Audio Visual Scene-Aware Dialog Generation with Transformer-based Video Representations

Yoshihiro Yamazaki, Shota Orihashi, Ryo Masumura, Mihiro Uchida, Akihiko Takashima

NTT Computer and Data Science Laboratories, NTT Corporation, Japan
yoshihiro.yamazaki.df@hco.ntt.co.jp

Abstract

There have been many attempts to build multimodal dialog systems that can respond to a question about given audio-visual information, and the representative task for such systems is the Audio Visual Scene-Aware Dialog (AVSD). Most conventional AVSD models adopt the Convolutional Neural Network (CNN)-based video feature extractor to understand visual information. While a CNN tends to obtain both temporally and spatially local information, global information is also crucial for boosting video understanding because AVSD requires long-term temporal visual dependency and whole visual information. In this study, we apply the Transformer-based video feature that can capture both temporally and spatially global representations more efficiently than the CNN-based feature. Our AVSD model with its Transformer-based feature attains higher objective performance scores for answer generation. In addition, our model achieves a subjective score close to that of human answers in DSTC10. We observed that the Transformer-based visual feature is beneficial for the AVSD task because our model tends to correctly answer the questions that need a temporally and spatially broad range of visual information.

Introduction

While many researchers have recently tackled the text-based conversational agent (**??**), multimodal dialog systems that incorporate text, audio, and visual cues to determine the response have also been getting attention. One of the advantages of using multimodal information is that the systems are able to consider more diverse interactions (e.g., dialog systems talking with the user about the events happening around them). For example, Audio Visual Scene-Aware Dialog (AVSD) has been proposed as the task of multi-turn question-answering based on given text, audio, and video signals (**???**). Figure 1 shows the overview of the AVSD task. The contents of the audio and video are daily indoor activities, and the dialog system is expected to correctly respond to questions about them.

The AVSD task has been adopted as the competition track in the Dialog System Technology Challenge (DSTC) three times in DSTC7, DSTC8, and DSTC10. The participants of the AVSD track on the DSTC build their own conversational model by using training data distributed by the organizers, then infer the answers for the test data. The participants compete with each other based on the objective and subjective performance scores for predicted answers. In this paper, we aim to build a model for the AVSD track in DSTC10.

Many of the conventional studies employ neural-based autoregressive text generation for the AVSD model. Those models encode the text, audio, and video information into latent representations and generate response sentences. In the previous competition of DSTC8, **?** presented fine-tuning of a pre-trained Transformer-based language model; it showed remarkable performance. They indicated that pre-training of text generation is beneficial for AVSD, but the quality of visual understanding remains an issue.

Furthermore, the captions and summaries concerning the events involving audio and video are annotated to the dataset of the AVSD, and the participants of the competitions were allowed to use them as input features in DSTC7 and DSTC8. However, in actual usage of dialog systems, the captions do not exist before the conversation. In DSTC10, the participants are not allowed to use the captions for inference, only for training. The captions and summaries are powerful clues in understanding the events because they are concrete and concise representations of the video scenes. Therefore, in DSTC10, it is necessary to develop a more complete visual understanding than the conventional methods.

Most conventional methods focused on the network architecture in modality fusion or response generation module and relatively little on the video features. The typical video features for AVSD were extracted from the I3D (**?**) based on 3D-CNN. CNN-based visual feature extractors tend to output more local information than global information (**?**). The conventional AVSD model using a CNN-based video representation had difficulty in answering correctly to questions that need a temporally and spatially broad range of spatio-temporal information. Thus, promoting the ability to capture that information also seems to be necessary to develop correct answers. In recent years, TimeSformer (**?**), a Transformer-based video feature extractor, showed better performance than CNN-based models in action recognition tasks owing to its abilities such as capturing global representations. Therefore, TimeSformer appears to be effective in developing an AVSD model that can correctly understand visual information and precisely generate answers.

In this paper, we propose to apply TimeSformer-based video features to Transformer-based autoregressive response generation model to enhance visual understanding without

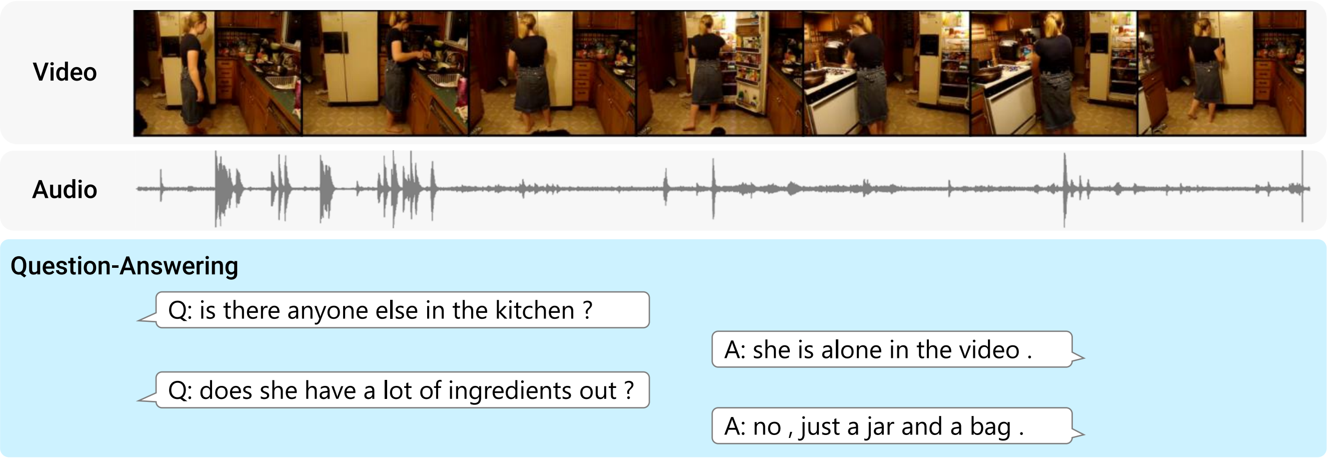

Video
Audio
Question-Answering
Q: is there anyone else in the kitchen ?
A: she is alone in the video .
Q: does she have a lot of ingredients out ?
A: no , just a jar and a bag .

Figure 1: Overview of the Audio Visual Scene-Aware Dialog

recourse to captions of the dialog. We utilize a pre-trained TimeSformer as a video feature extractor and compare its performances with that of I3D. Experiments show the improvements in the objective scores. One of the reason for the results is the ability of TimeSformer to capture a temporally and spatially broad range of visual information.

Related Studies

Network Architectures of AVSD Models

Almost all AVSD models proposed to date are based on end-to-end neural text generation. In DSTC7, the encoder-decoder model based on an RNN and an attention module was often adopted (**??**). Many of the models for DSTC7 contained the attention mechanism for multimodal feature fusion and generated natural sentences to some extent. Transformer-based generation models appeared in DSTC8. In particular, the pre-trained Transformer-based language model such as GPT-2 (**?**) or BERT (**?**) that was fine-tuned received the top two human-rated scores (**??**). Therefore, we employ pre-training and fine-tuning of the Transformer-based language model for our response generation model as it can generate fluent sentence.

Visual Features Used in AVSD

As mentioned in the previous section, many AVSD attempts tried to improve the network architectures in modality fusion or response generation module, but not video feature extraction. The video features based on I3D (**?**) were provided by the organizers of the AVSD track, and many studies have used them. I3D is a visual understanding model based on 3D-CNN; it considers a spatio-temporal relation in converting RGB images or optical flow sequences into visual feature vectors. On the other hand, there have been some methods that base their AVSD models on other visual features. **???** utilized intermediate representations of ResNet-50 (**?**), ResNeXt-101 (**?**), and 3D ResNeXt (**?**). Faster-RCNN (**?**) has been used to extract region representations for each object (**?**).

All the visual features mentioned above are based on CNN. Transformers can capture temporally and spatially global information more effectively than CNN and so offer better performance. The state-of-the-art video understanding model based on Transformer, TimeSformer (**?**), surpasses the CNN-based models in terms of action recognition. TimeSformer learns the space-time relationships in the video by using spatial and temporal self-attention. Thus, it is considered that applying TimeSformer to AVSD leverages the response performance.

AVSD Response Generation Model Using TimeSformer Features

We propose a Transformer-based autoregressive response generation model using TimeSformer video feature for AVSD. We extract the video features from a pre-trained TimeSformer model, then train the response generation model. This section overviews our response generation model, TimeSformer, and its video feature extraction process.

Proposed Response Generation Model

Figure 2 overviews the proposed Transformer-based response generation model that uses TimeSformer video feature. The network is based on GPT-2 pre-trained by only text corpora as is done in **?**. The model takes I frames of video feature $\boldsymbol{V} = \{\boldsymbol{v}_{(1)}, \cdots, \boldsymbol{v}_{(I)}\}$, where $\boldsymbol{v}_{(i)}$ is the video feature vector at the i-th frame, dialog history $\boldsymbol{H}$, and current question $\boldsymbol{Q}$ as inputs, and generates J tokens of response sentence $\boldsymbol{R} = \{r_{(1)}, \cdots, r_{(J)}\}$, where $r_{(j)}$ is the j-th token. In this paper, “dialog history” represents the multi-turn question-answer sequence up to the current question $\boldsymbol{Q}$. To decode the response sentence $\boldsymbol{R}$, the model predicts the output probability as:

$$P_{\boldsymbol{\Theta}}(\boldsymbol{R} \mid \boldsymbol{V}, \boldsymbol{H}, \boldsymbol{Q}) = \prod_{j=1}^{J} P_{\boldsymbol{\Theta}}(r_{(j)} \mid \boldsymbol{V}, \boldsymbol{H}, \boldsymbol{Q}, r_{(<j)}) \quad (1)$$

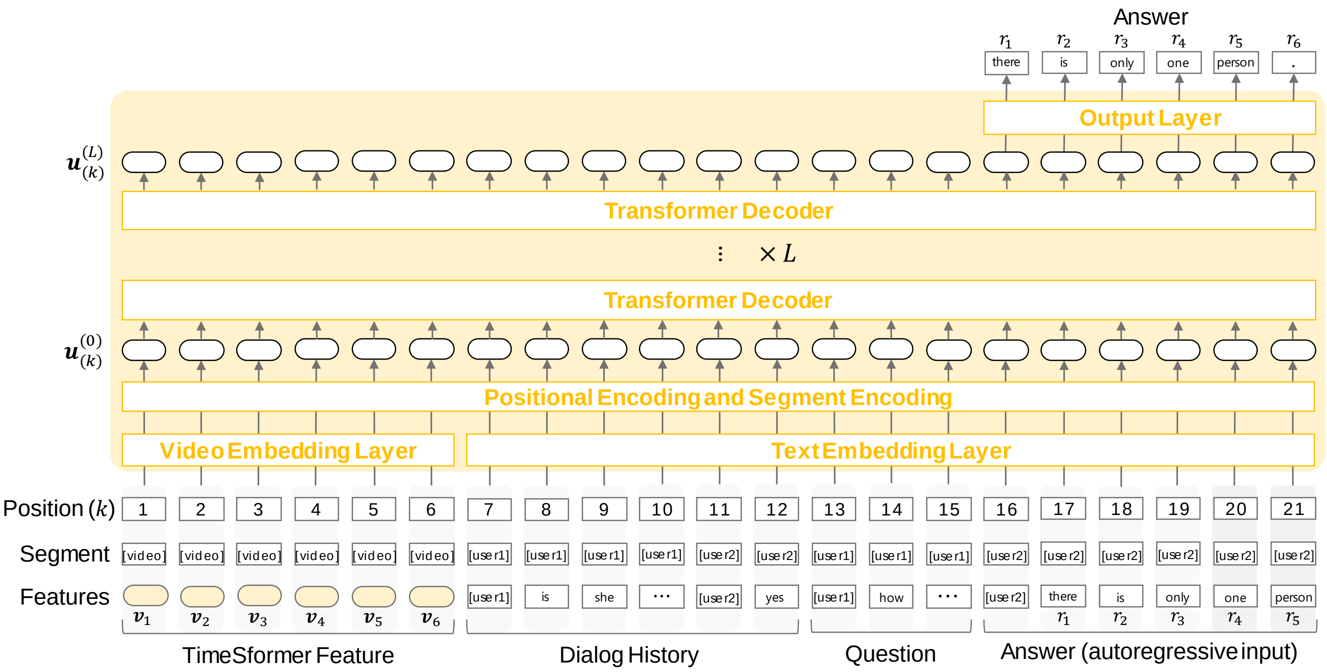

Answer
r_1 r_2 r_3 r_4 r_5 r_6
there
is
only
one
person
.
Output Layer
$u^{(L)}_{(k)}$
Transformer Decoder
⋮ × L
Transformer Decoder
$u^{(0)}_{(k)}$
Positional Encoding and Segment Encoding
Video Embedding Layer
Text Embedding Layer
Position (k)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
Segment
[video] [video] [video] [video] [video] [video] [user1] [user1] [user1] [user1] [user2] [user2] [user1] [user1] [user1] [user2] [user2] [user2] [user2] [user2] [user2]
Features
v_1 v_2 v_3 v_4 v_5 v_6
[user1] is she ⋯ [user2] yes [user1] how ⋯ [user2] there is only one person
r_1 r_2 r_3 r_4 r_5
TimeSformer Feature
Dialog History
Question
Answer (autoregressive input)

Figure 2: Our response generation model using TimeSformer video feature

where $\mathbf{\Theta}$ denotes a trainable network parameter.

While the conventional method (?) uses I3D feature as its video feature $\boldsymbol{V}$, we use the video feature extracted from the pre-trained TimeSformer model described in the following section. Input features for the response generation model are the concatenation of TimeSformer video feature, dialog history, and question.

The k-th feature embedding $\boldsymbol{e}_{\text{feat}(k)}$ from the k-th TimeSformer and text features are obtained by the video embedding layer and text embedding layer, respectively. Both of those layers are trainable linear projections. To capture positional information and explicitly distinguish feature type, the embedding vectors $\boldsymbol{e}_{\text{pos}(k)}$ and $\boldsymbol{e}_{\text{segm}(k)}$ are also obtained from the k-th positional token and segment token, respectively. The projection matrix for segment embedding is shared with that of the text embedding. The input of Transformer decoder blocks $\boldsymbol{u}^{(0)}_{(k)}$ is the sum of the feature, position, and segment embedding as follows.

$$\boldsymbol{u}^{(0)}_{(k)} = \boldsymbol{e}_{\text{feat}(k)} + \boldsymbol{e}_{\text{pos}(k)} + \boldsymbol{e}_{\text{segm}(k)} \tag{2}$$

Then, L layers of Transformer decoder blocks convert $\boldsymbol{u}^{(0)}_{(k)}$ into $\boldsymbol{u}^{(L)}_{(k)}$ as follows:

$$\boldsymbol{u}^{(l)}_{(k)} = \text{TransformerDecoderBlock}^{(l)}(\boldsymbol{u}^{(l-1)}_{(k)}) \tag{3}$$

where l denotes the layer index of the Transformer decoder blocks. The output probability for the j-th token $r_{(j)}$ is obtained by the output layer which consists of linear projection and the softmax function as follows.

$$P_{\mathbf{\Theta}}(r_{(j)} \mid \boldsymbol{V}, \boldsymbol{H}, \boldsymbol{Q}, r_{(<j)}) = \text{softmax}(\boldsymbol{W}(\boldsymbol{u}^{(L)}_{(j+\tau)})) \tag{4}$$

$$\tau = I + |\boldsymbol{H}| + |\boldsymbol{Q}| \tag{5}$$

where $\boldsymbol{W}$ is a trainable matrix.

The network parameter $\mathbf{\Theta}$ is optimized to minimize the cross-entropy loss between the output probabilities of predicted and reference tokens as:

$$\hat{\mathbf{\Theta}} = \underset{\mathbf{\Theta}}{\arg\min} - \log P_{\mathbf{\Theta}}(\boldsymbol{R} \mid \boldsymbol{V}, \boldsymbol{H}, \boldsymbol{Q}) \tag{6}$$

where $\hat{\mathbf{\Theta}}$ denotes an optimized network parameter.

Architecture of TimeSformer

? proposed TimeSformer, a video feature extractor that applies spatio-temporal self-attention to sequences of image patches. They introduced several variants of methods involving self-attention. In this study, we use “Divided Space-Time Attention”, in which temporal and spatial attention are used separately, because this approach yielded the best performance on action recognition tasks. Figure 3 shows the architecture of TimeSformer with “Divided Space-Time Attention”.

Fixed T frames of RGB image sequences are sampled from the raw video. Each image is split into N patches, each of which has size of $D \times D$ pixels following process used in ViT (**?**). Each patch is flatten into a vector $\boldsymbol{x}_{(t,n)} \in \mathbb{R}^{3D^2}$ where t and n denote frame index and patch index, respectively. The input feature of the network is the series of the vectors $\{\boldsymbol{x}_{(1,1)}, \boldsymbol{x}_{(1,2)}, \cdots, \boldsymbol{x}_{(2,1)}, \boldsymbol{x}_{(2,2)}, \cdots, \boldsymbol{x}_{(T,N)}\}$. Then, $\boldsymbol{x}_{(t,n)}$ is linearly embedded into the inner representation, and the positional embedding is added to it to capture the sequential information. Let $\boldsymbol{z}^{(0)}_{(t,n)} \in \mathbb{R}^d$ be the obtained vector, where d is the number of units of the hidden layer. Here, vector $\boldsymbol{z}^{(0)}_{(0,0)}$ is taken as the classification

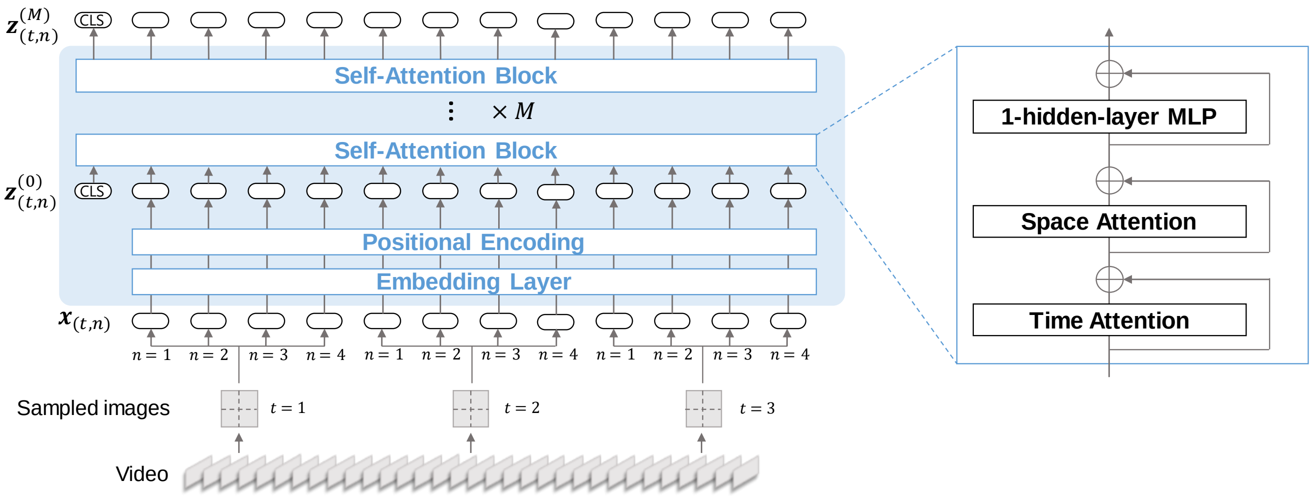


Figure 3: TimeSformer (?), a video feature extractor, with "Divided Space-Time Attention". The figure illustrates the example in sampling three image frames from a video ($T = 3$) and splitting each frame into four patches ($N = 4$).

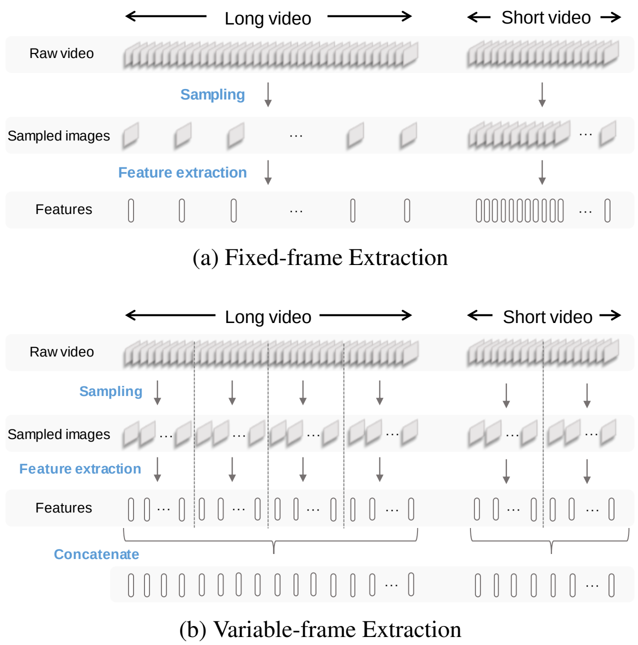
(CLS) token appended to the top of the series of $\boldsymbol{z}_{(t,n)}^{(0)}$ as in BERT (?). The sequence of $\boldsymbol{z}_{(t,n)}^{(0)}$ passes through M layers of self-attention blocks and results in $\boldsymbol{z}_{(t,n)}^{(M)}$. In a self-attention block, each patch embedding is used for time attention with the patch on the same position across different frames to extract temporal dependency. Space attention is applied to each patch embedding with all the patches in the same frame to capture spatial dependency followed by the one-hidden-layer of the feed-forward network. Layer normalization and residual connections are used for each operation.

Extraction of TimeSformer feature

The original TimeSformer model is trained for action recognition, and the CLS token on the last layer $\boldsymbol{z}_{(0,0)}^{(M)}$ is used for prediction. However, it is desired to obtain the feature vectors for each frame to more fully utilize the rich information of the video. Thus, we used the averaged vector $\hat{\boldsymbol{v}}_{(t)}$ along with all patches $\boldsymbol{z}_{(t,n)}^{(M)}$ in each frame as the visual feature for AVSD.

$$\hat{\boldsymbol{v}}_{(t)} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{z}_{(t,n)}^{(M)} \tag{7}$$

Note that T, the number of frames, is constant number throughout their pre-training phase of TimeSformer. Fixed T frames of feature vectors are obtained regardless of the length of the video, where the density of information depends on the length of the video, which is considered to have a negative impact on learning our AVSD model. Thus, we compare two methods to extract video features $\boldsymbol{V}$. One is to simply extract a fixed number of frames, and the other is to extract a variable number of frames such that all features from the sequence with different durations have equal density.



(a) Fixed-frame Extraction

(b) Variable-frame Extraction

Figure 4: Feature extraction methods using TimeSformer

Fixed-frame Extraction First, we consider a simple method to extract the vectors $\boldsymbol{V}$ for a fixed number of frames. Figure 4a shows how to extract the vectors. The sampling method is to extract fixed T frames from the entire video. In this method, $\boldsymbol{V}$ and I, the number of frames of $\boldsymbol{V}$, are indicated as follows.

$$\boldsymbol{V} = \{\hat{\boldsymbol{v}}_{(1)}, \cdots, \hat{\boldsymbol{v}}_{(T)}\} \tag{8}$$

$$I = T \tag{9}$$

Since the number of frames to be extracted remains the same regardless of the length of the video, the frame rate of feature extraction changes for each video. It is expected that this inconsistency in density degrades model performance.

Variable-frame Extraction On the other hand, we also consider a method to extract vectors for a variable number of frames at equal intervals depending on the length of the video using the pre-trained TimeSformer; the number of input and output frames is assumed to be fixed. Figure 4b shows the vector extraction method. In order to obtain vectors for a variable number of frames from the pre-trained TimeSformer, we split the video into S segments, extract fixed T frames of feature vectors from each segment, and combine them. First, the video segments that do not meet the fixed time length are complemented by copies of the last frame. Then, for each segment, fixed T frames of feature vectors are obtained using the pre-trained model of TimeSformer. Let $\hat{\boldsymbol{v}}_{(s,t)}$ be a feature vector obtained from the t-th frame in the s-th segment. Finally, by combining the vectors extracted from all segments and excluding the vectors corresponding to the complemented T_{copy} frames, video representations $\boldsymbol{V}$ can be obtained at equal intervals that depend on video length. In this method, $\boldsymbol{V}$ and I are indicated as follows.

$$\begin{aligned}\boldsymbol{V} = \{&\hat{\boldsymbol{v}}_{(1,1)}, \cdots, \hat{\boldsymbol{v}}_{(1,T)}, \hat{\boldsymbol{v}}_{(2,1)}, \cdots, \\ &\hat{\boldsymbol{v}}_{(S-1,T)}, \hat{\boldsymbol{v}}_{(S,1)}, \cdots, \hat{\boldsymbol{v}}_{(S,T-T_{\text{copy}})}\} \end{aligned} \tag{10}$$

$$I = ST - T_{\text{copy}} \tag{11}$$

Experimental Setup

We trained and evaluated the conventional response generation model that uses the I3D video feature (**?**) and our model using TimeSformer video feature by using the test sets of DSTC7 and DSTC8. We compared two conditions for TimeSformer feature extraction (fixed-frame and variable-frame).

Here, the conventional response generation model utilized not only video but also the audio features from Vggish (**?**) as non-linguistic information. Although the use of Vggish improved the objective scores to some extent, the improvement was slight. Therefore, we did not use the Vggish features with TimeSformer. To compare the performances under the conditions without the Vggish, we evaluated the model only uses I3D. Since a model ensemble is effective to improve generalization performance, we also built an ensemble of models of each condition.

We submitted the predicted answers for the test set of the DSTC10 by using our model. We reported the performances in the DSTC10.

Datasets

The AVSD track uses a multimodal dialog corpus holding questions and answers about daily life activities. The AVSD corpus was constructed by annotating the questions-answers to the videos in the Charades dataset (**?**). The annotators consist of a questioner and an answerer. The questioner asked the answerer about the contents of the audio and video material, and the answerer responded to the question posed. For each video, ten question-answering turns were conducted. Here, six ground truth answers were annotated by six different answerers in the test set. The corpus also contains the captions and summaries about the events except for the test set of DSTC10. The number of videos in the training and validation sets were 7,659 and 1,787, respectively, and that in the test sets was 1,710 in DSTC7 and DSTC8 and 1,804 in DSTC10.

Video Feature Extraction

We used the I3D-flow, I3D-rgb, and Vggish features distributed by the organizers of DSTC10 as the conventional methods. I3D-flow and I3D-rgb were trained to solve the action recognition task by using the Kinetics dataset. The number of dimensions per frame was 2,048. Vggish was trained to predict the audio class label from Youtube videos. The number of dimensions per frame was 128.

TimeSformer feature vectors were obtained from the model[1] pre-trained by using the HowTo100M action recognition dataset (**?**). The model's input consisted of thirty two 224×224 image sequences. Thus, when we extracted the visual features from the pre-trained TimeSformer, we sampled 32 frames from the original videos from the Charades dataset (**?**) and resized them to 224×224. Each patch had size of 16×16 pixels. The number of dimensions per frame was 768.

Response Generation Model

We fine-tuned the pre-trained GPT-2 (12-layer, 768-hidden, 12-heads, 117M parameters) released by the Huggingface Transformers (**?**) for the AVSD dataset. The text features were tokenized by using WordPieces (**?**) as was done in **?**.

The batchsize was four, and the optimization algorithm was AdamW (**?**) with learning rate of 6.25×10^{-5}. The models were trained using four epochs to minimize the cross-entropy loss. In decoding the answer sentence, we applied beam-search with a beam width of five, a max length of 20, and a length penalty of 0.3. We only examined beam-search because **?** reported that it is more suitable to AVSD than other decoding methods such as greedy-search and nucleus sampling. We also built an ensemble of models to improve generalization performance. We trained the four different models by varying the random seed, averaged the probability distributions of the trained models, and decoded the sentence.

Results

An objective evaluation was conducted by using the answers generated for the test sets of DSTC7 and DSTC8. We calculated the metrics based on word overlaps, such as BLEU, METEOR, ROUGE-L, and CIDEr, using the MSCOCO evaluation toolkit[2].

Tables 1 and 2 show the objective evaluation results for the test sets of DSTC7 and DSTC8, respectively. Note that "I3D and Vggish (**?**)" represents our experimental results, not the scores reported in their paper. On the whole,

[1] https://github.com/facebookresearch/TimeSformer

[2] https://github.com/tylin/coco-caption

Table 1: Objective result from the test set of DSTC7

Conditions	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
I3D and Vggish (**?**)	0.673	0.544	0.445	0.366	0.248	0.527	0.904
I3D and Vggish (ensemble)	0.694	**0.573**	**0.478**	**0.404**	0.255	0.544	1.043
I3D	0.670	0.542	0.440	0.359	0.246	0.526	0.929
I3D (ensemble)	0.675	0.547	0.447	0.368	0.247	0.529	0.942
TimeSformer fixed-frame	0.692	0.569	0.473	0.398	**0.256**	0.546	1.044
TimeSformer fixed-frame (ensemble)	0.692	0.570	0.476	0.402	**0.256**	0.546	**1.057**
TimeSformer variable-frame	0.691	0.567	0.471	0.397	0.255	0.543	1.048
TimeSformer variable-frame (ensemble)	**0.695**	0.572	0.477	0.403	0.255	**0.547**	1.049

Table 2: Objective result from the test set of DSTC8

Conditions	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
I3D and Vggish (**?**)	0.670	0.548	0.452	0.378	0.245	0.539	0.979
I3D and Vggish (ensemble)	0.675	0.557	0.465	0.392	0.248	0.542	1.018
I3D	0.669	0.547	0.454	0.379	0.244	0.534	0.970
I3D (ensemble)	0.671	0.552	0.459	0.385	0.247	0.539	0.992
TimeSformer fixed-frame	0.679	0.559	0.466	0.392	**0.252**	0.547	1.032
TimeSformer fixed-frame (ensemble)	**0.682**	**0.563**	**0.471**	**0.398**	0.251	**0.548**	**1.049**
TimeSformer variable-frame	0.681	0.561	0.467	0.393	0.250	0.546	1.037
TimeSformer variable-frame (ensemble)	0.680	0.561	0.467	0.393	0.250	0.546	1.028

Table 3: Objective and subjective result from the test set of DSTC10 evaluated by the organizers

Conditions	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr	Human
DSTC10 Baseline (**?**)	0.572	0.422	0.320	0.247	0.191	0.439	0.566	2.851
TimeSformer fixed-frame (ensemble)	0.680	0.558	0.461	0.385	0.247	0.539	0.957	3.567
TimeSformer variable-frame (ensemble)	0.679	0.554	0.456	0.379	0.246	0.536	0.945	-
Ground Truth	-	-	-	-	-	-	-	3.958

TimeSformer-based models were superior to the I3D-based models, though the ensemble of I3D and Vggish was competitive with TimeSformer in the test set of DSTC7. This result indicates that TimeSformer is a more suitable visual feature extractor than I3D for AVSD. TimeSformer fixed-frame and variable-frame performed well equally. The fixed-frame feature captures the whole information of the video, but it is sparse or dense depending on video duration. To the contrary, the variable-frame feature is uniformly extracted from the entire video, but it is unlikely to capture the temporal dependency of the whole video because feature extraction is based on segments. Since the two extraction methods have pros and cons, there was not so many differences between them. In addition, the ensemble models achieved better scores in almost all conditions.

The answers for the test set of DSTC10 were generated by using the ensemble of TimeSformer fixed-frame and variable-frame, and we submitted the prediction results to the organizers who conducted objective and subjective evaluations. The metrics of the objective evaluation are the same as used in the experiments for the test sets of DSTC7 and DSTC8. The evaluators rated the generated responses considering correctness, naturalness, and informativeness using a five-grade scale (one: very poor, five: very good).

Table 3 shows our competition results for DSTC10. The table also shows the results of the baseline system by the organizers based on a Transformer encoder-decoder using I3D and Vggish (**?**) and the subjective score for the ground truth answers. The subjective evaluation for our models examined only the fixed-frame model. Our TimeSformer-based model surpassed the baseline in both subjective and objective scores. Moreover, our model achieved a close-to-human rating against the ground truth, which indicates the suitability of the Transformer-based video feature to AVSD.

Discussion

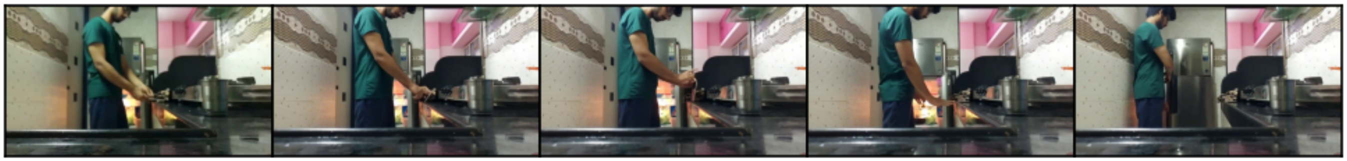
We investigated the tendency of the TimeSformer-based model in order to discern the cause of the improved performance and the remaining challenges.

The TimeSformer-based model correctly answered the question of how many people were shown in the video more often than the I3D-based model (e.g., Figure 5). To grasp the number of people in the video, the model must capture the global spatio-temporal dependency so as to detect people in each frame and track each person across the frames. The TimeSformer fixed-frame model correctly determined and answered the number of people due to its ability to catch a broad range of temporal relationships. However, the TimeSformer-based model tended to incorrectly answer the questions that needed local visual information. As shown in Figure 6, the TimeSformer-based model failed to recognize whether the man was wearing glasses or not. To answer this question correctly, the model must pay attention to the man's head, which the CNN-based I3D model is proficient at. These tendencies suggest that the model should extract local or global features of the video depending on question content for further improvement.

In addition, all models had trouble answering when the video was unclear or viewpoint movement was rapid. Therefore, stable feature extraction from low-quality or complex

Question		how many people are in the room?
Answer	I3D and Vggish (?)	there are two people in the room.
	I3D and Vggish (ensemble)	there are two people in the room.
	I3D	there are two people in the room
	I3D (ensemble)	there are two people in the room.
	TimeSformer fixed-frame	there is only one person in the room.
	TimeSformer fixed-frame (ensemble)	there is only one person in the room.
	TimeSformer variable-frame	there are two people in the room.
	TimeSformer variable-frame (ensemble)	there are two people in the room.
	Ground Truth	there is just one man

Figure 5: A question-answering sample that TimeSformer could answer correctly



Question		is this person wearing glasses?
Answer	I3D and Vggish (?)	yes, he is wearing glasses.
	I3D and Vggish (ensemble)	yes, he is wearing glasses.
	I3D	yes, he is wearing glasses.
	I3D (ensemble)	yes, he is wearing glasses.
	TimeSformer fixed-frame	no, he is not wearing glasses.
	TimeSformer fixed-frame (ensemble)	no, he is not wearing glasses.
	TimeSformer variable-frame	yes, he is wearing glasses.
	TimeSformer variable-frame (ensemble)	no, he is not wearing glasses.
	Ground Truth	yes he is wearing glasses

Figure 6: A question-answering sample that TimeSformer had difficulty in answering correctly

movement videos is required.

Conclusion

In this paper, we proposed to apply the Transformer-based video representations instead of the CNN-based representations to the autoregressive response generation model for AVSD. The results of a subjective evaluation for the test sets of DSTC7 and DSTC8 showed that the Transformer-based model outperformed the CNN-based model. Our model was competitive with the ground truth answers for DSTC10. The Transformer-based model was likely to answer properly the question about the number of people shown in the video; a task that needs the spatio-temporal global dependencies of the video.

In the future, we will construct a model that flexibly extracts local or global visual information depending on the pattern of the question. In addition, we plan to improve the visual understanding of low-quality and/or complex videos via data expansion.

Rem vel deserunt dicta architecto hic a magnam, ipsum accusantium quis ducimus commodi, nisi nam nesciunt reiciendis tempore odit suscipit voluptatum quibusdam deleniti nostrum nihil, voluptatem velit laboriosam culpa ab fugiat distinctio quibusdam corrupti, doloribus error aliquam eveniet quia obcaecati voluptates quam illo?Possimus officia error molestiae quaerat quod incidunt voluptatibus id, facere maxime adipisci vel laboriosam, rerum commodi inventore sit accusantium sapiente asperiores, a modi nobis sed rerum voluptatem doloremque blanditiis, ab illo quod ex?Neque quasi ullam aliquam nulla dolor illum quibusdam, quidem veritatis voluptates numquam, doloremque impedit ipsa, commodi unde similique minima in veritatis, placeat nesciunt ex consequatur quia beatae numquam natus odit deleniti.Dignissimos repellat culpa dolorem similique, at tenetur neque vitae?Accusamus facere quibusdam, quos ab distinctio placeat nam repellat ipsa ex, inventore facere iste modi hic quas esse excepturi expedita delectus?Ipsum voluptates facere officia autem, excepturi corrupti ducimus placeat ea magnam sit quidem fugit exercitationem nam, nulla reprehenderit voluptas molestiae delectus?Ipsam provident numquam alias delectus