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

Low-Latency Streaming Scene-aware Interaction Using Audio-Visual Transformers

Hori, Chiori; Hori, Takaaki; Le Roux, Jonathan

TR2022-116 October 04, 2022

Abstract

To apply scene-aware interaction technology to real-time dia- log systems, we propose an online low-latency response gen- eration framework for scene-aware interaction using a video question answering setup. This paper extends our prior work on low-latency video captioning to build a novel approach that can optimize the timing to generate each answer under a trade- off between latency of generation and quality of answer. For video QA, the timing detector is now in charge of finding a tim- ing for the question-relevant event, instead of determining when the system has seen enough to generate a general caption as in the video captioning case. Our audio visual scene-aware dialog system built for the 10th Dialog System Technology Challenge was extended to exploit a low-latency function. Experiments with the MSRVTT-QA and AVSD datasets show that our ap- proach achieves between 97% and 99% of the answer quality of the upper bound given by a pre-trained Transformer using the entire video clips, using less than 40% of frames from the beginning.

Interspeech 2022

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Low-Latency Streaming Scene-aware Interaction Using Audio-Visual Transformers

Chiori Hori, Takaaki Hori, Jonathan Le Roux

Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA

{chori, leroux}@merl.com

Abstract

To apply scene-aware interaction technology to real-time dialog systems, we propose an online low-latency response generation framework for scene-aware interaction using a video question answering setup. This paper extends our prior work on low-latency video captioning to build a novel approach that can optimize the timing to generate each answer under a trade-off between latency of generation and quality of answer. For video QA, the timing detector is now in charge of finding a timing for the question-relevant event, instead of determining when the system has seen enough to generate a general caption as in the video captioning case. Our audio visual scene-aware dialog system built for the 10th Dialog System Technology Challenge was extended to exploit a low-latency function. Experiments with the MSRVTT-QA and AVSD datasets show that our approach achieves between 97% and 99% of the answer quality of the upper bound given by a pre-trained Transformer using the entire video clips, using less than 40% of frames from the beginning.

Index Terms: online streaming video QA, AVSD, low-latency, audio-visual, transformer

1. Introduction

We are pursuing scene-aware interaction technologies which allow machines to interact with humans based on shared knowledge obtained through recognizing and understanding their surroundings using various kinds of sensors as introduced in [1]. Towards this goal, we have investigated audio-visual scene-aware dialog (AVSD) since 2018.

In the early stages of the research, end-to-end approaches were shown to better handle flexible conversations between a user and a system by training models on large conversational data sets [2, 3]. Such approaches have been extended to carry out conversations about objects and events taking place around the machines or the users, based on understanding of the dynamic scenes captured by multimodal sensors such as a video camera and a microphone. This extension allows users to ask questions about what is happening in their surroundings. Using this end-to-end framework, visual question answering (VQA) for one-shot QA about a static image [4–7], and Visual dialog in which an AI agent holds a meaningful dialog with humans about a static image using natural, conversational language [8] have been intensively researched. More recently, the framework has been extended to address video clips, where systems need to understand what, when, how, and by whom events took place based on a time series of audio-visual features. New technologies have been actively investigated to tackle these scene-aware interaction problems within the context of the Video QA and AVSD tasks [9–11]. Video QA [12] is a QA task which consists in answering a single question about video clips such as those found on YouTube, and was formalized within the MSVD [13] and MSRVTT [14] datasets. AVSD is a multi-turn dialog task which consists in generating responses to a user's questions about daily-life video clips, and has been addressed in the Dialog Systems Technology Challenges (DSTC) [15, 16].

However, such scene-aware interaction tasks assume that the system can access all frames of the video clip when predicting an answer to a question. This assumption is not practical for real-time dialog systems that are monitoring an ongoing audio-visual scene, where it is essential not only to predict an answer accurately but also to respond to the user as soon as possible by finding the question-related events quickly in the online video stream and generating an appropriate answer. Such functionality requires the development of new low-latency QA techniques.

In previous work, we proposed a low-latency audio-visual captioning method, which can describe events accurately and quickly without waiting for the end of video clips, optimizing the timing for captioning [17]. In parallel, we recently introduced a new AVSD task for the third AVSD challenge at DSTC10[1], which notably asks systems to demonstrate temporal reasoning by finding evidence from the video to support their answers. This was based on a new extension of the AVSD dataset for DSTC10, for which we collected human-generated temporal reasoning data [18]. This work proposes to extend our low-latency captioning approach to the scene-aware interaction task, combining it with our reasoning AVSD system to develop a low-latency online scene-aware interaction system.

In the same spirit as our low-latency captioning approach, the proposed system optimizes the output timing for each answer based on a trade-off between latency and answer quality. We train a low-latency audio-visual Transformer composed of (1) a Transformer-based response generator which tries to generate the ground-truth answer after only seeing a small portion of all video frames, and also to mimic the outputs of a similar pre-trained response generator that is allowed to see the entire video, and (2) a CNN-based timing detector that can find the best timing to output an answer for the input question, such that the responses ultimately generated by the above two Transformers become sufficiently close to each other. The proposed jointly-trained response generator and timing detector can generate answer responses in an early stage of a video clip, as soon as a relevant event happens, and may even forecast future frames. Thanks to the combination of information from multiple modalities, the system has more opportunities to recognize an event at an earlier timing by relying on the earliest cue in one of the modalities.

Experiments with the MSR-VTT QA and AVSD datasets show that our approach achieves low-latency video QA with competitive answer quality to an offline video QA baseline that utilizes the entire video frames.

[1] https://dstc10.dstc.community/aaai-22-workshop

2. Related work

To realize real-time systems generating sentences such as closed captioning and interpretation, some works have explored low-latency end-to-end sentence generation for machine translation (MT) and automatic speech recognition (ASR). In the field of MT, simultaneous translation using greedy decoding was investigated, as well as issues on streaming for Neural MT (NMT) [19–23]; an output timing when a phrase is fully translated into a target language was incrementally determined. The approaches proposed in [24, 25] iteratively retranslated by concatenating subsequent words and updating the output. This framework allows to generate a partial translation in the meantime before a full source sentence is translated. In the field of ASR, a real-time technology is also essential for applications such as closed captions. In [26–29], end-to-end systems that regularize or penalize the emission delay using endpoint detection, and penalty terms that constrain alignments were proposed. The target of these approaches is to output a complete transcription as soon as or even slightly earlier than the end of an utterance.

Online video captioning is also essential for scene-understanding deployed in surveillance systems. For example, [30] has attempted to anticipate caption generation for future frames, exploiting the current event features as a contextual feature and using them as input into a captioning module to generate future captions. This technology relies on the temporal dependency between events in a sequence. Our prior work [17] addressed low-latency video captioning, where we designed a model architecture with a caption generator and a timing detector to generate video captions as early as possible.

However, what we address in this paper is video QA, which is different from video captioning. Our aim is here to generate appropriate answers for a user's questions as soon as possible. To this end, we introduce a question encoder to provide question embeddings to the timing detector, and extend the sentence generator to accept a question as contextual information. Thus, while the proposed method utilizes the same underlying mechanism for low-latency processing, the model is extended for video QA tasks.

3. Low-latency video QA model

3.1. Architecture

We build our proposed model for low-latency video QA upon the DSTC10-AVSD system, which employs an AV-transformer architecture [31, 32]. For the DSTC10-AVSD challenge, we extended the AV-transformer with joint student-teacher learning and attentional multimodal fusion to achieve state-of-the-art performance [33]. As in our low-latency video captioning system [17], the low-latency QA model receives video and audio features in a streaming manner, and a timing detector decides when to generate an answer response for the feature sequence the model has received until that moment. Figure 1 illustrates the model architecture, which consists of a question encoder, an AV encoder, a timing detector, and a response decoder, where the AV encoder is shared by the timing detector and the response decoder.

Given a video stream and a question text as inputs, the AV encoder encodes VGGish and I3D features extracted from the audio and video tracks, while a Transformer-based text encoder encodes the question. The sequences of audio and visual features from a starting point to the current time are fed to the encoder, and converted to hidden vector sequences through self-attention, bi-modal attention, and feed-forward layers. This encoder block is repeated N times, and the final encoded representation is obtained via the N-th encoder block. The question word sequence is also encoded via a word embedding layer followed by a Transformer with N' blocks.

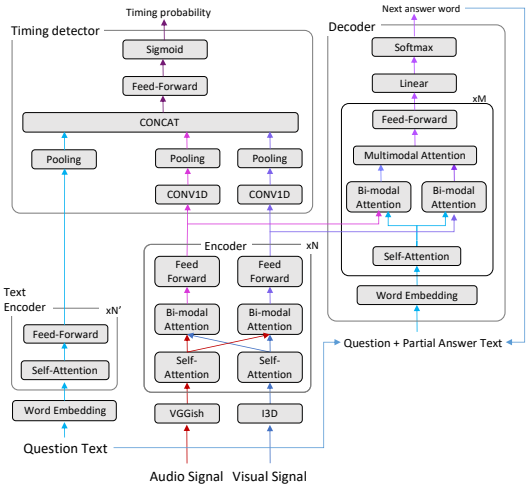

Figure 1: *Online Audio-Visual Transformer for Online Streaming Video QA.*

With the AV encoder and the question encoder, we obtain encoded representations A and V of the audio-visual features, and Q of the question word sequence $\bar{Q}$. The timing detector receives the encoded representations A and V available up to the current time and the encoding Q of the input question. The role of the timing detector is to decide whether the system should generate an answer or not for the given encoded features and the question. The detector first processes the encoded vector sequence from each modality with stacked 1D-convolution layers (Conv1d) as

$$A' = \mathrm{Conv1d}(A), \quad V' = \mathrm{Conv1d}(V). \tag{1}$$

These time-convoluted sequences and the question encoding are then summarized into a single vector through mean pooling and concatenation operations:

$$\mathcal{H} = \mathrm{Concat}(\mathrm{Mean}(A'), \mathrm{Mean}(V'), \mathrm{Mean}(Q)) \tag{2}$$

A feed-forward layer FFN and sigmoid function σ convert the summary vector to the probability of d, where d indicates whether a relevant answer can be generated or not:

$$P(d = 1|A, V, Q) = \sigma(\mathrm{FFN}(\mathcal{H})). \tag{3}$$

Once the timing detector outputs a probability higher than a threshold, e.g., $P(d = 1|A, V, Q) > 0.5$, the decoder generates an answer based on the encoded representations.

The decoder iteratively predicts the next word, starting from question word sequence $\bar{Q}$ plus a starting token <sos>. At each iteration step, it receives the partial answer sentence that has already been generated, and predicts the next word by applying M decoder blocks and a prediction network. Let Y_i be a partial answer sentence $<\mathrm{sos}>, y_1, \ldots, y_i$ after i iterations and $\mathcal{Y}_i^0$ be its question-conditioned sequence $\bar{Q}^0 Y_i^0$, where each word in the question and the partial answer sentence is converted to a word embedding vector. Each decoder block has

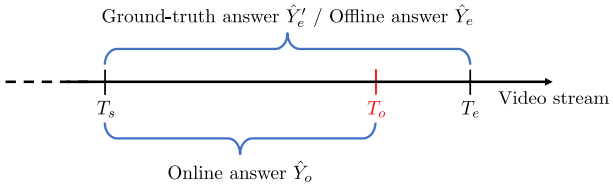

Figure 2: *Online and offline answer generation.*

self-attention, bi-modal source attention, and feed-forward layers:

$$\bar{\mathcal{Y}}_i^m = \mathcal{Y}_i^{m-1} + \mathrm{MHA}(\mathcal{Y}_i^{m-1}, \mathcal{Y}_i^{m-1}, \mathcal{Y}_i^{m-1}), \quad (4)$$

$$\bar{\mathcal{Y}}_i^{Am} = \bar{\mathcal{Y}}_i^m + \mathrm{MHA}(\bar{\mathcal{Y}}_i^m, A, A), \quad (5)$$

$$\bar{\mathcal{Y}}_i^{Vm} = \bar{\mathcal{Y}}_i^m + \mathrm{MHA}(\bar{\mathcal{Y}}_i^m, V, V), \quad (6)$$

$$\tilde{\mathcal{Y}}_i^m = \mathrm{Concat}(\bar{\mathcal{Y}}_i^{Am}, \bar{\mathcal{Y}}_i^{Vm}), \quad (7)$$

$$\mathcal{Y}_i^m = \tilde{\mathcal{Y}}_i^m + \mathrm{FFN}(\tilde{\mathcal{Y}}_i^m), \quad (8)$$

where MHA denotes a multi-head attention network. The self-attention layer maps the word embedding vectors to high-level representations that contain their temporal dependency in (4). The bi-modal attention layers update the word representations based on the relevance to the encoded bi-modal representations in (5) and (6). The feed-forward layer converts the concatenated outputs of the bi-modal attention layers in (7) and (8). These operations are repeated to the M-th block. The linear transform and the softmax operation are applied to the M-th output to obtain the probability distribution of the next word as

$$P(\mathbf{y}_{i+1}|\bar{Q}Y_i, A, V) = \mathrm{Softmax}(\mathrm{Linear}(y_i^M)), \quad (9)$$

$$\hat{y}_{i+1} = \underset{y \in \mathcal{V}}{\mathrm{argmax}}\, P(\mathbf{y}_{i+1} = y|\bar{Q}Y_i, A, V), \quad (10)$$

where $\mathcal{V}$ denotes the vocabulary. The partial answer is extended by appending the best word $\hat{y}_{i+1}$ to the previous partial sentence as $Y_{i+1} = Y_i, \hat{y}_{i+1}$. This is a greedy search process repeated until it receives an end-of-sentence token <eos> as $\hat{y}_{i+1}$. We can also use a beam search technique, which selects top-K words with highest probabilities and keeps multiple word sequences. Finally, the best word sequence is selected as the final answer for the input question.

3.2. Training and inference

We jointly train the AV encoder, the question encoder, the response decoder, and the timing detector, so that the system achieves an answer quality comparable to that for a complete video, even if the given video is shorter than the original one by truncating the later part.

As we proposed in [17], we utilize two loss functions, a response loss to improve the answer quality and a timing detection loss to detect a proper timing to emit an answer sentence. Figure 2 illustrates a video stream, where the video has started at time T_s and ends at T_e, which are associated with ground-truth answer Y'_e. If time T_o is picked as the emission timing, the response decoder generates an answer based on the audio-visual signals $X_{T_s:T_o} = (A_{T_s:T_o}, V_{T_s:T_o})$ from time T_s to time T_o, and the question $\bar{Q}$.

The response loss is based on a standard cross-entropy loss for the ground-truth answer Y'_e,

$$\mathcal{L}_{CE} = -\log P(Y'_e|X_{T_s:T_o}, \bar{Q}; \theta_R), \quad (11)$$

and a Kullback–Leibler (KL) divergence loss between predictions from a pre-trained model allowed to process the complete video and the target model that can only process incomplete videos, i.e.,

$$\mathcal{L}_{KL} = -\sum_{i=1}^{|Y'_e|} \sum_{y \in \mathcal{V}} P(y|Y'_{e,i}, X_{T_s:T_e}, \bar{Q}; \bar{\theta}_R) \log P(y|Y'_{e,i}, X_{T_s:T_o}, \bar{Q}; \theta_R). \quad (12)$$

This student-teacher learning approach [34] can exploit the superior description power of the teacher model $\bar{\theta}_R$, which predicts an answer using entire video clip $X_{T_s:T_e}$, by pushing the student model θ_R to mimic the teacher's predictions using only the truncated video clip $X_{T_s:T_o}$. This improves the training stability and leads to better performance.

The timing detection loss is based on a binary cross-entropy for appropriate timings. In general, however, such timing information does not exist in the training data set. As in [17], we decide the right timing based on whether or not the response decoder can generate an appropriate answer, that is, an answer sufficiently close to the ground-truth Y'_e or the answer $\hat{Y}_e$ generated for the entire video clip $X_{T_s:T_e}$ using the pre-trained model $\bar{\theta}_R$. The detection loss is computed as

$$\mathcal{L}_D = -\log P(d|X_{T_s:T_o}, \bar{Q}; \theta_D), \quad (13)$$

where d is determined based on

$$d = \begin{cases} 1 & \text{if } \max(\mathrm{Sim}(Y'_e, \hat{Y}_o), \mathrm{Sim}(\hat{Y}_e, \hat{Y}_o)) \geq S, \\ 0 & \text{otherwise}, \end{cases} \quad (14)$$

where $\mathrm{Sim}(\cdot,\cdot)$ denotes a similarity measure between two word sequences. In this work, we use the matched word percentage computed in a teacher-forcing manner, where we obtain $\hat{Y}_e$ and $\hat{Y}_o$ as sequences of highest-probability words given the ground-truth word sequence as the left context, and count the matched words between them. $S \in (0, 1]$ is a pre-determined threshold which judges whether or not the online answer $\hat{Y}_o$ is sufficiently close to the references Y'_e and $\hat{Y}_e$.

The model $\theta = (\theta_R, \theta_D)$ is trained by repeating the steps of sampling the emission timing $T_o \sim \mathrm{Uniform}(T_s, T_e)$, computing the loss $\mathcal{L} = \alpha\mathcal{L}_{CE} + \beta\mathcal{L}_{KL} + \gamma\mathcal{L}_D$, and updating the parameters θ using $\nabla_\theta \mathcal{L}$.

At inference time, the emission time $\hat{T}_o$ is determined as the first time that meets $P(d = 1|X_{T_s:\hat{T}_o}, \bar{Q}; \theta_D) > F$, where F is a pre-determined threshold controlling the sensitivity of timing detection. An answer is then generated based on

$$\hat{Y}_o = \underset{Y \in \mathcal{V}^*}{\mathrm{argmax}}\, P(Y|X_{T_s:\hat{T}_o}, \bar{Q}; \theta_R). \quad (15)$$

Note that we assume that T_s is already determined.

4. Experiments

We evaluate our low-latency video QA method using the MSRVTT-QA [12] and AVSD [9–11] datasets.

MSRVTT-QA is based on the MSR-VTT dataset [14], which contains 10k video clips and 243k question-answer (QA) pairs. The QA pairs are generated automatically from the manually-annotated captions for each video clip, where the question is a sentence and the answer is a single word. We follow the data split in the MSR-VTT dataset which is 65% for training, 5% for validation, and 30% for test.

AVSD is a set of text-based dialogs on short videos from the Charades dataset [35], which consists of untrimmed multi-action videos, which each include an audio track. In AVSD, two

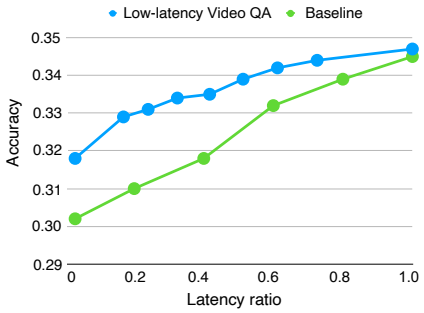

Figure 3: *Latency ratio vs. Accuracy [%] on MSRVTT-QA.*

parties, dubbed *questioner* and *answerer*, have a dialog about events in the provided video. The job of the answerer, who has already watched the video, is to answer questions asked by the questioner [36]. We follow the AVSD challenge setting, where the train, validation, and test sets consist of 7.7k, 1.8k, and 1.8k dialogs, respectively, and each dialog includes 10-turn QA pairs, where both questions and answers are sentences. The duration of video clips ranges from 10 to 40 seconds.

The VGGish features were configured to form a 128-dimensional vector sequence for the audio track of each video, where each audio frame corresponds to a 0.96 s segment without overlap. The I3D features were configured to form a 2048-dimensional vector sequence for the video track, where each visual frame corresponds to a 2.56 s segment without overlap.

We first trained a multi-modal Transformer with entire video clips and QA pairs. This model was used as a baseline and teacher model. We used $N = 2$ audio-visual encoder blocks, $N' = 4$ question encoder blocks, $M = 4$ decoder blocks, and set the number of attention heads to 4. The vocabulary size was 7,599 for MSRVTT-QA and 3,669 for AVSD. The dimension of the word embedding vectors was 300.

The proposed model for low-latency video QA was trained with incomplete video clips according to the steps in Section 3.2. The architecture was the same as the baseline/teacher model except for the addition of the timing detector. In the training process, we consistently used $\alpha = \beta = \gamma = 1/3$ for the loss function and threshold $S = 0.9$ in Eq. (14). We set the dimensions of the hidden activations in the audio and visual attention layers to 256 and 1024, respectively, the dropout rate to 0.1, and we applied a label smoothing technique. The timing detector consisted of 2 stacked 1D-convolution layers with a ReLU non-linearity in between. The performance was measured by answer accuracy for MSRVTT-QA, and BLEU4 and METEOR scores for AVSD.

Figure 3 shows the relationship between latency ratio and answer accuracy on MSRVTT-QA. The latency ratio denotes the ratio of actually used frames (from the beginning) to the entire video frames. The baseline results were obtained with the baseline (teacher) model by simply omitting the future frames at various ratios. Results for the proposed models were obtained by changing the detection threshold F. The accuracy for MSRVTT-QA shows a percentage of one-word answers matching with the ground-truths. The result demonstrates that our proposed method achieves low-latency video QA with much smaller accuracy degradation compared to the baseline. Our approach achieves 97% of the answer quality of the upper bound given by the pre-trained Transformer using the entire video clips, using only 40% of frames from the beginning.

Table 1 compares the quality of answer sentences on the

Table 1: *Answer quality of baseline and proposed systems on AVSD task.*

		AVSD-DSTC7		AVSD-DSTC8	
Latency	Method	BLEU4	METEOR	BLEU4	METEOR
1.0	Baseline	0.394	0.252	0.372	0.237
	Proposed	**0.396**	**0.253**	**0.381**	**0.243**
0.5	Baseline	0.387	0.238	0.362	0.226
	Proposed	**0.396**	**0.254**	**0.379**	**0.243**
0.2	Baseline	0.362	0.221	0.345	0.212
	Proposed	**0.390**	**0.250**	**0.374**	**0.239**

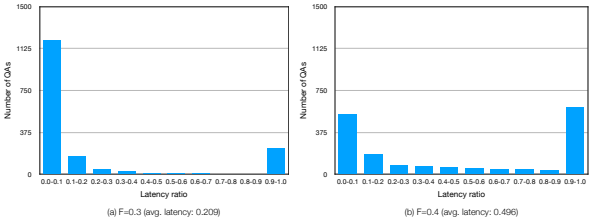

Figure 4: *Histogram of detected timings for detection thresholds $F = 0.3$ (left) and $F = 0.4$ (right) in DSTC7.*

AVSD task. We controlled the latency ratio by setting the detection threshold F to obtain average ratios of 1.0, 0.5, and 0.2 for the proposed method. We omitted the future frames with the above fixed ratios for the baseline system. As shown in the table, our proposed method slightly outperforms the baseline even with 1.0 latency. This could be due to the increased robustness of the model by training with randomly truncated video. Moreover, the proposed method keeps the same level of BLEU4 and METEOR scores at the 0.5 latency and achieves competitive scores even at the 0.2 latency with little degradation, reaching 98% to 99% of the scores at the 1.0 latency condition.

Figure 4 shows the distribution of the QAs over the latency with detection thresholds $F = 0.3$ and $F = 0.4$ on the AVSD-DSTC7 task, which correspond to the results for latency 0.2 (left) and 0.5 (right) in Table 1. These results illustrate that most of the QAs of AVSD require either only the early frames or all frames to generate accurate answers. We investigated the reason for the polarized distribution. The most frequent pattern for questions leading to an early decision is "How does the video starts?". Furthermore, there are some consistent answers such as "one person" in response to "How many people are in the videos?" in the training data. Such frequent linguistic patterns could also be a cause for the early decision. The late decision case contains such patterns as "How does the video ends?". Such a question is natural for questioners who need to generate video captions through 10 QAs without watching the full videos.

5. Conclusion

We proposed a low-latency video QA method, which can answer a user's questions accurately and quickly without waiting for the end of video clips. The proposed method optimizes each answer's output timing based on a trade-off between latency and answer quality. We have demonstrated that the proposed system can generate answers in early stages of video clips using the MSRVTT-QA and AVSD datasets, achieving between 97% and 99% of the answer quality of the upper bound given by a pre-trained Transformer using the entire video clips, using less than 40% of frames from the beginning.

6. References

[1] C. Hori and A. Vetro, "At last, a self-driving car that can explain itself," in *IEEE Spectrum*, Feb. 2022. [Online]. Available: https://spectrum.ieee.org/at-last-a-self-driving-car-that-can-explain-itself

[2] O. Vinyals and Q. Le, "A neural conversational model," *arXiv preprint arXiv:1506.05869*, 2015.

[3] C. Hori, J. Perez, R. Higasinaka, T. Hori, Y.-L. Boureau, M. Inaba, Y. Tsunomori, T. Takahashi, K. Yoshino, and S. Kim, "Overview of the sixth dialog system technology challenge: DSTC6," *Comput. Speech Lang.*, 2018.

[4] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual Question Answering," in *Proc. ICCV*, 2015.

[5] P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh, "Yin and Yang: Balancing and answering binary visual questions," in *Proc. CVPR*, 2016.

[6] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering," in *Proc. CVPR*, 2017.

[7] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler, "MovieQA: Understanding stories in movies through question-answering," in *Proc. CVPR*, 2016, pp. 4631–4640.

[8] A. Das, S. Kottur, J. M. Moura, S. Lee, and D. Batra, "Learning cooperative visual dialog agents with deep reinforcement learning," in *Proc. ICCV*, 2017.

[9] H. Alamri, V. Cartillier, R. G. Lopes, A. Das, J. Wang, I. Essa, D. Batra, D. Parikh, A. Cherian, T. K. Marks *et al.*, "Audio visual scene-aware dialog (AVSD) challenge at DSTC7," *arXiv preprint arXiv:1806.00525*, 2018.

[10] C. Hori, A. Cherian, T. Hori, and T. K. Marks, "Audio visual scene-aware dialog (AVSD) track for natural language generation in DSTC8," in *Proc. DSTC8 Workshop at AAAI*, 2020.

[11] C. Hori, A. P. Shah, S. Geng, P. Gao, A. Cherian, T. Hori, J. Le Roux, and T. K. Marks, "Overview of audio visual scene-aware dialog with reasoning track for natural language generation in DSTC10," in *Proc. DSTC10 Workshop at AAAI*, 2022.

[12] D. Xu, Z. Zhao, J. Xiao, F. Wu, H. Zhang, X. He, and Y. Zhuang, "Video question answering via gradually refined attention over appearance and motion," in *Proc. ACM Multimedia*, 2017, pp. 1645–1653.

[13] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko, "Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition," in *Proc. ICCV*, Dec. 2013.

[14] J. Xu, T. Mei, T. Yao, and Y. Rui, "MSR-VTT: A large video description dataset for bridging video and language," in *Proc. CVPR*, 2016, pp. 5288–5296.

[15] L. F. D'Haro, K. Yoshino, C. Hori, T. K. Marks, L. Polymenakos, J. K. Kummerfeld, M. Galley, and X. Gao, "Overview of the seventh dialog system technology challenge: DSTC7," *Comput. Speech Lang.*, vol. 62, p. 101068, 2020.

[16] S. Kim, M. Galley, C. Gunasekara, S. Lee, A. Atkinson, B. Peng, H. Schulz, J. Gao, J. Li, M. Adada, M. Huang, L. Lastras, J. K. Kummerfeld, W. S. Lasecki, C. Hori, A. Cherian, T. K. Marks, A. Rastogi, X. Zang, S. Sunkara, and R. Gupta, "Overview of the eighth dialog system technology challenge: DSTC8," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 2529–2540, 2021.

[17] C. Hori, T. Hori, and J. Le Roux, "Optimizing latency for online video captioningusing audio-visual transformers," in *Proc. Interspeech*, 2021.

[18] C. Hori, H. Alamri, J. Wang, G. Wichern, T. Hori, A. Cherian, T. K. Marks, V. Cartillier, R. G. Lopes, A. Das *et al.*, "End-to-end audio visual scene-aware dialog using multimodal attention-based video features," in *Proc. ICASSP*, May 2019, pp. 2352–2356.

[19] K. Cho and M. Esipova, "Can neural machine translation do simultaneous translation?" *arXiv preprint arXiv:1606.02012*, 2016.

[20] J. Gu, G. Neubig, K. Cho, and V. O. Li, "Learning to translate in real-time with neural machine translation," in *Proc. EACL*, Apr. 2017.

[21] M. Ma, L. Huang, H. Xiong, R. Zheng, K. Liu, B. Zheng, C. Zhang, Z. He, H. Liu, X. Li, H. Wu, and H. Wang, "STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework," in *Proc. ACL*, Jul. 2019.

[22] F. Dalvi, N. Durrani, H. Sajjad, and S. Vogel, "Incremental decoding and training methods for simultaneous translation in neural machine translation," in *Proc. NAACL HLT*, Jun. 2018.

[23] N. Arivazhagan, C. Cherry, W. Macherey, C.-C. Chiu, S. Yavuz, R. Pang, W. Li, and C. Raffel, "Monotonic infinite lookback attention for simultaneous machine translation," in *Proc. ACL*, Jul. 2019.

[24] J. Niehues, N.-Q. Pham, T.-L. Ha, M. Sperber, and A. Waibel, "Low-latency neural speech translation," in *Proc. Interspeech*, Sep. 2018, pp. 1293–1297.

[25] N. Arivazhagan, C. Cherry, I. Te, W. Macherey, P. Baljekar, and G. Foster, "Re-translation strategies for long form, simultaneous, spoken language translation," in *Proc. ICASSP*, May 2020.

[26] B. Li, S.-y. Chang, T. N. Sainath, R. Pang, Y. He, T. Strohman, and Y. Wu, "Towards fast and accurate streaming end-to-end ASR," in *Proc. ICASSP*, May 2020, pp. 6069–6073.

[27] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," *Proc. Interspeech*, pp. 1468–1472, Sep. 2015.

[28] T. N. Sainath, R. Pang, D. Rybach, B. Garcıa, and T. Strohman, "Emitting word timings with end-to-end models," in *Proc. Interspeech*, Oct. 2020, pp. 3615–3619.

[29] J. Yu, C.-C. Chiu, B. Li, S.-y. Chang, T. N. Sainath, Y. He, A. Narayanan, W. Han, A. Gulati, Y. Wu *et al.*, "FastEmit: Low-latency streaming ASR with sequence-level emission regularization," *arXiv preprint arXiv:2010.11148*, 2020.

[30] M. Hosseinzadeh and Y. Wang, "Video captioning of future frames," in *Proc. WACV*, Jan. 2021, pp. 980–989.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, Dec. 2017, pp. 5998–6008.

[32] V. Iashin and E. Rahtu, "A better use of audio-visual cues: Dense video captioning with bi-modal transformer," in *Proc. BMVC*, 2020.

[33] A. P. Shah, S. Geng, P. Gao, A. Cherian, T. Hori, T. K. Marks, J. Le Roux, and C. Hori, "Audio-visual scene-aware dialog and reasoning using audio-visual transformers with joint student-teacher learning," in *Proc. ICASSP*, 2022.

[34] C. Hori, A. Cherian, T. K. Marks, and T. Hori, "Joint Student-Teacher Learning for Audio-Visual Scene-Aware Dialog," in *Proc. Interspeech*, Sep. 2019, pp. 1886–1890.

[35] G. A. Sigurdsson, G. Varol, X. Wang, I. Laptev, A. Farhadi, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," *arXiv preprint arXiv:1604.01753*, 2016.

[36] H. Alamri, V. Cartillier, A. Das, J. Wang, A. Cherian, I. Essa, D. Batra, T. K. Marks, C. Hori, P. Anderson, S. Lee, and D. Parikh, "Audio visual scene-aware dialog," in *Proc. CVPR*, Jun. 2019.