

Attention-Based Multimodal Fusion for Video Description

Chiori Hori Takaaki Hori Teng-Yok Lee Ziming Zhang
Bret Harsham John R. Hershey Tim K. Marks Kazuhiko Sumi*

Mitsubishi Electric Research Laboratories (MERL)

{chori, thori, tlee, zzhang, harsham, hershey, tmarks}@merl.com, sumi@it.aoyama.ac.jp

Abstract

Current methods for video description are based on encoder-decoder sentence generation using recurrent neural networks (RNNs). Recent work has demonstrated the advantages of integrating temporal 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. Such methods typically use two different types of features: image features (from an object classification model), and motion features (from an action recognition model), combined by naïve concatenation in the model input. Because different feature modalities may carry task-relevant information at different times, fusing them by naïve concatenation may limit the model’s ability to dynamically determine the relevance of each type of feature to different parts of the description. In this paper, we incorporate audio features in addition to the image and motion features. To fuse these three modalities, we introduce a multimodal attention model that can selectively utilize features from different modalities for each word in the output description. Combining our new multimodal attention model with standard temporal attention outperforms state-of-the-art methods on two standard datasets: YouTube2Text and MSR-VTT.

1. Introduction

Automatic video description, also known as video captioning, refers to the automatic generation of a natural language description, such as 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.

Recent work in video description has demonstrated the ad-

vantages of integrating temporal attention mechanisms into encoder-decoder neural networks, in which the decoder network predicts each word in the description by selectively giving more weight to encoded features from different times in the video. Typically, two different types of features are used: image features (learned from an object classification task), and motion features (learned from an action recognition task). These are combined by naïve concatenation in the input to the video description model. Because different feature modalities may carry task-relevant information at different times, fusing them by naïve concatenation may limit the model’s ability to dynamically determine the relevance of each type of feature to different parts of the description. In this paper, we expand the feature set to include the audio modality, in addition to the image and motion features.

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. Different modalities of input may be important for selecting each word 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 complementary, 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.

*On sabbatical from Aoyama Gakuin University.

A longstanding area of research addresses how to effectively combine information from multiple modalities for machine perception tasks [10]. Previous methods typically used stream weights (e.g., [8]) or Bayesian adaptation approaches (e.g., [21]). As far as we know, our approach is the first to fuse multimodal information using attention between modalities in a neural network. Our method dynamically adjusts the relative importance of each modality to generate better descriptions. The benefits of attentional multimodal fusion include: (1) the modalities that are most helpful to discriminate each word in the description can dynamically receive a stronger weight, and (2) the network can detect interference (e.g., noise) and other sources of uncertainty in each modality and dynamically down-weight the modalities that are less certain. Not only does our proposed method achieve these benefits, but it does so using a model that can be discriminatively trained end-to-end.

In this work, we present results of video description on two large datasets: YouTube2Text and the subset of MSR-VTT that was still available at the time of the experiments. We show that combining our new multimodal attention model with temporal attention outperforms state-of-the-art methods, which are based on temporal attention alone.

2. Related Work

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 [30, 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. They employed a recurrent neural network (RNN)-based soft attention model that enables 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 [30], and dialog management [19].

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 [20, 30]. To improve performance, [33] 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 [29]. In this task, the inputs to the encoder network are video information features that may include static im-

age features extracted using convolutional neural networks (CNNs), temporal dynamics of videos extracted using spatiotemporal 3D CNNs [27], dense trajectories [31], optical flow, and audio features [15]. From the encoder outputs, the decoder network generates word sequences using recurrent neural networks (RNNs) with long short-term memory (LSTM) units [11] 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, the order in which objects and actions appear over time in the video may be different from their order in the sentence. When choosing the right words to describe something, the features that directly correspond to that object or action are most relevant, and other features may be 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 [33], and dialog management [14]. In recent work, these attention mechanisms have been applied to video description [34, 35]. 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.

We first described the proposed method in an arXiv paper [12]. In this paper, we expand upon [4] by testing on an additional dataset and precisely analyzing the significance of the improvements due to our method. The approach we describe here is not limited to the modalities of video and audio. It could also be applied to other types of sources, such as text for machine translation and summarization, or to information from multiple sensors to predict user status (e.g., driver confusion) [13]. In this work, we tested our attentional multimodal fusion using MSR-VTT and precisely analyzed significance of improvements.

2.1. Encoder-Decoder-Based Sentence Generator

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

Given a sequence of images, $X = x_1, x_2, \dots, x_L$, 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 [18], VGG-16 [24], or C3D [27]. The sequence of image features, $X' = x'_1, x'_2, \dots, x'_L$, 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 state of the LSTM is given by

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

where $\text{LSTM}(h, x; \lambda)$ represents an LSTM function of hidden and input vectors h and x , which is computed with parameters λ . In Eq. (1), λ_E denotes the encoder's parameters.

The decoder predicts the next word iteratively beginning with the start-of-sentence token, $\langle \text{sos} \rangle$, until it predicts the end-of-sentence token, $\langle \text{eos} \rangle$. Given decoder state s_{i-1} , the decoder network λ_D infers the next word probability distribution as

$$P(y|s_{i-1}) = \text{softmax} \left(W_s^{(\lambda_D)} s_{i-1} + b_s^{(\lambda_D)} \right), \quad (2)$$

and generates the word y_i that has the highest probability according to

$$y_i = \underset{y \in V}{\text{argmax}} P(y|s_{i-1}), \quad (3)$$

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

$$s_i = \text{LSTM}(s_{i-1}, y'_i; \lambda_D), \quad (4)$$

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 $y'_0 = \text{Embed}(\langle \text{sos} \rangle)$.

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

$$\begin{aligned} \hat{Y} &= \underset{Y \in V^*}{\text{argmax}} P(Y|X) \\ &= \underset{y_1, \dots, y_M \in V^*}{\text{argmax}} P(y_1|s_0)P(y_2|s_1) \cdots \\ &\quad P(y_M|s_{M-1})P(\langle \text{eos} \rangle|s_M). \end{aligned} \quad (5)$$

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

2.2. 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

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

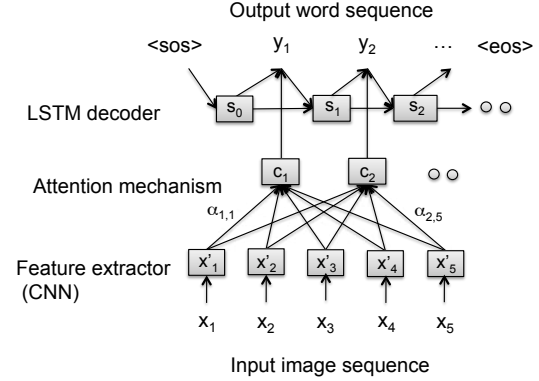


Figure 1. An encoder-decoder based sentence generator with temporal attention mechanism.

predicted more accurately. Compared to the basic approach described in Section 2.1, 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 1 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

$$h_t = \begin{bmatrix} h_t^{(f)} \\ h_t^{(b)} \end{bmatrix}, \quad (6)$$

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

$$h_t^{(f)} = \text{LSTM}(h_{t-1}^{(f)}, x'_t; \lambda_E^{(f)}) \quad (7)$$

$$h_t^{(b)} = \text{LSTM}(h_{t+1}^{(b)}, x'_t; \lambda_E^{(b)}). \quad (8)$$

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

$$h_t = \tanh(W_p x'_t + b_p), \quad (9)$$

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 in-

put 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 i th output word and the t th input feature vector. For the i th output, the vector representing the relevant content of the input sequence is obtained as a weighted sum of hidden unit activation vectors:

$$c_i = \sum_{t=1}^L \alpha_{i,t} h_t. \quad (10)$$

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 (4).

Then, the output label probability is computed as

$$P(y|s_{i-1}, c_i) = \text{softmax} \left(W_s^{(\lambda_D)} s_{i-1} + W_c^{(\lambda_D)} c_i + b_s^{(\lambda_D)} \right), \quad (11)$$

and word y_i is generated according to

$$y_i = \underset{y \in V}{\text{argmax}} P(y|s_{i-1}, c_i). \quad (12)$$

In contrast to Equations (2) and (3) of the basic encoder-decoder, the probability distribution is conditioned on the content vector c_i , which emphasizes specific features that are most relevant 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:

$$g_i = \tanh \left(W_s^{(\lambda_D)} s_{i-1} + W_c^{(\lambda_D)} c_i + b_s^{(\lambda_D)} \right), \quad (13)$$

and

$$P(y|s_{i-1}, c_i) = \text{softmax} (W_g^{(\lambda_D)} g_i + b_g^{(\lambda_D)}). \quad (14)$$

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

$$\alpha_{i,t} = \frac{\exp(e_{i,t})}{\sum_{\tau=1}^L \exp(e_{i,\tau})} \quad (15)$$

and

$$e_{i,t} = w_A^T \tanh(W_A s_{i-1} + V_A h_t + b_A), \quad (16)$$

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

3. Attention-Based Multimodal Fusion

We propose 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, combinations of multiple

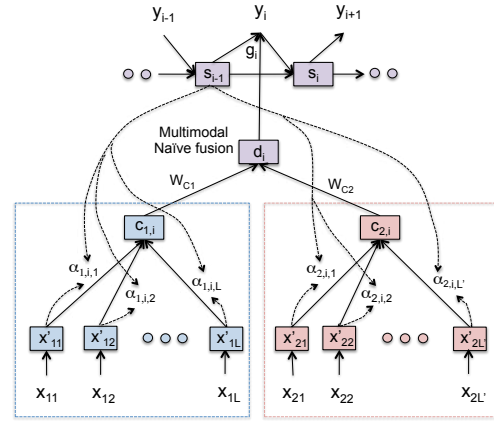


Figure 2. Naïve Fusion of multimodal features.

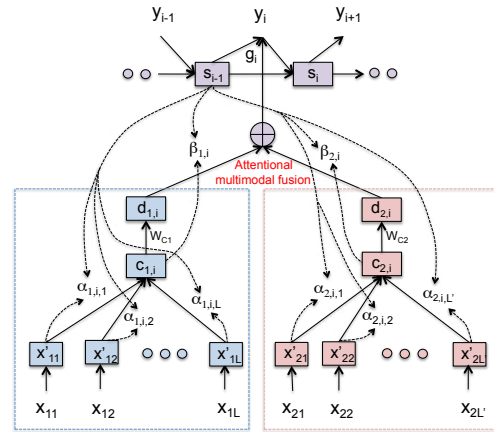


Figure 3. Our Attentional Fusion of multimodal features.

features from different feature extraction methods are often effective to improve the accuracy of descriptions.

In [35], content vectors from VGG-16 (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. (13):

$$g_i = \tanh \left(W_s^{(\lambda_D)} s_{i-1} + d_i + b_s^{(\lambda_D)} \right), \quad (17)$$

where

$$d_i = W_{c1}^{(\lambda_D)} c_{1,i} + W_{c2}^{(\lambda_D)} c_{2,i}, \quad (18)$$

and $c_{1,i}$ and $c_{2,i}$ are two feature vectors obtained using different feature extractors and/or different input modalities. Figure 2 illustrates this approach, which we call Naïve Fusion, in which multimodal feature vectors are combined using one projection matrix W_{c1} for the first modality (input sequence x_{11}, \dots, x_{1L}), and a different projection matrix W_{c2} for the second modality (input sequence $x'_{21}, \dots, x'_{2L'}$).

However, these feature vectors are combined in the sentence generation step with projection matrices W_{c1} and

W_{c2} , which do not depend on time. Consequently, for each modality (or each feature type), all of the feature vectors from that modality are given the same weight during fusion, independent of the decoder state. Note that Naïve Fusion is a type of late fusion, because the inherent difference in sampling rate of the three feature streams precludes early fusion (concatenation of input features). The Naïve Fusion architecture lacks the ability to exploit multiple types of features effectively, because it does not allow the relative weights of each modality (of each feature type) to change based on the context of each word in the sentence.

Our proposed method extends the attention mechanism to multimodal fusion. We call it *attentional fusion*, or *multimodal attention*. In our model, 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

$$g_i = \tanh \left(W_s^{(\lambda_D)} s_{i-1} + \sum_{k=1}^K \beta_{k,i} d_{k,i} + b_s^{(\lambda_D)} \right), \quad (19)$$

where

$$d_{k,i} = W_{ck}^{(\lambda_D)} c_{k,i} + b_{ck}^{(\lambda_D)}. \quad (20)$$

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

$$\beta_{k,i} = \frac{\exp(v_{k,i})}{\sum_{\kappa=1}^K \exp(v_{\kappa,i})}, \quad (21)$$

where

$$v_{k,i} = w_B^T \tanh(W_B s_{i-1} + V_{Bk} c_{k,i} + b_{Bk}), \quad (22)$$

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

Figure 3 shows the architecture of our sentence generator, including the multimodal attention mechanism. Unlike in the Naïve multimodal fusion method shown in Figure 2, in our method (shown in Figure 3) the multimodal attention weights can change according to the decoder state and the feature vectors. This enables the decoder network to attend to a different set of features and/or modalities when predicting each subsequent word in the description.

4. Experiments

4.1. Datasets

We evaluated our proposed feature fusion using the YouTube2Text [9] and MSR-VTT [32] video datasets. YouTube2Text has 1,970 video clips with multiple natural language descriptions. 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.

MSR-VTT [32] consists of 10,000 web video clips with 41.2 hours and 200,000 clip-sentence pairs in total, covering a comprehensive list of 20 categories and a wide variety of video content. Each clip was annotated with about 20 natural sentences. The dataset is split into training, validation, and testing sets of 65%, 5%, 30%, corresponding to 6,513, 497, and 2,990 clips respectively. However, because the video clips are hosted on YouTube, some of the MSR-VTT videos have been removed due to content or copyright issues. At the time we downloaded the videos (February 2017), approximately 12% were unavailable. Thus, we trained and tested our approach using just the subset of the MSR-VTT dataset that were available, which consist of 5,763, 419, and 2,616 clips for train, validation, and test respectively.

4.2. Video Processing

The image data are extracted from each video clip at 24 frames per second and rescaled to 224×224-pixel images. For extracting image features, we use a VGG-16 network [24] that was pretrained on the ImageNet dataset [17]. The hidden activation vectors of fully connected layer fc7 are used for the image features, which produces a sequence of 4096-dimensional feature vectors. To model motion and short-term spatiotemporal activity, we use the pretrained C3D [27] (which was trained on the Sports-1M dataset [16]). 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.

4.3. Audio Processing

Unlike previous methods that use the YouTube2Text dataset [34, 22, 35], we additionally incorporate audio features. Since the packaged YouTube2Text dataset does not include the audio from the YouTube videos, we extracted the audio data via the original video URLs. Although some of the videos were no longer available on YouTube, we were able to collect audio data for 1,649 video clips, which is 84% of the dataset. 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 sequence of 13-dimensional MFCC features are then concatenated into one vector for every group of 20 consecutive frames, resulting in a sequence of 260-dimensional vectors. The MFCC features are normalized so

Table 1. Evaluation results on the YouTube2Text test set. The top three rows of the upper table present results of previous state-of-the-art methods for YouTube2Text, which use only visual features and only temporal attention. The rest of the tables show results from our own implementations. Naïve Fusion indicates the conventional approach using temporal attention only (see Figure 2). Attentional Fusion is our proposed Modal-attention approach (see Figure 3). The symbol (V) denotes methods that only use the visual modalities (image features and spatiotemporal features). The symbol (AV) denotes our methods that use all three modalities (audio features as well as the two types of video features). Our baseline method “Naïve Fusion (V)” is very similar to the approach of [35]. In the second table, we evaluate our methods on the subset of the YouTube2Text videos whose audio is not obscured by overdubbed music.

YouTube2Text Full Dataset							
Method	Attention	Modalities (feature types)			Evaluation metric		
		Image	Spatiotemporal	Audio	BLEU4	METEOR	CIDEr
LSTM-E [22]		VGG-16	C3D		0.453	0.310	–
TA [34]	Temporal	GoogLeNet	3D CNN		0.419	0.296	0.517
h-RNN [35]	Temporal	VGG-16	C3D		0.499	0.326	0.658
Naïve Fusion (V)	Temporal	VGG-16	C3D		0.515	0.313	0.659
Naïve Fusion (AV)	Temporal	VGG-16	C3D	MFCC	0.506	0.309	0.637
Attentional Fusion (V)	Temporal & Multimodal	VGG-16	C3D		0.524	0.320	0.688
Attentional Fusion (AV)	Temporal & Multimodal	VGG-16	C3D	MFCC	0.539	0.322	0.674

YouTube2Text Subset without Overdubbed Music							
Method	Attention	Modalities (feature types)			Evaluation metric		
		Image	Spatiotemporal	Audio	BLEU4	METEOR	CIDEr
Naïve Fusion (V)	Temporal	VGG-16	C3D		0.527	0.333	0.695
Naïve Fusion (AV)	Temporal	VGG-16	C3D	MFCC	0.534	0.331	0.695
Attentional Fusion (V)	Temporal & Multimodal	VGG-16	C3D		0.549	0.342	0.704
Attentional Fusion (AV)	Temporal & Multimodal	VGG-16	C3D	MFCC	0.568	0.343	0.724

Table 2. Evaluation results on MSR-VTT Subset. Approximately 12% of the MSR-VTT videos have been removed from YouTube, so we train and test on the remaining Subset of MSR-VTT videos that we were able to download. We cannot directly compare with the results in [32], because they used the full MSR-VTT dataset. Our Naïve Fusion (V) baseline method is extremely similar to the method of [32], so it may be viewed as our implementation of their method using the available subset of the MSR-VTT dataset.

MSR-VTT Subset							
Fusion method	Attention	Modalities (feature types)			Evaluation metric		
		Image	Spatiotemporal	Audio	BLEU4	METEOR	CIDEr
Naïve Fusion (V)	Temporal	VGG-16	C3D		0.379	0.242	0.379
Naïve Fusion (AV)	Temporal	VGG-16	C3D	MFCC	0.376	0.240	0.332
Attentional Fusion (V)	Temporal & Multimodal	VGG-16	C3D		0.394	0.257	0.404
Attentional Fusion (AV)	Temporal & Multimodal	VGG-16	C3D	MFCC	0.397	0.255	0.400

that the mean and variance vectors are 0 and 1 in the training set. The validation and test sets are also adjusted using the original mean and variance vectors from the training set. Unlike for the image features, we apply a BLSTM encoder network for MFCC features, which is trained jointly with the decoder network. If audio data are not available for a video clip, then we feed in a sequence of dummy MFCC features, which is simply a sequence of zero vectors.





4.4. Experimental Setup

The similarity between ground truth (human-generated) and automatic video description results is evaluated using two metrics that were motivated by machine translation, BLEU [23] and METEOR [7], as well as a newly proposed metric for image description, CIDEr [28]. We used the publicly available evaluation script prepared for the image captioning challenge [3]. Each video in YouTube2Text has multiple “ground-truth” descriptions, but *some* “ground-

truth” answers are incorrect. Since BLEU and METEOR scores for a video do not consider frequency of words in the ground truth, they can be strongly affected by one incorrect ground-truth description. METEOR is even more susceptible, since it also accepts paraphrases of incorrect ground-truth words. In contrast, CIDEr is a voting-based metric that is robust to errors in ground truth.

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 compared the AdaDelta optimizer [36] and RMSprop [25] to update the

Table 3. Sample video description results on YouTube2Text. The first row of descriptions were generated by a unimodal system with only image features (VGG-16) and temporal attention. The other model names are the same as in Table 1.

Sample Image				
Unimodal (VGG-16)	a monkey is running	a man is slicing a potato	a woman is riding a horse	a man is singing
Naïve Fusion (V)	a dog is playing	a woman is cutting an onion	a girl is riding a horse	a man is singing
Naïve Fusion (AV)	a monkey is running	a woman is peeling an onion	a girl is riding a horse	a man is playing a guitar
Attentional Fusion (V)	a monkey is pulling a dogs tail	a man is slicing a potato	a man is riding a horse	a man is playing a guitar
Attentional Fusion (AV)	a monkey is playing	a woman is peeling an onion	a girl is riding a horse	a man is playing a violin
Discussion	Attentional Fusion (V) (i.e., Multimodal attention on visual features) worked best.	Our inclusion of audio features enabled the “peeling” action to be identified.	Attentional fusion is best. Audio hurts performance due to overdubbed music.	Both audio features and multimodal attention are needed to identify “violin”.

parameters, which is widely used for optimizing attention models. In this video description task, we used L2 regularization for all experimental conditions and compared RMSprop and AdaDelta. RMSprop outperformed Adadelta for all experimental conditions, so we report the results using RMSprop in Tables 1 and 2. The LSTM and attention models were implemented using Chainer [26].

5. Results and Discussion

Tables 1 and 2 show the evaluation results on the YouTube2Text and MSR-VTT Subset datasets. On each dataset, we compare the performance of our multimodal attention model (Attentional Fusion), which integrates temporal and multimodal attention mechanisms, to a naïve additive multimodal fusion (Naïve Fusion). We test versions of our system that use only visual (image and spatiotemporal) features “(V)”, and versions that additionally use audio features “(AV)”. Our baseline system is the “Naïve Fusion (V)” method that uses only temporal attention and only visual features (no audio). This baseline is extremely similar to the methods used in [35] and [32], which are the current state-of-the-art methods on the two datasets.

The results demonstrate the effectiveness of our proposed model. In Table 1, the proposed methods outperform the previously published results in all but one evaluation metric of one previous method. In both Tables 1 and 2, our proposed methods outperform the “Naïve Fusion (V)” baseline, which is our implementation of the state-of-the-art methods [35] and [32]. Furthermore, our proposed Attentional Fusion model outperforms the corresponding Naïve Fusion model, both with audio features (AV) and without audio features (V), on both datasets. These results clearly demonstrate the benefits of our proposed multimodal attention model. Table 3 shows generated descriptions for four example videos from the YouTube2Text data set. These and more examples, including the original videos with sound, are in the supplementary material.

5.1. Significance of Improvements

To understand performance improvements via the metrics, we measured the relative improvement in performance, defined as $P = (Proposed - Baseline) / Baseline$, where *Proposed* is the score for Attentional Fusion (AV), and *Baseline* refers to Naïve Fusion (AV). The relative improvements P for all metrics on the YouTube2Text Full Dataset and MSR-VTT Subset are shown in part (A) of Table 4. The use of relative scores highlights the significance of the improvements due to Attentional Fusion. In addition, to establish an upper bound related to human performance, we evaluated inter-rater reliability of the human captions in leave-one-out fashion: we compared each reference sentence for each video to the remaining set of reference sentences for that video, using all three metrics. The mean of these “Human” scores are shown in part (B) of Table 4. Our scores are quite close to this inter-rater reliability upper bound. Furthermore, our model scores significantly close the gap between the baseline and this “Human” upper bound. We can quantify the gap in terms of the relative reduction, R , defined as $R = (Proposed - Baseline) / (Human - Baseline)$. The relative gap reduction, R , for all metrics is shown in part (C) of Table 4. These scores indicate that our model makes significant progress from the baseline toward human-level performance. Note that for BLEU4 on the MSR-VTT Subset, both the baseline and our system are “super-human” by this standard, so there is no gap to close. Nevertheless, our model still outperforms the “Naïve Fusion” baseline.

5.2. Impact of Audio Features

In some experiments, including audio features (AV) improves performance over the corresponding visual-only (V) case, but in other cases it does not. Including audio features can degrade performance for some video clips because some YouTube videos include unrelated noise that was not in the original scene, such as overdubbed music that was added to the video in post-production. Attentional Fusion

Table 4. Significance of Improvement by Attentional Fusion (AV) in terms of (A) Relative Improvement, P , compared to the Naïve Fusion (AV) baseline, (B) Mean of the “Human” Scores, and (C) Relative Gap Reduction, R , compared to the “Human” Scores.

Data set	BLEU4	METEOR	CIDEr
(A) Relative Improvement in Performance, P			
YouTube2Text Full Dataset	6.5%	4.2%	5.8%
MSR-VTT Subset	5.6%	6.3%	20.5%
(B) Mean of the “Human” Scores			
YouTube2Text Full Dataset	0.56	0.42	1.19
MSR-VTT Subset	0.34	0.30	0.50
(C) Relative Gap Reduction to Human, R			
YouTube2Text Full Dataset	63%	11%	7%
MSR-VTT Subset	NA	27%	40%

mitigated the degradation by the audio feature. On the other hand, the audio feature contributed to the performance for both Naïve and Attentional fusion models.

We found the negative impact of audio features on some evaluation metrics—i.e., cases in which (AV) methods perform worse than their (V) counterparts in Tables 1 and 2. We hypothesized that this degradation due to audio features was due to overdubbed sound that was not present in the original scene. To test this hypothesis, we performed an experiment in which we manually removed all of the YouTube2Text videos in which overdubbed music obscured the sound that was captured during filming. The subsection of Table 1 titled “YouTube2Text Subset without Overdubbed Music” shows the results for the remaining subset of YouTube2Text (380 videos). The results show that whereas the Naïve fusion baseline did not make good use of the audio features in these videos, our proposed Attentional Fusion method does, yielding a significant score improvement over the baseline for all metrics.

5.3. Impact of Multimodal Attention

A particular advantage of the proposed multimodal attention is that we can easily inspect the attention distributions over modalities produced by the network for each word. Table 5 shows the average attention weights used for each modality when generating various words, sorted in descending order by weight. The image features, which were trained for object classification (VGG-16 ImageNet), are strongly selected for the words that describe generic object types. The motion features (C3D), which were trained to identify different sports scenes, appear to be selected when describing objects and scenes that tend to be in motion, such as sports and vehicles. The audio features, which were not pretrained (MFCC), overall have smaller weights and were less strongly selected. Nevertheless, the words with the strongest audio weights appear to be action verbs associated with sound, such as talking, singing, and driving. Thus the overall pattern of weights is consistent with our hypothesis about the role of attention to different modalities in selecting different types of words.

Table 5. A list of words with strong average attention weights for each modality, obtained on the the MSR-VTT Subset using our “Attentional Fusion (AV)” multimodal attention method.

Image (VGG-16)		Motion (C3D)		Audio (MFCC)	
bowl	0.9701	track	0.9887	talking	0.3435
pan	0.9426	motorcycle	0.9564	shown	0.3072
recipe	0.9209	baseball	0.9378	playing	0.2599
piece	0.9136	football	0.9275	singing	0.2465
paper	0.9098	horse	0.9212	driving	0.2284
kitchen	0.8827	soccer	0.9099	working	0.2004
toy	0.8758	basketball	0.9096	walking	0.1999
folding	0.8423	tennis	0.8958	riding	0.1900
makeup	0.8326	player	0.8720	showing	0.1836
guitar	0.7723	two	0.8345	dancing	0.1832
applying	0.7691	video	0.8237	wrestling	0.1735
food	0.7547	men	0.8198	running	0.1689
making	0.7470	running	0.7680	applying	0.1664
cooking	0.7464	wrestling	0.7462	cooking	0.1646
working	0.6837	people	0.7374	making	0.1636
showing	0.6229	stroller	0.7314	characters	0.1245
computer	0.5837	game	0.7293	folding	0.1079
band	0.5791	group	0.7205	program	0.0886
cartoon	0.5728	riding	0.7133	character	0.0747
character	0.5298	girl	0.6779	something	0.0696
cat	0.5287	man	0.6761	makeup	0.0590
characters	0.4826	walking	0.6759	game	0.0525
car	0.4757	dancing	0.6703	player	0.0518
song	0.4522	stage	0.6346	tennis	0.0367
person	0.4274	table	0.6315	food	0.0313
something	0.4179	driving	0.6127	two	0.0141
woman	0.4070	dog	0.6114	men	0.0119
program	0.4025	woman	0.5905	people	0.0118
dog	0.3876	person	0.5702	stage	0.0110
table	0.3651	song	0.5463	cartoon	0.0091

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. In addition, Attentional Fusion enables us to analyze the attention weights for each word to examine how each modality contributes to each word. We evaluated our method on the YouTube2Text and MSR-VTT datasets, achieving results that are competitive with current state-of-the-art methods that employ temporal attention models. More importantly, we demonstrate that our model incorporating multimodal attention as well as temporal attention outperforms the state-of-the-art baseline models that use temporal attention alone. The attention mechanism also provides a means for introspection in the model, in the sense that the weights across modalities that are used in generating each word can be used to explore what features are useful in various contexts. Examination of these attention weights confirms that the focus of attention on the appropriate modality is well aligned to the semantics of the words.

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 Merriënboer, 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 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 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] G. Gravier, S. Axelrod, G. Potamianos, and C. Neti. Maximum entropy and mce based hmm stream weight estimation for audio-visual asr. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 1, pages I–853. IEEE, 2002.
- [9] S. Guadarrama, N. Krishnamoorthy, G. Malkar-nenkar, 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.
- [10] M. E. Hennecke, D. G. Stork, and K. V. Prasad. Visionary speech: Looking ahead to practical speechreading systems. In *Speechreading by Humans and Machines*, pages 331–349. Springer, 1996.
- [11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [12] C. Hori, T. Hori, T. Lee, K. Sumi, J. R. Hershey, and T. K. Marks. Attention-based multimodal fusion for video description. *CoRR*, abs/1701.03126, 2017.
- [13] C. Hori, S. Watanabe, T. Hori, B. A. Harsham, J. R. Hershey, Y. Koji, Y. Fujii, and Y. Furumoto. Driver confusion status detection using recurrent neural networks. In *IEEE International Conference on Multimedia and Expo, ICME 2016, Seattle, WA, USA, July 11-15, 2016*, pages 1–6, 2016.
- [14] 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*.
- [15] Q. Jin, J. Liang, and X. Lin. Generating Natural Video Descriptions via Multimodal Processing. In *Inter-speech*, 2016.
- [16] 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.
- [17] 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.
- [18] M. Lin, Q. Chen, and S. Yan. Network in network. *CoRR*, abs/1312.4400, 2013.
- [19] 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.
- [20] 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.
- [21] J. R. Movellan and P. Mineiro. Robust sensor fusion: Analysis and application to audio visual speech recognition. *Machine Learning*, 32(2):85–100, 1998.
- [22] 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.

- [23] 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.
- [24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [25] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.
- [26] 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 (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [33] 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.
- [34] 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.
- [35] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. *CoRR*, abs/1510.07712, 2015.
- [36] M. D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.