Temporal Attention Feature Encoding for Video Captioning

Nayoung Kim†, Seong Jong Ha‡, and Je-Won Kang†
†Department of Electronic and Electrical Engineering, Ewha W. University, Seoul, Korea
‡Vision AI Lab, AI Center, NCSOFT, Korea
E-mail: †1210513skdud@ehwain.net, ‡seongjongha@ncsoft.com, †jewonk@ewha.ac.kr

***Abstract*—In this paper, we propose a novel video captioning algorithm including a feature encoder (FENC) and a decoder architecture to provide more accurate and richer representation. Our network model incorporates feature temporal attention (FTA) to efficiently embed important events to a feature vector. In FTA, the proposed feature is given as the weighted fusion of the video features extracted from 3D CNN, and, therefore it allows a decoder to know when the feature is activated. In a decoder, similarly, a feature word attention (FWA) is used for weighting some elements of the encoded feature vector. The FWA determines which elements in the feature should be activated to generate the appropriate word. The training is further facilitated by a new loss function, reducing the variance of the frequencies of words. It is demonstrated with experimental results that the proposed algorithms outperforms the conventional algorithms in VATEX that is a recent large-scale dataset for long-term video sentence generation.**

I. INTRODUCTION

Video captioning has gained significant attention from both research communities of computer vision and natural language processing. The task describing a video scene in a natural sentence is challenging because it involves two different process of scene analysis and sentence generation. The recent advance in deep learning sheds some light on the link of image/video understanding and translation. Convolutional Neural Network (CNN) is used for extracting useful image/video features, and the features are outputted in the form of sentences through the Recurrent Neural Network (RNN). The deep learning architecture is called an encoder-decoder structure [11].

Image captioning techniques have been extensively studied [4], [5], [6], [11]. In earlier studies, researchers tried to develop more efficient encoder-decoder structures, applied to the captioning task. Sutskever *et al.* [11] proposed to use a multi-layer long-short term memory (LSTM). As the LSTM is more robust to an over-fitting problem than the RNN, they could learn deeper multi-layers and generate longer sentences. Later, a visual attention model has been used for a CNN-based encoder because the generated sentence needs to reflect the areas in which humans are paying more attention in an image [4], [5], [6]. In [4], the attention model is applied to choosing semantic conceptual proposals. In [5], an adaptive attention encoder-decoder model is developed to automatically determine when to depend on visual contents in a region. In [6], the attention has been more accurate by precisely localizing the corresponding regions in an image.

In video captioning, the encoder consider temporal features as well as appearance features. In other words, instead of captioning a frame independently, an encoder uses correlation among successive frames to understand actions or temporal dynamics better. In the beginning research, an encoder extracts features by using average-pooling of all the video frames stacked in a temporal order [14]. In [12], they choose few key frames to show important remarks in a temporal feature. Those works are valid for short video clips when similar visual contents are repeatedly shown. However, they are difficult to apply to frames with dynamic scene changes. To overcome the drawbacks, Veugopalan *et al.* [15] proposed stacked video encoder and decoder, consisting of two LSTM modules. In [17], the authors showed that the 3D CNN networks which is pretrained by video action classification tasks works for video captioning tasks as well. Yao *et al.* [13] proposed an attention model in a decoder to select an appropriate word in some specific moment.

Video features extracted from 3D CNN [9] are widely used for video captioning networks. However, because the feature is encoded to contain both dynamic actions and stationary scenes, the conventional approach cannot fully consider the temporal dynamics in a video sequence. In other words, an encoder needs to inform a decoder when to focus in a video sequence. Therefore, in this paper, the proposed algorithm uses a feature encoder (FENC) to apply temporal attention to the 3D video feature to reflect more important moments, efficiently. In a feature decoder, the proposed algorithm applies word attention to let a decoder know where to focus in a feature. We also develop a new loss function to facilitate the learning.

The paper is organized as follows. We describe the proposed model in detail in Section II. We show experimental results in Section III and the concluding remarks in Section IV.

II. PROPOSED ALGORITHM

A. Overview of Proposed Algorithm

We define a descriptive sentence $U = [u_1, \ldots, u_{N_u}]$ as a sequence of word vectors. u_i is a word vector to indicate an index in a dictionary. The word-to-vector is defined in Glove [18], widely used for natural language processing. When a video sequence V is given, the goal is to learn a mapping function $\mathcal{M} : V \longrightarrow U$ in an end-to-end manner. To this aim, our deep learning model is built on an encoder-decoder structure based on 3D-CNN [9] and Long Short-Term Memory

(LSTM). The encoder transforms a video sequence into video feature vectors $[v_1, \ldots, v_{N_v}]$, and the decoder interprets the features to a sentence. Generally, given with a video feature v_t, u_i is sequentially given by maximizing the likelihood function : $p(u_i|v_t, u_{1:i-1}; \theta)$, where θ is the network parameters. The problem can be recursively solved with the previous word outputs $u_{1:i-1}$ and the video feature through the LSTM.

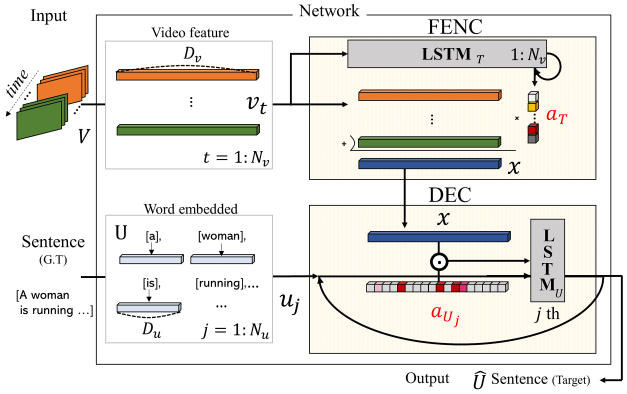

Fig. 1. Blockdiagram of the proposed network architecture.

Beside to the architectural similarity to the conventional works, the proposed architecture uses a feature encoder (FENC) including a feature temporal attention (FTA) module to enhance the learning. Our belief is that the proposed feature x that is encoded using both v_t and its associated temporal attention can help the captioning task, by letting the decoder to know when the feature is activated. In a decoder, similarly, a feature word attention (FWA) is used for weighting some elements of the encoded feature vector to activate an accurate word. The blockdiagram of the proposed encoder-decoder is shown in Fig.1.

B. Feature Encoder with Temporal Attention

In the proposed algorithm, the FENC is built on an LSTM module denoted by LSTM_T and the corresponding attention at each time stamp. The LSTM determines the next hidden states h^v_{t+1} and the output o^v_t when the current video feature v_t goes through the LSTM cells, given as

$$o^v_t, h^v_{t+1} = \text{LSTM}_T(v_t, h^v_t), \tag{1}$$

where $v_t \in \mathbb{R}^{D_v}$ is sequentially fed with t.

In FTA, a weight vector $a_T \in \mathbb{R}^{N_v}$ is trained to identify more important ingredients of a video feature in temporal domain. In the model, the attention is calculated as follows:

$$a_T = \sigma(W_T \cdot o_T + b_T), \tag{2}$$

where W_T is a projection matrix and b_T is an offset as learnable parameters in the encoder, and $o_T = [o_1, \ldots, o_{N_v}]^{\mathrm{T}}$ is all the outputs obtained after the end of the iteration of LSTM_T. $\cdot$ is the matrix multiplication.

Then, a feature vector x with the temporal attention is computed as,

$$x = \sum_{t=1}^{N_v} a_{T_t} v_t, \tag{3}$$

where x is given as the weighted fusion of all the coded video feature vectors with the temporal weights.

C. Feature Decoder with Word Attention

In a decoder side, another LSTM module denoted by LSTM_U is used for convert the coded feature vector into a sequence of a word as shown in Fig.1. In decoding, a feature word attention (FWA) is used for emphasizing a part of the feature vector to give a better chance to activate an appropriate word. In Fig.1, the weight vector a_{U_j} is applied to when the j-th word is generated. a_{U_j} is mathematically given as,

$$a_{U_j} = \sigma(W_U \cdot [h^u_j, x] + b_U), \tag{4}$$

where $W_U \in \mathbb{R}^{D_v \times 2D_v}$ and $b_U \in \mathbb{R}^{D_v}$ are the learnable matrix and offset, respectively. In (4), $[,]$ is an operation to concatenate h^u_j and x. In LSTM_U, the output gate o^u_{j+1} and the next hidden state h^u_{j+1} are evolved through the cells. The operations are as follows:

$$o^u_{j+1}, h^u_{j+1} = \text{LSTM}_U([a_{U_j} \odot x, u_j], h^u_j), \tag{5}$$

where $\odot$ is the element-wise multiplication. In this manner, a_{U_j} gives more weights to the j-th elements of x.

In final, an estimated word $\hat{u}_{j+1}$ in the next iteration is computed as

$$\hat{u}_{j+1} = \text{Softmax}(W_o \cdot o^u_{j+1}), \tag{6}$$

where W_o are learnable projection matrix from the domain of the feature to the domain of the word vector.

D. Training and Loss Function

We train our network to optimize a set of learning parameters $\theta = \{W, b\}$ to output a sentence $\hat{U}$ as close as the ground-truth U. For this, we define the cost function L considering the quality of generated sentences, given as

$$L = \lambda_1 L_{var} + \lambda_2 L_{prob} + \lambda_3 R(W), \tag{7}$$

where λ_1,λ_2 and λ_3 are the parameter for training, which are set to 0.01, 1, and 0.001, respectively.

Some conventional algorithms use beam-search [7] to determine the best word by allowing for multiple candidates of $\hat{u}_j$ in training. However, it is also shown in [8] that such the mechanism may give a wrong result and take more time to search the best candidate. Therefore, instead of using the beam search, we develop a new loss function L_{var} defined as:

$$L_{var} = \frac{1}{N_u} \sum_{j=1}^{N_u} (\hat{u}_j - \bar{u}_j)^2 \tag{8}$$

where $\bar{u}_j$ is a mean value, and N_u is the number of words in a sentence. As shown in (8), L_{var} reduces the variance to avoid confusion in activating the best candidate of the words.

We also use a regularization term $R(W)$ as $||W||_2$. The minimization of the L_2-norm reduces the number of non-zero elements in the projection matrix, and it helps avoid over-fitting. L_{prob} is a cross-entropy function defined as $E(-U log\hat{U})$.

In training, we use Adam Optimizer with a variable learning rate 10^{-3} to 10^{-5}. We set a batch size to 256 and an epoch to 100. We set N_v, N_u, D_u, and D_v to 32, 30, 300 and 512, respectively. We use Tensorflow and TITAN XP GPU to learn the proposed network.

III. Experimental results

A. Dataset

We use a VaTeX [20] that is a very recent large-scale videos with sufficient lengths of sentences for captioning. The average number of the words in a sentence is 15.23. The dataset contains 25,991 training data, 3,000 validation data, and 6,000 test data. Each video clip has 10 different captions. We calculate each frequency of words in all the training dataset, and mark "Unknown" tag if a word is shown only 20 times in the whole set.

B. Performance Evaluation

We conduct experiments to show the qualitative performance of the proposed algorithm using four diverse machine translation metrics: BLEU [21], Meteor [22], $\mathrm{ROUGE_L}$ [23] and CIDEr [24]. Those metrics are widely used for comparing the performance of video captioning algorithms. Specifically, MS-COCO code [19] is used in the evaluation. Because there are 10 different sequences in the dataset, we calculate an average score of the metrics.

TABLE I
QUANTITATIVE PERFORMANCE OF THE PROPOSED ALGORITHM AS COMPARED TO THE CONVENTIONAL RESEARCH, USING VATEX VALIDATION SET AND VARIOUS METRICS.

model	BLEU@4	METEOR	$\mathrm{ROUGE_L}$	CIDEr
SA[13]	12.17± 0.8	18.75± 0.8	32.12± 2.2	26.10± 4.7
CRF-prob[12]	11.98± 0.5	18.49± 0.3	31.18± 0.5	25.18± 1.5
CRF-max[12]	11.91± 1.3	18.28± 0.8	31.17± 1.6	27.12± 7.7
S2VT[15]	12.20± 1.6	19.68± 1.3	33.45± 2.3	37.12± 9.6
ours	15.20± 0.9	22.19± 1.2	37.93± 2.8	55.21± 8.5
ours - L_{var}	15.17± 0.8	21.08± 1.3	37.21± 2.7	54.88± 8.3
ours - FTA	14.10± 1.2	20.12± 1.0	35.77± 2.8	52.79±10.5
ours - FWA	13.90± 1.0	20.08± 1.5	36.66± 2.8	53.21± 7.7

Table I shows the quantitative results. As shown, the proposed algorithm denoted by "Ours" shows the best results around 15.20, 22.19, 37.93, and 55.21 on the average in BLEU, Meteor, $\mathrm{ROUGE_L}$, and CIDEr, respectively. Our result is improved around 7.02% as compared to S2VT as the 2nd best algorithm when averaging all the values. The results imply that the proposed algorithm embeds richer features with the temporal attention model. SA and CRF do not consider temporal correlation in video features. They provide a sum of frame-level features in encoders. As compared to SA and CRF, our model uses FTA efficiently to give more weights to important moment in a video. As a result, the proposed algorithm outperforms around 10.50% over SA and CRF.

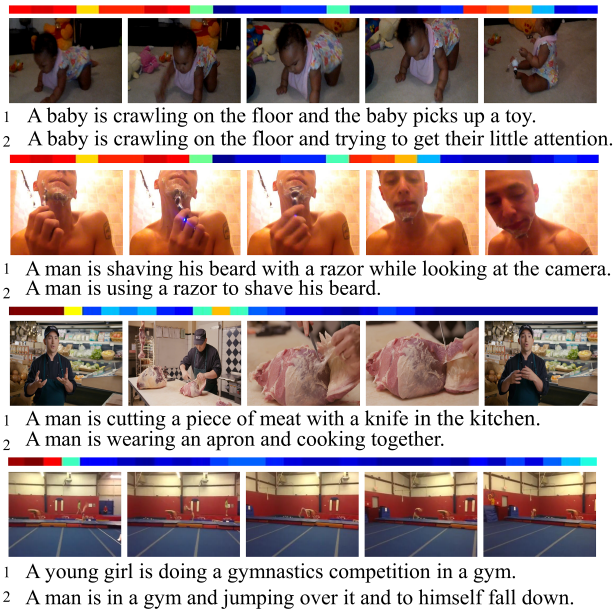

Fig. 2. The captioning results in VATEX dataset. The color bar means how much FTA is activated. Red colors mean higher values in attention and vice versa in blue colors. The first sentence in each video clip is generated from the proposed algorithm, and the second sentence is generated from the same model without FTA.

We also show qualitative results of the proposed algorithm in Fig. 2. As shown, the proposed algorithm shows accurate sentence as in the first line of each video clip. Some videos have abrupt scene changes more than one event in the sequence or unstable camera movements. For instance, the video clip in the third row shows a person talking about cooking in the beginning, but the scene is changed to the specimen. Nevertheless, the proposed algorithm outputs a descriptive sentence to appropriately explain the scenes. The words are naturally combined with "and" or "while" in the sentences.

C. Ablation Study

In this subsection, we conduct several ablation studies. First, we verify our proposed modules FTA, and FWA by removing the a_U in (4), a_T in (3) respectively. As shown in Table I, each result without FTA, FWA is decreased around 1.93% and 1.67. We also visualize the activation of FTA as color bar as shown is Fig.2. The color bar on each video frame is an indicator of how much each video frames are activated for generating captioning: red colors mean higher values in attention and vice versa in blue colors. Usually, FTA is highly activated on early video feature, and reflects intuitively temporal characteristics. For instance, the early frames of video clip in the first row are highly activated with caption: "a baby is crawling on the floor", and then little activated since the same information is repeated. The last few frames of video clip are re-activated with new caption:"a baby pick up a toy". As more interesting, sentences(second line) shown in Fig.2 without the temporal attention a_T make sense, but are not fully representative

overall video.

We also verify the effectiveness of the new loss term by removing L_{var} in (7). The result shows that the overall performance is slightly decreased compared to "our" base model. As a result, we are able to prove that our L_{var} has helped to reduce the confusion rate to produce correct answer.

Fig. 3. Case study: The limitation of video captioning.

However, our algorithm does not make an appropriate caption in some cases. As shown in Fig.3, it is difficult to distinguish between similar color backgrounds or object, *e.g.* 'sky' and 'water'. Additionally, our algorithm hardly outputs accurate captions when the video contains fast motions.

IV. Conclusions

In this paper, a novel video captioning algorithm including a feature encoder (FENC) and a decoder architecture was proposed. The proposed algorithm provided more accurate and richer representation in the sentence generation. Our network model incorporated feature temporal attention (FTA) to efficiently embed important events to a feature vector. In a decoder, similarly, a feature word attention (FWA) was used for weighting some elements of the encoded feature vector. It has been demonstrated with experimental results that the proposed algorithms provided significantly improved performance over the conventional algorithms in VaTeX [20] dataset.

Acknowledgment

This work has been supported by NCSOFT and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(No.NRF-2019R1C1C1010249).

References

[1] N. Krishnamoorthy, G. Malkarnenkar, R. Mooney, K. Saenko, and S. Guadarrama, "Generating natural-language video descriptions using text-mined knowledge," *Twenty-Seventh AAAI Conference on Artificial Intelligence,* 2013.

[2] S. Venugopalan,H. Xu,J. Donahue,M. Rohrbach,R. Mooney, and K. Saenko , "Translating videos to natural language using deep recurrent neural networks," *arXiv preprint arXiv:1412.4729,* 2014.

[3] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2013.

[4] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2016, pp.4651-4659

[5] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2017, pp. 375-383.

[6] M. Pedersoli, T. Lucas, C. Schmid, and J. Verbeek, "Areas of attention for image captioning," in *Proceedings of the IEEE International Conference on Computer Vision,* 2017, pp. 1242-1250.

[7] J.S. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2017, pp. 7008-7024.

[8] Z. Ren, X. Wang, N. Zhang, X. Lv, and J.L. Li, "Deep reinforcement learning-based image captioning with embedding reward," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2017, pp. 290-298.

[9] J. Carreira, and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2017, pp. 6299-6308.

[10] K. Cho, B. Van Merrinboer, C. Gulcehre, D. Bahdanau, F, Bougares, H. Schwenk, and Y.Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *arXiv preprint arXiv:1406.1078,* 2014.

[11] I. Sutskever, O. Vinyals, and Q.V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems,* 2014, pp. 3104-3112.

[12] J, Donahue,L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan,K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2015, pp. 2625-2634.

[13] L. Yao, A. Torabi,K. Cho, N, Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *Proceedings of the IEEE international conference on computer vision,* 2015, pp. 4507-4515.

[14] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," in *arXiv preprint arXiv:1412.4729,* 2014.

[15] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proceedings of the IEEE international conference on computer vision,* 2015, pp. 4534-4542.

[16] M. Sundermeyer, R. Schlter, and H. Ney, "LSTM neural networks for language modeling," in *Thirteenth annual conference of the international speech communication association,* 2012.

[17] R. Krishna, K. Hata, F. Ren, and L. Fei-Fei , "Dense-captioning events in videos," in *Proceedings of the IEEE international conference on computer vision,* 2017, pp. 706-715.

[18] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP),* 2014, pp. 1532-1543.

[19] X, Chen, H. Fang, T.Y. Lin, R. Vedantam, S. Gupta, P. Dollr, and C. L. Zitnick , "Microsoft coco captions: Data collection and evaluation server," in *arXiv preprint arXiv:1504.00325,* 2015.

[20] X. Wang, J. Wu, J. Chen, L. Li, Y. F. Wang, and W. Y. Wang,, "VATEX: A Large-Scale, High-Quality Multilingual Dataset for Video-and-Language Research," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2019.

[21] K. Papineni, S. Roukos, T. Ward, and W.J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics,* 2002, pp. 311-318.

[22] M. Denkowski, A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proceedings of the ninth workshop on statistical machine translation,* 2014, pp. 376-380.

[23] C. Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out,* 2004, pp. 74-81.

[24] R. Vedantam, C.Z. Lawrence and D. Parikh r, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2015, pp. 4566-4575.