Recurrent Memory Addressing for describing videos

Arnav Kumar Jain* Abhinav Agarwalla* Kumar Krishna Agrawal* Pabitra Mitra
Indian Institute of Technology Kharagpur
{arnavkj95, abhinavagarawalla, kumarkrishna, pabitra}@iitkgp.ac.in

Abstract

In this paper, we introduce Key-Value Memory Networks to a multimodal setting and a novel key-addressing mechanism to deal with sequence-to-sequence models. The proposed model naturally decomposes the problem of video captioning into vision and language segments, dealing with them as key-value pairs. More specifically, we learn a semantic embedding (v) corresponding to each frame (k) in the video, thereby creating (k, v) memory slots. We propose to find the next step attention weights conditioned on the previous attention distributions for the key-value memory slots in the memory addressing schema. Exploiting this flexibility of the framework, we additionally capture spatial dependencies while mapping from the visual to semantic embedding. Experiments done on the Youtube2Text dataset demonstrate usefulness of recurrent key-addressing, while achieving competitive scores on BLEU@4, METEOR metrics against state-of-the-art models.

1. Introduction

Generating natural language descriptions for images and videos is a long-standing problem, in the intersection of computer vision and natural language processing. Solving the problem requires developing powerful models capable of extracting visual information about various objects in an image, while deriving semantic relationships between them in natural language. For video captioning, the models are additionally required to find compact representations of the video to capture the temporal dynamics across image frames.

The recent advances in training deep neural architectures have significantly improved in the state-of-the-art across computer vision and natural language understanding. With impressive results in object detection and scene understanding, Convolution Neural Networks (CNNs) [22] have become the staple for extracting feature representations from images. Recurrent Neural Networks (RNNs) with Long Short Term Memory (LSTM) [15] units or Gated Recurrent Units (GRUs)[10], have similarly emerged as generative models of choice for dealing with sequences in domains ranging from language modeling, machine translation to speech recognition. Advancements in these fundamental problems make tackling challenging problems, like captioning [16, 44], dialogue [31] and visual question answering [1] more viable.

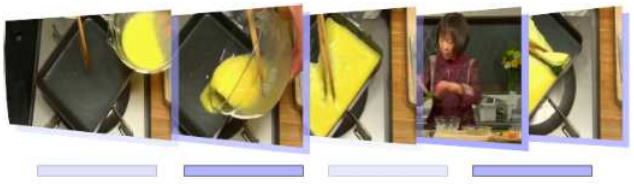

Figure 1. Our model employs a temporal attention mechanism on the visual features to identify key frames in the video. These are mapped to semantic features in the language domain for better context to the language model, which then generates the output sequence. Previously attended frames and generated words identify the key frames for generating the next word.

Despite the fundamental complexities of these problems, there has been an increasing interest in solving them. A common underlying approach in these proposed models is the notion of "attention mechanisms", which refers to selectively focusing on segments of sequences [3, 46] or images [34] to generate corresponding outputs. Such attention based approaches are specially attractive for captioning problems, since they allow the network to focus on patches of the image conditioned on the previously generated tokens [44, 16], often referred to as spatial attention.

Models with spatial attention, however cannot be readily used for video description. For instance, in the Youtube2Text dataset, a video clip stretches around 10 seconds, or around 150 frames. Applying attention on patches in these individual frames provides the network with local spatial context. This however, does not take ordering of the frame sequence or events ranging across frames, into consideration. To incorporate this *temporal attention* into the model, [46, 48, 26] extend this *soft alignment* to video

*denotes equal contribution

captioning. Most of these approaches, treat the problem of video captioning in the sequence-to-sequence paradigm [36] with attentive encoders and decoders. This requires finding a compact representation of the video, which is passed as context to the RNN decoder.

However, we identify two primary issues with these approaches. First, applying attention sequentially provides the model with local context at the generative decoder [45]. As a result the decoder would be unable to deal with long-term dependencies while captioning videos of longer duration. Secondly, these models jointly learn the multimodal embedding in a visual-semantic space [27, 48] at the RNN decoder. With the annotated sentences being the only supervisory signal, learning a mapping from a sequence of images to a sequence of words is difficult. This is specially true for dealing with video sequences, as the underlying probability distribution is distinctively multimodal. While [27] tries to address this issue with an auxiliary loss, the model suffers from the first drawback.

To address the aforementioned issues, we introduce a model which generalizes Key-Value Memory Networks [24] to a multimodal setting for video captioning. At the same time, the framework provides an effective way to deal with the complex transformation from the visual to language domain. Using a pre-trained model to explicitly transform individual frames (*keys*) to semantic embedding (*values*), we construct memory slots with each slot being a tuple (*key, value*). This allows us to provide a weighted pooling of textual features as context to the decoder RNN, which is closer to the language model. The proposed model naturally tackles the problem of maintaining long-term temporal dependencies in videos, by explicitly providing all image frames for selection at each time step. We also propose a novel key-addressing scheme (see Section 4), allowing us to find the new relevance scores conditioned on the previous attention distribution. It keeps track of previous attention distribution and provides a global context to the decoder. This allows us to exploit the temporal dependencies in the recurrent key-addressing and in the language decoder.

In summary, our key contributions are following:

- We generalize Key-Value Memory Networks (KV-MemNN) in a multimodal setting to generate natural descriptions for videos, and more generally deal with sequence-to-sequence models (Section 3).
- We propose a novel key-addressing schema to find the attention weights for key-value memory slots conditioned on the previous attention distribution (Section 4).
- The proposed model is evaluated on the YouTube dataset [7], where we outperform strong baselines while reporting competitive results against state-of-art models (Section 5.6).

2. Related Work

Following the success of end-to-end neural architectures and attention mechanisms, there is a growing body of literature for captioning tasks, in images and more recently videos. To deal with the multimodal nature of the problem, classical approaches relied on manually engineered templates [19, 11]. And while some recent approaches in this direction show promise [13], but the models lack generalization to deal with complex scenes, videos.

As an alternative approach, [14, 18] suggest learning a joint visual-semantic embedding, effectively a mapping from the visual to language space. The motivation of our work is strongly aligned with [30], who generate semantic representations for images using CRF models, as context for the language decoder. However, our approach significantly differs in the essence that we capture spatio-temporal dynamics in videos while generating the text description.

Building on this, and Encoder-Decoder [3, 9] models for machine translation , [41, 40] develop models which compute average fixed-length representations (for images, videos respectively) from image features. These *context* vectors are provided at each time step to the decoder language model, for generating descriptions. The visual representations for the images are usually transferred from pre-trained convolution networks [32, 37].

A major drawback of the above approach is induced by mean pooling, where context features across image frames are collapsed. For one, this looses the temporal structure across frames by treating them as "bag-of-images" model. Addressing this, [36] propose Sequence-to-Sequence models for accounting for the temporal structure, and [39] extend it to a video-captioning setting. However, passing a fixed vector as context at each time step, creates a bottleneck for the flow of gradients using Backpropagation Through Time (BPTT) [42] at the encoder.

The notion of *visual attention* has a rich literature in Psychology and Neuroscience, and has recently found application in computer vision [25] and machine translation [3]. Allowing the network to selectively focus on the patches of images or segments of the input sequences, representative works [44, 46, 16, 4, 48, 26, 27] have significantly pushed the state-of-the-art in their domain. The issues of fixed length representation and gradient bottleneck are largely addressed by selectively conditioning the decoder outputs on encoder states:

$$p(y_i|y_{i-1}, ..., y_1, x) = g(y_{i-1}, s_i, c_i) \tag{1}$$

where, y_i is the readout, c_i is the context from the encoder and $s_i = f(s_{i-1}, y_{i-1}, c_i)$, is the hidden state of decoder RNN (See [3] for details).

However, as discussed in Section 1, sequential attention provides the decoder with local context [45]. Additionally, providing a semantic input which is closer to language

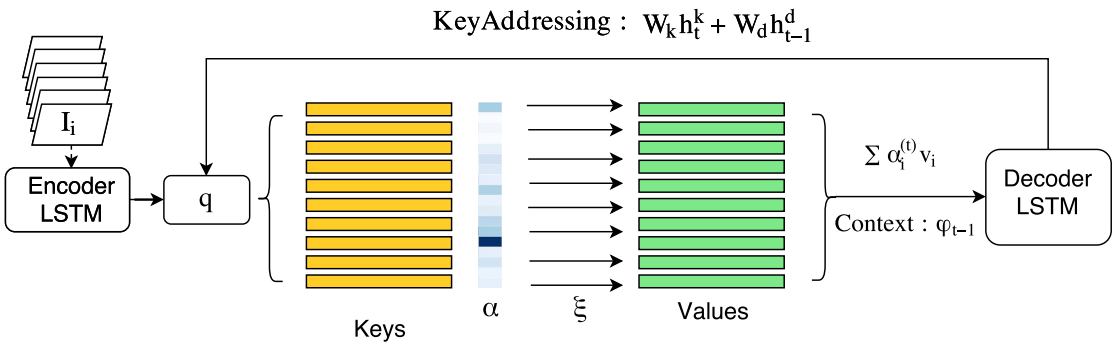

Figure 2. The video is considered as a sequence of image frames $\{I_1, ..., I_n\}$. The memory is filled with key-value pairs (k_i, v_i) that capture the relationships between visual features and textual descriptions. The α_i^t corresponds to the attention weights associated with the memory slots $(h_{t-1}, h_t, ...)$ being the hidden states of the decoder RNN. The memory is then queried over and over again to produce a weighted sum of the values to be decoded using a standard LSTM RNN decoder into a word in the description.

space, as context to the decoder significantly improves on the capability of the model [47]. Our work closely brings these advances together in a Memory Networks framework [43, 35, 20]. While we introduce Key-Value Memory Networks [24] in a multimodal setting, there are several other key differences from previous works. For one, to our knowledge this is the first work which introduces video captioning in light of Memory Networks. This automatically deals with problems of maintaining long-term dependencies in the input stream by explicitly storing image representations. Meanwhile we also tackle the "vanishing gradient problem" typical with training RNN Encoder-Decoder modules for long input sequences.

Key-Value MemNNs [24] were originally proposed for QA task in the language domain, providing the last time-step hidden state, as input to the classifier. In this work, we address a more complex problem of video captioning by proposing a novel key-addressing scheme (details in Section 4) and *(key, value)* setup for exploiting the spatio-temporal structures. The model tracks the attention distribution at previous time steps, thereby providing a strong context on where to attend on the complete video sequence. This implicitly provides a global temporal structure at each readout. While similar in motivation to [47, 29], the model architecture and domain of application, especially on capturing global temporal dynamics in videos as opposed to images or entailment, is significantly different.

3. Key-Value Memory Networks for videos

Our work is based on the encoder-decoder framework [9, 3, 44, 41, 17], in a Memory Networks [43, 24] setting to generate descriptions of videos. The encoder network learns a mapping from the input sequence to a fixed-length vector representation, which is passed to the decoder to generate output sequences. Similar to standard Encoder-Decoder with soft attention mechanism, our model (see Fig. 2) comprises an encoder module, key-value memories and a decoder module.

3.1. Encoder

The encoder network E maps a given input sequence of images in a video $X = \{I_1, ..., I_T\}$ of length T to the corresponding sequence of fixed size context representation vectors. As we are dealing with videos (sequence of images), we define two different encoders to achieve the mapping.

CNN Encoder: Given an input image $I_i \in \mathbb{R}^{NxM}$, the CNN encoders learn a mapping f from image I_i to context representations of size D given by $f : \mathbb{R}^{NxM} \rightarrow \mathbb{R}^D$. The output of either fully-connected layers[32] or feature maps of convolutional layers[46] of standard ConvNet architectures is considered.

RNN Encoder: The RNN encoder processes the features extracted from CNN Encoder of the frames sequentially, generating hidden states h_i^e at each time step which summarizes the sequence of images seen so far, where

$$h_i^e = g(f(I_i), h_{i-1}^e) \tag{2}$$

While maintaining temporal dependencies, this allows us to map variable length sequences to fixed length context vectors. In this work, we use modified version of LSTM [15] unit, as proposed in [49] to implement g. The RNN Encoder allows the model to capture the temporal variation between frames and take the ordering of actions and events into consideration. Implicitly, this also helps in preserving high level information about motion in the video [4]. We extract the features from the CNN Encoder, and pass these extracted feature vectors through the RNN Encoder.

3.2. Key-Value Memories

The model is built around a Key-Value Memory Network [24] with memory slots as key-value pairs

$(k_1, v_1), ..., (k_T, v_T)$. The keys and values serve the purpose of transforming visual space context into language space, and effectively capture the relationships between the visual features and textual descriptions. The memory definition, addressing and reading schema is outlined below:

Keys (K): Using CNN Encoder, visual context k_i is generated for each frame I_i of the video. These appearance feature vectors are passed through a RNN Encoder to incorporate sequential structure (video being a sequence of images), and hidden state h_i^e at each timestep is extracted as key k_i, given by $k_i = h_i^e$.

Values (V): For each image frame I_i, a semantic embedding v_i representing the textual meaning of a particular frame/key is generated. It is difficult to jointly learn visual-semantic embedding in Encoder-Decoder models, with supervisory signal only from annotated descriptions [47]. To mitigate this, we explicitly precompute semantic embeddings corresponding to individual frames in the video. In our case, we obtained v_i from a pretrained model ψ which jointly models visual and semantic embedding for images [44, 16, 41], given by $v_i = \psi(I_i)$. Now, for each frame I_i in the video, we have a key-value memory slot (k_i, v_i).

Key Addressing: This corresponds to the soft-attention mechanism deployed to assign a relevance probability α_i to each of the memory slots. These relevance probabilities are used for value reading. We have introduced a new Key Addressing scheme which is described in Section 4.

Value Reading: The value reading of the memory slots is the weighted sum of the key-value feature vectors: $\phi_t(K)$ and $\phi_t(V)$ at each time step. $\phi_t(K)$ is used for key addressing at the next time step(details in Section 4) and $\phi_t(V)$ is passed as input to the decoder RNN for generating the next word.

$$\phi_t(K) = \sum_{i=1}^{T} \alpha_i^{(t)} k_i, \phi_t(V) = \sum_{i=1}^{T} \alpha_i^{(t)} v_i \tag{3}$$

3.3. Decoder

Recurrent Neural Networks is used as decoder because they have been widely used for natural language generation tasks like machine translation, image captioning and video description generation. Since, vanilla RNNs are difficult to train for long range dependencies as they suffer from the *vanishing gradient problem* [5], Long Short Term Memory(LSTM) [15] is used. The LSTM units are capable of memorizing context for longer period of time using controllable memory units.

The LSTM model has a memory cell c_t in addition to the hidden state h_t in RNNs, which effectively summarizes the information observed up to that time step. There are primarily three gates which control the flow of information i.e (input, output, forget). The input gate i_t controls the current input x_t, forget gate f_t adaptively allows to forget old memory and output gate o_t decides the extent of transfer of cell memory to hidden state. The recurrences at the decoder in our case are defined as:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{h}_{t-1} + \mathbf{U}_i \mathbf{x}_t + \mathbf{A}_i \phi_t(V) + \mathbf{b}_i) \tag{4}$$
$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{h}_{t-1} + \mathbf{U}_f \mathbf{x}_t + \mathbf{A}_f \phi_t(V) + \mathbf{b}_f) \tag{5}$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{h}_{t-1} + \mathbf{U}_o \mathbf{x}_t + \mathbf{A}_o \phi_t(V) + \mathbf{b}_o) \tag{6}$$
$$\tilde{\mathbf{c}}_\mathbf{t} = \tanh(\mathbf{W}_c \mathbf{h}_{t-1} + \mathbf{U}_c \mathbf{x}_t + \mathbf{A}_c \phi_t(V) + \mathbf{b}_c) \tag{7}$$
$$\mathbf{c}_t = \mathbf{i}_t \odot \tilde{\mathbf{c}}_\mathbf{t} + \mathbf{f}_t \odot \mathbf{c}_{t-1} \tag{8}$$
$$\mathbf{h}_t = \mathbf{o}_t \odot \mathbf{c}_t \tag{9}$$

where $\odot$ is an element wise multiplication, σ is the sigmoidal non-linearity. $\mathbf{W}_x, \mathbf{U}_x, \mathbf{A}_x$ and $\mathbf{b}_x$, are the weight matrices for the previous hidden state, input, value context and bias respectively.

Following standard sequence-to-sequence models with generative decoders, we apply a single layer network on the hidden state h_t followed by softmax function to get the probability distribution over the set of possible words.

$$\mathbf{p}_t = \text{softmax}(\mathbf{U}_p[h_t, x_t, \phi_t(V)] + \mathbf{b}_p) \tag{10}$$

Here p_t is the probability distribution over the vocabulary for sampling the current word and [...] denotes vector concatenation. Sentences with high probability are found using Beam Search[36].

4. Key Addressing

Soft attention mechanism have been successful in image captioning[44] and video captioning [46] because they focus on the most important segments, and weights them accordingly. Previous work based on soft attention mechanism[46] use the decoder's hidden state h_t to find attention weights of each memory unit. We propose a new key addressing mechanism which looks at the previous attention distribution over keys in addition to h_t to select relevant frames for generating the next word. The attention distribution over keys denotes the importance of frames attended so far and the hidden state of the decoder summarizes the previously generated words. This allows us to take into consideration the previously generated words, the attention distribution at previous time steps and the individual key representations k_i's to find relevance score for keys.

We experiment with two different Key-Addressing methods. In first method, we use the previous weighted sum of the keys $\phi_{t-1}(K)$ directly to find next step attention distribution. In second method, we have a Key-Addressing RNN (referred to as Memory LSTM in Fig. 3) which takes previous value read over keys $\phi_{t-1}(K)$ as input.

$$\mathbf{h}_t^k = \mathbf{f}^k(\phi_{t-1}(K), h_{t-1}^k) \tag{11}$$

where f^k is the recurrent unit. For first method, h_t^k is essentially $\phi_{t-1}(K)$. The next step attention weights α_i^t are

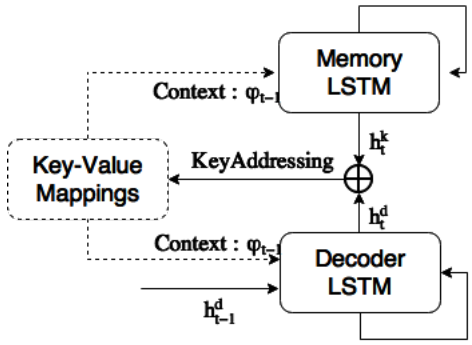

Figure 3. The key addressing LSTM is shown here. The memory LSTM updates its hidden state using the last attention distribution over keys. The new hidden state is used with decoder's last hidden state to get new relevance scores which are combined with values and passed to the decoder to generate the next word.

obtained using the hidden state $\mathbf{h}_t^k$ of this RNN-LSTM. The hidden state of Key-Addressing RNN at initial time step is the mean-pooled average of all the keys.

The query vector q is a weighted combination of the decoder and key-addressing hidden states. It summarizes the frames seen so far and the generated outputs.

$$q = W_k h_t^k + W_d h_{t-1} \tag{12}$$

For obtaining the attention weights, the relevance score e_i^t of i-th temporal feature k_i is obtained using the decoder RNN hidden state h_{t-1}, key addressing RNN hidden state h_t^k and the i-th key vector k_i:

$$e_i^t = \mathbf{w}_t \tanh(\mathbf{q} + \mathbf{U}_a k_i) \tag{13}$$

where $\mathbf{w}_t$, $\mathbf{W}_d$, $\mathbf{W}_k$ and $\mathbf{U}_a$ are parameters of the model.

These relevance scores are normalised using a softmax function to obtain the new attention distribution α^t, where:

$$\alpha_i^t = exp\{e_i^t\} / \sum_{j=1}^{N} exp\{e_j^t\} \tag{14}$$

The segregation of the vision and language components into key-value pairs provides a better context for the RNN decoder. Also, the explicit memory structure provides access to the image frames at all time steps allowing the model to assign weights to the key-frames without losing information.

5. Experimental Setup

5.1. Dataset

Youtube2Text The proposed approach is benchmarked on the Youtube2Text[7] dataset which consists of 1,970 Youtube videos with multiple descriptions annotated through Amazon Mechanical Turk. The videos are generally short (9 seconds on an average), and depict a single activity. Activities depicted are open domain ranging from everyday objects to animals, scenarios, actions, landscapes. etc. The dataset consists of 80,839 annotations with an average of 41 annotations per clip and 8 words per sentence respectively. The training, validation and test sets have 1,200, 100 and 670 videos respectively which is exactly the same splits as in previous work on video captioning [46, 4, 26].

Key-Value Memories We select 28 equally spaced frames and pass them through a pretrained VGG-16[32] and GoogleNet[37] because of their state of the art performance in object detection on Imagenet[12] database. For an input image of size WXH, visual features with shape $(\lfloor \frac{W}{16} \rfloor, \lfloor \frac{W}{16} \rfloor, C)$ with C as 512 are extracted from the *conv5_3* layer of VGG-16. We simply average over the feature maps which results in a feature vector of size C. The visual features extracted from the *pool5/7x7_s1* layer of GoogLeNet is a 1024 dimensional vector. The feature vectors are either directly used as keys or are passed to encoder RNN to generate keys as described in Section 3.

The values are generated from a pre-trained Densecap [16] model, which jointly models the task of object localization and textual description. The model identifies salient regions in an image and generates a caption for each of these regions. We extract the output of Recognition Network which is encoded as region codes of size BxD, where B is the number of salient regions or boxes, and D is the representation with dimension 4096. Along with the features, a score S is assigned to each of the regions which denotes its confidence. A weighted sum of features of top 5 scores is calculated to get values.

Preprocessing: The video descriptions are tokenized using the wordpunct_tokenizer from the NLTK toolbox[23]. The number of unique words were 15,903 in the Youtube2Text dataset.

5.2. Model Specifications

We test on four different variations of the model which help us identify changes in architecture that lead to large improvements on the evaluation metric. *VGG-Encoder* uses features encoded from the last convolution layer in VGG-16 [32] network, and *GoogLeNet-Encoder* uses features extracted from GoogLeNet[37] as input to the model. There is no Key Addressing in the above two models which means attention weights are obtained using last hidden state of decoder. *t-KeyAddressing* extends the *GoogLeNet-Encoder* by addressing keys using the previous attention distribution over keys. Finally, *m-KeyAddressing* addresses keys using key addressing RNN instead of using the last attention distribution.

Table 1. Experiment results on the Youtube2Text Dataset.

Model	BLEU@4	METEOR	CIDEr	Feat.	Fine
VGG-Encoder	0.404	0.295	0.515	No	No
GoogLeNet-Encoder	0.427	0.303	0.534	No	No
t-KeyAddressing	0.436	0.308	0.545	No	No
m-KeyAddressing (Memory LSTM)	**0.457**	**0.319**	**0.573**	No	No
Enc-Dec Basic(Yao et al. [46])	0.3869	0.2868	0.4478	No	No
GoogLeNet + HRNE(Pan et al. [26])	0.438	**0.321**	.	No	No
LSTM-E(VGG + C3D)(Pan et al. [27])	0.453	0.310	.	No	No
C3D(Yao et al. [46])	0.4192	0.2960.	0.5167	Yes	No
VGG + C3D + p-RNN(Yu et al.[48]	**0.499**	**0.326**	-	Yes	No
S2VT(Venugopalan et al. [39])	-	.298	-	Yes	No
GRU-RCN(Ballas et al. [4])	**0.490**	0.3075	**0.5937**	Yes	Yes

5.3. Model Comparisons

We compare the model performance with previous state of the art approaches and some strong baselines. Pan et al. [27] explicitly learn a visual-semantic joint embedding model for exploiting the relationship between visual features and generated language, which is then used in a encoder-decoder framework. Yao et al.[46] utilizes a temporal attention mechanism for global attention apart from local attention using 3-D Convolution Networks. Ballas et al.[4] proposed an encoder to learn spatial-temporal features across frames, introducing a variant GRU with convolution operations (GRU-RCN). In the current state-of-art Yu et al. [48] models the decoder as a paragraph generator, describing the videos over multiple sentences using stacked LSTMs.

5.4. Evaluation Metrics

We evaluate our approach using standard evaluation metrics to compare the generated sequences with the human annotations, namely BLEU [28], METEOR [21] and CIDEr[38]. We use the code accompanying the Microsoft COCO Evaluation script [8] to obtain the results reported in the paper.

5.5. Training Details

The model predicts the next output word conditioning on previously generated words and the input video. Thus, the goal is to maximize the log likelihood of the loss function:

$$L = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{|y^i|}\log p(y_j^i|y_{<j}^i, \mathbf{x}^n, \theta) \qquad (15)$$

where N is the total number of video-description pairs and length of each description y^i is $|y^i|$. Here x^n refers to the input video provided as context to the decoder. We train our network parameters θ through first order stochastic gradient-based optimization with an adaptive learning rate using the Adadelta [50] optimizer. The batch size is set to be 64 and we optimize hyper-parameters, which include number of hidden units in Decoder LSTM, key addressing LSTM, learning rate and word embedding dimension for the log loss using random search [6].

5.6. Results

In the first block of Table 5.1, we present the performances of different variations of the model followed by results of prior work in subsequent lines. The *VGG-Encoder* model outperforms S2VT [39] and the Basic Enc-Dec model [46] on all three metrics, which shows that it is beneficial to use Key-Value Memory Networks in a multimodal setting. We observe that using features from pretrained GoogleNet further improves the results. Using our approach, we further outperform the Enc-Dec model [46].

Results on *t-KeyAddressing* and *m-KeyAddressing* shows further boost in performances on all the metrics demonstrating the effectiveness of using Key Addressing scheme. *m-KeyAddressing* outperforms Pan et al. [27] by a significant margin on BLEU@4. While the improvements on METEOR are significant compared to *t-KeyAddressing*, [26] performs slightly better. In the current setting, our model is unable to outperform Yu et al.[48] and Ballas et al.[4]. It must be noted that using more sophisticated regularizers for training the decoder, as proposed in [2] and using a better encoder[4] should lead to increased evaluation scores.

In Table 5.1 we also provide comparison on whether the models use finetuning on the CNN encoder (represented by *Fine*) or if they use external features, like on action recognition, optical flow (represented by *Feat*). It is to be noted, that we do not finetune the encoderas compared to [4], which finetunes the encoder CNN on UCF101 action recognition set[33]. Also, no additional features are extracted for gaining more information about motion, actions etc. as in [48], [4], [39].

Fig 4 shows examples of some of the input frames and generated outputs, along with ground truths. Some of the

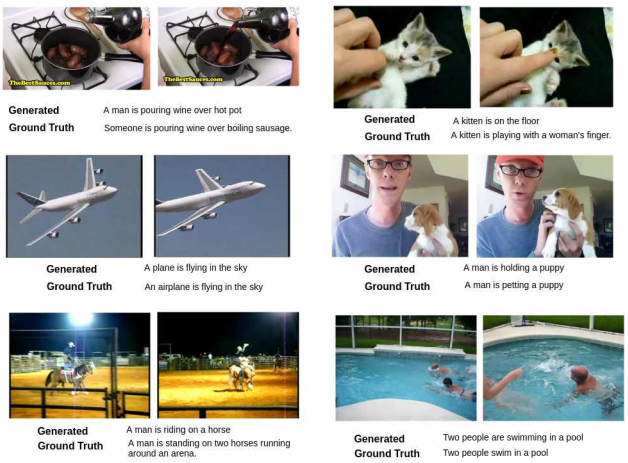

Figure 4. Samples generated on the Youtube2Text dataset.

examples demonstrate that the model is able to infer the activities from the video frames, like "swimming", "riding" and "flying" which is distributed across multiple frames.

6. Conclusion

We demonstrate the potential of Memory Networks, specifically Key-Value Memory Networks for video captioning task by decomposing memory into visual and language components as key-value pairs. This paper also proposes a key addressing system for dealing with sequence-to-sequence models, which considers the previous attention distribution over the keys to calculate the new relevance scores. Experiments done on the proposed model outperform strong baselines across several metrics. To the best of our knowledge this is the first proposed work for video-captioning in a Memory Networks setting, and does not rely heavily on annotated videos to generate intermediate semantic-embedding for supporting the decoder. Further work would be exploring the effectiveness of the model on longer videos and generating fine-grained descriptions with more sophisticated decoders.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015. 1

[2] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 6

[3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 1, 2, 3

[4] N. Ballas, L. Yao, C. Pal, and A. Courville. Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432*, 2015. 2, 3, 5, 6

[5] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. 4

[6] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012. 6

[7] D. L. Chen and W. B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 190–200. Association for Computational Linguistics, 2011. 2, 5

[8] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 6

[9] K. Cho, B. Van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. 2, 3

[10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 1

[11] P. Das, C. Xu, R. F. Doell, and J. J. Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2634–2641, 2013. 2

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 5

[13] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1473–1482, 2015. 2

[14] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013. 2

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1, 3, 4

[16] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. *arXiv preprint arXiv:1511.07571*, 2015. 1, 2, 4, 5

[17] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015. 3

[18] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014. 2

[19] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions*

on Pattern Analysis and Machine Intelligence, 35(12):2891–2903, 2013. 2

[20] A. Kumar, O. Irsoy, J. Su, J. Bradbury, R. English, B. Pierce, P. Ondruska, I. Gulrajani, and R. Socher. Ask me anything: Dynamic memory networks for natural language processing. *arXiv preprint arXiv:1506.07285*, 2015. 3

[21] M. D. A. Lavie. Meteor universal: language specific translation evaluation for any target language. *ACL 2014*, page 376, 2014. 6

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[23] E. Loper and S. Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. 5

[24] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*, 2016. 2, 3

[25] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, pages 2204–2212, 2014. 2

[26] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. *arXiv preprint arXiv:1511.03476*, 2015. 1, 2, 5, 6

[27] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. *arXiv preprint arXiv:1505.01861*, 2015. 2, 6

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002. 6

[29] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiskỳ, and P. Blunsom. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*, 2015. 3

[30] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 433–440, 2013. 2

[31] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, 2016. 1

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 3, 5

[33] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 6

[34] M. F. Stollenga, J. Masci, F. Gomez, and J. Schmidhuber. Deep networks with internal selective attention through feedback connections. In *Advances in Neural Information Processing Systems*, pages 3545–3553, 2014. 1

[35] S. Sukhbaatar, J. Weston, R. Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015. 3

[36] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. 2, 4

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 2, 5

[38] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575, 2015. 6

[39] 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 (ICCV)*, 2015. 2, 6

[40] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL HLT*, 2015. 2

[41] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015. 2, 3, 4

[42] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990. 2

[43] J. Weston, S. Chopra, and A. Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014. 3

[44] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, abs/1502.03044, 2015. 1, 2, 3, 4

[45] Z. Yang, Y. Yuan, Y. Wu, R. Salakhutdinov, and W. W. Cohen. Encode, review, and decode: Reviewer module for caption generation. *arXiv preprint arXiv:1605.07912*, 2016. 2

[46] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4507–4515, 2015. 1, 2, 3, 4, 5, 6

[47] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. *arXiv preprint arXiv:1603.03925*, 2016. 3, 4

[48] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. *arXiv preprint arXiv:1510.07712*, 2015. 1, 2, 6

[49] W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014. 3

[50] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 6