DSTC10-AVSD Submission System with Reasoning using Audio-Visual Transformers with Joint Student-Teacher Learning

Ankit Parag Shah[†§], **Takaaki Hori**[†], **Jonathan Le Roux**[†], **Chiori Hori**[†]

[†]Mitsubishi Electric Research Laboratories (MERL)
[§]Carnegie Mellon University

Abstract

We participated in the third challenge for the Audio-Visual Scene-Aware Dialog (AVSD) task in DSTC10. The target of the task was updated by two modifications: 1) the human-created description is unavailable at inference time, and 2) systems must demonstrate temporal reasoning by finding evidence from the video to support each answer. The baseline system built using an AV-transformer was released along with the new dataset including temporal reasoning for DSTC10-AVSD. This paper introduces a new system that extends the baseline system with attentional multimodal fusion, joint student-teacher learning (JSTL), and model combination techniques, achieving state-of-the-art performances on the AVSD datasets for DSTC7, DSTC8, and DSTC10. We also propose two temporal reasoning methods for AVSD: one attention-based, and one based on a time-domain region proposal network (RPN). We confirmed our system outperformed the baseline system and the previous state of the art for the AVSD test sets for DSTC7, DSTC8, and DSTC10. Furthermore, the temporal reasoning using RPN outperformed the attention method of the baseline system.

Introduction

To encourage development of dialog system technologies that enable an agent to discuss audio-visual scenes with humans, two challenges on audio-visual Scene-Aware Dialog (AVSD) at DSTC7 and DSTC8 (D'Haro et al. 2020; Kim et al. 2021) were held using the Question Answering (QA) dialog about videos in the Charades dataset (Sigurdsson et al. 2016). The AVSD task defined and the dataset prepared in DSTC were the first attempt to promote the combination of audio-visual question-answering systems and conversation systems into a single framework (Hori et al. 2019a; Alamri et al. 2019). The AVSD task proposed in DSTC is to generate a system response to a query, where the query is part of a multi-turn dialog about a video. Challenge participants used the video, its associated audio, and the dialog text to train end-to-end deep learning models to produce the answers. In addition, the systems had access to human-created video captions. The AVSD task can be seen as an extension to video data of both the *visual question answering* (VQA) task (Antol et al. 2015; Zhang et al. 2016; Goyal et al. 2017; Tapaswi et al. 2016), in which the goal is to generate answers to questions about a scene in a static image, and the *visual dialog* task (Das et al. 2016), in which an AI agent holds a meaningful dialog with humans about a static image using natural, conversational language (Das et al. 2017). Another progenitor to AVSD is the task of *video description* (text summarization of videos), which (Hori et al. 2017) addressed utilizing multimodal attention mechanisms that selectively attend to different input modalities (feature types) such as spatio-temporal motion features and audio features, in addition to temporal attention. Combining video description technologies like these with end-to-end dialog systems enables scene-aware dialog systems that make use of multimodal information, such as audio and visual features. In a more recent work, spatio-temporal reasoning has been shown to improve performance on AVSD tasks (Geng et al. 2021). Recently, Transformer-based AVSD systems outperform LSTM-based ones (Le et al. 2019; Li et al. 2021).

The task setup for AVSD in DSTC7–8 allowed participants to use human-created video captions to help generate answers for the dialog questions, and systems that used these human-generated captions significantly outperformed systems that did not. However, since such human-created descriptions are not available in real-world applications of an AVSD system, in practice a system needs to learn to produce the answers without the captions. There are two other design difficulties that such text-based descriptions introduce that may skew the evaluation: (i) some descriptions already include parts of the answers that are used in the evaluations, making audio-visual inference redundant, and (ii) language models trained using a simple (and limited) QA dataset may generate answers using frequently-occurring text patterns in the training data, without needing to use audio-visual cues (e.g., Q: How many people are in the scene? A: Two people). The results from AVSD in DSTC7–8 suggest there is still an opportunity to design better audio-visual reasoning methods to approach the performance achieved when using manual video descriptions, but without using these descriptions at test time. Furthermore, real systems should ideally be able to show the evidence supporting their generated answers, by pointing to the relevant segments of the video. To encourage progress towards this end, a third AVSD challenge was proposed in DSTC10.

In this paper, we introduce the DSTC10-AVSD challenge

Table 1: Audio-Visual Scene-aware Dialog data set for DSTC10.

	training	validation	test
#dialogs	7,659	1,787	1,804
#turns	153,180	35,740	28,406
#words	1,450,754	339,006	272,606

task, the goals of which are: 1) answer generation without human-created captions at inference time, and 2) temporal reasoning (providing evidence) for the generated answers. Furthermore, we develop an AVSD baseline system using an AV-transformer (Iashin and Rahtu 2020). In addition, we propose a novel system that extends this AV-transformer using attentional multimodal fusion (Hori et al. 2017), joint student-teacher learning (JSTL) (Hori et al. 2019b), and model combination techniques. We also propose two temporal reasoning methods for AVSD: one attention-based, and one based on a region proposal network (RPN). Results show that our extended AV-transformer achieves state-of-the-art on DSTC 7, 8, and 10 when combined with our LSTM-based AVSD system (Hori et al. 2019b).

Audio-Visual Scene-Aware Dialog data set

We base the new Audio-Visual Scene-Aware Dialog (AVSD) task for DSTC10 on the AVSD dataset from DSTC7–8 (D'Haro et al. 2020; Kim et al. 2021). For the AVSD data, we collected text-based dialogs on short videos from the popular Charades dataset (Sigurdsson et al. 2016), which consists of untrimmed and multi-action videos (each video also has an audio track) and comes with human-generated descriptions of the scene. In the AVSD dialog case, two 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 (Alamri et al. 2019). Table 1 shows the size of the data used for DSTC10. For this year's challenge (DSTC10), we collected additional data for temporal reasoning, in which humans watched the videos and read the dialogues, then identified segments of the video containing evidence to support a given answer. Humans identifying the reasoning for the identified segments had to identify the segments based on visual evidence and/or audio evidence with appropriate fields to provide reasoning.

Baseline Model

Our DSTC10-AVSD baseline model is an AV-transformer architecture (Iashin and Rahtu 2020), shown in Fig. 1. The system employs a transformer-based encoder-decoder, including a bimodal attention mechanism (Bahdanau, Cho, and Bengio 2014; Chorowski et al. 2015) that lets it learn interdependencies between audio and visual features.

Given a video stream, the audio-visual encoder extracts VGGish (Hershey et al. 2017) and I3D (Carreira and Zisserman 2017) features from the audio and video tracks, respectively, and encodes these using self-attention, bimodal attention, and feed-forward layers. Typically, this encoder block is repeated N times, e.g., $N \geq 6$. More formally, let X^A

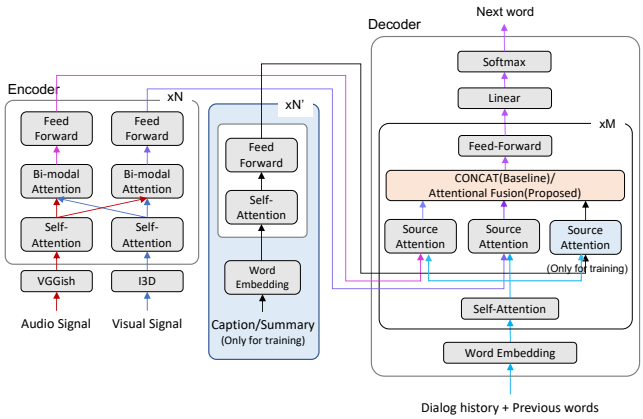


Figure 1: Baseline and extended AV-transformer. Our extended system adds the JSTL modules (blue and orange boxes) to the baseline.

and X^V denote audio and visual signals. First, the feature extraction module extracts VGGish and I3D feature vector sequences from the input signals:

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

The nth encoder block computes hidden vector sequences as:

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

where MHA and FFN denote multi-head attention and feed-forward network, respectively. Layer normalization (Ba, Kiros, and Hinton 2016) is applied before every MHA and FFN layer, but it is omitted from the equations for simplicity. MHA takes three arguments: query, key, and value vector sequences (Vaswani et al. 2017). 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 bimodal 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 decoder receives the encoder outputs and the dialog history until the current question, and starts generating the answer sentence from the beginning token (<sos>) placed at the end of the last question. At each iteration step, it receives the preceding word sequence and predicts the next word by applying M decoder blocks and a prediction network. In each decoder block, the encoded audio-visual features are combined with each word using the bimodal attention layers. Let Y_i be a dialog history plus preceding

word sequence $h_1, \ldots, h_L, \texttt{<sos>}, y_1, \ldots, y_i$ after i iterations and Y_i^0 be a word embedding vector sequence given by $Y_i^0 = \mathrm{Embed}(Y_i)$.

Each decoder block has self-attention, bimodal source attention, and feed-forward layers. Computations within the m-th block are as follows:

$$\bar{Y}_i^m = Y_i^{m-1} + \mathrm{MHA}(Y_i^{m-1}, Y_i^{m-1}, Y_i^{m-1}), \quad (8)$$

$$\bar{Y}_i^{Am} = \bar{Y}_i^m + \mathrm{MHA}(\bar{Y}_i^m, A^N, A^N), \quad (9)$$

$$\bar{Y}_i^{Vm} = \bar{Y}_i^m + \mathrm{MHA}(\bar{Y}_i^m, V^N, V^N), \quad (10)$$

$$\tilde{Y}_i^m = \mathrm{Concat}(\bar{Y}_i^{Am}, \bar{Y}_i^{Vm}), \quad (11)$$

$$Y_i^m = \tilde{Y}_i^m + \mathrm{FFN}(\tilde{Y}_i^m). \quad (12)$$

The self-attention layer converts the word vectors to high-level representations considering their temporal dependency in (8). The bimodal source attention layers update the word representations based on the relevance to the encoded multimodal representations in (9) and (10). A feed-forward layer is then applied to the outputs of the bimodal attention layers in (11) and (12). Finally, a linear transform and softmax operation are applied to the output of the M-th decoder block to obtain the probability distribution of the next word as

$$P(\mathbf{y}_{i+1}|Y_i, X^A, X^V) = \mathrm{Softmax}(\mathrm{Linear}(Y_i^M)). \quad (13)$$

At inference time, we can pick the one-best word $\hat{y}_{i+1}$ for y_{i+1} as

$$\hat{y}_{i+1} = \underset{y \in \mathcal{V}}{\mathrm{argmax}}\, P(\mathbf{y}_{i+1} = y|Y_i, X^A, X^V), \quad (14)$$

where $\mathcal{V}$ denotes the vocabulary, and the answer sentence is extended by adding the selected word to the already generated word sequence as $Y_{i+1} = Y_i, \hat{y}_{i+1}$. This is a greedy search process that ends if $\hat{y}_{i+1} = \texttt{<eos>}$, which represents an end token. It is also possible to pick multiple words with highest probabilities and consider multiple candidates for the answer sentence using a beam search.

Extended AV-transformer

We extend the baseline AV-transformer by applying attentional multimodal fusion (Hori et al. 2017) and joint student-teacher learning (JSTL) (Hori et al. 2019b), which have successfully been applied to an LSTM-based AVSD system (Hori et al. 2019a) but have not previously been applied to transformer-based systems. In this paper, we propose to extend the AV-transformer with these techniques and test their effectiveness.

Fig. 1 shows the teacher model of the extended AV-transformer, which has a caption/summary encoder in the encoder and an attentional fusion layer in the decoder. In student-teacher learning, a student model without the caption/summary encoder and its attention module in the decoder is trained using the teacher model output as the target distribution.

To further improve the performance, we combine the extended AV-transformer with the LSTM-based model trained with student-teacher learning as well, where the two decoder outputs are linearly combined in the log domain during the beam search. This system combination method aims to exploit the complementary information between the two models to improve the performance.

Attentional Multimodal Fusion

The baseline AV-transformer in Fig. 1 concatenates multimodal encoder outputs in each decoder block, assuming that the audio and visual features have equal contribution to the next word prediction regardless of the given question and the generated answer. However, prior work has shown that attentional multimodal fusion is effective for LSTM-based systems. In this work, we apply the attentional fusion technique to the AV-transformer. In the case of Transformer, we can use single-head attention (SHA) in each decoder block as

$$\tilde{\bar{Y}}_i^m = \mathrm{SHA}(\bar{Y}_i^m, \tilde{Y}_i^m, \tilde{Y}_i^m), \quad (15)$$

where $\tilde{Y}_i^m$ is here a concatenation of $\bar{Y}_i^{Am}$ and $\bar{Y}_i^{Vm}$. If the model has a caption/summary encoder, its output $\bar{Y}_i^{Cm}$ is also concatenated. In this case, $\tilde{Y}_i^m$ is a $3 \times D$ tensor including three modalities, each of which has a D-dimensional vector. Then, the fused vector $\tilde{\bar{Y}}_i^m$ is fed to the feed-forward layer.

Student-Teacher Learning

The goal of student-teacher learning is to obtain a student model that does not make use of the video caption or summary, which is trained to mimic a teacher model that has already been trained using the caption/summary text. Accordingly, the student model can be used to generate system responses without relying on the caption text, while hopefully achieving similar performance to the teacher model.

The student-teacher loss is a cross entropy loss with *soft* targets:

$$\mathcal{L}_{\mathrm{ST}} = -\sum_{i=1}^{|Y|} \sum_{y \in \mathcal{V}} \hat{P}(y|Y_{i-1}, X^A, X^V, X^C) \log P(y|Y_{i-1}, X^A, X^V), \quad (16)$$

where $\hat{P}(y|Y_{i-1}, X^A, X^V, X^C)$ denotes the probability distribution for the ith word obtained by the teacher network. Here, $P(y|Y_{i-1}, X^A, X^V)$ is the posterior distribution from the current student network (which is being trained), which is predicted without the caption text X^C.

Following our prior work, we also incorporate a decoder state similarity loss and a cross-entropy loss on the teacher for joint student-teacher learning as

$$\mathcal{L}_{\mathrm{JST}} = \mathcal{L}_{\mathrm{ST}} + \lambda_c \mathcal{L}_{\mathrm{MSE}} + \mathcal{L}_{\mathrm{CE}}^{(T)}, \quad (17)$$

where $\mathcal{L}_{\mathrm{MSE}} = \sum_{i=1}^{|Y|} \mathrm{MSE}(Y_i^m, \hat{Y}_i^m)$. Here, $\mathrm{MSE}(\cdot, \cdot)$ denotes the mean square error between two vectors, λ_c denotes a scaling factor, and the MSE loss is computed for the m-th block of the decoder to make the teacher's and student's hidden vectors closer. In this work, we set $m = M/2$, which indicates the block in the middle of the M decoder blocks. We aim here to compensate for missing input features at the decoder state level, so that the student model can hopefully exploit other modalities more actively. Furthermore, joint student-teacher learning updates not only the student network but also the teacher network. We use the standard cross entropy $\mathcal{L}_{\mathrm{CE}}^{(T)}$ for a *hard* target, only for the teacher network. Likewise, $\mathcal{L}_{\mathrm{ST}}$ is used only for the student network, while $\mathcal{L}_{\mathrm{MSE}}$ is used for both networks.

Temporal Reasoning

Temporal reasoning is the task of finding evidence supporting the generated answers, where the evidence corresponds to human-annotated time regions of the video that have been identified as supporting each ground-truth answer. Human annotators were allowed to choose multiple time regions for each question-answer pair, but most of the reasons consist of a single region.

Attention-based method

We built a baseline method for temporal reasoning based on attention weights obtained during decoding. The attention weights are computed to predict each word, where each attention weight corresponds to a certain time frame of input audio/visual features. Thus, a high weight means that the corresponding time frame is strongly correlated to a word in the generated answer. Given an attention weight distribution, we can compute mean μ and standard deviation σ of the distribution, and roughly estimate the time region as $\mu \pm \nu\sigma$, where ν is a hyperparameter. Since we have multiple attention distributions over the word sequence, attention heads, and layers, we use their averaged distribution. This method finds only one time region for each answer, and it requires no special training to select time regions.

RPN-based time region detection

We also built a CNN-based temporal reasoning model, which accepts encoder outputs of the AV-transformer and an embedded QA pair to predict temporal regions that support the answer. The model employs a time-domain region proposal network (RPN) (Ren et al. 2015; Iashin and Rahtu 2020), where Conv1D modules with different kernel sizes accept frame-level outputs of the multimodal encoders, each of which is concatenated with the QA pair embedded by the decoder followed by mean pooling. It predicts the center position, the region length, and the confidence score of each region candidate. We pick high-confidence regions from the candidates using a predetermined threshold.

Experiments

We evaluate our AV-transformer using the AVSD datasets from DSTC7, DSTC8, and DSTC10. Training and validation sets are common across the three challenges, but the test sets are different.

Conditions

We extracted VGGish audio features (Hershey et al. 2017) and I3D video features (Carreira and Zisserman 2017) from each video clip, where I3D features consisted of sequences of 2040-dimensional RGB and flow vectors, and VGGish features were sequences of 128-dimensional vectors. The RGB and flow features were concatenated before feeding them to the encoder.

The baseline AV-Transformer has projection layers before the first audio-visual encoder block, where the audio and visual features are projected to 64 and 128 dimensional vectors, respectively. The encoder has 2 encoder blocks, in which the audio and visual attention layers have 64 and 128 dimensions, and their feed-forward layers have 256 and 512 dimensions, respectively. The decoder has 2 decoder blocks, in which 300-dimensional GloVe word vectors (Pennington, Socher, and Manning 2014) are projected to 256-dimensional embedding vectors and fed to 256-dimensional attention layers followed by 1024-dimensional feed-forward layers. The baseline system employs greedy search to generate the answers.

The quality of the automatically generated sentences was evaluated with objective measures to compare the similarity between the generated sentences and the ground truth sentences. We used the evaluation code for MS COCO caption generation[1] for objective evaluation of system outputs, which supports automated metrics such as BLEU, METEOR, ROUGE_L, and CIDEr.

Table 2: Evaluation results on DSTC7-AVSD test set. DSTC7's best system (Sanabria, Palaskar, and Metze 2019) does not report results without captions, so we report their results with captions.

Model	BLEU4	METEOR	ROUGE_L	CIDEr
Baseline AV-transformer	0.296	0.214	0.485	0.771
+ Hyperparameter tuning	0.362	0.237	0.522	0.974
+ Beam search	0.380	0.239	0.530	0.998
+ Attentional MM fusion	0.391	0.248	0.536	1.013
+ JST learning	0.401	0.256	0.549	1.051
+ Comb. with LSTM	**0.406**	**0.262**	**0.554**	**1.079**
LSTM + JST learning (Hori et al. 2019b)	0.382	0.254	0.537	1.005
DSTC7 best w/ cap. (Sanabria et al. 2019)	0.394	0.267	0.563	1.094

Table 3: Evaluation results on DSTC8-AVSD test set.

Model	BLEU4	METEOR	ROUGE_L	CIDEr
Baseline AV-transformer	0.281	0.203	0.468	0.701
Extended AV-transformer	0.380	0.242	0.535	0.957
+ Comb. with LSTM	**0.394**	**0.250**	**0.545**	0.997
DSTC8 best (Li et al. 2021) w/o cap.	0.387	0.249	0.544	**1.022**

Results and Discussion

Table 2 shows the evaluation results on the DSTC7 test set. To improve the performance from the baseline, we first tuned the hyperparameters using the validation set, where we made the decoder network deeper to 6 blocks and reduced the dimension of the attention layers to 200. We shrank the dialog history given to the decoder into just the previous question. In addition, we applied a learning rate control that halves the learning rate of Adam optimizer if the validation loss did not decrease after each training epoch. With this tuning, we obtained substantial improvement, e.g., $0.296 \rightarrow 0.332$ in BLEU4. Then, we applied the beam search technique with beam size 5, which further improved the performance.

We extend the AV-transformer by adding attentional multimodal (MM) fusion and joint student-teacher (JST) learning, achieving further performance improvement. Finally, we combine our AV-transformer with our LSTM-based model from (Hori et al. 2019b), which also employed attentional MM fusion and JST learning. The LSTM-based model

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

Table 4: Evaluation results on DSTC10-AVSD test set.

Model	BLEU4	METEOR	ROUGE_L	CIDEr
Baseline AV-transformer	0.247	0.191	0.437	0.566
Extended AV-transformer	0.371	0.245	0.535	0.869
+ Comb. with LSTM	**0.385**	**0.247**	**0.539**	**0.888**

Table 5: Evaluation results on temporal reasoning for DSTC10-AVSD test set (Unofficial for challenge).

Model	IoU-1	IoU-2
Attention method	0.361	0.380
Region Proposal Net (RPN)	**0.521**	**0.550**

had a two-layer bidirectional LSTM encoder for question encoding, a single projection layer for each audio or visual feature, and a two-layer unidirectional LSTM decoder including attentional MM fusion. The number of LSTM cells was 256 for each layer of the encoder and the decoder. When we combine the word posterior probabilities of the Transformer and LSTM decoders in the log domain, we obtain the best results, which outperform the prior method (Hori et al. 2019b) and even achieve competitive performance to the best DSTC7 system that used the caption/summary information.

Table 3 shows the evaluation results on the DSTC8 test set. As in the DSTC7 results, the AV-transformer including all the extensions shows substantial improvements on all the performance metrics. Furthermore, the table shows that combination of the AV-transformer and the LSTM model achieves the state-of-the-art performance in BLEU4, METEOR, and ROUGE_L in comparison with the DSTC8 best system (Li et al. 2021) based on a large-scale Transformer initialized with GPT-2 (Radford et al. 2019), for the condition in which caption/summary information were not available.

Finally, we evaluated our model with the DSTC10-AVSD test set. The sentence generation performance is shown in Table 5, and we see improvements similar to the ones in the DSTC7 and DSTC8 results. We also evaluated the reasoning performance of the attention-based and RPN-based methods introduced in the section for Temporal Reasoning. The RPN had 3-layer Conv1D modules with 10 different kernel sizes for each modality and 256 dimensions in each internal layer. Table 5 shows the reasoning performance measured by Intersection over Union (IoU), which indicates the ratio of overlap between the predicted and ground-truth time regions (higher is better). Since there may be multiple valid reasons for each answer, we designed two IoU measures, where IoU-1 is obtained as an average IoU computed between each ground truth and the predicted region that gives the highest IoU to the ground truth. IoU-2 is computed by frame-level matching among all predicted and ground-truth regions for each answer, i.e., frames included in both predicted and ground-truth regions are counted as intersections while those included in both or either of them are counted as union. Table 5 shows that the RPN outperforms the naive attention-based approach, which suggests that model training with ground-truth annotations for temporal reasoning is important for temporal reasoning in the AVSD task. We did not get the above reasoning results in the official challenge timeline, and thus the result was not officially approved by the organizers.

Figures 2 and 3 show examples of temporal reasoning obtained by the baseline and proposed systems in comparison with the ground truth. We selected samples as shown below: (1) Reasoning for how the video ends, (2) Reasoning for actions, and (3) Reasoning for "where" question, (4) Reasoning for the entire video: Answers need to be generated based on the information in all frames. These examples clearly show that the RPN method in the proposed system provides much better reasoning than the attention method of the baseline system.

Conclusions

In this paper, we introduced the DSTC10-AVSD task and dataset, which promote further advancements into real-world applications of AVSD, in which human-created descriptions are not available at inference time and where temporal reasoning is required to provide evidence supporting the answers. We proposed extending the baseline system for DSTC10-AVSD with attentional multimodal fusion, joint student-teacher learning, and model combination techniques, achieving state-of-the-art performance resulting in an improvement across all evaluation metrics. Our experiments compared the performance of the baseline system and our extended system with the previous state of the art, testing on the AVSD test sets for DSTC7, DSTC8, and DSTC10.

References

Alamri, H.; Cartillier, V.; Das, A.; Wang, J.; Cherian, A.; Essa, I.; Batra, D.; Marks, T. K.; Hori, C.; Anderson, P.; Lee, S.; and Parikh, D. 2019. Audio Visual Scene-Aware Dialog. In *Proc. CVPR*.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*.

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. In *Proc. NIPS Deep Learning Symposium*.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, abs/1409.0473.

Carreira, J.; and Zisserman, A. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *Proc. CVPR*.

Chorowski, J. K.; Bahdanau, D.; Serdyuk, D.; Cho, K.; and Bengio, Y. 2015. Attention-Based Models for Speech Recognition. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28*, 577–585. Curran Associates, Inc.

Das, A.; Kottur, S.; Gupta, K.; Singh, A.; Yadav, D.; Moura, J. M. F.; Parikh, D.; and Batra, D. 2016. Visual Dialog. *CoRR*, abs/1611.08669.

(1) Reasoning for how the video ends

In [ ]:
```
view('CD6ZT')
```

```
Q1: how many people are in the video ?
A1: there appears to be just one there .
Q2: where does the video begin ?
A2: it begins in the room . maybe living
Q3: what does the man first do ?
A3: he picks up a box of food .
Q4: what happens next ?
A4: he laughs and runs out of hte room
Q5: so you can hear sound ?
A5: i cannot hear any sound
Q6: does he runs out very fast ?
A6: he does , but not too fast
Q7: does he look happy the entire time ?
A7: yea , he pretty much deos
Q8: does it end with him running out ?
---------------
(1) Groundtruth - A8: yes , that is exactly what it does
(2) Baseline    - A8: yes , he is in the same room
(3) Proposed    - A8: yes , the video ends with him walking out of the room .
```

(1) Groundtruth

(2) Baseline

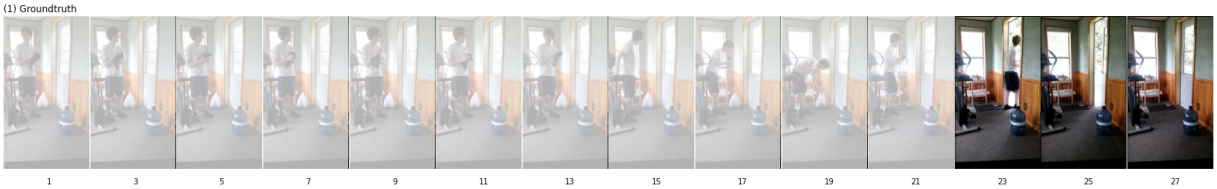
(3) Proposed

(2) Reasoning for actions

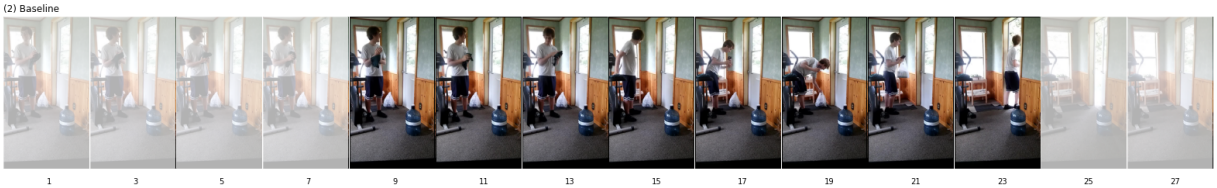
In [ ]:
```
view('EEOE7')
```

```
Q1: who is in the clip ?
A1: there is just one boy in the clip .
Q2: how old would you say that he is ?
A2: i think he might be around 17 years old .
Q3: what type of room is he in ?
A3: he looks to be in a weight room of some sort .
Q4: what is in his hand ?
A4: he is holding a towel in his hands .
Q5: what does he do first ?
A5: he wipes his hands then picks up a bag and phone .
Q6: does he say anything ?
A6: no he doesn 't say anything in the video .
Q7: what does he do with the bag and the phone ?
---------------
(1) Groundtruth - A7: he picks it up and walks out the door .
(2) Baseline    - A7: he just holds it and looks at it .
(3) Proposed    - A7: he picks it up and looks at it .
```

(1) Groundtruth



(2) Baseline



(3) Proposed

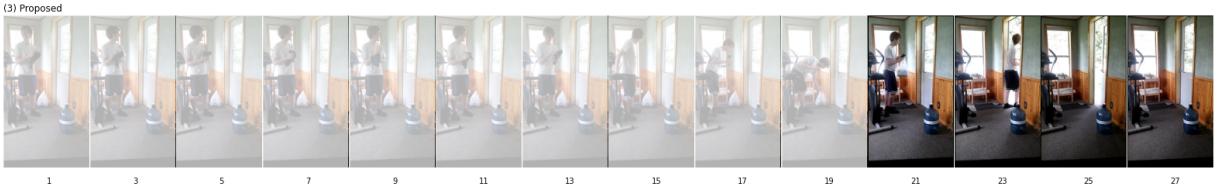


Figure 2: Example of reasoning results (1/2)

(3) Reasoning for "where" question

In [ ]:
```
view('27O1M')
```

```
Q1: how does the video start ?
A1: the video starts by the man dusting with a rug .
Q2: what is he dusting ?
A2: the sofa . then he grabs a box and puts items inside and places the box in the corner of the wall .
Q3: then ?
A3: then he takes the vacuum and starts vacuuming the floor and the video ends
Q4: anything else ?
A4: no that 's it the video ends there
Q5: so a man is dusting a sofa . he grabs a box , puts items inside and places it in a corner . then he starts to vaccum ?
A5: yes that 's it in the video
Q6: does he look happy ?
A6: not really . he has a neutral expression
Q7: what room is it ?
---------------
(1) Groundtruth - A7: looks like the living room
(2) Baseline    - A7: it looks like a living room
(3) Proposed    - A7: it looks like a living room .
```

(4) Reasoning for the entire video: Answers need to be generated based on the information in all frames.

In [ ]:
```
view('QT4ET')
```

```
Q1: is the girl in the frame when the video begins ?
A1: yes she is in the scene the entire time
Q2: does she pick up the towel ?
A2: she has it in her hands to begin with and then wraps it around her neck , it stays around her neck the entire time
Q3: what does she do after she puts the towel on ?
A3: she walks across the room and stands on a chair facing a wall
Q4: does she pick up anything in the room ?
A4: no she picks up nothing else , she does adjust the chair before she stands on it and does fidget with something on the wall she is facing
Q5: does she move the chair to the wall ?
A5: yes just a little bit , it was already facing the wall before she got to it
Q6: does she fidget with the something on the wall for long ?
A6: i would say about 10 seconds , it is above her head , it looks like she is adjusting perhaps
Q7: does she say anything during the video ?
---------------
(1) Groundtruth - A7: there is no dialogue of any kind throughout
(2) Baseline    - A7: no she does not say anything
(3) Proposed    - A7: no , she does not say anything .
```

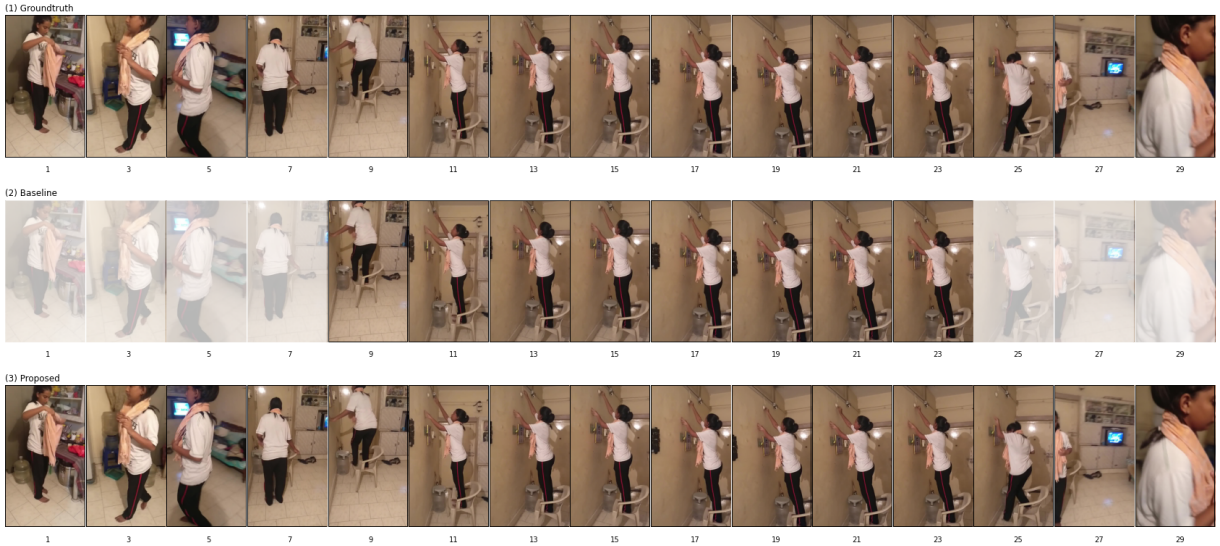


Figure 3: Example of reasoning results (2/2)

Das, A.; Kottur, S.; Moura, J. M.; Lee, S.; and Batra, D. 2017. Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning. In *International Conference on Computer Vision (ICCV)*.

D'Haro, L. F.; Yoshino, K.; Hori, C.; Marks, T. K.; Polymenakos, L.; Kummerfeld, J. K.; Galley, M.; and Gao, X. 2020. Overview of the seventh Dialog System Technology Challenge: DSTC7. *Computer Speech & Language*, 62: 101068.

Geng, S.; Gao, P.; Chatterjee, M.; Hori, C.; Le Roux, J.; Zhang, Y.; Li, H.; and Cherian, A. 2021. Dynamic graph representation learning for video dialog via multi-modal shuffled transformers. In *Proc. AAAI Conference on Artificial Intelligence*.

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

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

Hori, C.; Cherian, A.; Marks, T. K.; and Hori, T. 2019b. Joint Student-Teacher Learning for Audio-Visual Scene-Aware Dialog.

Hori, C.; Hori, T.; Lee, T.-Y.; Zhang, Z.; Harsham, B.; Hershey, J. R.; Marks, T. K.; and Sumi, K. 2017. Attention-Based Multimodal Fusion for Video Description. In *The IEEE International Conference on Computer Vision (ICCV)*.

Iashin, V.; and Rahtu, E. 2020. A Better Use of Audio-Visual Cues: Dense Video Captioning with Bi-modal Transformer. In *Proc. BMVC*.

Kim, S.; Galley, M.; Gunasekara, C.; Lee, S.; Atkinson, A.; Peng, B.; Schulz, H.; Gao, J.; Li, J.; Adada, M.; Huang, M.; Lastras, L.; Kummerfeld, J. K.; Lasecki, W. S.; Hori, C.; Cherian, A.; Marks, T. K.; Rastogi, A.; Zang, X.; Sunkara, S.; and Gupta, R. 2021. Overview of the Eighth Dialog System Technology Challenge: DSTC8. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29: 2529–2540.

Le, H.; Sahoo, D.; Chen, N.; and Hoi, S. 2019. Multimodal Transformer Networks for End-to-End Video-Grounded Dialogue Systems. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Li, Z.; Li, Z.; Zhang, J.; Feng, Y.; and Zhou, J. 2021. Bridging text and video: A universal multimodal transformer for video-audio scene-aware dialog. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28: 91–99.

Sanabria, R.; Palaskar, S.; and Metze, F. 2019. CMU Sinbad's Submission for the DSTC7 AVSD Challenge. In *DSTC7 at AAAI2019 workshop*.

Sigurdsson, G. A.; Varol, G.; Wang, X.; Laptev, I.; Farhadi, A.; and Gupta, A. 2016. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. *ArXiv*.

Tapaswi, M.; Zhu, Y.; Stiefelhagen, R.; Torralba, A.; Urtasun, R.; and Fidler, S. 2016. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4631–4640.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Proc. NeurIPS*, 5998–6008.

Zhang, P.; Goyal, Y.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2016. Yin and Yang: Balancing and Answering Binary Visual Questions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.