11-2018

Learning generalized video memory for automatic video captioning

Poo-Hee CHANG

Ah-hwee TAN
Singapore Management University, ahtan@smu.edu.sg

Citation

CHANG, Poo-Hee and TAN, Ah-hwee. Learning generalized video memory for automatic video captioning. (2018). *Multi-disciplinary International Conference on Artificial Intelligence: 12th International Conference: MIWAI 2018, Hanoi, Vietnam, November 18-20: Proceedings*. 11248 LNAI, 187-201.

Available at: https://ink.library.smu.edu.sg/sis_research/6076

Learning Generalized Video Memory for Automatic Video Captioning

Poo-Hee Chang(✉) and Ah-Hwee Tan(✉)

School of Computer Science and Engineering, Nanyang Technological University,
Singapore 639798, Singapore
{phchang,asahtan}@ntu.edu.sg

Abstract. Recent video captioning methods have made great progress by deep learning approaches with convolutional neural networks (CNN) and recurrent neural networks (RNN). While there are techniques that use memory networks for sentence decoding, few work has leveraged on the memory component to learn and generalize the temporal structure in video. In this paper, we propose a new method, namely Generalized Video Memory (GVM), utilizing a memory model for enhancing video description generation. Based on a class of self-organizing neural networks, GVM's model is able to learn new video features incrementally. The learned generalized memory is further exploited to decode the associated sentences using RNN. We evaluate our method on the YouTube2Text data set using BLEU and METEOR scores as a standard benchmark. Our results are shown to be competitive against other state-of-the-art methods.

Keywords: Memory model · Video captioning · Deep learning
Adaptive Resonance Theory · LSTM · CNN

1 Introduction

Automatic video captioning has a wide array of applications, such as artificial consciousness, videos categorization and aids for the visually impaired. It involves the understanding and translation of temporal visual features into words. While video captioning is a challenging task for both computer vision and language, recent progress with deep neural networks have led to many possibilities in regards to automatic video captioning.

Current deep learning approach to the video captioning task typically involves a deep visual encoder using convolutional neural networks (CNN) such as AlexNet [17] and a sentence decoder using Long-Short Term Memory (LSTM) [14], a variant of recurrent neural network (RNN). For example, the mean pool approach [34] which takes the average of the AlexNet features across the video frames. The mean pooled vector trains the LSTM network and decodes a sequence of words. The mean pooling approach serves as a baseline to many of the recent state-of-the-art algorithms. Further research has explored

M. Kaenampornpan et al. (Eds.): MIWAI 2018, LNAI 11248, pp. 187–201, 2018.
https://doi.org/10.1007/978-3-030-03014-8_16

the temporal representation of the videos [33], the temporal representation of the sentence decoder [41] and the visual attention mechanism [18] for improving the quality of video captions.

Another area of research involves the introduction of memory into deep neural models. Weston et al. introduced the Memory Networks [40] which enabled RNNs to memorize long sequences. The general model of the memory network is to generalize inputs, to retrieve memories, and to interpret the stored memories. Memory networks have shown to be able to tackle textual and visual question and answering tasks [39]. While there are some video captioning work involving the memory component, such as Iterative Attention/Memory [9] which is a memory model based attention mechanism, there is a lack of a memory model that focused on generalizing and storing the temporal structure of the video. Previous memory model for video captioning have a fixed pre-defined number of memory slots which may limit the number of useful memories stored.

To address the issue, we adapt from an earlier work, the Adaptive Resonance Theory (ART) [2] which is a class of self-organizing neural network. The ART model is able to stores input patterns in a content addressable way with unlimited categories or memory slots. To our best knowledge, the use of ART neural network as the external memory module for general deep learning tasks are not explored. One possible reason may be the difficulty of integrating the ART network with deep learning methods as the ART model does not learn by back-propagation.

In this paper, we present the video captioning architecture, with a memory model named the Generalized Video Memory (GVM) that is able to generalize and store the temporal video features using the ART framework. GVM is able to store memory incrementally, in which are retrieved for improving caption decoding. Our main contribution of this paper is to show that how a memory model based on the ART is integrated with the deep learning approach. The GVM is able to generalize and retrieve the temporal structure of the video to improve the quality of the video description base on a deep learning framework. We construct our framework that is based on the mean pool approach [34], with the integration of the GVM model. The mean pooled features are the representation of the visual features within a video. Our method explores on the representation and storage of similar video features into memories. The idea is analogous to a human drawing past experiences and knowledge to conduct an informed judgement based on a limited sensory information. By recalling memory of similar videos, the LSTM caption decoder is able to utilize the additional information and generate better quality sentence. Using the publicly available YouTube2Text data set [5], we show that by combining GVM with the basic mean pool approach, we can obtain competitive results as compared to the current state-of-the-art methods.

The organization of this paper is as follows. We report the related work for video captioning in Sect. 2. In Sect. 3, we introduce our GVM network and the video captioning framework. In Sect. 4, we discuss the details of the experiment set up. We then illustrate the performance of GVM by comparing other state-of-the-art methods in Sect. 5. We then finalize our paper with concluding remarks in Sect. 6.

2 Related Work

Early research on video captioning has focused on using various image processing techniques to extract the best subject, verb and object tuples from the video. Together with rule based, statistical modeling and sentence templates [12,16], they are able to produce grammatically correct sentences. However, these methods has focused on a narrow domain with limited vocabularies describing the objects and its the activities.

Recent success with large-scale image recognition using convolutional neural networks (CNN) [17,22,27] and language modeling and translation using variants of the RNN [10,25,26] have inspired researchers to combine both deep learning domains for work regarding image captioning [15,35].

The natural progression of image captioning using deep learning approach is to extend to the video domain. Venugopalan et al. [34] used the AlexNet to extracts frame by frame features. Mean pooling or averaging is applied to the frame features. The mean pooled vector is presented as input to a two-layer LSTM network for generating descriptions. While this method of averaging frame features loses the representation in the temporal aspects, it has provided a good baseline result for other video captioning work.

To address the temporal representations of video for generating description, a sequence to sequence LSTM framework is proposed for temporal modeling of videos and language [33]. Recent work include the attention mechanism [18,41] which are able to selectively focus on the given input video features. However, these models do not attend to other videos which have similar visual features.

There is a trend in deep learning to integrate the use of memory into neural network models. Weston et al. introduced the Memory Networks [40] which can help RNNs to memorize long sequences. This memory model is known to be difficult to train by backpropagation. Sukhbaatar et al. [24] proposed an end-to-end memory network that requires less supervision in training. Memory networks have been shown to be able to do textual and visual question and answering tasks [39]. However, these memory models do have limited slots to store memories.

The ART network [2] was proposed to learn memory or cognitive nodes by encoding input patterns and to support the recognition and recall of the stored patterns. A vigilance parameter is used to control the level of generalization on the stored patterns. Tan et al. proposed the fusion ART [30], which is an ART variant with multi input fields. The fusion ART model has been applied to the modeling of episodic memory [4,23,37,38] as well as to the reinforcement learning [28,29,31,36] domain. Given that the ART network does not learn by backpropagation, it may present a challenge to integrate the ART model into an end-to-end neural network. Our work focuses on how the ART-based memory module is able to integrate into a deep learning approach to the video captioning domain.

3 Video Captioning Using Generalized Video Memory

The video captioning task encodes visual features from an image sequences $(v_1, v_2, ..., v_n)$ and decodes a sequence of words $(y_1, y_2, ...y_m)$. The length of both input and output sequences are variable. In this paper, we propose a memory model named Generalized Video Memory (GVM) network for storing and recalling of the generalized video features. Our automatic video captioning framework is based on Venugopalan et al. [34] mean pooling approach. Firstly, a CNN based image encoder is used for extracting frame features. The extracted frames features are averaged (mean pooled). The mean pooled vector representing the video are presented to the GVM for memory generalization storage. The mean pool vector and the generalized video memory features are provided as the inputs to a two-layer LSTM caption decoder. By using additional information from the memory, the caption decoder is able to generate better captions. Figure 1 shows our video captioning framework integrating with the GVM network. In the following sections, we describe our video captioning framework with the GVM network.

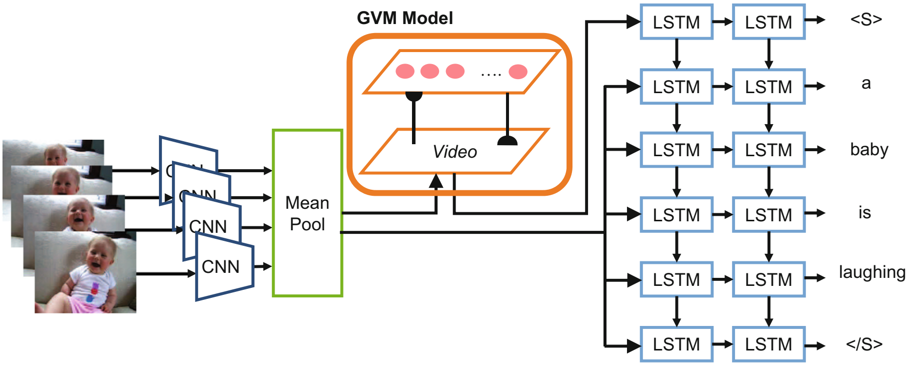


Fig. 1. Video captioning framework with Generalized Video Memory (GVM) model. The token <S> represents the start-of-sentence and the token </S> represents the end-of-sentence.

3.1 The CNN Video Encoder

The 16-layer CNN based image encoder, VGG16 [22], is used for encoding the image features from the video clip. The VGG16 network is loaded with pre-trained parameters trained with the 1.2M subset images from the ImageNet data set [21]. We use the publicly available VGG16 implementation from Caffe [8] and converted the implementation to tensorflow. Image features (4096 dimensional vector) are extracted from the fully connected layer (fc2) after ReLU activation. To encode the entire video, image features extracted are averaged (mean pooled) across the video frames to form a mean pooled vector fc_{mean}.

3.2 The Generalized Video Memory Network

The GVM network proposed in this paper is based on Adaptive Resonance Theory (ART) neural network [2] and fusion ART [30]. Figure 2 shows architecture of the GVM network. The GVM network consists of a input field, namely the *video* field, and a category field. The network is designed to learn cognitive nodes at the category layer (F_2), while encoding the input patterns at the input field layer (F_1). During learning, the input vector presented to the input field is matched against the cognitive nodes at the F_2 layer. The matching criteria is controlled by the vigilance parameter. When a match is found, the matched cognitive node adapts its weights to the new input vector. If no match is found, a new cognitive node is recruited which learns the newly presented input vector. Thus, the GVM performs fast and stable learning in response to a continual stream of input patterns, and learns new patterns incrementally. It supports the recognition and recall of the stored patterns based on similarity of the search cue. For completeness, the network dynamics are described below.

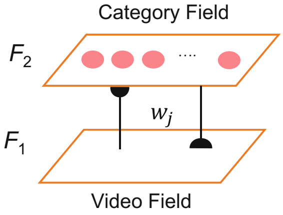


Fig. 2. The Generalized Video Memory (GVM) network.

Input vectors: Let $\mathbf{I} = (I_1, I_2, \ldots, I_n)$ denote an input vector, where $I_i \in [0, 1]$ indicates the i^{th} input element. Let the complement coded vector be $\bar{\mathbf{I}}$, such that $\bar{I}_i = 1 - I_i$.

Input fields: Let F_1 denote an input field that holds the input pattern for the video features. Let $\mathbf{x} = (x_1, x_2, \ldots, x_{2n})$ be the activity vector of F_1 receiving the input vector $\mathbf{I}$ (including the complement) such that $\mathbf{x} \leftarrow (I_1, I_2, \ldots, I_n, \bar{I}_1, \bar{I}_2, \ldots, \bar{I}_n)$.

Category field: Let F_2 denote the category field. Let $\mathbf{y} = (y_1, y_2, \ldots, y_m)$ be the activity vector of F_2.

Weight vectors: Let $\mathbf{w}_j$ denote the weight vector associated with the jth node in F_2 for learning the input pattern in F_1.

Parameters: Each field's dynamics is determined by choice parameters $\alpha \geq 0$, learning rate parameters $\beta \in [0, 1]$, contribution parameters $\gamma \in [0, 1]$ and vigilance parameters $\rho \in [0, 1]$.

The dynamics of the GVM network can be considered as a system of continuous resonance search processes comprising the basic operations as follows.

Code activation: A node j in F_2 is activated by the choice function

$$T_j = \gamma \frac{|\mathbf{x} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|}, \tag{1}$$

where the fuzzy AND operation $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv min(p_i, q_i)$, and the norm $|.|$ is defined by $|\mathbf{p}| \equiv \sum_i p_i$ for vectors $\mathbf{p}$ and $\mathbf{q}$.

Code competition: A code competition process follows to select a F_2 node with the highest choice function value. The winner is indexed at J where

$$T_J = \max\{T_j : \text{for all } F_2 \, \text{node } j\}. \tag{2}$$

When a category choice is made at node J, $y_J = 1$; and $y_j = 0$ for all $j \neq J$ indicating a *winner-take-all* strategy.

Template matching: A template matching process checks if resonance occurs. It checks if the *match function* m_J of the chosen node J meets its vigilance criterion such that

$$m_J = \frac{|\mathbf{x} \wedge \mathbf{w}_J|}{|\mathbf{x}|} \geq \rho. \tag{3}$$

If the vigilance constraint is violated, a mismatch reset occurs and T_J is set to 0 for the duration of the input presentation. Another F_2 node J is selected using choice function and code competition until a resonance is achieved. If no selected node in F_2 meets the vigilance, an uncommitted node is recruited in F_2 as a new category node.

Template learning: Once a resonance occurs, the weight vector $\mathbf{w}_J$ is modified by the following learning rule:

$$\mathbf{w}_J^{(\text{new})} = (1 - \beta)\mathbf{w}_J^{(\text{old})} + \beta(\mathbf{x} \wedge \mathbf{w}_J^{(\text{old})}). \tag{4}$$

Activity readout: The chosen F_2 node J may perform a readout of its weight vectors to an input field F_1 such that $\mathbf{x}^* = \mathbf{w}_J$.

Using the described network dynamics, our memory model is able to generalize video representations and store memories incrementally at the category field F_2. For our framework, we use the mean pooled vector, fc_{mean} as inputs to the GVM model. The mean pool vector, fc_{mean} is normalized by dividing the vector by a scaling factor of M, subjected to a ceiling of one. The normalized mean pool vector $\mathbf{I_{norm}}$, is complement coded such that the input vector $\mathbf{x}$ is $[\mathbf{I_{norm}}, \overline{\mathbf{I_{norm}}}\ ]$ (8192 dimensional vector). The overall learning process is summarized in Algorithm 1. Each created category node weight vector $\mathbf{w_j}$ represents a new memory of a class video features. During learning, the closeness of which the category nodes are categorized are determined by the vigilance parameter ρ. The higher the vigilance parameter, the more specific the category nodes are learned.

Algorithm 1. Encoding of the Generalized Video Memory

```
Input: The normalized mean pooled vector
Present the normalized mean pooled vector to the *video* field at $F_1$
Perform code activation in the category field $F_2$    ▷ see (1)
repeat
    Perform code competition and template matching    ▷ see (2 & 3)
until resonance occurs    ▷ see (3)
Perform template learning    ▷ see (4)
```

To recall the generalized video memory, the vigilance parameter, ρ is set to zero. The recalling process is described in Algorithm 2 for retrieving the matching memory $\mathbf{x}^*$ from the input $\mathbf{x}$. The retrieved memory represents a generalized feature most similar to the input. With the use of complement coding and fuzzy AND operations, the memory node is able to represent the range values of the stored category [3]. The generalized memory is complement coded, such that $\mathbf{x}^*$ $=[w^*,\overline{w_c^*}]$, where w^* is the lower bound of the memory vector, and $1-\overline{w_c^*}$ is upper bound of memory vector. We averaged both the upper and lower bound memory vector and rescaled the vector back by M to form the *generalized memory vector*, $\mathbf{m}^*$. In this case, $\mathbf{m}^*$ has a dimension of 4096. The vector $\mathbf{m}^*$ is used as part of the input to the caption decoder as described in later sections.

Algorithm 2. Retrieval of the Generalized Video Memory

```
Input: The mean pooled vector to the *video* field
Perform code activation in the category field    ▷ see (1)
Select the winner code with highest choice value    ▷ see (2)
return The readout of the winner's memory $\mathbf{x}^*$
```

3.3 Caption Decoder

A standard RNN in principle is able to map a sequence of inputs to a sequence of outputs. For our work, a RNN is useful in mapping out a sequence of words to form the video description. Practically however, training a standard RNN with long-term dependency is difficult as it suffers from vanishing gradient problem [1]. The Long Short-Term Memory (LSTM) [14], a variant of RNN addresses the vanishing gradient problem with nonlinear gating units and memory cell c_t to maintain its state over time. As there are many variants of LSTM [11], to avoid confusion, we denote the exact LSTM equations used in our work. The LSTM unit used are described by Graves et al. [10]. The vector formulas for the LSTM unit are written as:

$$\begin{aligned}
i_t &= \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i)\\
f_t &= \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f)\\
o_t &= \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o)\\
c_t &= f_t \odot c_{t-1} + i_t \odot \phi(W_{cx}x_t + W_{ch}h_{t-1} + b_c)\\
h_t &= o_t \odot \phi(c_t)
\end{aligned} \tag{5}$$

where σ is the sigmoid function, ϕ is the hyperbolic tangent function and $\odot$ denotes pointwise multiplication of two vectors. The LSTM weight matrices are denoted byW_{ij}, and its biases bj. Each LSTM unit has three gates to compute the hidden state h_t. For an input of x_t at time step t, the input gate i_t controls how much of the input x_t is to be considered. The forget gate f_t controls how much to forget on the previous memory state c_{t-1}. The output gate o_t controls how much information in the memory state c_t is to be transferred to the hidden state h_t.

To have a direct comparison with the original mean-pool technique, we use the same two-layer LSTM designs as described in the mean pool approach [34]. A two-layer LSTM captures the structure of time series more naturally than a single-layer LSTM [13]. From the mean pool method, the mean pooled vector fc_{mean} is repeatedly presented as inputs to the LSTM caption decoder. In our work, during training, we present the *generalized memory vector*, $\mathbf{m}^*$ as the first input of the sequence, followed by the fc_{mean} vector repeatedly. This allows to prime the LSTM model to learn similar experiences from the generalized video memory vector, while learning the exact representation from the mean pooled vector. The LSTM model outputs the hidden state per time step. To predict a word from the hidden states, a word embedding is trained. The word embedding formula is represented as:

$$z_t = \phi(Wh_t + b) \tag{6}$$

where z_t is the predicted word output vector, W is the word embedding vector, b is the bias for the embedding vector, and h_t is the hidden states from the second LSTM layer. The words are represented as one-hot vectors y_t with a vector length D, where D is the size of the vocabulary. The sentence is prepended with a start-of-sentence (<S>) token, and appended with a end-of-sentence </S>) token. We use the softmax function to compute the probability distribution of the words w:

$$p(w|h_t) = \frac{\exp(W_w h_t + b)}{\sum_{w' \in D} \exp(W_{w'} h_t + b)} \tag{7}$$

During training, the LSTM model maximizes the log-likelihood objective function which is formulated as,

$$\max_{\theta} \sum_{t=1}^{T} \log \; p(y_t|z_t, y_{t-1}; \theta) \tag{8}$$

where θ denotes the model parameter which is to be optimized over the entire training data set.

During testing, the first input of the sequence to the LSTM model is the *generalized memory vector* $\mathbf{m}^*$, followed by the fc_{mean} vector repeatedly until the (</S>) token is emitted.

4 Experiment

4.1 Data Set

The YouTube2Text data set [5] contains 1,970 YouTube video clips annotated with multiple language descriptions. The descriptions are created by crowd sourcing using the Amazon's Mechanical Turk. We use only the English descriptions from the data set, which is about 80,000 video-sentence pairs. Each clip is usually less than 10 seconds long which depicts a main activity, accompanied with about 40 sentences. Following mean pool approach [34], we split the data set by randomly picking a training set of 1,200 videos, a testing set of 670 videos and a validation set of 100 videos.

4.2 Preprocessing

Video Preprocessing: We conducted frame sampling for one in every ten frames. We resized the sampled frames to 224×224 pixels, which is the input size for the VGG16 network. The video clips are zero padded to maintain the original aspect ratio.

Text Preprocessing: We tokenized the sentences, removed punctuations and converted the words to lower case. The vocabulary size is about 5,000 after removal of rare words that appeared less than four times. Due to batch training, sentences are padded with </S>) tokens to align with the longest length of the word sequence of each batch.

4.3 Training Details

For the visual encoder, we fix the weights of the VGG16 network to reduce computation work load. For each video, the outputs vectors of the VGG16 fc2 layer are mean pooled to form the mean pooled vector.

To learn the generalized video memory vector, the mean pooled vectors are used for memory encoding. The scaling factor M for normalization is set to 64.0. One shot learning is enabled by setting the GVM's learning rate β to 1.0. The choice parameter, α is set to 0.001. With the vigilance parameter ρ set at 0.99, GVM learns a total of 576 categories after learning training set.

The training of the caption decoder proceeds after learning the GVM's memories. The two-layer LSTM caption decoder has 1,000 hidden units for each layer. To avoid over-fitting, a dropout of 0.5 is used on both the inputs and the outputs of both LSTM layers. Training of the caption decoder with GVM model is described in Sect. 3.3. We stopped training the LSTM caption decoder when the validation loss does not improve. For our baseline comparison, we have also replicated the results using the mean pool method with VGG16 video encoder with our data set splits.

4.4 Evaluation Metrics

We use two model-free evaluation metrics, BLEU [20] and METEOR [7] to evaluate the results against the ground truth sentences. The two metrics are chosen as most prior work with YouTube2Text data set report their results with BLEU and METEOR scores, therefore a direct comparison can be done. Both BLEU and METEOR are typically used for evaluating machine translation and image captioning tasks. Generally, the higher the scores, the better the correlation of the predicted descriptions are against human judgement. We employed the codes from the Microsoft COCO Caption Evaluation Server [6] to obtain both BLEU and METEOR scores.

5 Experimental Results

The evaluation metric scores are shown in Table 1. While other methods may have trained with more image/video captioning data set, we only compare the evaluation results that are trained purely on the YouTube2Text data set, with the use of pre-trained visual encoder. We report the scores of the compared algorithms along with the type of visual encoder as the evaluation scores can differ by employing a different visual encoder. The BLEU scores for both the mean pool (VGGNet) and the S2VT methods are omitted as they are not reported in the original work. The results are shown in Table 1.

5.1 Compared Algorithms

The following describes briefly on the compared algorithms:

- **Factor Graph Model (FGM)** [32]. FGM first employs vision recognizers to obtain the subject, object, activity and place (SOVP) elements. The Factor

Table 1. Evaluation results based on the YouTube2Text data set, with compared methods with its visual encoder. (*) represents our replicated results.

Model	BLEU@4	METEOR
FGM	13.68	23.9
Mean pool (AlexNet)	31.2	26.9
Mean pool (VGGNet)	-	27.7
Our Experiments		
Mean pool (VGGNet) *	42.5	27.6
GVM (VGGNet)	**42.5**	**28.1**
Recent Methods		
S2VT (VGGNet) (AlexNet)	-	29.8
LSTM-E (VGGNet) (C3D)	45.3	31.0
HRNE-Attention (GoogLeNet) (C3D)	46.7	33.9

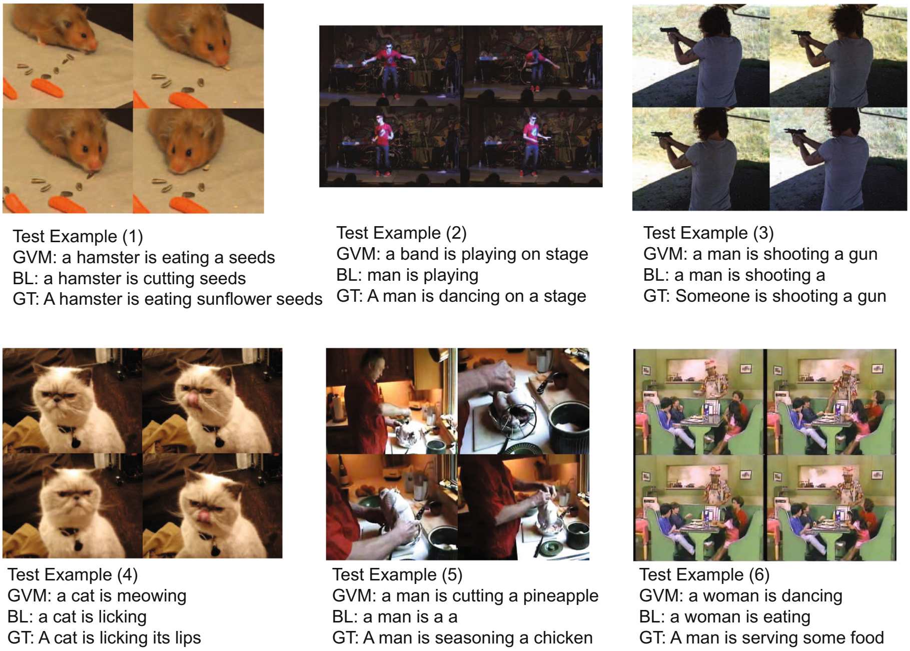
Graph Model further refines the co-occurring SOVP elements by maximum a posteriori (MAP) estimation. A template is used for sentence generation based on the refined SOVP elements.
- **Mean Pool (AlexNet)** [34]. The visual features are extracted using AlexNet. The features are averaged (mean pooled) across the frames of the entire video. The mean pooled feature is used as the input to the LSTM caption decoder continuously until a end-of-sentence token is omitted.
- **Mean Pool (VGG)** [33]. Similar to the above, with the exception of utilising the VGGNet visual encoder.
- **S2VT (VGG & AlexNet)** [33]. The VGGNet is used to extract RGB features, and the AlexNet for to extract optical flow features. Both RGB and flow features are presented as the input to an encoder-decoder model of LSTM. The first encoding phase processes the sequences of the visual features. The second decoding phase generates the captions until a end-of-sentence token is omitted.
- **LSTM-E (VGG & C3D)** [19]. Visual-semantic embeddings using LSTM are used to maximize the probability of next word given the previous word and visual content. A joint learning of relevance and coherence objective functions are utilized to minimize losses between the visual and textual content.
- **HRNE-Attention (GoogLeNet & C3D)** [18]. The Hierarchical Recurrent Neural Encoder (HRNE) exploits the temporal structure of the video. The 2-layer hierarchical LSTM structure is analogical to a convolutional network. Information flows to the next layer by a fixed time step. Along with attention mechanism, HRNE decodes the captions from the video feature sequences.

As the data set split of the training, testing and validation set is randomly picked, it may affect the metric scores when compared to other work due to the data set differences. For a fair comparison, we also replicated the result of the mean pool (VGG16) approach using our data set splits.

5.2 Analysis

With training done on the YouTube2Text data set alone, our framework with GVM model is able to achieve a METEOR score of 28.1. This is better than the reported baseline method Mean pool (VGG) at 27.7 and our replicated result at 27.6. It should be noted that these results are trained only with the YouTube2Text data set. The result shows that the GVM model is able to enhance the video captioning quality. While the magnitude of the evaluation scores may not be intuitive to interpret, by comparing with the latest algorithms that combines multiple visual features, we can gauge that minor differences in score do make significant contribution to the quality of captions.

When inspecting the test output, there are a number of sentences that do not contain the verbs of the activities. We attribute this issue to the mean pooling method as it does not fully capture the order of visual events which is important for activity recognition.



Test Example (1)
GVM: a hamster is eating a seeds
BL: a hamster is cutting seeds
GT: A hamster is eating sunflower seeds

Test Example (2)
GVM: a band is playing on stage
BL: man is playing
GT: A man is dancing on a stage

Test Example (3)
GVM: a man is shooting a gun
BL: a man is shooting a
GT: Someone is shooting a gun

Test Example (4)
GVM: a cat is meowing
BL: a cat is licking
GT: A cat is licking its lips

Test Example (5)
GVM: a man is cutting a pineapple
BL: a man is a a
GT: A man is seasoning a chicken

Test Example (6)
GVM: a woman is dancing
BL: a woman is eating
GT: A man is serving some food

Fig. 3. Example 1 to 3 (top left to right) and Example 4 to 6 (bottom left to right) with screenshots of the video against text generated by the Generalized Video Memory Network (GVM); the replicated mean pool baseline (BL); and the ground truth (GT).

The one drawback of the ART based network is that the category nodes learned are dependent on the order of which the inputs are presented. This issue is mitigated by shuffling the order of the training data during the encoding stage of the GVM network.

5.3 Test Examples

Figure 3 shows some examples of the generated captions with our framework using GVM, the replicated mean pool method and the ground truth. Interestingly for test example 2, while the focus is on the dancing man, GVM is able to pick up the background activities depicting a group of musicians performing on the stage. Our current framework do not have the attention mechanism which may improve the focus of the subjects in the video clips. For the test example 4 and 5, while our GVM model captions are invalid, it generates related captions that are similar to the video scene. The invalid output suggests a down side of over-generalization of the video memory.

6 Conclusion

We have proposed a memory framework GVM for video captioning. Our experiments have shown that GVM is able to enhance on the BLEU and METEOR scores using a similar baseline design based on mean pooling. We have demonstrated the potential of enhancing the accuracies of a deep learning model using memory modules based on ART. We believe by integrating the concepts investigated in this paper to the latest state-of-the-art video captioning architecture, we can further enhance the scores as well. In future work, we will like to introduce GVM into the latest state-of-the-art methods and expand the use of multi-channel input fields to represent multi-modal memories.

Acknowledgments. This research is supported in part by a research grant (DSOCL16006) from DSO National Laboratories, Singapore and a joint project funded by ICT Virtual Organization of ASEAN Institutes and NICT (ASEAN IVO).

References

1. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. **5**(2), 157–166 (1994)
2. Carpenter, G.A., Grossberg, S.: Adaptive Resonance Theory. In: Arbib, M.A. (ed.) The Handbook of Brain Theory and Neural Networks, pp. 87–90. MIT Press, Cambridge (2003)
3. Carpenter, G.A., Grossberg, S., Rosen, D.B.: Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Netw. **4**(6), 759–771 (1991)
4. Chang, P.-H., Tan, A.-H.: Encoding and recall of spatio-temporal episodic memory in real time. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 1490–1496 (2017)
5. Chen, D.L., Dolan, W.B.: Collecting highly parallel data for paraphrase evaluation. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 190–200 (2011)
6. Chen, X., et al.: Microsoft COCO captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325 (2015)
7. Denkowski, M., Lavie, A.: Meteor universal: language specific translation evaluation for any target language. In: Proceedings of the ninth workshop on Statistical Machine Translation, pp. 376–380 (2014)
8. Donahue, J., et al.: DeCAF: a deep convolutional activation feature for generic visual recognition. In: International Conference on Machine Learning, pp. 647–655 (2014)
9. Fakoor, R., Mohamed, A., Mitchell, M., Kang, S.B., Kohli, P.: Memory-augmented attention modelling for videos. arXiv preprint arXiv:1611.02261 (2016)
10. Graves, A., Mohamed, A., Hinton, G.: Speech recognition with deep recurrent neural networks. In: Proceedings of the IEEE international conference on Acoustics, Speech and Signal Processing, pp. 6645–6649 (2013)
11. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: a search space odyssey. IEEE Trans. Neural Netw. Learn. Syst. **28**(10), 2222–2232 (2017)
12. Guadarrama, S., et al.: YouTube2Text: recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2712–2719 (2013)
13. Hermans, M., Schrauwen, B.: Training and analysing deep recurrent neural networks. In: Advances in Neural Information Processing Systems, pp. 190–198 (2013)

14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
15. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3128–3137 (2015)
16. Krishnamoorthy, N., Malkarnenkar, G., Mooney, R.J., Saenko, K., Guadarrama, S.: Generating natural-language video descriptions using text-mined knowledge. In: Association for the Advancement of Artificial Intelligence, vol. 1, p. 2 (2013)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
18. Pan, P., Xu, Z., Yang, Y., Wu, F., Zhuang, Y.: Hierarchical recurrent neural encoder for video representation with application to captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1029–1038 (2016)
19. Pan, Y., Mei, T., Yao, T., Li, H., Rui, Y.: Jointly modeling embedding and translation to bridge video and language. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4594–4602 (2016)
20. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318 (2002)
21. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proceedings of the International Conference on Learning Representations (2015)
23. Subagdja, B., Tan, A.-H.: Neural modeling of sequential inferences and learning over episodic memory. Neurocomputing **161**, 229–242 (2015)
24. Sukhbaatar, S., Weston, J., Fergus, R., et al.: End-to-end memory networks. In: Advances in Neural Information Processing Systems, pp. 2440–2448 (2015)
25. Sundermeyer, M., Schlüter, R., Ney, H.: LSTM neural networks for language modeling. In: Conference of the International Speech Communication Association (2012)
26. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
27. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
28. Tan, A.-H.: Falcon: a fusion architecture for learning, cognition, and navigation. In: Proceedings of the IEEE International Joint Conference on Neural Network, vol. 4, pp. 3297–3302. IEEE (2004)
29. Tan, A.-H.: Direct code access in self-organizing neural networks for reinforcement learning. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 1071–1076 (2007)
30. Tan, A.-H., Carpenter, G.A., Grossberg, S.: Intelligence through interaction: towards a unified theory for learning. In: Liu, D., Fei, S., Hou, Z.-G., Zhang, H., Sun, C. (eds.) ISNN 2007. LNCS, vol. 4491, pp. 1094–1103. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72383-7_128
31. Teng, T.H., Tan, A.-H., Zurada, J.M.: Self-organizing neural networks integrating domain knowledge and reinforcement learning. IEEE Trans. Neural Netw. Learn. Syst. **26**(5), 889–902 (2015)

32. Thomason, J., Venugopalan, S., Guadarrama, S., Saenko, K., Mooney, R.: Integrating language and vision to generate natural language descriptions of videos in the wild. In: Proceedings of the International Conference on Computational Linguistics, pp. 1218–1227 (2014)
33. Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., Saenko, K.: Sequence to sequence-video to text. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4534–4542 (2015)
34. Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., Saenko, K.: Translating videos to natural language using deep recurrent neural networks. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1494–1504 (2015)
35. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: a neural image caption generator. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3156–3164 (2015)
36. Wang, P., Zhou, W.J., Wang, D., Tan, A.-H.: Probabilistic guided exploration for reinforcement learning in self-organizing neural networks. In: Proceedings of International Conference on Agents, pp. 109–112 (2018)
37. Wang, W., Subagdja, B., Tan, A.-H., Starzyk, J.A.: A self-organizing approach to episodic memory modeling. In: Proceedings of the IEEE International Joint Conference on Neural Networks, pp. 1–8. IEEE (2010)
38. Wang, W., Subagdja, B., Tan, A.-H., Starzyk, J.A.: Neural modeling of episodic memory: encoding, retrieval, and forgetting. IEEE Trans. Neural Netw. Learn. Syst. **23**(10), 1574–1586 (2012)
39. Weston, J., et al.: Towards ai-complete question answering: a set of prerequisite toy tasks. arXiv preprint arXiv:1502.05698 (2015)
40. Weston, J., Chopra, S., Bordes, A.: Memory networks. In: Proceedings of the International Conference on Learning Representations (2015)
41. Yu, H., Wang, J., Huang, Z., Yang, Y., Xu, W.: Video paragraph captioning using hierarchical recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4584–4593 (2016)