Delving Deeper into the Decoder for Video Captioning

Haoran Chen and **Jianmin Li** and **Xiaolin Hu** [1]

Abstract. Video captioning is an advanced multi-modal task which aims to describe a video clip using a natural language sentence. The encoder-decoder framework is the most popular paradigm for this task in recent years. However, there exist some problems in the decoder of a video captioning model. We make a thorough investigation into the decoder and adopt three techniques to improve the performance of the model. First of all, a combination of variational dropout and layer normalization is embedded into a recurrent unit to alleviate the problem of overfitting. Secondly, a new online method is proposed to evaluate the performance of a model on a validation set so as to select the best checkpoint for testing. Finally, a new training strategy called *professional learning* is proposed which uses the strengths of a captioning model and bypasses its weaknesses. It is demonstrated in the experiments on Microsoft Research Video Description Corpus (MSVD) and MSR-Video to Text (MSR-VTT) datasets that our model has achieved the best results evaluated by BLEU, CIDEr, METEOR and ROUGE-L metrics with significant gains of up to 18% on MSVD and 3.5% on MSR-VTT compared with the previous state-of-the-art models.

1 Introduction

The video captioning task aims to automatically generate a human-readable sentence to describe the content of a video clip that is usually 10 to 30 seconds long. Videos "in the wild" cover a variety of scenes and actions. Objects and relations between them in a video clip are required to be captured in order to determine nouns and space relations in a sentence. It is also necessary to model temporal relations in order to describe an event that lasts for a few seconds. It is therefore a difficult challenge to learn a model to generate an adequate description for a short video clip.

The encoder-decoder framework has become the mainstream approach in the field of video captioning motivated by the success in machine translation and image captioning. For the decoder, it is most common to utilize long-short term memory (LSTM) [19] or gated recurrent unit (GRU) [10] to map low-dimensional video representation, produced by the encoder, to a variable length sequence. Attention mechanism is also widely applied in video captioning models to generate visual features dynamically as input to recurrent neural network (RNN) units according to different contexts [27, 28, 15, 20, 36].

However, there are still some non-negligible problems in the decoder for video captioning. Firstly, video caption decoders usually suffer from the serious problem of overfitting. Dropout [32] is a common technique to prevent overfitting in convolutional network (ConvNet), but its application in recurrent networks is not so effective as researchers expected[46]. Several methods have been proposed to apply dropout to recurrent networks [25, 31]. Variational dropout is a theoretically grounded method that has been adopted widely in kinds of deep learning frameworks [13]. However, variational dropout slows down convergence speed and increases training time significantly. Layer normalization is proposed to accelerate convergence speed of RNN by stabilizing internal dynamics [2]. As far as we know, the effect of layer normalization has not been explored in the context of video captioning before.

Then, it is a common practice that the loss value on the validation set is used as a metric to choose a model for testing. However, some metrics from natural language processing (NLP), e.g. BLEU[26], METEOR[5], CIDEr[34] and ROUGE-L[23], are exploited to assess model's performance in testing. The divergence of evaluation between validation and testing phases leads to deteriorated performance in inference. Some researchers utilize one of those metrics to choose the best model for testing, such as BLEU and CIDEr. However, single metric can't reflect the overall performance of video captioning system since all of these metrics are reported in video captioning literature.

Last but not least, most of the previous training algorithms have a common defect that they treat all the training samples equally which leads to what we call "absolute equalitarianism" in learning. The model trained in this way is likely to learn an intersection of the annotations for each video which consists of frequent words and phrases and is inclined to forget advanced words and complicated sentence structures since they vary too much from sentence to sentence.

In this work, we propose three methods to solve each of these problems accordingly. First, variational dropout is used to reduce overfitting of the decoder and layer normalization is employed to counteract the prolonged training time brought on by dropout usage. Secondly, a new selecting method is proposed to choose the best model for testing based on comprehensive consideration of the various metrics. Besides, a novel training strategy, called *professional learning*, is presented which trains the model in *teacher forcing* way to learn basic knowledge using all the annotations equally and then optimize the same model with emphasis upon the samples it is good at. We perform extensive experiments on MSVD (YouTube2Text) [17] and MSR-VTT [43] datasets. And empirical results prove the effectiveness of the proposed methods in video captioning.

2 Related Work

We pay our attention to regularization and training strategies for RNN as well as encoder-decoder-based video captioning literature in deep learning for the reason that they are highly correlated with our work.

---

[1] State Key Laboratory of Intelligent Technology and Systems, Institute for Artificial Intelligence, Beijing National Research Center for Information Science and Technology, THBI, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China., email: chr17@mails.tsinghua.edu.cn, {lijianmin, xlhu}@tsinghua.edu.cn, corresponding author: Xiaolin Hu.

2.1 Video Captioning

Semantic information has been widely applied to assist video captioning models in generating annotations. Semantic SVO triplets and semantic hierarchies are exploited to output a brief sentence that summarizes the content in a video in [17]. Tags of a video, i.e. key words from human annotations, are joined together with RNN parameters by matrix factorization technique to gain better sense of themes for images/videos [14]. In another work, visual features and sentences embedding are projected to a joint low-dimension space. Semantic consistency between sentence content and video visual information is guaranteed by minimizing the distance between two embedded vectors in the joint space [15]. Higher-order object interactions are modeled to improve the performance of video understanding [24]. High level semantic features derived from a video action classifier and an object detector is utilized to enrich video representation features in [1]. We notice that classification results produced by video action classifiers and image classifiers have hardly utilized in previous works and we find that they can be naturally integrated into the semantic information provided by video tagging networks. With enhanced semantic information for video, a model is able to describe it in a more comprehensive way.

Inspired by successful application of attention mechanism in machine translation [4], object detection [3] and image captioning [44], attention mechanism has been applied to video captioning task in various ways[45, 15, 20, 27, 28, 36, 37, 41, 29]. Attention mechanism contributes to the caption generation by distilling useful information dynamically according to the runtime context. Though it is extremely popular in recent years, we find attention mechanism does not always contribute to the performance of a model since the video captioning system may overfit on the training set more easily.

External or internal knowledge from a dataset is also utilized to provide more information for video captioning models. In MARN, memory structure is proposed to learn the relationship between a word and its various related visual features in order to achieve a more comprehensive understanding of the video content [29]. In TAMoE, external Wikipedia corpus is explored by primitive experts and it helps to transfer the knowledge learned from seen topics to unseen topics as well as improve the quality of generated captions on seen topics [38].

2.2 Regularization and Training Strategies for Recurrent Neural Network

Dropout and normalization operation are two common regularization methods for RNN. Dropout is a simple technique to reduce overfitting in neural networks but it does not work very well in RNN [32]. A new method of applying dropout shows that the dropout operator should only be applied to the non-recurrent connections [46]. Dropout is proved to be a Bayesian approximation for representing uncertainty [12] and variational dropout is proposed for RNN accordingly [13]. Another method, that dropout mask should only be used on the update vector in RNN, is only supported by experiments [31]. It is a common phenomenon that dropout prolongs training time. Batch normalization is proposed to tackle the problem of internal covariate shift so that training process is accelerated [22]. However, it requires different running averages of input statistics at different time steps when applied to an RNN unit which hinders its application in variable length sequence training. It has been supported by experiments that RNN, especially with long sequences, benefits significantly from Layer Normalization [2].

The most common and intuitive training strategy is *teacher forcing*

which can be traced back to the end of 80s in 20th century[39]. The ground truth $d(t)$ for a sequence s is exploited as a part of the input $x(t + 1)$ to the recurrent unit during training. This leaves wide divergence between training process and inference phase since the unit output $y(t)$ is utilized as the input $x(t + 1)$ to the recurrent unit. In another word, it is called exposure bias. A model trained by *teacher forcing* is unlike to adapt to testing process because of the divergence. A training method called scheduled sampling is proposed to minimize the gap between training phase and inference phase, which gently transfers the training phase from using ground truth as part of input for the recurrent unit to using model-generated tokens as part of input[6]. Although the problem of the divergence between training process and testing process is alleviated, the optimization goal of the scheduled sampling deviates from the natural-looking sentences [21]. Adversarial domain technique, called *professor forcing*, is exploited to align dynamics of RNN during training and inference. *Professor forcing* can, to some extent, act as a regularizer and has better ability to capture the long-term dependencies reported in [16]. Besides, reinforcement learning methods are proposed to train recurrent networks, e.g. self-critical sequence training (SCST) [30], CIDEnt-reward model [28], Hierarchical Reinforcement Learning (HRL) [36, 40]. Multi-task learning helps a model produce better input representation and improves generalization of a particular task by jointly training a model with related tasks [27]. In curriculum learning, samples are introduced to train the model according to a predefined and fixed schedule called curriculum. All the samples are treated equally once introduced during optimization process [7].

Unfortunately, from the perspective of the whole training process, all of these training strategies treat training samples equally which leads the model to learn from a small intersection of annotations corresponding to each video. It results to limited vocabulary and repeated sentences in generated captions.

3 Model and Proposed Methods

The encoder-decoder framework [10, 35] for video captioning is adopted in our work. In the encoder, a video vid is split into K frames. A pre-trained image classifier is applied on each frame and it outputs feature maps $v_{img,k}, k = 1, 2, ..., K$ and classification results $cls_{img,k}$ for each frame. Spatial feature map v_{img} and image classification result cls_{img} are obtained for each video by applying average operation over time axis on $v_{img,k}$ and $cls_{img,k}$. A pre-trained video action classifier is employed to produce global spatio-temporal feature map v_{vid} and action classification result cls_{vid} over all the frames. A tagging network is trained to generate semantic tags s_t for each video. On MSVD dataset, the output of the encoder for a video clip vid_i is composed of visual spatio-temporal features v_i and semantic information s_i:

$$v_i = v_{vid}, \quad s_i = \text{concat}(s_t, cls_{img}). \tag{1}$$

MSR-VTT dataset is relatively complicated compared to MSVD dataset. Thus, extra features are utilized in the experiments on MSR-VTT dataset:

$$v_i = \text{concat}(v_{vid}, v_{img}), s_i = \text{concat}(s_t, cls_{vid}, cls_{img}). \tag{2}$$

3.1 Semantic-GRU-based Decoder with Variational Dropout and Layer Normalization

Traditional RNNs often suffer from serious overfitting on training set which deteriorates their performance on inference. Variational

dropout is proposed in [13] based on the mathematical grounding in deep Gaussian process. It has been widely applied in RNNs to prevent overfitting but it slows down the training process. Layer normalization has been proved to be very effective in stabilizing internal state dynamics in recurrent networks and Transformer [2]. In the consideration of preventing overfitting as well as time and energy efficiency, variational dropout and layer normalization are embedded in our decoder together.

The traditional gated recurrent unit (GRU) which has one fewer gate than LSTM and is capable of learning to acquire temporal dependencies across various scales [10, 11]. The unit consists of two gates: an update gate z_t and a reset gate r_t(3). Suppose we have input $\boldsymbol{x}_t \in \mathbb{R}^{n_x}$ and previous hidden state $\boldsymbol{h}_{t-1} \in \mathbb{R}^{n_h}$ at time step $t-1$, and then z_t and r_t can be computed as follows:

$$z_t = \sigma\left(\mathbf{W}_z \boldsymbol{x}_t + \mathbf{U}_z \boldsymbol{h}_{t-1}\right), \quad r_t = \sigma\left(\mathbf{W}_r \boldsymbol{x}_t + \mathbf{U}_r \boldsymbol{h}_{t-1}\right), \quad (3)$$

where $\mathbf{W}_z \in \mathbb{R}^{n_h \times n_x}$, $\mathbf{U}_z \in \mathbb{R}^{n_h \times n_h}$, $\mathbf{W}_r \in \mathbb{R}^{n_h \times n_x}$ and $\mathbf{U}_r \in \mathbb{R}^{n_h \times n_h}$. The candidate activation $\tilde{\boldsymbol{h}}_t$ is computed as follows:

$$\tilde{\boldsymbol{h}}_t = \tanh\left(\mathbf{W}\boldsymbol{x}_t + \boldsymbol{r}_t \odot \left(\mathbf{U}\boldsymbol{h}_{t-1}\right)\right), \quad (4)$$

where $\mathbf{W} \in \mathbb{R}^{n_h \times n_x}$, $\mathbf{U} \in \mathbb{R}^{n_h \times n_h}$ and $\odot$ denotes element-wise multiplication. Unlike the popular version of the candidate activation $\tilde{\boldsymbol{h}}_t = \tanh\left(\mathbf{W}\boldsymbol{x}_t + \mathbf{U}\left(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}\right)\right)$ [11], the original approach [9] is chosen to compute $\tilde{\boldsymbol{h}}_t$ for the sake of consistency. The activation of the gated recurrent unit at time t is a linear interpolation between the previous activation $\boldsymbol{h}_{t-1}$ and the candidate activation $\tilde{\boldsymbol{h}}_t$:

$$\boldsymbol{h}_t = (1 - z_t)\boldsymbol{h}_{t-1} + z_t\tilde{\boldsymbol{h}}_t. \quad (5)$$

Inspired by [14], the weights $\mathbf{W}, \mathbf{U}, \mathbf{V}$ are transformed into semantics-dependent weight matrices to improve the quality of the generated captions as following shows.

$$\hat{\boldsymbol{x}}_{t,\star} = \mathbf{W}_{\star 1}\boldsymbol{s} \odot \mathbf{W}_{\star 2}\boldsymbol{x}_t, \quad (6)$$

$$\hat{\boldsymbol{h}}_{t,\star} = \mathbf{U}_{\star 1}\boldsymbol{s} \odot \mathbf{U}_{\star 2}\boldsymbol{h}_t, \quad (7)$$

$$\hat{\boldsymbol{v}}_\star = \mathbf{V}_{\star 1}\boldsymbol{s} \odot \mathbf{V}_{\star 2}\boldsymbol{v}, \quad (8)$$

$$z_t = \sigma\left(\mathbf{W}_{z3}\hat{\boldsymbol{x}}_{t,z} + \mathbf{U}_{z3}\hat{\boldsymbol{h}}_{t-1,z} + \mathbf{V}_{z3}\hat{\boldsymbol{v}}_z\right), \quad (9)$$

$$\boldsymbol{r}_t = \sigma\left(\mathbf{W}_{r3}\hat{\boldsymbol{x}}_{t,r} + \mathbf{U}_{r3}\hat{\boldsymbol{h}}_{t-1,r} + \mathbf{V}_{r3}\hat{\boldsymbol{v}}_r\right), \quad (10)$$

$$\tilde{\boldsymbol{h}}_t = \tanh\left(\mathbf{W}_{h3}\hat{\boldsymbol{x}}_{t,h} + \boldsymbol{r}_t \odot \left(\mathbf{U}_{h3}\hat{\boldsymbol{h}}_{t-1,h}\right) + \mathbf{V}_{h3}\hat{\boldsymbol{v}}_h\right), (11)$$

$$\boldsymbol{h}_t = (1 - z_t)\boldsymbol{h}_{t-1} + z_t\tilde{\boldsymbol{h}}_t, \quad (12)$$

where $\Psi_{\star 1} \in \mathbb{R}^{n_f \times n_s}$, $\mathbf{W}_{\star 2} \in \mathbb{R}^{n_f \times n_x}$, $\mathbf{U}_{\star 2} \in \mathbb{R}^{n_f \times n_h}$, $\mathbf{V}_{\star 2} \in \mathbb{R}^{n_f \times n_v}$, $\Psi_{\star 3} \in \mathbb{R}^{n_h \times n_f}$, $\star \in \{z, r, h\}$, $\Psi \in \{\mathbf{W}, \mathbf{U}, \mathbf{V}\}$.

With the consideration of preventing overfitting and training efficiency, layer normalization and variational dropout are applied to the semantic GRU.

$$\hat{\boldsymbol{x}}_{t,\star} = \mathbf{W}_{\star 1}\left(\boldsymbol{s} \odot \boldsymbol{m}_{\star,s}\right) \odot \mathbf{W}_{\star 2}\left(\boldsymbol{x}_t \odot \boldsymbol{m}_{\star,x}\right), \quad (13)$$

$$\hat{\boldsymbol{h}}_{t,\star} = \mathbf{U}_{\star 1}\left(\boldsymbol{s} \odot \boldsymbol{m}_{\star,s}\right) \odot \mathbf{U}_{\star 2}\left(\boldsymbol{h}_t \odot \boldsymbol{m}_{\star,h}\right), \quad (14)$$

$$\hat{\boldsymbol{v}}_\star = \mathbf{V}_{\star 1}\left(\boldsymbol{s} \odot \boldsymbol{m}_{\star,s}\right) \odot \mathbf{V}_{\star 2}\left(\boldsymbol{v} \odot \boldsymbol{m}_{\star,v}\right), \quad (15)$$

$$z_t = \sigma\left(\text{LN}\left(\mathbf{W}_{z3}\hat{\boldsymbol{x}}_{t,z} + \mathbf{U}_{z3}\hat{\boldsymbol{h}}_{t-1,z} + \mathbf{V}_{z3}\hat{\boldsymbol{v}}_z\right)\right), \quad (16)$$

$$\boldsymbol{r}_t = \sigma\left(\text{LN}\left(\mathbf{W}_{r3}\hat{\boldsymbol{x}}_{t,r} + \mathbf{U}_{r3}\hat{\boldsymbol{h}}_{t-1,r} + \mathbf{V}_{r3}\hat{\boldsymbol{v}}_r\right)\right), \quad (17)$$

$$\tilde{\boldsymbol{h}}_t = \tanh\left(\text{LN}\left(\mathbf{W}_{h3}\hat{\boldsymbol{x}}_{t,h} + \boldsymbol{r}_t \odot \left(\mathbf{U}_{h3}\hat{\boldsymbol{h}}_{t-1,h}\right) + \mathbf{V}_{h3}\hat{\boldsymbol{v}}_h\right)\right), \quad (18)$$

$$\boldsymbol{h}_t = (1 - z_t)\boldsymbol{h}_{t-1} + z_t\tilde{\boldsymbol{h}}_t, \quad (19)$$

where $\boldsymbol{m}$ is time-invariant dropout mask and LN denotes layer normalization. Inspired by the "Go deeper, not wider" principle of philosophy[33, 18], we stack two variational-normalized semantic GRU (VNS-GRU) layers together and reduce the internal embedding dimension n_f. As a result of it, the model has stronger decoding ability than the one-layer model, even though it has fewer parameters than the latter one.

3.2 Selection of Model for Testing: Comprehensive Selection Method

Given n metrics $M = [m_0, m_1, ..., m_{n-1}]$ with the corresponding weights $W = [w_0, w_1, ..., w_{n-1}]$, the overall performance o of a model can be evaluated as follows:

$$o = \sum_i \frac{w_i \cdot v_i}{b_i}, \quad (20)$$

where v_i is the value of metric m_i and b_i is by far the best value for metric m_i. At the end of each epoch, v_i is computed based on the output of the model on the validation set. b_i for each metric and o_b are updated subsequently if necessary. A checkpoint of the model is saved whenever o_b is updated during training process.

In our method, the performance of a model is evaluated by n metrics on the validation set with predefined weights W. If $M = [CrossEntropy]$, then the best model is chosen solely based on cross entropy loss for testing. Our method can be embedded into an end-to-end training framework to assess a checkpoint by arbitrary number of metrics. Most of the existing selection methods can be regarded as special cases of this method.
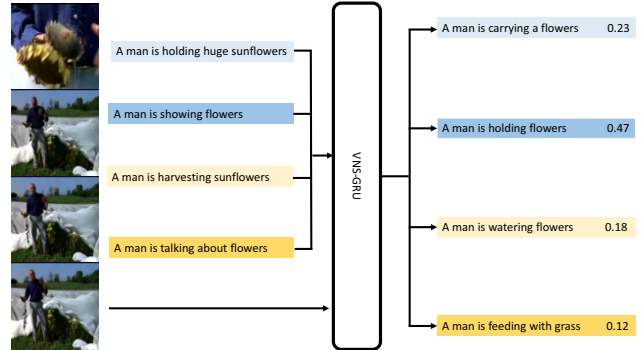
Figure 1. VNS-GRU stands for semantic gated recurrent unit model with variational dropout and layer normalization. Video features are fed into our model and the model is optimized by different annotations with corresponding weights and it is called *professional learning*.

3.3 Training Strategy: *Professional Learning*

For previous training strategies, models are optimized on all training samples equally which leads to "absolute equalitarianism" implicitly. "Absolute equalitarianism" in video captioning task is a phenomenon that a model's knowledge for the common part of all training samples is enhanced iteratively and the model is inclined to forget advanced words and complicated language grammar since they vary too much from sentence to sentence. The intersection of human annotations for

a video usually consists of limited number of frequent vocabulary and elementary language grammar. It partially explains why an ordinary video captioning model cannot generate a sentence that is competitive with human annotations.

Inspired by the higher education system in the real word, we propose a novel training method called *professional learning* (Alg. 1). University students get a liberal education to consolidate and widen their basic knowledge and skills first and then choose a particular specialty to develop their professional skills. Similarly, in *professional learning*, a model will be trained by optimizing losses computed with training samples equally in the first phase, which is called *teacher forcing* or *general learning*. In the second phase, n annotations $\mathbf{A}^{(k)} = [\boldsymbol{a}_0^{(k)}, \boldsymbol{a}_1^{(k)}, ..., \boldsymbol{a}_{n-1}^{(k)}]$ are sampled for the video k. The possibility distributions $\mathbf{P}^{(k)} = [\boldsymbol{p}_0^{(k)}, \boldsymbol{p}_1^{(k)}, ..., \boldsymbol{p}_{n-1}^{(k)}]$ for each token in the generated captions $\mathbf{C}^{(k)} = [\boldsymbol{c}_0^{(k)}, \boldsymbol{c}_1^{(k)}, ..., \boldsymbol{c}_{n-1}^{(k)}]$ are computed by the model, where $\boldsymbol{c}_i^{(k)}$ is guided by $\boldsymbol{a}_i^{(k)}$. Cross entropy loss $\boldsymbol{l}^{(k)}$ is produced for pairs of possibility distribution $\boldsymbol{p}_i^{(k)}$ and human annotation $\boldsymbol{a}_i^{(k)}$:

$$\boldsymbol{l}^{(k)} = [l_0^{(k)}, l_1^{(k)}, ..., l_{n-1}^{(k)}]^\mathsf{T}, l_i^{(k)} = \text{mean}\left(-\boldsymbol{a}_i^{(k)} \log \boldsymbol{p}_i^{(k)}\right). \quad (21)$$

The cross entropy loss $\boldsymbol{l}^{(k)}$ with weights $\boldsymbol{\beta}^{(k)}$ is utilized to optimize a video captioning system and the weighted loss is formulated as:

$$Loss\left(\mathbf{A}, \mathbf{S}, \mathbf{V}; \Theta\right) = \frac{1}{bs} \sum_{i=0}^{bs-1} \boldsymbol{\beta}^{(i)} \boldsymbol{l}^{(i)}, \quad (22)$$

where $\mathbf{A}$, $\mathbf{S}$, $\mathbf{V}$ and Θ denote human annotations, semantic information, visual features, and model parameters respectively.

Given loss $l_i^{(k)}$, the corresponding annotation $\boldsymbol{a}_i^{(k)}$ and the corresponding caption $\boldsymbol{c}_i^{(k)}$, small loss $l_i^{(k)}$ indicates the model "is good at" generating a caption $\boldsymbol{c}_i^{(k)}$ which is similar to $\boldsymbol{a}_i^{(k)}$; large loss $l_i^{(k)}$ indicates the model is inclined to generate a caption which is dissimilar to $\boldsymbol{a}_i^{(k)}$. The former property is called a strength of the model and the latter is called a weakness. The generated caption $\boldsymbol{c}_i^{(k)}$ has large weight $\beta_i^{(k)}$ in optimization if the corresponding loss $l_i^{(k)}$ is small; $\boldsymbol{c}_i^{(k)}$ has small $\beta_i^{(k)}$ if $l_i^{(k)}$ is large. Strengths of a model are enhanced and weaknesses are bypassed in this way. The model is able to learn unique words and advanced grammar rules for they pay more attention on the samples they do well in. Larger size of vocabulary and more diverse sentence structures are expected to be observed in the captions produced by the model trained in this way. Weights $\boldsymbol{\beta}$ consists of two parts:

$$\begin{aligned}\boldsymbol{\beta}^{(i)} &= \gamma \, \text{softmax}\left(-\boldsymbol{l}^{(i)}\right) \\ &+ (1-\gamma) \, \text{softmax}\left(-\text{abs}\left(\boldsymbol{len}^{(i)} - \bar{len}\right)\right),\end{aligned} \quad (23)$$

where γ is a hyper-parameter to modulate the balance between the cross-entropy-related and length-related probability distribution. The first part is probability distribution determined by softmax value of cross entropy so that samples with higher loss values have lower probability and vice versa. The second part is $L1$ distance between machine-generated and average sentence length in a dataset which is to encourage the generation of average-length sentences.

It is easier to generate short sentence with small loss than to generate long sentence with the same loss because of accumulation of errors in RNN. Without the second term in (23), short sentences may gain unproportionally large weights $\boldsymbol{\beta}$ in optimization. If γ is close

to 1, a model is likely to generate short and simple captions for it is relatively easy to fit on such kind of captions and loss values for those samples are small; vice versa.

Algorithm 1: Professional Learning Algorithm

Data: inputs $\mathbf{i}$, human annotations $\mathbf{a}$, total training epoch $epoch_{total}$ and the switch point $epoch_{sw}$

1   $idx_{cur} \leftarrow 0$;
2   **while** $idx_{cur} < epoch_{total}$ **do**
3   **if** $idx_{cur} < epoch_{sw}$ **then** // *general learning* scheme
4   Optimize the model by each video and annotation pair equally;
5   **else** // *professional learning* scheme
6   Compute and optimize the model by weighted loss function with idx_{cur}, human annotations $\mathbf{A}$, semantic information $\mathbf{S}$ and video features $\mathbf{V}$;
7     $idx_{cur} = idx_{cur} + 1$;

For the sake of a smooth transition, a proper schedule for the sampling number n is needed. Examples of the schedule for n can be as follows:

1. Fixed sampling size: $n = c \in \mathbb{Z}^+$.
2. Exponential schedule:

$$n = \begin{cases} 1, & \text{if } idx_{cur} <= epoch_{sw}, \\ b^{\text{floor}((idx_{cur} - epoch_{sw})/\sigma)}, & \text{if } idx_{cur} > epoch_{sw}. \end{cases},$$

where $\sigma \in \mathbb{Z}^+$ is a hyper-parameter that depends on the estimated rate of convergence.

GT: (1) a man is walking (2) the person is walking

GT: (1) a man falls down (2) a man falls over

GT: (1) a man falling down stairs (2) a man who is drunk falls down

Figure 2. Duplicate captions for different videos. GT is short for ground truth. The machine-generated captions for these three video clips are identical: "a man is running". Two actions "walk" and "fall down" are recognized as "run" incorrectly.

4   Experiment

We implement our models and perform experiments under the Tensor-Flow framework. Its source code can be found on GitHub [2].

4.1 Dataset

Two popular datasets for video captioning in recent research are used in experiments, MSVD and MSR-VTT.

4.1.1 MSVD Dataset

The MSVD dataset consists of 1970 short video snippets. Each video clip has length of 10s on average and each of them corresponds to 40 English human annotations in MSVD [17]. The average number of words for each annotation in the dataset is around 7.1 which implies those annotations are composed of relatively short and simple sentences. Following the split setting in [17, 20, 37, 41, 1, 29], we take 1200 video clips as training set, 100 video snippets as validation set and 670 clips as testing set. We tokenize 80839 English sentences and obtain vocabulary with 12596 English words from the training set.

4.1.2 MSR-VTT Dataset

The MSR-VTT 1.0 dataset is composed of 10000 short video clips, which are divided into twenty predefined categories [43]. Each video snippet has roughly 20 English descriptions. The average number of words for each annotation in the dataset is around 9.3 which implies that human annotations in MSR-VTT are more complicated than those in MSVD. We follow data split setting provided by MSRA and take 6513 video clips as training set, 497 video clips as validation set and 2990 video clips as testing set. We tokenize 200000 English human annotations and obtain vocabulary with 13796 English words only from the training set.

4.2 Model Architecture

In the encoder, ResNeXt-101 with 64 paths in each block pre-trained on ImageNet is used as our frame-level feature generator [42]. The 2048-dim feature map for each frame is taken from the output of the global pooling layer in ResNeXt. We also collect the probability distribution of classification for each frame and apply average pooling operation on the frames from the same video. The averaged probability distribution, a 1000-dim vector for each video, is used as part of the semantic features. We choose Efficient Convolutional Network (ECN) [47] pre-trained on Kinetics-400 as our video-level feature generator. The feature map for each video clip is taken from the output of concatenation of the global pooling layers in ECN. Both feature maps from ResNeXt and ECN are scaled to $[0, 1]$. The probability distribution for actions from ECN, which is a 400-dim vector for each video clip, is taken as (a part of) semantic information. In addition, we select 300 key words from dataset vocabulary as tags (semantic clues) for each video. Our tagging network is trained on training and validation sets and is utilized to predict tags for each video and each of them is a 300-dim vector. In all, we have video feature with dimension 1536 and semantic feature with dimension 1300 for each video segment in MSVD dataset (1), and video feature with dimension 3584 and semantic feature with dimension 1700 for each of those in MSR-VTT dataset (2). Note that we reuses the video and semantic features in SAM-SS[8] for the sake of convenience.

For MSVD dataset, our model has settings as follows: $n_v = 1536$, $n_s = 1300$, $n_f = 64$, $n_h = n_x = 512$ and $\gamma = 0.8$. For MSR-VTT dataset, our model has settings as follows: $n_v = 3584$, $n_s = 1700$, $n_f = 128$, $n_h = n_x = 1024$ and $\gamma = 0.4$. Given that all the other hyper-parameters are equal, the number of parameters of a model with $n_f = \frac{n_h}{8}$ is about one-eighth of the one with $n_f = n_h$, while the performance of the former can be comparable with the latter.

4.3 Training Detail

The model is trained for 50 epochs on MSVD dataset and for 80 epochs on MSR-VTT dataset. And the best model is chosen based on the performance on the validation set as described in Section 3.2. In our experiment, B4, CIDEr, METEOR and ROUGEL are used to evaluate the performance of a model and the weights for each of them are set to 0.25(24).

$$overall_i = \left(\frac{B4_i}{B4_b} + \frac{C_i}{C_b} + \frac{M_i}{M_b} + \frac{R_i}{R_b} \right) /4, \qquad (24)$$

where m_b is the current best score on metric $m \in [B4, C, M, R]$[8] and subscript i denotes checkpoint i. The training strategy is switched from *general learning* scheme to *professional learning* scheme at 16th epoch. For MSVD, the sampling number n of annotations for each video is fixed to 16. The sampling number n in MSR-VTT is computed as follows:

$$n = 2^{\text{floor}(e/16)}, \qquad (25)$$

where e is the epoch index during training. The reason for the choice of the sampling schedules is described in Section 4.4.1. A GeForce GTX 1080 Ti is employed to speed up the training process for each of those experiments. The model finishes its training on MSVD dataset within two hours and on MSR-VTT dataset within six hours. Our models are optimized by Adam Algorithm with initial learning rate 2×10^{-4} and global norm gradient clip of 40. We use a weight decay of 0.861 every 1000 steps for MSVD and 0.9455 every 1000 steps for MSR-VTT.

4.4 Ablation Study

4.4.1 Influence of Sampling Schedule in Professional Learning

We perform experiments with different fixed sampling size $n \in \{2, 4, 8, 16\}$ and exponential sampling schedule on MSVD and MSR-VTT dataset. As demonstrated by Table 1, on MSVD dataset, the best performance evaluated by overall score is obtained with $n = 16$. It also can be inferred from Table 1 that, on MSR-VTT dataset, the best performance among different fixed sampling size is obtained with $n = 8$ which is smaller than the one on MSVD dataset. The best performance on MSR-VTT among all schedules is obtained with exponential sampling schedule.

The human annotations in MSVD dataset are simpler than those in MSR-VTT and the diversity of sentences in MSVD is less than that in MSR-VTT. For a video, annotations have more words and sentence structures in common in MSVD than in MSR-VTT. A model is able to achieve better performance with more attention focused on few sentences in MSVD but it needs to allocate its attention more evenly in MSR-VTT to have better metric values. Exponential schedule optimizes the model on different sampling size so that it helps the model learn better hidden patterns from diversified annotations in MSR-VTT.

4.4.2 Effectiveness of Components

As shown by Table 2, we have high-level baseline model, by virtue of enhanced semantic features and highly qualified visual features. If cross entropy is used to select the best model for testing, a low-quality model will be chosen. The chosen model will have better performance if BLEU-4 is used for selecting. Comprehensive Selection Method is

Table 1. Model performance with different fixed sampling number n on MSVD and MSR-VTT dataset. EXP denotes exponential schedule(25). Size denotes sampling size.

DS	Size	B4	C	M	R	Overall(24)
	2	64.0	117.8	41.4	79.3	0.978
	4	62.9	118.9	41.4	79.2	0.975
MSVD	8	64.5	121.0	41.6	79.1	0.987
	16	**66.5**	**121.5**	**42.1**	**79.7**	**1.000**
	EXP	64.1	119.1	41.5	79.1	0.981
	2	44.5	51.8	29.4	62.9	0.980
	4	45.0	51.7	29.4	62.9	0.982
MSR-VTT	8	**46.1**	51.4	29.3	62.8	0.985
	16	45.8	49.6	28.6	62.5	0.968
	EXP	45.3	**53.0**	**29.9**	**63.4**	**0.996**

able to find the model with the satisfying overall performance based on the metric values in validation set. The combination of variational dropout and layer normalization makes a profound impact on the model performance in MSVD and MSR-VTT. Our model is improved by *professional learning* significantly on both datasets which proves the validity of the proposed method, as demonstrated by Table 2. The sampling schedules for *professional learning* are described in Section 4.3. In overall, the performance of our model is improved with the increase in the number of modules or methods.

Table 2. The results of the experiments performed on MSVD and MSR-VTT with or without some components. SEL, VD, LN and PL are the abbreviations for selection method described in Section 3.2, variational dropout and *professional learning*, respectively. XE in the column of SEL denotes cross entropy loss which is used for selecting the model. BLEU4 means that BLEU-4 is used to select the model for testing.

Dataset	SEL	VD&LN	PL	B4	C	M	R
	XE	×	×	60.9	100.3	38.7	76.7
	BLEU4	×	×	57.4	104.4	38.3	75.7
MSVD	√	×	×	57.4	104.4	38.3	75.7
	√	√	×	64.3	117.5	40.9	78.9
	√	√	√	**66.5**	**121.5**	**42.1**	**79.7**
	XE	×	×	40.0	44.3	27.0	60.4
	BLEU4	×	×	40.0	46.6	27.5	60.4
MSR-VTT	√	×	×	41.5	48.8	28.3	61.5
	√	√	×	44.0	50.1	28.7	62.3
	√	√	√	**45.3**	**53.0**	**29.9**	**63.4**

4.4.3 Diversity of the Generated Captions

Poor vocabulary and repeated captions are two serious problems in video captioning task. We count the number of distinct sentences and the number of distinct words in the captions generated by our model for MSVD and MSR-VTT test sets respectively.

As demonstrated by Table 3, for the VS-GRU baseline, only $196/12596 = 1.6\%$ and $342/13796 = 2.5\%$ of all the vocabulary appears in the generated captions for the MSVD and MSR-VTT test sets respectively. For the same model, $670/326 = 2.1$ and $2990/793 = 3.8$ clips of video share the same caption on average in MSVD and MSR-VTT testing respectively. As shown in Fig. 2, three video clips are described with identical captions: "a man is running". Layer normalization enlarges the vocabulary of the model by 13.3% on MSVD test set and 13.5% on MSR-VTT test set. *Professional learning* enlarges vocabulary by 14.9% on MSVD test set and 7.2%

on MSR-VTT test set. These two methods also increase the number of distinct sentences accordingly. Layer normalization stabilizes the internal states of the model so that the model is able to learn delicate pattern hidden in features. In this way, the model can generate different words in response to the slight changes brought by inputs. In some degree, *professional learning* and layer normalization alleviate the problems of poor vocabulary and repeated captions in testing.

Sometimes, our method can generate a caption which is competitive with or even better than a human annotation. In the first video clip of Fig. 3, our model uses "cleaning" to describe the action of human which expresses the purpose of "brushing". In the third video clip, "a group of", which is grammatically correct, is used by our model instead of "group of" in the ground truth, which is a syntax error.

Table 3. Statistics data of distinct sentences and vocabulary size for models without certain component(s). LN and PL denote layer normalization and *professional learning* respectively. VS-GRU stand for a semantic GRU model with variational dropout.

Dataset	LN	PL	Distinct Sentences	Vocabulary Size
	×	×	326	196
MSVD	√	×	376	222
	√	√	**397**	**255**
	×	×	793	342
MSR-VTT	√	×	910	388
	√	√	**915**	**416**

4.5 Comparison with Previous Models

To demonstrate the superiority of our method, we list the performance of our model along with the previous state-of-the-art results from existing computer vision literature. Four metrics in natural language processing, called BLEU-4, CIDEr, METEOR and ROUGE-L, are applied to evaluate the performance of those models numerically.

4.5.1 Comparison on MSVD

Table 4. The results of the models on MSVD. VNS-GRU is a model trained without cls_{img} while VNS-GRU^r is a model trained with it (1). The rest configuration of two models is the same and is described in Section 3. B4, C, M and R stand for BLEU-4, CIDEr, METEOR and ROUGE-L respectively.

Models	B4	C	M	R	Overall(24)
SCN[14]	51.1	77.7	33.5	-	-
MTVC[27]	54.5	92.4	36.0	72.8	0.837
CIDEnt-RL[28]	54.4	88.6	34.9	72.2	0.821
HATT[41]	52.9	73.8	33.8	-	-
ECN[47]	53.5	85.8	35.0	-	-
GRU-EVE[1]	47.9	78.1	35.0	71.5	0.773
MARN[29]	48.6	92.2	35.1	71.9	0.806
SAM-SS[8]	61.8	103.0	37.8	76.8	0.910
VNS-GRU	66.3	116.0	41.4	79.1	0.982
VNS-GRU^r	**66.5**	**121.5**	**42.1**	**79.7**	**1.000**

To the best of our knowledge, our model outperforms all the previous models on all the metrics (Table 4). SCN[14] takes advantage of semantic information produced by video tagging network and ensemble the RNN weights with video tags. Its performance is evaluated on the ensemble of five models. Multi-task video captioning model

without: a cat is playing.
with: a person is cleaning a cat.
GT: a man is brushing a cat.
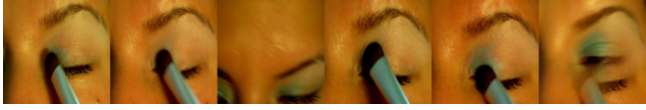
without: a woman is applying a eye.
with: a woman is applying makeup.
GT: a woman is putting makeup.

without: a man is singing and dancing.
with: a group of people are singing and dancing.
GT: group of people of singing and walking on streets.

without: a woman and a woman are talking.
with: a woman and a woman are talking to each other.
GT: a man and woman sitting in bed talking in a fireign language.

Figure 3. Comparison between models with or without *professional learning*. GT is short for ground truth. Errors or mistakes in captions are indicated by red color, such as "playing". The model trained by *professional learning* learns to describe video clips more accurately. It is able to make use of advanced words or phrases, indicated by green color, to generate sentences.

(MTVC) [27] is trained with unsupervised video prediction task to learn resilient video encoder representation and language entailment task to produce logic-enhanced annotation decoder feature. CIDEnt-RL[28] is trained by reinforcement method using mixed-loss methods and entailment-enhanced reward which outperforms CIDEr-reward models. HATT [41] utilizes temporal features, motion features, audio information and semantic information by hierarchical attention-based fusion to generate captions for videos. Efficient Convolutional Network (ECN) [47] is an 3D convolutional network for video action classification task and meaningful video features outputed by it are fed into SCN to produce annotations for videos. In GRU-EVE [1], Hierarchical Short Fourier Transform is applied to frame-level features in order to derive high-quality temporal dynamics and rich high-level semantic information is obtained from an object detection model. MARN [29] employs a memory block to store all the related visual and contextual information over the training set for each word. SAM-SS [8] is trained by scheduled sampling method with the assistance of semantics.

The experiment results displayed in Table 4 show that our model outperforms all the other methods on all the metrics with a large margin. Our model VNS-GRU^r achieves gains over the previously best model SAM-SS by 7.6% on BLEU-4, by 18.0% on CIDEr, by 11.4% on METEOR and by 3.8% on ROUGE-L.

Table 5. The results of the models on MSR-VTT. ⋆ denotes that the model is from MSR-VTT Challenge 2017. VNS-GRU is trained by the features shown in (1). Subscript r denotes feature $\boldsymbol{v}_{img}$ is used in training. Superscript e and r denote semantic feature $\boldsymbol{cls}_{vid}$ and $\boldsymbol{cls}_{img}$ are used in training respectively (2).

Models	B4	C	M	R	Overall
v2t_navigator⋆	40.8	44.8	28.2	60.9	0.911
Aalto⋆	39.8	45.7	26.9	59.8	0.894
VideoLAB⋆	39.1	44.1	27.7	60.6	0.893
CIDEnt-RL[28]	40.5	51.7	28.4	61.4	0.945
HACA[37]	43.4	49.7	29.5	61.8	0.963
HATT[41]	41.2	44.7	28.5	60.7	0.914
GRU-EVE[1]	38.3	48.1	28.4	60.7	0.914
MARN[29]	40.4	47.1	28.1	60.7	0.918
TAMoE[38]	42.2	48.9	29.4	62.0	0.952
SAM-SS[8]	43.8	51.4	28.9	62.4	0.970
VNS-GRU	45.3	50.3	29.2	62.6	0.977
VNS-GRU$_r$	45.4	51.7	29.3	62.9	0.986
VNS-GRU$_r^e$	**45.6**	51.8	29.7	63.2	0.992
VNS-GRU$_r^{er}$	45.3	**53.0**	**29.9**	**63.4**	**0.998**

4.5.2 Comparison on MSR-VTT

The first three models in Table 5 are the top-3 winners in MSR-VTT 2017 competition. HACA [37] is a model with multi-level aligned multi-modal attention framework. TAMoE [38] learns to embed multiple topic-based experts into the model and implicitly transfers knowledge in seen activities to unknown ones.

Our model VNS-GRU$_r^{er}$ also outperforms all the previous models on four metrics: BLEU-4, CIDEr, METEOR and ROUGE-L, and achieves gains over the closest rival SAM-SS by 3.4%, 3.1%, 3.5% and 1.6%, respectively. Note that our model surpasses CIDEnt-RL, which is directly optimized on CIDEr-related reward, on CIDEr.

5 Conclusion

In this work, we propose three methods to improve the decoder of video captioning model. The first is to embed variational dropout and layer normalization in RNN unit to prevent overfitting and sustain convergence speed. The second is an online method to select the best model for testing with comprehensive consideration on kinds of metrics. The last is a novel training scheme called *professional learning*. In its first phase, all the training annotations for each video is equally treated in the process of optimization. In its second phase, the training algorithm aims to strengthen the strong points of the model, in other words, the samples with lower loss values have higher weights in optimization. Our model achieves state-of-the-art results on the popular video captioning benchmarks with a rich vocabulary and diversified sentences. However, in theory, *profession learning* can be applied with other training algorithms together and it may further improve the performance of a model which we leave it for future research.

ACKNOWLEDGEMENTS

REFERENCES

[1] Nayyer Aafaq, Naveed Akhtar, Wei Liu, Syed Zulqarnain Gilani, and Ajmal Mian, 'Spatio-temporal dynamics and semantic attribute enriched visual encoding for video captioning', in *CVPR*, pp. 12487–12496, (2019).

[2] Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton, 'Layer normalization', *ArXiv*, **abs/1607.06450**, (2016).

[3] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu, 'Multiple object recognition with visual attention', in *ICLR*, (2015).

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, 'Neural machine translation by jointly learning to align and translate', in *ICLR*, (2015).

[5] Satanjeev Banerjee and Alon Lavie, 'METEOR: an automatic metric for MT evaluation with improved correlation with human judgments', in *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL*, pp. 65–72, (2005).

[6] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer, 'Scheduled sampling for sequence prediction with recurrent neural networks', in *NeurIPS*, pp. 1171–1179, (2015).

[7] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston, 'Curriculum learning', in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 41–48, New York, NY, USA, (2009). ACM.

[8] Haoran Chen, Ke Lin, Alexander Maye, Jianming Li, and Xiaolin Hu, 'A Semantics-Assisted Video Captioning Model Trained with Scheduled Sampling', *arXiv e-prints*, arXiv:1909.00121, (Aug 2019).

[9] Kyunghyun Cho, Bart Van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio, 'On the properties of neural machine translation: Encoder-decoder approaches', *arXiv: Computation and Language*, (2014).

[10] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, 'Learning phrase representations using RNN encoder-decoder for statistical machine translation', in *EMNLP*, pp. 1724–1734, (2014).

[11] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio, 'Empirical evaluation of gated recurrent neural networks on sequence modeling', *arXiv: Neural and Evolutionary Computing*, (2014).

[12] Yarin Gal and Zoubin Ghahramani, 'Dropout as a bayesian approximation: Representing model uncertainty in deep learning', in *ICML*, pp. 1050–1059, (2016).

[13] Yarin Gal and Zoubin Ghahramani, 'A theoretically grounded application of dropout in recurrent neural networks', in *NeurIPS*, pp. 1019–1027, (2016).

[14] Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng, 'Semantic compositional networks for visual captioning', in *CVPR*, pp. 1141–1150, (2017).

[15] Lianli Gao, Zhao Guo, Hanwang Zhang, Xing Xu, and Heng Tao Shen, 'Video captioning with attention-based LSTM and semantic consistency', *IEEE Trans. Multimedia*, **19**(9), 2045–2055, (2017).

[16] Anirudh Goyal, Alex Lamb, Ying Zhang, Saizheng Zhang, Aaron C. Courville, and Yoshua Bengio, 'Professor forcing: A new algorithm for training recurrent networks', in *NeurIPS*, pp. 4601–4609, (2016).

[17] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond J. Mooney, Trevor Darrell, and Kate Saenko, 'Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition', in *ICCV*, pp. 2712–2719, (2013).

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, 'Deep residual learning for image recognition', in *CVPR*, pp. 770–778, (2016).

[19] Sepp Hochreiter and Jürgen Schmidhuber, 'Long short-term memory', *Neural Computation*, **9**(8), 1735–1780, (1997).

[20] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R. Hershey, Tim K. Marks, and Kazuhiko Sumi, 'Attention-based multimodal fusion for video description', in *ICCV*, pp. 4203–4212, (2017).

[21] Ferenc Huszar, 'How (not) to train your generative model: Scheduled sampling, likelihood, adversary?', *CoRR*, **abs/1511.05101**, (2015).

[22] Sergey Ioffe and Christian Szegedy, 'Batch normalization: Accelerating deep network training by reducing internal covariate shift', in *ICML*, pp. 448–456, (2015).

[23] Chin-Yew Lin, 'ROUGE: A package for automatic evaluation of summaries', in *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, (July 2004). Association for Computational Linguistics.

[24] Chih-Yao Ma, Asim Kadav, Iain Melvin, Zsolt Kira, Ghassan AlRegib, and Hans Peter Graf, 'Attend and interact: Higher-order object interactions for video understanding', in *CVPR*, pp. 6790–6800, (2018).

[25] Taesup Moon, Heeyoul Choi, Hoshik Lee, and Inchul Song, 'RNNDROP: A novel dropout for RNNS in ASR', in *IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 65–70, (2015).

[26] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, 'Bleu: a method for automatic evaluation of machine translation', in *ACL*, pp. 311–318, (2002).

[27] Ramakanth Pasunuru and Mohit Bansal, 'Multi-task video captioning with video and entailment generation', in *ACL*, pp. 1273–1283, (2017).

[28] Ramakanth Pasunuru and Mohit Bansal, 'Reinforced video captioning with entailment rewards', in *EMNLP*, pp. 979–985, (2017).

[29] Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai, 'Memory-attended recurrent network for video captioning', in *CVPR*, pp. 8347–8356, (2019).

[30] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel, 'Self-critical sequence training for image captioning', in *CVPR*, pp. 1179–1195, (2017).

[31] Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth, 'Recurrent dropout without memory loss', in *COLING*, pp. 1757–1766, (2016).

[32] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, 'Dropout: a simple way to prevent neural networks from overfitting', *J. Mach. Learn. Res.*, **15**(1), 1929–1958, (2014).

[33] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, 'Going deeper with convolutions', in *CVPR*, pp. 1–9, (2015).

[34] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh, 'Cider: Consensus-based image description evaluation', in *CVPR*, pp. 4566–4575, (2015).

[35] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko, 'Translating videos to natural language using deep recurrent neural networks', in *NAACL*, pp. 1494–1504, Denver, Colorado, (May–June 2015). Association for Computational Linguistics.

[36] Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang, 'Video captioning via hierarchical reinforcement learning', in *CVPR*, pp. 4213–4222, (2018).

[37] Xin Wang, Yuan-Fang Wang, and William Yang Wang, 'Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning', in *NAACL-HLT*, pp. 795–801, (2018).

[38] Xin Wang, Jiawei Wu, Da Zhang, Yu Su, and William Yang Wang, 'Learning to compose topic-aware mixture of experts for zero-shot video captioning', in *AAAI*, pp. 8965–8972, (2019).

[39] Ronald J. Williams and David Zipser, 'A learning algorithm for continually running fully recurrent neural networks', *Neural Computation*, **1**(2), 270–280, (1989).

[40] Chen Wu, Xuancheng Ren, Fuli Luo, and Xu Sun, 'A hierarchical reinforced sequence operation method for unsupervised text style transfer', in *ACL*, pp. 4873–4883, (2019).

[41] Chunlei Wu, Yiwei Wei, Xiaoliang Chu, Weichen Sun, Fei Su, and Leiquan Wang, 'Hierarchical attention-based multimodal fusion for video captioning', *Neurocomputing*, **315**, 362–370, (2018).

[42] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, 'Aggregated residual transformations for deep neural networks', in *CVPR*, pp. 5987–5995, (2017).

[43] Jun Xu, Tao Mei, Ting Yao, and Yong Rui, 'MSR-VTT: A large video description dataset for bridging video and language', in *CVPR*, pp. 5288–5296, (2016).

[44] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio, 'Show, attend and tell: Neural image caption generation with visual attention', in *ICML*, pp. 2048–2057, (2015).

[45] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher J. Pal, Hugo Larochelle, and Aaron C. Courville, 'Describing videos by exploiting temporal structure', in *ICCV*, pp. 4507–4515, (2015).

[46] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals, 'Recurrent neural network regularization', *CoRR*, **abs/1409.2329**, (2014).

[47] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox, 'ECO: efficient convolutional network for online video understanding', in *ECCV*, pp. 713–730, (2018).