Video Interactive Captioning with Human Prompts

Aming Wu[1] , **Yahong Han**[1] and **Yi Yang**[2,3*]
[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]School of Computer Science, University of Technology Sydney, Australia
[3]Baidu Research
{tjwam, yahong}@tju.edu.cn, yi.yang@uts.edu.au

Abstract

Video captioning aims at generating a proper sentence to describe the video content. As a video often includes rich visual content and semantic details, different people may be interested in different views. Thus the generated sentence always fails to meet the ad hoc expectations. In this paper, we make a new attempt that, we launch a round of interaction between a human and a captioning agent. After generating an initial caption, the agent asks for a short prompt from the human as a clue of his expectation. Then, based on the prompt, the agent could generate a more accurate caption. We name this process a new task of video interactive captioning (ViCap). Taking a video and an initial caption as input, we devise the ViCap agent which consists of a video encoder, an initial caption encoder, and a refined caption generator. We show that the ViCap can be trained via a full supervision (with ground-truth) way or a weak supervision (with only prompts) way. For the evaluation of ViCap, we first extend the MSRVTT with interaction ground-truth. Experimental results not only show the prompts can help generate more accurate captions, but also demonstrate the good performance of the proposed method.

1 Introduction

Video captioning aims at automatically generating a natural language sentence to describe a video accurately. As a video clip often includes rich visual content and semantic details, different people may be interested in different views. Thus, one sentence generated by a video captioning model usually fails to meet the ad hoc expectations. Inspired by the relevance feedback in information retrieval [Rocchio, 1971], we propose to launch a round of interaction between a human and a captioning agent, so as to refine the initial captions.

In this paper, we present a new task of Video interactive Captioning (ViCap): When a caption generated by a pre-trained video captioning model does not meet the expectation, the ViCap agent asks for a short prompt from the human as a clue of his expectation. Then based on the prompt, the agent could generate a more accurate caption. In Fig. 1, we show an example. Firstly, given as input a video clip to a pre-trained video captioning model, an initial caption is generated, i.e., 'Someone is showing cartoon clip from tv'. As the human may be interested in certain frames of the clip, the description fails to meet his expectation. Then he gives a short prompt, i.e., 'A squidward'. Finally, based on the prompt, the ViCap agent generates a more accurate caption 'A squidward is talking to the spongebob at the crusty crab'. From a practical view, a human could provide any prompts in any form, e.g, in different length. And the prompts could appear anywhere in the generated captions. In this paper, we move forward a first step in that we set the prompts to be the first two words of the generating caption. Thus ViCap with human prompts is different from traditional captioning tasks, which may support various applications, such as video

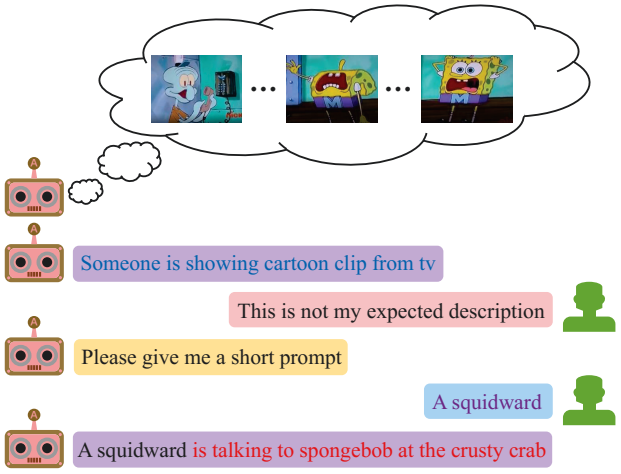

Figure 1: A human and a ViCap agent are interacting. First, the agent generates an initial description of a video clip (the blue words). As the human may be interested in certain frames of the clip, the description fails to meet his expectation. Then he gives a short prompt, i.e., the first two words of the expecting caption (the purple words). Finally, based on the prompt, the agent generates a more accurate caption (the red words).

*Part of this work was done when Yi Yang was visiting Baidu Research during his Professional Experience Program. Code is publicly available on GitHub: https://github.com/ViCap01/ViCap.

retrieval [Kordopatis-Zilos *et al.*, 2018], video chat robot, and even human-robot interaction [Pasunuru and Bansal, 2018].

The challenges of this task are two-fold: First, given as input the video clip, the initial caption, and the human prompts, the ViCap model should be able to generate a sentence not only starts with the prompts but also faithfully describes the video content. Second, as there launches a round of interaction, the prompts and the expecting captions are certainly semantically different from the initial ones, which therefore lack the ability of guiding the generation of the refined sentences. In this paper, we devise the ViCap to include a video encoder, an initial caption encoder, and a refined caption generator. Moreover, we show that the ViCap can be trained via a full supervision way, i.e., supervised with the ground-truth captions, or a weak supervision way, i.e., weakly supervised with only the prompts.

In particular, we first utilize a GRU network [Cho *et al.*, 2014] to encode the video and the initial caption, respectively. For the caption generator, the commonly used methods are based on LSTM [Li *et al.*, 2017] or GRU network. However, these generators are prone to dilute the long-term information [Gehring *et al.*, 2017]. Recent works [Gehring *et al.*, 2017] have demonstrated that employing convolution operation as the decoder could alleviate the problem of long-term information dilution. Thus, in this paper, we stack multiple dilated convolutional layers [Yu and Koltun, 2015] followed by gated activation units [Oord *et al.*, 2016] as the refined caption generator, the goal of which is to capture dependencies among frame and word sequences. For the training of ViCap with weak supervision of prompts, we devise a convolutional reconstruction network to reconstruct the current input of decoders. During the training, the reconstruction loss and the prompt-supervision loss are merged together.

For the evaluation of ViCap with human prompts, we first extend the MSRVTT with interaction ground-truth. Then the experimental results not only show the prompts can help generate more accurate captions, but also demonstrate the good performance of the proposed method.

2 Related Work

Recent advances towards video captioning mainly follow the encoding-decoding framework. Besides efforts made on devising effective encoders and decoders [Baraldi *et al.*, 2017; Wu and Han, 2018], Wang et al. [Wang *et al.*, 2016] proposed a new multimodal memory encoder and the method in [Pan *et al.*, 2017] utilized the transferred semantic attributes to help models generate better captions. Towards efficiency and a more compact representation of videos, Chen et al. [Chen *et al.*, 2018] proposed a reinforcement learning-based method to pick some keyframes to generate video caption. Besides, as the current most methods about video captioning suffer from the problems of exposure bias and inconsistency between train/test measurement [Keneshloo *et al.*, 2018], these works [Wang *et al.*, 2018; Phan *et al.*, 2017] proposed to leverage reinforcement learning to solve these problems.

As a video often includes rich visual content and semantic details, e.g., diverse objects and events, the existing captioning methods are hard to generate one sentence to fully describe these content and details. In this paper, we propose a new task of ViCap, i.e., when the initial caption does not meet human expectation, models could generate refined caption based on human prompts. As the existing captioning methods often employ a GRU or an LSTM unit as the decoder and only take a video clip as the input to generate a sentence, we could not directly employ GRU or LSTM decoder to solve the task. The reason is that the new task needs the decoder could leverage the video content, the initial caption, and the prompts to generate a more accurate caption. To solve this task, we devise a convolutional decoder, which uses the initial caption and the prompt to generate accurate caption. Finally, the experimental results on our extended MSRVTT dataset demonstrate the effectiveness of the convolutional decoder.

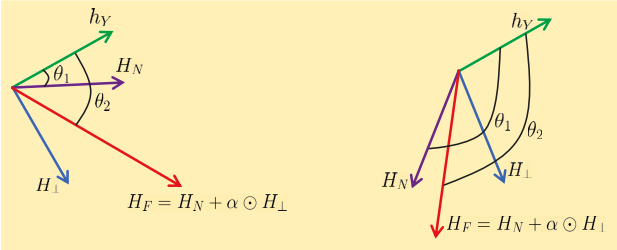

Figure 2: Illustration of enlarging the semantic gap. Here we use H_N to indicate the representation (the purple vector) generated by a decoder and use H_F to indicate the final representation (the red vector). h_Y indicates the initial caption representation. α is a scalar. The blue vector represents $H_\perp$. We can see that by adding $H_\perp$ to H_N, the angle between H_F and $H_\perp$ becomes smaller, which indicates the semantic gap between H_F and h_Y is enlarged.

3 The Framework of ViCap

In this paper, we devise the ViCap model which includes a video encoder, an initial caption encoder, and a refined caption generator. Moreover, we show that the ViCap can be trained via a full supervision way or a prompt supervision (weak supervision) way.

Concretely, we employ a GRU unit to encode the video and the initial caption, respectively. For the video, we use $H_X = \{h_X^1 \in \mathbb{R}^k, ..., h_X^m \in \mathbb{R}^k\}$ to denote the GRU output set and use H_X^{mean} to indicate the mean visual feature of H_X. For the initial caption, we take the GRU output $h_Y \in \mathbb{R}^o$ at the last time step as the representation of the initial caption.

3.1 The Convolutional Decoder of ViCap Model

As said in the Introduction, the new task aims at generating an accurate sentence which is semantically different from the initial caption. Thus, in order to enlarge the semantic gap, we propose to use the vertical representation $H_\perp$ of the initial caption representation h_Y in the decoder ($H_\perp \cdot h_Y = 0$, where $\cdot$ indicates the dot product), which indicates $H_\perp$ and h_Y are irrelevant. In Fig. 2, we show two examples of enlarging the semantic gap. The left part is the case where the output H_N of a decoder is positively related to h_Y. And the right part is the negatively correlated case. We can see that by adding $H_\perp$ to H_N ($H_F = H_N + \alpha \odot H_\perp$, where $\odot$ indicates the element

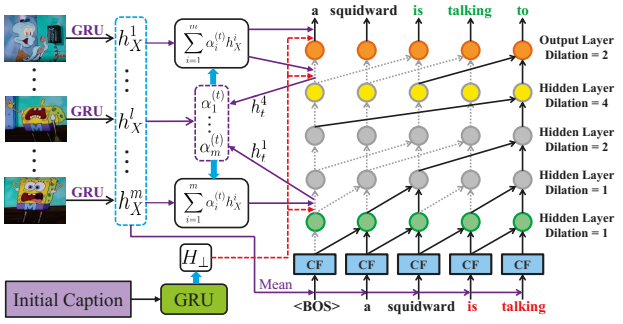

Figure 3: Illustration of our convolutional sequential decoder. This decoder consists of five dilated convolutional layers. The dilated rate is respectively set to 1, 1, 2, 4, 2. 'CF' represents the concatenation fusion. The green words are the ones generated by the decoder. By the hierarchical convolutional architecture, the decoder could capture variant dependencies among the sequence.

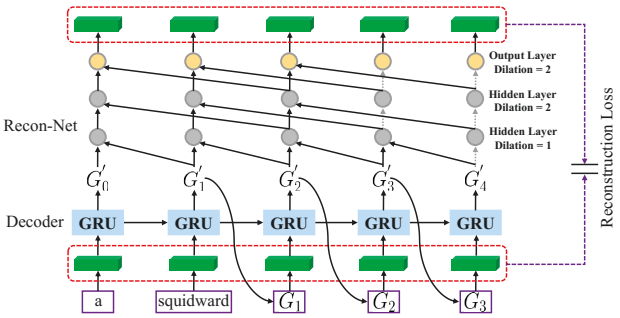

Figure 4: An example of convolutional reconstruction network. Here, we take GRU decoder as an example. And we stack three dilated convolutional layers as the reconstruction network. The dilated rate is respectively set to 1, 2, 2.

product), the semantic gap between H_F and h_Y is obviously enlarged, i.e., compared with the angle between H_N and $H_\perp$, the angle between H_F and $H_\perp$ becomes smaller. Based on this idea, we devise a convolutional refined decoder.

Computation of $H_\perp$

We define $h_Y = [y_1, y_2, ..., y_o]$, where y_1, y_2, and y_o are elements of the vector h_Y. When the dimension o is set to an even number, the computation of $H_\perp$ is shown as follows:

$$H_\perp = [-y_o, ..., -y_{\frac{o}{2}}, y_{\frac{o}{2}-1}, ..., y_1] \tag{1}$$

The goal of this operation is to keep $H_\perp \cdot h_Y = 0$.

Convolutional Decoder

In this paper, we stack five dilated convolutional layers to form convolutional sequential decoder (Fig. 3). In the following, we denote by $\hat{Y}^C = \{\hat{Y}_0^C, ..., \hat{Y}_{L-1}^C\}$ the predicted word sequence. We denote by $Y^C = \{Y_0^C, ..., Y_{L-1}^C\}$ the target word sequence, where L denotes sequence length. $[a, b]$ represents the concatenation of a and b.

As shown in Fig. 3, at each step t, the operations of each layer are shown as follows:

$$\begin{aligned} H_t^l &= [h_t^{l-1}, h_{t-rl}^{l-1}] \\ h_t^l &= tanh(w_f^l * H_t^l + b_f^l) \odot \sigma(w_g^l * H_t^l + b_g^l) \\ &\quad + (1.0 - \sigma(w_g^l * H_t^l + b_g^l)) \odot H_\perp \end{aligned} \tag{2}$$

where rl represents dilated rate of the l-th layer. h_t^{l-1} denotes the output of the $(l-1)$-th layer at time step t. w_f^l and w_g^l denote convolutional filters on the l-th layer. b_f^l and b_g^l are bias. Adding $H_\perp$ is to enlarge the semantic gap.

Note that for video caption generation, there is no future information available for the decoder. Besides, based on the different size of both filter and dilated rate, we use different number of zero vectors to pad the input of each layer.

Finally, the t-th generated word $\hat{Y}_t^C$ is computed as:

$$\begin{aligned} \alpha &= ReLU(\frac{h_t^5 \cdot h_Y}{\|h_t^5\| \|h_Y\|}) \\ o_t &= h_t^5 + \alpha \odot H_\perp \\ \hat{Y}_t^C &\sim softmax(w_p o_t + b_p) \end{aligned} \tag{3}$$

where h_t^5 indicates the output of the fifth layer. w_p and b_p are learnable parameters. The goal of $ReLU$ operation is to keep h_Y and h_t^5 are positively related.

3.2 ViCap with Full Supervision

In order to reduce the risk of vanishing gradients, inspired by the work [Zhang *et al.*, 2016], we enforce intermediate supervision for some hidden layers. In our network, since the function of the first layer and the fifth layer is to process the input and generate the output, respectively, we enforce supervision for these two layers. For each layer $j \in \{1, 5\}$, we employ a cross-entropy loss.

$$L_{CE}^j = -\sum_{t=0}^{L-1} log(p(\hat{Y}_t^C | Y_{0:t-1}^C, H_X, h_Y)) \tag{4}$$

where $p(\hat{Y}_t^C | Y_{0:t-1}^C, H_X, h_Y)$ is the output probability of the predicted word $\hat{Y}_t^C$ given the previous word $Y_{0:t-1}^C$, visual feature H_X, and the encoding h_Y of the initial caption. By summing the loss of each layer, we obtain the overall loss:

$$L_{CE} = \lambda_1 L_{CE}^1 + \lambda_5 L_{CE}^5 \tag{5}$$

where λ_1 and λ_5 are hyper-parameters. And we set $\lambda_1 + \lambda_5 = 1$. Meanwhile, as the fifth layer is the output layer, we should keep λ_5 is bigger than λ_1.

As said in the Introduction, we should keep the generating captions could faithfully describe their corresponding video content. Thus, we define a new loss function to ensure the consistency between the caption and video content, which is

named consistency loss (L_s). The details are as:

$$C_s = \frac{1}{L}\sum_{i=0}^{L-1} o_t, V_s = \frac{1}{m}\sum_{i=1}^{m} X_i$$
$$L_s = \|w_r C_s - V_s\| \tag{6}$$

where $\|\cdot\|$ indicates ℓ_2-norm. w_r is a learnable parameter.

Finally, the training loss L_{train} is computed as follows:

$$L_{train} = L_{CE} + L_s \tag{7}$$

3.3 ViCap with Prompt Supervision

For the case of prompt supervision, since we have no other word as a supervision in addition to the prompt words, we could not completely use the teacher forcing mechanism [Lamb *et al.*, 2016] to train the network. If we only use the prompt-supervision loss to train the network, as there only exist two prompt words, it is easy to make the network overfitting. And the performance is not good. In order to reduce the risk of overfitting and improve the performance, we propose a recurrent reconstruction network (Recon-Net) which uses the output of the decoder to reconstruct the current input of the decoder (Fig. 4). Finally, the reconstruction loss will be jointly trained with the prompt-supervision loss.

Convolutional Reconstruction Network

As shown in Fig. 4, we stack three dilated convolutional layers as the reconstruction network. At time step t, we take the output o_{L-t} of the decoder and the mean visual feature H_X^{mean} as the input of the first layer. The details of this layer are as:

$$M_t^r = w_c^r[H_X^{mean}, o_{L-t}] + b_c^r, \quad H_t^1 = [M_{t-1}^r, M_t^r]$$
$$R_t^1 = tanh(w_p^1 * H_t^1 + b_p^1) \odot \sigma(w_q^1 * H_t^1 + b_q^1) \tag{8}$$

where w_c^r and b_c^r are learnable parameters. w_p^1 and w_q^1 are convolutional filters. b_p^1 and b_q^1 are bias. Then the operations of the next two layers are similar to those of the first layer. As same as the convolutional decoder, we enforce intermediate supervision for the first and output layer to reduce the risk of vanishing gradients. Thus, the loss of the convolutional reconstruction network is as:

$$L_C^t = \beta_1\|R_t^3 - G_t\| + \beta_2\|R_t^1 - G_t\| \tag{9}$$

where G_t indicates the input of the decoder at time step t. β_1 and β_2 are hyper-parameters. And we set $\beta_1 + \beta_2 = 1$. Besides, as R_t^3 is the output of top layer, we should keep β_1 is bigger than β_2. For the prompt-supervision loss, we still use cross-entropy loss. Finally, the training loss is as:

$$L_{CNN} = L_P + \lambda\sum L_C^t \tag{10}$$

where λ is a hyper-parameter. L_P indicates the prompt-supervision loss.

Such a reconstruction network, which is similar to the regularized auto-encoder [Bengio *et al.*, 2012], promotes our caption decoders to learn effective word-level representation.

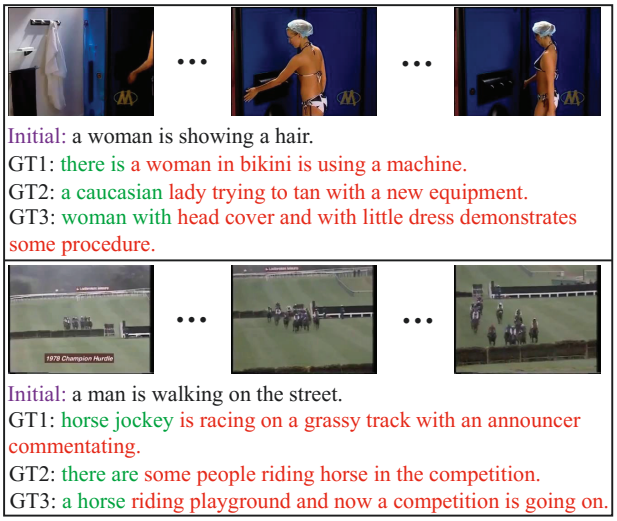

Figure 5: Examples of the extended MSRVTT-2016 dataset. The two green words indicate the prompts. The red words indicate the ground-truth of evaluation. 'Initial' indicates the initial caption. 'GT1', 'GT2', and 'GT3' indicate the three annotations.

Here, the caption decoder is equivalent to the encoder. And the reconstruction network is equivalent to the decoder. The reconstruction loss is helpful for regularizing the caption decoder during training, which reduces the risk of overfitting.

Besides, it is worth noting that the direction of the reconstruction network is opposite to that of the caption decoder. The goal of this processing is to promote the output representation generated by the caption decoder could embrace much information of both video content and the generating caption, which will be further demonstrated and discussed in the following experiment.

4 Experiments

In this section, we perform extensive experiments to evaluate the proposed methods. All results are evaluated by metrics of BLEU [Papineni *et al.*, 2002], METEOR [Denkowski and Lavie, 2014], and CIDEr [Vedantam *et al.*, 2015].

4.1 Extended Dataset and Implementation Details

MSRVTT-2016 [Xu *et al.*, 2016] is the recently released largest dataset for video captioning. The dataset contains 10,000 web video clips and each clip is annotated with approximately 20 natural language sentences.

In order to demonstrate this attempt and evaluate the proposed methods, we extend this dataset. And we take the 1st to 4500th clip as the training set of the pre-trained model and use the 4501st to 5000th clip as the validation set. For ViCap models, we take the 5001st to 8500th clip as the training set. And we take the 8501st to 9000th clip as the validation set and use 9001st to 10000th clip as the test set. Besides, for the 20 annotations of the training set of ViCap models, we remove duplicate annotations, semantically similar annotations, and short annotations. For each clip of the validation and test set of ViCap models, we give three different semantic sentences as the annotations. Finally, when we evaluate

Figure 6: Examples of ViCap. 'Initial' indicates the initial caption generated by the S2VT model. 'Final' indicates the refined caption generated by the convolutional decoder. 'GT' indicates the ground-truth. The red words indicate the final generated words.

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
S2VT+IGRU-D	22.24	11.56	6.76	3.86	8.96	22.26	38.50
S2VT+CNN-D	23.23	11.98	7.10	4.42	9.24	24.47	46.04
HRNE+IGRU-D	22.10	11.45	6.52	3.80	8.95	22.10	38.18
HRNE+CNN-D	23.17	12.26	7.08	4.29	9.33	24.03	45.06

Table 1: Performance of full supervision. Here we use GoogLeNet feature. 'S2VT+CNN-D' represents we use the S2VT model as the pre-trained model and use convolutional decoder as the final caption generator. All values are measured by percentage (%).

the performance of ViCap models, for each one of the annotations in the test set, we remove the first two words and take the remaining words as the ground-truth of evaluation. In Fig. 5, we show two examples which are from the test set. And the red words are the ground-truth of evaluation.

In this paper, we choose S2VT [Venugopalan *et al.*, 2015] and HRNE [Pan *et al.*, 2016] as examples of the pre-trained models to generate initial captions. Of course, we can also use other caption models as the pre-trained models. Finally, we take the model with the best performance on the validation set (4501st to 5000th video clips) as the initial caption generator. As two contributions of this paper are refined convolutional decoder (CNN-D) and convolutional reconstruction network (CNN-R), in order to compare fairly, we design an improved GRU decoder (IGRU-D) and an GRU reconstruction network (GRU-R).

In the following experiments, we select 20 equally-spaced frames from each video and feed them into GoogLeNet [Szegedy *et al.*,] to extract a 1,024-dimensional frame-wise representation. For the encoding network of both video and initial caption, the number of output channel is all set to 512. For CNN-D, the number of output channel of each layer is respectively set to 512, 256, 256, 512, and 512. For CNN-R, the number of output channel of each layer is set to 512, 256, and 512. For IGRU-D and GRU-R, the number of output channel is set to 512. Finally, during training, we use Adam optimizer with an initial learning rate of 1×10^{-3}. λ_1 and λ_5 are respectively set to 0.4 and 0.6. β_1, β_2, and λ are respectively set to 0.6, 0.4, and 0.001. Note that we do not conduct beam search in testing.

The details of IGRU-D are shown as follows:

$$
\begin{aligned}
r_t &= \sigma(U_r Y_G^{t-1} + V_r m_{t-1} + A_r \varphi_t(H_X) + b_r) \\
z_t &= \sigma(U_z Y_G^{t-1} + V_z m_{t-1} + A_z \varphi_t(H_X) + b_z) \\
\bar{h}_t &= \phi(U_{\bar{h}} Y_G^{t-1} + V_{\bar{h}}(r_t \odot m_{t-1} + (1 - r_t) \odot H_\perp) \\
&\quad + A_{\bar{h}} \varphi_t(H_X) + b_{\bar{h}}) \\
m_t &= (1 - z_t) \odot m_{t-1} + z_t \odot \bar{h}_t \\
\alpha &= ReLU(\frac{m_t \cdot h_Y}{\|m_t\| \|h_Y\|}), \quad o_t = m_t + \alpha \odot H_\perp
\end{aligned}
\tag{11}
$$

where Y_G^{t-1} represents the word embedding result of the ground-truth word at time step $t-1$. $\varphi_t(H_X)$ represents the visual attention [Yao *et al.*, 2015]. The goal of $ReLU$ operation is to keep h_Y and m_t are positively related.

The details of GRU-R are shown as follows:

$$
\begin{aligned}
r_t &= \sigma(U_{rr} o_{L-t} + V_{rr} R_{t-1} + A_{rr} H_X^{mean}) \\
z_t &= \sigma(U_{rz} o_{L-t} + V_{rz} R_{t-1} + A_{rr} H_X^{mean}) \\
\bar{h}_t &= \phi(U_{r\bar{h}} o_{L-t} + V_{r\bar{h}}(r_t \odot R_{t-1}) + A_{rr} H_X^{mean}) \\
R_t &= (1 - z_t) \odot R_{t-1} + z_t \odot \bar{h}_t
\end{aligned}
\tag{12}
$$

where o_{L-t} represents the output of the decoder at time step $L-t$. L indicates the length of the generated caption. The reconstruction loss is defined as:

$$
L_R^t = \|R_t - G_t\| \tag{13}
$$

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
S2VT+IGRU-D+GRU-R	9.42	3.66	1.31	0.43	3.93	12.16	2.61
S2VT+IGRU-D+CNN-R	10.56	3.87	1.32	0.43	4.34	12.91	3.47
S2VT+CNN-D+GRU-R	12.45	4.37	1.36	0.46	4.73	13.23	4.43
S2VT+CNN-D+CNN-R	14.64	5.53	1.83	0.58	4.97	14.43	6.25
HRNE+IGRU-D+GRU-R	9.37	3.54	1.31	0.44	3.93	12.16	2.61
HRNE+IGRU-D+CNN-R	10.02	3.96	1.47	0.58	4.31	12.27	3.55
HRNE+CNN-D+GRU-R	12.14	4.66	1.33	0.46	4.72	14.57	5.15
HRNE+CNN-D+CNN-R	13.69	5.28	1.78	0.56	4.81	14.87	5.99

Table 2: Performance of prompt supervision. 'S2VT+CNN-D+CNN-R' indicates we use the S2VT model to generate initial captions and use the convolutional decoder and convolutional reconstruction network. All values are measured by percentage (%).

where G_t indicates the input of the decoder at time step t.

Method	BLEU-4	METEOR	CIDEr
GRU+$H_\perp$	3.86	8.96	38.50
GRU	3.10	8.16	33.35
GRU+h_Y	2.97	7.80	31.86
CNN+$H_\perp$	4.42	9.24	46.04
CNN	3.91	8.83	39.54
CNN+h_Y	3.76	8.49	40.13

Table 3: Ablation analysis of $H_\perp$. 'CNN+$H_\perp$' indicates we use the $H_\perp$ in the convolutional decoder. 'CNN' indicates we do not use $H_\perp$ and h_Y in the convolutional decoder.

Loss Type	BLEU-4	METEOR	CIDEr
out-layer	0.00	2.11	0.95
out-layer+consist	0.00	2.80	1.62
multi-layer	3.95	8.96	39.41
multi-layer+consist	4.42	9.24	46.04

Table 4: The effect of different types of convolutional decoder loss. 'out-layer' indicates the loss which is from the output layer. 'multi-layer+consist' indicates the loss L_{CE} and L_s.

4.2 Performance of Full Supervision

Table 1 shows the performance of full supervision. We can see that based on different pre-trained models, the performance of convolutional decoder outperforms that of the IGRU decoder. This shows that convolutional decoder is effective. In Fig. 6, we show some examples of ViCap. We can see that when the initial caption does not meet our expectation, based on the given prompts, our method indeed generate more accurate descriptions. These not only show the proposed scene is meaningful but also demonstrate the effectiveness of the proposed method.

Ablation Analysis of $H_\perp$

We first analyze $H_\perp$. In Table 3, we show the comparative results. We can see that adding $H_\perp$ in the decoder could improve the performance obviously.

The Effect of Different Types of Convolutional Loss

In order to analyze the effect of different types of loss, we make some comparative experiments. The results are shown in Table 4. Firstly, we can see that the performance of using the multi-layer loss outperforms that of only using the output layer loss obviously. This shows that using multi-layer loss really reduce the risk of vanishing gradients and improve the performance. Besides, for the multi-layer loss and the out-layer loss, we can see that adding the loss L_s improve the performance. This shows the loss L_s is effective.

4.3 Performance of Prompt Supervision

For the case of prompt supervision, we propose a GRU and convolution reconstruction network. Combined with the improved GRU decoder and convolutional decoder, there are four different architectures of prompt supervision. Table 2 shows their performance. We can see that the performance of the convolutional reconstruction network outperforms the GRU reconstruction network. This further demonstrates the effectiveness of convolutional architecture. Finally, the performance of the combination of the convolutional decoder and convolutional reconstruction network is the best.

The Effect of Reconstruction Loss

Here, we analyze the effect of the reconstruction loss. The results are shown in Table 5. We can see that for different decoder, adding reconstruction network could improve the performance obviously. For the generation of one natural language sentence, only using two ground-truth words (prompts) as the supervision information is not enough, which makes the network is prone to overfitting. Obviously, for our network, adding the reconstruction loss reduces the risk of overfitting and promotes the decoders to generate better word-level representation.

Method	BLEU-4	METEOR	CIDEr
IGRU-D+NO-R	0.11	2.31	1.72
IGRU-D+CNN-R	0.43	4.34	3.47
CNN-D+NO-R	0.33	4.17	2.79
CNN-D+CNN-R	0.58	4.97	6.25

Table 5: The effect of reconstruction loss. 'CNN-D+NO-R' indicates we only use the convolutional decoder.

5 Conclusion

In this paper, we propose a new task of ViCap. Meanwhile, we devise ViCap models. Finally, the experimental results not only demonstrate the effectiveness of our proposed methods but also show the new task is meaningful.

Acknowledgements

This work is supported by the NSFC (under Grant 61876130, U1509206).

References

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

[Bengio *et al.*, 2012] Yoshua Bengio, Aaron C Courville, and Pascal Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR, abs/1206.5538*, 1:2012, 2012.

[Chen *et al.*, 2018] Yangyu Chen, Shuhui Wang, Weigang Zhang, and Qingming Huang. Less is more: Picking informative frames for video captioning. *arXiv preprint arXiv:1803.01457*, 2018.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[Denkowski and Lavie, 2014] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380, 2014.

[Gehring *et al.*, 2017] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*, 2017.

[Keneshloo *et al.*, 2018] Yaser Keneshloo, Tian Shi, Chandan K Reddy, and Naren Ramakrishnan. Deep reinforcement learning for sequence to sequence models. *arXiv preprint arXiv:1805.09461*, 2018.

[Kordopatis-Zilos *et al.*, 2018] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. Fivr: Fine-grained incident video retrieval. *arXiv preprint arXiv:1809.04094*, 2018.

[Lamb *et al.*, 2016] Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *NIPS*, pages 4601–4609, 2016.

[Li *et al.*, 2017] Xuelong Li, Bin Zhao, and Xiaoqiang Lu. Mam-rnn: multi-level attention model based rnn for video captioning. In *IJCAI*, 2017.

[Oord *et al.*, 2016] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[Pan *et al.*, 2016] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*, pages 1029–1038, 2016.

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

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318. ACL, 2002.

[Pasunuru and Bansal, 2018] Ramakanth Pasunuru and Mohit Bansal. Game-based video-context dialogue. *arXiv preprint arXiv:1809.04560*, 2018.

[Phan *et al.*, 2017] Sang Phan, Gustav Eje Henter, Yusuke Miyao, and Shin'ichi Satoh. Consensus-based sequence training for video captioning. *arXiv preprint arXiv:1712.09532*, 2017.

[Ramakrishna *et al.*, 2015] Vedantam Ramakrishna, Parikh Devi, and C Lawrence Zitnick. Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575, 2015.

[Rocchio, 1971] Joseph John Rocchio. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*, pages 313–323, 1971.

[Szegedy *et al.*,] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, and Dragomir Anguelov. Going deeper with convolutions. In *CVPR*, pages 1–9.

[Venugopalan *et al.*, 2015] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *ICCV*, pages 4534–4542, 2015.

[Wang *et al.*, 2016] Junbo Wang, Wei Wang, Yan Huang, Liang Wang, and Tieniu Tan. Multimodal memory modelling for video captioning. *arXiv preprint arXiv:1611.05592*, 2016.

[Wang *et al.*, 2018] Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. Video captioning via hierarchical reinforcement learning. In *CVPR*, pages 4213–4222, 2018.

[Wu and Han, 2018] Aming Wu and Yahong Han. Multimodal circulant fusion for video-to-language and backward. In *IJCAI*, volume 3, page 8, 2018.

[Xu *et al.*, 2016] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, pages 5288–5296, 2016.

[Yao *et al.*, 2015] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *ICCV*, pages 4507–4515, 2015.

[Yu and Koltun, 2015] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[Zhang *et al.*, 2016] Yuting Zhang, Kibok Lee, and Honglak Lee. Augmenting supervised neural networks with unsupervised objectives for large-scale image classification. In *ICML*, pages 612–621, 2016.