Attention-Based Multimodal Fusion for Video Description

Hori, C.; Hori, T.; Lee, T.-Y.; Sumi, K.; Hershey, J.R.; Marks, T.K.

Abstract

Currently successful methods for video description are based on encoder-decoder sentence generation using recurrent neural networks (RNNs). Recent work has shown the advantage of integrating temporal and/or spatial attention mechanisms into these models, in which the decoder network predicts each word in the description by selectively giving more weight to encoded features from specific time frames (temporal attention) or to features from specific spatial regions (spatial attention). In this paper, we propose to expand the attention model to selectively attend not just to specific times or spatial regions, but to specific modalities of input such as image features, motion features, and audio features. Our new modality-dependent attention mechanism, which we call multimodal attention, provides a natural way to fuse multimodal information for video description. We evaluate our method on the Youtube2Text dataset, achieving results that are competitive with current state of the art. More importantly, we demonstrate that our model incorporating multimodal attention as well as temporal attention significantly outperforms the model that uses temporal attention alone.

arXiv

Attention-Based Multimodal Fusion for Video Description

Chiori Hori   Takaaki Hori   Teng-Yok Lee   Kazuhiro Sumi*   John R. Hershey

Tim K. Marks

Mitsubishi Electric Research Laboratories (MERL)

{chori, thori, tlee, sumi, hershey, tmarks}@merl.com

Abstract

Currently successful methods for video description are based on encoder-decoder sentence generation using recurrent neural networks (RNNs). Recent work has shown the advantage of integrating temporal and/or spatial attention mechanisms into these models, in which the decoder network predicts each word in the description by selectively giving more weight to encoded features from specific time frames (temporal attention) or to features from specific spatial regions (spatial attention). In this paper, we propose to expand the attention model to selectively attend not just to specific times or spatial regions, but to specific modalities of input such as image features, motion features, and audio features. Our new modality-dependent attention mechanism, which we call multimodal attention, provides a natural way to fuse multimodal information for video description. We evaluate our method on the Youtube2Text dataset, achieving results that are competitive with current state of the art. More importantly, we demonstrate that our model incorporating multimodal attention as well as temporal attention significantly outperforms the model that uses temporal attention alone.

1. Introduction and Related Work

Automatic video description, also known as video captioning, refers to the automatic generation of a natural language description (e.g., a sentence) that summarizes an input video. Video description has widespread applications including video retrieval, automatic description of home movies or online uploaded video clips, and video descriptions for the visually impaired. Moreover, developing systems that can describe videos may help us to elucidate some key components of general machine intelligence. Video description research depends on the availability of videos labeled with descriptive text. A large amount of such data is becoming available in the form of audio description prepared for visually impaired users. Thus there is an opportunity to make significant progress in this area. We propose a video description method that uses an attention-based encoder-decoder network to generate sentences from input video.

Sentence generation using an encoder-decoder architecture was originally used for neural machine translation (NMT), in which sentences in a source language are converted into sentences in a target language [25, 5]. In this paradigm, the encoder takes an input sentence in the source language and maps it to a fixed-length feature vector in an embedding space. The decoder uses this feature vector as input to generate a sentence in the target language. However, the fixed length of the feature vector limited performance, particularly on long input sentences, so [1] proposed to encode the input sentence as a sequence of feature vectors, employing a recurrent neural network (RNN)-based soft attention model to enable the decoder to pay attention to features derived from specific words of the input sentence when generating each output word.

The encoder-decoder based sequence to sequence framework has been applied not only to machine translation but also to other application areas including speech recognition [2], image captioning [25], and dialog management [16].

In image captioning, the input is a single image, and the output is a natural-language description. Recent work on RNN-based image captioning includes [17, 25]. To improve performance, [27] added an attention mechanism, to enable focusing on specific parts of the image when generating each word of the description.

Encoder-decoder networks have also been applied to the task of video description [24]. In this task, the inputs to the encoder network are video information features that may include static image features extracted using convolutional neural networks (CNNs), temporal dynamics of videos extracted using spatiotemporal 3D CNNs [22], dense trajectories [26], optical flow, and audio features [12]. The decoder network takes the encoder outputs and generates word

---

sequences based on language models using recurrent neural networks (RNNs) based on long short-term memory (LSTM) units [9] or gated recurrent units (GRUs) [4]. Such systems can be trained end-to-end using videos labeled with text descriptions.

One inherent problem in video description is that the sequence of video features and the sequence of words in the description are not synchronized. In fact, objects and actions may appear in the video in a different order than they appear in the sentence. When choosing the right words to describe something, only the features that directly correspond to that object or action are relevant, and the other features are a source of clutter. It may be possible for an LSTM to learn to selectively encode different objects into its latent features and remember them until they are retrieved. However, attention mechanisms have been used to boost the network's ability to retrieve the relevant features from the corresponding parts of the input, in applications such as machine translation [1], speech recognition [2], image captioning [27], and dialog management [10]. In recent work, these attention mechanisms have been applied to video description [28, 29]. Whereas in image captioning the attention is spatial (attending to specific regions of the image), in video description the attention may be temporal (attending to specific time frames of the video) in addition to (or instead of) spatial.

In this work, we propose a new use of attention: to fuse information across different modalities. Here we use modality loosely to refer to different types of features derived from the video, such as appearance, motion, or depth, as well as features from different sensors such as video and audio features. Video descriptions can include a variety of descriptive styles, including abstract descriptions of the scene, descriptions focused on objects and their relations, and descriptions of action and motion, including both motion in the scene and camera motion. The soundtrack also contains audio events that provide additional information about the described scene and its context. Depending on what is being described, different modalities of input may be important for selecting appropriate words in the description. For example, the description "A boy is standing on a hill" refers to objects and their relations. In contrast, "A boy is jumping on a hill" may rely on motion features to determine the action. "A boy is listening to airplanes flying overhead" may require audio features to recognize the airplanes, if they do not appear in the video. Not only do the relevant modalities change from sentence to sentence, but also from word to word, as we move from action words that describe motion to nouns that define object types. Attention to the appropriate modalities, as a function of the context, may help with choosing the right words for the video description.

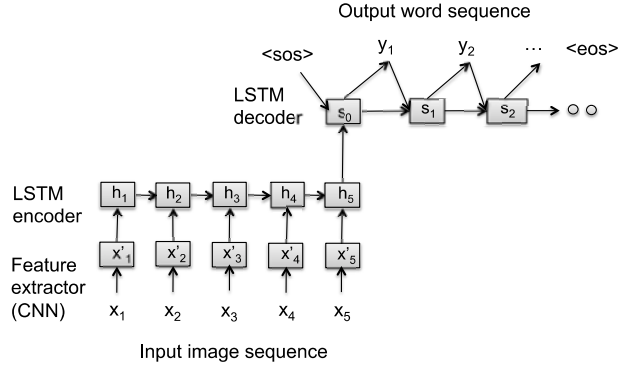Often features from different modalities can be comple-

Figure 1. An encoder-decoder based video description generator.

mentary, in that either can provide reliable cues at different times for some aspect of a scene. Multimodal fusion is thus an important longstanding strategy for robustness. However, optimally combining information requires estimating the reliability of each modality, which remains a challenging problem. In this work, we propose that this estimation be performed by the neural network, by means of an attention mechanism that operating across different modalities (in addition to any spatio-temporal attention). By training the system end-to-end to perform the desired description of the semantic content of the video, the system can learn to use attention to fuse the modalities in a context-sensitive way. We present experiments showing that incorporating multimodal attention, in addition to temporal attention, significantly outperforms a corresponding model that uses temporal attention alone.

2. Encoder-decoder-based sentence generator

One basic approach to video description is based on sequence-to-sequence learning. The input sequence, i.e., image sequence, is first encoded to a fixed-dimensional semantic vector. Then the output sequence, i.e., word sequence, is generated from the semantic vector. In this case, both the encoder and the decoder (or generator) are usually modeled as Long Short-Term Memory (LSTM) networks. Figure 1 shows an example of the LSTM-based encoder-decoder architecture.

Given a sequence of images, $X = x_1, x_2, \ldots, x_L$, each image is first fed to a feature extractor, which can be a pre-trained CNN for an image or video classification task such as GoogLeNet [15], VGGNet [20], or C3D [22]. The sequence of image features, $X' = x'_1, x'_2, \ldots, x'_L$, is obtained by extracting the activation vector of a fully-connected layer of the CNN for each input image.[1] The sequence of feature vectors is then fed to the LSTM encoder, and the hidden

---

[1]In the case of C3D, multiple images are fed to the network at once to capture dynamic features in the video.

state of the LSTM is given by

$$h_t = \text{LSTM}(h_{t-1}, x_t'; \lambda_E), \quad (1)$$

where the LSTM function of the encoder network λ_E is computed as

$$\text{LSTM}(h_{t-1}, x_t; \lambda) = o_t \tanh(c_t), \quad (2)$$

$$\text{where} \quad o_t = \sigma\big(W_{xo}^{(\lambda)} x_t + W_{ho}^{(\lambda)} h_{t-1} + b_o^{(\lambda)}\big) \quad (3)$$

$$c_t = f_t c_{t-1} + i_t \tanh\big(W_{xc}^{(\lambda)} x_t$$
$$+ W_{hc}^{(\lambda)} h_{t-1} + b_c^{(\lambda)}\big) \quad (4)$$

$$f_t = \sigma\big(W_{xf}^{(\lambda)} x_t + W_{hf}^{(\lambda)} h_{t-1} + b_f^{(\lambda)}\big) \quad (5)$$

$$i_t = \sigma\big(W_{xi}^{(\lambda)} x_t + W_{hi}^{(\lambda)} h_{t-1} + b_i^{(\lambda)}\big), \quad (6)$$

where $\sigma()$ is the element-wise sigmoid function, and i_t, f_t, o_t and c_t are, respectively, the input gate, forget gate, output gate, and cell activation vectors for the tth input vector. The weight matrices $W_{zz}^{(\lambda)}$ and the bias vectors $b_z^{(\lambda)}$ are identified by the subscript $z \in \{x, h, i, f, o, c\}$. For example, W_{hi} is the hidden-input gate matrix and W_{xo} is the input-output gate matrix. We did not use peephole connections in this work.

The decoder predicts the next word iteratively beginning with the start-of-sentence token, "`<sos>`" until it predicts the end-of-sentence token, "`<eos>`." Given decoder state s_{i-1}, the decoder network λ_D infers the next word probability distribution as

$$P(y|s_{i-1}) = \text{softmax}\left(W_s^{(\lambda_D)} s_{i-1} + b_s^{(\lambda_D)}\right), \quad (7)$$

and generates word y_i, which has the highest probability, according to

$$y_i = \underset{y \in V}{\arg\max}\, P(y|s_{i-1}), \quad (8)$$

where V denotes the vocabulary. The decoder state is updated using the LSTM network of the decoder as

$$s_i = \text{LSTM}(s_{i-1}, y_i'; \lambda_D), \quad (9)$$

where y_i' is a word-embedding vector of y_m, and the initial state s_0 is obtained from the final encoder state h_L and $y_0' = \text{Embed}(\text{`<sos>`})$ as in Figure 1.

In the training phase, $Y = y_1, \ldots, y_M$ is given as the reference. However, in the test phase, the best word sequence needs to be found based on

$$\hat{Y} = \underset{Y \in V^*}{\arg\max}\, P(Y|X) \quad (10)$$

$$= \underset{y_1, \ldots, y_M \in V^*}{\arg\max}\, P(y_1|s_0) P(y_2|s_1) \cdots$$
$$P(y_M|s_{M-1}) P(\text{`<eos>`}|s_M). \quad (11)$$

Accordingly, we use a beam search in the test phase to keep multiple states and hypotheses with the highest cumulative probabilities at each mth step, and select the best hypothesis from those having reached the end-of-sentence token.
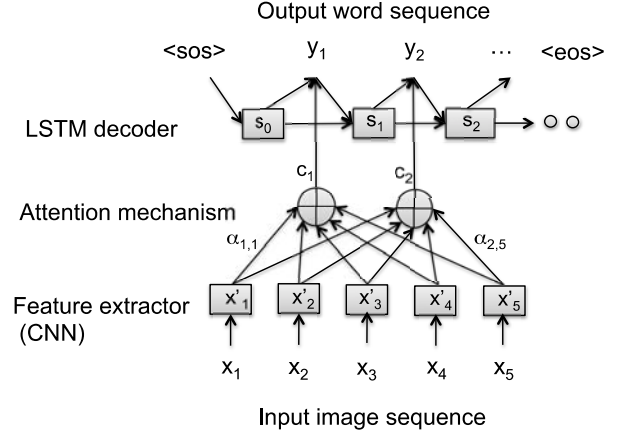

Figure 2. An encoder-decoder based sentence generator with temporal attention mechanism.

3. Attention-based sentence generator

Another approach to video description is an attention-based sequence generator [6], which enables the network to emphasize features from specific times or spatial regions depending on the current context, enabling the next word to be predicted more accurately. Compared to the basic approach described in Section 2, the attention-based generator can exploit input features selectively according to the input and output contexts. The efficacy of attention models has been shown in many tasks such as machine translation [1].

Figure 2 shows an example of the attention-based sentence generator from video, which has a temporal attention mechanism over the input image sequence. The input sequence of feature vectors is obtained using one or more feature extractors. Generally, attention-based generators employ an encoder based on a bidirectional LSTM (BLSTM) or Gated Recurrent Units (GRU) to further convert the feature vector sequence so that each vector contains its contextual information. In video description tasks, however, CNN-based features are often used directly, or one more feed-forward layer is added to reduce the dimensionality.

If we use an BLSTM encoder following the feature extraction, then the activation vectors (i.e., encoder states) are obtained as

$$h_t = \left[\begin{array}{c} h_t^{(f)} \\ h_t^{(b)} \end{array}\right], \quad (12)$$

where $h_t^{(f)}$ and $h_t^{(b)}$ are the forward and backward hidden activation vectors:

$$h_t^{(f)} = \text{LSTM}(h_{t-1}^{(f)}, x_t'; \lambda_E^{(f)}) \quad (13)$$

$$h_t^{(b)} = \text{LSTM}(h_{t+1}^{(b)}, x_t'; \lambda_E^{(b)}). \quad (14)$$

If we use a feed-forward layer, then the activation vector is

calculated as

$$h_t = \tanh(W_p x'_t + b_p), \qquad (15)$$

where W_p is a weight matrix and b_p is a bias vector. If we use the CNN features directly, then we assume $h_t = x'_t$.

The attention mechanism is realized by using *attention weights* to the hidden activation vectors throughout the input sequence. These weights enable the network to emphasize features from those time steps that are most important for predicting the next output word.

Let $\alpha_{i,t}$ be an attention weight between the ith output word and the tth input feature vector. For the ith output, the vector representing the relevant content of the input sequence is obtained as a weighted sum of hidden unit activation vectors:

$$c_i = \sum_{t=1}^{L} \alpha_{i,t} h_t. \qquad (16)$$

The decoder network is an Attention-based Recurrent Sequence Generator (ARSG) [1][6] that generates an output label sequence with content vectors c_i. The network also has an LSTM decoder network, where the decoder state can be updated in the same way as Equation (9).

Then, the output label probability is computed as

$$P(y|s_{i-1}, c_i) = \mathrm{softmax}(W_s^{(\lambda_D)} s_{i-1} + W_c^{(\lambda_D)} c_i + b_s^{(\lambda_D)}), \qquad (17)$$

and word y_i is generated according to

$$y_i = \underset{y \in V}{\arg\max}\, P(y|s_{i-1}, c_i). \qquad (18)$$

In contrast to Equations (7) and (8) of the basic encoder-decoder, the probability distribution is conditioned on the content vector c_i, which emphasizes specific features that are most relant to predicting each subsequent word. One more feed-forward layer can be inserted before the softmax layer. In this case, the probabilities are computed as follows:

$$g_i = \tanh\left(W_s^{(\lambda_D)} s_{i-1} + W_c^{(\lambda_D)} c_i + b_s^{(\lambda_D)}\right), \qquad (19)$$

and

$$P(y|s_{i-1}, c_i) = \mathrm{softmax}(W_g^{(\lambda_D)} g_i + b_g^{(\lambda_D)}). \qquad (20)$$

The attention weights are computed in the same manner as in [1]:

$$\alpha_{i,t} = \frac{\exp(e_{i,t})}{\sum_{\tau=1}^{L} \exp(e_{i,\tau})} \qquad (21)$$

and

$$e_{i,t} = w_A^\intercal \tanh(W_A s_{i-1} + V_A h_t + b_A), \qquad (22)$$

where W_A and V_A are matrices, w_A and b_A are vectors, and $e_{i,t}$ is a scalar.
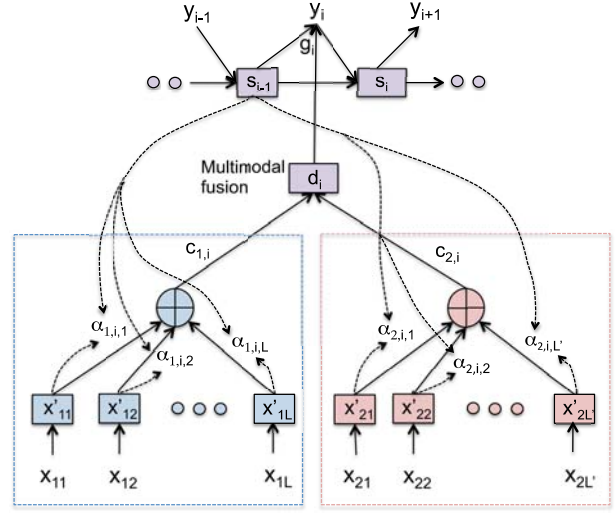

Figure 3. Simple feature fusion.

4. Attention-based multimodal fusion

This section proposes an attention model to handle fusion of multiple modalities, where each modality has its own sequence of feature vectors. For video description, multimodal inputs such as image features, motion features, and audio features are available. Furthermore, combination of multiple features from different feature extraction methods are often effective to improve the description accuracy.

In [29], content vectors from VGGNet (image features) and C3D (spatiotemporal motion features) are combined into one vector, which is used to predict the next word. This is performed in the fusion layer, in which the following activation vector is computed instead of Eq. (19),

$$g_i = \tanh\left(W_s^{(\lambda_D)} s_{i-1} + d_i + b_s^{(\lambda_D)}\right), \qquad (23)$$

where

$$d_i = W_{c1}^{(\lambda_D)} c_{1,i} + W_{c2}^{(\lambda_D)} c_{2,i}, \qquad (24)$$

and $c_{1,i}$ and $c_{2,i}$ are two different content vectors obtained using different feature extractors and/or different input modalities.

Figure 3 shows the simple feature fusion approach, in which content vectors are obtained with attention weights for individual input sequences $x_{11}, \ldots, x_{1L}$ and $x_{21}, \ldots, x_{2L'}$, respectively. However, these content vectors are combined with weight matrices W_{c1} and W_c2, which are commonly used in the sentence generation step. Consequently, the content vectors from each feature type (or one modality) are always fused using the same weights, independent of the decoder state. This architecture lacks the ability to exploit multiple types of features effectively, because it does not allow the relative weights of each feature type (of each modality) to change based on the context.
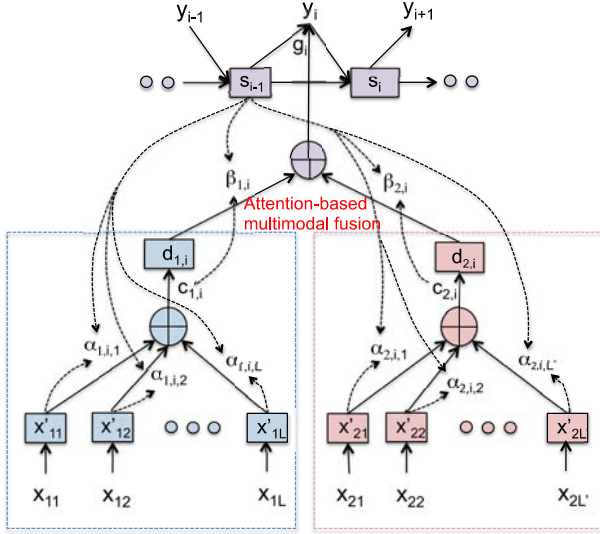
Figure 4. Our multimodal attention mechanism.

This paper extends the attention mechanism to multimodal fusion. Using this multimodal attention mechanism, based on the current decoder state, the decoder network can selectively attend to specific modalities of input (or specific feature types) to predict the next word. Let K be the number of modalities, i.e., the number of sequences of input feature vectors. Our attention-based feature fusion is performed using

$$g_i = \tanh \left(W_s^{(\lambda_D)} s_{i-1} + \sum_{k=1}^K \beta_{k,i} d_{k,i} + b_s^{(\lambda_D)} \right), \quad (25)$$

where

$$d_{k,i} = W_{ck}^{(\lambda_D)} c_{k,i} + b_{ck}^{(\lambda_D)}. \quad (26)$$

The multimodal attention weights $\beta_{k,i}$ are obtained in a similar way to the temporal attention mechanism:

$$\beta_{k,i} = \frac{\exp(v_{k,i})}{\sum_{\kappa=1}^K \exp(v_{\kappa,i})}, \quad (27)$$

where

$$v_{k,i} = w_B^\mathsf{T} \tanh(W_B s_{i-1} + V_{Bk} c_{k,i} + b_{Bk}), \quad (28)$$

where W_B and V_{Bk} are matrices, w_B and b_{Bk} are vectors, and $v_{k,i}$ is a scalar.

Figure 4 shows the architecture of our sentence generator, including the multimodal attention mechanism. Unlike the simple multimodal fusion method in Figure 3, in Figure 4, the feature-level attention weights can change according to the decoder state and the content vectors, which enables the decoder network to pay attention to a different set of features and/or modalities when predicting each subsequent word in the description.

5. Experiments

5.1. Dataset

We evaluated our proposed feature fusion using the Youtube2Text video corpus [8]. This corpus is well suited for training and evaluating automatic video description generation models. The dataset has 1,970 video clips with multiple natural language descriptions. Each video clip is annotated with multiple parallel sentences provided by different Mechanical Turkers. There are 80,839 sentences in total, with about 41 annotated sentences per clip. Each sentence on average contains about 8 words. The words contained in all the sentences constitute a vocabulary of 13,010 unique lexical entries. The dataset is open-domain and covers a wide range of topics including sports, animals and music. Following [38], we split the dataset into a training set of 1,200 video clips, a validation set of 100 clips, and a test set consisting of the remaining 670 clips.

5.2. Video Preprocessing

The image data are extracted from each video clip, which consist of 24 frames per second, and rescaled to 224×224 pixel images. For extracting image features, a pretrained GoogLeNet [15] CNN is used to extract fixed-length representation with the help of the popular implementation in Caffe [11]. Features are extracted from the hidden layer pool5/7x7_s1. We select one frame out of every 16 frames from each video clip and feed them into the CNN to obtain 1024-dimensional frame-wise feature vectors.

We also use a VGGNet [20] that was pretrained on the ImageNet dataset [14]. The hidden activation vectors of fully connected layer fc7 are used for the image features, which produces a sequence of 4096-dimensional feature vectors. Furthermore, to model motion and short-term spatiotemporal activity, we use the pretrained C3D [22] (which was trained on the Sports-1M dataset [13]). The C3D network reads sequential frames in the video and outputs a fixed-length feature vector every 16 frames. We extracted activation vectors from fully-connected layer fc6-1, which has 4096-dimensional features.

5.3. Audio Processing

Unlike previous methods that use the YouTube2Text dataset [28, 18, 29], we also incorporate audio features, to use in our attention-based feature fusion method. Since YouTube2Text corpus does not contain audio track, we extracted the audio data via the original video URLs. Although a subset of the videos were no longer available on YouTube, we were able to collect the audio data for 1,649 video clips, which covers 84% of the corpus. The 44 kHz-sampled audio data are down-sampled to 16 kHz, and Mel-Frequency Cepstral Coefficients (MFCCs) are extracted from each 50 ms time window with 25 ms shift. The

Table 1. Evaluation results on the YouTube2Text test set. The last three rows of the table present previous state-of-the-art methods, which use only temporal attention. The rest of the table shows results from our own implementations. The first three rows of the table use temporal attention but only one modality (one feature type). The next two rows do multimodal fusion of two modalities (image and spatiotemporal) using either Simple Multimodal fusion (see Figure 3) or our proposed Multimodal Attention mechanism (see Figure 4). The next two rows also perform multimodal fusion, this time of three modalities (image, spatiotemporal, and audio features). In each column, the scores of the top two methods are shown in boldface.

Fusion method		Modalities (feature types)			Evaluation metric					
	Attention	Image	Spatiotemporal	Audio	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	CIDEr
Unimodal	Temporal	GoogLeNet			0.763	0.644	0.549	0.449	0.294	0.534
Unimodal	Temporal	VGGNet			0.789	0.673	0.578	0.472	0.305	0.595
Unimodal	Temporal		C3D		0.784	0.664	0.568	0.467	0.296	0.549
Simple Multimodal	Temporal	VGGNet	C3D		0.800	0.679	0.581	0.478	0.309	0.614
Multimodal Attention	**Temporal & Multimodal**	VGGNet	C3D		**0.811**	**0.703**	**0.607**	**0.503**	**0.318**	0.634
Simple Multimodal	Temporal	VGGNet	C3D	MFCC	0.774	0.649	0.552	0.448	0.306	0.578
Multimodal Attention	**Temporal & Multimodal**	VGGNet	C3D	MFCC	0.801	0.688	0.593	0.496	0.310	**0.655**
TA [28]	Temporal	GoogLeNet	3D CNN		0.800	0.647	0.526	0.419	0.296	0.517
LSTM-E [18]		VGGNet	C3D		0.788	0.660	0.554	0.453	0.310	-
h-RNN [29]	Temporal	VGGNet	C3D		**0.815**	**0.704**	**0.604**	**0.499**	**0.326**	**0.658**

sequence of 13-dimensional MFCC features are then concatenated into one vector from every group of 20 consecutive frames, which results in a sequence of 260-dimensional vectors. The MFCC features are normalized so that the mean and variance vectors are 0 and 1 in the training set. The validation and test sets are also adjusted with the original mean and variance vectors of the training set. Unlike with the image features, we apply a BLSTM encoder network for MFCC features, which is trained jointly with the decoder network. If audio data are missing for a video clip, then we feed in a sequence of dummy MFCC features, which is simply a sequence of zero vectors.

5.4. Experimental Setup

The caption generation model, i.e. the decoder network, is trained to minimize the cross entropy criterion using the training set. Image features are fed to the decoder network through one projection layer of 512 units, while audio features, i.e. MFCCs, are fed to the BLSTM encoder followed by the decoder network. The encoder network has one projection layer of 512 units and bidirectional LSTM layers of 512 cells. The decoder network has one LSTM layer with 512 cells. Each word is embedded to a 256-dimensional vector when it is fed to the LSTM layer. We apply the AdaDelta optimizer [30] to update the parameters, which is widely used for optimizing attention models. The LSTM and attention models were implemented using Chainer [21].

The similarity between ground truth and automatic video description results are evaluated using machine-translation-motivated metrics: BLEU [19], METEOR [7], and the newly proposed metric for image description, CIDEr [23]. We used the publicly available evaluation script prepared

for image captioning challenge [3].

5.5. Results and Discussion

Table 1 shows the evaluation results on the Youtube2text data set. We compared the performance for our multimodal attention model (Multimodal Attention) which integrated temporal and multimodal attention mechanisms with a simple additive multimodal fusion (Simple Multimodal), unimodal models with temporal attention (Unimodal), and baseline systems that used temporal attention.

The Simple Multimodal model performed better than the Unimodal models. However the proposed Multimodal Attention model outperformed Simple Multimodal. The audio feature degrades the performance of the baseline because some YouTube data includes noise such as background music, which is unrelated to the video content. As expected, the Multimodal Attention model mitigated the impact of the noise of the audio features. Moreover, combining the audio features using our proposed method reached the best performance of CIDEr for all experimental conditions.

In contrast to the existing systems, our temporal attention system which used only static image features (Unimodal) outperformed TA using combination of static image and dynamic video features [28]. Our proposed attention mechanisms outperformed LSTM-E [18] which does not use attention mechanisms. Our Simple Multimodal system using temporal attention is the same basic structure used by h-RNN as well as the same features extracted from VGGNet [20] and C3D [22]. However, Simple Multimodal did not perform as well as h-RNN, which may indicate that it was not as well tuned as h-RNN. While we employed AdaDelta for optimizers and LSTM-based RNN for decoder

networks, h-RNN employed RMSprop and Gated Recurrent Units (GRUs). Importantly, our Multimodal Attention model improved upon Simple Multimodal and achieved comparable performance with h-RNN.

6. Conclusion

We proposed a new modality-dependent attention mechanism, which we call multimodal attention, for video description based on encoder-decoder sentence generation using recurrent neural networks (RNNs). In this approach, the attention model selectively attends not just to specific times, but to specific modalities of input such as image features, spatiotemporal motion features, and audio features. This approach provides a natural way to fuse multimodal information for video description. We evaluate our method on the Youtube2Text dataset, achieving results that are competitive with current state-of-the-art methods that employ temporal attention models, in which the decoder network predicts each word in the description by selectively giving more weight to encoded features from specific time frames. More importantly, we demonstrate that our model incorporating multimodal attention as well as temporal attention outperforms the model that uses temporal attention alone.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[2] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio. End-to-end attention-based large vocabulary speech recognition. pages 4945–4949, 2016.

[3] X. Chen, H. Fang, T. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325, 2015.

[4] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014.

[5] K. Cho, B. van Merrienboer, 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 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734, 2014.

[6] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 577–585. Curran Associates, Inc., 2015.

[7] M. J. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*, pages 376–380, 2014.

[8] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2712–2719, 2013.

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

[10] T. Hori, H. Wang, C. Hori, S. Watanabe, B. Harsham, J. L. Roux, J. Hershey, Y. Koji, Y. Jing, Z. Zhu, and T. Aikawa. Dialog state tracking with attention-based sequence-to-sequence learning. In *2016 IEEE Spoken Language Technology Workshop, SLT 2016, San Diego, CA, USA, December 13-16, 2016*.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[12] Q. Jin, J. Liang, and X. Lin. Generating Natural Video Descriptions via Multimodal Processing. In *Interspeech*, 2016.

[13] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[15] M. Lin, Q. Chen, and S. Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[16] R. Lowe, N. Pow, I. Serban, and J. Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2-4 September 2015, Prague, Czech Republic*, pages 285–294, 2015.

[17] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *CoRR*, abs/1412.6632, 2014.

[18] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. *CoRR*, abs/1505.01861, 2015.

[19] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 311–318, 2002.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[21] S. Tokui, K. Oono, S. Hido, and J. Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (Learn-*

ingSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS), 2015.

[22] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 4489–4497, 2015.

[23] R. Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575, 2015.

[24] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. J. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1494–1504, 2015.

[25] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3156–3164, 2015.

[26] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 3169–3176, Colorado Springs, United States, June 2011.

[27] 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. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 2048–2057, 2015.

[28] L. Yao, A. Torabi, K. Cho, N. Ballas, C. J. Pal, H. Larochelle, and A. C. Courville. Describing videos by exploiting temporal structure. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 4507–4515, 2015.

[29] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. *CoRR*, abs/1510.07712, 2015.

[30] M. D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.