End-to-End Multimodal Dialog Systems with Hierarchical Multimodal Attention on Video Features

Hung Le\textsuperscript{1}, Steven C.H. Hoi\textsuperscript{1}, Doyen Sahoo\textsuperscript{1}, Nancy F. Chen\textsuperscript{2}
\textsuperscript{1}Singapore Management University, Singapore
\textsuperscript{2}Institute for Infocomm Research (I2R), Singapore
hungle.2018@phdis.smu.edu.sg, {chhoi,doyens}@smu.edu.sg, nfychen@i2r.a-star.edu.sg

Abstract

We present our work on the Dialog System Technology Challenges 7 (DSTC7). We participated in Track 3, which evaluated how dialog systems understand video scenes and respond to users about the video visual and audio content. Our system is built upon the baseline system [Hori et al. 2018] with changes adopted similarly to [Anderson et al. 2018]. The model utilizes different types of attentions on video caption and the video audio and visual input that contribute to the improved evaluation results. We also applied a nonlinear feature fusioning of the visual and audio features to improve the results further. Our proposed model showed improved performance in terms of both objective evaluation and human rating, surpassing the performance of the baseline.

Introduction

The Dialog System Technology Challenge 7 (DSTC7) proposed a track that focused on multi-modal dialog systems. Arising from the related tasks in visual Question-Answering (VQA) [Antol et al. 2015][Goyal et al. 2017], image captioning [Vinyals et al. 2015][Xu et al. 2015], video captioning [Hori et al. 2017][Li et al. 2018], and visual dialogs [Das et al. 2017a][Das et al. 2017b], the track offered an interesting dialog setting that integrates not only visual features but also audio features from video input. Compared to visual dialog [Das et al. 2017b], the proposed task in this track consists of more modalities with much larger feature space. Our entries to Track 3 of DSTC7 follow this setting and was trained exclusively from the provided official data and did not utilize any external data.

Our approach for this track is summarized in this paper. Our two entries to this track were built upon the baseline model [Hori et al. 2018], and exploited different attention mechanisms 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 techniques on caption and video features. In the VQA setting, the usage of these attentions was shown to improve the accuracy in selecting the correct answers. In the context of DSTC7, we aims to explore how these attention mechanisms could be utilized in a dialog context to generate system responses rather than in a QA setting. The provision of both visual and audio features also allowed us to explore fusion techniques to combine these features better than the baseline. Our experiments showed that using linear layer with ReLU activation and Hadamard-product helped to fuse the features and increased the results significantly.

In this report we detailed our proposed model and different parameter settings of the model. We also provided some qualitative study to analyze how our system improved the quality of the responses from the baseline model.

End-to-End Multimodal Dialog System

This section details several changes we made from the baseline approach [Hori et al. 2018]. The overview of the model can be seen in Figure 1.

Gated Recurrent Unit

Instead of using Long short-term memory (LSTM) as the unit module for the recurrent network, we replaced LSTM with Gated Recurrent Unit (GRU) in the encoders (for question and dialog 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 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:

\begin{align}
    r_n &= \sigma(I_r s_n + H_r h_{n-1}), \\
    u_n &= \sigma(I_u s_n + H_u h_{n-1}), \\
    \overline{h}_n &= \tanh(I s_n + H (r_n \cdot h_{n-1})), \\
    h_n &= (1 - u_n) \cdot h_{n-1} + u_n \cdot \overline{h}_n
\end{align}

where $\sigma$ is the logistic sigmoid, $\cdot$ represents the element-wise scalar product between vectors, $I, I_u, I_r \in \mathbb{R}^{d_s \times V}$ 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, $\overline{h}_n$ 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 dialog turn $t$, for each question $Q_t$, the question encoder reads the words of the question sequentially and updates its hidden state according to:

$$h_{t,n}^{qes} = GRU_{qEnc}(h_{t,n-1}^{qes}, s_{t,n}), n = 1, \ldots, N_t^{qes} \tag{5}$$

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

$$h_{t,n}^{qa} = GRU_{qaEnc}(h_{t,n-1}^{qa}, s_{t,n}), n = 1, \ldots, N_t^{qa} \tag{6}$$

A separate GRU takes as input the sequence of past question and answer representations $Q_1, A_1, \ldots, Q_{t-1}, A_{t-1}$ and computes the sequence of dialog-turn recurrent states to summarize the dialog up to turn $t$ into $h_{t}^{his}$. For all encoders, we initialized the hidden states to zero.

$$h_{t,0}^{qes} = 0 \tag{7}$$

$$h_{t,0}^{qa} = 0 \tag{8}$$

$$h_{0}^{his} = 0 \tag{9}$$

**Caption Encoder**

Instead of concatenating the video caption as the first turn in the dialog history like in the baseline [Hori et al. 2018], we decided to 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:

$$h_{t,n}^{cap} = GRU_{capEnc}(h_{t,n-1}^{cap}, s_{t,n}), n = 1, \ldots, N_t^{cap} \tag{10}$$

We also initialized the hidden state $h_{t,0}^{cap} = 0$.

**Question Self-Attention**

We added 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 $h_{n}^{qes}(n = 1, \ldots, N_t^{qes})$. The set of all question hidden states $h_{n}^{qes}$ are passed through two convolutional layers with kernel size 1 and ReLU and softmax activation. The result scalar attention $\alpha_n^{qes}$ is associated with the position $n^{th}$ in the question.

$$\alpha_n^{qes} = softmax(Conv(ReLU(Conv(h_{n}^{qes}(n = 1, \ldots, N_t^{qes})))) \tag{11}$$

$$\hat{h}_{n}^{qes} = \sum_{n=1}^{N_t^{qes}} \alpha_n^{qes} h_{n}^{qes} \tag{12}$$

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

**Question-to-Multimodal Attention**

We extended 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 $l = 1, \ldots, L$:

$$\hat{q}_k = ReLU(W_q q + b_q) \tag{13}$$

$$\hat{x}_{kl} = ReLU(W_{z_k} x_{kl} + b_{z_k}) \tag{14}$$

where $W_{q_k} \in \mathbb{R}^{D_k \times d_q}$ and $W_{z_k} \in \mathbb{R}^{D_k \times d_z}$. The question features are then expanded to have the same sequential dimension $L$ as the modality feature $\hat{q}_{k}^{exp} \in \mathbb{R}^{L \times D_k}$ and 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 $\alpha_{kl}$ associated with input sequence $x_{kl}$.

$$f_k = \hat{x}_k \cdot \hat{q}_{k}^{exp} \tag{15}$$

$$\alpha_k = softmax(Conv(ReLU(Conv(f_k)))) \tag{16}$$

$$\hat{x}_k = \sum_{l=1}^{L} \alpha_{kl} x_{kl} \tag{17}$$

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 $\hat{x}_k$ representing the attended features of the input video for a modality $k$.

After obtaining the attended modality features for all modalities, we combined 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. The result is a single vector $\hat{z}$ representing the combined modality features of the input video.

$$\hat{z}_k = ReLU(weightNorm(W_{z_k} \hat{x}_k + b_{z_k})) \tag{18}$$

$$\hat{z} = \prod_{k=1}^{K} \hat{z}_k \tag{19}$$

**Question-to-Caption Attention**

We also used 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 are then expanded to have the same sequential dimension $N_{cap}$ as the caption features $q_{cap}^{exp} \in \mathbb{R}^{N_{cap} \times D_{cap}}$.
\(\mathbb{R}^{N_{\text{cap}} \times D_{\text{cap}}}\) and we then used Hadamard product to create a vector for question-caption features \(f_{\text{cap}}\). The rest of the attention is similar to our Question-to-Multimodal Attention described above.

\[
f_{\text{cap}} = \hat{h}^m_{\text{cap}} \cdot \hat{q}_{\text{exp}}^m
\]

(20)

\[
ao_{\text{cap}} = \text{softmax}(\text{ReLU}(\text{Conv}(f_{\text{cap}})))
\]

(21)

\[
\hat{h}^m_{\text{cap}} = \sum_{n=1}^{N_{\text{cap}}} \alpha_{n_{\text{cap}}}^m h_{n_{\text{cap}}}
\]

(22)

**Response Decoder**

To generate each system response, each dialog 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 \(m\):

\[
h^m_{\text{res}} = \text{GRU}_{\text{resDec}}(h^m_{\text{res}-1}, [y_{m-1}, g])
\]

(23)

\[
g = \hat{h}^q_{\text{res}} \odot \hat{z} + \hat{h}^h_{\text{res}} \odot f_{\text{cap}}
\]

(24)

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

\[
p(T | H, Q, V) = \frac{\exp(f(h^m_{\text{res}} - 1, e_{ym}))}{\sum_y \exp(f(h^m_{\text{res}} - 1, e_{ym}))}
\]

(25)

where \(e_{ym}\) is the output word embedding, \(h^m_{\text{res}} - 1\) is the output hidden vector of the decoder at decoding step \(m - 1\), and \(f\) is the activation function between \(h^m_{\text{res}} - 1\) and \(e_{ym}\).

Question, dialog history, and video caption encoders and the response decoder use different GRUs with separate parameters to capture different patterns of word composition. Similarly to \(\text{Hori et al. 2018}\), we use a beam search technique with beam size 5.

**Experiments**

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

\[
\hat{T} = \arg\max_T \{\log p(T | H, Q, V)\}
\]

(26)

For each encoder and decoder, we used 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 question, dialog history, caption encoders and response decoder. We chose to initialize all word embeddings with 200-dimensional Glove embedding \(\text{Pennington, Socher, and Manning 2014}\) pre-trained on Wikipedia and Gigaword\(\text{[1]}\). 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 \(\text{Kingma and Ba 2014}\) 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.

**Data**

The main objective for Track 3 of DSTC7 is training an end-to-end multimodal dialog system based on Charades videos \(\text{Sigurdsson et al. 2016}\). We downloaded the data from the official links provided by the organizers. Table \(\text{1}\) summarizes the data provided for this track. Each dialog consists of 10 questions about a given video and corresponding 10 responses. Each dialog 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 dialog history and the input video (including audio and visual features and/or video caption). For each dialog of the official test set, we generated a response corresponding to the position of the \(\text{UNDISCLOSED}\) token i.e. 1710 responses in total for each of our submissions. 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 \(\text{Hori et al. 2018}\). We also utilized the audio and visual feature extractors provided by the organizers. Particularly, we used the I3D\_rgb and I3D\_flow features from the “Mixed_5c” layer of the I3D network \(\text{Carreira and Zisserman 2017}\) for visual features and Audio Set VGGish \(\text{Hershey et al. 2017}\) for audio features.

<table>
<thead>
<tr>
<th>Official Training</th>
<th>Official Validation</th>
<th>Official Test</th>
<th>Prototype Test</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Dialogs</td>
<td>7,659</td>
<td>1,787</td>
<td>1,710</td>
</tr>
<tr>
<td># of Turns</td>
<td>153,180</td>
<td>35,740</td>
<td>13,490</td>
</tr>
<tr>
<td># of Words</td>
<td>1,450,754</td>
<td>339,006</td>
<td>110,252</td>
</tr>
</tbody>
</table>

**Official Results**

We evaluated our submissions and the baselines using corpus-level BLEU1 to BLUE4 (\text{Papineni et al. 2002}), CIDEr (\text{Vedantam, Lawrence Zitnick, and Parikh 2015}), ROUGE-L \(\text{Lin 2004}\), and METEOR \(\text{Banerjee and Lavie 2005}\). Results for these metrics were provided by the DSTC organizers themselves. We submitted two systems to Track 3, representing the two settings: Video+Text and Text Only. For Video+Text setting, in addition to the dialog 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 testing with the prototype test data. For Text Only setting, we used the dialog 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 \(\text{Hori et al. 2018}\).
The objective and subjective evaluation results are shown in Table 2 and 3 respectively. The ground truth responses from the official test data was 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 completely suitable to evaluate generated dialog responses. 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 system could have been 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 a significant gap of human rating (0.858 difference) between our generated responses and the official test responses.

**Prototype Results**

Table 4 shows the results of our proposed models trained on the official training data and evaluated on the prototype test data. The evaluation metrics are the same as the official results, including BLEU1-4, METEOR, ROUGE-L, and CIDEr. The evaluation codes were provided by the organizer and based on MS COCO caption generation [2]. Here we analyzed how changes in different modules affect the model performance. Model #1 is essentially our prototype results running with the baseline model [Hori et al. 2017]. As we changed from LSTM to GRU (Model #2) in all encoders and decoder, we did not observed much changes in terms of evaluation metrics. However, as GRU is more computationally efficient, we applied GRU in the remaining experiments. As we applied Question-to-Multimodal Attention (Model #3), the performance increased slightly across all the metrics except for BLEU1. When we combined Question-to-Multimodal Attention with Non-linear Multimodal Feature Fusioning as described above (Model #4), the results increased significantly in terms of BLEU scores. However, as we added I3D_flow features of the input video (Model #5), the performance became worse. We speculate that our Multimodal Feature Fusioning method is not suitable to combined more than two modalities, and hence, adding a third feature such as I3D_flow affected the results. When we added caption features with question-guided attention mechanism, the model performance clearly improved (Model #6 and #7). We also experimented with Caption-to-Multimodal Attention by replace q in Equation 13 to h_{cap} (Model #7). However, the results were worse than using the proposed Question-to-Multimodal Attention.

When using pretrained Glove embedding, we observed improved results with 200-dimensional embedding (Model #9). With 100-dimensional Glove embedding (Model #8), 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 enough useful semantic meaning in the training corpus. Similarly to (Model #5), we did not see improvement when adding I3D_flow into the input video features with pretrained word embedding (Model #11). Surprisingly, as we added caption features with attention (Model #12), the performance became worse except for BLEU1. This is inconsistent with our previous finding in cases without pretrained word embedding. Among the Video+Text setting models, Model #10 showed the best performance and was used as our submission to the DSTC7. We also experimented with only input text without the input video (Model #12 and #13). As we used pretrained 200-dimensional Glove embedding (Model #13), we achieved better performance and used this model as our submission for the Text Only setting.

**Discussion**

Using the prototype test data, we tested our best Video+Text setting model and compared some sample responses with the baseline model responses as well as the reference responses in Table 5. In terms of correctness, our responses are able...
Table 4: Objective evaluation results for models trained on official data and evaluated on prototype test data

<table>
<thead>
<tr>
<th>Model No.</th>
<th>Video</th>
<th>Text</th>
<th>RNN</th>
<th>cap-att</th>
<th>mm-att</th>
<th>mm-fusion</th>
<th>word-emb</th>
<th>BLEU B-1</th>
<th>BLEU B-2</th>
<th>BLEU B-3</th>
<th>BLEU B-4</th>
<th>METEOR</th>
<th>ROUGE-L</th>
<th>CIDEr</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I3D</td>
<td>rgb+VGGish</td>
<td>Dialog</td>
<td>LSTM</td>
<td>-</td>
<td>Baseline</td>
<td>No</td>
<td>0.272</td>
<td>0.176</td>
<td>0.120</td>
<td>0.086</td>
<td>0.122</td>
<td>0.298</td>
<td>0.842</td>
</tr>
<tr>
<td>2</td>
<td>I3D</td>
<td>rgb+VGGish</td>
<td>Dialog</td>
<td>GRU</td>
<td>-</td>
<td>Baseline</td>
<td>No</td>
<td>0.266</td>
<td>0.174</td>
<td>0.120</td>
<td>0.086</td>
<td>0.121</td>
<td>0.299</td>
<td>0.843</td>
</tr>
<tr>
<td>3</td>
<td>I3D</td>
<td>rgb+VGGish</td>
<td>Dialog</td>
<td>GRU</td>
<td>-</td>
<td>QuesProj+Conv</td>
<td>Baseline No</td>
<td>0.269</td>
<td>0.175</td>
<td>0.121</td>
<td>0.087</td>
<td>0.123</td>
<td>0.301</td>
<td>0.851</td>
</tr>
<tr>
<td>4</td>
<td>I3D</td>
<td>rgb+VGGish</td>
<td>Dialog</td>
<td>GRU</td>
<td>-</td>
<td>QuesProj+Conv</td>
<td>FC+HdmProd No</td>
<td>0.291</td>
<td>0.186</td>
<td>0.126</td>
<td>0.090</td>
<td>0.127</td>
<td>0.301</td>
<td>0.824</td>
</tr>
<tr>
<td>5</td>
<td>I3D</td>
<td>rgb+VGGish</td>
<td>Dialog</td>
<td>GRU</td>
<td>-</td>
<td>QuesProj+Conv</td>
<td>FC+HdmProd No</td>
<td>0.284</td>
<td>0.183</td>
<td>0.125</td>
<td>0.090</td>
<td>0.123</td>
<td>0.296</td>
<td>0.802</td>
</tr>
<tr>
<td>6</td>
<td>I3D</td>
<td>rgb+VGGish</td>
<td>Dialog+Caption</td>
<td>GRU</td>
<td>QuesProj+Conv</td>
<td>QuesProj+Conv</td>
<td>FC+HdmProd No</td>
<td>0.304</td>
<td>0.198</td>
<td>0.137</td>
<td>0.100</td>
<td>0.131</td>
<td>0.312</td>
<td>0.891</td>
</tr>
<tr>
<td>7</td>
<td>I3D</td>
<td>rgb+VGGish</td>
<td>Dialog+Caption</td>
<td>GRU</td>
<td>QuesProj+Conv</td>
<td>CapProj+Conv</td>
<td>FC+HdmProd No</td>
<td>0.298</td>
<td>0.194</td>
<td>0.135</td>
<td>0.097</td>
<td>0.129</td>
<td>0.309</td>
<td>0.867</td>
</tr>
<tr>
<td>8</td>
<td>I3D</td>
<td>rgb+VGGish</td>
<td>Dialog</td>
<td>GRU</td>
<td>-</td>
<td>QuesProj+Conv</td>
<td>FC+HdmProd Glove100</td>
<td>0.276</td>
<td>0.181</td>
<td>0.125</td>
<td>0.091</td>
<td>0.124</td>
<td>0.304</td>
<td>0.870</td>
</tr>
<tr>
<td>9</td>
<td>I3D</td>
<td>rgb+VGGish</td>
<td>Dialog</td>
<td>GRU</td>
<td>-</td>
<td>QuesProj+Conv</td>
<td>FC+HdmProd Glove200</td>
<td>0.307</td>
<td>0.204</td>
<td>0.144</td>
<td>0.106</td>
<td>0.136</td>
<td>0.320</td>
<td>0.995</td>
</tr>
<tr>
<td>10</td>
<td>I3D</td>
<td>rgb+VGGish</td>
<td>Dialog</td>
<td>GRU</td>
<td>-</td>
<td>QuesProj+Conv</td>
<td>FC+HdmProd Glove200</td>
<td>0.303</td>
<td>0.201</td>
<td>0.141</td>
<td>0.103</td>
<td>0.133</td>
<td>0.317</td>
<td>0.962</td>
</tr>
<tr>
<td>11</td>
<td>I3D</td>
<td>rgb+VGGish</td>
<td>Dialog+Caption</td>
<td>GRU</td>
<td>QuesProj+Conv</td>
<td>QuesProj+Conv</td>
<td>FC+HdmProd Glove200</td>
<td>0.314</td>
<td>0.204</td>
<td>0.142</td>
<td>0.102</td>
<td>0.136</td>
<td>0.317</td>
<td>0.940</td>
</tr>
<tr>
<td>12</td>
<td>-</td>
<td>Dialog+Caption</td>
<td>GRU</td>
<td>QuesProj+Conv</td>
<td>-</td>
<td>-</td>
<td>No</td>
<td>0.293</td>
<td>0.194</td>
<td>0.136</td>
<td>0.100</td>
<td>0.130</td>
<td>0.313</td>
<td>0.933</td>
</tr>
<tr>
<td>13</td>
<td>-</td>
<td>Dialog+Caption</td>
<td>GRU</td>
<td>QuesProj+Conv</td>
<td>-</td>
<td>-</td>
<td>Glove200</td>
<td>0.312</td>
<td>0.203</td>
<td>0.141</td>
<td>0.102</td>
<td>0.135</td>
<td>0.316</td>
<td>0.931</td>
</tr>
</tbody>
</table>

Figure 1: Overview of the proposed end-to-end multimodal dialog systems with attention mechanisms on multimodal features and video caption embedding
Table 5: Sample dialogues for an input video with responses generated by the best baseline model $A_{\text{baseline}}$, our best Video+Text model $A_{\text{ours}}$, and the reference response $A_{\text{ref}}$ from the prototype test dataset.

<table>
<thead>
<tr>
<th>Video Caption</th>
<th>$Q_1$: how many people in the video?</th>
<th>$A_{\text{ref}}$: there are two persons in video</th>
<th>$A_{\text{baseline}}$: there is one person in the video</th>
<th>$A_{\text{ours}}$: there are two people in the video</th>
</tr>
</thead>
<tbody>
<tr>
<td>“A person is working on their laptop, another person comes in carrying a bag of groceries. They set the groceries down and start cooking.”</td>
<td>$Q_2$: what gender are they?</td>
<td>$A_{\text{ref}}$: both males in video</td>
<td>$A_{\text{baseline}}$: they are both men in the video</td>
<td>$A_{\text{ours}}$: one is a male and the other is a male</td>
</tr>
<tr>
<td>$Q_3$: are they talking in the video?</td>
<td>$A_{\text{ref}}$: no, both are not talking to each other</td>
<td>$A_{\text{baseline}}$: yes they are talking in the video</td>
<td>$A_{\text{ours}}$: no one is talking in the video</td>
<td></td>
</tr>
<tr>
<td>$Q_4$: what is the second person doing?</td>
<td>$A_{\text{ref}}$: second person bring a vegetable bag come to kitchen room</td>
<td>$A_{\text{baseline}}$: he is eating a sandwich</td>
<td>$A_{\text{ours}}$: the person is standing in front of the refrigerator</td>
<td></td>
</tr>
<tr>
<td>$Q_5$: are they drinking something in the video?</td>
<td>$A_{\text{ref}}$: no, they are both not drinking</td>
<td>$A_{\text{baseline}}$: no, he is eating a sandwich</td>
<td>$A_{\text{ours}}$: no, they are not drinking anything in the video</td>
<td></td>
</tr>
</tbody>
</table>

To express the correct answers better than the baseline responses in $A_1$, $A_3$, and $A_5$, this indicates that our model is able to 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 are able to 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, we could argue that our response is a better possible response in this context as “standing in front of the refrigerator” is more appropriate than “eating a sandwich” action for this particular input video.

In addition, we also observe that our generated responses have a large proportion of negative answers i.e. answers that respond “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 “yes that is all happening in the video” to questions such as “is that all happened in the video?” This type of questions might require further cross-references to reason over the dialog history before generating a correct response. Using a hierarchical encoder for the dialog history might not be sufficient for this type of questions.

**Conclusion**

DSTC7 Track 3 offered a valuable opportunity to investigate multimodal dialog systems in a video-oriented setting rather than just visual setting (Das et al. 2017b) (Das et al. 2017a). It set a good framework to explore how state-of-the-art feature extraction models such as VGGish and I3D can be pretrained to extract the visual and audio features efficiently. We also found that techniques used in visual QA models such as (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 dialog setting further in the future with larger scale datasets and probably in other variations of dialog settings e.g. open-domain dialogs, task-oriented dialogs. Besides bootstrapping with pretrained word embeddings, we could also pretrain parts of the model on a larger dialog corpus that covers similar topics and types of questions and responses. An example corpus is the Movie QA dataset (Tapaswi et al. 2016) containing Q-A pairs constructed to query about movie contents. This corpus can be used to pretrain the model before further training with the DSTC7 training dataset. Alternatively, unsupervised pretraining with language models such as BERT (Devlin et al. 2018) has shown significant improvement in multiple NLP tasks and could be applied into multimodal dialog settings.

**Acknowledgements**

The first author is supported by A*STAR Graduate Scholarship. We also thank the insightful comments from the reviewers and E. Hovy.
References


