The A VSD task has been adopted as the competition track for the DSTC7, DSTC8, and DSTC10. The organizers specified that the participants are not allowed to use captions for inference, only 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 (Carreira and Zisserman 2017) based on 3D-CNN. CNN-based visual feature extractors tend to output more local information than global information (Raghu et al. 2021). 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, Transformers (Bertasius, Wang, and Torresani 2021), 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 cor-
rectly 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 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 (Nguyen et al. 2019; Hori et al. 2019a). 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 (Radford et al. 2019) or BERT (Devlin et al. 2019) that was fine-tuned received the top two human-rated scores (Li et al. 2021; Chen et al. 2020). 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 (Carreira and Zisserman 2017) 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. Sanabria, Palaskar, and Metze (2019); Le et al. (2020); Geng et al. (2021) utilized intermediate representations of ResNet-50 (He et al. 2016), ResNeXt-101 (Xie et al. 2017), and 3D ResNeXt (Hara, Kataoka, and Satoh 2018). Faster-RCNN (Ren et al. 2015) has been used to extract region representations for each object (Le, Chen, and Hoi 2021).

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 (Bertasius, Wang, and Torresani 2021), 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 Li et al. (2021). The model takes $I$ frames of video feature $V = \{v_1, \cdots, v_I\}$, where $v_i$ is the video feature vector at the $i$-th frame, dialog history $H$, and current question $Q$ as inputs, and generates $J$ to-
kens of response sentence \( R = \{ r_{(1)}, \cdots, r_{(j)} \} \), 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 \( Q \). To decode the response sentence \( R \), the model predicts the output probability as:

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

where \( \Theta \) denotes a trainable network parameter.

While the conventional method (Li et al. 2021) uses I3D feature as its video feature \( V \), 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 \( e_{\text{feat}(k)} \) 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 \( e_{\text{pos}}(k) \) and \( e_{\text{segm}}(k) \) 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 \( u^{(0)}_{(k)} \) is the sum of the feature, position, and segment embedding as follows:

\[
\begin{align*}
  u^{(0)}_{(k)} &= e_{\text{feat}(k)} + e_{\text{pos}(k)} + e_{\text{segm}(k)}
\end{align*}
\]

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

\[
  u^{(l)}_{(k)} = \text{TransformerDecoderBlock}^{(l)}(u^{(l-1)}_{(k)})
\]

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:

\[
  P_{\Theta}(r_{(j)} \mid V, H, Q, r_{(<j)}) = \text{softmax}(W(u^{(L)}_{(j+r_j)}))
\]

where \( W \) is a trainable matrix.

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

\[
  \hat{\Theta} = \arg \min_{\Theta} - \log P_{\Theta}(R \mid V, H, Q)
\]
The original TimeSformer model is trained for action recognition, and the CLS token on the last layer $z^{(M)}_{(t,n)}$ 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 $\hat{v}_{(t)}$ along with all patches $z^{(M)}_{(t,n)}$ in each frame as the visual feature for AVSD.

\[
\hat{v}_{(t)} = \frac{1}{N} \sum_{n=1}^{N} z_{(t,n)}^{(M)}
\]  

(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 $V$. 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.

**Fixed-frame Extraction** First, we consider a simple method to extract the vectors $V$ 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, $V$ and $I$, the number of frames of $V$, are indicated as follows:

\[
V = \{ \hat{v}_{(1)}, \cdots, \hat{v}_{(T)} \}
\]

(8)

\[
I = T
\]

(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 \( \hat{v}_{(s,t)} \) 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 \( V \) can be obtained at equal intervals that depend on video length. In this method, \( V \) and \( I \) are indicated as follows.

\[
V = \{ \hat{v}_{(1,1)}, \cdots, \hat{v}_{(1,T)}, \cdots, \hat{v}_{(S-1,T)}, \hat{v}_{(S,1)}, \cdots, \hat{v}_{(S,T-T_{copy})} \} \tag{10}
\]

\[
I = ST - T_{copy} \tag{11}
\]

**Experimental Setup**

We trained and evaluated the conventional response generation model that uses the I3D video feature (Li et al. 2021) 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 (Hershey et al. 2017) 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 (Sigurdsson et al. 2016). 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 pre-trained by using the HowTo100M action recognition dataset (Miech et al. 2019). The model’s input consisted of thirty two \( 224 \times 224 \) 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 (Sigurdsson et al. 2016) and resized them to \( 224 \times 224 \). Each patch had size of \( 16 \times 16 \) 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 (Wolf et al. 2020) for the AVSD dataset. The text features were tokenized by using WordPieces (Wu et al. 2016) as was done in Li et al. (2021).

The batchsize was four, and the optimization algorithm was AdamW (Loshchilov and Hutter 2019) with learning rate of \( 6.25 \times 10^{-5} \). 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 Li et al. (2021) reported that it is more suitable to AVSD than other decoding methods.

---

https://github.com/facebookresearch/TimeSFormer
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.1

Tables 1 and 2 show the objective evaluation results for the test sets of DSTC7 and DSTC8, respectively. Note that “I3D and Vggish (Li et al. 2021)” represents our experimental results, not the scores reported in their paper. On the whole, 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 such as greedy-search and nucleus sampling, 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
| Question               | Answer                                      |
|-----------------------|---------------------------------------------|
| how many people are in the room? | I3D and Vggish (Li et al. 2021) 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               | Answer                                      |
|-----------------------|---------------------------------------------|
| is this person wearing glasses? | I3D and Vggish (Li et al. 2021) 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.

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 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.
