

MEMORY-AUGMENTED ATTENTION MODELLING FOR VIDEOS

Rasool Fakoor[†], Abdel-rahman Mohamed,^{††}, Margaret Mitchell^{‡‡}, Sing Bing Kang^{‡‡}, Pushmeet Kohli^{††}

^{††}Microsoft Research [†]University of Texas at Arlington ^{‡‡}Google

[†]rasool.fakoor@mavs.uta.edu, ^{††}{asamir, singbing.kang, pkohli}@microsoft.com ^{‡‡}mmitchellai@google.com,

ABSTRACT

Recent works on neural architectures have demonstrated the utility of attention mechanisms for a wide variety of tasks. Attention models used for problems such as image captioning typically depend on the image under consideration, as well as the previous sequence of words that come before the word currently being generated. While these types of models have produced impressive results, they are not able to model the higher-order interactions involved in problems such as video description/captioning, where the relationship between parts of the video and the concepts being depicted is complex. Motivated by these observations, we propose a novel memory-based attention model for video description. Our model utilizes memories of past attention when reasoning about where to attend to in the current time step, similar to the *central executive* system proposed in human cognition (Baddeley & Hitch, 1974). This allows the model to not only reason about local attention more effectively, it allows it to consider the entire sequence of video frames while generating each word. Evaluation on the challenging and popular MSVD and Charades datasets show that the proposed architecture outperforms all previously proposed methods and leads to a new state of the art results in the video description.

1 INTRODUCTION

Deep neural architectures have led to remarkable progress in computer vision and natural language understanding problems. Image captioning is one such application that has seen the combination of convolutional structures (Krizhevsky et al., 2012; LeCun et al., 1998), which have been shown to be very effective for problems like object detection, with sequential recurrent structures, which have been shown to be very effective for problems like machine translation (Sutskever et al., 2014). One of the key modelling paradigms shared by most models for image captioning is the notion of an attention mechanism that guide the model to attend to certain parts of the image while generating.

The attention models used for problems such as image captioning typically depend on the single image under consideration and the partial output generated so far, jointly capturing one region of an image and the words being generated. However, such models cannot capture the temporal reasoning necessary to effectively produce words that refer to actions and events taking place over multiple frames in a video. For example, in a video depicting “someone waving a hand”, the “waving” action can start from any frame and can continue on for a variable number of following frames. More importantly, it is likely in a given video quite few frames do not contain any useful information or motion in regard to a given task. Given this, it is not surprising that even with recent advancements in image captioning Xu et al. (2015a); Johnson et al. (2016); Vinyals et al. (2015), video captioning has remained challenging.

Motivated by these observations, we introduce a memory-based attention mechanism for video captioning and description. Our model utilizes memories of past attention in the video when reasoning about where to attend to in a current time step. This allows the model to not only effectively leverage local attention, but also to consider the entire video as it generates each word. This mechanism is similar to the proposed *central executive* system in human cognition, which is thought to permit human performance on two simultaneous tasks (e.g., seeing and saying) using two separate percep-

tual domains (e.g., visual and linguistic) by binding information from both sources into coherent structure that enables coordination, selective attention, and inhibition.

Our work shares the same goals as recent work on attention mechanisms for sequence-to-sequence architectures, such as Rocktäschel et al. (2016) and Yang et al. (2016). However, there are major differences between this work and our current work. Rocktäschel et al. (2016) considers the domain of entailment relations, where the goal is to determine entailment given two input sentences. They propose a soft attention model that is not only focused on the current state, but the previous as well. In our model, we explicitly store all previous attention into memory. In addition, our memory memorizes the encoded version of the input videos conditioned on previously seen words. Even though Yang et al. (2016) and our work both try to solve the problem of locality of attention, our work is different from them in how the memory architecture is modelled. More specifically, they incorporate discriminative supervision into their "reviewer" mechanism, which is not the case in our model. Further, their model is applied to image caption generation, which is to some extent simpler than video caption generation because there is no temporal structure to model.

We apply our model on the video captioning problem and evaluate it on the MSVD (Chen & Dolan, 2011) and the Charades (Sigurdsson et al., 2016) datasets. Experimental results show that our proposed architecture outperforms all previous methods and leads to new state of the art results. While we have chosen the video captioning problem for our experiment, the model is general enough that it can be applied to other problems where attention models are used.

2 RELATED WORK

One of the primary challenges in learning a mapping from a visual space (i.e., video or image) to a language space is learning a representation that not only effectively represents each of these modalities, but is also able to translate a representation from one space to the other. Rohrbach et al. (2013) developed a model that generates a semantic representation of visual content that can be used as the source language for the language generation module. Venugopalan et al. (2015b) proposed a deep method to translate a video into a sentence where an entire video is represented with a single vector based on the mean pool of frame features. However, representing a video by an average of its frames misses the temporal structure of the video. To address this problem, recent work (Yao et al., 2015; Pan et al., 2016a; Venugopalan et al., 2015a; Andrew Shin, 2016; Pan et al., 2016b; Xu et al., 2015b; Ballas et al., 2016; Yu et al., 2016) proposed methods to model temporal structure of video as well as language.

The majority of these methods are inspired by sequence to sequence (Sutskever et al., 2014) and attention (Bahdanau et al., 2015) models. Sequence learning (Sutskever et al., 2014) was originally proposed to map the input sequence of a source language to a target language. Even though applying this method with a combination of attention to the problem of translating a video to a description shows promising results, there are some shortcomings. First of all, modelling the video content with a fixed-length vector in order to map it to a language space is a much harder problem than mapping from a language to a language given the complexity of visual content. Since not all frames in a video are equally salient for a short description, and an event can happen in multiple frames, it is important for a model to identify which frames are most salient. Further, the model should be able to focus on points of interest within these frames to select what to talk about. Even using a variable-length vector to represent a video using attention (Yao et al., 2015) can have some problems. More specifically, current attention methods are local Yang et al. (2016), since the attention mechanism works in a sequential structure, and lacks the ability to capture global structure. Moreover, combining a video and a language description as a sequence to sequence is usually done by some variant of a recurrent neural network (RNN) (Hochreiter & Schmidhuber, 1997). Given the limited capacity of a recurrent network to model very long sequences, memory networks (Weston et al., 2014; Sukhbaatar et al., 2015) have been introduced to help the RNN memorize sequences. However, one problem these memory networks suffer from is the difficulty in training the model. The model in Weston et al. (2014) requires supervision at each layer which makes training with backpropagation a challenging task. Even though Sukhbaatar et al. (2015) proposed a memory network that can be trained end-to-end, working with memory is still a challenging problem in deep learning especially with write operation (Graves et al., 2014).

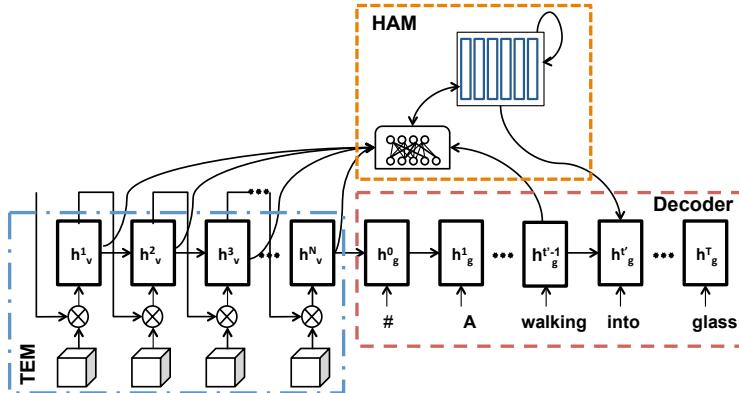


Figure 1: Our proposed architecture. Each component of our model is described in 3.1 through 3.3

To address these problems, we propose a memory-based attention sequence-to-sequence model that not only can learn hierarchical attention relationships, but provides a simple and effective memory structure. In the next section, we explain our model in more detail.

3 LEARNING TO ATTEND AND MEMORIZ

Our goal is to design an architecture that learns where to look and what to look for in a video, in order to talk about it in the description. To achieve this goal, we formulate the problem as sequence learning to maximize the probability of generating a correct description given a video:

$$\Theta^* = \arg \max_{\Theta} \sum_{(S, f_1, f_2, \dots, f_N)} \log p(S|f_1, f_2, \dots, f_N; \Theta) \quad (1)$$

where S is the description, f_1, f_2, \dots, f_N are the input video frames, and Θ is the model parameter vector. The main modelling challenges in video description are to develop a system that can model the temporal structure of the video, learn to attend to the important parts of a video, how to memorize the video that was described given all the generated words so far, then generate a new word by looking at the entire video. To address these issues, we propose an end-to-end network that has three components (Figure 1): Temporal Model (TEM), Hierarchical Attention/Memory (HAM), and the Decoder. The goal in TEM is to capture the temporal structure and track motion in a video. The HAM component acts as a hierarchical attention or memory between an input video and the description. More specifically, the HAM learns a hierarchical attention structure that learns where to attend in video given all previously generated words and previous states. The HAM can be interpreted as a memory structure as well, where it learns to memorize an encoded version of a video with language. HAM provides the decoder with the ability to look at an entire video plus all the previously generated words before generating any new words. This is important because a single action normally exhibits multiple frames in the input video. By employing the HAM, the model can effectively model the action over these frames. One of the main contributions of this work is to use a global state to generate any new word. This global state aggregates information from previously generated words and all input frames. We will first describe each component of our model, then explain details of training and inference.

3.1 TEMPORAL MODELER (TEM)

One important question is how to encode the temporal structure of the input video for caption generation. Recently, it has been shown that Recurrent Neural Networks (RNN) has the ability to model the temporal structure in sequential data such as video (Ballas et al., 2016; Sharma et al., 2015; Venugopalan et al., 2015a) and speech (Graves & Jaitly, 2014). Moreover, since frame-to-frame temporal variation tend to be local (Brox & Malik, 2011) and critical in the motion modeling (Ballas et al., 2016), it is important to consider a frame representation that can preserve frame-to-frame temporal variation. Even though using features extracted from the fully connected layers of Convolutional Neural Networks (CNNs) have shown state of the art results in image classification and

recognition (Simonyan & Zisserman, 2014; He et al., 2016), these features tend to discard the low level information useful in modeling the motion in the video (Ballas et al., 2016).

To address the temporal modeling and video representation problems, we use an RNN to model the temporal structure of the video where at each time step, a frame encoding with size of R^D is used as an input to the RNN. Instead of extracting features from a top layer of the pretrained CNN, intermediate convolutional maps have been extracted for the video frames. Specifically, for a given video, X , with N frames $X = [X^1, X^2, \dots, X^N]$, N convolutional maps of size $R^{L \times D}$ are extracted where D is dimension of a feature corresponding to L locations in the input frame.

In order to let the network selectively focus on these L locations of each frame given the hidden state of RNN, we apply a soft attention model Bahdanau et al. (2015); Xu et al. (2015a); Sharma et al. (2015), called “Location Attention (**Latt**)”. More specifically, by using a softmax, each hidden state produces L probabilities to specify which part of the input is more important, and then creates an input map for the RNN using these probabilities. The f_{Latt} is defined as follow:

$$\rho_j^t = \frac{\exp((h_v^{t-1})^T W_p^j)}{\sum_{k=1}^L \exp((h_v^{t-1})^T W_p^k)} \quad (2)$$

$$F^t = \sum_{j=1}^L \rho_j^t X_j^t \quad (3)$$

where $h_v^{t-1} \in R^K$ is hidden state of RNN at $t-1$, $W_p \in R^{K \times L}$, and $F^t \in R^D$. At each time step, TEM learns a vector representation for each frame, looking at the frame convolution map, and applying the location attention on this map conditioned on all previously seen frames.

$$F^t = f_{\text{Latt}}(X_t, h_v^{t-1}; W_p) \quad (4)$$

$$h_v^t = f_v(F^t, h_v^{t-1}; \Theta_v) \quad (5)$$

where f_v can be a vanilla RNN, LSTM, or GRU and Θ_v is the parameters of the f_v . Due to the fact that vanilla RNNs have gradient vanishing and exploding problems (Pascanu et al., 2013), we use gradient clipping to deal with gradient exploding, and an LSTM with the following flow to deal with the gradient vanishing problem:

$$\begin{aligned} i^t &= \sigma(F^t W_{xi} + h_v^{t-1} W_{hi}) \\ f^t &= \sigma(F^t W_{xf} + h_v^{t-1} W_{hf}) \\ o^t &= \sigma(F^t W_{xo} + h_v^{t-1} W_{ho}) \\ g^t &= \tanh(F^t W_{xg} + h_v^{t-1} W_{hg}) \\ c_v^t &= f^t \odot c_v^{t-1} + i^t \odot g^t \\ h_v^t &= o_t \odot \tanh(c^t) \end{aligned}$$

where $W_{h*} \in R^{K \times K}$, $W_{x*} \in R^{D \times K}$, and we define $\Theta_v = \{W_{h*}, W_{x*}\}$.

3.2 HIERARCHICAL ATTENTION/MEMORY (HAM)

One problem with using sequence-to-sequence style architectures (Sutskever et al., 2014), to model a task such as video language description, is how to find a mapping from a video space to a language space that can capture the relationship between a word and video, or more specifically, the connection between an entire video and an entire sentence where there might not be a clear alignment between the two sequences, as opposed to machine translation and speech recognition. Furthermore, the model should be able to identify which part of the video is more relevant to the description because captions normally focus on a tiny fraction of the facts present in the video. More importantly, once the model starts generating the description, it should still be reminded with the video frames to generate meaningful descriptions. In order to address these problems, we propose a memory-based attention that encodes a video into memory, built as a function of the state of the language generation network (a.k.a. Decoder) and the state of the TEM network. More specifically, our Hierarchical Attention/Memory can be formulated as two following steps:

- Attention update [$\hat{\mathbf{F}}(\Theta_a)$]:

$$\mathbf{Q}_A = \tanh(\mathbf{H}_v \mathbf{W}_v + \mathbf{H}_g^{t'-1} \mathbf{W}_g + \mathbf{H}_m^{t'-1} \mathbf{W}_m) \quad (6)$$

$$\alpha_{t'} = \text{softmax}(\mathbf{U}^T \mathbf{Q}_A) \quad (7)$$

$$\hat{\mathbf{F}} = \alpha_{t'}^T \mathbf{H}_v \quad (8)$$

- Memory update:

$$\mathbf{h}_m^{t'} = f_m(\mathbf{h}_m^{t'-1}, \hat{\mathbf{F}}; \Theta_m) \quad (9)$$

where $H_v \in R^{N \times K}$, W_v and W_g are $\in R^{K \times K}$, $U \in R^K$, $W_m \in R^{M \times K}$, and $\Theta_a = \{W_v, W_g, W_m, U\}$. N is number of frames in a given video, $H_v = [h_v^1, \dots, h_v^N]$, $H_g^{t'-1} = [h_g^{t'-1}, \dots, h_g^{t'-1}]$, and $H_m^{t'-1} = [h_m^{t'-1}, \dots, h_m^{t'-1}]$. $\alpha_{t'}$ is the set of probabilities in a given time step that specifies the attention over an input video state (H_v), memory state (H_m), and decoder state (H_g). In order to let the network remember what has been attended before and the temporal structure of a video, we propose f_m to memorize the previous attention and encoded version of an input video with language model. Using f_m not only enables the network to memorize previous attention and frames, but also to learn multi-layer attention over an input video and corresponding language. The output of the memory-attention is then used as input to the Decoder.

3.3 DECODER

In order to generate a new word conditioned on all previous words and HAM states, a recurrent structure is modelled as follows:

$$\mathbf{h}_g^{t'} = f_g(s^{t'}, \mathbf{h}_m^{t'}, \mathbf{h}_g^{t'-1}; \Theta_g) \quad (10)$$

$$\hat{s}_{t'} = \text{softmax}(\mathbf{W}_e \mathbf{h}_g^{t'}) \quad (11)$$

where $s^{t'}$ is a word vector at t' , $\mathbf{W}_e \in R^{K \times C}$ and C is the vocabulary size. In addition, $\hat{s}_{t'}$ assigns a probability to each word in the language. We use LSTMs for both f_m and f_g .

3.4 TRAINING AND OPTIMIZATION

The goal in our network is to predict the next word given all previously seen words and an input video. In order to optimize our network parameters, $\Theta = \{W_p, \Theta_v, \Theta_a, \Theta_m, \mathbf{W}_e\}$, we minimize a negative log likelihood loss function, formulated as follow:

$$L(\mathbf{S}, \mathbf{X}; \Theta) = - \sum_j^T \sum_i^{|V|} s_{j,i} \log(\hat{s}_{j,i}) + \lambda \|\Theta\|_2^2 \quad (12)$$

where $|V|$ is the dictionary size. We fully train our network in an *end-to-end* fashion using first-order stochastic gradient-based optimization method with an adaptive learning rate. More specifically, in order to optimize our network parameters, we use Adam Kingma & Ba (2015) with learning rate 2×10^{-5} and set β_1, β_2 to 0.8 and 0.999, respectively. At the training, we use a batch size of 16.

4 EXPERIMENTS

DATASET We evaluate our proposed model on the *Charades* (Sigurdsson et al., 2016) dataset and the Microsoft Video Description Corpus (MSVD) (Chen & Dolan, 2011). *Charades* contains 9,848 videos (in total) and provides 27,847¹ video descriptions, with 7569 training, 1,863 test, 400 for the validation. We follow the same split (i.e. training and test splits) as Sigurdsson et al. (2016). It is worth noting that one major difference between this dataset and others is that they use a “Hollywood in Homes” approach to collecting the data (Sigurdsson et al., 2016), where “actors” are crowdsourced, yielding a diverse scene and actor videos. One reason that we report results on this

¹Only 16087 out of 27,847 are used as captions for our evaluation since the 27,847 refers to script of the video as well as captions.

dataset is because each video has a specific action in it and would be a suitable testbed to evaluate our model.

MSVD is a set of Youtube videos that are annotated by a Mechanical Turker,² who was asked to pick a clip from a video that represents an activity. In this dataset, each clip is annotated by multiple workers with a single sentence. This dataset contains 1,970 videos and about 80,000 descriptions, where 1,200 of the videos are training data, 670 are test data, and the rest (i.e., 100 videos) are assigned for validation. In order to make the results comparable with other papers, we follow the exact training/validation/test split provided by Venugopalan et al. (2015b).

EVALUATION METRICS Below, we report results on the video caption generation task. In order to evaluate captions generated by our model, we use model-free automatic evaluation metrics. We adopt *METEOR*, *BLEU@N*, and *CIDEr* metrics available from the Microsoft COCO Caption Evaluation code³ to score the system.

VIDEO AND CAPTION PREPROCESSING We preprocess the captions for both datasets using the Natural Language Toolkit (NLTK)⁴. Beyond this, no other type of preprocessing is used.

We extract sample frames for each video and pass each frame through VGGnet (Simonyan & Zisserman, 2014) without any fine-tuning. For the experiments in this paper, we use the feature maps from *conv5_3* layer after applying *ReLU*. The feature map in this layer is $14 \times 14 \times 512$. Our TEM component operates on the flattened 196×512 of this feature cubes. For the ablation studies, features from fully connected layer are used as well where the features in this layer have 4096 dimension.

HYPER-PARAMETER OPTIMIZATION We use random search (Bergstra & Bengio, 2012) on validation set to select hyper-parameters on both datasets. The word-embedding size, hidden layer size (for both TEM and Decoder), and memory size of the best model on Charades are: 237, 1316, and 437, respectively. These values are 402, 1479, and 797 for the model on MSVD dataset. A stack of two LSTMs are used in the Decoder and TEM.

4.1 VIDEO CAPTION GENERATION

We first present an ablation analysis to elucidate the contribution of the different components of our proposed model. Next, we compare the overall performance of our model on video caption generation task to other models.

ABALATION ANALYSIS

We first perform a series of ablation studies in order to show the contributions of the different components of our model. Specifically, we show that the importance of each components in our model in caption generation task on MSVD dataset. One ablation (denoted as **Att + No TEM**) corresponds to a simpler version of our model in which we remove the TEM component and instead we pass each frame of a video through a CNN and extract features from the last fully-connected hidden layer (e.g., *fc7*). In addition, we replace our HAM component with a simpler version where the model only memorizes the current step instead of all previous steps. In another ablation (denoted as **No HAM + TEM**), we remove the HAM component from our model and keep the rest of our model as it is. In the next variation (denoted as **HAM + No TEM**), we remove the TEM component and calculate features for each frame, similar to **Att + No TEM**. Finally, the last row in the table is our proposed model (denoted **HAM + TEM**) with all its components.

Table 1 reports the result of this study. In this experiment, we sample 40 frames per video and use them as the inputs to a network. As the results show, HAM plays a critical role in our proposed model, and removing it causes a drop in performance. On the other hand, removing TEM by itself does not drop performance as much as dropping the HAM. When we put the two together, they complement one another, resulting in better performance.

²<https://www.mturk.com/mturk/welcome>

³<https://github.com/tylin/coco-caption>

⁴<http://www.nltk.org/>

Table 1: Ablation of our model with and without the HAM component on the test set of 670 videos

Method	METEOR	BLEU@1	BLEU@2	BLEU@3	BLEU@4	CIDEr
Att + No TEM	31.20	77.90	65.10	55.3	44.90	63.90
No HAM + TEM	30.5	78.10	65.20	55.10	44.60	60.50
HAM + No TEM	31.0	78.70	66.90	57.40	47.0	62.10
HAM + TEM	31.70	79.0	66.20	56.0	45.6	62.20

PERFORMANCE COMPARISON

Next, to extensively evaluate our model, we compare our model with state-of-the-art models and baselines for the video caption generation task on the MSVD dataset. In this experiment, we use 8 frames per video as the inputs to the TEM module. Table 2⁵ shows the results for this experiment. As the results show, our model gets state-of-the-art scores either in BLEU-4 or METEOR, compared to other methods. This is particularly noteworthy because we do not use external features for the video, such as Optical Flow (Brox et al., 2004) (denoted as Flow in table), 3-Dimensional Convolutional Network features (Tran et al., 2015) (denoted as C3D), or fine-tuned CNN features (denoted as FT) on the action recognition task with dataset such as UCF-101. The only exception happens when we compare our model with (Yu et al., 2016), who uses C3D features. In this method, adding C3D features leads to a huge improvement in their results (compare row 4 with 11 in Table 2). On the other hand, our method without using any external features can achieve better results in comparison with all other methods. This is important because our proposed architecture can alone not only learn a representation for video that can model the temporal structure of a video sequence, but also a representation that can effectively map visual space to the language space.

Table 2: Video captioning evaluation on the test set of 670 videos in MSVD.

Method	METEOR	BLEU@1	BLEU@2	BLEU@3	BLEU@4	CIDEr
Venugopalan et al. (2015b)	27.7	—	—	—	—	—
Venugopalan et al. (2015a)	29.2	—	—	—	—	—
Pan et al. (2016b)	29.5	74.9	60.9	50.6	40.2	—
Yu et al. (2016)	31.10	77.30	64.50	54.60	44.30	—
Pan et al. (2016a)	33.10	79.20	66.30	55.10	43.80	—
Our Model	31.80	79.40	67.10	56.80	46.10	62.70
Yao et al. (2015) + C3D	29.60	—	—	—	41.92	51.67
Venugopalan et al. (2015a) + Flow	29.8	—	—	—	—	—
Ballas et al. (2016) + FT	30.75	—	—	—	49.0	59.37
Pan et al. (2016b) + C3D	31.0	78.80	66.0	55.4	45.3	—
Yu et al. (2016) + C3D	32.60	81.50	70.40	60.4	49.90	—

In addition, we report results on the Charades dataset for video caption generation. This dataset is challenging because only a few captions per video (about 2 per video) are available. In this experiment, we use 16 frames per video as the inputs to the TEM module. Table 3 shows the performance of our method on this dataset. Our method can achieve 10% improvement over Venugopalan et al. (2015a) in the caption generation task. It is worth noting that a human can only achieve a score of 24 in METEOR for this dataset, which illustrated the level of difficulty in this dataset.

QUALITATIVE RESULTS

We show some captions generated by our model in 2. The model mostly generates correct captions for cases where content and ground truth captions are consistent. There are some cases in which our model makes some mistakes. For example, in a ‘a dog is on a trampoline’ video, our model generated “a man is washing a bath” as a caption. This is interesting because the ‘man’ object only appears in a few frames (1 or 2), but our model can still recognize the man object in the video.

⁵— in the Table 2 means, that score was not reported by the corresponding paper.

Table 3: Video captioning evaluation on the test set of 1863 videos in Charades.

Method	METEOR	BLEU@1	BLEU@2	BLEU@3	BLEU@4	CIDEr
Human(Sigurdsson et al., 2016)	24	62	43	29	20	53
Sigurdsson et al. (2016)	16	49	30	18	11	14
Our Model	17.6	50	31.1	18.8	11.5	16.7

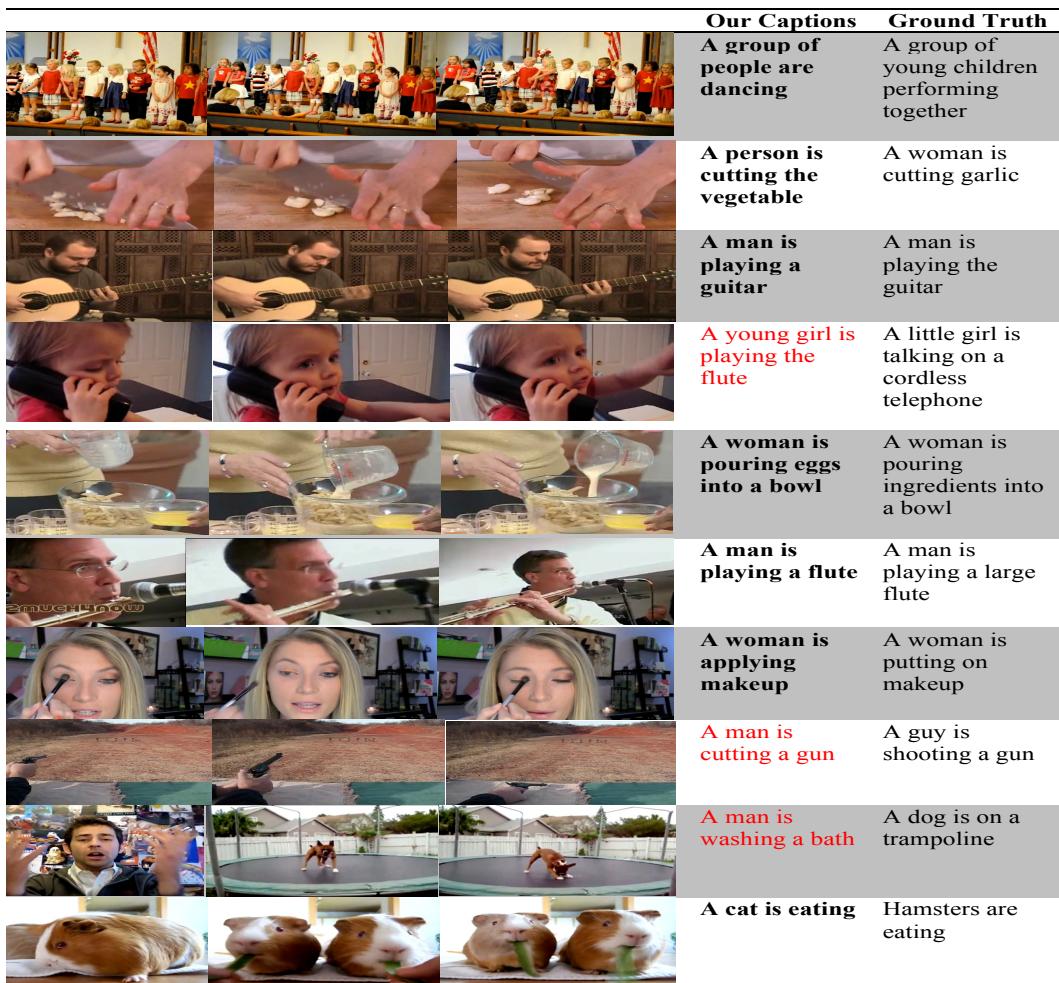


Figure 2: Example captions generated by our model on the test video for MSVD. Incorrect caption cases are shown in red.

5 CONCLUSION

We introduce an end-to-end memory-based attention model to describe an input video using natural language description, similar to the *central executive* system proposed in human cognition. Our model utilizes memories of past attention when reasoning about where to attend to in the current time step. This allows the model to not only reason about local attention more effectively, but also allows it to consider the entire sequence of video frames while generating each word. Our experiments have confirmed that the memory components in our architecture play a significant role in improving the performance of the entire network. It is worth noting that in this paper, we consider the problem of video caption generation, but our architecture can be applied to any sequence learning problem, which we hope to explore in the future.

REFERENCES

- Tatsuya Harada Andrew Shin, Katsunori Ohnishi. Beyond caption to narrative: Video captioning with multiple sentences. *ICIP*, 2016.
- A.D. Baddeley and G. Hitch. Working memory. *G.A. Bower (Ed.), The psychology of learning and motivation*, 8(4):47–89, 1974.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.
- Nicolas Ballas, Li Yao, Chris Pal, and Aaron C. Courville. Delving deeper into convolutional networks for learning video representations. In *ICLR*, 2016.
- James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13:281–305, 2012.
- T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *TPAMI*, 33(3):500–513, March 2011. ISSN 0162-8828.
- T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.
- David L. Chen and William B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, Portland, OR, June 2011.
- Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML-14*, pp. 1764–1772, 2014.
- Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *CoRR*, abs/1410.5401, 2014.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, June 2016.
- Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667.
- Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *CVPR*, 2016.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (eds.), *NIPS*, pp. 1097–1105. 2012.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueteng Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*, June 2016a.
- Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. *CVPR*, 2016b.
- Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. *ICML-13*, 28:1310–1318, 2013.
- Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. Reasoning about entailment with neural attention. In *ICLR*, 2016.
- M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In *ICCV*, pp. 433–440, Dec 2013.

- Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. *CoRR*, abs/1511.04119, 2015.
- Gunnar A. Sigurdsson, Gü̈l Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016.
- K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. *NIPS*, 2015.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NIPS*, pp. 3104–3112. 2014.
- Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.
- Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence – video to text. In *ICCV*, 2015a.
- Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL HLT*, 2015b.
- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, June 2015.
- Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *CoRR*, abs/1410.3916, 2014.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML-15*, pp. 2048–2057, 2015a.
- Ran Xu, Caiming Xiong, Wei Chen, and Jason J. Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, 2015b.
- Zhilin Yang, Ye Yuan, Yuexin Wu, Ruslan Salakhutdinov, and William W. Cohen. Encode, review, and decode: Reviewer module for caption generation. *CoRR*, abs/1605.07912, 2016.
- Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *ICCV*, 2015.
- Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, June 2016.