Storytelling from an Image Stream Using Scene Graphs

Ruize Wang,[1] Zhongyu Wei,[2,4*] Piji Li,[5] Qi Zhang,[3] Xuanjing Huang[3]
[1]Academy for Engineering and Technology, Fudan University, China
[2]School of Data Science, Fudan University, China
[3]School of Computer Science, Fudan University, China
[4]Research Institute of Intelligent and Complex Systems, Fudan University, China
[5]Tencent AI Lab, China
{rzwang18, zywei, qz, xjhuang}@fudan.edu.cn, lipiji.pz@gmail.com

Abstract

Visual storytelling aims at generating a story from an image stream. Most existing methods tend to represent images directly with the extracted high-level features, which is not intuitive and difficult to interpret. We argue that translating each image into a graph-based semantic representation, i.e., scene graph, which explicitly encodes the objects and relationships detected within image, would benefit representing and describing images. To this end, we propose a novel graph-based architecture for visual storytelling by modeling the two-level relationships on scene graphs. In particular, on the within-image level, we employ a Graph Convolution Network (GCN) to enrich local fine-grained region representations of objects on scene graphs. To further model the interaction among images, on the cross-images level, a Temporal Convolution Network (TCN) is utilized to refine the region representations along the temporal dimension. Then the relation-aware representations are fed into the Gated Recurrent Unit (GRU) with attention mechanism for story generation. Experiments are conducted on the public visual storytelling dataset. Automatic and human evaluation results indicate that our method achieves state-of-the-art.

1 Introduction

For most people, showing them images and ask them to compose a reasonable story about the images is not a difficult task. Though the recent advances in deep neural networks have achieved encouraging results, it is still non-trivial for the machine to summarize the meanings from images and generate a narrative story. Recently, visual storytelling has attracted increasing attention from the areas of both Computer Vision (CV) and Natural Language Processing (NLP) (Huang et al. 2016; Yu, Bansal, and Berg 2017; Wang et al. 2018a; Huang et al. 2019). Different from image captioning (Karpathy and Fei-Fei 2015; Vinyals et al. 2017; Yao et al. 2018; Fan et al. 2019) which aims at generating a literal description for a single image, visual storytelling is more challenging, which further investigates machine's capabilities of understanding a sequence of images and generate a coherent story with multiple sentences.

Existing methods (Huang et al. 2016; Wang et al. 2018a; Huang et al. 2019) for visual storytelling employ encoder-decoder structure to translate images to sentences directly, with CNN-based models for visual feature extraction and RNN-based models for text generation. However, it is not intuitive to represent all the visual information of the images with an abstract high-level feature, and this also hurts the interpretability and reasoning ability of the model. Recall that when we humans telling stories for an image sequence, we will recognize the objects in each image, reason about their visual relationships, and then abstract the content into a scene. Next, we will observe the images in order and reason the relationship among images.

Taking this idea as motivation, we propose a novel graph-based architecture named SGVST for visual storytelling, which first translates each image into a graph-based semantic representation, i.e., scene graph, and then models the relationship on within-image level and cross-images level, as shown in Figure 1. Specifically, inspired by the success of scene graph generation (Xu et al. 2017; Li et al. 2018;

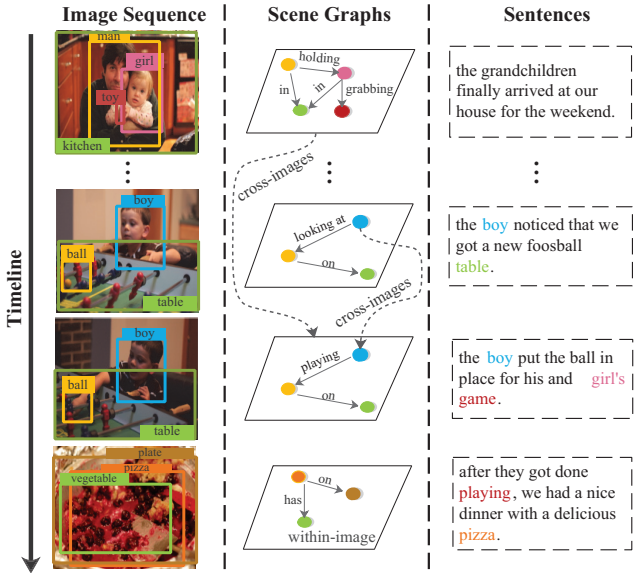

Figure 1: A scene graph based example for visual storytelling from VIST dataset. The story presented is from a human annotator. (Best viewed in color)

*Corresponding author

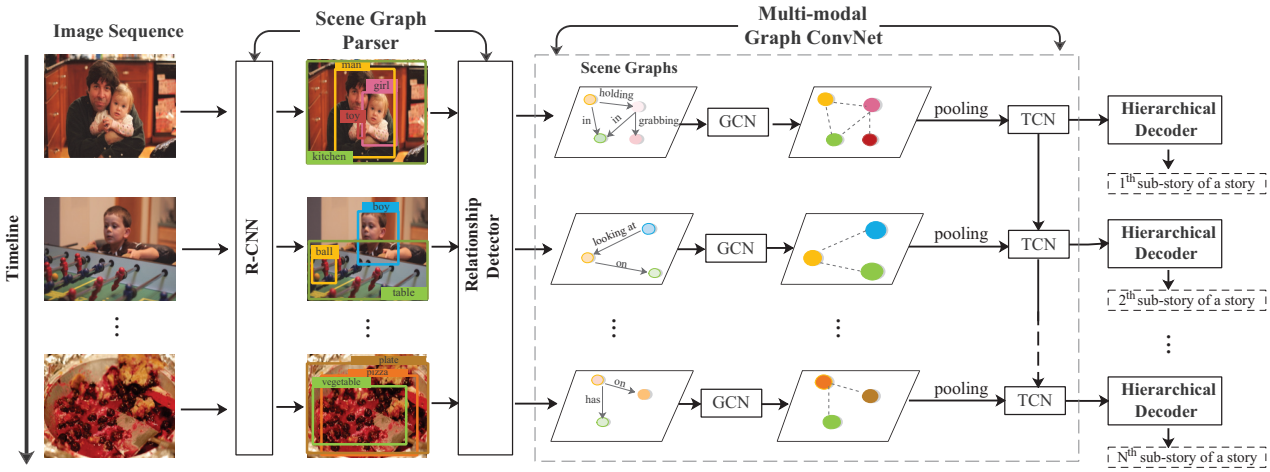

Figure 2: An overview of our SGVST model (better viewed in color).

Zellers et al. 2018), a scene graph parser, consisting of Faster R-CNN (Ren et al. 2015) and relationship detector, is firstly implemented to parse images into scene graphs. In each scene graph, vertexes represent different regions and directed edges denote relationships between them, which can be represented as tuples <*subject-predicate-object*>, e.g., <*man-holding-girl*>, explicitly encoding the objects and relationships detected within an image. Then for processing the scene graphs to enrich region representations, we employ Graph Convolution Network (GCN) which passes the information along graph edges. After processing the local region representations for each image, we further utilize Temporal Convolution Network (TCN) (Bai, Kolter, and Koltun 2018) to process the region representations along the temporal dimension, which models relationships on cross-images level. To this end, the relation-aware representations are integrated with the information on both within-image level and cross-images level. In order to make full use of image information, we use a bidirectional-GRU (Chung et al. 2014) (biGRU) to encode the feature maps obtained from Faster R-CNN as high-level visual features, and then fuse them with the relation-aware representations to get new representations. Finally, the obtained new relation-aware representations are fed into the hierarchical decoder to conduct the story generation.

The main contributions can be summarized as follows:

- We first propose to translate images into graph-based semantic representations called the scene graphs to benefit representing images and high-quality story generation.
- We propose a framework based on scene graphs to realize enriching fine-grained representations by modeling the visual relationships through GCN on the within-image level and through TCN on the cross-images level.
- Extensive experiments on the VIST dataset (Huang et al. 2016) demonstrate that our method achieves the state-of-the-art performance.

2 Method

The overall architecture of our proposed model is shown in Figure 2. Here we have an image stream $I = \{I_1, \ldots, I_N\}$, we aim to output a story $y = \{y_1, \ldots, y_N\}$, where N is the number of images in the image stream and sentence $y_n = \{w_1, \ldots, w_T\}$ consisting of T words in the vocabulary $\mathbb{V}_s$ of all output words. We argue that modeling relationships on within-image and cross-images levels would help for understanding and describing images. To this end, we propose a graph-based architecture. First, scene graphs $G = \{G_1, \ldots, G_N\}$ are first generated by a pre-trained scene graph parser, where the vertex (object) represents each region and the edge denotes the visual relationship between them. Then the scene graphs are passed through Multi-modal Graph ConvNet to obtain the relation-aware representations $\bar{v} = \{\bar{v}_1, \ldots, \bar{v}_N\}$, which integrate both within-image and cross-images levels information. In the story generation state, we feed the relation-aware representations $\bar{v}$ into a hierarchical decoder to generate the story. Each of these modules will be described in details in the following sections.

2.1 Scene Graph Parser

Scene graph parser is proposed to parse an image to a scene graph. Thanks to the recent advances in visual relationship detection (Xu et al. 2017; Zellers et al. 2018), detecting the relationship can be simplified as a semantic relation classification task on visual relationship datasets. Formally, a scene graph is a tuple $G_n = (V_n, E_n)$, where $n \in N$ denotes n-th scene graph for n-th image I_n, $V_n = \{v_{n,1}, \ldots, v_{n,K}\}$ is a set of K detected objects with each region representation $v_{n,i} \in \mathbb{R}^{D_V}$, and E_n is a set of directed edges of the form $(v_{n,i}, r_{n,(i,j)}, v_{n,j})$, assigning two directional edges from $v_{n,i}$ to $r_{n,(i,j)}$ and from $r_{n,(i,j)}$ to $v_{n,j}$, where $r_{n,(i,j)}$ denotes a relationship categories (labels). The details of parsing an image to scene graph are given as follows.

Object Detector. We use pre-trained Faster-RCNN (Ren et al. 2015) as the object detector to produce and classify objects in an image I_n. To this end, for each image, we get the set of region representations $V_n = \{v_{n,1}, \ldots, v_{n,K}\}$ and labels $O = \{o_{n,1}, \ldots, o_{n,K}\}$ of detected objects, where each $v_{n,i} \in \mathbb{R}^{D_v}$ denotes the D_v dimension feature, and each $o_{n,i} \in C$ denotes object categories (labels).

Relationship Detector. We use the LSTM-based model proposed by Zellers et al. (2018) as our relationship detector to classify relationships between objects. Then we follow them to train our relationship detector on Visual Genome dataset (Krishna et al. 2017).

In subsequent experiments, the parameters of scene graph parser will be fixed. We directly employ the pre-trained scene graph parser to construct the corresponding scene graph $G_n = (V_n, E_n)$ for image I_n, where a directional edge from the subject region to object region is established and the relation class with maximum probability is regarded as the label of this edge. As a first stage of processing, we apply a embedding layer on each region representation $v_{n,i}$ of object and categorical label $r_{n,(i,j)}$ of edge of the graph, converting them to $v_{n,i} \in \mathbb{R}^{D_v}$ and a dense vector $v_r \in \mathbb{R}^{D_r}$, respectively.

2.2 Multi-modal Graph ConvNet

Inspired by the recent advances in spatial Graph Convolution Network (GCN), we can enrich the fine-grained region-level features by modeling the relations on scene graphs, allowing our model to explicitly reason about objects and their relationships. Furthermore, we employ Temporal Convolution Network (TCN) (Bai, Kolter, and Koltun 2018) to model temporal interaction within an image stream. To this end, we get the relation-aware representations which integrated with both within-image and cross-images levels information.

Graph Convolution Network. For enriching each region representation, we follow the way similar to Johnson, Gupta, and Fei-Fei (2018), aggregating the information of its local neighbors through a graph convolution layer.

For enriching each node by aggregating the information of its local neighbors through a graph convolution layer, we follow the way similar as Johnson, Gupta, and Fei-Fei (2018). Given an input graph with vectors of each node and edge, it computes new vectors for each node and edge. Each graph convolution layer propagates information along edges of the graph.

Formally, given input vectors $v_{n,i} \in \mathbb{R}^{D_v}$, $v_r \in \mathbb{R}^{D_r}$ for all objects and edges, we compute output vectors $v^{'}_{n,i}$, $v^{'}_r \in R^{D_{out}}$ for all nodes and edges using three functions g_s , g_p and g_o, which take as input the triple of vectors $(v_{n,i}, r_{n,(i,j)}, v_{n,j})$ for an edge and output new vectors for objects and edges.

For the output edges vectors $v^{'}_r$, we simply compute via $v^{'}_r$ = $g_p(v_{n,i}, v_r, v_{n,j})$. Then the output object vectors $v^{'}_{n,i}$ depend on all features of objects which connected via edges. To this end, for each edge starting at $v_{n,i}$ we use g_s to compute a candidate vector, collecting all such candidates in the set $V^s_{n,i}$; we similarly use g_o to compute a set of candidate vectors V^o_i for all edges terminating at $v_{n,i}$ as follows:

$$\begin{aligned} V^s_{n,i} &= \{g_s\left(v_{n,i}, v_r, v_{n,j}\right)\} \\ V^o_{n,i} &= \{g_o\left(v_{n,j}, v_r, v_{n,i}\right)\} \end{aligned} \tag{1}$$

In our implementation, we concatenate its three input vectors as the input for functions g_s, g_p and g_o, and feed them to a MLP, and computes three output vectors for objects and edges. The output vector is then calculated as $v^{'}_{n,i} = h(V^s_{n,i} \cup V^o_{n,i})$ where h denotes an average pooling function after with a MLP layer which converts a set of vectors to a single output vector. After passing all scene graphs through GCN, the enriched region representations $v^{'}_{n,i}$ are integrated with the inherent visual relation information at object level.

Temporal Convolution Network. With the help of GCN, we enrich representation for each object which aggregates information across all objects and relationships in the graph. In order to capture the interaction among images, we now advance to the task of modeling temporal relationships among images. To this end, we use Temporal Convolution Network (TCN) (Bai, Kolter, and Koltun 2018) to process region representations along temporal dimension.

Notably, before using TCN, we calculate the mean-pooled region vectors over K object regions $\{\mathbf{v}^{'}_{n,i}\}^K_{i=1}$ via follows:

$$\bar{\mathbf{v}}_n = \frac{1}{K}\sum_{i=1}^{K} \mathbf{v}^{'}_{n,i} \tag{2}$$

Specifically, TCN employs dilated causal convolutions that enable an exponentially large receptive field. For a 1-D sequence input $\{\bar{\mathbf{v}}_n\}^N_{n=1} \in \mathbb{R}^{D_v}$ and fully-convolutional network (FCN) (Long, Shelhamer, and Darrell 2015) as filter $f : \{0, \ldots, k-1\} \rightarrow \mathbb{R}$, the dilated convolution operation F on each $\bar{\mathbf{v}}_n$ is defined as

$$F(\bar{\mathbf{v}}_n) = \sum_{i=0}^{k-1} f(i) \cdot \bar{\mathbf{v}}_{n-d\cdot i} \tag{3}$$

where d denotes the dilation factor, k denotes the filter size, and $\bar{\mathbf{v}}_{n-d\cdot i}$ denotes the $\bar{\mathbf{v}}_n$ pointing to $d \cdot i$-th dilated convolution layer. Then with the help of a residual structure (He et al. 2016), the region representations can be updated via follows:

$$\bar{\mathbf{v}}_n = \mathrm{ReLU}(\bar{\mathbf{v}}_n + F(\bar{\mathbf{v}}_n)) \tag{4}$$

where $\bar{\mathbf{v}}_n$ denotes n-th relation-aware representations. After modeling interaction among images through TCN, we get the relation-aware representations which integrated with the information on both within-image and cross-images levels.

High-level Encoder. Although the scene graph abstracts away most of the informative characteristics of an image, there is still some image information lost in the process. In order to make full use of image information, we use a bidirectional gated recurrent unit (biGRU) to encode the feature

maps obtained from the previous Faster R-CNN as high-level visual features, and then fuse with the relation-aware representations to get new relation-aware representations.

At this stage, the high-level visual vectors h_n^v can be calculated as:

$$\begin{aligned} \overrightarrow{h_{n,t}} &= \overrightarrow{\text{GRU}}(f_n, \overrightarrow{h_{n,t-1}}) \\ \overleftarrow{h_{n,t}} &= \overleftarrow{\text{GRU}}(f_n, \overleftarrow{h_{n,t+1}}) \\ h_n^v &= \text{ReLU}([\overleftarrow{h_n}; \overrightarrow{h_n}] + f_n) \end{aligned} \tag{5}$$

where $[\cdot]$ indicates concatenation, $\overrightarrow{h_{n,t}}$ is the forward hidden state at time step t of n-th high-level feature f_n, while the $\overleftarrow{h_{n,t}}$ is the backward one.

At the end of encoding state, we fuse relation-aware representations with high-level visual vectors to update relation-aware representations. Formally,

$$\begin{aligned} \mathbf{v}_{mul} &= \text{ReLU}(W_{mul}(\bar{\mathbf{v}}_n \odot h_n^v)) \\ \mathbf{v}_{minus} &= \text{ReLU}(W_{minus}(\bar{\mathbf{v}}_n - h_n^v)) \\ \bar{\mathbf{v}}_n &= \text{ReLU}(W_{final}[v_{mul}, v_{minus}]) \end{aligned} \tag{6}$$

where $[\cdot]$ indicates concatenation, W_{mul}, W_{minus}, W_{final} are the projection matrix, $\odot$ denotes Hadamard product.

2.3 Hierarchical Story Decoder

We devise our hierarchical story decoder by injecting all of the relation-aware representations $\bar{\mathbf{v}}$ into a two-layer GRU with attention mechanism. Specifically, we concatenate relation-aware representations $\bar{\mathbf{v}}_n$ with the previous word token $w_{n,t-1}$ and the previous output $h^2_{n,t-1}$ of the second-layer GRU, as the input of the first layer GRU. Formally, the output of first layer GRU is generated through this process:

$$h^1_{n,t} = \text{GRU}(h^1_{n,t-1}, [W_s w_{n,t-1}, \bar{\mathbf{v}}_n, h^2_{n,t-1}]) \tag{7}$$

where $[\cdot]$ indicates concatenation, W_s is the projection matrix for the input word. Then we use a traditional soft attention mechanism (Rocktäschel et al. 2015). Given the output $h^1_{n,t}$ of the first layer GRU, the attention mechanism will produce normalized attention weights a_{att} over all the relation-aware features via following:

$$\text{Z} = \tanh\left(\mathbf{W}_v \bar{\mathbf{v}}_n + \mathbf{W}_h \mathbf{h}^1_{n,t}\right) \tag{8}$$

$$a_{att} = \text{softmax}(\mathbf{W}_z \text{Z}) \tag{9}$$

where $\text{W}_v, \text{W}_h, \text{W}_z$ are the projection matrix, a_{tt} denotes the attention weights. Based on the above attention weights, the attended relation-aware representations $\hat{\mathbf{v}}_n$ as calculated as the weighted sum:

$$\hat{\mathbf{v}}_n = \bar{\mathbf{v}}_n a^T_{att} \tag{10}$$

At last, we concatenate the attended relation-aware representations $\hat{\mathbf{v}}_n$ with the output $h^1_{n,t}$ of first layer GRU, and then feed them into second layer GRU. Then we leverage $h^2_{n,t}$ to generate a next word w_t through a softmax layer. Formally, the generation process can be written as:

$$h^2_{n,t} = \text{GRU}\left(h^2_{n,t-1}, [w_{n,t-1}, \hat{\mathbf{v}}_n]\right) \tag{11}$$

$$p(w_{n,t}|w_{n,1:t-1}) = \text{softmax}\left(\text{MLP}(h^2_{n,t})\right) \tag{12}$$

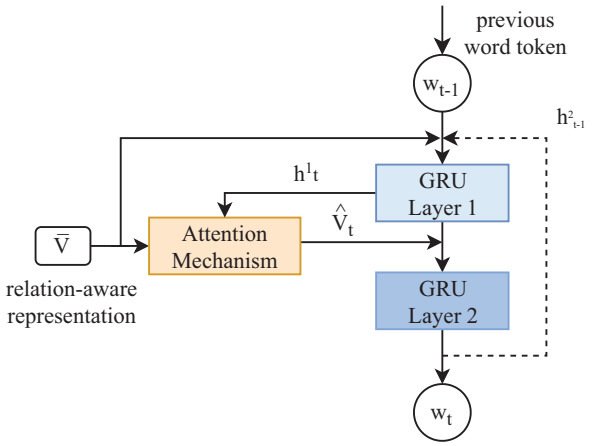

Figure 3: An overview of our hierarchical story decoder.

where $h^2_{n,t}$ denotes the t-th hidden state of second layer GRU of n-th hierarchical decoder. The output p is a probability distribution over the whole story vocabulary $\mathbb{V}_s$. Eventually, the final story y is the concatenation of the sub-stories $y_n = \{w_1, \ldots, w_T\}$ consisting of T words in $\mathbb{V}_s$.

2.4 Training and Inference

In the training stage, we fix the parameters of our pre-trained scene graph parser as described in Section 2.1, and other components of our model are trained and evaluated on VIST dataset for visual storytelling task. We define cross-entropy (MLE) loss for the training process, as shown in Equation 13:

$$L(\theta) = -\sum_{t=1}^{T} log\left(p_\theta(y^*_t|y^*_1, ..., y^*_{t-1})\right) \tag{13}$$

where θ is the parameters of our model; y^* is the ground-truth story and y^*_t denotes the t-th word in y^*. During training, our goal is minimizing L using stochastic gradient descent.

For inference in story generation, we adopt the beam search strategy to produce story with a beam size of 3.

3 Experimental Evaluation

3.1 Experimental Setup

Datasets. **VIST** (Huang et al. 2016) dataset includes 10,117 Flicker albums with 210,819 images. In our experiments, we follow the same split settings as (Huang et al. 2016; Yu, Bansal, and Berg 2017; Wang et al. 2018b). Thus, the samples have been split into three parts, 40,098 for training, 4,988 for validation and 5,050 for testing, respectively. Each sample (album) contains five images and a story with five sentences. We train and evaluate our models (except the scene graph parser) on VIST.

Visual Genome (VG) (Krishna et al. 2017) comprises 108,077 images annotated with scene graphs, which can be exploited to train the object detector and relationship detector. We follow the setting as Xu et al. (2017), containing

Table 1: Overall performance of story generation on VIST dataset for different models in terms of BLEU (B), METEOR (M), ROUGE-L (R-L), and CIDEr-D (C). $*$ directly optimized with RL rewards, e.g., the CIDEr Metric, † optimized with cross-entropy (MLE). **Bolded** numbers are the best performance in each category.

Methods	B-1	B-2	B-3	B-4	R-L	C	M
seq2seq† (Huang et al. 2016)	–	–	–	3.5	–	6.8	31.4
BARNN† (Liu et al. 2017)	–	–	–	–	–	–	33.3
h-attn-rank† (Yu, Bansal, and Berg 2017)	–	–	21.0	–	29.5	7.5	34.1
HPSR† (Wang et al. 2019)	61.9	37.8	21.5	12.2	**31.2**	8.0	34.4
AREL* (Wang et al. 2018b)	63.7	39.0	23.1	14.0	29.6	9.5	35.0
HSRL* (Huang et al. 2019)	-	-	-	12.3	30.8	**10.7**	35.2
SGVST w/o GCN or TCN†	62.8	38.4	22.8	13.9	29.6	8.5	35.1
SGVST w/o GCN†	63.1	39.0	23.3	14.1	29.8	8.8	35.2
SGVST w/o TCN†	**65.4**	39.8	23.5	14.2	29.6	9.3	35.4
SGVST w/ single-dec†	64.5	39.7	23.5	14.4	29.7	9.4	35.5
SGVST w/o high-level-enc†	64.9	40.0	23.6	14.5	29.8	9.6	35.6
SGVST†	65.1	**40.1**	**23.8**	**14.7**	29.9	9.8	**35.8**

150 object classes and 50 relation classes. The VG dataset is only used to train the relationship detector in our scene graph parser.

Automatic Metrics. We adopt four automatic metrics in our experiments: BLEU (Papineni et al. 2002), ROUGE-L (Lin and Och 2004), METEOR (Banerjee and Lavie 2005), and CIDEr-D (Vedantam, Lawrence Zitnick, and Parikh 2015).

3.2 Implementation Details

In the scene graph parser, we use Faster RCNN with a VGG backbone as our object detector and use MOTIFS (Zellers et al. 2018) as relationship detector. For each scene graph, we set the max number of objects as 10 and the max number of relationship as 20. The dimension of region feature for each object and the high-level feature of an image is 4096. In Multi-modal Graph ConvNet, we use a 5 layers GCN, whose the input and output dimension both as 512; for TCN, we set the dilation factor=5 and filter size=7; for high-level encoder, we use a bi-GRU with the hidden dimension of 512. We build a story vocabulary with a size of 9,837 words which contain those words appearing more than three times in the training set. All the parameters are initialized by a kaiming-normal distribution (He et al. 2015).

We set the batch size as 100 during the whole experiments. We use Adam (Kingma and Ba 2015) to optimize our models with the initial learning rate of 0.0004. We select the best model which achieves the highest METEOR score on the validation set. The reason is that METEOR is proved to correlate better with human judgment than CIDEr-D in the small references case and superior to BLEU@N and ROUGE all the time (Vedantam, Lawrence Zitnick, and Parikh 2015; Wang et al. 2018a).

3.3 Models for Comparison

We compare our proposed methods with several baselines for visual storytelling. Moreover, five variants of our method are provided to reveal the impact of each component. Each of these models will be described as follows.

seq2seq (Huang et al. 2016): This model is the ordinary seq2seq model, which encodes an image sequence by running an RNN, and decodes sentences with a RNN decoder.

BARNN (Liu et al. 2017): BARNN is a new-designed sGRU model, with attention on semmatic relation extracted from space space to enhance the textual coherence in story generation.

h-attn-rank (Yu, Bansal, and Berg 2017): h-attn-rank is a hierarchically-attentive RNN based model consisting of three RNN stages, i.e., encoding photo stage, photo selection stage and generation stage.

HPSR (Wang et al. 2019): HPSR is a model includes the hierarchical photo-scene encoder, decoder, and reconstructor.

AREL (Wang et al. 2018b): AREL is a model based on reinforcement learning. It takes a CNN-RNN architecture as the policy model for story generation, while the reward model aims to learn the reward function from human demonstrations.

HSRL (Huang et al. 2019): HSRL develops a hierarchically structured reinforcement learning approach, which propose to generate a local semantic concept for each image in the sequence and generate a sentence for each image using a semantic compositional network.

SGVST w/o GCN or TCN: This model is the basic baseline, which is ablated from our full model by removing GCN and TCN.

SGVST w/o GCN: To investigate the role of the GCN and its what effect it has for modeling the relationships between objects, in this baseline, we ablate our model by removing the GCN.

SGVST w/o TCN: To investigate the role of the TCN and its what effect it has for modeling the interaction among images, in this baseline, we ablate our model by removing the TCN.

SGVST w/ single-dec: Again, we ablate our model by replacing hierarchical decoder with single-layer GRU decoder.

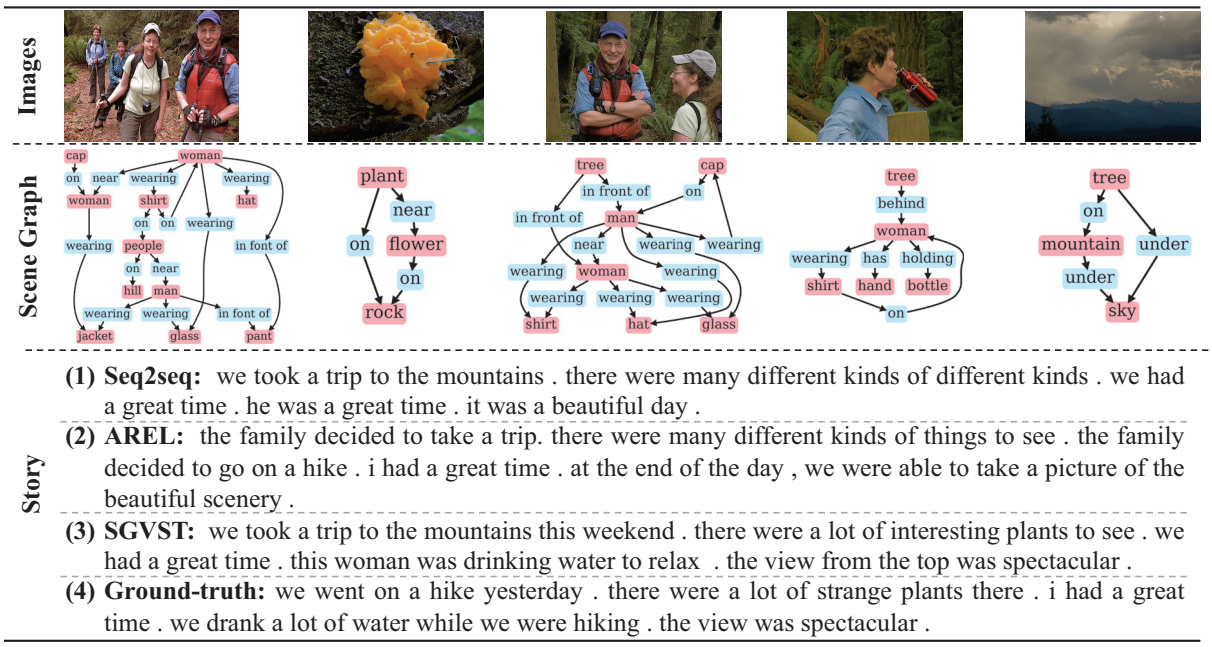

Figure 4: Qualitative example of different models with an image stream, scene graph, ground-truth story and generated story by three approaches, i.e., seq2seq, AREL and our SGVST.

SGVST w/o high-level-enc: Again, we ablate our model by removing high-level encoder.

SGVST: SGVST is the complete method in this paper.

3.4 Quantitative Results

Comparing with state-of-the-art. Table 1 shows the performances of different models on seven automatic evaluation metrics. Some works (Wang et al. 2018a; Modi and Parde 2019) have confirm that CIDEr do not correlate well with human evaluations in this task, but here we still adopt this metric for reference. Overall, the results indicate that our proposed *SGVST* model achieves superior performances over other state-of-the-art models optimized with MLE and RL, which directly demonstrates our graph-based model can help for story generation. In particular, the BLEU-1, BLEU-4 and METEOR scores of our *SGVST* makes the relative improvement over the best method optimized with cross-entropy loss by 3.2%, 2.5% and 1.4%, respectively, which is considered as significant progress on this dataset. It is worth noting that, our *SGVST* also outperforms state-of-the-art model optimized with RL rewards.

Comparing with ablations. As shown in Table 1, we conduct experiments on five ablations with our proposed model. Overall, we find that all our models achieve almost the same performance on ROUGE, which indicates ROUGE is not very suitable for evaluation in this task as shown in Wang et al. (2018b). In particular, (1) *SGVST w/o GCN* slightly outperforms our basic baseline *SGVST w/o GCN or TCN*. This demonstrates that only modeling the relationships among images is effective but not obvious. (2) *SGVST w/o TCN* significantly outperforms our basic baseline *SGVST w/o GCN or TCN*. This demonstrates that modeling the visual relationships between objects in each image can enhance the fine-grained region representations and help to describe images. (3) The performance of *SGVST* in BLEU@3-4, CIDER and METEOR is clearly better than *SGVST w/o TCN*. This indicates modeling the interaction among the images can refine the relation-aware representations on cross-images level. (4) *SGVST* makes obvious improvement over BLEU@1-2 comparing with *SGVST w/ single-dec*, which indicates that this two-layer GRU decoder with attention mechanism can help generate story in word (entity) level; (5) *SGVST w/o high-level-enc* achieves a comparable performance, which slightly loses compared with *SGVST*. This demonstrates from another aspect that our graph-based model has the ability to learn high-level information through reasoning the relationships.

3.5 Qualitative Results

Qualitative Examples. Figure 4 shows some examples with the an image stream, scene graphs, ground-truth story and generated story by three approaches, i.e., *seq2seq*, *AREL* and our *SGVST*, where the *seq2seq* (Huang et al. 2016) is implemented by us and *AREL* (Wang et al. 2018b) is trained and evaluated according to its publicly available code. From these examples, it is easy to find that the story generated by our *SGVST* is more coherent, informative and descriptive.

Human Evaluation. To better evaluate the qualities of the generated story, we conduct two kinds of human evaluation through Amazon Mechanical Turk (AMT). Specifically, we randomly select 150 stories, each evaluated by 3

Table 2: Human evaluation results. Workers on AMT rate the quality of the story by telling how much they Agree or Disagree with each question, on a scale of 1-5.

Methods	Focused	Coherent	Share	Human-like	Grounded	Detailed
seq2seq	2.30	2.33	2.12	2.22	2.30	2.30
AREL	3.51	3.53	3.37	3.43	3.31	3.39
SGVST	**3.97**	**4.01**	**3.91**	**3.99**	**4.02**	**4.07**
GT	4.37	4.40	4.21	4.38	4.32	4.39

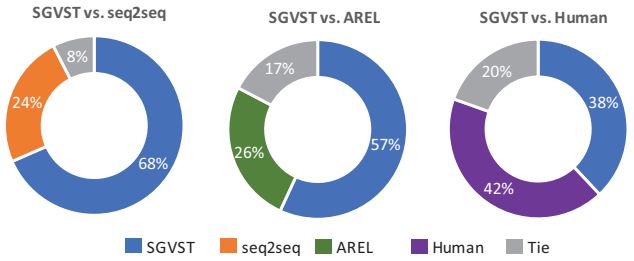

Figure 5: Pairwise comparison results, where the charts each comparing two methods in human evaluation. Each color represents the percentage of works who consider the story generated by the corresponding method is more human-like and descriptive. "Tie" in grey color indicates hard to tell.

crowd workers. **(1) Pairwise Comparison** In pairwise comparison, the workers are asked to compare two stories generated by corresponding methods and choose the one that more human-like and descriptive. Figure 5 shows the stories generated by our *SGVST* are significantly better than stories generated by other machines, and achieve competitive performance compared with human. **(2) Human Rating** For a more detailed comparison of different stories generated from different models, we conduct human rating survey corresponding to the following characteristics modified from Visual Storytelling Challenge (NAACL 2018): ① Focused: the story is focused, ② Coherent: the story is coherent, ③ Share: inclination to share, ④ Human-like: the story sounds like written by a human, ⑤ Grounded: the story is visually grounded, and ⑥ Detailed: the story is detailed. The workers are asked to rate the quality of the story by telling how much they Agree or Disagree with each question, on a scale of 1-5. The results are shown in Table 2. The scores reported show that our *SGVST* model outperforms in all six characteristics, which further proves the storied generated by our model are more informative and high-quality.

4 Related work

There are many works focus on vision-to-language, e.g., VQA (Fan et al. 2018a; 2018b) and image captioning. Some earlier works (Karpathy and Fei-Fei 2015; Vinyals et al. 2017) propose CNN-RNN frameworks for image captioning. Further, some works (Yao et al. 2018; Lu et al. 2018) explore visual relationship for image captioning. Different from image captioning, visual storytelling aims at generating a narrative story from an image stream. The pioneering work was done by Park and Kim (2015). Huang et al. (2016) introduces the first dataset (VIST) for visual storytelling task. Yu, Bansal, and Berg (2017) designs a hierarchically-attentive RNN structure. Wang et al. (2018a) propose a reinforcement learning framework with two discriminators. Due to the bias can be brought by the hand-coded evaluation metrics, Wang et al. (2018b) proposes an adversarial reward learning framework to uncover a reward function from human demonstrations. Wang et al. (2019) propose a model with a hierarchical photo-scene encoder and a re-constructor. Huang et al. (2019) develops a hierarchically reinforcement learning approach, which introduces a local semantic concept to model. However, these methods tend to represent images with high-level features, which is not intuitive and difficult to interpret.

Scene graphs present scenes as directed graphs, where vertexes represent objects and edges represent relationships between objects. Recently, scene graphs have been used for many tasks, e.g., image generation (Johnson, Gupta, and Fei-Fei 2018), image captioning (Yao et al. 2018; Yang et al. 2019) and image retrieval (Johnson et al. 2015). There are many works (Xu et al. 2017; Zellers et al. 2018) focus on scene graph parsing, which aims at producing structured graph representations of visual scenes. Inspired by the booming in scene graphs, we propose to encode images into graphs, which contains objects and corresponding visual relationships, and this eventually helps for story generation.

5 Conclusion

In this paper, we propose a novel graph-based method named SGVST for visual storytelling, which parses images to scene graphs, and models the relationships on scene graphs at two levels, i.e., within-image and cross-images levels. Extensive experiments demonstrate that our method achieves state-of-the-art, and the stories generated by our method are more informative and fluent. In the further, we would explore our method to other multi-modal tasks, e.g., video captioning.

Acknowledgment

This work is partially supported by National Natural Science Foundation of China (No. 61751201, No. 61702106) and Science and Technology Commission of Shanghai Municipality Grant (No.18DZ1201000, No.17JC1420200, No.16JC1420401).

References

Bai, S.; Kolter, J. Z.; and Koltun, V. 2018. An empirical evaluation of generic convolutional and recurrent networks

for sequence modeling. *arXiv:1803.01271*.

Banerjee, S., and Lavie, A. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop*, 65–72.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Fan, Z.; Wei, Z.; Li, P.; Lan, Y.; and Huang, X. 2018a. A question type driven framework to diversify visual question generation. In *IJCAI*, 4048–4054.

Fan, Z.; Wei, Z.; Wang, S.; Liu, Y.; and Huang, X.-J. 2018b. A reinforcement learning framework for natural question generation using bi-discriminators. In *COLING*, 1763–1774.

Fan, Z.; Wei, Z.; Wang, S.; and Huang, X.-J. 2019. Bridging by word: Image grounded vocabulary construction for visual captioning. In *ACL*, 6514–6524.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 1026–1034.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Huang, T.-H. K.; Ferraro, F.; Mostafazadeh, N.; Misra, I.; Agrawal, A.; Devlin, J.; Girshick, R.; He, X.; Kohli, P.; Batra, D.; et al. 2016. Visual storytelling. In *NAACL*, 1233–1239.

Huang, Q.; Gan, Z.; Celikyilmaz, A.; Wu, D.; Wang, J.; and He, X. 2019. Hierarchically structured reinforcement learning for topically coherent visual story generation. In *AAAI*, 8465–8472.

Johnson, J.; Krishna, R.; Stark, M.; Li, L.-J.; Shamma, D.; Bernstein, M.; and Fei-Fei, L. 2015. Image retrieval using scene graphs. In *CVPR*, 3668–3678.

Johnson, J.; Gupta, A.; and Fei-Fei, L. 2018. Image generation from scene graphs. In *CVPR*, 1219–1228.

Karpathy, A., and Fei-Fei, L. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 3128–3137.

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

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV* 123(1):32–73.

Li, Y.; Ouyang, W.; Bolei, Z.; Jianping, S.; Chao, Z.; and Wang, X. 2018. Factorizable net: An efficient subgraph-based framework for scene graph generation. In *ECCV*, 346–363.

Lin, C.-Y., and Och, F. J. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *ACL*, 605.

Liu, Y.; Fu, J.; Mei, T.; and Chen, C. W. 2017. Let your photos talk: Generating narrative paragraph for photo stream via bidirectional attention recurrent neural networks. In *AAAI*, 1445–1452.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*, 3431–3440.

Lu, J.; Yang, J.; Batra, D.; and Parikh, D. 2018. Neural baby talk. In *CVPR*, 7219–7228.

Modi, Y., and Parde, N. 2019. The steep road to happily ever after: An analysis of current visual storytelling models. In *NAACL Workshop on SiVL*, 47–57.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 311–318.

Park, C. C., and Kim, G. 2015. Expressing an image stream with a sequence of natural sentences. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *NIPS*. Curran Associates, Inc. 73–81.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 91–99.

Rocktäschel, T.; Grefenstette, E.; Hermann, K. M.; Kociský, T.; and Blunsom, P. 2015. Reasoning about entailment with neural attention. *CoRR* abs/1509.06664.

Vedantam, R.; Lawrence Zitnick, C.; and Parikh, D. 2015. Cider: Consensus-based image description evaluation. In *CVPR*, 4566–4575.

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2017. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *PAMI* 39(4):652–663.

Wang, J.; Fu, J.; Tang, J.; Li, Z.; and Mei, T. 2018a. Show, reward and tell: Automatic generation of narrative paragraph from photo stream by adversarial training. In *AAAI*, 7396–7403.

Wang, X.; Chen, W.; Wang, Y.-F.; and Wang, W. Y. 2018b. No Metrics Are Perfect: Adversarial Reward Learning for Visual Storytelling. In *ACL*, 899–909.

Wang, B.; Ma, L.; Zhang, W.; Jiang, W.; and Zhang, F. 2019. Hierarchical photo-scene encoder for album storytelling. In *AAAI*, 8909–8916.

Xu, D.; Zhu, Y.; Choy, C.; and Fei-Fei, L. 2017. Scene graph generation by iterative message passing. In *CVPR*.

Yang, X.; Tang, K.; Zhang, H.; and Cai, J. 2019. Auto-encoding scene graphs for image captioning. In *CVPR*, 10685–10694.

Yao, T.; Pan, Y.; Li, Y.; and Mei, T. 2018. Exploring visual relationship for image captioning. In *ECCV*, 684–699.

Yu, L.; Bansal, M.; and Berg, T. 2017. Hierarchically-attentive rnn for album summarization and storytelling. In *EMNLP*, 966–971.

Zellers, R.; Yatskar, M.; Thomson, S.; and Choi, Y. 2018. Neural motifs: Scene graph parsing with global context. In *CVPR*.