Bridge Video and Text with Cascade Syntactic Structure

Guolong Wang, Zheng Qin, Kaiping Xu, Kai Huang and **Shuxiong Ye**
Tsinghua University
wanggl16@mails.tsinghua.edu.cn, qingzh@tsinghua.edu.cn
{xkp13, huang-k15, ysx15}@mails.tsinghua.edu.cn

Abstract

We present a video captioning approach that encodes features by progressively completing syntactic structure (*LSTM-CSS*). To construct basic syntactic structure (i.e., subject, predicate, and object), we use a Conditional Random Field to label semantic representations (i.e., motions, objects). We argue that in order to improve the comprehensiveness of the description, the local features within object regions can be used to generate complementary syntactic elements (e.g., attribute, adverbial). Inspired by redundancy of human receptors, we utilize a Region Proposal Network to focus on the object regions. To model the final temporal dynamics, Recurrent Neural Network with Path Embeddings is adopted. We demonstrate the effectiveness of *LSTM-CSS* on generating natural sentences: 42.3% and 28.5% in terms of BLEU@4 and METEOR. Superior performance when compared to state-of-the-art methods are reported on a large video description dataset (i.e., MSR-VTT-2016).

1 Introduction

Video has become a ubiquitous way of communication on the Internet, podcast channels, as well as mobile devices. Accelerated by the explosive spread of video data, automatic analysis of semantic video content remains a promising area. The advances of this task can provide comparatively favorable preconditions for subsequent tasks, such as video retrieval, human-perception analysis, keyframe recommendation(Chen et al., 2017). To encapsulate the informative dynamics in the video, researchers have started focusing on recognizing videos based on the predefined templates, such as (Kojima et al., 2002; Guadarrama et al., 2014). Rohrbach et al. (2013) learned a CRF to assemble between different components of the input video and generated descriptions for videos. Xu et al. (2015) utilized a unified framework that jointly models video and the corresponding text sentences.

Another line of work uses Recurrent Neural Network. In light of its success, the representations (e.g. key objects, locations, motions, and scenes) can be concatenated into an input sequence and then translated to a natural sentence (Donahue et al., 2015). Follow-up works investigate the modeling of not only video contents and their spatio-temporal relationships, but also the syntactical structure (Venugopalan et al., 2015). More recently, a visual-semantic embedding learning technique has been proposed to model video content and textual semantics as a regularizer in Long Short-Term Memory architecture (Pan et al., 2016). This was extended by (Hori et al., 2017) where they utilized a modality-dependent attention mechanism.

Over the past years, researchers have studied multiple strategies to effectively bridge the visual content and textual description based on some fresh ideas. (Pan et al., 2017) incorporated the transferred semantic attributes learned from images and videos into the *CNN plus RNN* framework. (Baraldi et al., 2017) proposed a novel LSTM cell which can modify the temporal connections of the encoding layer according to the identified discontinuity points among frames. (Kaufman et al., 2017) transferred the semantics of the selected reference clips to test clips, which keeps consistent and maintains temporal coherence. However, the existing methods fail to take advantage of the local constraints which can extract compressed features for generating complementary syntactic elements.

Proceedings of the 27th International Conference on Computational Linguistics, pages 3576–3585
Santa Fe, New Mexico, USA, August 20-26, 2018.

Input video:

Output Sentence:

- LSTM-CSS [ours]: a fashion show is going on the runway.
- Human: a woman is modelling clothes. / a fashion show with women walking. / fashion show runway walk involving several models.

Figure 1: Examples of video description generation

This paper proposes a novel deep architecture, named Cascade Syntactic Structure (CSS), which takes advantages of incorporating global representations and local object regions into sequence learning for video captioning. Take the given video in Figure 1 as an example, the object regions of interest can be extracted to depict primary objects with locations (e.g., "woman") while motions from global features convey the temporal dynamics (e.g., "walking", "showing"). As "fashion show" and "runway" are not in the motion and object candidate set, we believe it is the local features that convey this complementary information. This has facilitated video captioning to disentangle the global features (motions and objects) and local features (object regions) enabling independent specification of both within the generation process. Our exploration of object regions as a modality for video captioning complements recent advances. Concretely, we propose a hierarchical *CNN plus RNN* architecture to learn a low-dimensional joint embedding for global and local features. The Convolutional Neural Network (*CNN*) has two discriminatively trained streams, motion stream and object stream. The outputs of these two streams are motions and objects respectively, which are used by Conditional Random Field (*CRF*) (Lafferty et al., 2001) to formulate the basic syntactic structure. The Long Short-Term Memory Network, together with Path embeddings (i.e., *LSTM-PE*) is used to generate a final sentence with optimal semantic structure.

As the final performance of our description generation relies on the accuracy of motion classification and object detection, we argue that the two stream *CNN* needs to be discriminative enough to provide information for following sub-structures. Its output global representation (i.e., a triplet of motion and objects) is used to construct basic syntactic structure which determines the comprehensibility of a sentence. As actions in MSR-VTT-2016 dataset (Xu et al., 2016) have a small range of movement and 3D convolutional neural networks (*C3D*) (Tran et al., 2015) performs not well enough on these actions, we prefer to use Temporal Segment Networks (*TSN*) (Wang et al., 2016) to extract motion features with additional optical flows (Z. et al., 2017). A *CRF* model is then used to learn the optimal global representation which is sent to a Long Short-term Memory (*LSTM*) network. Inspired by object masks proposed by Mask R-CNN (He et al., 2017), we believe that local constraints (i.e. object regions) can contribute to generating more comprehensive information for the result. We apply region-based network to our object stream and the output local constraints will be directly sent to aforementioned *LSTM* for generating the final sentence. The path embeddings can model the relationship between word vectors and contribute to the prediction of next word. This will help our model to cope with some recursive structures (e.g., nested conjunctions). Its efficacy is analyzed in section 2. Given a video, our framework can generate a sentence invariant to recursive structures based on both global representations and local constraints extracted from the video.

Our main technical contributions are three-fold:

- We propose a multi-task convnet to learn local embedding and global embedding for objects, which outperforms, by a large margin, previous attempts that use deep convnets for learning only global features.

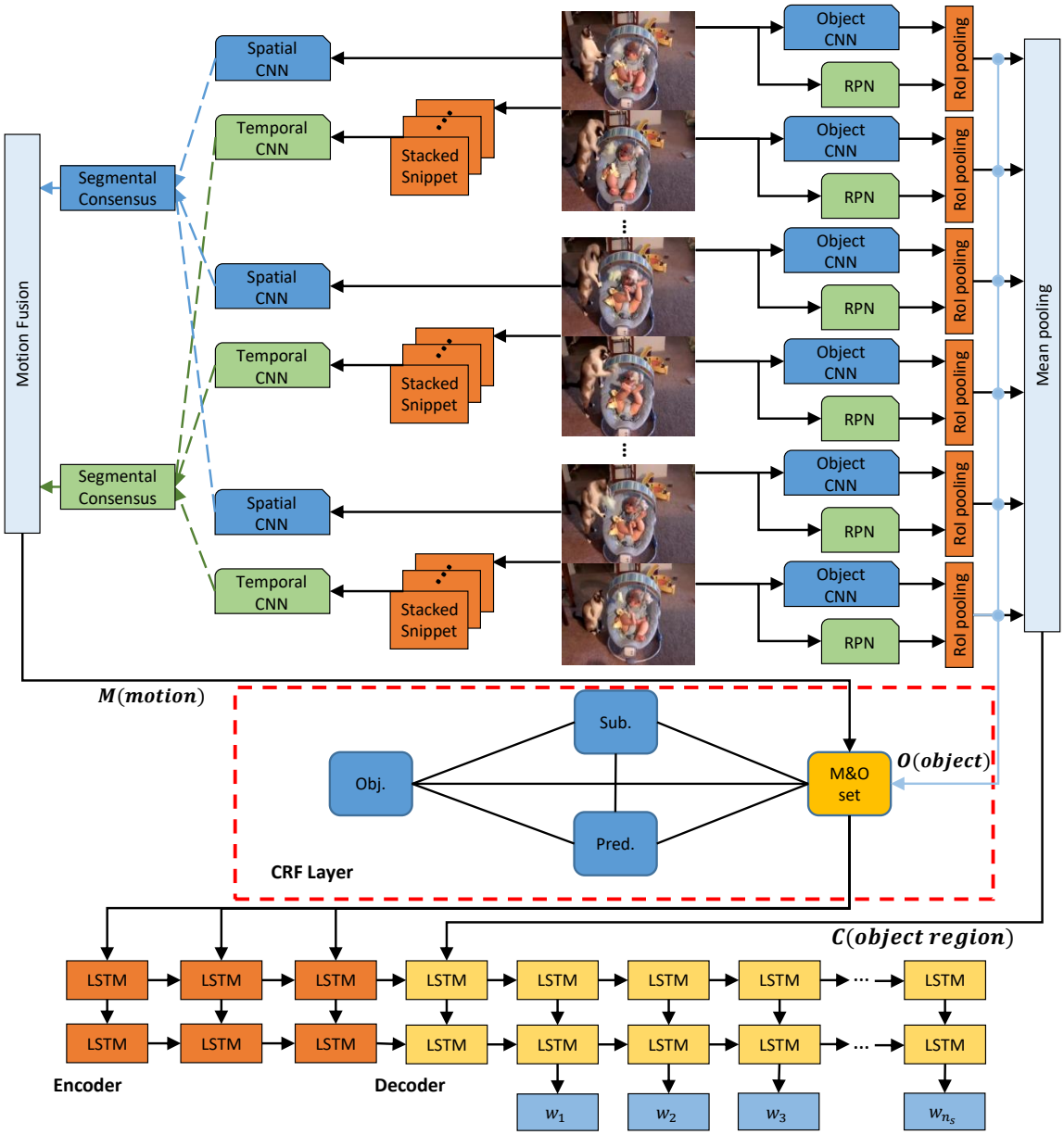

Figure 2: Hierarchical network combining global vectors (motions and objects) and local constraints (object regions) to generate video description with CRF and LSTM. Each sub-structure is trained independently (Section 2). M&O set denotes the semantic representation of motions and objects. Sub, Pred, and Obj denotes subject, predicate, and Object respectively in CRF.

- We empirically show the advantage of our unified video description generation framework whereby local constraints can be used to generate complementary syntactic elements, which can in turn improve the comprehensiveness of the description.
- We show that giving too much information to *LSTM* decoder could lead to chaos in the final structure. Hence, our method also input path embeddings to the decoder as context.

2 Video Captioning with Cascade Syntactic Structure

The goal of our network is to generate a sentence that describes the contents and their relationships. The problem is formulated as follows: given a video $\mathbf{V}$ with n_v segmented clips, our objective is to describe it by a textual sentence $\mathbf{S}$ with n_s words noted as $\mathbf{S} = \{\mathbf{w}_1, \cdots, \mathbf{w}_{n_s}\}$ with each word in it as

its column vector. $\mathbf{M} = \{\mathbf{m}_1, \cdots, \mathbf{m}_{n_m}\}$ and $\mathbf{O} = \{\mathbf{o}_1, \cdots, \mathbf{o}_{n_o}\}$ respectively denote the candidate of motions (resp. objects). $\mathbf{m}_i$ and $\mathbf{o}_i$ are represented by a one-hot vector. n_m and n_o refer to the size of the candidates. The local constraints $\mathbf{C}$ are D_c-dimensional local features, which are the mean pooling of selected frames in a video.

In the remaining of this section, we describe our network architecture and training methodology for generating a description for a specific video with local constraints. We first present two sub-structures of our architecture (Figure 2): 1) Feature extraction network (Section 2.1), which unify two discriminatively trained network streams that independently learn a motion (resp. object) feature embedding for a video; 2) Sentence generation network (Section 2.2), which uses CRF to learn optimal semantic representation and LSTM to generate the final description.

2.1 Feature extraction network

2.1.1 Motion stream

For each clip, the motion stream incorporates a Spatial CNN accepting a frame of RGB image and a Temporal CNN accepting a short snippet with 10 optical flow fields (5 for x directions and 5 for y directions) stacked in channel dimension extracted by warped TVL1 (Wang and Schmid, 2014). The frame sent to the Spatial CNN is randomly selected from a specific segment. The optical flow fields sent to the Temporal CNN are randomly selected from those computed for arbitrary two consecutive frames in each segment. Owing to the uncertain range of optical flow displacement, normalization is applied to constraint the displacements in $[0, 255]$ which is same as gray image. Thus, an optical flow field can be regarded as an image with 10 channels in the motion stream. The output of the motion stream, motion embedding $\mathbf{M}$, is averaged over all segments.

2.1.2 Object stream with local constraints

In conventional methods (Venugopalan et al., 2015), only embedding of motions and objects are extracted for LSTM. We believe that if visual features are sent to the LSTM, it is more generative to learn sentences for videos. In the human retina, visual features which represent natural signals formed by the peripheral receptors are very high-dimensional. However, we argue that the receptors to discover where and what objects only occupy a small fraction of the space of all possible receptor activation due to the statistical regularity and redundancy (Burton and Moorhead, 1987). We choose **ROI pooling** (Girshick, 2015) to extract local features specifically for the object regions as our local constraints.

As the object stream is the main sub-structure in our model, a careful training protocol is required to ensure convergence. For each clip, two images are randomly selected. For each input image, we train a network with aforementioned local constraints. The Object CNN can use many networks with the output discrete probability distribution $p = (p_0, \cdots, p_k, \cdots, p_{K-1})$, indexed by k. It is computed by a softmax layer over K object classes. The RPN network is a stack of convolutional filters of different sizes. Its outputs are the offsets of object regions, $r^k = (r^k_x, r^k_y, r^k_w, r^k_h)$, for the kth object class. The network can be described with a function $f(\cdot)$ minimising:

$$L = m + \sum_i L_{cls} + \lambda \sum_i L_{loc} \tag{1}$$

where m is a margin promoting convergence and i is the index of object region. The confidence loss L_{cls} is log loss for true class for each region. The location loss L_{loc} is the Smooth L1 loss (Girshick, 2015) between the predicted region and the ground truth region. Local constraint $\mathbf{C}$ (features in regions) and object embedding $\mathbf{O}$ are the output of object stream. Use of local Constraints is later shown to yield significant performance gain (Section 3.4.1).

2.2 Sentence generation

2.2.1 CRF: basic syntactic structure

The sentence generation network comprises two parts: CRF and LSTM. For CRF model, given $\mathbf{z} = \mathbf{M} \cup \mathbf{O}$ as input, let $\mathbf{y} = \{y_1, \cdots, y_{n_g}\}, n_g = n_m + n_s$ represents the label sequence of them, use the

following standard energy formulation:

$$E(\mathbf{y}, \mathbf{z}) = \sum_{i=1}^{n_g} E^u(y_i; \mathbf{z}) + \sum_{i,j} E^p(y_i, y_j) \tag{2}$$

where the first term defines the sum of quadratic unary terms and the second term describes the relationship between pairs of (y_i, y_j) (Hu et al., 2016).

Corresponding to our predicted $\mathbf{y}^*$, we rearrange $\mathbf{z}$ for optimal semantic representation $\mathbf{z}^*$. It is sent to the following LSTM model as well as local constraints $\mathbf{C}$. The LSTM model is based on an encoder-decoder framework (Sutskever et al., 2014). The encoder computes a intermediate representation $\mathbf{e}$ for the input semantic representation $\mathbf{z}^*$. Based on encoded intermediate representation, the decoder generates a translation, one target word at a time. The $\mathbf{S}$ is the output of the decoder. The log conditional probability is as follows:

$$\log\ p(\mathbf{S}|\mathbf{z}^*) = \sum_{t=1}^{n_s} \log\ p(\mathbf{w}_t|\mathbf{w}_{<t}, \mathbf{e}) \tag{3}$$

The objective is formulated as:

$$L = \sum_{(\mathbf{z}^*, \mathbf{S}) \in \mathbb{D}} -\log\ p(\mathbf{S}|\mathbf{z}^*) \tag{4}$$

where $\mathbb{D}$ refers to the parallel training corpus of source and target representation pairs $(\mathbf{z}^*, \mathbf{S})$.

2.2.2 LSTM-PE: final syntactic structure

The encoder and decoder share a common LSTM network with similar forward and backpropagation process. The difference is the decoder starts from the intermediate representation rather than zero states. In addition to the intermediate representation, the local constraints $\mathbf{C}$ are injected at the initial time of the decoder to inform the whole memory cells in LSTM. For timestep t, $\mathbf{x}_t$, $\mathbf{y}_t$ and $\mathbf{h}_t$ are the input vector, output vector, and hidden state respectively.

$$\begin{aligned}
&\mathbf{g}_t = \phi(\mathbf{U}_g\mathbf{x}_t + \mathbf{W}_g\mathbf{h}_{t-1} + \mathbf{b}_g), \mathbf{i}_t = \sigma(\mathbf{U}_i\mathbf{x}_t + \mathbf{W}_i\mathbf{h}_{t-1} + \mathbf{b}_i),\\
&\mathbf{f}_t = \sigma(\mathbf{U}_f\mathbf{x}_t + \mathbf{W}_f\mathbf{h}_{t-1} + \mathbf{b}_f), \mathbf{c}_t = \mathbf{g}_t \odot \mathbf{i}_t + \mathbf{c}_{t-1} \odot \mathbf{f}_t,\\
&\mathbf{o}_t = \sigma(\mathbf{U}_o\mathbf{x}_t + \mathbf{W}_o\mathbf{h}_{t-1} + \mathbf{b}_o), \mathbf{h}_t = \phi(\mathbf{c}_t) \odot \mathbf{o}_t,
\end{aligned} \tag{5}$$

where $\mathbf{g}_t$, $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{c}_t$, $\mathbf{o}_t$, and $\mathbf{h}_t$ are cell input, input gate, forget gate, cell state, output gate, and cell output of the LSTM. σ is the element-wise sigmoid function and $\odot$ is the element-wise product. $\mathbf{U}$ are the weight matrices of different gates for input $\mathbf{x}_t$, $\mathbf{W}$ are the recurrent weight matrices for hidden state $\mathbf{h}_t$, and $\mathbf{b}$ are bias vectors.

To make full use of aforementioned optimal label sequence $\mathbf{y}^*$, we learn the path embeddings from sequences (Chen and Manning, 2014) and use them as context (Jiang Guo and Xu, 2016) for decoder. We pre-train path embeddings together with word embeddings. At the initial time, the path embeddings stand for generic dependencies between labels (i.e., subject, predicate, and object). The path embeddings are then enlarged and refined during the learning phase. Let $\mathbf{P}_t$ denote the path embeddings, which is the concatenation of vectors representing labels:

$$\begin{aligned}
&\mathbf{p}_{-1} = [\mathbf{y}^*],\\
&\mathbf{p}_t = [\mathbf{p}_{t-1}\mathbf{y}_{t-1}],
\end{aligned} \tag{6}$$

The LSTM updating procedure of decoder is as:

$$\begin{aligned}
&\mathbf{x}_{-1} = \mathbf{U}_C\mathbf{C},\\
&\mathbf{x}_t = \mathbf{U}_s\mathbf{e},\\
&\mathbf{h}_t = f(\mathbf{W}_i\mathbf{x}_t + \mathbf{W}_h\mathbf{h}_{t-1} + \mathbf{W}_p\mathbf{P}_t),\\
&\mathbf{y}_t = softmax(\mathbf{W}_o\mathbf{h}_t),
\end{aligned} \tag{7}$$

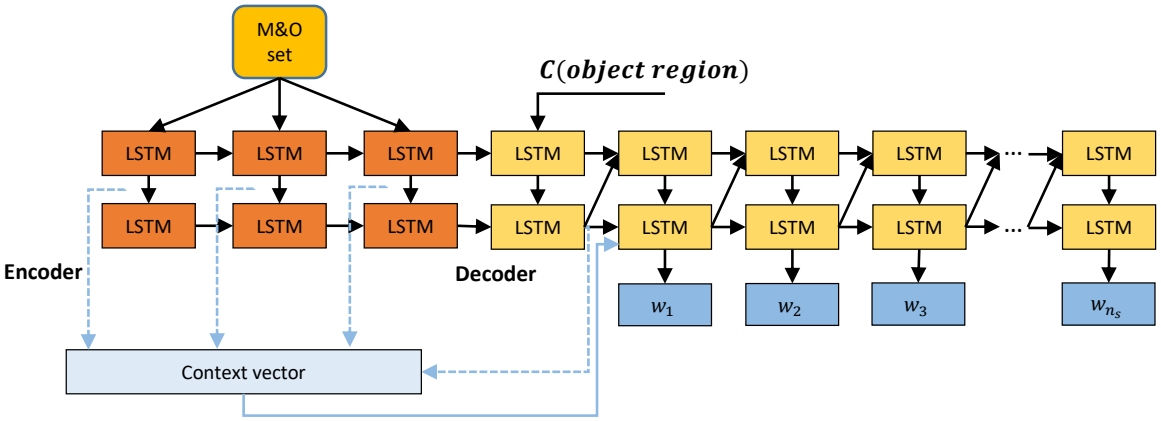

Figure 3: Detail architecture of LSTM-PE (The input of encoders is from M&O set, Local constraints $\mathbf{C}$ are input to initialize decoder as complementary information, $\mathbf{p}_t$ goes from the output layer to the input layer).

Figure 4: Examples of generated captions on MSR-VTT. GT1 and GT2 are ground truth captions.

where $\mathbf{U}_C$ and $\mathbf{U}_s$ are the transformation matrices for local constraints and intermediate representation respectively. $\mathbf{W}_i$, $\mathbf{W}_h$, $\mathbf{W}_p$, $\mathbf{W}_o$ are the weight matrices. f is the updating function within LSTM unit. We also apply an attention mechanism proposed by (Bahdanau et al., 2015) to LSTM units. The details can be illustrated in Figure 3. The output of LSTM is the final sentence $\mathbf{S}$ generated for video $\mathbf{V}$.

3 EXPERIMENTS AND DISCUSSION

We evaluate and compare our proposed LSTM-CSS with state-of-the-art approaches (Section 3.3) and analyze the effect of local constraints (Section 3.4.1). We also qualitatively analyze the effect of path embeddings (Section 3.4.2).

3.1 Datasets and settings

Dataset. We use MSR-VTT-2016 (Xu et al., 2016) which contains 10,000 web-collected video clips. There are roughly 20 available descriptions per video. In our experiment, our video captioning models are trained, hyperparameter selected, and evaluated using the official partitioned training, validation, and test set with 65%:5%:30% corresponding to 6,513, 497, 2,990 video clips, respectively.

Network detail. The motion stream is a reproduced model of TSN, fine-tuned over MSR-VTT-2016. For a single CNN, it closely resembles BN Inception network (Ioffe and Szegedy, 2015). We take the output of softmax layer from the combination of multiple CNNs as the motion representation $\mathbf{M}$. The object

Model	BLEU@4	METEOR	ROUGE-L	CIDEr-D
SA (Yao et al., 2015)	32.3	23.4	-	-
S2VT (Venugopalan et al., 2015)	35.2	25.2	-	-
Xu et al. (Xu et al., 2016)	36.6	25.9	-	-
MTLM (Pasunuru and Bansal, 2017)	40.8	**28.8**	60.2	47.1
WS (Shen et al., 2017)	41.4	28.3	61.1	**48.9**
Rank1: v2t_navigator	40.8	28.2	60.9	44.8
Rank2: Aalto	39.8	26.9	59.8	45.7
Rank3: VideoLAB	39.1	27.7	60.6	44.1
LSTM-CSS	**42.3**	28.5	**61.2**	46.5

Table 1: BLEU@4, METEOR, CIDEr-D, and ROUGE-L of our LSTM-CSS and other state-of-the-art methods on MSR-VTT-2016 dataset. All values are reported as percentage (%) and (-) indicates unknown scores

Model	BLEU@4	METEOR	ROUGE-L	CIDEr-D
LSTM-SR	38.1	25.6	59.0	40.7
LSTM-CSS-E-N	40.4	26.6	59.8	43.2
LSTM-CSS-N	41.1	26.9	60.2	44.8
LSTM-CSS-E	42.0	27.0	60.2	46.7
LSTM-CSS	42.3	28.5	61.2	46.5

Table 2: Captioning accuracy of LSTM-CSS, LSTM-SR, LSTM-CSS-E, LSTM-CSS-N, and LSTM-CSS-E-N on MSR-VTT-2016 test set

stream is trained from scratch, using an COCO (Lin et al., 2014) to evenly detect objects $\mathbf{O}$ belonging to 90 classes and extract local features. In each video, 5 frames are selected and local constraints $\mathbf{C}$ are the average of local features extracted from these frames. We take the output of softmax layer and output of 4096-way fc6 layer from the Resnet-50 (He et al., 2016) as $\mathbf{O}$ and $\mathbf{C}$ respectively. The margin m in Equation 1 is set to 1. The dimension of the input and hidden layers in LSTM are both set to 512. In test stage, we set the beam size to 4 in beam search.

Evaluation metric. We adopt four common metrics in video captioning task for quantitative evaluation of our proposed model: BLEU@4, METEOR, CIDEr-D, and ROUGE-L (from MS-COCO evaluation server (Chen et al., 2015)).

3.2 Compared methods

We compare our LSTM-CSS model with the following methods to evaluate the efficacy of our model.

SA (**S**oft-**A**ttention) (Yao et al., 2015): utilizes a weighted attention mechanism to dynamically attend to specific temporal segments of the video with the input of both frame representation (2-D CNN) and video clip representation (3-D CNN). In the test, the frame representation is extracted from the same feature, while the video clip representation is extracted from a C3D network.

S2VT (**S**equence to **S**equence - **V**ideo to **T**ext) (Venugopalan et al., 2015): incorporates both RGB and optical flow inputs. The encoding and decoding of inputs and word representations are jointly learned. In the test, the RGB and optical flows are extracted from same structure with ours.

MTLM (**M**ulti-**T**ask **L**earning **M**ethod) (Pasunuru and Bansal, 2017): improves video captioning by utilizing sharing representations with a temporally-directed unsupervised video prediction task to learn richer context-aware video encoder representations, and a logically-directed language entailment generation task to learn better caption decoder. In the test, the representation is extracted from same structure with ours.

WS (**W**eakly **S**upervised) (Shen et al., 2017): links video regions with lexical labels by utilizing lexical fully convolutional neural networks with weakly supervised learning. It also trains a sequence-to-sequence language model with the weakly supervised information. In the test, the CNN model is trained starting from the same structure with ours.

We also compare the baseline criterion and top-3 rank (i.e. v2t_navigator, Aalto, and VideoLAB) results proposed by (Xu et al., 2016).

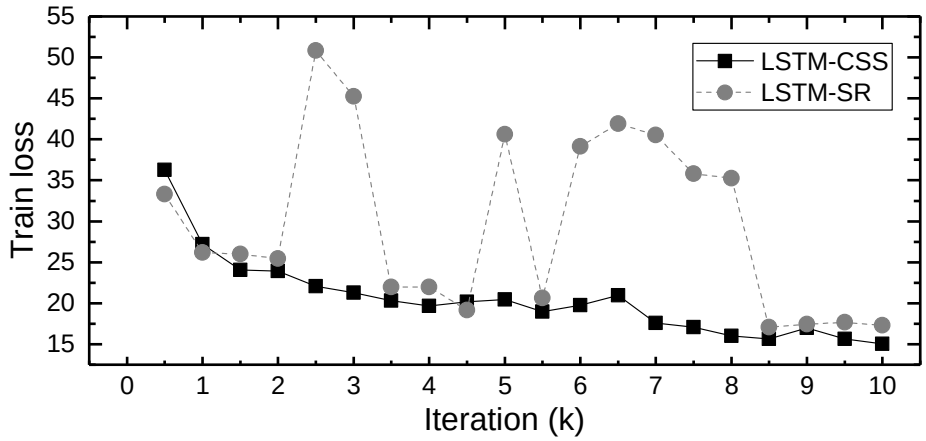

Figure 5: Train loss of LSTM-CSS and LSTM-SR

3.3 Performance comparison

Table 1 shows our primary results on MSR-VTT-2016 dataset. Overall, our proposed LSTM-CSS outperforms the other methods. Specifically, it reaches 42.3% BLEU@4, which makes the relative improvement over the two state-of-the-art methods S2VT by 20.1% and MTLM by 3.7%, respectively. We are also the new Rank1 on the MSR-VTT-2016 leaderboard, based on their ranking criteria. Figure 4 shows several example captions generated by our approach for MSR-VTT-2016 videos. The result indicates the utilizing of local constraints can contribute to the generation of comprehensive sentences.

In terms of BLEU scores, our method obtains bigger gains than others. Since the preference of longer captions tend to obtain lower BLEU scores, we argue that our method can generate comprehensive but concise captions. In terms of METEOR scores, our method does not perform best. From the aforementioned example, we can see that the subject of our caption is "fashion show" and the adverbial is "on the runway". In this example, there are no reference captions that have same syntactic structure with our caption. This may influence our recall rate and further affect the METEOR score.

3.4 Ablation study

3.4.1 Effect of Complementary syntactic elements

Our model **LSTM-CSS** depicted in Figure 2 uses local constraints to construct the final syntactic structure with complementary elements. To perform the ablation studies for our local constraints, we also test degraded versions of our model as follows: 1) **LSTM-SR**: the input of LSTM are semantic representation of motions and objects. Only global features are included; 2) **LSTM-CSS-E**: the local constraints are added as another input modal of LSTM and they are input to the encoder; 3) **LSTM-CSS-N**: the network without attention mechanism; 4) **LSTM-CSS-E-N**: the network remove attention mechanism from **LSTM-CSS-E**; 5) **LSTM-CSS w/o PE**: the network remove path embeddings from **LSTM-CSS**.

Figure 5 shows that LSTM-CSS achieves better training loss than LSTM-SR, which shows the training stability of LSTM-CSS. We can draw a conclusion that the local constraints can contribute to ensuring the convergence rate and alleviate the fluctuation of training loss. Table 2 summarizes the results on the MSR-VTT-2016 test set. As can be seen, in all the cases, LSTM-SR performs worse than the models with local features. Base on these results, the validation of local constraints is certificated. From the comparison of LSTM-CSS-E-N and LSTM-CSS-N, we can conclude that without attention mechanism, modeling global and local information at the same level will weaken the ability of local constraints. We can also find that LSTM-CSS and LSTM-CSS-E nearly perform equally. Therefore, we believe that attention mechanism can make the LSTM translation network focus on relevant content and alleviate the influence of the places where local constraints are input.

LSTM-CSS: a woman is talking on a show.
LSTM-CSS w/o PE: a woman is talking.

LSTM-CSS: a man is talking about a man who in black.
LSTM-CSS w/o PE: a man is talking about.

Figure 6: Examples with recursive syntactic structure

3.4.2 Qualitative analysis of path embeddings

As we can see, in the MSR-VTT-2016 dataset, some videos are about news, products, slides. In fact, people appearing in these videos are not the subject. Through visual analyzing, existing video caption algorithms can tell people are talking but not understand the content they talking about. Though some advanced methods can find both introducers and their contents, they are hard to model the relationship and organize sentences. In cases like 'someone is talking about something', our method can generate a completed sentence as shown in Figure 6. We can also see that in sentences generated by LSTM-CSS w/o PE, the content people shows are not included.

4 Conclusions

Inspired by the human receptors for object detection, we propose a hierarchical model for video captioning. Our model not only precisely captures motions and objects in videos but also learns a sentence constrained by cascade syntactic structure (*CRF* for the basic structure and *LSTM-PE* for the complementary elements). The experimental results demonstrate that our model performs on par with state-of-the-art methods on a large video description dataset in terms of accuracy.

However, the structure of generated sentence is relatively fixed for all videos. It should be better to improve the diversity of the sentences considering different object locations and make the syntactic structure more logical. We will address this issue in our future work.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. 2017. Hierarchical boundary-aware neural encoder for video captioning. In *CVPR*.

G. J. Burton and I. R. Moorhead. 1987. Color and spatial structure in natural scenes. *Applied Optics*, 26(1):157.

D. Chen and C. D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Conference on Empirical Methods in Natural Language Processing*.

Xinlei Chen, Hao Fang, Tsung Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *Computer Science*.

Xu Chen, Yongfeng Zhang, Qingyao Ai, Hongteng Xu, Junchi Yan, and Zheng Qin. 2017. Personalized key frame recommendation. In *SIGIR*.

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

Ross Girshick. 2015. Fast r-cnn. In *ICCV*.

S Guadarrama, N Krishnamoorthy, G Malkarnenkar, S Venugopalan, R Mooney, T Darrell, and K Saenko. 2014. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. 2017. Mask r-cnn. In *ICCV*.

Chiori Hori, Takaaki Hori, Teng Yok Lee, Kazuhiro Sumi, John R Hershey, and Tim K Marks. 2017. Attention-based multimodal fusion for video description. *arXiv preprint arXiv:1701.03126*.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. In *ACL*.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*.

Haifeng Wang Ting Liu Jiang Guo, Wanxiang Che and Jun Xu. 2016. A unified architecture for semantic role labeling and relation classification. In *Conference on Computational Linguistics*.

Dotan Kaufman, Gil Levi, Tal Hassner, and Lior Wolf. 2017. Temporal tessellation: A unified approach for video analysis. In *ICCV*.

Atsuhiro Kojima, Takeshi Tamura, and Kunio Fukunaga. 2002. Natural language description of human activities from video images based on concept hierarchy of actions. *IJCV*.

John D. Lafferty, Andrew Mccallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollr, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. 2016. Jointly modeling embedding and translation to bridge video and language. In *CVPR*.

Yingwei Pan, Ting Yao, Houqiang Li, and Tao Mei. 2017. Video captioning with transferred semantic attributes. In *CVPR*.

Ramakanth Pasunuru and Mohit Bansal. 2017. Multi-task video captioning with video and entailment generation. In *ACL*.

Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. 2013. Translating video content to natural language descriptions. In *ICCV*.

Zhiqiang Shen, Jianguo Li, Zhou Su, Minjun Li, Yurong Chen, Yu Gang Jiang, and Xiangyang Xue. 2017. Weakly supervised dense video captioning. In *CVPR*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

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

Heng Wang and Cordelia Schmid. 2014. Action recognition with improved trajectories. In *ICCV*.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*.

Ran Xu, Caiming Xiong, Jason J. Corso, and Jason J. Corso. 2015. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*.

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

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

Ren Z., Yan J., Ni B., Zha H., and Yang X. 2017. Unsupervised deep learning for optical flow estimation. In *AAAI*.