MLVC_HDU@TRECVID 2022: Video to Text (VTT) and Activities in Extended Video (ActEV) Tasks

Ping Li, Tao Wang, Xingchao Ye
School of Computer Science, Hangzhou Dianzi University
Hangzhou, China, 310018
{lpcs, wangtao000213, yxc_}@hdu.edu.cn

Abstract

In this notebook paper, we present our solutions to the Video to Text (VTT) task and Activities in Extended Video (ActEV) task released in TRECVID 2022. For the VTT task, we propose a Semantic Alignment Network (SAN), which attempts to (1) establish a mapping relation between generated words and video frames by attention mechanism and then to (2) decode these video frames in predicting the next word. SAN learns to capture the most discriminative phrase of the partially decoded caption and also the mapping that aligns each phrase with the relevant frames. For the ActEV task, we adopt the Dynamic Interactive Aggregation Network (DIAN), which considers multiple interactive relationships (such as person-person, person-object, and temporal interaction) by dense serial connection, and dynamically updates memory features by iterative self-learning. The 2022 edition of the TRECVID benchmark has been a fruitful participation for the MLVC_HDU team. Our runs rank the second place on METEOR and SPICE, and the third place on BLEU in the VTT task. Meanwhile, our runs rank the third in the ActEV task.

1 Video to Text (VTT)

1.1 Introduction

The TRECVID Video to Text (VTT) is the task of understanding the scenes in a video and describing it in words, which is one of the most challenging computer vision tasks as it requires the model capable of associating video with text.

Recently, most video captioning methods adopt the encoder-decoder framework using convolutional neural networks (CNNs) and recurrent neural networks (RNNs) as the backbone. The CNN-based encoder takes a set of consecutive video frames and produces visual representations to generate the accurate caption that describes the video. Then, the RNN-based decoder takes the visually encoded features and the previously predicted word as input and generates one word at a time.

The current video frame is usually similar to the previous frame, and there exist redundancy among consecutive frames [2]. Therefore, someone claims that video is grouped in terms of semantics rather than frame. For instance, as shown in Figure 1, a woman meets a man and talks. The Phrase "A woman in gray shakes hands with a man in a suit" can be used to group the first three frames where the woman meets the man, and the last three frames are grouped to show they are talking. Given these two groups, the decoder can exploit the semantics of two groups in predicting the next word.

Notebook paper submitted to TRECVID 2022.

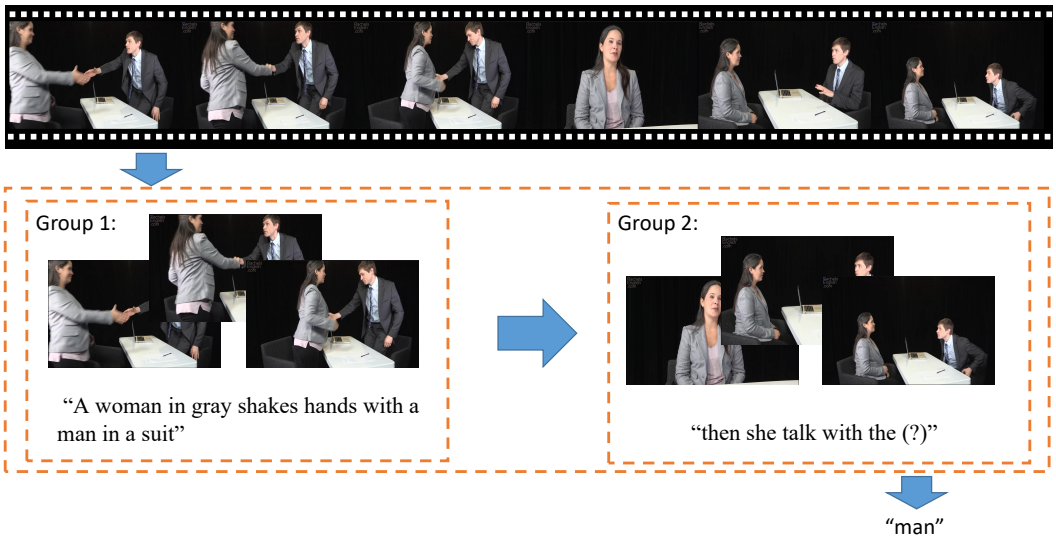

Figure 1: Phrases can gather their relevant frames in the video, forming groups that share common semantics within them. A decoder then exploits the necessary semantic group in predicting the next word ("man").

1.2 The Proposed Method

This notebook paper proposes a Semantic Alignment Network (SAN) to establish the mapping between each word and video frames to reduce redundancy. As shown in 2, SAN consists of three components: (a) Visual Encoder, (b) Semantic Aligner, and (c) Sentence Decoder. The details of each component are described in the following.

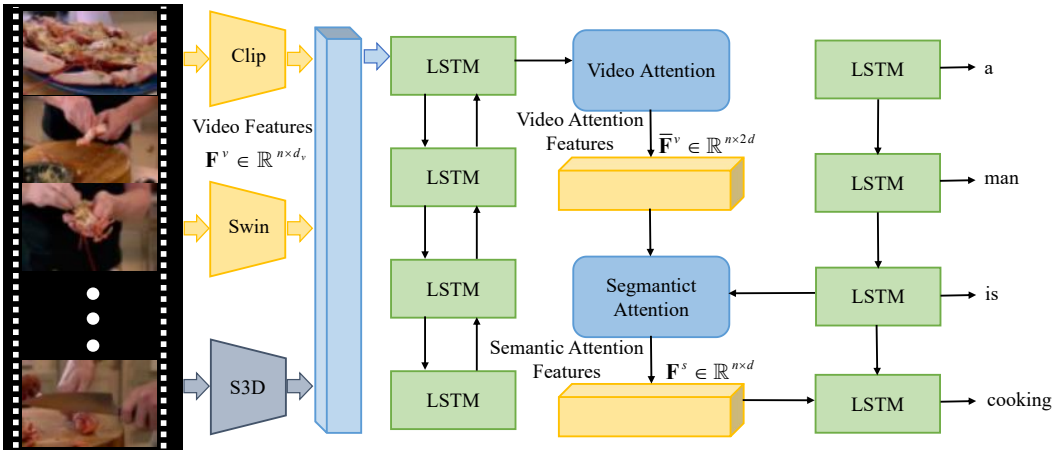

Figure 2: SAN consists of (a) Visual Encoder, (b) Semantic Aligner, and (c) Sentences Decoder. Video Encoder uses multiple different models to generate video embedding. Semantic Aligner consists of Bi-LSTM, video attention model, and semantic attention module, which establish a mapping relationship between words and video frames.

1.2.1 Visual Encoder

Given a video $\mathcal{V} = \{\mathbf{f}_k | 0 < k \leq n, \mathbf{f}_k \in \mathbb{R}^{w \times h \times c}\}$, where n is the number of frames, w, h, and c denote the width, height, and channel number of video frames, respectively. To encode videos, we extract three types of features by different backbone networks, including Clip[8], Swin-

Transformer[6] and S3D[11]. Then we concatenate the extracted features, resulting in video feature $\mathbf{F}^v \in \mathbb{R}^{n \times d_v}$.

1.2.2 Semantic Aligner

To build the semantic relation between generated words and video frames, we design a semantic aligner using the attention mechanism. First, we input the video features $\mathbf{F}^v \in \mathbb{R}^{n \times d_v}$ into a Bi-LSTM to get bidirectional video features $\hat{\mathbf{F}}^v \in \mathbb{R}^{n \times 2d}$. Then, we use the video attention module to get video attention features $\overline{\mathbf{F}}^v \in \mathbb{R}^{n \times 2d}$. The calculation process is as follows

$$\begin{aligned} \alpha_{i,j} &= \sigma(\hat{\mathbf{F}}_i^v(\hat{\mathbf{F}}_j^v)^\top), \\ \overline{\mathbf{F}}_i^v &= \textstyle\sum_{j=1}^{n} \alpha_{i,j} \hat{\mathbf{F}}_j^v \end{aligned} \tag{1}$$

where $\alpha_{i,j}$ is an attention weight, σ denotes an activation function such as hyperbolic tangent, and $\hat{\mathbf{F}}_i^v \in \mathbb{R}^{2d}$ denotes the bidirectional video features of i-th video frame.

Then, we design a semantic attention model to build semantic groups, which aligns frames around the phrases of partially decoded caption and describes the video by exploiting the semantic groups as processing units. Meanwhile, we use the attention mechanism to capture the semantic attention features $\mathbf{F}^s \in \mathbb{R}^{n \times d}$, i.e.,

$$\begin{aligned} \beta_{t-1,j} &= \mathbf{u}^\top \sigma(\mathbf{U}\mathbf{w}_{t-1} + \mathbf{H}\overline{\mathbf{F}}_j^v + \mathbf{b}) \\ \mathbf{F}_t^s &= \textstyle\sum_{j=1}^{n} \beta_{t-1,j} \overline{\mathbf{F}}_j^v \end{aligned} \tag{2}$$

where $\mathbf{w}_{t-1} \in \mathbb{R}^{n \times d}$ is the $(t-1)$-th word embeddings, $\beta_{i,j}$ is an attention weight, σ denotes an activation function, and $\overline{\mathbf{F}}_j^v$ denotes the semantic attention features of i-th video frame. In addition, $\mathbf{u}$, $\mathbf{U}$, $\mathbf{H}$ and $\mathbf{b}$ are trainable weights.

1.2.3 Sentence Decoder

Now we need to decode the semantic attention features to words. In particular, $\mathbf{F}^s$ is passed to an LSTM, and the t-th word embedding $\mathbf{w}_t$ is generated by a fully connected layer followed by a softmax layer as

$$\begin{aligned} \mathbf{h}_t &= LSTM(\mathbf{F}^s, \mathbf{h}_{t-1}) \\ \mathbf{w}_t &= Softmax(\mathbf{U}_h \mathbf{h}_t + \mathbf{b}_h) \end{aligned} \tag{3}$$

where $\mathbf{h}_t \in \mathbb{R}^d$ is the LSTM hidden state at the t-th time instance; $\mathbf{U}_h$ and $\mathbf{b}_h$ are trainable weights.

1.3 Experimental Setup

Firstly, we uniformly sample n=40 and change the size of each video frame to $224 \times 224 \times 3$ (i.e. w=224, h=224, c=3). Then, clip appearance features are extracted from the "Layer-normalization" of Clip, 1024D Swin appearance features are extracted from the "Flatten-layer" of Swin-Transformer, and 1024D motion features is extracted from the 'Fully- connected Layer' of S3D. Then we concatenate these three features to get 2816D video features (i.e. d_v=2816). We also set the hidden size of LSTM to 512D (i.e. d=512). In addition, we built the vocabulary based on those words appearing at least three times. For each paragraph, we remove punctuation marks and convert all alphabets to the lowercase. We also truncate the sequences longer than 28 for paragraphs and use 512D word embeddings for each word.

During the training stage, we optimize the model by the Adam with an initial learning rate of 1e-4 and a weight decay of 0.8. We train the model for at most 600 epochs with an early stopping using CIDEr-D. The batch size is se to 20.

1.4 Experiments

We employ the TRECVID VTT 2022 training data (10862 videos) to train the model, and the TRECVID VTT 2022 test data (2008 videos) to evaluate the model. Table 1 shows the results of the submission. Our model ranks in the second place on METEOR [4] and SPICE [1] metrics, and the third place on BLEU [7] metric.

Table 1: The results of VTT submissions.

Organization	SPICE	METEOR	BLEU
Renmin University of China	0.184	0.414	0.135
Ours	**0.107**	**0.290**	**0.071**
Elyadata	0.102	0.248	0.069
Waseda University	0.100	0.287	0.037
Nagaoka University of Technology	0.097	0.281	0.081
Carnegie Mellon University	0.077	0.222	0.030

(a) Graph structure detection mode

(b) Serial interactive aggregation mode

Figure 3: The object interaction relation is evaluated by different weights. Aggregating different types of object interactions helps to adjust the relation weights to obtain more accurate attention.

1.5 Conclusion

In this notebook paper, we propose a Semantic Alignment Network (SAN) for video captioning. It builds a mapping relationship between generated words and video frames to form semantic groups. The semantic groups are composed by the video frames with the coherent semantics, which are employed to predict the next word. Experimental results on VTT datasets show that our proposed model has excellent performance.

2 Activities in Extended Video (ActEV)

2.1 Introduction

Spatio-temporal action detection has been an active research area in computer vision due to its potential in a wide range of applications such as public safety and security. The Activity Extended Video (ActEV) challenge mainly focuses on human activity detection in multi-camera video streams. Recently, most spatio-temporal action detection models use a graph-based approach to characterize the interaction between targets. However, they fail to fully use the object interactions, resulting in less attention on the critical objects. Therefore, we model and integrate multiple types of interactions based on dense serials, and enhance the object features by aggregating different types of interactions. For example, as shown in Figure 3, men and women in conversation, when the object action features are not obvious, we need to use the object interactions to further boost the performance action classification.

2.2 The Adopted Method

In this task, we adopt a Dynamic Interactive Aggregation Network (DIAN), i.e. asynchronous interactive aggregation network [10], which aggregates different types of target interaction relations, and dynamically updates memory features by iterative self-learning. As shown in Figure 4, DIAN contains two components: (a) Interactive aggregation module; (b) Iterative memory update module. The details of each component are described in the following.

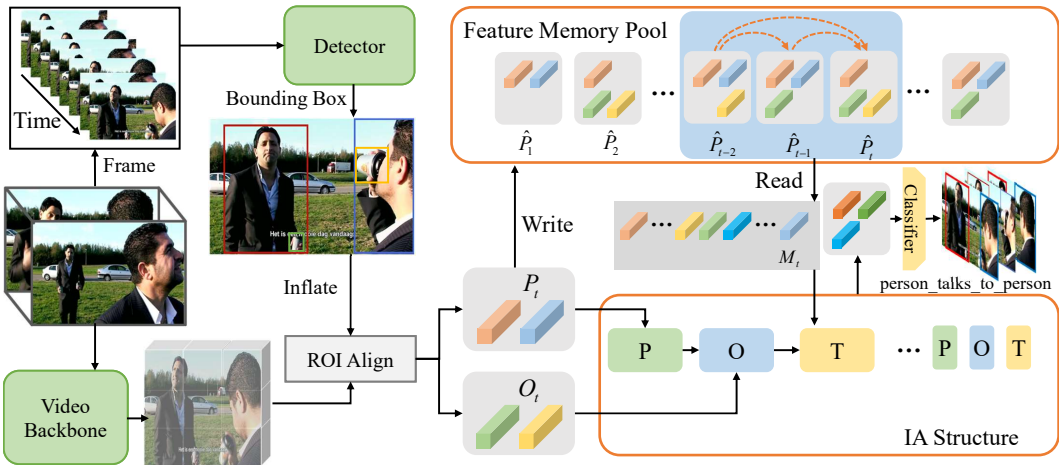

Figure 4: The DIAN [10] contains of (a) Interactive aggregation module: person features, object features, and memory features from the feature pool Ω in c are fed to IA to integrate multiple interactions. The output of IA is passed to the final classifier for predictions; (b) Iterative memory update module: read memory features from the feature pool and writes fresh person features to it.

2.2.1 Interactive aggregation module

In Dense Serial IA, each interaction block takes all the outputs of previous blocks and aggregates them using a learnable weight. Formally, the query of the i^{th} block is represented by

$$Q_{t,i} = \sum_{j \in \mathbf{C}} W_j \odot E_{t,j} \tag{4}$$

where $\odot$ denotes the element-wise multiplication, C is the set of indices of previous blocks, W_j is a learnable vector normalized by a Softmax function of C, $E_{t,j}$ is the enhanced output features from the j^{th} block.

2.2.2 Iterative memory update module

READ operation: At the beginning of each iteration, given a video clip $v_t^{(i)}$ from the i^{th} video, memory features before the target clip are read from the memory pool Ω, which is $\left[\hat{P}_{t-L}^{(i)}, \ldots, \hat{P}_{t-1}^{(i)}\right]$.

WRITE operation: At the end of each iteration, personal features for the target clip $P_t^{(i)}$ are written back to the memory pool Ω as estimated memory features $\hat{P_t^{(i)}}$, tagged with current loss value.

REWEIGHTING operation: The features we READ are written at different training steps. Therefore, some early written features are extracted from the model whose parameters are much different from current ones. Therefore, we use a simple yet effective way to compute such penalty factor by the loss tag. The calculation process is as follows

$$w_{t'}^{(i)} = \min\left\{err/\delta_{t'}^{(i)}, \delta_{t'}^{(i)}/err\right\} \tag{5}$$

where $w_{t'}^{(i)}$ denotes a penalty factor that discards poorly estimated features, the loss value $\delta_{t'}^{(i)}$ affects the convergence state of the whole network, err denotes the difference between the loss tag and current loss value.

2.3 Experimental Setup

We apply Faster R-CNN [9] framework to detect persons and objects on the key frames of each clip, and select the backbone SlowFast [5] network with ResNet-50 as our baseline model. The inputs

Table 2: The results of ActEV submissions.

Organization	AOD mean Pmiss@0.1rfa	AOD mean nMODE@0.1rfa	AOD mean nAUDC@0.2rfa	AD mean PMiss@0.1rfa	AD mean nAUDC@0.2rfa
BUPT	0.6309	0.0538	0.6705	0.5805	0.6231
UMD	0.8131	0.1620	0.8300	0.7789	0.7995
Ours	**0.9921**	**0.0303**	**0.9922**	**0.9728**	**0.9732**
Waseda	0.9961	0.1080	0.9964	0.9829	0.9850
M4D team	-	-	-	0.9823	0.9819

BUPT: Beijing University of Posts and Telecommunications

Waseda: Waseda University, Meimei University, SoftBank Corpotation

of our network are 32 frames, clips are scaled such that the shortest side becomes 256. We set L to 30 for memory features in our experiments. We train the model for 27.5k iterations with an initial learning rate 0.004 and the learning rate is reduced by a factor 10 at 17.5k and 22.5k iteration. A linear warm-up scheduler is applied for the first 2k iterations. We set the batch size to 32.

2.4 Experiments

We use the MEVA[3] video dataset to train the model and use the ActivityNet ActEV SRL Test dataset (201 videos) to evaluate the model. Table 2 shows the results submitted. Our model ranks the third among the teams of ActEV task.

2.5 Conclusion

For the ActEV task, we adopt the Dynamic Interactive Aggregation Network (DIAN), which integrates different types of interactions in the same segment in a dense series manner to adjust the relationship weight between objects. The memory features are dynamically updated by iterative self-learning to obtain long-term temporal interaction features. Experimental results on ActEV dataset show that our proposed model enjoy good performance.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: semantic propositional image caption evaluation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 9909, pages 382–398, 2016.

[2] Yangyu Chen, Shuhui Wang, Weigang Zhang, and Qingming Huang. Less is more: Picking informative frames for video captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 11217, pages 367–384, 2018.

[3] Kellie Corona, Katie Osterdahl, Roderic Collins, and Anthony Hoogs. Meva: A large-scale multiview, multimodal video dataset for activity detection. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1060–1068, January 2021.

[4] Michael J. Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Workshop on Statistical Machine Translation (WMT@ACL)*, pages 376–380, 2014.

[5] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6201–6210, 2019.

[6] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021.

[7] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, 2002.

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 139, pages 8748–8763, 2021.

[9] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Transactions Pattern Analysis and Machine Intelligence (TPAMI)*, 39(6):1137–1149, 2017.

[10] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu. Asynchronous interaction aggregation for action detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 12360, pages 71–87, 2020.

[11] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 11219, pages 318–335, 2018.