Dense Procedure Captioning in Narrated Instructional Videos

Botian Shi[1*†], **Lei Ji**[2,3†], **Yaobo Liang**[3], **Nan Duan**[3], **Peng Chen**[4], **Zhendong Niu**[1‡], **Ming Zhou**[3]
[1]Beijing Institute of Technology, Beijing, China
[2]Institute of Computing Technology, Chinese Academy of Science, Beijing, China
[3]Microsoft Research Asia, Beijing, China
[4]Microsoft Research and AI Group, Beijing, China
botianshi@bit.edu.cn,{leiji,yalia,nanduan,peche}@microsoft.com
zniu@bit.edu.cn,mingzhou@microsoft.com

Abstract

Understanding narrated instructional videos is important for both research and real-world web applications. Motivated by video dense captioning, we propose a model to generate procedure captions from narrated instructional videos which are a sequence of step-wise clips with description. Previous works on video dense captioning learn video segments and generate captions without considering transcripts. We argue that transcripts in narrated instructional videos can enhance video representation by providing fine-grained complimentary and semantic textual information. In this paper, we introduce a framework to (1) extract procedures by a cross-modality module, which fuses video content with the entire transcript; and (2) generate captions by encoding video frames as well as a snippet of transcripts within each extracted procedure. Experiments show that our model can achieve state-of-the-art performance in procedure extraction and captioning, and the ablation studies demonstrate that both the video frames and the transcripts are important for the task.

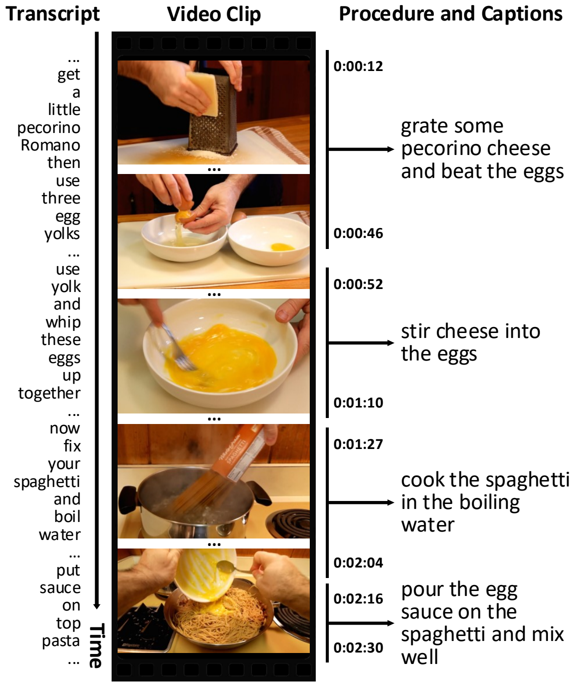

Transcript
Video Clip
Procedure and Captions
... get a little pecorino Romano then use three egg yolks ... use yolk and whip these eggs up together ... now fix your spaghetti and boil water ... put sauce on top pasta ...
Time
0:00:12
grate some pecorino cheese and beat the eggs
0:00:46
0:00:52
stir cheese into the eggs
0:01:10
0:01:27
cook the spaghetti in the boiling water
0:02:04
0:02:16
pour the egg sauce on the spaghetti and mix well
0:02:30

Figure 1: A showcase of video dense procedure captioning. In this task, the video frames and the transcript are given to (1) extract procedures in the video, (2) generate a descriptive and informative sentence as the caption of each procedure.

1 Introduction

Narrated instructional videos provide rich visual, acoustic and language information for people to easily understand how to complete a task by procedures. An increasing amount of people resort to narrated instructional videos to learn skills and solve problems. For example, people would like to watch videos to *repair a water damaged plasterboard / drywall ceiling*[1] or *cook Cottage Pie*[2]. This motivates us to investigate whether machines can understand narrated instructional videos like humans. Besides, watching a long video is time-consuming, captions of videos provide a quick overview of video content for people to learn the main steps rapidly. Inspired by this, our task is to generate procedure captions from narrated instructional videos which are a sequence of step-wise clips with a description as shown in Figure 1.

Previous works on video understanding tend to recognize actions in video clips by detecting pose (Wang et al., 2013a; Packer et al., 2012) and motion (Wang et al., 2013b; Yang et al., 2013) or both (Wang et al., 2014) and fine-grained features(Rohrbach et al., 2016). These works take low-level vision features into account and can

[*]This work was done during the first author's intership in MSR Asia

[†]Equal contribution

[‡]Corresponding Author

[1] https://goo.gl/QZFsfR

[2] https://goo.gl/2Z4Kb8

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6382–6391
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

only detect human actions, instead of complicated events that occur in the scene. To deeply understand the video content, Video Dense Captioning (Krishna et al., 2017) is proposed to generate semantic captions for a video. The goal of this task is to identify all events inside a video and our target is the video dense captioning on narrated instructional videos which we call *dense procedure captioning*.

Different from videos in the open domain, instructional videos contain an explicit sequential structure of procedures accompanied by a series of shots and descriptive transcripts. Moreover, they contain fine-grained information including actions, entities, and their interactions. According to our analysis, many fine-grained entities and actions also present in captions which are ignored by previous works like (Krishna et al., 2017; Zhou et al., 2018b). The procedure caption should be detailed and informative. Previous works (Krishna et al., 2017; Xu et al., 2016) for video captioning usually consist of two stages: (1) temporal event proposition; and (2) event captioning. However, there are two challenges for narrated instructional videos: one of the challenges is that video content fails to provide semantic information so as to extract procedures semantically; the other challenge is that it is hard to recognize fine-grained entities from the video content only, and thus tends to generate coarse captions.

Previous models for dense video captioning only use video signals without considering transcripts. We argue that transcripts in narrated instructional videos can enhance video representation by providing fine-grained complimentary and semantic textual information. As shown in Figure 1, the task takes a video with a transcript as input and extracts the main procedures as well as these captions. The whole video is divided into four proposal procedure spans in sequential order including: (1) *grate some pecorino cheese and beat the eggs* during time span [0:00:12-0:00:46], (2) *then stir cheese into the eggs* during [0:00:52-0:01:10], and so on. Besides video content, transcripts can provide semantic information. Our model embeds transcript using a pre-trained context-aware model to provide rich semantic information. Furthermore, with the transcript, our model can directly ”copy” many fine-grained entities, e.g. *pecorino cheese* for procedure captioning.

In this paper, we propose utilizing multi-modal content of videos including frame features and transcripts to conduct procedure extraction and captioning. First, we use the transcript of instructional videos as a global text feature and fuse it with video signals to construct context-aware features. Then we use temporal convolution to encode these features and generate procedure proposals. Next, the fused features of video and transcript tokens within the proposed time span are used to generate the final caption via a recurrent model. Experiments on the YouCookII dataset (Zhou et al., 2018a) (a cooking-domain instructional video corpus) are conducted to show that our model can achieve state-of-the-art results and the ablation studies demonstrate that the transcript can not only improve procedure proposition performance but also be very effective for procedure captioning.

The contributions of this paper are as follows:

1. We propose a model fusing transcript of narrated instructional video during procedure extraction and captioning.

2. We employ the pre-trained BERT(Devlin et al., 2018) and self-attention(Vaswani et al., 2017) layer to embed transcript, and then integrate them to visual encoding during procedure extraction.

3. We adopt the sequence-to-sequence model to generate captions by merging tokens of the transcript with the aligned video frames.

2 Related Works

Narrated Instructional Video Understanding Previous works aim to ground the description to the video. (Malmaud et al., 2015) adopted an HMM model to align the recipe steps to the narration. (Naim et al., 2015) utilize latent-variable based discriminative models (CRF, Structured Perceptron) for unsupervised alignment. Besides the alignment of transcripts with video, (Alayrac et al., 2016, 2018) propose to learn the main steps from a set of narrated instructional videos for five different tasks and formulate the problem into two clustering problems. Graph-based clustering is also adopted to learn the semantic storyline of instructional videos in (Sener et al., 2015). These works assume that ”one task” has the same procedures. Different from previous works, we focus on learning more complicated procedures for

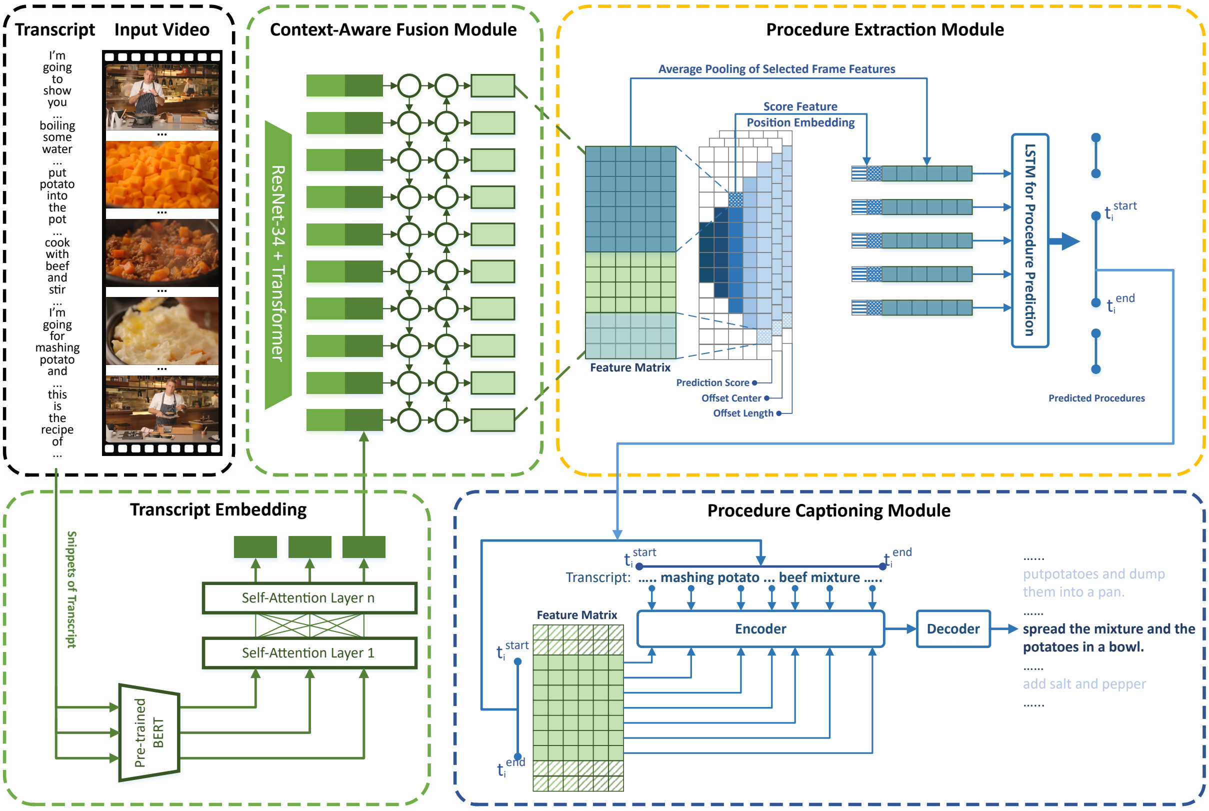

Transcript
Input Video
I'm going to show you ... boiling some water ... put potato into the pot ... cook with beef and stir ... I'm going for mashing potato and ... this is the recipe of ...
Context-Aware Fusion Module
ResNet-34 + Transformer
Procedure Extraction Module
Average Pooling of Selected Frame Features
Score Feature
Position Embedding
Feature Matrix
Prediction Score
Offset Center
Offset Length
LSTM for Procedure Prediction
t_i^{start}
t_i^{end}
Predicted Procedures
Transcript Embedding
Snippets of Transcript
Self-Attention Layer n
Self-Attention Layer 1
Pre-trained BERT
Procedure Captioning Module
t_i^{start}
t_i^{end}
Transcript: ..... mashing potato ... beef mixture .....
Feature Matrix
t_i^{start}
t_i^{end}
Encoder
Decoder
......
putpotatoes and dump them into a pan.
......
spread the mixture and the potatoes in a bowl.
......
add salt and pepper
......

Figure 2: The main structure of our model.

each video and propose a neural network model for step-wise summarization.

Temporal action proposal is designed to divide a long video into contiguous segments as a sequence of actions, which is similar to the first stage of our model. (Shou et al., 2016) adopt 3D convolutional neural networks to generate multi-scale proposals. DAPs in (Escorcia et al., 2016) apply a sliding window and a Long Short-Term Memory (LSTM) network for video content encoding and predicting proposals covered by the window. SST in (Buch et al., 2017) effectively generates proposals in a single pass. However, previous methods do not consider context information to produce non-overlapped procedures. (Zhou et al., 2018a) is the most similar work to ours, which is designed to detect long complicated event proposals rather than actions. We adopt this framework and inject the textual transcript of narrated instructional videos as our first step.

Dense video caption aims to generate descriptive sentences for all events in the video. Different from video captioning and paragraph generation, dense video caption requires segmenting of each video into a sequence of temporal proposals with corresponding captions. (Krishna et al., 2017) resorts to the DAP method (Escorcia et al., 2016) for event detection and apply the context-aware S2VT model (Venugopalan et al., 2015). (Yu et al., 2018) propose to generate long and detailed description for sport videos. (Li et al., 2018) train jointly on unifying the temporal proposal localization and sentence generation for dense video captioning. (Xiong et al., 2018) assembles temporally localized description to produce a descriptive paragraph. (Duan et al., 2018) propose weakly supervised dense event captioning, which does not require temporal segment annotations, and decomposes the problem into a pair of dual tasks. (Wang et al., 2018a) exploit both past and future context for predicting accurate event proposals. (Zhou et al., 2018b) adopt a transformer for action proposing and captioning simultaneously. Besides, there are also some works try to incorporate multi-modal information (e.g. audio stream) for dense video captioning task(Ramanishka et al., 2016; Xu et al., 2017; Wang et al., 2018b). The major difference is that our work adopts a different model structure and fuses transcripts to further enhance semantic representation. Experiments show that transcripts can improve both procedure extraction and captioning.

3 Model

In this section, we describe our framework and model details as shown in Figure 2. First, we adopt a context-aware video-transcript fusion module to generate features by fusing video information and transcript embedding; Then the procedure extraction module takes the embedded features and predicts procedures with various lengths; Finally, the procedure captioning module generates captions for each procedure by an encoder-decoder based model.

3.1 Context-Aware Fusion Module

We first encode transcripts and video frames separately and then extract cross-modal features by feeding both embeddings into a context-aware model.

To embed transcripts, we first split all tokens in the transcript by a sliding window and input them into a uncased BERT-large (Devlin et al., 2018) model. Next, we encode these sentences by a Transformer (Vaswani et al., 2017) and take the first output as the context-aware transcript embedding $\mathbf{e} \in \mathbb{R}^e$.

To embed the videos, we uniformly sample T frames and encode each frame $\mathbf{v}_t$ in $\mathbf{V} = \{\mathbf{v}_1, \cdots, \mathbf{v}_T\}$ to an embedding representation by an ImageNet-pre-trained ResNet-32 (He et al., 2016) network. Then we adopt another Transformer model to further encode the context information, and output $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_T\} \in \mathbb{R}^{T \times d}$.

Finally, we combine each of the frame features in $\mathbf{X}$ with transcript feature $\mathbf{e}$ to get the fused feature $\mathbf{C} = \{\mathbf{c}_1, \cdots, \mathbf{c}_t, \cdots, \mathbf{c}_T | \mathbf{c}_t = \{\mathbf{x}_t \circ \mathbf{e}\}\}$ and feed it into a Bi-directional LSTM (Hochreiter and Schmidhuber, 1997) in order to encode past and future contextual information of video frames: $\mathbf{F} = \text{Bi-LSTM}(\mathbf{C})$ where $\mathbf{F} = \{\mathbf{f}_1 \cdots \mathbf{f}_T\} \in \mathbb{R}^{T \times f}$, and f is the hidden size of the LSTM layers.

3.2 Procedure Extraction Module

We take the encoded T feature vectors $\mathbf{F}$ of each video as the elementary units to generate procedure proposals. We follow the idea in (Zhou et al., 2018a; Krishna et al., 2017) that (1) generate a lot of anchors, i.e. proposals, with different lengths and (2) use the frame features within a proposal span to predict plausible scores.

3.2.1 Procedure Proposal Generation

In order to generate different-sized procedure proposals, we adopt a 1D (temporal) convolutional layer with the setting of K different kernels; three output channels and zero padding to generate procedure candidates. The layer takes $\mathbf{F} \in \mathbb{R}^{T \times f}$ as input and outputs a list of $\mathbf{M}^{(k)} \in \mathbb{R}^{T \times 3}$ for each k-th kernel. All these results are stacked as a tensor $\mathbf{M} \in \mathbb{R}^{K \times T \times 3}$.

Next, the tensor $\mathbf{M}$ is divided into three matrices: $\mathbf{M} = \left[\hat{\mathbf{M}}_m, \hat{\mathbf{M}}_l, \hat{\mathbf{M}}_s\right]$ where $\hat{\mathbf{M}}_m$, $\hat{\mathbf{M}}_l$, $\hat{\mathbf{M}}_s \in \mathbb{R}^{K \times T}$, They are designed to represent the offset of the proposal's midpoint; the offset of the proposal's length and the prediction score. We calculate the starting and ending timestamp of each proposal by the offset of midpoint and length. Finally, a non-linear projection is applied on each matrix: $\mathbf{M}_m = \tanh(\hat{\mathbf{M}}_m)$, $\mathbf{M}_l = \tanh(\hat{\mathbf{M}}_l)$, $\mathbf{M}_s = \sigma(\hat{\mathbf{M}}_s)$ where σ is the Sigmoid projection.

3.2.2 Procedure Proposal Prediction

It is obvious that all proposed procedure candidates are co-related to each other. In order to encode this interaction, we follow the method in (Zhou et al., 2018a) which uses an LSTM model to predict a sequence from the $K \times T$ generated procedure proposal.

The input of the recurrent prediction model for each time step consists of three parts: frame features, the position embedding, the plausibility score feature.

Frame Features For a generated procedure proposal, the corresponding feature vectors $\mathbf{F}^{(k,t)}$ are calculated as follows:

$$\mathbf{F}^{(k,t)} = \left\{\mathbf{f}_{C(k,t)-L(k,t)}, \cdots, \mathbf{f}_{C(k,t)+L(k,t)}\right\} \quad (1)$$

$$C(k,t) = \lfloor t + \mathbf{k}^{(k)} \times \mathbf{M}_m^{(k,t)} \rfloor \quad (2)$$

$$L(k,t) = \lfloor \mathbf{k}^{(k)} \times \frac{\mathbf{M}_l^{(k,t)}}{2} \rfloor \quad (3)$$

where $\mathbf{k} = \{k_1, \cdots, k_K\}$ is a list of different kernel sizes. The $\mathbf{M}_m^{(k,t)}$ and $\mathbf{M}_l^{(k,t)}$ represent the midpoint and length offset of the span for k-th kernel and t-th frame respectively and $\mathbf{k}^{(k)}$ is the length of the k-th kernel.

Position Embedding We treat all possible positions as a list of tokens and use an embedding layer to get a continuous representation. The [BOS] and [EOS], i.e. the *begin of sentence* and the *end of sentence*, are also added into the vocabulary for sequence prediction.

Score Feature The score feature is a flatten of matrix $\mathbf{M}_s$, i.e. $\mathrm{s} \in \mathbb{R}^{K \cdot T \times 1}$.

The input embedding of each time step is the concatenation of:

1. The averaged features of the proposal predicted in the previous step t:

$$\overline{\mathbf{F}^{(k,t)}} = \frac{1}{2L(k,t)} \sum_{t'=-L(k,t)}^{L(k,t)} \mathbf{f}_{C(k,t)+t'} \tag{4}$$

2. The position embedding of the proposal.
3. The score feature s.

Specifically, for the first step, the input frame feature is the averaged frame features of the entire video. $\overline{\mathbf{F}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{f}_t$ and the position embedding is the encoding of [BOS]. The procedure extraction finishes when [EOS] is predicted, and the output of this module is a sequence of indexes of frames: $\mathbf{P} = \{p_1 \cdot p_L\}$ where L is the maximum count of the predicted proposals.

3.3 Procedure Captioning Module

We design an LSTM based sequence-to-sequence model (Sutskever et al., 2014) to generate captions for each extracted procedure.

For the (k, t)-th extracted procedure, we calculate the starting time t_s and ending time t_e separately and retrieve all tokens within the time span $[t_s, t_e]$: $E(t_s, t_e) = \{e_{t_s}, \cdots, e_{t_e}\} \subset \{e_1, \cdots, e_Q\}$ where Q is the total word count of a video's transcript.

On each step, we concatenate the embedding representation of each token $q \in E(t_s, t_e)$, i.e. $\mathbf{q}$, with the nearest video frame feature $\mathbf{f}_{\hat{q}}$ into the input vector $\mathbf{e}_q = \{\mathbf{q} \circ \mathbf{f}_{\hat{q}}\}$ of the encoder. We employ the hidden state of the last step after encoding all tokens in $E(t_s, t_e)$ and decode the caption of this extracted procedure as $\mathbf{W} = \{w_1, \cdots, w_Z\}$ where Z is the word count of the decoded procedure caption.

3.4 Loss Functions

The target of the model is to extract procedures and generate captions. The loss function consists of four parts: (1) $\mathcal{L}_s$: a binary cross-entropy loss of each generated positive and negative procedure; (2) $\mathcal{L}_r$: the regression loss with a smooth $l1$-loss (Ren et al., 2015) of a time span between the extracted and the ground-truth procedure. (3) $\mathcal{L}_p$: the cross-entropy loss of each proposed procedure in the predicted sequence of proposals. (4) $\mathcal{L}_c$: the cross-entropy loss of each token in the generated procedure captions. Here are the formulations:

$$\mathcal{L} = \alpha_s \mathcal{L}_s + \alpha_r \mathcal{L}_r + \alpha_p \mathcal{L}_p + \alpha_c \mathcal{L}_c \tag{5}$$

$$\begin{aligned} \mathcal{L}_s = &- \frac{1}{C_P} \sum_{i=1}^{C_P} \log(\mathbf{M}_s^P) \\ &- \frac{1}{C_N} \sum_{i=1}^{C_N} \log(1 - \mathbf{M}_s^N) \end{aligned} \tag{6}$$

$$\mathcal{L}_r = \frac{1}{C_P} \sum_{i=1}^{C_P} ||B_i^{pred} - B_i^{gt}||_{s-l1} \tag{7}$$

$$\mathcal{L}_p = - \frac{1}{L} \sum_{l=1}^{L} \log(p_l \mathbb{1}_l^{(gt_l)}) \tag{8}$$

$$\mathcal{L}_c = - \frac{1}{L} \sum_{l=1}^{L} \frac{1}{|\mathbf{W}_l|} \sum_{w \in \mathbf{W}_l} \log(w \mathbb{1}^{(gt_w)}) \tag{9}$$

where $\mathbf{M}_s^P$ and $\mathbf{M}_s^N$ are the scoring matrix of positive and negative samples in a video, and C_P and C_N represent the count separately. Here we regard a sample as positive if its IoU (Intersection of Union) with any ground-truth procedure is more than 0.8. If the IoU is less than 0.2, we treat it as negative. The loss $\mathcal{L}_s$ aims to enlarge the score of all positive samples and decrease the score otherwise.

The B_i^{pred} and B_i^{gt} represent the boundary (calculated by the offset of midpoint and length) of the positive sample and ground-truth procedure separately. We only take positive samples into account and conduct the regression with $\mathcal{L}_r$ to shorten the distance between all positive samples and the ground-truth procedures.

The p_l is the classification result of the procedure extraction module and the value of $\mathbb{1}$ will be 1 if the predicted class of extracted procedure proposal is identical to the class of the ground-truth proposal with the maximal IoU and 0 otherwise. The cross-entropy loss $\mathcal{L}_p$ aims to exploit the model to correctly select the most similar proposal of each ground-truth procedure from many positive samples.

Finally, $\mathbf{W}$ stores all decoded captions of procedures of a video. The $\mathcal{L}_c$ is designed for the captioning module based on the extracted procedures.

4 Experiment and Case Study

4.1 Evaluation Metrics

We separately evaluate the procedure extraction and captioning module.

For procedure extraction, we adopt the widely used mJacc (mean of Jaccard) (Bojanowski et al., 2014) and mIoU (mean of IoU) metrics for evaluating the procedure proposition. The Jaccard calculates the intersection of the predicted and ground-truth procedure proposals over the length of the latter. The IoU replaces the denominator part with the union of predicted and ground-truth procedures.

For procedure captioning, we adopt BLEU-4(Papineni et al., 2002) and METEOR(Banerjee and Lavie, 2005) as the metrics to evaluate the performance on the result of captioning based on both extracted and ground-truth procedures.

4.2 Dataset

In this paper, we use the YouCookII[3] (Zhou et al., 2018a) dataset to conduct experiments. It contains 2000 videos dumped from YouTube which are all instructional cooking recipe videos. For each video, human annotators were asked to first label the starting and ending time of procedure segments, and then write captions for each procedure.

This dataset contains pre-processed frame features ($T = 500$ frames for each video, each frame feature is a 512-d vector, extracted by ResNet-32) which were used in (Zhou et al., 2018a). In this paper, we also use these pre-computed video features for our task.

Besides the video content, our proposed model also relies on transcripts to provide multi-modality information. Since the YouCookII dataset does not have transcripts, we crawl all transcripts automatically generated by YouTube's ASR engine.

YouCookII provides a partition on these 2000 videos: 1333 for training, 457 for validation and 210 for testing. However, the labels of 210 testing videos are unpublished, we can only adopt the training and validation dataset for our experiment. We also remove several videos which are unavailable on YouTube. In all, we use 1387 videos from the YouCookII dataset. We split these videos into 967 for training, 210 for validation and 210 for testing. As shown in Table 1, even though we use less data for training, we can still obtain comparable results.

	validation		testing	
Methods	mJacc	mIoU	mJacc	mIoU
	YouCookII Partition			
SCNN-prop	46.3	28.0	45.6	26.7
vsLSTM	47.2	33.9	45.2	32.2
ProcNets	51.5	37.5	50.6	37.0
	Our Partition			
ProcNets	50.9	38.2	49.1	37.0
Ours (Video Only)	53.3	38.0	52.8	37.1
Ours (Full Model)	**56.5**	**41.4**	**56.4**	**41.8**

Table 1: Result on Procedure Extraction

4.3 Implementation Details

For the procedure extraction module, we follow the method in (Zhou et al., 2018a) to use 16 different kernel sizes for the temporal convolutional layer, i.e. from 3 to 123 with the interval step of 8, which can cover the different lengths. We also used a max-pooling layer with a kernel of [8, 5] after the convolutional layer.

We extract at most 16 procedures for each video, and the maximum caption length of each extracted procedure is 50. The hidden size of all recurrent model (LSTM) is 512 and we conduct a dropout for each layer with a probability of 0.5. We use two transformer models with 2048 inner hidden sizes, 8 heads, and 6 layers to encode context-aware transcripts and video frame features separately.

We adopt an Adam optimizer (Kingma and Ba, 2015) with a starting learning rate of 0.000025 and $\alpha = 0.8$ and $\beta = 0.999$ to train the model. The batch size of training is 4 for each GPU and we use 4 GPUs to train our model so the overall batch size is 16.

4.4 Result on Procedure Extraction

	Ground-Truth Procedures		Predicted Procedures	
Methods	B@4	M	B@4	M
Bi-LSTM +TempoAttn	0.87	8.15	0.008	4.62
End-to-End Transformer	1.42	11.20	0.30	6.58
Ours (Video Only)	2.20	17.59	1.70	16.71
Ours (Full Model)	**2.76**	**18.08**	**2.61**	**17.43**

Table 2: Result on Procedure Captioning

We demonstrate the result of the procedure extraction model by Table 1. We compare our model with several baseline methods: (1) SCNN-prop (Shou et al., 2016) is the Segment CNN for pro-

[3] http://youcook2.eecs.umich.edu/

	Procedure Extraction		Procedure Captioning			
			Ground-Truth Procedures		Predicted Procedures	
Methods	mJacc	mIoU	B@4	M	B@4	M
1. Video Only Model Proposal by Video Only & Caption by Video Only	52.80	37.13	2.20	17.59	1.70	16.72
2. Transcript Only Model Proposal by Transcript Only & Caption by Transcript Only	48.25	31.66	2.43	17.66	1.09	15.23
3. Caption by Video Model Proposal by Video+Transcript & Caption by Video Only	53.83	37.72	**3.12**	**18.24**	2.59	17.38
4. Caption by Transcript Model Proposal by Video+Transcript & Caption by Transcript Only	52.66	36.54	2.12	17.27	1.85	15.80
5. Full Model Proposal by Video+Transcript & Caption by Video+Transcript	**56.37**	**41.76**	2.76	18.08	**2.61**	**17.43**

Table 3: Ablation experiments of our model. (All experiments are conducted on testing dataset)

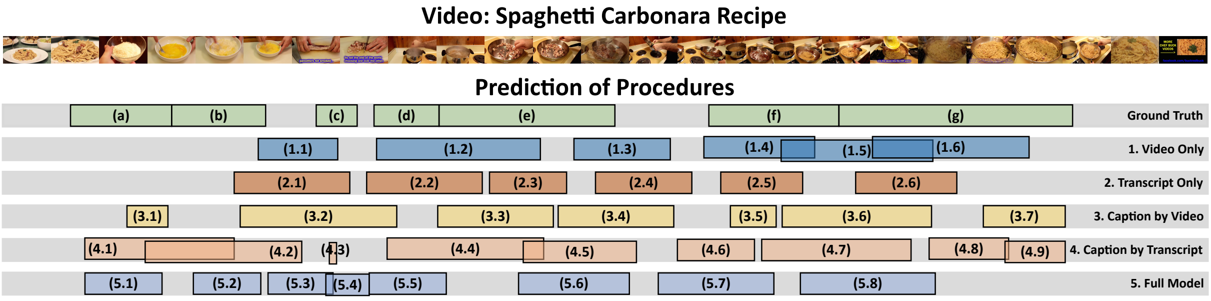


Figure 3: The ground-truth and extracted procedures, which are generated by our full and ablated models. (best viewed in color)

posals; (2) vsLSTM is an LSTM based video summarization model (Zhang et al., 2016); (3) ProcNets (Zhou et al., 2018a) which is the previous SOTA method.

As shown in Table 1, we first show the results reported in (Zhou et al., 2018a) which use the full dataset with 2000 videos. In order to ensure a fair comparison, we first run the ProcNets on the validation dataset of YouCookII and get a comparable result. In further experiments, we directly use the subset (the *our partition* in the table) described in the previous section.

Moreover, we conduct two experiments to demonstrate the effectiveness of incorporating transcripts in this task. The *Ours (Full Model)* is the final model we propose, which achieves state-of-the-art results. The *Ours (Video Only)* model considers video content without transcripts in the procedure extraction module. Compared with ProcNets, our *video only* model adds a captioning module, which helps the procedure extraction module to get a better result.

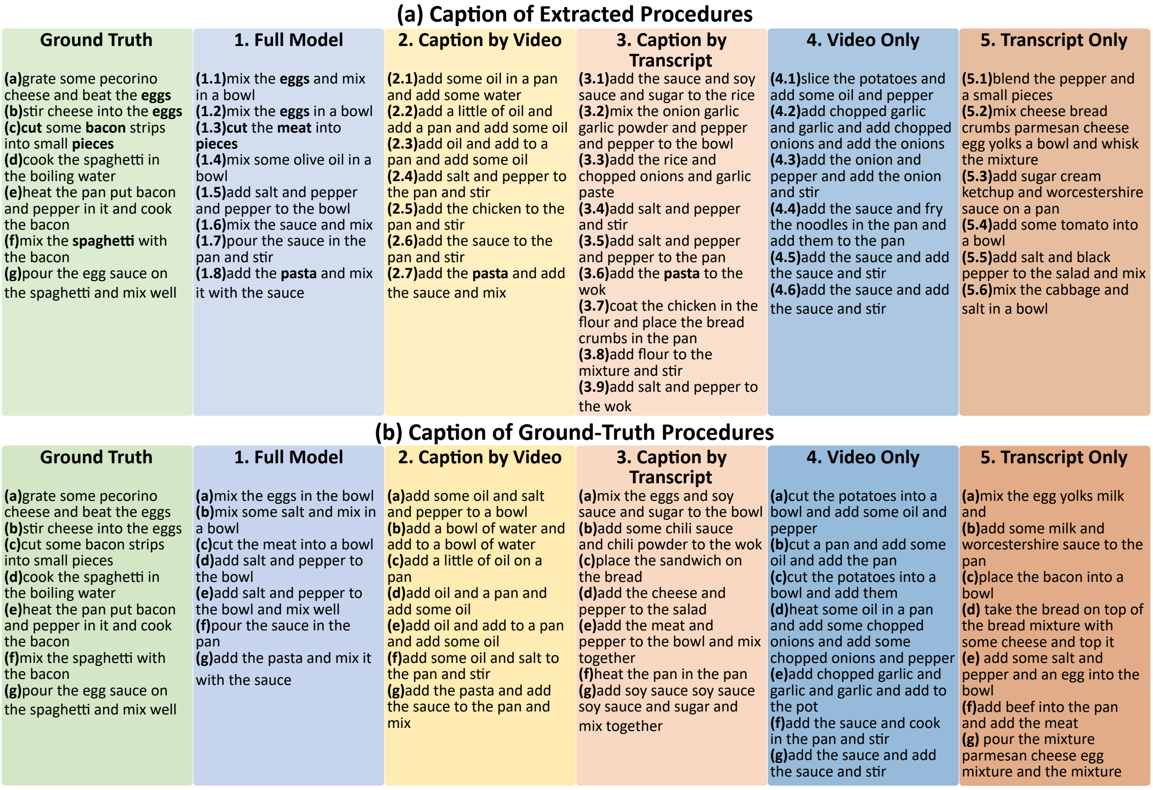
4.5 Result on Procedure Captioning

For evaluating procedure captioning, we consider two baseline models: (1) Bi-LSTM with temporal attention (Yao et al., 2015) (2) an end-to-end transformer based video dense captioning model proposed in (Zhou et al., 2018b). We evaluate the performance of captioning on two different procedures: (1) the ground-truth procedure; (2) the procedure extracted by models. In Table 2, we demonstrate that using ground-truth procedures can generate better captions. Additionally, our model achieves the SOTA result on BLEU-4 and METEOR metrics when using the ground-truth procedures as well as the extracted procedures.

4.6 Ablation and analysis

We conduct the ablation experiments to show the effectiveness of utilizing transcripts. Table 3 lists the results.

The Video Only Model only relies on video information for all modules. The Captioning by Video Model fuses transcripts during the procedure extraction which shows the transcript is effective for the extracting procedure. The Caption by Transcript Model only uses transcripts for captioning. Compared with the Caption by Video Model, we find that only using transcripts for captioning decreases performance. The reason is that only using transcripts for captioning will miss several actions appearing in the video but not mentioned in the transcript. The full Model achieves state-



(a) Caption of Extracted Procedures

Ground Truth	1. Full Model	2. Caption by Video	3. Caption by Transcript	4. Video Only	5. Transcript Only
(a)grate some pecorino cheese and beat the **eggs** **(b)**stir cheese into the **eggs** **(c)cut** some **bacon** strips into small **pieces** **(d)**cook the spaghetti in the boiling water **(e)**heat the pan put bacon and pepper in it and cook the bacon **(f)**mix the **spaghetti** with the bacon **(g)**pour the egg sauce on the spaghetti and mix well	**(1.1)**mix the **eggs** and mix in a bowl **(1.2)**mix the **eggs** in a bowl **(1.3)cut** the **meat** into **pieces** **(1.4)**mix some olive oil in a bowl **(1.5)**add salt and pepper and pepper to the bowl **(1.6)**mix the sauce and mix **(1.7)**pour the sauce in the pan and stir **(1.8)**add the **pasta** and mix it with the sauce	**(2.1)**add some oil in a pan and add some water **(2.2)**add a little of oil and add a pan and add some oil **(2.3)**add oil and add to a pan and add some oil **(2.4)**add salt and pepper to the pan and stir **(2.5)**add the chicken to the pan and stir **(2.6)**add the sauce to the pan and stir **(2.7)**add the **pasta** and add the sauce and mix	**(3.1)**add the sauce and soy sauce and sugar to the rice **(3.2)**mix the onion garlic garlic powder and pepper and pepper to the bowl **(3.3)**add the rice and chopped onions and garlic paste **(3.4)**add salt and pepper and stir **(3.5)**add salt and pepper and pepper to the pan **(3.6)**add the **pasta** to the wok **(3.7)**coat the chicken in the flour and place the bread crumbs in the pan **(3.8)**add flour to the mixture and stir **(3.9)**add salt and pepper to the wok	**(4.1)**slice the potatoes and add some oil and pepper **(4.2)**add chopped garlic and garlic and add chopped onions and add the onions **(4.3)**add the onion and pepper and add the onion and stir **(4.4)**add the sauce and fry the noodles in the pan and add them to the pan **(4.5)**add the sauce and add the sauce and stir **(4.6)**add the sauce and add the sauce and stir	**(5.1)**blend the pepper and a small pieces **(5.2)**mix cheese bread crumbs parmesan cheese egg yolks a bowl and whisk the mixture **(5.3)**add sugar cream ketchup and worcestershire sauce on a pan **(5.4)**add some tomato into a bowl **(5.5)**add salt and black pepper to the salad and mix **(5.6)**mix the cabbage and salt in a bowl

(b) Caption of Ground-Truth Procedures

Ground Truth	1. Full Model	2. Caption by Video	3. Caption by Transcript	4. Video Only	5. Transcript Only
(a)grate some pecorino cheese and beat the eggs **(b)**stir cheese into the eggs **(c)**cut some bacon strips into small pieces **(d)**cook the spaghetti in the boiling water **(e)**heat the pan put bacon and pepper in it and cook the bacon **(f)**mix the spaghetti with the bacon **(g)**pour the egg sauce on the spaghetti and mix well	**(a)**mix the eggs in the bowl **(b)**mix some salt and mix in a bowl **(c)**cut the meat into a bowl **(d)**add salt and pepper to the bowl **(e)**add salt and pepper to the bowl and mix well **(f)**pour the sauce in the pan **(g)**add the pasta and mix it with the sauce	**(a)**add some oil and salt and pepper to a bowl **(b)**add a bowl of water and add to a bowl of water **(c)**add a little of oil on a pan **(d)**add oil and a pan and add some oil **(e)**add oil and add to a pan and add some oil **(f)**add some oil and salt to the pan and stir **(g)**add the pasta and add the sauce to the pan and mix	**(a)**mix the eggs and soy sauce and sugar to the bowl **(b)**add some chili sauce and chili powder to the wok **(c)**place the sandwich on the bread **(d)**add the cheese and pepper to the salad **(e)**add the meat and pepper to the bowl and mix together **(f)**heat the pan in the pan **(g)**add soy sauce soy sauce soy sauce and sugar and mix together	**(a)**cut the potatoes into a bowl and add some oil and pepper **(b)**cut a pan and add some oil and add the pan **(c)**cut the potatoes into a bowl and add them **(d)**heat some oil in a pan and add some chopped onions and add some chopped onions and pepper **(e)**add chopped garlic and garlic and garlic and add to the pot **(f)**add the sauce and cook in the pan and stir **(g)**add the sauce and add the sauce and stir	**(a)**mix the egg yolks milk and **(b)**add some milk and worcestershire sauce to the pan **(c)**place the bacon into a bowl **(d)** take the bread on top of the bread mixture with some cheese and top it **(e)** add some salt and pepper and an egg into the bowl **(f)**add beef into the pan and add the meat **(g)** pour the mixture parmesan cheese egg mixture and the mixture

Figure 4: The procedure captions, which are generated based on the **Extracted Procedures** and the **Ground-Truth Procedures**. (best review in color)

of-the-art results on procedure extraction and captioning, while Caption by Video Model gets better results on captioning for the ground-truth procedure. To sum up, both video frame frames and transcripts are important for the task.

We study several captioning results and find that the *Caption by Video Model* tends to generate general descriptions such as ”*add ...*” for all steps. Nonetheless, our model tends to generate various fine-grained captions. Motivated by this, we conduct another experiment to use cherry picked sentence like *add the chicken (or beef, carrot, onion, etc.) to the pan and stir* or *add pepper and salt to the bowl* as the captions for all procedures and can still achieve a good result on BLEU (4.0+) and METEOR (16.0+). We find that the distribution of captions in this dataset is biased because there are many similar procedure descriptions even in different recipes.

4.7 Case study

We also present a qualitative analysis based on the case study shown in Figures 3 and 4 (best viewed in color).

Figure 3 visualizes the ground-truth procedures and the predicted procedures. The horizontal axis is the time and the number on each small ribbon is the ID of the procedure. We have slightly shifted the overlapping procedures in order to show the results more clearly. It can be seen that the extracted procedures by our full model have the most similar trend with the ground-truth procedures.

Figure 4 presents the generated captions on extracted procedures (Fig.4a) and ground-truth procedures (Fig.4b) separately. Each column shows captioning results from one model, and the first column is the ground-truth result. On one hand, only the full model can generate *eggs* in the procedure (1.1) and (1.2), which is also an important ingredient entity in the ground-truth captions. On the other hand, the ingredient *bacon* in ground-truth caption (c) is ignored by all models. In fact, our *Full Model* predicts *meat* synonyms of bacon. Besides, the *Full Model* can also generate the action *cut* and the final state of ingredient *pieces* mentioned in transcript, while it is hard to recognize using only video signals.

5 Conclusion

In this paper, we propose a framework for procedure extraction and captioning modeling in instructional videos. Our model use narrated transcripts of each video as the supplementary information and can help to predict and caption procedures better. The extensive experiments demonstrate that our model achieves state-of-the-art results on the YouCookII dataset, and ablation studies indicate the effectiveness of utilizing transcripts.

Acknowledgments

We thank the reviewers for their carefully reading and suggestions. This work was supported by the National Natural Science Foundation of China (No. 61370137), the National Basic Research Program of China (No.2012CB7207002), the Ministry of Education - China Mobile Research Foundation Project (2016/2-7).

References

Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. 2016. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583.

Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. 2018. Learning from narrated instruction videos. *IEEE transactions on pattern analysis and machine intelligence*, 40(9):2194–2208.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. 2014. Weakly supervised action labeling in videos under ordering constraints. In *European Conference on Computer Vision*, pages 628–643. Springer.

Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. 2017. Sst: Single-stream temporal action proposals. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2911–2920.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Xuguang Duan, Wenbing Huang, Chuang Gan, Jingdong Wang, Wenwu Zhu, and Junzhou Huang. 2018. Weakly supervised dense event captioning in videos. In *Advances in Neural Information Processing Systems*, pages 3063–3073.

Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. 2016. Daps: Deep action proposals for action understanding. In *European Conference on Computer Vision*, pages 768–784. Springer.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 706–715.

Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. 2018. Jointly localizing and describing events for dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7492–7500.

Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin P Murphy. 2015. What's cookin'? interpreting cooking videos using text, speech and vision. *North American Chapter of the Association for Computational Linguistics*, pages 143–152.

Iftekhar Naim, Young C Song, Qiguang Liu, Liang Huang, Henry Kautz, Jiebo Luo, and Daniel Gildea. 2015. Discriminative unsupervised alignment of natural language instructions with corresponding video segments. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 164–174.

Benjamin Packer, Kate Saenko, and Daphne Koller. 2012. A combined pose, object, and feature model for action understanding. In *CVPR*, pages 1378–1385. Citeseer.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Vasili Ramanishka, Abir Das, Dong Huk Park, Subhashini Venugopalan, Lisa Anne Hendricks, Marcus Rohrbach, and Kate Saenko. 2016. Multimodal video description. In *Proceedings of the 24th*

ACM international conference on Multimedia, pages 1092–1096. ACM.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Marcus Rohrbach, Anna Rohrbach, Michaela Regneri, Sikandar Amin, Mykhaylo Andriluka, Manfred Pinkal, and Bernt Schiele. 2016. Recognizing fine-grained and composite activities using hand-centric features and script data. *International Journal of Computer Vision*, 119(3):346–373.

Ozan Sener, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. 2015. Unsupervised semantic parsing of video collections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4480–4488.

Zheng Shou, Dongang Wang, and Shih-Fu Chang. 2016. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1049–1058.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond J Mooney, and Kate Saenko. 2015. Translating videos to natural language using deep recurrent neural networks. *North American Chapter of the Association for Computational Linguistics*, pages 1494–1504.

Chunyu Wang, Yizhou Wang, and Alan L Yuille. 2013a. An approach to pose-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 915–922.

Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. 2018a. Bidirectional attentive fusion with context gating for dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7190–7198.

LiMin Wang, Yu Qiao, and Xiaoou Tang. 2013b. Motionlets: Mid-level 3d parts for human motion recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2674–2681.

Limin Wang, Yu Qiao, and Xiaoou Tang. 2014. Video action detection with relational dynamic-poselets. In *European Conference on Computer Vision*, pages 565–580. Springer.

Xin Wang, Yuanfang Wang, and William Yang Wang. 2018b. Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning. *North American Chapter of the Association for Computational Linguistics*, 2:795–801.

Yilei Xiong, Bo Dai, and Dahua Lin. 2018. Move forward and tell: A progressive generator of video descriptions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 468–483.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5288–5296.

Jun Xu, Ting Yao, Yongdong Zhang, and Tao Mei. 2017. Learning multimodal attention lstm networks for video captioning. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 537–545. ACM.

Yang Yang, Imran Saleemi, and Mubarak Shah. 2013. Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1635–1648.

Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. 2015. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, pages 4507–4515.

Huanyu Yu, Shuo Cheng, Bingbing Ni, Minsi Wang, Jian Zhang, and Xiaokang Yang. 2018. Fine-grained video captioning for sports narrative. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6006–6015.

Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. 2016. Video summarization with long short-term memory. In *European conference on computer vision*, pages 766–782. Springer.

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018a. Towards automatic learning of procedures from web instructional videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. 2018b. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748.