MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Early and Late Integration of Audio Features for Automatic Video Description

Hori, C.; Hori, T.; Marks, T.K.; Hershey, J.R.

TR2017-183 December 2017

Abstract

This paper presents our approach to improve video captioning by integrating audio and video features. Video captioning is the task of generating a textual description to describe the content of a video. State-of-the-art approaches to video captioning are based on sequence-to-sequence models, in which a single neural network accepts sequential images and audio data, and outputs a sequence of words that best describe the input data in natural language. The network thus learns to encode the video input into an intermediate semantic representation, which can be useful in applications such as multimedia indexing, automatic narration, and audio-visual question answering. In our prior work, we proposed an attention-based multi-modal fusion mechanism to integrate image, motion, and audio features, where the multiple features are integrated in the network. Here, we apply hypothesis-level integration based on minimum Bayes-risk (MBR) decoding to further improve the caption quality, focusing on well-known evaluation metrics (BLEU and METEOR scores). Experiments with the YouTube2Text and MSR-VTT datasets demonstrate that combinations of early and late integration of multimodal features significantly improve the audio-visual semantic representation, as measured by the resulting caption quality. In addition, we compared the performance of our method using two different types of audio features: MFCC features, and the audio features extracted using SoundNet, which was trained to recognize objects and scenes from videos using only the audio signals.

IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)

EARLY AND LATE INTEGRATION OF AUDIO FEATURES FOR AUTOMATIC VIDEO DESCRIPTION

Chiori Hori, Takaaki Hori, Tim K. Marks, John R. Hershey

Mitsubishi Electric Research Laboratories (MERL)
201 Broadway, Cambridge, MA 02139, USA
{chori, thori, tmarks, hershey}@merl.com

ABSTRACT

This paper presents our approach to improve video captioning by integrating audio and video features. Video captioning is the task of generating a textual description to describe the content of a video. State-of-the-art approaches to video captioning are based on sequence-to-sequence models, in which a single neural network accepts sequential images and audio data, and outputs a sequence of words that best describe the input data in natural language. The network thus learns to encode the video input into an intermediate semantic representation, which can be useful in applications such as multimedia indexing, automatic narration, and audio-visual question answering. In our prior work, we proposed an attention-based multi-modal fusion mechanism to integrate image, motion, and audio features, where the multiple features are integrated in the network. Here, we apply hypothesis-level integration based on minimum Bayes-risk (MBR) decoding to further improve the caption quality, focusing on well-known evaluation metrics (BLEU and METEOR scores). Experiments with the YouTube2Text and MSR-VTT datasets demonstrate that combinations of early and late integration of multimodal features significantly improve the audio-visual semantic representation, as measured by the resulting caption quality. In addition, we compared the performance of our method using two different types of audio features: MFCC features, and the audio features extracted using SoundNet, which was trained to recognize objects and scenes from videos using only the audio signals.

***Index Terms*—** video description, audio feature, SoundNet, MFCC, encoder-decoder, deep learning

1. INTRODUCTION

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. Recent work in video description has demonstrated the advantages 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 (extracted by a network that was trained to perform object classification), and spatiotemporal motion features (extracted by a network that was trained to perform action recognition). These two types of features are typically 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 limits 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 image and motion features.

In our prior work, we proposed a new use of attention: to fuse information across different modalities [1]. We use the term *modality* loosely: In addition to referring to features from different types of sensors, such as video and audio features, we also refer to different types of features derived from the image sequence, such as features describing image appearance , motion, or depth, as different modalities. Depending on the context, different modalities of input may be important for selecting the next word in different parts of the description. Not only do the relevant modalities change from sentence to sentence, but also from word to word, e.g., 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 word's context, may help with choosing the right words for the video description. Often features from different modalities can be complementary, in that they can provide reliable cues at different times for some aspect of a scene. Multimodal fusion is thus an important strategy for robustness. However, optimally combining information requires estimating the reliability of each modality, which remains a challenging problem.

In this work, we introduce hypothesis-level fusion across different modalities, which we call late integration of multimodal features, to the video description task. Our late integration approach is a form of system combination, where each component system generates sentence hypotheses based on a single modality, and the generated sentences are combined across systems to generate a better sentence. In this work, we apply a minimum Bayes-risk (MBR) framework to optimize the sentence combination to explicitly improve an evaluation metric such as the BLEU or METEOR scores. Moreover, we combine the early integration method proposed in our prior work with the MBR-based late integration, which increases the robustness of video description and can exploit relatively unreliable but nevertheless useful features such as audio features. In addition, we compared basic mel-frequency cepstral coefficients (MFCCs) to more advanced audio features extracted using a multi-modal network called SoundNet. SoundNet was trained from 2,000,000 unlabeled videos to recognize objects and scenes of video using only the audio signal [2]. We present results on two large datasets: YouTube2Text, and the subset of MSR-VTT that was available at the time of the experiments. We show that our combined early+late integration approach including audio features significantly improves caption quality.

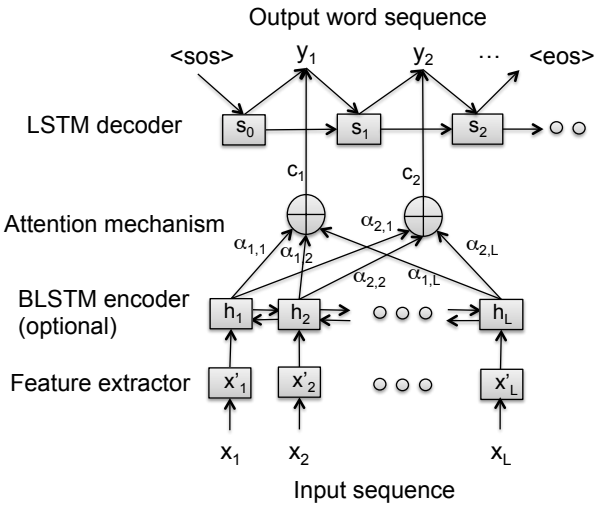

Fig. 1. An encoder-decoder based sentence generator with temporal attention mechanism.

2. VIDEO DESCRIPTION WITH TEMPORAL ATTENTION

This section describes the basic architecture of our video description system for single modality inputs. The system is based on an attention-based sequence generator [3], which enables the network to emphasize features from specific time frames depending on the current context, enabling the next word to be predicted more accurately. 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 [4].

Figure 1 shows an example of the attention-based sentence generator. The input features are extracted from the video sequence at each time step, and the temporal attention mechanism selectively weights input features from different time steps.

Given an input sequence, $X = x_1, x_2, \ldots, x_L$, each frame sample is first fed to a feature extractor. For an image sequence, the feature extractors may be pre-trained CNNs that were trained for image classification, such as GoogLeNet [5] or VGG-16 [6], or for video classification, such as C3D [7]. If audio data are available, we can extract classical audio features such as mel-frequency cepstral coefficients (MFCCs) or more advanced audio features obtained by a pre-trained CNN such as SoundNet [2]. When using a CNN for feature extraction, the sequence of feature vectors, $X' = x'_1, x'_2, \ldots, x'_L$, is obtained by extracting the activation vector of a fully-connected layer of the CNN for each input frame.[1]

The feature vectors are then optionally fed to a bidirectional LSTM (BLSTM) encoder, to obtain a sequence of hidden vectors. The LSTM decoder iteratively receives a semantic vector that contains a summary (via an attention mechanism) of the hidden state vector sequence, and predicts the next word based on that semantic vector and the current decoder state. When using CNN-based feature extractors, the feature vectors already provide an effective representation that may be fed directly to the LSTM decoder, so the BLSTM encoder is optional. When the BLSTM encoder is not used, it may be replaced by a feed-forward layer to reduce the dimensionality of the feature vectors.

If we use the CNN features directly, without a BLSTM encoder or additional feedforward layer, then we simply set $h_t = x'_t$. If the feature extractor is followed by a feed-forward layer, however, then the hidden activation vector is calculated as

$$h_t = \tanh(W_p x'_t + b_p), \tag{1}$$

where W_p is a weight matrix and b_p is a bias vector.

On the other hand, if we use a BLSTM encoder following feature extraction, then the activation vectors (i.e., encoder states) are obtained as

$$h_t = \begin{bmatrix} h_t^{(f)} \\ h_t^{(b)} \end{bmatrix}, \tag{2}$$

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

$$h_t^{(f)} = \text{LSTM}(h_{t-1}^{(f)}, x'_t; \lambda_E^{(f)}) \tag{3}$$

$$h_t^{(b)} = \text{LSTM}(h_{t+1}^{(b)}, x'_t; \lambda_E^{(b)}). \tag{4}$$

Here, LSTM() denotes a function to update the hidden vectors in the forward or backward direction using the matrix parameters $\lambda_E^{(f)}$ and bias parameters $\lambda_E^{(b)}$.

The attention mechanism uses *attention weights* to perform a weighted average of the hidden activation vectors across 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 the attention weight to the ith output word from the tth input feature vector. For the ith output word, the vector representing the relevant content of the input sequence is obtained as a weighted average of hidden unit activation vectors:

$$c_i = \sum_{t=1}^{L} \alpha_{i,t} h_t. \tag{5}$$

The decoder network is an attention-based recurrent sequence generator (ARSG) [4, 3] that uses content vectors c_i to generate an output word sequence. 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} and content vector c_i, the decoder network λ_D infers the next word probability distribution as

$$P(y|s_{i-1}, c_i) = \text{softmax}\left(W_s^{(\lambda_D)} s_{i-1} + W_c^{(\lambda_D)} c_i + b_s^{(\lambda_D)}\right), \tag{6}$$

where the decoder network λ_D is defined by weight matrices $W_s^{(\lambda_D)}$, $W_c^{(\lambda_D)}$ and bias vector $b_s^{(\lambda_D)}$. Word y_i, is generated using

$$y_i = \arg\max_{y \in V} P(y|s_{i-1}, c_i), \tag{7}$$

where V denotes the vocabulary.

The probability distribution is conditioned on the content vector c_i, which emphasizes specific features that are most relevant to predicting each subsequent word. An additional feed-forward layer may optionally be inserted before the softmax layer. In this case, the probabilities are computed as follows:

$$g_i = \tanh\left(W_s^{(\lambda_D)} s_{i-1} + W_c^{(\lambda_D)} c_i + b_s^{(\lambda_D)}\right), \tag{8}$$

and

$$P(y|s_{i-1}, c_i) = \text{softmax}(W_g^{(\lambda_D)} g_i + b_g^{(\lambda_D)}). \tag{9}$$

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

To prepare for predicting the next word, the decoder state is updated:

$$s_i = \text{LSTM}(s_{i-1}, y'_i; \lambda_D), \tag{10}$$

where $y'_i = \text{embed}(y_i) \stackrel{\text{def}}{=} W_y^{(\lambda_D)} \text{onehot}(y_i)$ is a word-embedding vector selected from the columns of a dictionary matrix $W_y^{(\lambda_D)}$, and $\text{onehot}(y_i)$ is the one-hot vector representation of y_i. The initial deocder state s_0 is obtained from the final encoder state h_L and $y'_0 = \text{embed}(\texttt{<sos>})$.

The attention weights are computed from the output context and input context as in [4]:

$$\alpha_{i,t} = \frac{\exp(e_{i,t})}{\sum_{\tau=1}^{L} \exp(e_{i,\tau})}, \tag{11}$$

where

$$e_{i,t} = w_A^{\intercal} \tanh(W_A s_{i-1} + V_A h_t + b_A). \tag{12}$$

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

In the training phase, $Y = y_1, \ldots, y_M$ is a known sentence, but in the test phase, the best word sequence needs to be found based on

$$\hat{Y} = \arg\max_{Y \in V^*} P(Y|X) \tag{13}$$

$$= \arg\max_{y_1,\ldots,y_M \in V^*} P(y_1|s_0)P(y_2|s_1)\cdots P(y_M|s_{M-1})P(\texttt{<eos>}|s_M).$$

Accordingly, we use a beam search in the test phase to keep multiple state sequence hypotheses with the highest cumulative probabilities at each step $m \leq M$, pruning out those with lower probability. The best state sequence is then selected from among those that reach the end-of-sentence token.

3. MULTIMODAL FEATURE INTEGRATION

3.1. Early Integration

We utilize an attention model to handle early integration 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. However, these different feature types are defined on asynchronous time scales, and hence it is not straightforward to fuse them directly at the feature level. In order to fuse them, some sort of cross-modal alignment is necessary, and this is provided by the attention model.

We use a method for multimodal fusion that we proposed in our prior work [1]. Using this multimodal attention mechanism, based on the current decoder state, the decoder network can selectively attend to specific modalities of input (i.e., specific feature types) to predict the next word. Let K be the number of modalities, or equivalently the number of sequences of input feature vectors. Our attention-based feature fusion is performed using

$$g_i = \tanh\left(W_s^{(\lambda_D)} s_{i-1} + \sum_{k=1}^{K} \beta_{k,i} d_{k,i} + b_s^{(\lambda_D)}\right), \tag{14}$$

where

$$d_{k,i} = W_{ck}^{(\lambda_D)} c_{k,i} + b_{ck}^{(\lambda_D)}. \tag{15}$$

The multimodal attention weights $\beta_{k,i}$ are obtained in a similar way to the temporal attention mechanism of Equations (11) and (12):

$$\beta_{k,i} = \frac{\exp(v_{k,i})}{\sum_{\kappa=1}^{K} \exp(v_{\kappa,i})}, \tag{16}$$

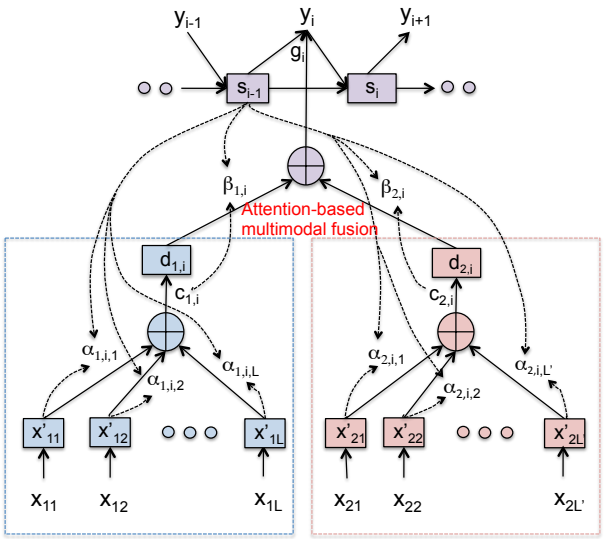

Fig. 2. Our multimodal attention mechanism.

where

$$v_{k,i} = w_B^{\intercal} \tanh(W_B s_{i-1} + V_{Bk} c_{k,i} + b_{Bk}), \tag{17}$$

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

Figure 2 shows the architecture of our sentence generator, including the multimodal attention mechanism. Like the temporal attention weights α, the feature-level attention weights β can change according to the decoder state and the content 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.

3.2. Late Integration

We now introduce a method that is novel to video description, which we call late integration of multi-modal features. This technique, which combines different features at the hypothesis level, is a kind of system combination. Each component system generates sentence hypotheses based on a single modality, and the hypotheses of multiple systems are combined to generate a better description. Although system combination has already been applied to speech recognition [8, 9] and machine translation [10], it has not yet been used for video description (to the best of our knowledge).

To perform late integration, we apply a minimum Bayes-risk (MBR) decoding [11, 12], which can improve the caption quality by focusing on a specific evaluation metric. Here we use BLEU [13] and METEOR [14] scores.

In MBR decoding, the decoding objective is defined as

$$\hat{Y} = \arg\max_{Y \in V^*} \sum_{Y' \in V^*} P(Y'|X) E(Y', Y), \tag{18}$$

where $E(Y', Y)$ denotes an evaluation metric assuming Y' is a reference (ground-truth) and Y is a hypothesis (generated description). For the BLEU [13] score, the evaluation metric can be computed as

$$E(Y', Y) = \exp\left(\sum_{n=1}^{N} \log \frac{p_n(Y', Y)}{N}\right) \times \gamma(Y', Y), \tag{19}$$

where N is the order of the BLEU score (usually $N = 4$), and $p_n(Y', Y)$ is the precision of n-grams in hypothesis Y. The penalty term, $\gamma(Y', Y) = 1$ if $\text{len}(Y') < \text{len}(Y)$ and $\exp(1 - \text{len}(Y)/\text{len}(Y))$ otherwise, penalizes hypotheses Y that are shorter than reference Y'. Since it is intractable to enumerate all possible word sequences in vocabulary V, we usually limit them to the n-best hypotheses generated by the system. Although in theory the distribution $P(Y'|Y)$ should be the true distribution, we instead estimate it using the encoder-decoder model.

In our late integration model, each modality has its own unimodal encoder-decoder based sentence generator with temporal attention (described in Figure 1 and Section 2). To combine these unimodal systems, we first merge the n-best lists generated by the multiple systems, then apply MBR decoding to choose the best description.

3.3. Early+Late Integration

In general, system combination provides a better hypothesis when using an ensemble of complementary systems in which every system has good performance to some extent. However, when one of the systems strongly under-performs the other systems, such as when one modality suffers from interference, it will not contribute to the final result at all and may even degrade the result.

To overcome this problem, we combine early and late integration approaches. That is, we apply our late integration scheme to the results of different early integration systems, for instance, early integration systems that use different subsets of features.

4. EXPERIMENTS

4.1. Datasets

We evaluated our proposed feature fusion methods using the YouTube2Text [15] and MSR-VTT [16] video datasets.

4.1.1. YouTube2Text

This dataset has 1,970 video clips, each annotated with multiple descriptions (sentences) provided by different Amazon Mechanical Turk workers. There are 80,839 sentences in total, with about 41 annotated sentences per clip. The average sentence length is 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.

4.1.2. MSR-VTT

MSR-VTT [16] consists of 10,000 web video clips, each annotated with about 20 natural language sentences by Amazon Mechanical Turk workers. The dataset contains a total of 41.2 hours of video, containing a wide variety of video content in 20 categories. The dataset is split into training (65%), validation(5%), and testing (30%) sets, respectively corresponding to 6,513, 497, and 2,990 clips. Because the video clips are hosted on YouTube, however, 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 MSR-VTT that was available, which consists of 5,763, 419, and 2,616 clips respectively for train, validation, and test. We call this the *MSR-VTT Subset*.

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 [6] 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 network [7], which was trained on the Sports-1M dataset [18]. The C3D network reads sequential frames in the video and outputs a fixed-length feature vector every 16 frames. We extracted 4096-dimensional feature vectors from fully connected layer fc6-1.

4.3. Audio Processing

Unlike previous methods that used the YouTube2Text dataset [19, 20, 21], we additionally incorporate audio features. Since the packaged YouTube2Text dataset does not include the audio track 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 covers 84% of the dataset. The 44 kHz-sampled audio data are downsampled 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, which results in a sequence of 260-dimensional vectors. The MFCC features are normalized so that the mean and variance vectors are $\mathbf{0}$ and $\mathbf{1}$ 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 to the MFCC features, which is trained jointly with the decoder network. If audio data were not available for a video clip, then we feed in a sequence of dummy MFCC features (zero vectors).

We also extracted SoundNet features using a pre-trained CNN [2]. We extracted 1024-dimensional feature vectors (using fully connected layer conv7) from each video's audio track. Unlike for MFCC features, we do not apply a BLSTM encoder for SoundNet features.

4.4. Experimental Setup

The similarity between ground truth and automatic video description results is evaluated using two metrics that were motivated by machine translation: BLEU [13] and METEOR [14]. We used the publicly available evaluation script prepared for the image captioning challenge [22].

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 feed-forward projection layer of 512 units. The MFCC audio features are fed to the BLSTM encoder followed by the decoder network. The encoder network has one projection layer of 512 units and BLSTM layers of 512 cells. The SoundNet audio features are fed to the decoder network through only one projection layer of 512 cells without BLSTM layers. 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 used the RMSprop optimizer [23] with L2 regularization. The LSTM and attention models were implemented using Chainer [24].

In our late integration approach, for each video clip we generate a 100-best sentences list using each unimodal description system, then merge the multiple 100-best lists from the target systems into one list. The best MBR result is selected from the merged list according to Eq. (18). When evaluating system performance by BLEU or METEOR score, we use the result of BLEU-based or METEOR-based MBR decoding, respectively.

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 unimodal systems to that of early and late integration multimodal systems. Early integration refers to our multimodal attention model (attentional fusion). The results for late integration were obtained by MBR decoding over the unimodal systems.

Unimodal system results show that image-only and motion-only features provide significantly better BLEU4 and METEOR scores than audio-only features. Since video description mainly relies on objects and background scene in the video, it seems to be difficult to generate appropriate descriptions only using audio features. Furthermore, some YouTube videos include unrelated sound that was not in the original scene, such as overdubbed music that was added to the video in post-production, and some video clips have no audio track. In such cases, it is almost impossible to generate related sentences.

However, by performing early integration of audio features (MFCC, SoundNet) along with the image and motion features, both BLEU4 and METEOR scores improved over unimodal systems and over multimoal systems based only on image and motion features. This result demonstrates that audio features are useful for video description when they are used as additional information. However, audio features do not contribute to the performance in late integration. This is because poor hypotheses from audio-only systems degrade the combined N-best list for MBR decoding.

Next, we evaluate several combinations of early+late integration. Tables 3 and 4 show the results on the YouTube2Text and MSR-VTT Subset datasets. In the experiments, we used late integration to combine a range of early-integration systems with different sets of features. All the systems had at least image (VGG-16) and motion (C3D) features, and optionally included audio features (MFCC or SoundNet). As shown in the tables, the BLEU4 and METEOR scores substantially improve as the number of systems increases. Thus, our early+late integration approach is effective for video description tasks, even when incorporating audio features that are not always reliable.

6. CONCLUSION

In our prior work, we proposed a new modality-dependent attention mechanism which is used as the early integration strategy in this paper. That approach provides a natural way to fuse multimodal information for video description. In this work, we also applied hypothesis-level late integration based on minimum Bayes-risk decoding to further improve description quality using BLEU and METEOR scores. Experiments on the Youtube2Text and MSR-VTT datasets demonstrate that combinations of early and late integration of multimodal features significantly improve the audio-visual semantic representation, as measured by the resulting caption quality. We also compared the performance using MFCC features to that using audio features extracted by SoundNet, which was trained to recognized objects and scenes from video using only the audio signal. Contrary to our expectations, the audio features extracted using SoundNet did not always improve the video description performance. This may be because the semantic space represented by SoundNet does not match well with the datasets used in this study, a hypothesis that could be tested by fine-tuning of SoundNet for these datasets.

Table 1. Results of feature integration on YouTube2Text dataset

	feature type				Evaluation metric	
	Image	Motion	Audio		BLEU4	METEOR
Unimodal systems	VGG-16				0.464	0.309
		C3D			0.464	0.304
			MFCC		0.267	0.228
				SoundNet	0.216	0.177
Early integration	VGG-16	C3D			0.507	0.318
	VGG-16	C3D	MFCC		0.517	**0.320**
	VGG-16	C3D		SoundNet	0.517	0.315
	VGG-16	C3D	MFCC	SoundNet	**0.519**	0.312
Late integration	VGG-16	C3D			0.499	**0.320**
	VGG-16	C3D	MFCC		0.461	0.307
	VGG-16	C3D		SoundNet	0.496	0.319
	VGG-16	C3D	MFCC	SoundNet	0.459	0.305

Table 2. Results of feature integration on MSR-VTT Subset dataset

	feature type				Evaluation metric	
	Image	Motion	Audio		BLEU4	METEOR
Unimodal systems	VGG-16				0.361	0.244
		C3D			0.362	0.246
			MFCC		0.248	0.209
				SoundNet	0.218	0.198
Early integration	VGG-16	C3D			0.394	0.257
	VGG-16	C3D	MFCC		**0.397**	**0.258**
	VGG-16	C3D		SoundNet	0.395	0.253
	VGG-16	C3D	MFCC	SoundNet	0.390	0.254
Late integration	VGG-16	C3D			0.383	0.257
	VGG-16	C3D	MFCC		0.379	0.254
	VGG-16	C3D		SoundNet	0.374	0.251
	VGG-16	C3D	MFCC	SoundNet	0.356	0.245

Table 3. Results of early+late integration on YouTube2Text dataset

Early integrated systems used for late integration			Evaluation metric	
System-1	System-2	System-3	BLEU4	METEOR
VGG-16 & C3D & MFCC			0.517	0.320
VGG-16 & C3D & MFCC	VGG-16 & C3D & SoundNet		0.525	0.322
VGG-16 & C3D & MFCC	VGG-16 & C3D & SoundNet	VGG-16 & C3D	**0.529**	**0.333**

Table 4. Results of early+late integration on MSR-VTT Subset dataset

Early integrated systems used for late integration			Evaluation metric	
System-1	System-2	System-3	BLEU4	METEOR
VGG-16 & C3D & MFCC			0.397	0.258
VGG-16 & C3D & MFCC	VGG-16 & C3D & SoundNet		0.405	0.272
VGG-16 & C3D & MFCC	VGG-16 & C3D & SoundNet	VGG-16 & C3D	**0.408**	**0.273**

7. REFERENCES

[1] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Kazuhiro Sumi, John R Hershey, and Tim K Marks, "Attention-based multimodal fusion for video description," *arXiv preprint arXiv:1701.03126*, 2017.

[2] Yusuf Aytar, Carl Vondrick, and Antonio Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in Neural Information Processing Systems*, 2016, pp. 892–900.

[3] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., pp. 577–585. Curran Associates, Inc., 2015.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.

[5] Min Lin, Qiang Chen, and Shuicheng Yan, "Network in network," *CoRR*, vol. abs/1312.4400, 2013.

[6] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[7] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 2015, pp. 4489–4497.

[8] Jonathan G Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*. IEEE, 1997, pp. 347–354.

[9] Gunnar Evermann and PC Woodland, "Posterior probability decoding, confidence estimation and system combination," in *Proc. Speech Transcription Workshop*. Baltimore, 2000, vol. 27, p. 78.

[10] Khe Chai Sim, William J Byrne, Mark JF Gales, Hichem Sahbi, and Philip C Woodland, "Consensus network decoding for statistical machine translation system combination," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. IEEE, 2007, vol. 4, pp. IV–105.

[11] Andreas Stolcke, Yochai Konig, and Mitchel Weintraub, "Explicit word error minimization in n-best list rescoring.," in *Eurospeech*, 1997, vol. 97, pp. 163–166.

[12] Shankar Kumar and William Byrne, "Minimum bayes-risk decoding for statistical machine translation," Tech. Rep., JOHNS HOPKINS UNIV BALTIMORE MD CENTER FOR LANGUAGE AND SPEECH PROCESSING (CLSP), 2004.

[13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing 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.*, 2002, pp. 311–318.

[14] Michael J. Denkowski and Alon 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*, 2014, pp. 376–380.

[15] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko, "Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2712–2719.

[16] Jun Xu, Tao Mei, Ting Yao, and Yong 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.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., pp. 1097–1105. Curran Associates, Inc., 2012.

[18] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[19] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher J. Pal, Hugo Larochelle, and Aaron C. Courville, "Describing videos by exploiting temporal structure," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 2015, pp. 4507–4515.

[20] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui, "Jointly modeling embedding and translation to bridge video and language," *CoRR*, vol. abs/1505.01861, 2015.

[21] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu, "Video paragraph captioning using hierarchical recurrent neural networks," *CoRR*, vol. abs/1510.07712, 2015.

[22] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick, "Microsoft COCO captions: Data collection and evaluation server," *CoRR*, vol. abs/1504.00325, 2015.

[23] 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.

[24] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin 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.