*INTERSPEECH 2021*
30 August – 3 September, 2021, Brno, Czechia

# Optimizing Latency for Online Video Captioning Using Audio-Visual Transformers

*Chiori Hori, Takaaki Hori, Jonathan Le Roux*

Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA
{chori, thori, leroux}@merl.com

## Abstract

Video captioning is an essential technology to understand scenes and describe events in natural language. To apply it to real-time monitoring, a system needs not only to describe events accurately but also to produce the captions as soon as possible. Low-latency captioning is needed to realize such functionality, but this research area for online video captioning has not been pursued yet. This paper proposes a novel approach to optimize each caption's output timing based on a trade-off between latency and caption quality. An audio-visual Transformer is trained to generate ground-truth captions using only a small portion of all video frames, and to mimic outputs of a pre-trained Transformer to which all the frames are given. A CNN-based timing detector is also trained to detect a proper output timing, where the captions generated by the two Transformers become sufficiently close to each other. With the jointly trained Transformer and timing detector, a caption can be generated in the early stages of an event-triggered video clip, as soon as an event happens or when it can be forecasted. Experiments with the ActivityNet Captions dataset show that our approach achieves 94% of the caption quality of the upper bound given by the pre-trained Transformer using the entire video clips, using only 28% of frames from the beginning.

**Index Terms**: online video captioning, low-latency, audio-visual, transformer

## 1. Introduction

At any time instant, countless events that happen in the real world are captured by cameras and stored as massive video data resources. To effectively retrieve such recordings, whether in offline or online settings, video captioning is an essential technology thanks to its ability to understand scenes and describe events in natural language.

Since the S2VT system [1,2] was first proposed, video captioning has been actively researched in the field of computer vision [3–7] using sequence-to-sequence models in an end-to-end manner [8]. Its goal is to generate a video description (caption) about objects and events in a video clip. To further leverage audio features to identify events, [9] proposed the multimodal attention approach to fuse audio and visual features such as VGGish [10] and I3D [11] to generate video captions. Such video clip captioning technologies have been expanded to offline video stream captioning technologies such as dense video captioning [12] and progressive video description generator [13], where all salient events in a video stream are temporally localized, and event-triggered captions are generated in a multi-thread manner. While all video captioning technologies had so far been based on LSTM, [14] successfully applied the Transformer [15–17] together with the audio-visual attention framework [9]. In that work, the audio-visual Transformer was tested using the ActivityNet Captions dataset [12] within an offline video captioning system and achieved the best performance for the dense video captioning task. However, such offline video captioning technologies are not practical in real-time monitoring or surveillance systems, in which it is essential not only to describe events accurately but also to produce captions as soon as possible to find and report the events quickly. Low-latency captioning is required to realize such functionality, but this research area has not been pursued yet.

This paper proposes a novel approach that optimizes the output timing for each caption based on a trade-off between latency and caption quality. We train a low-latency audio-visual Transformer composed of (1) a Transformer-based caption generator which tries to generate ground-truth captions after only seeing a small portion of all video frames, and also to mimic the outputs of a similar pre-trained caption generator that is allowed to see the entire video, and (2) a CNN-based timing detector that can find the best timing to output a caption, such that the captions ultimately generated by the above two Transformers become sufficiently close to each other.

The proposed jointly-trained caption generator and timing detector can generate captions in an early stage of a video clip, as soon as an event happens. Additionally, this framework has the potential to forecast future events in online captions. Furthermore, by combining multimodal sensing information, an event can be recognized at an earlier timing triggered by the earliest cue in one of the modalities without waiting for other cues in other modalities. For example, the proposed approach has the potential to generate captions earlier than a visual cue's timing based on the timing of an audio cue. Such a low-latency online video captioning using multimodal sensing information will contribute not only to retrieve events quickly but also to answer questions about scenes earlier [18, 19].

## 2. Related work

There are some works on low-latency end-to-end sentence generation for machine translation (MT) and automatic speech recognition (ASR). To realize real-time interpretation systems, simultaneous translation using greedy decoding was proposed and opened up the issue of streaming for neural MT (NMT) [20–24]; an emission point when a phrase is fully translated into a target language was incrementally determined. Another approach iteratively retranslates by concatenating subsequent words and updating the output [25,26]. The goal is to generate a partial translation in the meantime before a full source sentence is translated. In contrast, our goal is to generate a full caption as soon as the system believes enough cues have been captured before seeing the entire video. Real-time ASR technology is also essential for applications such as closed captions. Some end-to-end systems regularize or penalize the emission delay using endpoint detection, and penalty terms that constrain alignments were proposed [27–30]. There, the target is to gen-

Copyright © 2021 ISCA

http://dx.doi.org/10.21437/Interspeech.2021-1975

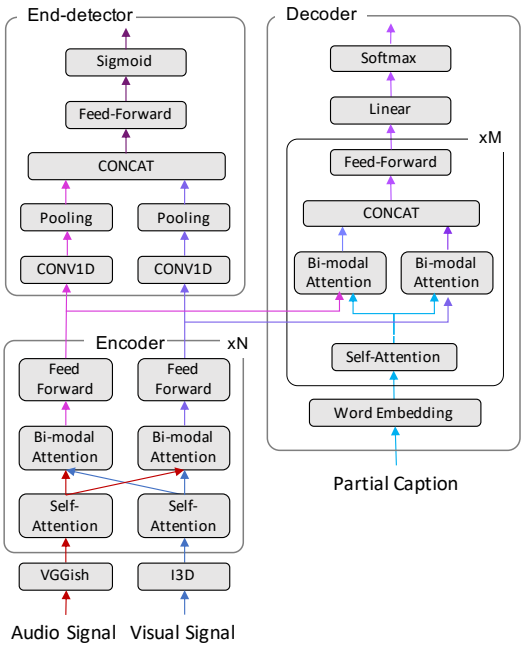


Figure 1: *Online multi-modal captioning Transformer.*

erate a transcription slightly earlier from the end of an utterance. In contrast, our target is to generate video captions as early as possible before the end of events.

In the field of computer vision, PickNet was proposed to find salient visual frames sufficient to generate video captions, where the target number of frames was given, and the captioning capability using only the selected frames was evaluated [31]. The paper mentions that it may be possible to apply PickNet to online captioning, showing a sample use case, but no quantitative evaluation was performed. Another work relevant to online video captioning attempts to anticipate caption generation for future frames [32]. This approach exploited the current event features as a contextual feature and input them into a captioning module to generate future captions. This technology uses temporal dependency between events in a sequence.

## 3. Online multi-modal captioning Transformer

### 3.1. Architecture

We describe the proposed low-latency video captioning model. Figure 1 illustrates the model architecture, which consists of an audio-visual encoder, an end detector, and a caption decoder, where the encoder is shared by the detector and the decoder. Our model is based on the Transformer architecture [15] and its multimodal extension [14], but it receives video and audio features in a streaming manner, and the end detector decides when to generate a caption for the feature sequence the model has received until that moment.

Given a video stream, the audio-visual encoder extracts VGGish and I3D features from the audio and video tracks, respectively, where the frame rate may be different on each track. 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. Typically, this encoder block is repeated $N$ times, e.g., $N = 6$ or greater. The final encoded representation is obtained via the $N$-th encoder block.

Let $X^A$ and $X^V$ be audio and visual signals. First, the feature extraction module is applied to the input signals as

$$A^0 = \mathrm{VGGish}(X^A), \quad V^0 = \mathrm{I3D}(X^V), \tag{1}$$

to obtain feature vector sequences corresponding to the VGGish and I3D features, respectively. Each encoder block computes hidden vector sequences as

$$\bar{A}^n = A^{n-1} + \mathrm{MHA}(A^{n-1}, A^{n-1}, A^{n-1}), \tag{2}$$
$$\bar{V}^n = V^{n-1} + \mathrm{MHA}(V^{n-1}, V^{n-1}, V^{n-1}), \tag{3}$$
$$\tilde{A}^n = \bar{A}^n + \mathrm{MHA}(\bar{A}^n, \bar{V}^n, \bar{V}^n), \tag{4}$$
$$\tilde{V}^n = \bar{V}^n + \mathrm{MHA}(\bar{V}^n, \bar{A}^n, \bar{A}^n), \tag{5}$$
$$A^n = \tilde{A}^n + \mathrm{FFN}(\tilde{A}^n), \tag{6}$$
$$V^n = \tilde{V}^n + \mathrm{FFN}(\tilde{V}^n), \tag{7}$$

where MHA and FFN denote multi-head attention and feed-forward network, respectively. Layer normalization [33] is applied before every MHA and FFN layers, but it is omitted from the equations for simplicity. MHA takes three arguments, query, key, and value vector sequences [15]. The self-attention layer extracts temporal dependency within each modality, where the arguments for MHA are all the same, i.e., $A^{n-1}$ or $V^{n-1}$, as in (2) and (3). The bi-modal attention layers further extract cross-modal dependency between audio and visual features, taking the keys and values from the other modality as in (4) and (5). After that, the feed-forward layers are applied in a point-wise manner. The encoded representations for audio and visual features are obtained as $A^N$ and $V^N$.

The end detector receives the encoded representation based on the audio-visual information available at the moment. The role of the end detector is to decide whether the system should generate a caption or not for the given encoded features. The detector first processes the encoded vector sequence from each modality with stacked 1D-convolution layers as

$$A_c = \mathrm{Conv1D}(A^N), \quad V_c = \mathrm{Conv1D}(V^N). \tag{8}$$

Each time-convoluted sequences are then summarized into a single vector through pooling and concatenation operations:

$$H = \mathrm{Concat}(\mathrm{MeanPool}(A_c), \mathrm{MeanPool}(V_c)) \tag{9}$$

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

$$P(d = 1 | X^A, X^V) = \sigma(\mathrm{FFN}(H)). \tag{10}$$

Once the end detector provides a higher probability than a threshold, e.g., $P(d = 1|X^A, X^V) > 0.5$, the decoder generates a caption based on the encoded representation $(A^N, V^N)$.

The decoder iteratively predicts the next word from a starting token (<sos>). At each iteration step, it receives a partial caption that has already been generated, and predicts the next word by applying $M$ decoder blocks and a prediction network, where each word is assumed to be converted to a word embedding vector.

Let $Y_i^0$ be partial caption <sos>, $y_1, ..., y_i$ after $i$ iterations. Each decoder block has self-attention, bi-modal source attention, and feed-forward layers:

$$\bar{Y}_i^m = Y_i^{m-1} + \mathrm{MHA}(Y_i^{m-1}, Y_i^{m-1}, Y_i^{m-1}), \tag{11}$$

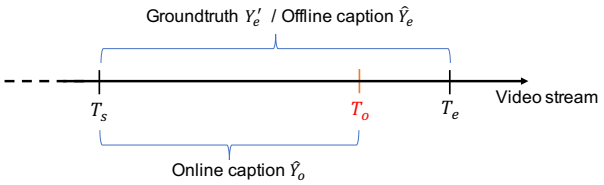


Figure 2: *Online and offline captions.*

$$\bar{Y}_i^{Am} = \bar{Y}_i^m + \mathrm{MHA}(\bar{Y}_i^m, A^N, A^N), \quad (12)$$

$$\bar{Y}_i^{Vm} = \bar{Y}_i^m + \mathrm{MHA}(\bar{Y}_i^m, V^N, V^N), \quad (13)$$

$$\tilde{Y}_i^m = \mathrm{Concat}(\bar{Y}_i^{Am}, \bar{Y}_i^{Vm}), \quad (14)$$

$$Y_i^m = \tilde{Y}_i^m + \mathrm{FFN}(\tilde{Y}_i^m). \quad (15)$$

The self-attention layer converts the word vectors to high-level representations considering their temporal dependency in (11). The bi-modal source attention layers update the word representations based on the relevance to the encoded multi-modal representations in (12) and (13). A feed-forward layer is then applied to the outputs of the bi-modal attention layers in (14) and (15). Finally, a linear transform and a softmax operation are applied to the output of the $M$-th decoder block to obtain the probability distribution of the next word as

$$P(\mathbf{y}_{i+1}|Y_i, X^A, X^V) = \mathrm{Softmax}(\mathrm{Linear}(Y_i^M)), \quad (16)$$

$$\hat{y}_{i+1} = \underset{y \in \mathcal{V}}{\mathrm{argmax}}\, P(\mathbf{y}_{i+1} = y|Y_i, X^A, X^V), \quad (17)$$

where $\mathcal{V}$ denotes the vocabulary.

After picking the one-best word $\hat{y}_{i+1}$, the partial caption is extended by adding the selected word to the previous partial caption as $Y_{i+1} = Y_i, \hat{y}_{i+1}$. This is a greedy search process that ends if $\hat{y}_{i+1} =$ <eos>, which represents an end token. It is also possible to pick multiple words with highest probabilities and consider multiple candidates of captions according to the beam search technique.

Similar architectures have been used for dense video captioning tasks [14, 34], where an event localization network is placed on top of the encoder similarly to our end detector. A difference with those models is that the localization network is assumed to access all frames of the video and chooses a set of regions, which potentially includes specific events, while our end detector can access only partial frames from the beginning or a certain point to the current frame and detect a timing at which the system should emit the caption. Thus, our model is designed and trained for online captioning.

### 3.2. Training

We learn the multi-modal encoder, the end detector, and the caption decoder jointly, so that the model achieves a caption quality comparable to that for a complete video, even if the given video is shorter than the original one by truncating the later part.

Two types of loss functions are combined, a captioning loss to improve the caption quality and an end detection loss to detect a right timing to emit a caption. Figure 2 shows an example of video stream, where an event has started at time $T_s$ and ends at $T_e$, and is associated with ground-truth caption $Y'_e$. If time $T_o$ is picked as the emission timing, the captioning decoder generates a caption based on the multi-modal input signal $X_{T_s:T_o} = (X^A_{T_s:T_o}, X^V_{T_s:T_o})$.

The captioning loss is based on a standard cross entropy loss for the ground-truth caption $Y'_e$,

$$\mathcal{L}_{CE} = -\log P(Y'_e|X_{T_s:T_o}; \theta_C), \quad (18)$$

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.,

$$\mathcal{L}_{KL} = -\sum_{i=1}^{|Y'_e|} \sum_{y \in \mathcal{V}} P(y|Y'_{e,i}, X_{T_s:T_e}; \bar{\theta}_C) \log P(y|Y'_{e,i}, X_{T_s:T_o}; \theta_C). \quad (19)$$

This is a student-teacher learning approach to exploit another model's superior description power [35], where the teacher model $\bar{\theta}_C$ predicts a caption using entire video clip $X_{T_s:T_e}$ and the student model $\theta_C$ tries to mimic the teacher's predictions using only the truncated video clip $X_{T_s:T_o}$. This makes the training more stable and achieves better performance.

The end 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. In this work, we decide the right timing based on whether or not the captioning decoder can generate a relevant caption, that is, a caption sufficiently close to the ground-truth $Y'_e$ or the caption $\hat{Y}_e$ generated for the entire video clip $X_{T_s:T_e}$ using the pre-trained model $\bar{\theta}_C$. The detection loss is computed as

$$\mathcal{L}_D = -\log P(d|X_{T_s:T_o}; \theta_D), \quad (20)$$

where $d$ is determined based on

$$d = \begin{cases} 1 & \text{if } \max(\mathrm{Sim}(Y'_e, \hat{Y}_o), \mathrm{Sim}(\hat{Y}_e, \hat{Y}_o)) \geq S, \\ 0 & \text{otherwise,} \end{cases} \quad (21)$$

where $\mathrm{Sim}(\cdot, \cdot)$ denotes a similarity measure between two word sequences. In this work, we use word accuracy computed in a teacher-forcing manner. $S \in (0, 1]$ is a pre-determined threshold which judges whether or not the online caption $\hat{Y}_o$ is sufficiently close to the references $Y'_e$ and $\hat{Y}_e$.

The training process for model $\theta = (\theta_C, \theta_D)$ repeats the following steps:

1. Sample $T_o \sim \mathrm{Uniform}(T_s, T_e)$,
2. Compute loss $\mathcal{L} = \alpha\mathcal{L}_{CE} + \beta\mathcal{L}_{KL} + \gamma\mathcal{L}_D$,
3. Update $\theta$ using $\nabla_\theta \mathcal{L}$.

### 3.3. Inference

The inference is performed in two steps:

1. Find $\hat{T}_o$ that first satisfies $P(d = 1|X_{T_s:\hat{T}_o}; \theta_D) > F$,
2. Generate a caption based on

$$\hat{Y}_o = \underset{Y \in \mathcal{V}^*}{\mathrm{argmax}}\, P(Y|X_{T_s:\hat{T}_o}; \theta_C), \quad (22)$$

where $F$ is a pre-determined threshold to control the sensitivity of end detection. Note that we assume that $T_s$ is already determined.

## 4. Experiments

We evaluate our low-latency caption generation method using the ActivityNet Captions dataset [12], which consists of 100k

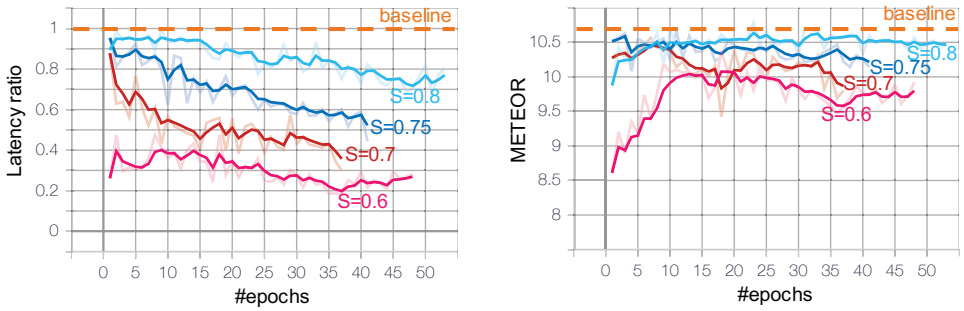


Figure 3: *Latency ratio (left) and METEOR score (right) in training with different thresholds $S$ in* (21).

caption sentences associated with temporal localization information based on 20k YouTube videos. Although the conventional video description dataset MSVD (YouTube2Text) [36] and MSR-VTT [37] have 41 and 20 ground-truth captions for each video clip respectively, ActivityNet only has one for each event. The dataset is split into 50%, 25%, and 25% for training, validation, and testing. However, since the ground-truth captions for the test set are not available, we split the validation set into two subsets on which we report the performance as done in a prior study [14]. The average duration of a video clip is 35.5, 37.7, and 40.2 seconds for the training set and the validation subsets 1 and 2, respectively. We used VGGish and I3D features provided by the author of [14]. 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 1024-dimensional vector sequence for the video track, where each visual frame corresponds to a 2.56 s segment without overlap.

A multi-modal Transformer was first trained with entire video clips and their ground-truth captions. This model was used as a baseline and teacher model. We used $N = 2$ encoder blocks and $M = 2$ decoder blocks, and the number of attention heads was 4. The vocabulary size was 10,172, and the dimension of word embedding vectors was 300.

The proposed model for online captioning 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 end detector. In the training process, we consistently used $\alpha = \beta = \gamma = 1/3$ for the loss function. The dimensions of hidden activations in audio and visual attention layers were 128 and 1024, respectively. The dropout rate was set to 0.1, and a label smoothing technique was also applied. The end detector had 2 stacked 1D-convolution layers, with a ReLU non-linearity in between. The performance was measured by BLEU3, BLEU4, and METEOR scores.

Figure 3 shows the latency ratio (left) and METEOR scores (right) on validation subset 1 when training with different $S$ in (21). The latency ratio indicates the ratio of the video duration used for captioning to the duration of the original video clip. With the baseline model, the latency ratio is always 1, which means all frames are used to generate captions. With our proposed method, the latency ratio and METEOR scores change depending on the value of $S$, where a larger $S$ gives a stricter condition on the caption accuracy, resulting in later detection, while a smaller $S$ results in earlier detection. As learning proceeds, the latency ratio gradually decreases, but the METEOR score tends to maintain high values close to the baseline. This result demonstrates that the learning process works to reduce the latency while maintaining caption quality.

Table 1: *Performance of baseline and proposed systems. The scores are averaged on the captions in validation subset 1. The average duration of a video clip is 37.7 seconds. ST denotes Student-Teacher learning.*

| Method | Latency | BLEU-3 | BLEU-4 | METEOR |
|---|---|---|---|---|
| Baseline [14] | 100% | 4.66 | 2.05 | 10.67 |
| Naive method | 55% | 4.31 | 1.76 | 10.20 |
| Naive method | 33% | 3.69 | 1.37 | 9.59 |
| Proposed (w/o ST) | 55% | 4.22 | 1.77 | 10.38 |
| Proposed | 56% | **4.40** | **1.82** | **10.45** |
| Proposed (w/o ST) | 29% | 3.75 | 1.52 | 9.93 |
| Proposed | 28% | **3.84** | **1.57** | **10.00** |
| Baseline (visual only) | 100% | 4.08 | 1.80 | 10.21 |
| Proposed (visual only) | 54% | 3.82 | 1.61 | 10.05 |
| Proposed (visual only) | 30% | 3.45 | 1.42 | 9.71 |

Table 1 compares captioning methods in BLEU and METEOR scores on validation subset 1. The model selected for evaluation was trained with $S = 0.6$ and had the best METEOR score on validation subset 2. We controlled the latency with the detection threshold $F$. As shown in the table, our proposed method at a 55% latency achieves 10.45 METEOR score with only a small degradation, which corresponds to 98% of the baseline score 10.67 [14]. It also achieves 10.00 METEOR score at a 28% latency, which corresponds to 94% of the baseline. We also evaluated a naive method which takes video frames from the beginning with a fixed ratio to the original video length and runs the baseline captioning on the truncated video clip. The results show that the proposed approach clearly outperforms the naive method at an equivalent latency.

The table also includes the results for a unimodal Transformer that receives only the visual feature. The results show that the proposed method works for the visual feature only, but the performance is degraded due to the lack of the audio feature. This result indicates that the audio feature is essential even in the proposed low-latency method.

## 5. Conclusions

In this paper, we proposed a low-latency audio-visual captioning method, which describes events accurately and quickly without waiting for the end of video clips. The proposed method optimizes each caption's output timing based on a trade-off between latency and caption quality. We have demonstrated that the proposed system can generate captions in early stages of event-triggered video clips, achieving 94% of the caption quality of the upper bound given by a Transformer processing the entire video clips, using only 28% of frames (10.6 seconds) on average from the beginning.

# 6. References


[1] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence – Video to text," in *Proc. ICCV*, Dec. 2015, pp. 4534–4542.

[2] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," in *Proc. NAACL HLT*, May 2015.

[3] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *Proc. ICCV*, Dec. 2015, pp. 4507–4515.

[4] A. Rohrbach, M. Rohrbach, and B. Schiele, "The long-short story of movie description," in *Proc. GCPR*, Oct. 2015, pp. 209–221.

[5] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *Proc. CVPR*, Jun. 2016, pp. 4594–4602.

[6] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, "Video paragraph captioning using hierarchical recurrent neural networks," in *Proc. CVPR*, Jun. 2016, pp. 4584–4593.

[7] M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya, "Learning joint representations of videos and sentences with web image search," in *Proc. ECCV*, Oct. 2016, pp. 651–667.

[8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *Proc. ICML*, Jul. 2015.

[9] C. Hori, T. Hori, T.-Y. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi, "Attention-based multimodal fusion for video description," in *Proc. ICCV*, Oct. 2017.

[10] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *Proc. ICASSP*, Mar. 2017.

[11] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proc. CVPR*, Jul. 2017.

[12] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. Carlos Niebles, "Dense-captioning events in videos," in *Proc. ICCV*, Oct. 2017, pp. 706–715.

[13] Y. Xiong, B. Dai, and D. Lin, "Move forward and tell: A progressive generator of video descriptions," in *Proc. ECCV*, Sep. 2018, pp. 468–483.

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

[15] 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.

[16] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang, "A comparative study on transformer vs RNN in speech applications," in *Proc. ASRU*, Dec. 2019.

[17] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. ICASSP*, Apr. 2018, pp. 5884–5888.

[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] 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.

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] Y. Chen, S. Wang, W. Zhang, and Q. Huang, "Less is more: Picking informative frames for video captioning," in *Proc. ECCV*, Sep. 2018, pp. 358–373.

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

[33] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," in *Proc. NIPS Deep Learning Symposium*, 2016.

[34] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-end dense video captioning with masked transformer," in *Proc. CVPR*, Jun. 2018, pp. 8739–8748.

[35] 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.

[36] 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.

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