S2VTACT: Enrich Sequence to Sequence Video to Text Captioning with Action Recognition

Peng Wang Zheng Huang
Department of Computer Science
{pw7nc, zh4zn}@virginia.edu

Yanan Gong
Department of Statistics
yg3ca@virginia.edu

1. Introduction

Action recognition plays a significant role in video understanding and captioning. Previous works in video captioning, though constructed on multimodal data, didn't explicitly take the information of actions. In our project, we propose to design a video captioning model which combine main stream encoder-decoder framework with action recognition. For the part of action recognition, we leverage on the nature connectivity between joints. The joints, or skeleton, can be represented by 2D/3D coordinators of human joints. According to the connectivity, frames of graphs can be built. Thus, one's action can be represented by graph-based model. Then, in order to extract human's action precisely, Graph Convolutional Networks (GCN) can be applied to achieve this goal. To be more specific, for a video, which contains time axis, we will extent GCN to spatial and temporal space. This GCN can extract high-level features both on spatial and temporal dimension and thus we can capture people's action information in a video. In order to fusion these 2 models and introduce the extracted information from the action generation model to the video captioning model, we introduce 2 methods. First one is quite simple which directly substitute the <PAD> symbol before the <BOS> symbol such that the initial state of the sentence generation model would have the information of the specific action. The second one adopts a novel variant of vanilla GRU, named GFRU, to fuse the action feature and context feature together. Result shows that, by introducing this action information, our model can generate captioning with better expression of action features inside the video. Our codes is available at $https://github.com/stillarrow/S2VT\_ACT$

2. Related Work

2.1. Video Captioning

Video captioning is one of the research hotspots in recent years. As a high-level learning and representation task, video captioning links the area of CV and NLP. Recent models mainly focus on the encoder-decoder paradigm, where in the encoder part, the model encodes the video information (such as a stack of images on the time axis) of the input and in the decoder part, the model decodes the extracted feature maps to generate text description. For the encoder which tries to extract temporal information from a series of video frames, previous works use 2D/3DCNNs-based video encoder [8] or a combination of CNN and recurrent model such as LSTM [7]. For the decoder which tries to generate a description sentence, sequence models such as LSTM [7] and Attention mechanism [6] are being used. Some recent works also take advantage of the rapid development of pre-trained models (PTM) in NLP and use them to generate better representations of videos [5].

2.2. Action Recognition

For action recognition, like mentioned in [11], importing GCN to model human skeleton is one of current state-of-the-art. By using the "NTU RGB+D" dataset [4, 3] which provides both the RGB video and also the skeleton map of the human in the video, they constructed an undirected graph on a skeleton sequence with N joints and T frames featuring both intra-body and inter-frame connection. They then applied GCN to extract spacial information. The temporal aspect of the graph, as shown on figure 1, is constructed by connecting the same joints across consecutive frames. Thus, convolution operation is applied on the same joints of different frames. With some sampling strategies to implement GCN and attention mechanism, we can finally finish our action recognition model.

3. Approach

We proposed a 2-stage fusion model for video description, named as **S2VTACT** (Enrich **S**equence to **S**equence **V**ideo to **T**ext Captioning with **ACT**ion Recognition), where the input is the sequence of video frames $(x_1, ..., x_n)$ and the output is the sequence of words $(y_1, ..., y_m)$. The input and output usually have different length and in most case, $n >> m$.

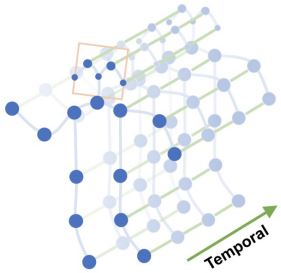


Figure 1. Example of skeleton graph.

Previous works usually focus on constructing a model to directly generate the text from given video frames. However, these kind of 1-stage model only leverage on sequence models (such as RNN, LSTM of even a stack of image frames for 3DCNN) in the encoder part to extract temporal information from the video. Previous works [9] have shown that encoder based on recurrent models, compared to encoder based on 3DCNN, is more powerful on extracting long-term temporal information, while 3DCNN has a better understanding of the action information. Thus, in order to remedy the defect of recurrent models, we turn to construct a 2-stage model.

The first stage is a traditional video captioning model with sequence-to-sequence structure. This is shown in the *video captioning* part of figure 2. The second stage is an action recognition model based on GCN. This is shown in the *Action Recognition* part of figure 2. We then add the action feature we generated from second stage (i.e. action recognition) into the decoder of the first stage (i.e. video captioning) so that the extracted action information can be fed and emphasized in our text decoder. In order to inject the action feature into text decoder, we proposed 2 methods. The first one is to directly introduce the action feature to the second GRU (colored green in Figure 2) at timesteps before the start of sentence generation. The second method adopts a variant of GRU called GFRU to combine both sequential information (including video frame features and sentences) and arbitrary specified features (action label here in our case). However, due to time limitation, we currently still haven't received the result of the model using GFRU, thus this part would be discussed in the Appendix. We believe this kind of 2-stage model will help the recurrent model better understanding the information of action which would lead to a better description of the video. The video captioning model, the action recognition model and model fusion are introduced in the next three sections step by step.

3.1. Video Captioning

The video captioning model we are using is similar to the work of S2VT [7]. Basically, as we can see in figure 2, it takes input of a sequence of video frames and then extracting information from them using a 2DCNN. Same as normal classification tasks, we can use many kinds of pretrained model to extract features of the video frames here, such as AlexNet, VGG, GoogleNet and ResNet. The generated sequence of video frame representations severs as the input sequence (colored yellow) and by using a stack of two GRUs, this model can generate a sentence description of the input video. In the original S2VT model, they used a recurrent unit of LSTM but here we replace it with GRU because it shows competitive performance with much better computational efficiency than LSTM.

Although there are 2 stages (i.e. encoder and decoder) here in S2VT, for video frames information and sentences information, the model uses the same GRU in these 2 stages, respectively, so that parameter sharing is allowed between the encoding and decoding stage which would be useful for extracting long-term dependent information. Another core design of S2VT is that the model uses two stacked GRUs which are unrolled over time and clearly shown in Figure 2. Since the 2 GRUs are stacked together, the hidden representation (h_t) from the first GRU (colored red) layer is provided as the input (x_t) to the second GRU (colored green). The first GRU layer is used to model the visual frame sequence, and the next layer is used to model the output word sequence.

3.2. Action Recognition

For the action recognition model, we use Spatial-Temporal Graph Convolutional Net- works (ST-GCN) proposed by [11]. This model takes a human skeleton graph as an input.

3.2.1 Spatial-Temporal Skeleton Graph Construction

Human's skeleton can be represented by an undirected spatial temporal graph G = (V, E) on a skeleton sequence with N joints and T frames featuring both intra-body and inter-frame connection. To be more specific, in this graph, the node set $V = \{v_{it} | t = 1, ..., T, i = 1, ..., N\}$ includes the all the joints in a skeleton sequence. Each node represent each joint of a people. The feature of each node contains coordinate vectors, as well as estimation confidence, of the i-th joint on frame t. First, the joints within one frame are connected with edges accord- ing to the connectivity of human body structure, which is illustrated in 1. Then each joint will be connected to the same joint in the consecutive frame. Thus the edge set is consist of two parts, the first subset depicts the intra-skeleton connection at each frame, denoted as $E_S = \{v_{ti}v_{tj} | (i, j) \in H\}$, where H is the set of naturally connected human body joints. The sec- ond subset contains the inter-frame edges, which connect the same joints in consecutive frames as $E_F = \{v_{ti}v_{(t+1)i}\}$. Therefore all edges in E_F for one particular joint i will represent

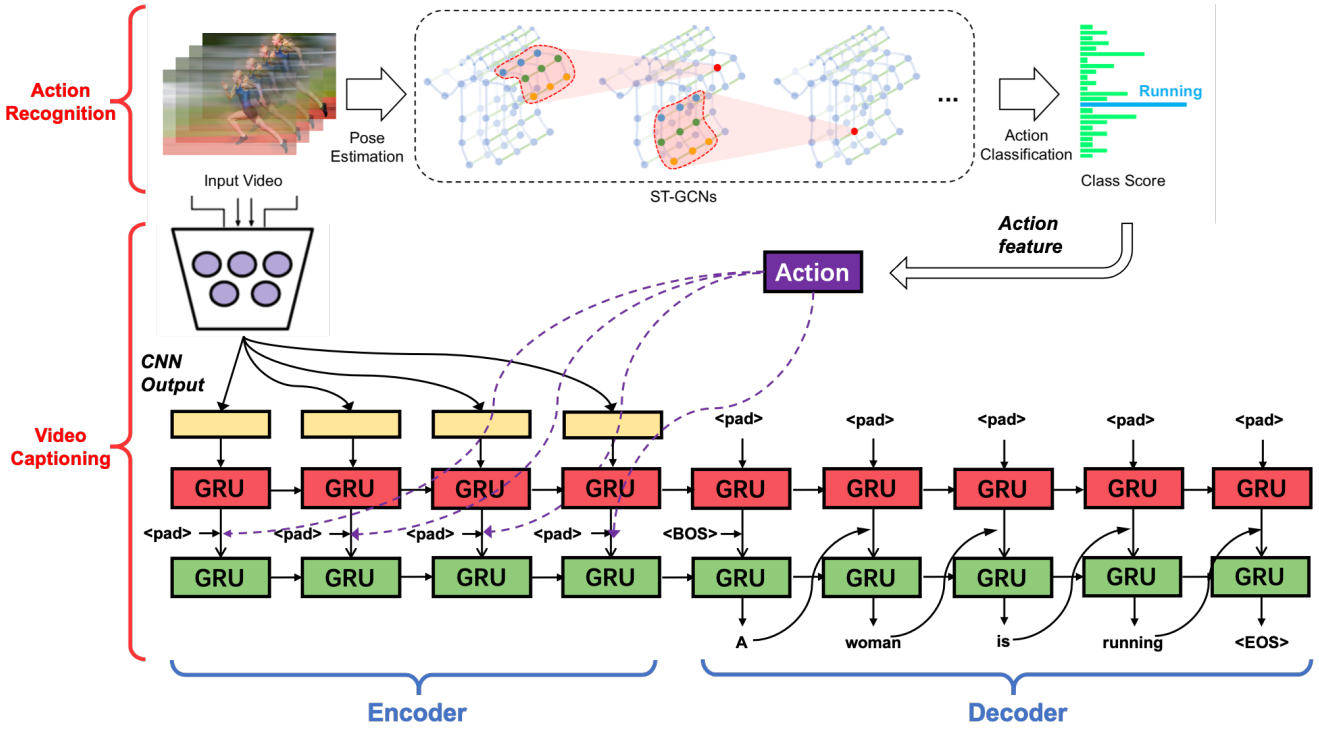


Figure 2. Structure of our proposed 2-stage model. For action recognition, we use exactly same model as *ST-GCN*. For video captioning, we follow the previous work of *S2VT*, we use a stack of 2 recurrent units (here we choose GRU) that learn a representation of the sequence of the frames and then decode it into a sentence. The top GRU layer (colored red) models visual feature inputs. The second GRU layer (colored green) models language given the text input and the hidden representation of the video sequence. We use <BOS> to indicate begin-of-sentence and <EOS> for the end-of-sentence tag. Zeros are used as a <pad> when there is no input at the time step. In order to introduce the action feature, we substitute the <pad> input during the encoder procedure of the second GRU with the predicted action label from the *ST-GCN* model.

its trajectory over time.

3.2.2 Spatial Graph Convolutional

Given a convolution operator with the kernel size of KK , and an input feature map $f_{in}$with the number of channels c, the output value for a single channel or a frame at the spatial location x can be written as

$$f_{out}(out) = \sum_{h=1}^{k} \sum_{w=1}^{k} f_{in}(p(x,h,w))w(h,w)$$

. Here P, $Z^2 * Z^2 \rightarrow Z^2$,is a sample function that enumerate the neighbors of location X. The weight function provides a weight vector in a specific frame for computing the inner product with the sampled input feature vectors f_{in}. In this work, we sample the neighbor of node $B(v_{ti} = \{v_{tj}|d(v_{tj}, v_{ti} <= 1)\})$. The function d(.) means the minimum length of any path from v_{tj} to v_{ti}, which means we only sample one-hop neighbor of node v_{ti}. Then we have a mapping function $l_{ti} : B(v_{ti}) \rightarrow \{0, 1, 2, ..K-1\}$ which maps a node in the neighborhood into a fixed number of K subsets. Each subset has a numeric label. Thus, we can align each sampled nodes with the weight matrix then calculate the final result of graph convolution.

3.2.3 Spatial Temporal Modeling

We define a very simple strategy to extend the spatial graph CNN to the spatial temporal domain. We sample the neighbor of node v_{it} at temporal domain. Therefore the neighbor set of v_{it} $B(v_{it}) = v_{qj}|d(v_{tj}, d(v_{ti}) <= K, |q-t| <= \Gamma/2)$, where Γ controls the temporal range to be included in the neighbor graph and can thus be called the temporal kernel size. Then we run a 1-D convolution to generate the result on temporal dimension.

3.2.4 ST-GCN Implementation

We follow the idea of [1]. For the spatial dimension, we use an adjacency matrix A to represent the connection of joints within a single frame and an identity matrix I representing self-connections. Thus with a specific sampling strategy, the final formula of Graph Convolution Neural Networks

can be represented as

$$f_{out} = \Lambda^{-\frac{1}{2}}(A+I)\Lambda^{-\frac{1}{2}} f_{in} W,$$

where $\lambda^{ii} = \sum_j (A^{ij} + I^{ij})$

3.3. Model Fusion

Many works in video captioning use multimodal data as input. Beside original video or image data, other modalities such as audio are also being fused. Instead of directly fusing data from different modality, we borrow ideas from multi-task learning and turn to combine different models. As described before, we propose to inject the action feature (which is generated from the action recognition model) into the text generation procedure in the model of video captioning. We proposed 2 methods to inject the action feature. The first one simply takes the predicted action label as a word and add this word into the encoding stage before caption sentence generation (i.e. decoding stage). The second method steps further. Instead of simply adding action feature into the first hidden state of the decoder, we follow the work of [2] which injects the feature information at each time step of the decoder. More detailed discussion on GFRU is provided in the Appendix.

For simplicity we use the first proposed method to inject action label into RNN cells. To fuse the model of video captioning and action recognition, we first introduce ResNet as a feature extractor of videos. To be more specific, we feed each frame of an input action video into a ResNet. Then each top layer GRU cell takes the extracted feature map as an input during the encoding stage. At the same time, for the same action video, we run ST-GCN based action recognition model to identify which action appeared within the video. Then the detected action label would be send as the input of each GRU cell within the second GRU layer(colored green). Thus, every encoder will contain information of the current video action, which we assume may improve the performance of captioning generation.

4. Experiment Setup

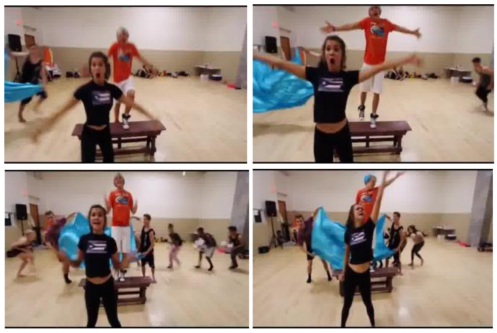
This section describes the evaluation of our approach. We first describe the 2 datasets we used, for the task of video captioning and action recognition, respectively and then the details of our models.

4.1. Datasets

Based on our idea, we will use two datasets: one for video captioning and another for action recognition.

4.1.1 Dataset 1: Video captioning

We use the MSR-VTT-10k[10] dataset for video captioning. It provides 10K web video clips (6,513 for train, 497 for validation and 2,990 for test) with 41.2 hours and 200K clip-sentence pairs in total (each video clip has 20 candidate captions), covering the most comprehensive categories and diverse visual content, and representing the largest dataset in terms of sentence and vocabulary. It contains 20 categories including music, people, gaming and sports. Although many classes don't seem to highly related with any action, it turns out that 98.34% videos actually has at least 1 candidate caption that contains one of the action label of the "NTU RGB+D" dataset which we are using for the action recognition model. Since the dataset is quite noisy, we filtered out all the words which only appeared once. The total size of remaining vocabulary is 16863 (including special tokens). The sample input and output is shown as Figure 3.



1. A man and a woman performing a musical.
2. A teenage couple perform in an amateur musical
3. Dancers are playing a routine.
4. People are dancing in a musical.
5. Some people are acting and singing for performance.

Figure 3. Example of input/output of MSR-VTT

4.1.2 Dataset 2: Action recognition

"NTU RGB+D" is the action recognition data which is constructed by 60 action classes and 56,880 video samples. It contains RGB videos, depth map sequences, 3D skeletal data, and infrared (IR) videos for each sample. The resolutions of RGB videos are 1920x1080, depth maps and IR videos are all in 512x424, and 3D skeletal data contains the 3D coordinates of 25 body joints at each frame. The 60 actions in this dataset are in three major categories: daily actions, mutual actions, and medical conditions. The sample input and output is shown as Figure 4.

4.2. Evaluation Metrics

For action recognition, we use top-1 and top-5 accuracy. For video captioning, we use BLEU-1 to BLEU-4, METEOR and ROUGE-L to evaluate the quality of our generated sentence. For all these metrics, the higher is better.

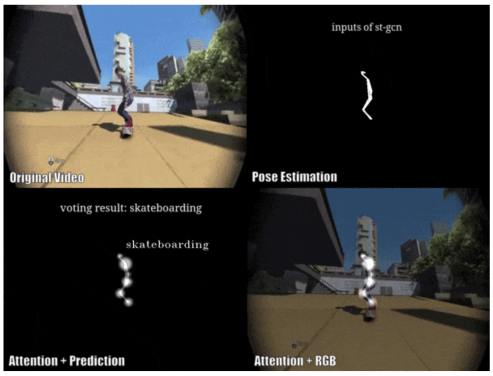


Figure 4. Example of input/output of NTU-RGB+D

4.3. Experimental details of our models

We run our action recognition model on NTU-RGB+D dataset. We evaluate the recognition performance by top-1 and top-5 classification accuracy. We report the top-1 and top-5 accuracy as 88.76% and 98.83%. The detail setting of the action recognition model is shown on table1.

Table 1. Parameter Setting for Action Recognition

Parameter	Value
# action classes	60
# epoch	80
Batch size	256
Test batch size	64
Dropout	0.5
Learning rate	0.01
Optimizer	SGD

We run this model on NTU-RGB+D to test its performance. NTU-RGB+D is currently the largest dataset with 3D joints annotations for human action recognition task. This dataset contains 56, 000 action clips in 60 action classes. These clips are all performed by 40 volunteers captured in a constrained lab environment, with three camera views recorded simultaneously. The provided annotations give 3D joint locations (X, Y, Z) in the camera coordinate system, detected by the Kinect depth sensors. We evaluate the recognition performance by top-1 and top-5 classification accuracy. We can see that the ST-GCN works well on this dataset and demonstrates the effectiveness of the proposed spatial temporal graph convolution operation, which means this proposed model is capable of action recognition task.

Since the NTU-RGB+D dataset doesn't have captioning sentences, this action recognition model is only used as a pre-trained model which can infer the action for video in the MSR-VTT dataset. We then train our model on MSR-VTT dataset and compare the performance result of our S2VTACT model with the original S2VT model. Since basically S2VTACT and S2VT shares the same network structure, thus the parameter settings for these 2 models are set to be the same. Details about the used parameters and their values are shown on the Table 2. During training, we use NLLLoss to measure the difference between generated captioning sentences and ground truth label. Since each video clip has 20 candidates captions, we randomly select one of them as the ground truth during training every-time we encounter with this specific video.

Table 2. Parameter Setting for Video Captioning

Parameter	Value
# max length of text	28
# epoch	1000
Batch size	256
Test batch size	64
Dropout	0.5
Learning rate	0.0004
Learning rate decay every (epochs)	200
Learning rate decay ratio	0.8
Optimizer	Adam

5. Results and Analysis

5.1. Quantitative Study

A quantitative result of our S2VTACT model campared with S2VT model is shown in Table 3.

Table 3. Performance comparisons between S2VT and S2VTACT on the test set of MSR-VTT dataset in terms of BLEU@1 to BLEU@4, METEOR and ROUGE_L scores (%), higher the better.

Model	B@1	B@2	B@3	B@4	M	R
S2VT	32.0	21.6	14.2	9.1	12.4	30.5
S2VTACT	32.5	21.7	14.2	9.1	12.5	30.3

As you can see, the performance of these 2 models are quite same. S2VTACT works slightly better than S2VT on lower rankded BLEU metrics which might because that S2VTACT would generate sentence with more accurate action expression. However, on the other metrics, these 2 models has a really close result. This might be because that although S2VTACT has more information about action and is kept emphasized by the 'forced' injected action feature, this kind of method might perturb the continuity and context of the sentence generation which leads to not that satisfied result. Another reason may be that both models only trained for 1,000 epochs (due to time limitation) while

the loss is still decreasing. Therefore, if we can train more epochs, then the result would become better.

5.2. Qualitative Study

A qualitative result of our S2VTACT model is shown in Figure 5. Here we show 3 example from the test set of MSR-VTT dataset. For each video clip, we provide 3 frames and 5 ground truth captions. We also show the generated caption from both S2VT model and our S2VTACT model.

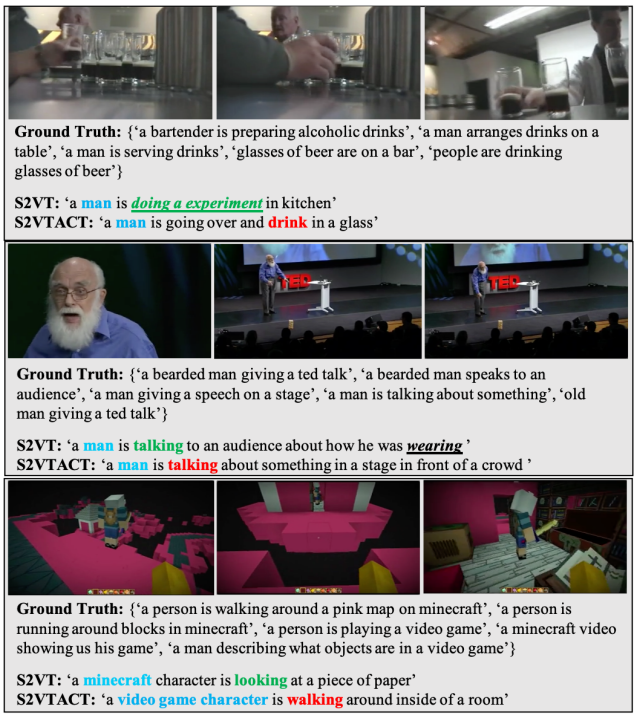


Figure 5. Qualitative comparison between the S2VT(Baseline) and our S2VTACT model by sampling from the test set of MSR-VTT datasets. Three frames are shown for each video clip. 6 human annotated descriptions are listed for illustration. Text in blue highlights the subject in a sentence. Words in green shows the and red show the predicted action by S2VT(Baseline) and by our S2VTACT, respectively. The bold italic text which is emphasized by underlines is the text we think irrelevant to the scene in the video.

Result shows that our S2VTACT model can better capture the action feature in the scene of the video ('drink' for the first example and 'walking' for the third example). We also find that S2VT model will sometimes generate irrelevant topics (such as 'doing experiment' in the first example and 'how he was wearing' in the second example). However, by injecting the action feature of this scene, S2VTACT model will focus more on generating captions which pay more attention to the action in this scene. Although this may result in losing some fine-grained features (like 'paper' in the third example), we believe that it's more important and valuable to focus on generating captions that is related to the main action of a specific video scene.

6. Conclusion and Future Work

In this paper, we introduce to model, one for action recognition, ST-GCN, another for video captioning, S2VT. We present an idea of combination and run them together and demonstrate that our proposed idea can effectively capture the action appeared in a video. More importantly, the video captioning model can preserve this information as well. The quantitative and qualitative experiments shows the combined model,S2VTACT, can further improve the performance of video captioning.

There are several directions for future work. One could inject labels into S2VT decoder and fine-tune the two models to predict the video captioning accurately. In addition, due to device limitation, more training epoch can be applied into future work. We believe that after more training epoch, the captioning performance will be higher. We provide more detailed future direction in appendix.

References

[1] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[2] L. Li, Y. Zhang, and L. Chen. Generate neural template explanations for recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 755–764, 2020.

[3] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[4] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

[5] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7464–7473, 2019.

[6] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[7] 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*, pages 4534–4542, 2015.

[8] J. Wang, W. Wang, Y. Huang, L. Wang, and T. Tan. M3: Multimodal memory modelling for video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7512–7520, 2018.

[9] Y. Wang, L. Jiang, M.-H. Yang, L.-J. Li, M. Long, and L. Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *International Conference on Learning Representations*, 2018.

[10] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[11] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1801.07455*, 2018.

A. Proposed Model for Future Work

As we have discussed before, directly injecting action features into the encoding stage of the top GRU layer is not an ideal solution. Here, we will explain the second method which uses a GFRU (first introduced in the work [2]) unit for the decoding stage of the second GRU layer (colored blue in Figure 7). An illustration of this novel GRU cell is shown in Figure 6. As we have discussed before, this GFRU cell is a combination of 2 GRU cells (one of them serves as the context cell and the other serves as the feature cell). *Context GRU* cell is used to process the sentence information while the *Feature GRU* is used to inject the action feature that we want to emphasize in this video scene. The Gated Fusion Unit (GFU) decides which GRU's word to be emitted at each time step. By doing so, the extracted action information is emphasized during the description generation procedure.

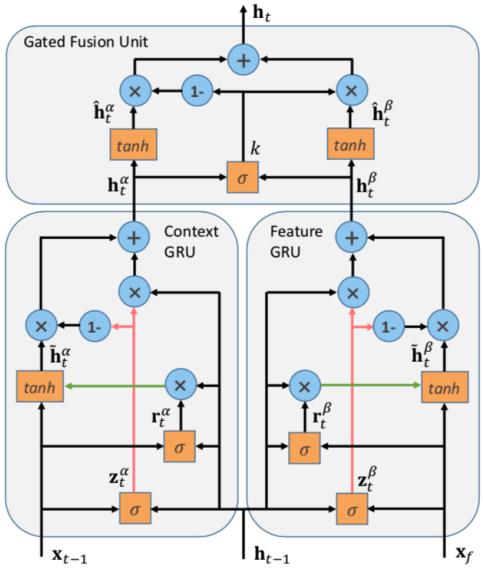


Figure 6. The structure of the GFRU decoder which is used for model fusion. This GFRU has three components. The word at the current time step and the action feature are processed by the bottom two GRUs, respectively, whose outputs are merged by the GFU component, which produces a final hidden state for the current step. This GFRU can be understood as a weighting model between context information and feature information.

Specifically, this GFRU cell can be written as $\mathbf{h}_t = g(\mathbf{x}_{t-1}, \mathbf{h}_{t-1}, \mathbf{x}_f)$ where $\mathbf{x}_f$ is the representation of the feature $f \in \mathcal{F}$, and $\mathcal{F} \subset \mathcal{V}$. In our case, f is the action feature, $\mathcal{F}$ is the total action set and $\mathcal{V}$ is the total vocabulary. $\mathbf{x}_{t-1}$ represents the last generated word and $\mathbf{h}_{t-1}$ is the last hidden state of the GFRU. Fore more detail of this model, please check their work [2].

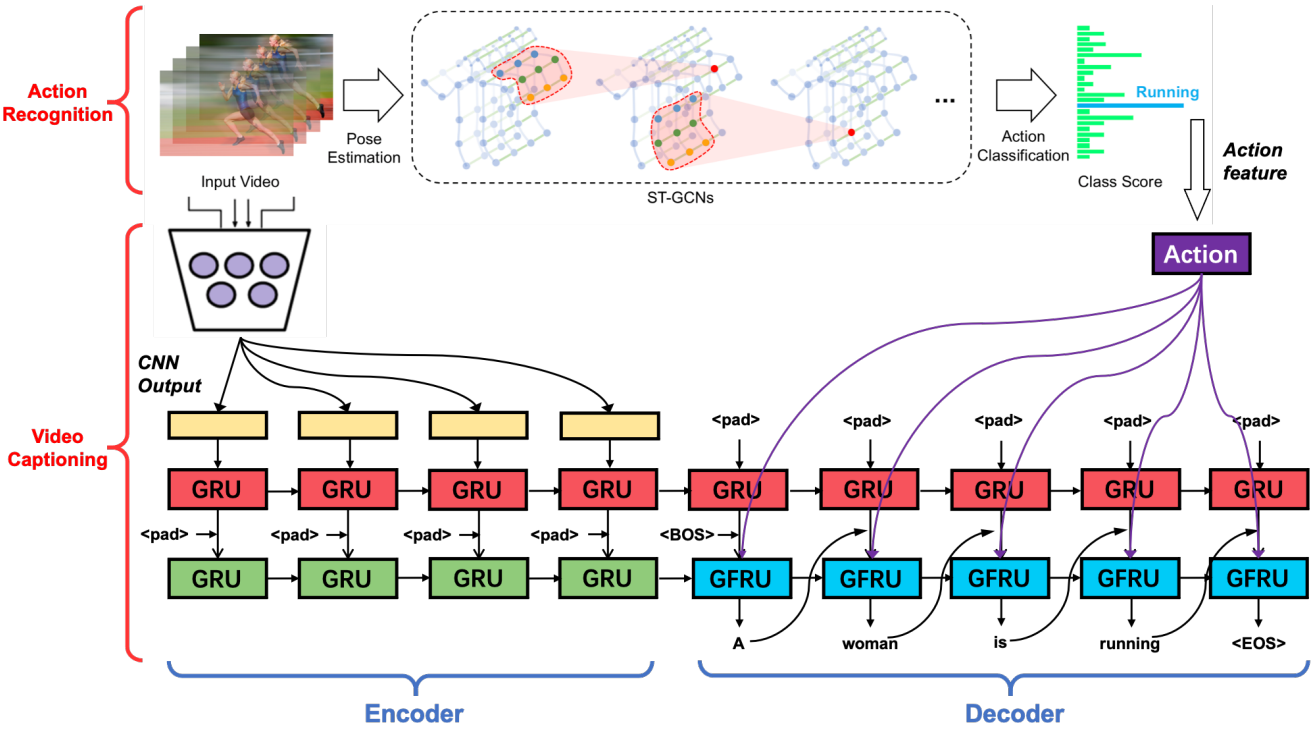


Figure 7. Structure of our proposed 2-stage model. For action recognition, we use exactly same model as *ST-GCN*. For video captioning, we follow the previous work of *S2VT*, we use a stack of 2 recurrent units (here we choose GRU) that learn a representation of the sequence of the frames and then decode it into a sentence. The top GRU layer (colored red) models visual feature inputs. The second GRU layer (colored green and blue) models language given the text input and the hidden representation of the video sequence. We use a variant of GRU called GFRU (colored blue) which enable the origin GRU to take 2 input features at each timestep. We use <BOS> to indicate begin-of-sentence and <EOS> for the end-of-sentence tag. Zeros are used as a <pad> when there is no input at the time step.