Dynamic Graph Representation Learning for Video Dialog via Multi-Modal Shuffled Transformers

Geng, Shijie; Gao, Peng; Chatterjee, Moitreya; Hori, Chiori; Le Roux, Jonathan; Zhang, Yongfeng; Li, Hongsheng; Cherian, Anoop

TR2021-010 March 11, 2021

Abstract

Given an input video, its associated audio, and a brief caption, the audio-visual scene aware dialog (AVSD) task requires an agent to indulge in a question-answer dialog with a human about the audio-visual content. This task thus poses a challenging multi-modal representation learning and reasoning scenario, advancements into which could influence several human-machine interaction applications. To solve this task, we introduce a semantics-controlled multi-modal shuffled Transformer reasoning framework, consisting of a sequence of Transformer modules, each taking a modality as input and producing representations conditioned on the input question. Our proposed Transformer variant uses a shuffling scheme on their multi-head outputs, demonstrating better regularization. To encode fine-grained visual information, we present a novel dynamic scene graph representation learning pipeline that consists of an intra-frame reasoning layer producing spatio-semantic graph representations for every frame, and an inter-frame aggregation module capturing temporal cues. Our entire pipeline is trained end-to-end. We present experiments on the benchmark AVSD dataset, both on answer generation and selection tasks. Our results demonstrate state-of-the-art performances on all evaluation metrics.

AAAI Conference on Artificial Intelligence
Dynamic Graph Representation Learning for Video Dialog via Multi-Modal Shuffled Transformers

Shijie Geng†, Peng Gao‡, Moitreya Chatterjee§, Chiori Hori‡,
Jonathan Le Roux†, Yongfeng Zhang‡, Hongsheng Li†, Anoop Cherian†

†Department of Computer Science, Rutgers University, Piscataway, NJ, USA
‡The Chinese University of Hong Kong  §University of Illinois at Urbana Champaign, Urbana, IL, USA
¶Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA

Abstract

Given an input video, its associated audio, and a brief caption, the audio-visual scene aware dialog (AVSD) task requires an agent to indulge in a question-answer dialog with a human about the audio-visual content. This task thus poses a challenging multi-modal representation learning and reasoning scenario, advancements into which could influence several human-machine interaction applications. To solve this task, we introduce a semantics-controlled multi-modal shuffled Transformer reasoning framework, consisting of a sequence of Transformer modules, each taking a modality as input and producing representations conditioned on the input question. Our proposed Transformer variant uses a shuffling scheme on their multi-head outputs, demonstrating better regularization. To encode fine-grained visual information, we present a novel dynamic scene graph representation learning pipeline that consists of an intra-frame reasoning layer producing spatio-semantic graph representations for every frame, and an inter-frame aggregation module capturing temporal cues. Our entire pipeline is trained end-to-end. We present experiments on the benchmark AVSD dataset, both on answer generation and selection tasks. Our results demonstrate state-of-the-art performances on all evaluation metrics.

Introduction

The success of deep learning in producing effective solutions to several fundamental problems in computer vision, natural language processing, and speech/audio understanding has provided an impetus to explore more complex multi-modal problems at the intersections of these domains, attracting wide interest recently (Zhu et al. 2020). A few notable ones include: (i) visual question answering (VQA) (Antol et al. 2015; Yang et al. 2003), the goal of which is to build an agent that can generate correct answers to free-form questions about visual content, (ii) audio/visual captioning (Hori et al. 2017; Venugopalan et al. 2015; Xu et al. 2015; Drossos, Lipping, and Virtanen 2019), in which the agent needs to generate a sentence in natural language describing the audio/visual content, (iii) visual dialog (Das et al. 2017), in which the agent needs to engage in a natural conversation with a human about a static image, and (iv) audio-visual scene-aware dialog (AVSD) (Alamri 2019; Hori et al. 2019) – that generalizes (i), (ii), and (iii) – in which the agent needs to produce a natural answer to a question about a given audio-visual clip, in a conversation setting or select the correct answer from a set of candidates. The ASVD task1 emulates a real-world human-machine conversation setting that is potentially useful in a variety of practical applications, such as building virtual assistants (Deruyttere et al. 2019) or in human-robot interactions (Thomas et al. 2019). See Figure 1 for an illustration of this task.

The generality of the AVSD task, however, poses a challenging multi-modal representation learning and reasoning problem. Specifically, some of the input modalities to this task may offer complementary information (such as video and audio), while a few others may be independent (audio and captions), or even conflict with each other, e.g., the provided text (Captions/dialogs) may include details from human experience that are absent in the video (e.g., “I think...”), or may include abstract responses (“happy”, “bored”, etc.) that may be subjective. Thus, the main quest in this task is to represent these modalities such that inference on them is efficient and effective. Previous approaches to this problem used holistic video features produced by a generic 3D convolutional neural network (Carreira and Zisserman 2017), and either focused on extending attention models on these features to include additional

1https://video-dialog.com/
Our proposed framework has similarities with prior works along three different axes, viz. (i) graph-based reasoning, (ii) multi-modal attention, and (iii) visual dialog methods.

**Scene Graphs:** (Johnson et al. 2015) combine objects detected in static images, their attributes, and object-object relationships (Lu et al. 2016) to form a directed graph that not only provides an explicit and interpretable representation of the image, but is also seen to be beneficial for higher-order reasoning tasks such as image captioning (Li and Jiang 2019; Yang et al. 2019), and visual question answering (Ghosh et al. 2019; Norcliffe-Brown, Vafeias, and Parisot 2018; Geng et al. 2019, 2020). There have been efforts (Wang et al. 2018; Girdhar et al. 2019; Jain et al. 2016; Herzig et al. 2019; Jiang et al. 2017; Tsai et al. 2019) at capturing spatio-temporal evolution of localized visual cues. In (Wang and Gupta 2018), a space-time graph reasoning framework is proposed for action recognition. Similarly, the efficacy of manually-labeled video scene graphs is explored in (Ji et al. 2020). Similar to ours, they use object detections per video frame, and construct a spatio-temporal graph based on the affinities of the features from the detected objects. Spatio-temporal graphs using knowledge distillation is explored for video captioning in (Pan et al. 2020). In contrast, our task involves several diverse modalities, demanding richer architectural choices. Specifically, we present a hierarchically organized intra/inter-frame reasoning pipeline for generating visual memories, trained via neural message passing, offering a powerful inference engine. Our ablation studies demonstrate the usefulness of these modules.

**Multi-modal Fusion/Attention:** has been explored in several prior works (Hori et al. 2017, 2018, 2019; Shi et al. 2020a), however does not use the power of Transformers. Self-attention and feature embeddings using Transformers is attempted in multi-modal settings (Gao et al. 2019a,b; Shi et al. 2020b), however only on static images. Bilinear fusion methods (Ben-Younes et al. 2019; Fukui et al. 2016) have been explored towards inter-modality semantic alignment, however they often result in high-dimensional interaction tensors that are computationally expensive during inference. In contrast, our pipeline is the first to leverage the power of Transformers in a hierarchical graph reasoning setup for video dialogs and is cheap to compute.

**Multi-modal Dialogs:** have been explored in various ways before. Free-form human-like answers were first considered in (Das et al. 2017), which also proposed the VisDial dataset, however on static images. A difficulty in designing algorithms on multi-modal data is in deriving effective attention mechanisms that can divulge information from disparate cues. To tackle this challenge, (Wu et al. 2018) proposed a sequential co-attention scheme in which the neural embeddings of various modalities are co-attended with visual embeddings in a specific order. (Schwartz et al. 2019) generalized the co-attention problem by treating the modalities as nodes of a graph, aggregating them as factors, using neural message passing. We use a combination of these two approaches; specifically we use Transformer encoders for embedding each modality, and attend on these multi-modal embeddings sequentially to generate the answer. Further, in
contrast to (Schwartz et al. 2019; Wu et al. 2018), that tackle solely the answer generation problem, we consider the answer selection task on AVSD as well. (Yeh et al. 2019) also proposed using Transformers (Vaswani et al. 2017) for fusing audio-visual features on the AVSD task. A multi-step reasoning scheme is proposed in (Gan et al. 2019) using joint attention via an RNN for generating a multi-modal representation. The Simple baseline (Schwartz, Schwing, and Hazan 2019) extends factor graphs (Schwartz et al. 2019) for the AVSD problem demonstrating promising results. A multi-modal Transformer for embedding various modalities and a query-aware attention is introduced in (Le et al. 2019). (Le and Hoi 2020) fine-tunes pretrained GPT-2 to obtain improved performance. However, these works neither consider richer visual representations using scene graphs, nor variations of the standard Transformers, like the shuffling scheme we present.

Proposed Method

In this section, we will first present our spatio-temporal scene graph representation (STSGR) for encoding the video sequences, following which we elaborate on our multi-modal shuffled Transformer architecture.

Overview of Spatio-Temporal Scene Graphs

Given a video sequence V, let C denote the associated human-generated video caption, and let (Q_i, A_i) represent the tuple of the text-based i-th question and its answer in the given human dialog about V (see Fig. 1). We will succinctly represent the dialog history by H = \{(Q_1, A_1), \ldots, (Q_{i-1}, A_{i-1})\}. Further, let Q_i represent the question under consideration. The audio-visual scene-aware dialog (AVSD) task requires the generation (or selection) of the answer denoted by A_i, corresponding to the question Q_i.

Our proposed pipeline to solve this task is schematically illustrated in Fig. 2. It consists of four components: (1) a scene graph construction module, which extracts objects and relation proposals from the video using pretrained neural network models, building a scene graph for every (temporally-sampled) video frame, (2) an intra-frame reasoning module, which conducts node-level and edge-level graph reasoning, producing compact feature representations for each scene graph, (3) an inter-frame information aggregation module, that aggregates these features within a temporal sliding window to produce a visual memory for each frame’s scene graph (at the center frame in that window), and (4) a semantics-controlled Transformer reasoning module, which performs multi-modal reasoning and language modelling based on a semantic controller. In this module, we also use a newly-proposed shuffle-augmented co-attention to enable head interactions in order to boost performance. Below, we describe in detail each of these modules.

Scene Graph Representation of Video

Our approach to generate scene graphs for the video frames is loosely similar to the ones adopted in recent works such as (Pan et al. 2020; Herzig et al. 2019; Wang and Gupta 2018), and has three components: (a) object detection, (b) visual-relation detection, and (c) region of interest (ROI) recrop on union bounding boxes. For (a), we train a Faster R-CNN model (Ren et al. 2015) on the Visual Genome dataset (Krishna et al. 2017) using the MMDetection repository (Chen et al. 2019). For a video frame I, this Faster-RCNN model produces: \( F_I, B_I, S_I = \text{RCNN}(I) \), where \( F_I \in \mathbb{R}^{N_x \times d_o} \) denotes the \( d_o \)-dimensional object features, \( B_I \in \mathbb{R}^{N_x \times 4} \) are the object bounding boxes, and \( S_I \) is a list of semantic labels associated with each bounding box. The pair \( (F_I, S_I) \) forms the nodes of our scene graph. Next, to find the graph edges, we train a relation model on the VG200 dataset (Krishna et al. 2017), which contains 150 objects and 50 predicates, and apply this learned model on the frames from the given video. The output of this model is a set of \( \langle S, P, O \rangle \) triplets per frame, where \( S, P, \) and \( O \) represent the subject, predicate, and object, respectively. We keep the \( \langle S, O \rangle \) pairs as relation proposals and discard the original predicate semantics, as the relation predicates of the model trained on VG200 are limited and fixed. Instead, we let our
reasoning model learn implicit relation semantics during our end-to-end training. For the detected \( (S, O) \) pairs, we regard the union box of the bounding boxes for \( S \) and \( O \) as the predicate region of interest. Next, we apply the ROI-align operator (Ren et al. 2015) on the last layer of the backbone network using this union box and make the resultant feature an extra node in the scene graph.

**Intra-Frame Reasoning**

Representing videos directly as sequences of scene graphs leads to a complex graph reasoning problem that can be computationally challenging. To avoid this issue, we propose to hierarchically reduce this complexity by embedding these graphs into learned representation spaces. Specifically, we propose an intra-frame reasoning scheme that bifurcates a scene graph into two streams: (i) a visual scene graph that generates a representation summarizing the visual cues captured in the graph nodes, and (ii) a semantic scene graph that summarizes the graph edges. Formally, let us define a scene graph as \( G = \{ (x_i, e_{ij}, x_j) \mid x_i, x_j \in V, e_{ij} \in E \} \), where \( V \) denotes the set of nodes consisting of single objects and \( E \) is the set of edges consisting of relations linking two objects. The triplet \( (x_i, e_{ij}, x_j) \) indicates that the subject node \( x_i \) and the object node \( x_j \) are connected by the directed relation edge \( e_{ij} \). We denote by \( G_v \) and \( G_s \) the visual scene graph and the semantic scene graph respectively: the former is a graph attention network (Velickovic et al. 2018) which computes an attention coefficient for each edge and updates node features based on these coefficients; the latter is based on EdgeConv (Wang et al. 2019), which computes an attention coefficient for each edge and aggregates the edge features linked to it.

**Visual Scene Graph Reasoning:** For \( M \) node features \( X = \{ x_1, x_2, \ldots, x_M \} \) in a scene graph, multi-head self-attention (Vaswani et al. 2017) is first performed for each pair of linked nodes. In each head \( k \), for two linked nodes \( x_i \) and \( x_j \), the attention coefficient \( \alpha_{ij}^k \) indicating the importance of node \( j \) to node \( i \) is computed by

\[
\alpha_{ij}^k = \frac{\exp \left( \sigma \left( \Theta^k \left[ W_i^k x_i \right] \right) \right)}{\sum_{k \in N_i} \exp \left( \sigma \left( \Theta^k \left[ W_i^k x_i \right] \right) \right)}.
\]

where \( \| \) denotes feature concatenation, \( \sigma \) is a nonlinearity (Leaky ReLU), \( N_i \) indicates the neighboring graph nodes of object \( i \), \( W_i^k \in \mathbb{R}^{d_h \times d_i} \) is a (learned) weight matrix transforming the original features to a shared latent space, and \( \Theta^k \in \mathbb{R}^{d_h \times 2d_i} \) is the (learned) attention weight vector. Using the attention weights \( \alpha_{ij}^k \) and a set of learned weight matrices \( W_2^k \in \mathbb{R}^{d_h / K \times d_k} \), we update the node features as:

\[
x_i' = \left( \sum_{k=1}^K \alpha_{ij}^k W_2^k x_i \right).
\]

Outputs of the \( K \) heads are concatenated to produce \( x_i' \in \mathbb{R}^{d_h} \), which is used as input to the semantic graph network.

**Semantic Scene Graph Reasoning:** This sub-module captures higher-order semantics between nodes in the scene graph. To this end, EdgeConv (Wang et al. 2019), which is a multi-layer fully-connected network \( h_A \), is employed to generate edge features \( e_{ij} \) from its two connected node features \( (x_i', x_j') \): \( e_{ij} = h_A(x_i', x_j') \), where \( h_A : \mathbb{R}^{d_h \times d_h} \rightarrow \mathbb{R}^{d_h} \) is a nonlinear transformation with learnable parameters \( \Lambda \). We then obtain the output node features \( x_i' \) by aggregating features from the edges that are directed to the object node \( i \), i.e.,

\[
x_i' = \max_{j(i,j) \in E} e_{ij},
\]

where \( E \) denotes the set of edges directed to node \( i \). All object features inside the scene graph are updated by the above intra-frame feature aggregation.

**Memory Generation with Graph Pooling:** After conducting intra-frame reasoning to obtain higher-level features for each node, we pool the updated graph into a memory for further temporal aggregation. Since different frame-level scene graphs have different numbers of nodes and edges, we adopt graph average pooling (GAP) and graph max pooling (GMP) (Lee, Lee, and Kang 2019) to generate two graph memories and concatenate them to produce \( V^* \):

\[
V^* = \text{GAP}(X^*, E) \big| \big| \text{GMP}(X^*, E),
\]

where \( E \) denotes the scene graph connection structure, and \( X^* \) the \( M \) node features \( \{ x_1', x_2', \ldots, x_M' \} \) from (3).

**Inter-Frame Information Aggregation**

Apart from the spatial graph representations described above, there is a temporal continuity of visual cues in the video frames that needs to be captured as well. To this end, we propose an inter-frame aggregation scheme that operates on the spatial graph embeddings. Specifically, for a sequence of scene graph memories \( \{ v_1', v_2', \ldots, v_L' \} \) of length \( L \) produced using (4) on a sequence of \( L \) frames, we use temporal sliding windows of size \( \tau \) to update the graph memory of the center frame in each window by aggregating the graph memories of its neighboring frames in that window, both in the past and in the future. Let \( F \in \mathbb{R}^{2d_h \times d_h} \) denotes a matrix of graph embeddings within this window of length \( \tau \), then we perform window-level summarization over all frame memories within \( F \) as: \( \beta = \text{softmax}^\top (\Gamma^\top \text{tanh}(W_r F)) \), where \( W_r \in \mathbb{R}^{2d_h \times 2d_h} \) is a learned weight matrix, \( \Gamma \in \mathbb{R}^{2d_h} \) is a weight vector, and \( \beta \) denotes the attention weights. We then use \( \beta \) to update the memory \( v_\tau' \) of the center frame (in this window) by aggregating information across this window, as: \( v_\tau' = F \beta^\top \). Repeating this step for all sliding windows, we get the final visual graph memory \( V' = \{ v_1', v_2', \ldots, v_L' \} \) aggregating both spatial and temporal information. We also augment these visual features with their temporally-aligned audio embeddings \( \{ s_1, s_2, \ldots, s_L \} \) produced using an AudioSet VGGish network (Hershey et al. 2017).

**Semantics-Controlled Transformer Reasoning**

The above modules encode a video into a sequence of graph memories via reasoning on visual and semantic scene graphs. Besides encoding audio-visual information, we also need to encode the text information available in the AVSD task. For the sentence generation task, we propose to generate the answer autoregressively (Anderson et al. 2018;
Figure 3: depicts the sequential attention flow in our semantics-controlled Transformer. MHA stands for multi-head attention. FFN is short for feed-forward networks. The acronyms A, V, H, and Q stand for the answers, visual memory, caption/dialog history, and the question, respectively.

Hori et al. (2018), i.e., predict the next word in the answer from the vocabulary based on source sequences including the visual memory, query Q, caption C, the dialog history H = {(Q1, A1), ..., (Ql−1, Al−1)}, and the partially generated answer so far, denoted A^p (see Fig. 2 and Fig. 3). This sub-answer A^p forms the semantics that control the attention on the various modalities to generate the next word. As shown in Fig. 3, our semantics-controlled Transformer module consists of a graph encoder, a text encoder, and a multi-modal decoder. It takes in source sequences and outputs the probability distribution of the next token for all tokens in the vocabulary. We detail the steps in this module next.

**Encoder:** We first use Transformers to embed all text sources (H, C, Q, A, A^p) using token and positional embeddings, generating feature matrices e_h, e_c, e_q, and e_a, each of the same feature dimensionality d_h. We also use a single-layer fully-connected network to transfer the audio-augmented visual memories in V^c to d_h-dimensional features e_v that match the dimension of the text sources. Next, for the answer generation task, the input sub-answer (generated so far) e_a is encoded with a Transformer consisting of multi-head self-attention to get hidden representations h_enc:

\[ h_{enc}^a = \text{FFN}^a(\text{Attention}(W^a_{e_a} h_{enc}^a, W^a_{e_a} h_{enc}),) \]  \tag{5}  

where W^a_{e_a}, W^a_{e_v} are weight matrices for query, key, and value respectively (Vaswani et al. 2017), FFN^a is a feed-forward module consisting of two fully-connected layers with ReLU in between. The Attention function is defined as in (Vaswani et al. 2017):

\[
\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad \tag{6}
\]

with a scaling factor \(\sqrt{d_k}\) that maintains the order of magnitude in features, and Q, K, V represent the query, key, and value triplets as described in (Vaswani et al. 2017). After encoding the input sub-answer, we conduct co-attention in turn for each of the other text and visual embeddings e_j, where j \in \{v, c, h, q\}, with a similar Transformer architecture. That is, the encoding h_{enc}^j for a given embedding type e_j is obtained by using the encoding h_{enc}^{j-1} for the previous embedding type e_{j-1} as query (Fig. 3):

\[
h_{enc}^j = \text{FFN}^j(\text{Attention}(W^j h_{enc}^{j-1}, W^j e_j, W^j V)). \quad \tag{7}
\]

In our implementation, the embeddings for history and caption are concatenated as e_{c+h} = [e_c || e_h], Processing occurs in the following order: starting from h_{enc}^a, we compute h_{enc}^{c+h}, then h_{enc}^{c+h}, and later h_{enc}. Finally, we get a feature vector h_{enc}^* that fuses all the information from the text and audio-visual sources by concatenating these multi-modal features.

**Multi-head Shuffled Transformer:** In this paper, we also propose to utilize head shuffling to further improve the performance of the Transformer structure as shown in Fig. 4. In the original Transformer (Vaswani et al. 2017), the feature vectors of all heads are concatenated before being fed into the last fully-connected layer. Thus, there is no interaction between those heads from the start to the end. To overcome this drawback, we propose to shuffle these head vectors before feeding them into fully-connected layers and concatenate them in a late fusion style. This scheme is similar to ShuffleNet (Zhang et al. 2018), with the key difference that here we conduct shuffling between different heads within the multi-head attention, while in ShuffleNet the shuffling is across channels. Our empirical results show that our shuffling operation results in better generalization of the model.

**Decoder:** For the generation setting, with the final encoded feature h_{enc}^*, we use a feed-forward network with softmax to predict the next token probability distribution P over all tokens in the vocabulary V; i.e., P = softmax(FFN(h_{enc}^*)). In the testing stage, we conduct beam search with b beams to generate an answer sentence.

**Loss Function:** Let \(\mathcal{P}\) denote the collection of all next-token probability distributions \(P_j \in \mathbb{R}^{\mid V\mid}, j = 1, \ldots, N\) for batch size N, and let \(\mathcal{G}\) be the collection of respective distributions \(G_j\) for the ground truth answer tokens. For the generation setting, we apply label smoothing (Müller, Kornblith, and Hinton 2019) to account for the sparsity of the token distributions, leading to \(\mathcal{G}\). We use the cross-entropy (CE) loss between the predicted and the smoothed ground truth distributions to train our model end-to-end:

\[
\mathcal{L} = \text{CE}(\mathcal{P} | \mathcal{G}) = -\frac{1}{N} \sum_{j=1}^{N} \sum_{u \in V} \hat{G}_j(u) \log P_j(u). \quad \tag{8}
\]

For the retrieval setting, we first concatenate the feature embeddings of the query and the various input modalities obtained from the Encoder module of our network (e_h, e_c, e_q, e_v). Next, the candidate answers are embedded into this joint space using LSTMs, and a dot product is taken between the concatenated inputs and embeddings of each of the answer candidates. We then train this model with the binary cross-entropy loss.
Experiments

In this section, we detail our experimental setup, datasets, and evaluation protocols, before furnishing our results.

**Dataset and Evaluation:** We use the audio-visual scene-aware dialog (AVSD) dataset (Alamri et al. 2019) for our experiments, which is the benchmark dataset for this task. This dataset emulates a real-world human-human natural conversation scenario about an audio-visual clip. See (Alamri et al. 2019) for details of this task and the dataset. We evaluate on two variants of this dataset corresponding to annotations available for the DSTC-7 and DSTC-8 challenges, consisting of 7,659, 1,787, 1,710, and 1,710 dialogs for training, validation, DSTC-7 testing, and DSTC-8 testing, respectively for the answer generation task. The quality of the generated answers is evaluated using the standard MS COCO evaluation metrics (Chen et al. 2015), such as BLEU, METEOR, ROUGE-L, and CIDEr. Apart from the answer generation task (Hori et al. 2018), we also report experiments on the answer selection task, described in (Alamri et al. 2019) using their annotations and ground truth answers. This task requires selecting the answer to a question from a set of 100 answers. Specifically, in this task, an algorithm is to present a ranking over a set of 100 provided answers, with ideally the correct answer ranked as the first. The evaluation is then based on the mean retrieval rank over the test set.

**Data Processing:** We follow (Le et al. 2019) to perform text preprocessing which include lowercasing, tokenization, and building a vocabulary by only selecting tokens that occur at least five times. Thus, we use a vocabulary with 3,254 words, both for the generation and retrieval tasks.

**Feature Extraction:** Motivated by (Anderson et al. 2018), we train a detector on Visual Genome with 1601 classes and 401 attributes, which incorporates a “background” label and a “no-attribute” label. We use ResNet-101 as the neural backbone with a multiscale feature pyramid network. We further use fine-grained ROI-alignment instead of ROI-pooling for better feature representation. We extract the 1024-D features for the 36 highest scoring regions, their class labels, and attributes. After extracting the region features, we apply a pretrained relationship detector (Zhang et al. 2019) to find visually-related regions. We calculate the minimal bounding box which covers two visually-related regions and perform ROI-alignment to get compact representations for relationship regions. In order to incorporate audio into the STSGR framework, we extract AudioSet VGG-ish features (Hershey et al. 2017) from the audio stream for every video. These are 128-D features obtained from the AudioSet VGG-ish CNN, pretrained on 0.96s Mel Spectrogram patches on the AudioSet data (Gemmeke et al. 2017).

**Model Training:** We set our Transformer hyperparameters following (Vaswani et al. 2017). The feature dimension is 512, while the inner-layer dimension of the feed-forward network is set to 2048. For multi-head attention, we maintain $h = 8$ parallel attention heads and apply shuffling to boost performance. For the semantic labels, we build a 300-D embedding layer for the 1651 words in the vocabulary (which is available with the dataset), and initialize the embeddings using GloVe word vectors (Pennington, Socher, and Manning 2014). For semantic labels consisting of more than one word, we use the average word embedding as the label embedding. Our model is trained on one Nvidia Titan XP GPU with Adam optimizer (Kingma and Ba 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$. The batch size is set to 16 and we adopt the warm-up strategy as suggested in (Vaswani et al. 2017) for learning rate adjustment with about 10,000 steps.

**Baselines:** We consider the following four baselines on the generation task: (i) Baseline (Hori et al. 2019), (ii) Multi-modal Attention (Hori et al. 2019), that uses attention over concatenated features, (iii) Simple (Schwartz et al. 2019) that uses factor-graph attention on the modalities, and (iv) MTN (Le et al. 2019) that applies self-attention and co-attention to aggregate multi-modal information. For the retrieval task, we compare our method against the state-of-the-art method of (Alamri et al. 2019) on the DSTC-7 split.

**Ablation Study:** To understand the importance of each component in our model, Table 1 details an ablation study. We analyze several key components: (i) shuffling in the Transformer structure, (ii) visual and semantic graph, (iii) ROI Recrop on the union bounding boxes, and (iv) temporal aggregation. From the table, we see that Graph Attention Network (GAT), which is used to produce the visual scene graph, is important to aggregate information from neighboring nodes (e.g., improving CIDEr from 1.125 to 1.265), while EdgeConv, used in the semantic graph, offers some improvement (e.g., CIDEr from 1.244 to 1.265). Moreover, the use of shuffling in the multi-head Transformer architecture boosts the performance significantly (from 1.208 to 1.265 for CIDEr). We can also conclude that union bounding boxes, semantic labels, and inter-frame aggregation contribute to stabilize the generation performance. Overall, by adopting all these key components, the full model outperforms all the ablations. From Tables 1 and 3, we notice that incorporation of audio helps improve the performance of our model. For instance, on the retrieval setting we observe that incorporating audio lowers the Mean-Retrieval Rank noticeably down to 4.08 from 4.33 for the full model and to 5.91 from 6.54 when no language context is available.

**Comparisons to the State of the Art:** In Table 2, we compare STSGR against baseline methods on various quality metrics based on ground-truth answers. As is clear, our approach achieves better performance against all the baselines. The performance on the answer selection task (mean retrieval rank) is provided in Table 3, demonstrating clearly state-of-the-art results against the baseline in (Alamri et al. 2019). We also show that including audio into the STSGR representation helps improve the mean retrieval rank.

![Table 1: Ablation study using AVSD@DSTC7 dataset.](https://sites.google.com/dstc.community/dstc8/home)
Table 2: Comparisons of our method against the state of the art on the AVSD test splits for DSTC7 and DSTC8.

<table>
<thead>
<tr>
<th>Method</th>
<th>AVSD@DSTC7</th>
<th>AVSD@DSTC8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (Hori et al. 2019)</td>
<td>0.075</td>
<td>0.110</td>
</tr>
<tr>
<td>Multi-modal Attention (Hori et al. 2019)</td>
<td>0.078</td>
<td>0.113</td>
</tr>
<tr>
<td>Simple (Schwartz, Schwing, and Hazan 2019)</td>
<td>0.091</td>
<td>0.125</td>
</tr>
<tr>
<td>MTN (Le et al. 2019)</td>
<td>0.128</td>
<td>0.162</td>
</tr>
<tr>
<td>Ours</td>
<td>0.133</td>
<td>0.165</td>
</tr>
</tbody>
</table>

Table 3: State-of-the-art comparisons on answer selection as measured by Mean Retrieval Rank (lower the better).

<table>
<thead>
<tr>
<th>Method</th>
<th>Full model</th>
<th>w/o caption</th>
<th>w/o cap.</th>
<th>diag. Hist.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alamri et al. (2019)</td>
<td>5.88</td>
<td>N/A</td>
<td>7.41</td>
<td></td>
</tr>
<tr>
<td>Hori et al. (2019)</td>
<td>5.60</td>
<td>N/A</td>
<td>7.23</td>
<td></td>
</tr>
<tr>
<td>MTN w/o audio</td>
<td>4.51</td>
<td>4.90</td>
<td>6.85</td>
<td></td>
</tr>
<tr>
<td>MTN w/o audio</td>
<td>4.29</td>
<td>4.78</td>
<td>6.46</td>
<td></td>
</tr>
<tr>
<td>STSGR</td>
<td>4.33</td>
<td>4.67</td>
<td>6.54</td>
<td></td>
</tr>
<tr>
<td>STSGR w/o audio</td>
<td>4.08</td>
<td>4.55</td>
<td>5.91</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Comparison of visual representations on DSTC7.

<table>
<thead>
<tr>
<th>Method</th>
<th>Feature</th>
<th>B4</th>
<th>M</th>
<th>K</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>i3d</td>
<td>0.091</td>
<td>0.125</td>
<td>0.301</td>
<td>0.833</td>
</tr>
<tr>
<td>MTN</td>
<td>VGG</td>
<td>0.095</td>
<td>0.126</td>
<td>0.309</td>
<td>0.919</td>
</tr>
<tr>
<td>MTN</td>
<td>i3d</td>
<td>0.118</td>
<td>0.160</td>
<td>0.348</td>
<td>1.151</td>
</tr>
<tr>
<td>STSGR (Ours)</td>
<td>N/A</td>
<td>0.121</td>
<td>0.152</td>
<td>0.353</td>
<td>1.277</td>
</tr>
<tr>
<td>STSGR (Ours)</td>
<td>Scene Graphs</td>
<td>0.133</td>
<td>0.165</td>
<td>0.361</td>
<td>1.265</td>
</tr>
</tbody>
</table>

Qualitative Results and Discussion: In Fig. 5, we provide two qualitative results from our STSGR model. For the first case, our model consistently detects the woman in the frames and finds that she maintains many connections with other objects inside the scene throughout the whole video, thus our model makes/selects the correct answer with high confidence. For the second case, the clutter background poses a challenge to our model. However, STSGR can still generate/rank the correct answer in top-2. In general, we find that STSGR can answer spatial and temporal questions very well. This is quantitatively evidenced by observing that both STSGR and MTN (Le et al. 2019) use similar backends, they differ in the input representations (I3D in (Le et al. 2019) vs. scene graphs in ours), and our model outperforms MTN noticeably (1.272 vs 1.249 on CIDEr, Table 2), substantiating the importance of our STSGR representation. In Table 4, we further compare STSGR representation against other visual representations (I3D, VGG) and different methods (Simple (Schwartz, Schwing, and Hazan 2019), MTN (Le et al. 2019) on the generation task, and demonstrate that our proposed scene graph representation by itself is a better way to characterize visual content.

Conclusions

We presented a novel hierarchical graph representation learning and Transformer reasoning framework for the problem of audio-visual scene-aware dialog. Specifically, our model generates object, frame, and video-level representations that are systematically integrated to produce visual memories, which are sequentially fused to the encodings of other modalities (dialog history, etc.) conditioned on the input question using a multi-head shuffled Transformer. Experiments demonstrate the benefits of our framework for both generation/selection tasks on the AVSD benchmark. Going forward, we plan to explore the use of richer text embeddings (such as GPT (Radford et al. 2019) and BERT (Devlin et al. 2019)) within our framework.

Ethical Impact: Our proposed STSGR framework presents a step towards improved modelling of multimodal data and we believe it could be a valuable addition to artificial intelligence research. However, STSGR does not explicitly account for biases in the training set, and thus may result in the model capturing social prejudices.

Acknowledgements: Shijie Geng, Peng Gao, and Moitreya Chatterjee worked on this project during MERL internships.
References


Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. ICLR.


