Hierarchical multimodal attention for end-to-end audio-visual scene-aware dialogue response generation

Hung LE
Singapore Management University, hungle.2018@phdis.smu.edu.sg

Doyen SAHOO
Singapore Management University, doyens@smu.edu.sg

Nancy F. CHEN

Steven C. H. HOI
Singapore Management University, chhoi@smu.edu.sg

Citation

LE, Hung; SAHOO, Doyen; CHEN, Nancy F.; and HOI, Steven C. H.. Hierarchical multimodal attention for end-to-end audio-visual scene-aware dialogue response generation. (2020). *Computer Speech and Language*. 63, 1-13.
Available at: https://ink.library.smu.edu.sg/sis_research/5259

Hierarchical multimodal attention for end-to-end audio-visual scene-aware dialogue response generation

Hung Le[ab], Doyen Sahoo[c], Nancy F. Chen[b], Steven C. H. Hoi[ac]

[a] Singapore Management University, 81 Victoria St, 188065, Singapore
[b] Institute for Infocomm Research, 1 Fusionopolis Way, 138632, Singapore
[c] Salesforce Research Asia, 5 Temasek Boulevard, Suntec Tower Five, 038985, Singapore

Published in Computer Speech & Language, 2020 September, 63, Article No. 101095, pp. 1-13

https://doi.org/10.1016/j.csl.2020.101095

Abstract

This work is extended from our participation in the Dialogue System Technology Challenge (DSTC7), where we participated in the Audio Visual Scene-aware Dialogue System (AVSD) track. The AVSD track evaluates how dialogue systems understand video scenes and responds to users about the video visual and audio content. We propose a hierarchical attention approach on user queries, video caption, audio and visual features that contribute to improved evaluation results. We also apply a nonlinear feature fusion approach to combine the visual and audio features for better knowledge representation. Our proposed model shows superior performance in terms of both objective evaluation and human rating as compared to the baselines. In this extended work, we also provide a more extensive review of the related work, conduct additional experiments with word-level and context-level pretrained embeddings, and investigate different qualitative aspects of the generated responses.

Keywords

Audio-visual scene-aware dialogue, Dialogue system, Multimodal attention, Neural network, Response generation

1. Introduction

The Dialogue System Technology Challenge (DSTC7) (D'Haro et al., 2020) proposed the Audio Visual Scene-aware Dialogue System (AVSD) track, which focuses on dialogue systems of multiple modalities. Arising from the related tasks in visual Question-Answering (VQA) (Antol, Agrawal, Lu, Mitchell, Batra, Lawrence Zitnick, Parikh, 2015, Goyal, Khot, Summers-Stay, Batra, Parikh, 2017), image captioning (Vinyals, Toshev, Bengio, Erhan, 2015, Xu, Ba, Kiros, Cho, Courville, Salakhudinov, Zemel, Bengio, 2015), video captioning (Hori, Hori, Lee, Zhang, Harsham, Hershey, Marks, Sumi, 2017, Li, Yao, Pan, Chao, Mei, 2018), and visual dialogues (Das, Kottur, Gupta, Singh, Yadav, Moura, Parikh, Batra, 2017, Das, Kottur, Moura, Lee, Batra, 2017), the AVSD track provides an interesting setting for dialogue research. In addition to processing information in traditional dialogue research such as dialogue context and user utterances, the dialogue agents in AVSD track are required to integrates not only visual features but also audio features from video input. Compared to visual dialogues (Das et al., 2017b), the proposed tasks in this track involve more complex features with larger feature space (i.e. temporal visual and audio features across multiple video frames).

Our approach for this track is delineated in this paper. Our two entries to this track are developed upon the baseline model (Hori et al., 2018), and exploit a hierarchical attention mechanism on question features, caption features, and visual and audio features of the input video. The attention strategies are adopted similarly to Anderson et al. (2018), including question-guided attention on caption and video features. In the VQA setting, using attention contributes to the increase of accuracy in selecting the correct answers. In the context of AVSD track, we aim to explore how this attention mechanism could be utilized in a dialogue setting in combination with visual and audio features of input videos to generate natural responses. We also integrate both visual and audio features using a nonlinear fusion technique to combine these features for better representations. Our proposed approach

approach shows superior performance as compared to the baseline model in terms of both automatic metrics and human evaluation. In this work, we also extended our submission to the DSTC7 in 3 areas:

- We include a review of related work in the research domains of dialogues and multiple modalities.
- We conduct additional experiments to examine the model performance with different word-level and context-level pretrained embeddings.
- We provide comprehensive qualitative analysis of the generated dialogue responses in several aspects: input video lengths, question types, and dialogue turn positions.

2. Related work

Dialogue research can be divided into 2 major categories: open-domain dialogues (Shang et al., 2015; Vinyals and Le, 2015; Yao et al., 2015; Li et al., 2016a; 2016b; Serban et al., 2017; 2016) which are modeled with holistic seq2seq models, and task-oriented dialogues (Bordes and Weston, 2016; Fatemi et al., 2016; Henderson et al., 2014; Liu, Lane) which have conventionally been modeled as a pipeline of components. More recent development includes exploring end-to-end task-orietned architetures such as memory network (Madotto, Wu, Fung, 2018) and an efficient two-stage CopyNet framework (Lei, Jin, Kan, Ren, He, Yin, 2018). Recent efforts on dialogue systems also include building conversational agents that can ground their responses on knowledge base such as online encyclopedias (Dinan et al., 2018), online chat platforms and e-commercial recommendation sites (Ghazvininejad et al., 2018). The dialogue agents can generate responses that are relevant to the dialogue context as well as exploiting information from the provided knowledge base e. g. recommend which restaurants are top-rated in a given area. Another extension in dialogue research is multi-turn QA, or Conversational QA (Zhu et al., 2018). The dialogue agents are typically not tasked to converse in trivial dialogues like open-domain dialogues, but restricted to a given text source such as a fictional story and Wikipedia page. The dialogue agents in this setting are expected to answer specific and complex questions from the users about the content of the text source.

There are a few recent efforts in the NLP domain where multimodal information needs to be incorporated. Major research directions include image captioning (Vinyals et al., 2015; Xu et al., 2015), video captioning (Hori et al., 2017; Li et al., 2018) and visual question-answering (VQA) (Antol et al., 2015; Goyal et al., 2017). The common challenge in multimodal settings is to obtain a model that can understand both natural language as well as non-text features such as vision and audio from images and videos. Specifically, in image captioning and video captioning, the dialogue agents are required to output description sentences about the content of an image or video respectively. Common approaches are to attend and align information from non-text features, e.g. pixels from images and temporal visual features from videos when decoding caption sentences. Visual QA is an extension from image captioning as more fine-grained understanding is required to extract the right information to answer questions from the users correctly. Recently, the proposed movie QA task (Tapaswi et al., 2016) has gained increasing attention. The task is similar to visual QA but the answers are grounded in movie videos. While VQA or movie QA tasks are related to our work, they are restricted to answering only individual queries. In AVSD, the dialogue agent is required to learn to process information of multiple modalities through multi-turn dialogues. We also focus on generating dialogue responses rather than selecting from a set of candidates. This requires the dialogue agents to model the semantics of the visual and/or audio contents to output appropriate responses.

Another related task is visual dialogues (Das et al., 2017a; 2017b; Kottur et al., 2018). This is similar to VQA but the conversational agent needs to process dialogue context potentially of multiple turns rather than just a single-turn question and output relevant responses. In this work, we focus on knowledge grounded in videos, which is more complex, considering the temporal visual and audio features extended across multiple video frames.

3. Approach

This section details several changes we made from the baseline approach (Hori et al., 2018). Given an input video V, its caption C_v, a dialogue context of $(t-1)$ turns, each including a pair of (question, answer) $(Q_1, A_1), \ldots, (Q_{t-1}, A_{t-1})$, and a factual query Q_t on the video content, the goal of AVSD task is to generate an appropriate dialogue response A_t that is relevant to the context and addresses the user query correctly. Our model follows the encoder-decoder framework, including 3 major components: (1) RNN Encoders that encode the dialogue context, the video caption, and the user query into fixed dimensional representations; (2) Hierarchical Attentions that include self-attention on question features, and question-guided attentions on video features (visual and audio) as well as on caption features. The final output of attentions are fused to create a joint feature representation of all input information from multiple modalities; and (3) RNN Decoder that uses the contextual joint features and generates system responses token-by-token. The overview of the model can be seen in Fig. 1.

We describe the individual components and changes as compared to the baseline (Hori et al., 2018) in the following.

3.1. Encoders

3.1.1. Gated recurrent unit

Instead of using Long short-term memory (LSTM) as the unit module for the recurrent network, we replace LSTM with Gated Recurrent Unit (GRU) in the encoders (for question and dialogue history). GRUs have shown to achieve superior performance at affordable computational cost (Cho et al., 2014). We describe here in mathematical details of the GRU for complete notation of

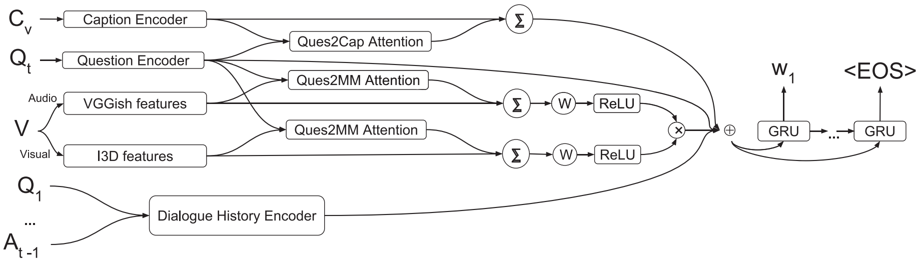


Fig. 1. Overview of the proposed end-to-end multimodal dialogue systems with attention mechanisms on multimodal features and video caption embedding. The model follows the encoder-decoder framework, including 3 major components: (1) RNN Encoders that encode the dialogue context, the video caption, and the user query into fixed dimensional representations; (2) Hierarchical Attentions that include self-attention on question features, and question-guided attentions on video features as well as on caption features; and (3) RNN Decoder that uses the contextual joint features and generates system responses token-by-token.

the proposed model. Given a sequence of input words S, in each encoding step n, the GRU will recurrently process the respective input s_n and the previous hidden state h_{n-1}. For simplicity, we denote s_n as both the real word as well as the representation vector of the word using an embedding matrix or one-hot representation. We denote the embedding dimension as V. The hidden state h_n for each encoding step n is given by:

$$r_n = \sigma(I_r s_n + H_r h_{n-1}), \tag{1}$$

$$u_n = \sigma(I_u s_n + H_u h_{n-1}), \tag{2}$$

$$h^{\cdot}_n = \tanh\Big(I s_n + H(r_n \cdot h_{n-1})\Big), \tag{3}$$

$$h_n = (1 - u_n) \cdot h_{n-1} + u_t \cdot h^{\cdot}_n \tag{4}$$

where σ is the logistic sigmoid, $\cdot$ represents the element-wise scalar product between vectors, $I, I_u, I_r \in \mathbb{R}^{d_h \times V}$ and $H, H_r, H_u \in \mathbb{R}^{d_h \times d_h}$. The I matrices encode the word s_n while the H matrices are used to retain or forget the information in h_{n-1}. Hence, r_n denotes the *reset gate*, u_n the *update gate*, $h^{\cdot}_n$ the *candidate update*, and h_n the *final update*.

The reset gate and update gate are computed in parallel. Provided the current word s_n, if it is learned to forget information of the previous sequence h_{n-1}, the elements of r_n will be closer to 0. The update gate u_n judges whether the current word contains relevant information that should be stored in h_n. In the final update, if the elements of u_n are close to 0, the network keeps the last recurrent state h_{n-1}. The gating behavior in GRU showed to provide robustness to noise in the source sequence.

At each dialogue turn t, for each question Q_t, the question encoder reads the words of the questions sequentially and updates its hidden state according to:

$$h^{qes}_{t,n} = GRU_{qEnc}(h^{qes}_{t,n-1}, s_{t,n}), n = 1, \dots, N^{qes}_t \tag{5}$$

To encode the dialogue history, each question and answer for each dialogue turn $1, \dots, t-1$ is encoded by a separate encoder.

$$h^{qa}_{t,n} = GRU_{qaEnc}(h^{qa}_{t,n-1}, s_{t,n}), n = 1, \dots, N^{qa}_t \tag{6}$$

A separate GRU takes as input the sequence of past question and answer representations $Q_1, A_1, \dots, Q_{t-1}, A_{t-1}$ and computes the sequence of dialog-turn recurrent states to summarize the dialogue up to turn t into h^{his}_t.

$$h^{his}_t = GRU_{hisEnc}(h^{his}_{t-1}, H^{qa}_t), t = 1, \dots, t-1 \tag{7}$$

Where H^{qa}_t is the hidden state in the last position of each question or answer as computed in Eq. (6). For all encoders, we initialize the hidden states to zero.

$$h^{qes}_{t,0} = 0 \tag{8}$$

$$h^{qa}_{t,0} = 0 \tag{9}$$

$$h^{his}_0 = 0 \tag{10}$$

3.1.2. Caption encoder

Instead of concatenating the video caption as the first turn in the dialogue history like in the baseline (Hori et al., 2018), we use a separate encoder to encode the video caption. For each dialog, a GRU encoder reads the words of the caption of the respective video input sequentially and updates its hidden states:

$$h^{cap}_{t,n} = GRU_{capEnc}(h^{cap}_{t,n-1}, s_{t,n}), n = 1, \ldots, N^{cap}_{t} \tag{11}$$

We also initialize the hidden state $h^{cap}_{t,0} = 0$.

3.2. Hierarchical attention

3.2.1. Question self-attention

We adde a self-attention mechanism in question encoder. Specifically, in each dialogue turn, the model attends over all positions in the question sequence, each represented by the question encoder hidden state $h^{qes}_{n}(n = 1, \ldots, N^{qes})$. The set of all question hidden states h^{qes} are passed through two convolutional layers with kernel size 1 and ReLU and softmax activation. The result scalar attention α^{qes}_{n} is associated with the position nth in the question.

$$\alpha^{qes} = softmax\Big(Conv\Big(ReLU\Big(Conv(h^{qes})\Big)\Big)\Big) \tag{12}$$

$$\widehat{h}^{qes} = \sum_{n=1}^{N^{qes}} \alpha^{qes}_{n} h^{qes}_{n} \tag{13}$$

The question hidden states are weighted by the softmax result and sum to obtain a single vector $\widehat{h}^{qes}$ representing the attended question features q.

3.2.2. Question-to-multimodal attention

We extend the baseline multimodal attention (Hori et al., 2018) by implementing a question-guided attention mechanism commonly used in many VQA models (Teney et al., 2017; Anderson et al., 2018). The attention mechanism is used to direct the model to specific input feature sequences in each modality k (input sequence $x_k = x_{k1}, \ldots, x_{kL}$ for $k = 1, \ldots, K$). The number of modalities is denoted by K and the number of feature sequences is L. First, both question features q and modality feature x_{kl} are passed through separate linear layers with ReLU activation to project them to the same dimensional space D_k. For each modality $k = 1, \ldots, K$ and $l = 1, ., L$:

$$\tilde{q}_k = ReLU(W_{kq}q + b_{kq}) \tag{14}$$

$$\tilde{x}_{kl} = ReLU(W_{kx}x_{kl} + b_{kx}) \tag{15}$$

where $W_{kq} \in \mathbb{R}^{D_k \times d_q}$, and $W_{kx} \in \mathbb{R}^{D_k \times d_k}$. The question features is then expanded to have the same sequential dimension L as the modality feature, resulting in $\tilde{q}^{\exp}_{k} \in \mathbb{R}^{L \times D_k}$ (the expansion is done by repeatedly stacking $\tilde{q}_k$ for L times). We then use Hadamard product to create a feature vector f_k to jointly combine question and modality features. The vector is then passed through two convolutional layers with kernel size 1 and ReLU and softmax activation to obtain a scalar attention weight α_{kl} associated with input sequence x_{kl}.

$$f_k = \tilde{x}_k \cdot \tilde{q}^{exp}_{k} \tag{16}$$

$$\alpha_k = softmax\Big(Conv\Big(ReLU\Big(Conv(f_k)\Big)\Big)\Big) \tag{17}$$

$$\widehat{x}_k = \sum_{l=1}^{L} \alpha_{kl} x_{kl} \tag{18}$$

The attention weights are normalized over all input sequence with the softmax function. The input features are then weighted by the normalized values and sum to obtain a single vector $\widehat{x}_k$ representing the attended features of the input video for a modality k.

After obtaining the attended modality features for all modalities, we combine these features by first passing each of them to a linear layer with weight normalization (Salimans and Kingma, 2016) followed by ReLU. All modalities are projected to the same dimensional space D. Then we use Hadamard product to combine the features from different modalities. Intuitively, we use Hadamard product to map all representations into a common feature space. The result is a single vector $\widehat{z}$ representing the joint modality features of the input video. We show in our experiments that using Hadamard product is better than simply concatenating all representations (as used in the baselines), resulting in superior performance and better quality system responses.

$$\tilde{z}_k = ReLU\Big(weightNorm(W_{kz}\widehat{x}_k + b_{kz})\Big) \tag{19}$$

$$\tilde{z} = \prod^{K} \tilde{z}_k \tag{20}$$

3.2.3. Question-to-caption attention

We also use a question-guided attention on the caption sequence. Here the attention attends to information from different positions in the caption, representing by hidden states obtained from the caption encoder ($h_1^{cap}, \ldots, h_{N^{cap}}^{cap}$). First, both question features q and caption hidden state h_n^{cap} are passed through separate linear layers with ReLU activation to project them to the same dimensional space D^{cap}. The question features is then expanded to have the same sequential dimension N^{cap} as the caption features $\tilde{q}_{cap}^{\exp} \in \mathbb{R}^{N^{cap} \times D^{cap}}$ and we then use Hadamard product to create a vector for question-caption features f_{cap}. The rest of the attention is similar to our Question-to-Multimodal Attention described above.

$$f_{cap} = \tilde{h}_n^{cap} \cdot \tilde{q}_{cap}^{exp} \tag{21}$$

$$\alpha^{cap} = softmax\Big(Conv\Big(ReLU\Big(Conv(f_{cap})\Big)\Big)\Big) \tag{22}$$

$$\widehat{h}^{cap} = \sum_{n=1}^{N^{cap}} \alpha_n^{cap} h_n^{cap} \tag{23}$$

3.3. Decoder

To generate each system response, each dialogue history H, question Q, and video V are paired with a sequence of output words to predict a target sequence T. A GRU decoder is used to define a distribution over output words. For each decoding step m:

$$h_m^{res} = GRU_{resDec}(h_{m-1}^{res}, [y_{m-1}, g]) \tag{24}$$

$$g = \widehat{h}^{qes} \oplus \tilde{z} \oplus h_T^{his} \oplus \widehat{h}^{cap} \tag{25}$$

where g is the concatenation of question encoding, audio-visual fusioned encoding, dialogue history encoding up to the last dialogue turn T, and caption encoding. The decoder sequentially predicts each token using softmax function:

$$p(T|H,Q,V) = \prod_{m=1}^{M} \frac{exp\Big(f(h_{m-1}^{res}, e_{y_m})\Big)}{\sum_{y'} exp\Big(f(h_{m-1}^{res}, e_{y'})\Big)} \tag{26}$$

where e_{y_m} is the output word embedding, h_{m-1}^{res} is the output hidden vector of the decoder at decoding step $m-1$, and f is the activation function between h_{m-1}^{res} and e_{y_m}.

The question, dialogue history, and video caption encoders and the response decoder use different GRUs with separate parameters to capture different semantic composition. We use a beam search technique with beam size 5 in the decoder.

4. Experiments

We use the standard objective function log-likelihood of the target sequence T given the dialogue history H, question Q, and video V, which at decoding time provides the statistical decision problem:

$$\widehat{T} = \arg\max_T \log p(T|H,Q,V) \tag{27}$$

For each encoder and decoder, we use an independent single forward GRU layer. The number of hidden units is set to 512 for all the encoders and decoder. We also separate the parameters of the word embedding for the question, dialogue history, caption encoders, and response decoder. We chose to initialize all word embeddings with 200-dimensional Glove embedding (Pennington et al., 2014) pretrained on Wikipedia and Gigaword.[1] The large size of the training dataset helps to bootstrap the embeddings to contain more meaningful semantic information in each word. We trained each model up to 15 epochs with a decaying learning rate schedule. The learning rate is initialized to 0.001. We used the ADAM optimizer (Kingma and Ba, 2014) to train the model. The batch size is set to 64 during training. For each training, we selected the best model with the lowest perplexity on the official validation dataset.

4.1. Data

Table 1 summarizes the data provided for the DSTC7 ASVD track. Each dialogue consists of 10 questions about a given video and corresponding 10 responses. Each dialogue was yielded by two Amazon Mechanical Turk (AMT) workers. One of the workers played the role of an answerer who already watched the entire video while the other did not. Each answerer had to answer the other worker's questions based on the previous dialogue history and the input video (including audio and visual features and/or video caption). For each dialogue in the test set, we generated a response corresponding to the position of the *UNDISCLOSED*

[1] https://nlp.stanford.edu/projects/glove/

Table 1
DSTC7 Video scene-aware dialogue dataset.

	Official training	Official validation	Official test	Prototype test
# of Dialogues	7659	1787	1710	733
# of Turns	76,590	17,870	6745	7330
# of Words	1,450,754	339,006	110,252	138,790

Table 2
Objective and subjective evaluation results on official test data. The highest value in each metric is highlighted in bold.

Model	Video	Text	BLEU-4	METEOR	ROUGE_L	CIDEr	Rating
Baseline	I3D_rgb_flow	Dialogue	0.305	0.217	0.481	0.733	N/A
Baseline	I3D_rgb_flow+VGGish	Dialogue	0.309	0.215	0.487	0.746	2.848
Ours	I3D_rgb+VGGish	Dialogue	**0.315**	0.239	0.509	0.848	-
Ours	None	Dialogue+Caption	0.310	**0.242**	**0.515**	**0.856**	**3.080**
Official Test	N/A	N/A	N/A	N/A	N/A	N/A	3.938

token i.e. 1710 responses in total. We used the official training dataset to train our system and the official validation dataset to validate and select the best models. We did not merge validation data to the official training data so that we can compare the results to the baselines (Hori et al., 2018). In addition to the official test data, we also reported experiments with the prototype test data. The prototype test data includes the full set of ground-truth responses and allows us to comprehensively test our models for ablation analysis. The official test data does not include ground-truth responses and all results on this data are received from the competition organizers. We also utilized pretrained audio and visual feature extractors. Particularly, we used the I3D_rgb and I3D_flow features from the “Mixed_5c” layer of the I3D network (Carreira and Zisserman, 2017) for visual features and Audio Set VGGish (Hershey et al., 2017) for audio features.

4.2. Results

We evaluated our submissions and the baselines using corpus-level metrics, including BLEU (Papineni et al., 2002), CIDEr (Vedantam et al., 2015), ROUGE-L (Lin, 2004), and METEOR (Banerjee and Lavie, 2005). Results for these metrics were provided by the DSTC organizers. We submitted two systems, representing the two settings: *Video+Text* and *Text Only*. For *Video+Text* setting, in addition to the dialogue data, we use the I3D_rgb features and VGGish features for visual and audio features. In this setting, we did not submit the system that also uses video caption data as we did not find significant improvement during validation. For *Text Only* setting, we used the dialogue data as well as the video caption to train our model. We did not use the video summary data. We compared these systems to the baseline (Hori et al., 2018).

The objective and subjective evaluation results are shown in Table 2. The ground truth responses from the official test data were also evaluated by human judges and the results were provided by the organizers. All of our submissions show improvement over the baselines in terms of BLEU, CIDEr, METEOR, and ROUGE-L. Among our systems' results, the *Video+Text* system performs better than the *Text Only* system in terms of BLEU scores, with an exception for BLEU-1 where *Text Only* system is slightly better than *Video+Text* system. The *Text Only* system outperforms the *Video+Text* system in terms of METEOR, ROUGE-L, and CIDEr. As ROUGE-L is a recall oriented metric designed for summarization and METEOR is a translation metric, they may not be ideal to evaluate the performance of dialogue response generation. This might explain the inconsistency between these metrics and BLEU scores when we compare *Video+Text* and *Text Only* system results. The difference between *Video+Text* and *Text Only* results is also not significant. As we expect the information conveyed from video visual and audio features is more than video caption alone, the performance of *Video+Text* could be further improved. For human evaluation, the results are consistent with objective scores in which our proposed *Text Only* model outperforms the baseline. However, there is still some difference in human rating (0.858 absolute score) between our generated responses and the official test responses.

4.3. Ablation analysis

Tables 3 and 4 show the results of our proposed models trained on the official training data and evaluated on the prototype test data. The evaluation metrics include BLEU, METEOR, ROUGE-L, and CIDEr. The evaluation codes were provided by the organizer and based on MS COCO caption generation task.[2] Here we analyze how changes in different components of the network architectures and input components (Video, Text) affect the model performance. *Model #1* is essentially the baseline model (Hori et al., 2017). As we change from LSTM to GRU (*Model #2*) in all encoders and decoder, we do not observe much changes in terms of evaluation metrics. However, as GRU is more computationally efficient, we apply GRU in all the models. As we adopt Question-to-Multimodal Attention (*Model #3*), the performance increases slightly across all the metrics except for BLEU1. When we

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

Table 3
We tested variants of our proposed approach with different combinations of input data (Video, Text) and model architectures.

Model#	Video	Text	RNN	cap-att	mm-att	mm-fusion	word-emb
1	I3D_rgb+VGGish	Dialogue	LSTM	-	Baseline	Baseline	No
2	I3D_rgb+VGGish	Dialogue	GRU	-	Baseline	Baseline	No
3	I3D_rgb+VGGish	Dialogue	GRU	-	QuesProj+Conv	Baseline	No
4	I3D_rgb+VGGish	Dialogue	GRU	-	QuesProj+Conv	FC+HdmProd	No
5	I3D_rgb_flow+VGGish	Dialogue	GRU	-	QuesProj+Conv	FC+HdmProd	No
6	I3D_rgb+VGGish	Dialog+Caption	GRU	-	QuesProj+Conv	FC+HdmProd	No
7	I3D_rgb+VGGish	Dialog+Caption	GRU	QuesProj+Conv	QuesProj+Conv	FC+HdmProd	No
8	I3D_rgb+VGGish	Dialog+Caption	GRU	QuesProj+Conv	CapProj+Conv	FC+HdmProd	No
9	I3D_rgb+VGGish	Dialogue	GRU	-	QuesProj+Conv	FC+HdmProd	Glove100
10	I3D_rgb+VGGish	Dialogue	GRU	-	QuesProj+Conv	FC+HdmProd	Glove200
11	I3D_rgb_flow+VGGish	Dialogue	GRU	-	QuesProj+Conv	FC+HdmProd	Glove200
12	I3D_rgb+VGGish	Dialogue+Caption	GRU	QuesProj+Conv	QuesProj+Conv	FC+HdmProd	Glove200
13	I3D_rgb+VGGish	Dialogue+Caption	GRU	QuesProj+Conv	QuesProj+Conv	FC+HdmProd	Glove300
14	-	Dialogue+Caption	GRU	QuesProj+Conv	-	-	No
15	-	Dialogue+Caption	GRU	QuesProj+Conv	-	-	Glove200
16	-	Dialogue+Caption	GRU	QuesProj+Conv	-	-	Glove300

Table 4
Automatic evaluation metrics on different variants of our proposed approach. The models are evaluated on the prototype test data. The best value in each metric is highlighted in bold.

Model#	BLEU				METEOR	ROUGE-L	CIDEr
	B-1	B-2	B-3	B-4			
1	0.272	0.176	0.120	0.086	0.122	0.298	0.842
2	0.266	0.174	0.120	0.086	0.121	0.299	0.843
3	0.269	0.175	0.121	0.087	0.123	0.301	0.851
4	0.291	0.186	0.126	0.090	0.127	0.301	0.824
5	0.284	0.183	0.125	0.090	0.123	0.296	0.802
6	0.294	0.192	0.130	0.094	0.124	0.300	0.865
7	0.304	0.198	0.137	0.100	0.131	0.312	0.891
8	0.298	0.194	0.135	0.097	0.129	0.309	0.867
9	0.276	0.181	0.125	0.091	0.124	0.304	0.870
10	0.307	**0.204**	**0.144**	**0.106**	**0.136**	**0.320**	**0.995**
11	0.303	0.201	0.141	0.103	0.133	0.317	0.962
12	**0.314**	**0.204**	0.142	0.102	**0.136**	0.317	0.940
13	0.307	0.202	0.142	0.103	0.134	0.316	0.935
14	0.293	0.194	0.136	0.100	0.130	0.313	**0.933**
15	**0.312**	**0.203**	**0.141**	**0.102**	**0.135**	**0.316**	0.931
16	0.304	0.197	0.139	0.101	0.133	0.314	0.930

combine Question-to-Multimodal Attention with Non-linear Multimodal Feature Fusioning (*Model #4*), the results increase significantly in terms of BLEU scores. However, as we add I3D_flow features of the input video (*Model #5*), the performance deteriorates. We speculate that our Multimodal Feature Fusioning method is not suitable to combine more than two types of features. Therefore, adding a third feature such as I3D_flow affects the results. When we add caption features and/or question-guided attention mechanism, the model performance clearly improves (*Model #6 and #7*). We also experiment with Caption-to-Multimodal Attention by replacing q in Eq. (14) to $\widehat{h}^{cap}$ (*Model #8*). However, the results are worse than using the proposed Question-to-Multimodal Attention. Without using pretrained word embedding, the best performance is achieved when using both video caption input and question-guided attention on the caption features.

When using pretrained Glove embedding, we observe increased results with 200-dimensional embedding (*Model #10*). With 100-dimensional Glove embedding (*Model #9*), the model is not as good as one without pretrained embedding (*Model #4*). This could be caused by the 100-dimensional embedding space not being able to capture comprehensive semantic meaning in the training corpus. Similarly to (*Model #5*), we do not see improvement when adding I3D_flow into the input video features (*Model #11*). Surprisingly, as we add caption features with question-guided attention (*Model #12*), the performance is not as good as the case without caption features (*Model #10*), except for BLEU1. Among the *Video+Text* setting models, *Model #10* shows the best performance and is used as our submission to the DSTC7 for the *Video+Text* setting. We also experiment with only input text without the input video (*Model #14, #15*, and *#16*). As we use pretrained 200-dimensional Glove embedding (*Model #15*), we achieve the best performance and use this model as our submission for the *Text Only* setting.

Table 5
Automatic evaluation metrics on the full models with finetuned or fixed pretrained embeddings. The models are evaluated on the prototype test data. For a pretrained embedding, the embedding dimension is indicated after the underscore. For FastText (Mikolov et al., 2018), the embedding weights are pretrained on Common Crawl (CC) (600B tokens) or Wikipedia, UMBC webbase corpus and statmt.org news dataset (Wiki) (16B tokens). The best value in each metric is highlighted in bold.

Pretrained Emb.	BLEU				METEOR	ROUGE-L	CIDEr
	B-1	B-2	B-3	B-4			
None	0.291	0.186	0.126	0.090	0.127	0.301	0.824
Word Embedding (Finetuned)							
Word2vec_300 (Mikolov et al., 2013)	0.284	0.188	0.133	0.098	0.128	0.315	0.956
Glove_50 (Pennington et al., 2014)	0.277	0.181	0.127	0.094	0.124	0.309	0.902
Glove_100 (Pennington et al., 2014)	0.276	0.181	0.125	0.091	0.124	0.304	0.870
Glove_200 (Pennington et al., 2014)	**0.307**	**0.204**	**0.144**	**0.106**	**0.136**	**0.320**	**0.995**
FT_300(CC) (Mikolov et al., 2018)	0.289	0.192	0.137	0.102	0.130	0.317	0.989
FT_300(Wiki) (Mikolov et al., 2018)	0.284	0.188	0.133	0.099	0.127	0.315	0.957
Contextual Embedding (Fixed)							
Elmo_1024 (Peters et al., 2018)	**0.286**	**0.190**	**0.136**	**0.101**	**0.130**	**0.317**	**0.986**
Elmo_2048 (Peters et al., 2018)	0.273	0.182	0.130	0.097	0.126	0.311	0.971
BERT_768 (Devlin et al., 2018)	0.273	0.180	0.127	0.094	0.123	0.305	0.912
BERT_1024 (Devlin et al., 2018)	0.278	0.182	0.129	0.095	0.124	0.306	0.910
GPT (Radford et al., 2018)	0.286	0.185	0.127	0.090	0.126	0.308	0.852
GPT2 (Radford et al., 2019)	0.279	0.182	0.127	0.093	0.123	0.301	0.870
Contextual Embedding (Finetuned)							
Elmo_1024 (Peters et al., 2018)	**0.284**	**0.189**	**0.135**	**0.100**	**0.129**	**0.316**	**0.989**
Elmo_2048 (Peters et al., 2018)	0.273	0.181	0.130	0.097	0.127	0.312	0.985
BERT_768 (Devlin et al., 2018)	0.278	0.185	0.131	0.097	0.126	0.310	0.961
BERT_1024 (Devlin et al., 2018)	0.277	0.182	0.129	0.095	0.125	0.308	0.933
GPT (Radford et al., 2018)	0.275	0.183	0.129	0.095	0.126	0.312	0.939
GPT2 (Radford et al., 2019)	0.280	0.186	0.132	0.098	0.128	0.312	0.971

4.4. *Impact of pretrained embedding*

In this section, we examine the impacts of pretrained embedding on model performance. The models are initialized with pretrained embedding to encode the sequences and the embedding weights are either finetuned or fixed during training time. We investigated the model performance when using only the video features (visual and audio) without the input video caption (Row 9 and 10 in Table 3 and 4). We consider the following pretrained word embedding models: Word2Vec[3] (Mikolov et al., 2013), Glove[4] (Pennington et al., 2014), and FastText[5] (Mikolov et al., 2018). For these pretrained embeddings, we initialize the embeddings with the pretrained weights and finetune the weights with dialogue data. We also experiment with contextual embedding models, including Elmo[6] (Peters et al., 2018), BERT (Devlin et al., 2018), GPT (Radford et al., 2018), and GPT2[7] (Radford et al., 2019). In each experiment, the contextual embedding is combined by using element sum to combine with the token-level embeddings. The token-level embeddings are learned from scratch in all experiments with pretrained contextual embedding. We experiment with either fixing the contextual embedding weights or finetune the weights with dialogue data.

As can be seen in Table 5, compared to not using any pretrained embedding, applying the pretrained word or contextual embeddings generally helps to improve the performance in most of the metrics, except for cases in BLEU1 and BLEU2. For word embedding, we observe the best performance is obtained when using 200-dimension Glove embedding (Pennington et al., 2014). When using the contextual embedding (either fixed or finetuned during training), the best performance is obtained when using 1024-dimensional Elmo embedding (Peters et al., 2018). Interestingly, we noted that the performance improvement with pretrained contextual embeddings is less than the cases with pretrained word embeddings. This could be explained by two reasons: (1) The dataset used for pretraining the embedding and the datasets used for learning dialogue models might have different distributions as the dialogue data is more of casual or spoken language while the pretrained data is typically composed of written language text. This difference could impact the contribution of the pretrained embedding in dialogue models and the impact is higher with context-based representations than token-based representations. (2) The contribution of contextual embeddings might also be limited by the multi-turn structure of dialogues as the embeddings are generally pretrained and tested on non-dialogue tasks such as language modeling, question-answering, machine translation, and text tagging (Peters et al., 2018; Devlin et al., 2018; Radford et al., 2018; 2019). In dialogues, each dialogue turn is usually shorter (of one-sentence length) than the input

[3] https://code.google.com/archive/p/word2vec/
[4] https://nlp.stanford.edu/projects/glove/
[5] https://fasttext.cc/docs/en/english-vectors.html
[6] https://github.com/allenai/allennlp/blob/master/tutorials/how_to/elmo.md
[7] https://github.com/huggingface/pytorch-pretrained-BERT

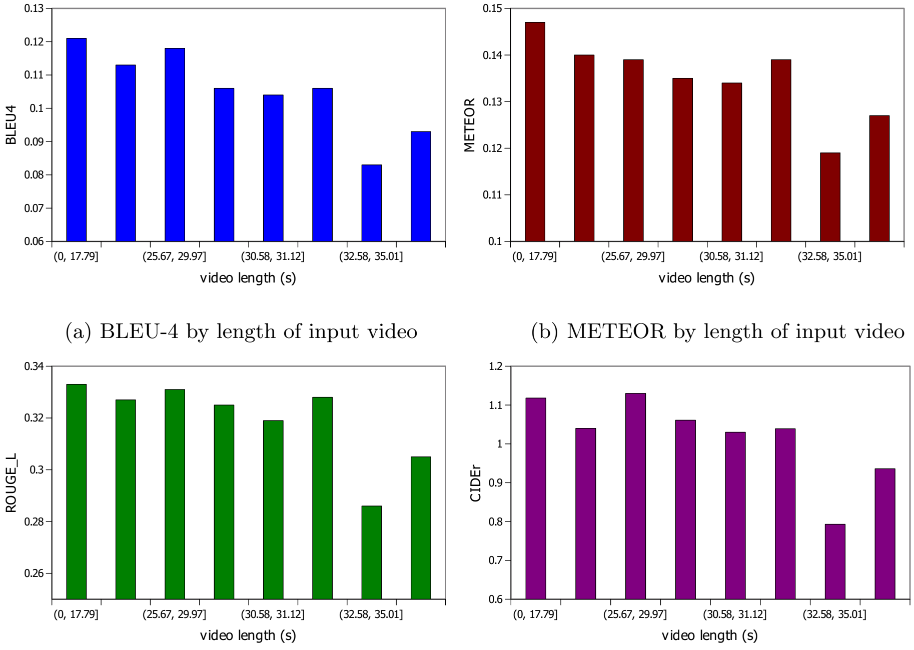
sequences in previous tasks. Dialogue turns are also structured as two-way or interactive communication rather than as continual paragraphs or documents in other tasks. Therefore, to apply pretrained context-level embedding into dialogues, other network approaches such as concatenating all dialogue turns into a single sequence (Wolf et al., 2019) might exploit more benefit of the pretrained embedding than the current hierarchical network structure.

5. Qualitative analysis

In this section, we analyze the model performance in several qualitative aspects. We examine our model with the highest BLEU-4 score in the official test result (See Table 2). All the results in this section are based on the generated responses in the prototype test set.

5.1. Video length

First we investigate whether the length of input video affects the model performance. As can be seen in Fig. 2 and Table 6, the performance generally decreases as the length of input video increases, except for slight surge of performance for video ranges (25.67, 29.97] and (31.12, 32.58]. There is a sharp decrease in performance at the video range (32.58, 35.01]. We speculate that the drop in performance could be caused by data bias as the testing population for this video range is quite small (See Table 6). The best performance is obtained at the shortest range of video lengths i.e. less than or equal to 17.79(s), except for CIDEr metric, which has the best value at video range (25.67, 29.97].



(a) BLEU-4 by length of input video

(b) METEOR by length of input video

(c) ROUGE-L by length of input video

(d) CIDEr by length of input video

Fig. 2. Plots of relationship between the length of input video and automatic metrics. The ranges of video length are selected based on intervals between 100/8 percentiles e.g. less than 12.5th percentile, from 12.5th to 25th precentile, and so on. (*a*, *b*] denotes a range of values more than *a* and less than or equal to *b*.

Table 6
Automatic metrics by video length. (*a*, *b*] denotes a range of values more than *a* and less than or equal to *b*. The best value in each metric is highlighted in bold.

Video Length (s)	# Dials	# Turns	BLEU4	METEOR	ROUGE-L	CIDEr
(0, 17.79]	95	950	**0.121**	**0.147**	**0.333**	1.118
(17.79, 25.67]	93	930	0.113	0.140	0.327	1.040
(25.67, 29.97]	100	1000	0.118	0.139	0.331	**1.130**
(29.97, 30.58]	100	1000	0.106	0.135	0.325	1.061
(30.58, 31.12]	91	910	0.104	0.134	0.319	1.030
(31.12, 32.58]	92	920	0.106	0.139	0.328	1.039
(32.58, 35.01]	76	760	0.083	0.119	0.286	0.793
(35.01, 72.04]	86	860	0.093	0.127	0.305	0.936

Table 7
Automatic metrics by question type in user query Q_t. The question type of each user query is identified by simple word matching (case insensitive). The best value in each metric is highlighted in bold.

Question Types	# Dials	# Turns	BLEU4	METEOR	ROUGE-L	CIDEr
which	9	9	0.000	0.119	0.303	0.709
what	622	1490	0.053	0.101	0.252	0.534
who	10	10	0.234	0.276	**0.602**	1.486
when	17	18	0.062	0.113	0.329	0.951
where	106	119	0.097	0.131	0.331	0.945
why	27	34	0.025	0.071	0.178	0.230
how	240	279	0.099	0.137	0.302	0.812
how many	212	225	**0.263**	**0.298**	0.539	**2.020**
yes/no	733	5146	0.116	0.142	0.332	1.074

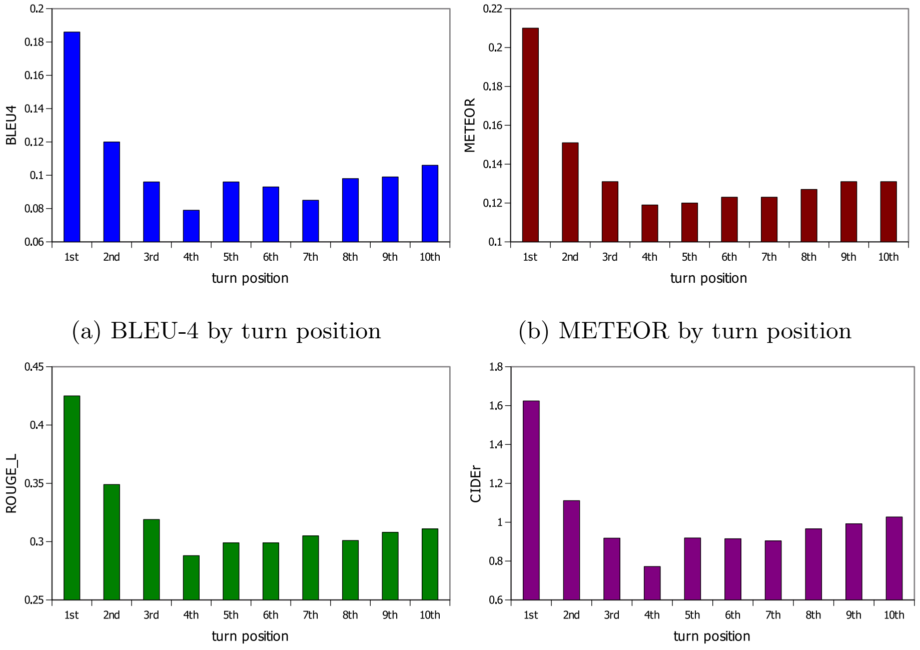
5.2. *Question type*

In Table 7, we examine the quality of the generated responses A_t by the types of questions in user query Q_t. The question type in each user query is identified by a simple word matching technique (case insensitive). The model performs better for *how many, who*, and *yes/no* types of questions in user query. In contrast, the model performance is lower for *which, what, why*, and *when* types of questions. However, We also note that the distribution of question types in the test data is quite imbalanced. For example, *yes/no* questions account for more than 70% of the test samples while *how many* questions only represent about 3%. The model performance is less representative in certain minor question types such as *which, who, when*, and *why*. While the model performs relatively well overall, it fails at some question types that require more complex reasoning and understanding of the input videos, such as *how* and *why* questions.

We also observe that the generated responses have a large proportion of negative answers i.e. answers that response "no" to *yes/no* questions. This might be due to the high frequency of negative responses in the training corpus. We also noticed our models tend to generate a universal answer such as "yes that is all happening in the video" to questions such as "is that all happened in the video?". This type of question might require further cross-references to reason over the dialogue history.

5.3. *Turn position*

Lastly, we examine whether the turn position of generated responses affects the performance in Fig. 3 and Table 8. As expected, we observe that the model achieves the best performance at the 1st turn and then reduces at later positions in the



(a) BLEU-4 by turn position

(b) METEOR by turn position

(c) ROUGE-L by turn position

(d) CIDEr by turn position

Fig. 3. plots of relationship between the turn position and automatic metrics.

Table 8
Automatic metrics by turn position of generated responses. The best value in each metric is highlighted in bold.

Turn Position	BLEU4	METEOR	ROUGE-L	CIDEr
1st	**0.186**	**0.210**	**0.425**	**1.624**
2nd	0.120	0.151	0.349	1.111
3rd	0.096	0.131	0.319	0.918
4th	0.079	0.119	0.288	0.772
5th	0.096	0.120	0.299	0.919
6th	0.093	0.123	0.299	0.915
7th	0.085	0.123	0.305	0.904
8th	0.098	0.127	0.301	0.966
9th	0.099	0.131	0.308	0.992
10th	0.106	0.131	0.311	1.027

dialogue. However, the trend of performance is not consistent throughout turn positions. For example, there are increases of performance observed at 5th, 6th, 9th, and 10th turn positions. We speculate that in addition to the turn position in the dialogue, other factors also affect the quality of the responses such as input video length and question type in user query. These factors might have more significant impacts on the quality of the responses than the turn positions and hence, result in an inconsistent trend of performance across turns. Still, considering the performance in the 1st turn as the upper bound, we aim to improve the models by reducing the gaps to this bound in 2nd and later turn positions.

5.4. Sample dialogue

We investigate one sample dialogue and compare the responses between our model and the baseline model as well as the reference responses in Table 9 (sample picture frames of the input video can be seen in Fig. 4). In terms of correctness, our responses express the answers better than the baseline responses in A_1, A_3, and A_5. This indicates that our model can capture the video features, either visual feature (e.g. action “drinking”, number of people) or audio feature (action “talking”) better than the baseline approach. For Q_2, both the baseline model and our model can express the correct answers. However, the baseline answer is preferred as its response is more concise than our response. For Q_4, both baseline response and our response do not match the reference response. However, our response is better in this context as “standing in front of the refrigerator” is more appropriate than “eating a sandwich” action for this particular dialogue context.

Table 9
A sample dialogue for an input video with responses generated by the best baseline model $A^{baseline}$, our best *Video+Text* model A^{ours}, and the reference response A^{ref} from the prototype test dataset.

Video Caption	“A person is working on their laptop, another person come in carrying a bag of groceries. they set the groceries down and start cooking.”
Q_1	how many people in the video?
A_1^{ref}	there are two person in video
$A_1^{baseline}$	there are one person in the video
A_1^{ours}	there are two people in the video
Q_2	what gender are they?
A_2^{ref}	there are both males in video
$A_2^{baseline}$	they are both men in the video
A_2^{ours}	one is a male and the other is a male
Q_3	are they talking in the video?
A_3^{ref}	no, both are not talks to each other
$A_3^{baseline}$	yes they are talking in the video
A_3^{ours}	no one is talking in the video
Q_4	what is the second person doing?
A_4^{ref}	second person bring a vegetable bag come to kitchen room
$A_4^{baseline}$	he is eating a sandwich
A_4^{ours}	the person is standing in front of the refrigerator
Q_5	are they drinking something in the video?
A_5^{ref}	no, they are both not a drink
$A_5^{baseline}$	no, he is eating a sandwich
A_5^{ours}	no, they are not drinking anything in the video

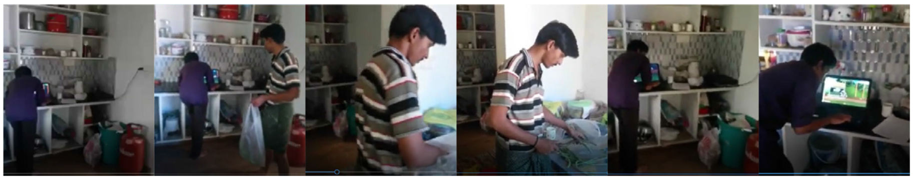


Fig. 4. Sample picture frames from an input video used in the AVSD validation set.

6. Discussion

DSTC7 AVSD Track proposes an interesting and challenging problem to investigate multimodal dialogue systems in a video-oriented rather than a visually-grounded setting (Das et al., 2017b; 2017a). It presents a framework to explore how the state-of-the-art feature extraction models such as VGGish and I3D can be used to extract the visual and audio features and be combined into a dialogue setting. We found that techniques used in visual QA models (Anderson et al., 2018; Teney et al., 2017) could be adapted into this setting to improve the model performance. We hope to explore in this multimodal dialogue setting further in the future with larger-scale datasets and in other variations of dialogue systems e.g. open-domain dialogues and task-oriented dialogues. Besides bootstrapping with pretrained word embeddings, we could also pretrain parts of the model on a larger dialogue corpus that covers similar topics and types of questions. An example corpus is the Movie QA dataset (Tapaswi et al., 2016) constructed to query about movie contents.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The first author is supported by A*STAR Computing and Information Science scholarship (formerly A*STAR Graduate scholarship). The third author is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046).

References

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L., 2018. Bottom-up and top-down attention for image captioning and visual question answering. CVPR, vol. 3, p. 6.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D., 2015. VQA: visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2425–2433.

Banerjee, S., Lavie, A., 2005. Meteor: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pp. 65–72.

Bordes, A., Weston, J., 2016. Learning end-to-end goal-oriented dialog. CoRR. abs/1605.07683.

Carreira, J., Zisserman, A., 2017. Quo vadis, action recognition? A new model and the kinetics dataset. Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. IEEE, pp. 4724–4733.

Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp. 1724–1734.

Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J.M., Parikh, D., Batra, D., 2017. Visual dialog. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, .

Das, A., Kottur, S., Moura, J.M.F., Lee, S., Batra, D., 2017. Learning cooperative visual dialog agents with deep reinforcement learning. 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2970–2979.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. BERT: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.

Dinan, E., Roller, S., Shuster, K., Fan, A., Auli, M., Weston, J., 2018. Wizard of wikipedia: knowledge-powered conversational agents. arXiv:1811.01241.

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

Fatemi, M., Asri, L.E., Schulz, H., He, J., Suleman, K., 2016. Policy networks with two-stage training for dialogue systems. In: Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue. Association for Computational Linguistics. https://doi.org/10.18653/v1/w16-3613.

Ghazvininejad, M., Brockett, C., Chang, M.-W., Dolan, B., Gao, J., Yih, S.W.-t., Galley, M., 2018. A knowledge-grounded neural conversation model. .

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D., 2017. Making the V in VQA matter: elevating the role of image understanding in visual question answering. CVPR, vol. 1, p. 3.

Henderson, M., Thomson, B., Young, S., 2014. Word-based dialog state tracking with recurrent neural networks. In: Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL). Association for Computational Linguistics. https://doi.org/10.3115/v1/w14-4340.

Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B., et al., 2017. CNN architectures for large-scale audio classification. Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. IEEE, pp. 131–135.

Hori, C., Alamri, H., Wang, J., Winchern, G., Hori, T., Cherian, A., Marks, T. K., Cartillier, V., Lopes, R. G., Das, A., et al., 2018. End-to-end audio visual scene-aware dialog using multimodal attention-based video features. arXiv:1806.08409.

Hori, C., Hori, T., Lee, T.-Y., Zhang, Z., Harsham, B., Hershey, J.R., Marks, T.K., Sumi, K., 2017. Attention-based multimodal fusion for video description. Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, pp. 4203–4212.

Kingma, D. P., Ba, J., 2014. Adam: a method for stochastic optimization. arXiv:1412.6980.

Kottur, S., Moura, J.M., Parikh, D., Batra, D., Rohrbach, M., 2018. Visual coreference resolution in visual dialog using neural module networks. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 153–169.

Lei, W., Jin, X., Kan, M.-Y., Ren, Z., He, X., Yin, D., 2018. Sequicity: simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 1437–1447.

Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B., 2016. A diversity-promoting objective function for neural conversation models. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics. https://doi.org/10.18653/v1/n16-1014.

Li, J., Galley, M., Brockett, C., Spithourakis, G., Gao, J., Dolan, B., 2016. A persona-based neural conversation model. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics. https://doi.org/10.18653/v1/p16-1094.

Li, Y., Yao, T., Pan, Y., Chao, H., Mei, T., 2018. Jointly localizing and describing events for dense video captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7492–7500.

Lin, C.-Y., 2004. ROUGE: a package for automatic evaluation of summaries. Text Summarization Branches Out.

Liu, B., Lane, I., 2017. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. arXiv:1708.05956.

Madotto, A., Wu, C.-S., Fung, P., 2018. Mem2Seq: effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, pp. 1468–1478.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv:1301.3781.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A., 2018. Advances in pre-training distributed word representations. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018).

Papineni, K., Roukos, S., Ward, T., Zhu, W.-J., 2002. BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, pp. 311–318.

Pennington, J., Socher, R., Manning, C.D., 2014. Glove: global vectors for word representation. Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543.

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations. In: Proc. of NAACL.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., 2018. Improving language understanding by generative pre-training. URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/languageunsupervised/language understanding paper.pdf.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., 2019. Language models are unsupervised multitask learners. OpenAI Blog 1 (8).

Salimans, T., Kingma, D.P., 2016. Weight normalization: a simple reparameterization to accelerate training of deep neural networks. Advances in Neural Information Processing Systems, pp. 901–909.

Serban, I., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A., Bengio, Y., 2017. A hierarchical latent variable encoder-decoder model for generating dialogues.

Serban, I.V., Sordoni, A., Bengio, Y., Courville, A., Pineau, J., 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. AAAI Press, pp. 3776–3783.

Shang, L., Lu, Z., Li, H., 2015. Neural responding machine for short-text conversation. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics. https://doi.org/10.3115/v1/p15-1152.

Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., Fidler, S., 2016. MovieQA: understanding stories in movies through question-answering. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Teney, D., Anderson, P., He, X., van den Hengel, A., 2017. Tips and tricks for visual question answering: learnings from the 2017 challenge. arXiv:1708.02711.

Vedantam, R., Lawrence Zitnick, C., Parikh, D., 2015. Cider: consensus-based image description evaluation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4566–4575.

Vinyals, O., Le, Q. V., 2015. A neural conversational model. CoRRabs/1506.05869.

Vinyals, O., Toshev, A., Bengio, S., Erhan, D., 2015. Show and tell: a neural image caption generator. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3156–3164.

Wolf, T., Sanh, V., Chaumond, J., Delangue, C., 2019. TransferTransfo: a transfer learning approach for neural network based conversational agents. CoRRabs/1901.08149.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y., 2015. Show, attend and tell: neural image caption generation with visual attention. International Conference on Machine Learning, pp. 2048–2057.

Yao, K., Zweig, G., Peng, B., 2015. Attention with intention for a neural network conversation model. CoRRabs/1510.08565.

Zhu, C., Zeng, M., Huang, X., 2018. SDNet: contextualized attention-based deep network for conversational question answering. arXiv:1812.03593.