioning with Listwise Supervision. In *Proceedings of AAAI 2017*, 4197-4203.

Guadarrama, S.; Krishnamoorthy, N.; Malkarnenkar, G.; Venugopalan, S.; Mooney, R.; Darrell, T.; and Saenko, K. 2013. Youtube2text: Recognizing and Describing Arbitrary Activities using Semantic Hierarchies and Zero-shot Recognition. In *Proceedings of ICCV 2013*, 2712-2719.

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Li F. 2014. Large-scale Video Classification with Convolutional Neural Networks. In *Proceedings of CVPR 2014*, 1725-1732.

Kojima, A.; Tamura, T.; and Fukunaga, K. 2002. Natural Language Description of Human Activities from Video Images based on Concept Hierarchy of Actions. *IJCV* 50(2): 171-184.

Long, X.; Gan, C.; and Melo, G. 2018. Video Captioning with Multi-faceted Attention. *TACL* 6: 173-184.

Pan, Y.; Mei, T.; Yao, T.; Li, H.; and Rui, Y. 2016. Jointly Modeling Embedding and Translation to Bridge Video and Language. In *Proceedings of CVPR 2016*, 4594-4602.

Pan, Y.; Yo, T.; Li, H.; and Mei, T. 2017. Video Captioning with Transferred Semantic Attributes. In *Proceedings of CVPR 2017*, 984-992.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W. J. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL 2002*, 311-318.

Sutskever, I.; Vinyals, O.; and Le, Q. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of NIPS 2014*, 3104-3112.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of CVPR 2016*, 2818-2826.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention Is All You Need. In *Proceedings of NIPS 2017*, 5998-6008.

Vedantam, R.; Lawrence Zitnick, C.; and Parikh, D. 2015. Cider: Consensus-based Image Description Evaluation. In *Proceedings of CVPR 2015*, 4566-4575.

Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R.; Darrell, T.; and Saenko, K. 2015a. Sequence to Sequence-Video to Text. In *Proceedings of ICCV 2015*, 4534-4542.

Venugopalan, S.; Xu, H.; Donahue, J.; Rohrbach, M.; Mooney, R.; and Saenko, K. 2015b. Translating Videos to Natural Language Using Deep Recurrent Neural Networks. In *Proceedings of NAACL 2015*, 1494-1504.

Wang, X.; Chen, W.; Wu, J.; Wang, Y.; and Wang, W. 2018a. Video Captioning via Hierarchical Reinforcement Learning. In *Proceedings of CVPR 2018*, 4213-4222.

Wang, B.; Ma, L.; Zhang, W.; and Liu, W. 2018b. Reconstruction Network for Video Captioning. In *Proceedings of CVPR 2018*, 7622-7631.

Wang, J.; Wang, W.; Huang, Y.; Wang, L.; and Tan, T. 2018c. M3: Multimodal Memory Modelling for Video Captioning. In *Proceedings of CVPR 2018*, 7512-7520.

Wu, X.; Li, G.; Cao, Q.; Ji, Q.; and Lin, L. 2018. Interpretable Video Captioning via Trajectory Structured Localization. In *Proceedings of CVPR 2018*, 6829-6837.

Xu, J.; Yao, T.; Zhang, Y.; and Mei, T. 2017. Learning Multimodal Attention LSTM Networks for Video Captioning. In *Proceedings of ACM MM 2017*, 537-545.

Xu, R.; Xiong, C.; Chen, W.; and Corso, J. 2015. Jointly Modeling Deep Video and Compositional Text to Bridge Vision and Language in a Unified Framework. In *Proceedings of AAAI 2015*, 2346-2352.

Yao, L.; Torabi, A.; Cho, K.; Ballas, N.; Pal, C.; Larochelle, H.; and Courville, A. 2015. Describing Videos by Exploiting Temporal Structure. In *Proceedings of ICCV 2015*, 4507-4515.

Yu, H.; Wang, J.; Huang, Z.; Yang, Y.; and Xu, W. 2016. Video Paragraph Captioning using Hierarchical Recurrent Neural Networks. In *CVPR*, 4584-4593.