Reactive Multi-Stage Feature Fusion for Multimodal Dialogue Modeling

Yi-Ting Yeh, Tzu-Chuan Lin, Hsiao-Hua Cheng, Yi-Hsuan Deng, Shang-Yu Su, Yun-Nung Chen
National Taiwan University, Taiwan
{r07922064,b04705003,b03902024,b04902013,f05921117}@csie.ntu.edu.tw  y.v.chen@ieee.org

Abstract
Visual question answering and visual dialogue tasks have been increasingly studied in the multimodal field towards more practical real-world scenarios. A more challenging task, audio visual scene-aware dialogue (AVSD), is proposed to further advance the technologies that connect audio, vision, and language, which introduces temporal video information and dialogue interactions between a questioner and an answerer. This paper proposes an intuitive mechanism that fuses features and attention in multiple stages in order to well integrate multimodal features, and the results demonstrate its capability in the experiments. Also, we apply several state-of-the-art models in other tasks to the AVSD task, and further analyze their generalization across different tasks.

Introduction
Nowadays, many real-world problems require to integrate information from multiple sources. Sometimes such problems involve distinct modalities, such as vision, language and audio. Therefore, many AI researchers are trying to build models that can deal with tasks using different modalities, and multimodal applications have become an increasing popular research topic. For example, visual question answering (VQA) (Antol et al. 2015) is the task that aims at testing whether an AI model can successfully summarize an image, text description of that image and then generate a correct response. The visual dialog task focuses on examining whether a system can interact with human via conversations (Das et al. 2017), where in a conversation, the system is expected to answer questions correctly given an input image. However, it is difficult to well converse with users when only accessing a single image without audio and dynamic scenes.

Motivated by the need for scene-aware dialogue systems, audio visual scene-aware dialogue (AVSD) is recently proposed by Alamri et al., providing more dynamic scenes with video modal rather than static images in VQA or Visual dialogue tasks. With richer information, machines can be trained to carry on a conversation with users about objects and events around them. To tackle this problem, this paper focuses on 1) better encoding multimodal features and 2) better decoding the responses with consideration of the encoded information. We propose an reactive encoder that is capable of fusing multimodal features and paying attention correctly, and a decoder that investigates different decoding mechanisms for better generating the dialogue responses. In this paper, our main contributions are 3-fold:
• This paper proposes a simple but effective way to fuse different modalities by the $1 \times 1$ convolution followed by a weighted sum operation.
• The proposed multi-stage fusion mechanism can encourage the model to thoroughly understand the question.
• This paper first attempts to generalize top-down attention proposed by Anderson et al. and investigates different attentional decoding methods in a principled way.

Task Description
The task is audio visual scene-aware dialogue (AVSD), which tests the capability of dialogue responses given audio, visual and dialogue contexts. The process of data collection is illustrated in the left part of Figure 1, where the collected dataset contains 11,156 visual dialogues. Each dialogue contains a video and dialogue texts, where there are pre-extracted feature modalities using VGGish (Hershey et al. 2017) (vggish) and I3D models (Carreira and Zisserman 2017) (i3d) for the video, and the dialogue texts contain a video caption, a summary written by the questioner after 10 rounds Q/A and dialogue history, denoted as caption, summary and dialogue respectively. Note that our model in the experiments only considers answer parts as our dialogue features. Our goal is to play the role of an answerer, who can reply the reasonable answer to the questioner illustrated in the right part of Figure 1.

Proposed Approach
Our system can be viewed as an encoder-decoder model, which is commonly used in conversation modeling (Sutskever, Vinyals, and Le 2014). We first explain the feature selection procedure, describe the basic fusion model, and then detail the novel design for the proposed model. In the encoder, we apply basic fusion, multi-stage fusion, and $1 \times 1$ convolution fusion methods. In the decoder, we apply an original decoder, an attention decoder, and a top-down attention LSTM decoder. The detail is described below.
Feature Selection
Considering that multiple modalities are involved in this task, it is important to choose proper and useful features for it. However, some features are too noisy to model and may result in ambiguity. In our proposed model, we decide to ignore \texttt{i3d} and \texttt{summary} for our feature set.

There are several reasons to drop \texttt{i3d}. The main reason is that \texttt{i3d} has too much information and thus contains many noises, which may hurt the stability of our model. We assume that the useful information contained in \texttt{i3d} can be obtained by combining \texttt{vggish} and texts. Because the audio tends to attract attention of people and be the key in the video, \texttt{vggish} provides more compact information and texts supply detailed description. In our experiments, we also found that removing \texttt{i3d} does not lead to a huge degradation of performance, implying that in fact we can answer most questions without this feature set in our model setup.

The reason of removing \texttt{summary} in our feature set is the concern about misleading and impracticality. We observe that \texttt{caption} usually contains more concise description and more rich information than \texttt{summary}, so our model only concentrates on \texttt{caption} to better answer questions. Another consideration is about how \texttt{summary} is generated. It is also more reasonable and practical to remove \texttt{summary} from our feature set, considering that the summary is generated by the questioner who does not see the video. For example, Table 1 shows a misleading \texttt{summary} example that contains incorrect information.

<table>
<thead>
<tr>
<th>Type</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Caption</strong></td>
<td>Man walks over to laptop and throws towel over his shoulder. He sits down and wipes and scratches his face with his hands and begins staring at the laptop.</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td>A man walks into a room and sits down while looking at a laptop on the floor.</td>
</tr>
</tbody>
</table>

Table 1: Misleading \texttt{summary} example.

Feature Encoder
After removing noisy features, our model utilizes several sequential features in different forms for the task. Let $z^i = \{x^i_1, \ldots, x^i_{T^i}\}$ represent the $i$-th feature, which can contain both video features and text word embeddings. For text-based features, \texttt{Glove} is applied to obtain word embeddings. Then for each temporal sequence of features, an uni-directional GRU encoder is used to encode all sequential features into a common space $\mathbb{R}^d$, producing $z^i = \{z^i_1, \ldots, z^i_{T^i}\}$, where $z^i_t \in \mathbb{R}^d$ and can be the representation of the whole video or the dialogue history. The illustration can be found in Figure 1.

Basic Fusion Model
In our basic fusion model, we use the last hidden states $\{z^1_{T^1}, \ldots, z^n_{T^n}\}$ for each modality as the encoded representation, and fuse different modalities into a global multimodal representation, $z$, using a weighted sum operation. $T_i$ indicates the number of hidden states of the $i$-th feature. The fused representation is computed by

$$z = \frac{\sum_{i=1}^{n} w_i z^i_{T^i}}{\sum_{i=1}^{n} w_i},$$

where $w_i$ is a trainable scalar weight and $n$ is the number of feature types.

We initialize the first hidden state $h_0 = z$ in the decoder and use a GRU to generate the answer. The decoding procedure follows $h_{t+1} = \text{GRU}(y_t, h_t)$, where $h_t$ is the $t$-th hidden state in the decoder and $y_t$ is the output in the $t$-th step.

$$\Pr(y_t \mid y_{1:t-1}) = \text{softmax}(W h_t + b),$$

where $W \in \mathbb{R}^{|\Sigma| \times d_h}$ and $b \in \mathbb{R}^{|\Sigma|}$ are learned weights. $\Sigma$ and $d_h$ are the vocabulary and the dimension of the hidden state respectively.

Multi-Stage Fusion
In addition to the above basic fusion model, here we introduce a novel feature fusion mechanism using multi-stage attention. The attention enables the model to focus on a targeted area within a context that is relevant to answer the question (Antol et al. 2015; Xiong, Merity, and Socher 2016). With the attention mechanism, we can enhance or modify vectors set with the information from other vectors; it is called the fusion process where information of some vectors is fused into other vectors (Huang et al. 2018).

An general attention function can be described as computing the weighted sum of values, where the weight is given by the attention score between a query and a key via a
score function. For simplicity, our model uses the same vector for the value and the key. Here we choose the recently popular multi-head attention as our attention score function (Vaswani et al. 2017):

$$
\text{MultiHead}(Q,V) = [\text{head}_1; \ldots; \text{head}_n]W^O,
$$

$$\text{head}_i = \text{Att}(QW^Q_i, VW^K_i, VW^V_i),$$

$$\text{Att}(Q,V,K) = \text{softmax} \left( \frac{QV^T}{\sqrt{d}} \right)K,$$

(3)

where $W^Q_i \in \mathbb{R}^{d \times d_k}$, $W^K_i \in \mathbb{R}^{d \times d_k}$, $W^V_i \in \mathbb{R}^{d \times d_k}$ and $W^O \in \mathbb{R}^{d \times d}$ are parameters. $[A; B]$ means the concatenation of two vectors $A$ and $B$. $n$ is the number of heads and $d$ is the dimension. $d_k = d/n$. Note that we add a residual connection after the attention to help model converge faster (He et al. 2016).

Our multi-stage fusion starts from fusing encoded question ($z^Q$) into encoded caption ($z^C$) and dialogue ($z^D$), and we consider the concatenation of caption and dialogue to be the context. We apply the multi-head attention as (3) to generate fusion vectors and then feed them into a bidirectional GRU reader to obtain the fused representation:

$$U^{fusion} = \text{MultiHead}(z^C; z^D; z^Q),$$

$$z^{fusion} = \text{BiGRU}(U^{fusion}),$$

(4)

While the context is necessary to infer the answer, an issue in our current fusion representation is lack of full understanding about the context. To address this problem, we first compute the attention $U^{self}$ using self-attention, which usually improves performance a lot in machine comprehension and question answering (Wang et al. 2017; Vaswani et al. 2017). We also feed $U^{self}$ into a bidirectional GRU and obtain $z^{self}$, where $z^{self}$ can be viewed as vectors that fuse all textual information together. Formally written as:

$$U^{self} = \text{MultiHead}(z^{fusion}, z^{fusion}),$$

$$z^{self} = \text{BiGRU}(U^{self}).$$

(5)

Then we have tried many different ways to leverage $z^{fusion}$ and $z^{self}$, and find the most efficient way is to use the last hidden state of them and add them into computation of the weighted sum in (1). Therefore, the multi-stage fusion mechanism intuitively offers richer information to enhance the first hidden state passed to the decoder.

1x1 Convolution Fusion

Instead of a simple weighted sum in (1), we find that it is beneficial to insert 1x1 convolution (Szegedy et al. 2015) before the weighted sum to help each feature channel interact with each other. The weighted sum operation can also be considered as a single channel 1x1 convolution with output normalization.

To model the multimodal features using $n$ channels, we utilize the last hidden state for each feature type, $\{z_1^{f_1}, \ldots, z_n^{f_n}\}$. We run $n_c$ channels 1x1 convolution on the inputs and obtain $\{z_1, \ldots, z_{n_c}\}$. Then we compute the weighted sum as (1):

$$z = \sum_{i=1}^{n_c} w_i z_i,$$

(6)

where $w_i$ is a trainable scalar weight.

Attention Decoder

In the decoding phase, attention is also commonly used to help model focus on important information in the encoder-decoder model (Luong, Pham, and Manning 2015). We
choose a commonly used multiplicative attention (Britz et al. 2017) and the computation follows:

\[
    a_{i,t} = v_i^T U^T V h_t, \\
    \alpha_t = \text{softmax}(a_t),
\]

where \( U \in \mathbb{R}^{k \times d_v}, V \in \mathbb{R}^{k \times d_h} \) are trainable weights and \( v_i \in \mathbb{R}^{d_v \times 1}, h_t \in \mathbb{R}^{d_h \times 1} \) are inputs. \( k \) is the attention dimension. Note that \( a_{i,t} = v_i^T U^T V h_t \) is also called low-rank bilinear method (Pirsiavash, Ramanan, and Fowlkes 2009; Kim et al. 2016).

In the decoding step \( t \), the attention of a query \( h_t \) and values \( v = \{v_1, \ldots, v_r\} \) is computed as:

\[
    C_t = \sum_i \alpha_i v_i.
\]

We then concatenate the context vector \( C_t \) with the next step input \( y_t \) as the attention-enhanced input and pass it through GRU, formally written as:

\[
    h_{t+1} = \text{GRU}([y_t; C_t], h_t).
\]

Here we choose a simple concatenation of all encoded features to form \( v = \{z_1, \ldots, z_r\} \) into this model. We consider this model as an enhanced attention decoder, where an additional attention LSTM is used to further track what has been attended and expect it can provide more useful information for the attention module.

Here the top-down attention LSTM has two LSTM layers, which are named attention LSTM and language LSTM. We consider the first LSTM layer as an attention model and the second LSTM layer as a language model. The main purpose of separating two LSTM cells is to modularize the learning process into two parts: capturing the importance of multimodal features (attention LSTM) and language generation (language LSTM).

The input vector to the attention LSTM at time step \( t \) consists of the previous output hidden states of the language LSTM concatenated with the mean pooled features \( \bar{v} = \frac{1}{r} \sum_i v_i \) and a word embedding of the previously generated word. \( x_t^1 = [h_t^1; \bar{v}; \text{embed}(y_{t-1})] \), where \( \text{embed}(y_{t-1}) \) is the word embedding of \( y_{t-1} \). Given the output \( h_t^1 \) of the attention LSTM, we compute the attention score as follows:

\[
    \beta_t = \text{softmax}(b_t),
\]

Then we calculate the context vector of attended features as:

\[
    C_t = \sum_i \beta_i v_i.
\]

The input to the language LSTM consists of the context vector of attended vectors concatenated with the output of the attention LSTM, \( x_t^2 = [h_t^2; C_t] \). We pass \( x_t^2 \) into the language LSTM to obtain \( h_t^2 \) and compute the output word distribution as:

\[
    \Pr(y_t | y_{1:t-1}) = \text{softmax}(Wh_t^2 + b),
\]

which is exactly same as the basic decoding setup in (2).

Training and Testing

The full model is trained in an end-to-end manner to optimize the dialogue generation results in (2). During testing, beam search is applied to generate the fluent answering sentences given the contexts.

Experiments

To evaluate whether the proposed model is capable of modeling dialogues with multimodal scenarios, a set of experiments is conducted.

Experimental Setup

We run our experiments using the official AVSD dataset (Alamri et al. 2018a), which consists of 7659, 1787, 1710 dialogues for train, dev, and test sets respectively. In the dataset, almost all dialogues contain 10 question/answer pairs. We use the evaluation script \textit{nlg-eval} \footnote{https://github.com/Maluuba/nlg-eval} to compute objective evaluation scores for the model, where the punctuations, ‘,’ and ‘.’, are not taken into account. In the experiments, the compared baselines include a naïve copy baseline and the released baseline (Alamri et al. 2018b), where the naïve baseline simply copies the questions as its answer and serves as a lower bound of this task. Note that because we found there are many word overlaps between the question and the answer, the copy baseline in fact has very strong performance under objective evaluation metrics.

Fine-Grained Response Revision

In the answering sentence to the yes/no question, adding “yes” or “no” at the beginning may affect the performance. Therefore, we train a classifier to predict whether “yes” or “no” tokens should be inserted. This classifier consists of an RNN encoder followed by a linear projection output layer. Note that we first filter out yes/no questions by rules, since yes/no questions should not be answered with this token.

Results

Table 2 shows the experimental results, where our models only take \textit{vggish}, \textit{caption}, and \textit{dialog} into account, while the released baseline additionally use \textit{i3d} as their input features. While the basic model is the simple architecture without any mechanism, it shows surprising performance specific in CIDEr (from 0.69 to 0.99) (Vedantam, Zitnick, and Parikh 2015). Because CIDEr measures the ability of capturing correct objects in the image and places less attention on words which frequently appear in answers such as...
stopwords, the result tells that the basic model already captures important information and objects in the video without the complicated design of attention. In the proposed model, we analyze the results for different encoding and decoding mechanisms.

**Encoding** We examine the effectiveness of the proposed two fusion methods, multi-stage fusion and 1x1 convolution fusion. The proposed multi-stage fusion provides model deeper understanding of input features and then further boosts the performance of CIDEr. The 1x1 convolution fusion also enable interactions between different modalities and improves the performance.

**Decoding** Furthermore, two decoding mechanisms, attention decoder and top-down attention LSTM, are investigated. Two variants of decoders have different advantages. The attention decoder mainly improves BLEU scores (Papineni et al. 2002), while the top-down attention boosts the performance for the ROUGE score (Lin 2004). It may be because the attention decoder enhances the ability of our model to capture the specific word usage in dialogues. On the other hand, the top-down attention LSTM adds the attention to fuse features into what is attended in previous decoding, and thus improves the ROUGE-L where the recall of generated words is relatively important. Hence, the attention decoder is applied as our decoder in the following experiments due to its strong ability of learning how to use particular word in certain context and think these modules will have complementary effects.

**Proposed Fusion Model** Our full fusion model is composed of multi-stage fusion, 1x1 convolution fusion, and the attention decoder, and further improvement is investigated and analyzed in the experiments shown in Table 2. Compared to the released baseline, we can find that even the proposed basic model can outperform it by a large margin in CIDEr, improve METEOR and ROUGE-L, and achieve comparable results in BLEU-4. Note that BLEU-1 and BLEU-2 are not good metrics to evaluate quality of generated sentences in dialogues, because they only measure unigram and bigram precision. The copy baseline shows that we can reach high BLEU-1 and BLEU-2 by simply copying the questions, implying that these are not good indicators. The comparable BLEU-4 shows that our basic model in fact have similar ability to capture the sentence structure compared to the released baseline. Furthermore, our proposed full fusion model can significantly outperforms the released baseline for almost all metrics.

Here we also conduct experiments on two variants of our full fusion model. The first one does not use vggish features, so it is a pure text model, and the second one additionally uses i3d features, which can be fairly compared with the released baseline. Table 2 shows that the pure text model also achieves better BLEU scores compared to the model that uses additional video features, implying that the language part of test data provides rich enough information for the model to answer most questions. On the other hand, using i3d features results in performance drop, confirming our concern about noisy i3d features and demonstrating the need of the proposed feature selection.

**Analyzing Fusion Operations**

In Table 3, we test different representation fusion methods to fuse features \( \{ z_{1i}, \ldots, z_{Ti} \} \) into \( v \), while we choose weighted sum in the simple model (Table 2). 3 different fusion methods are listed below:

- **Product**: \( z = \prod_i z_{1i} \)
- **Sum**
- **Concat**
<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU-1</th>
<th>BLEU-2</th>
<th>BLEU-3</th>
<th>BLEU-4</th>
<th>METEOR</th>
<th>ROUGE-L</th>
<th>CIDEr</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (i3d) (Alamri et al. 2018b)</td>
<td>.621</td>
<td>.48</td>
<td>.379</td>
<td>.305</td>
<td>.217</td>
<td>.481</td>
<td>.733</td>
<td>—</td>
</tr>
<tr>
<td>Baseline (i3d + vggish) (Alamri et al. 2018b)</td>
<td>.626</td>
<td>.485</td>
<td>.383</td>
<td>.309</td>
<td>.215</td>
<td>.487</td>
<td>.746</td>
<td>—</td>
</tr>
<tr>
<td>Basic Fusion Model (Official)</td>
<td>.636</td>
<td>.510</td>
<td>.417</td>
<td>.345</td>
<td>.224</td>
<td>.505</td>
<td>.877</td>
<td>—</td>
</tr>
<tr>
<td>Basic Fusion Model (Prototype)</td>
<td>.640</td>
<td>.513</td>
<td>.416</td>
<td>.342</td>
<td>.223</td>
<td>.504</td>
<td>.837</td>
<td>3.188</td>
</tr>
<tr>
<td>+ Fine-Grained Response Revision</td>
<td>.641</td>
<td>.513</td>
<td>.416</td>
<td>.342</td>
<td>.223</td>
<td>.504</td>
<td>.836</td>
<td>—</td>
</tr>
<tr>
<td>Fusion Text Model</td>
<td>.592</td>
<td>.468</td>
<td>.375</td>
<td>.304</td>
<td>.206</td>
<td>.475</td>
<td>.729</td>
<td>2.928</td>
</tr>
<tr>
<td>+ Fine-Grained Response Revision</td>
<td>.595</td>
<td>.477</td>
<td>.376</td>
<td>.304</td>
<td>.207</td>
<td>.477</td>
<td>.731</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 4: The submitted results in the official test set.

<table>
<thead>
<tr>
<th>Features</th>
<th>B-1</th>
<th>B-2</th>
<th>B-3</th>
<th>MET.</th>
<th>R-L</th>
<th>CIDEr</th>
</tr>
</thead>
<tbody>
<tr>
<td>dialogue</td>
<td>.228</td>
<td>.153</td>
<td>.110</td>
<td>.083</td>
<td>.116</td>
<td>.296</td>
</tr>
<tr>
<td>+ caption</td>
<td>.229</td>
<td>.154</td>
<td>.111</td>
<td>.084</td>
<td>.119</td>
<td>.305</td>
</tr>
<tr>
<td>+ vggish</td>
<td>.232</td>
<td>.157</td>
<td>.112</td>
<td>.084</td>
<td>.120</td>
<td>.305</td>
</tr>
<tr>
<td>+ i3d</td>
<td>.239</td>
<td>.161</td>
<td>.116</td>
<td>.087</td>
<td>.121</td>
<td>.309</td>
</tr>
</tbody>
</table>

Table 5: Results of the feature ablation tests. Model architectures are simple model using different input feature set.

<table>
<thead>
<tr>
<th>Features</th>
<th>B-1</th>
<th>B-2</th>
<th>B-3</th>
<th>B-4</th>
<th>MET.</th>
<th>R-L</th>
<th>CIDEr</th>
</tr>
</thead>
<tbody>
<tr>
<td>vggish</td>
<td>.233</td>
<td>.159</td>
<td>.115</td>
<td>.087</td>
<td>.121</td>
<td>.309</td>
<td>1.019</td>
</tr>
<tr>
<td>caption</td>
<td>.232</td>
<td>.158</td>
<td>.114</td>
<td>.087</td>
<td>.120</td>
<td>.307</td>
<td>1.007</td>
</tr>
<tr>
<td>dialogue</td>
<td>.234</td>
<td>.159</td>
<td>.115</td>
<td>.087</td>
<td>.120</td>
<td>.309</td>
<td>1.023</td>
</tr>
<tr>
<td>all</td>
<td>.245</td>
<td>.162</td>
<td>.115</td>
<td>.086</td>
<td>.119</td>
<td>.308</td>
<td>0.977</td>
</tr>
</tbody>
</table>

Table 6: Results of the attention ablation tests. Decoder is attention decoder which attends on different feature set of v.

- Sum: $z = \sum_{i}^{n} z_{T_i}^j$
- Concat: $z = W^{cat}[z_{T_1}^{1}; \ldots; z_{T_n}^{n}], W^{cat} \in \mathbb{R}^{d_h \times dn}$

As we expect, the weighted sum outperforms other methods except concat. While weighted sum is better in CIDEr, concat outperforms it in BLEU scores. Note that the number of parameters in the weighted sum is significantly smaller than the number in concat, because it only needs to learn n scalars and concat uses $d_h \times dn$ parameters. Considering the model size and extensibility, the weighted sum is considered as a better choice.

Ablation Test

In order to investigate the usefulness of features and attention, two sets of experiments are conducted for analysis.

Feature

In Table 5, we examine the effectiveness of each feature set by fixing the model architecture to the basic model. The results demonstrate that all feature sets contain semantically meaningful information, and the performance of the basic model gradually improves when we add more features. When using a simple model architecture, using i3d features can improve the performance, which is different from the result shown in Table 2. The reason is probably that i3d is too noise to learn, and thus we observe these different results when using i3d features.

Attention

We also test our attention decoder on different attention values $v$. Different from our expectation, results in Table 6 show that different attention values lead to little difference of performance. In sum, dialogue gives the best CIDEr score, and using all input features together leads to the best average BLEU score.

Official Results

We submit the following five predictions:

- Basic fusion model (official): a basic simple fusion model trained on official data
- Basic fusion model (prototype): a basic simple fusion model trained on prototype data (w/o and w/ fine-grained response revision)
- Fusion text model: a fusion model with multi-stage fusion and attention decoder that only take texts into consideration (w/o and w/ fine-grained response revision)

Note that the submitted models here used different experiment settings from the one proposed in this paper and the models were not well trained. Here feature selection is not applied, so the basic fusion model considers the i3d features. The results are shown in Table 4, and the performance is evaluated with consideration of multiple correct responses. Because the numbers in Table 4 are not comparable with ones in Table 2, we also show the submitted results in the last two rows of Table 2 for fair comparison. Our simple model outperforms the released baseline in all metrics, especially in CIDEr (by 12%) and human evaluation (by 11%). The proposed fine-grained response revision brings the small improvement, although the model which is not well trained.

Our proposed basic fusion architecture has less parameters but achieves better performance than the released baseline, which uses a complicated attention scheme. The results show that the simple $1 \times 1$ convolution fusion may have potential of disentangling each dimension on $z$ shown in Figure 2 and then resulting in more semantically meaningful representations for multiple modalities.

Conclusion

This paper proposes an intuitive and effective visual dialogue model based on an encoder-decoder design. We develop a set of modules that are capable of fusing multimodal features with and performing context-aware decoding. In the experiments of the audio visual scene-aware dialogue task, we validate the effectiveness of each module and analyze whether it is necessary to use all features to correctly answer questions. The attempt in this paper bridges different modalities and encourages further model exploration for advancing the important research area about multimodality.
References


Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In EMNLP.


<table>
<thead>
<tr>
<th>Video Frame</th>
<th>Type</th>
<th>Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>a person in the entryway is working on something on their phone. they start throwing some clothes at another person who is watching them oddly.</td>
<td>are they laughing in the video?</td>
</tr>
<tr>
<td>Question</td>
<td>they are not laughing out loud but are smiling and appear maybe to be flirting a bit. no, they are both talking to each other at the end of the video. no, they are not talking. no, they are not laughing.</td>
<td></td>
</tr>
</tbody>
</table>

| Caption     | one person walks into the room, pours a glass of something, then grasps a phone and leaves. | does he appear to be calm in the video?                                   |
| Question    | yes he appears to very calm in the video. no he does not show any emotion. he seems to be in a good mood. yes he seems to be calm. |

Table 7: The testing examples of the answers from our models.

Appendix

Hyperparameter Setting

For all text features, we split them by space and use pre-trained word vectors, Glove (Pennington, Socher, and Manning 2014), to obtain fixed word embedding for each word. The hidden state dimension of all uni-direction GRU is 256, and one of bidirectional GRU is set to 128 for dimension control. We use the Adam optimizer (Kingma and Ba 2014) with $\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The initial learning rate is $5 \times 10^{-4}$ and we multiply it by 0.1 every 6,000 update iterations. We add $10^{-4}$ L2 penalty and apply dropout (Srivastava et al. 2014) to the input word embeddings and video features, where the dropout rate is set to 0.5. The batch size varies from 8 to 16 according to our available computing resources as regularization. During training, the early stop mechanism is applied. During testing, we use beam search and set the beam size equal to 5 to generate our final answers.

Qualitative Analysis

We show some examples in Table 7 to qualitatively evaluate our model. In the first example, a woman is using her phone, while a man is staring at her in the video. One question in the dialogue about this video is “Are they laughing in the video?”, which should be easy for the model with incorporation of audio features (vggish). However, the released baseline says “no, they are both talking to each other at the end of the video”, which is totally wrong because they only appear to be happy and a little flirting but do not talk or laugh. On the other hand, our model is able to correctly capture the audio features and answer they don’t talk and laugh in the video. An interesting difference of the basic fusion model and the multi-stage fusion model is the word they choose. The basic fusion model uses “talking” while the multi-stage model picks the word “laughing”, which is more proper for this question. This example demonstrates that our proposed model can help system understand and catch what happens in the video and what the questioner asks.

In the second example, a person walks into the room, drink a water, picks a phone and leaves. It is difficult to tell whether the man is calm from few video frames, because face expression of the man is not clear in the video. However, our model can answer correctly using the fact that this man does not make large sound, which is the evidence that this man moves slowly, and calmly. The previous rounds of the dialogue mention that this man pour water to the coup and drink from it, which is another evidence that this man is not hurry to pick his phone and leave. To answer this hard question correctly, it is necessary to fuse evidences from both audio and texts, and our model is capable of generating the correct answer by fusing the multimodal features properly.