Hierarchically Structured Reinforcement Learning for Topically Coherent Visual Story Generation

Qiuyuan Huang,[1*] Zhe Gan,[1*] Asli Celikyilmaz,[1] Dapeng Wu,[2] Jianfeng Wang,[1] Xiaodong He[3]
[1]Microsoft Research, Redmond, WA, [2]University of Florida, [3]JD AI Research
{qihua, zhe.gan, aslicel, jianfw}@microsoft.com; dpwu@ieee.org; xiaodong.he@jd.com

Abstract

We propose a hierarchically structured reinforcement learning approach to address the challenges of planning for generating coherent multi-sentence stories for the visual storytelling task. Within our framework, the task of generating a story given a sequence of images is divided across a two-level hierarchical decoder. The high-level decoder constructs a plan by generating a semantic concept (*i.e.*, topic) for each image in sequence. The low-level decoder generates a sentence for each image using a semantic compositional network, which effectively grounds the sentence generation conditioned on the topic. The two decoders are *jointly* trained end-to-end using reinforcement learning. We evaluate our model on the visual storytelling (VIST) dataset. Empirical results from both automatic and human evaluations demonstrate that the proposed hierarchically structured reinforced training achieves significantly better performance compared to a strong flat deep reinforcement learning baseline.

Introduction

Visual storytelling is the task of generating a sequence of coherent sentences (*i.e.*, a story) for an ordered image stream (Park and Kim 2015; Huang et al. 2016; Liu et al. 2017b). Inspired by the successful use of recurrent neural network (RNN) based encoder-decoder models employed in machine translation tasks (Cho et al. 2014; Sutskever, Vinyals, and Le 2014), variants of encoder-decoder models have shown promising results on the task of story generation (Huang et al. 2016).

The fundamental challenge, however, is that the strong performance of neural encoder-decoder models does not generalize well for visual storytelling. The task requires a full understanding of the content of each image as well as the relation among different images. The motivation behind our approach is to build a context-aware text-synthesis model that can efficiently encode the sequence of images and generate a topically coherent multi-sentence paragraph (see Fig. 2). We design a two-level hierarchical structure, where a high-level decoder constructs a plan by generating a topic for each image in sequence, and the low-level decoder generates a sentence conditioned on that topic.

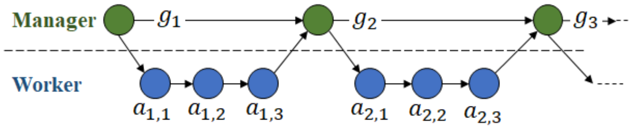


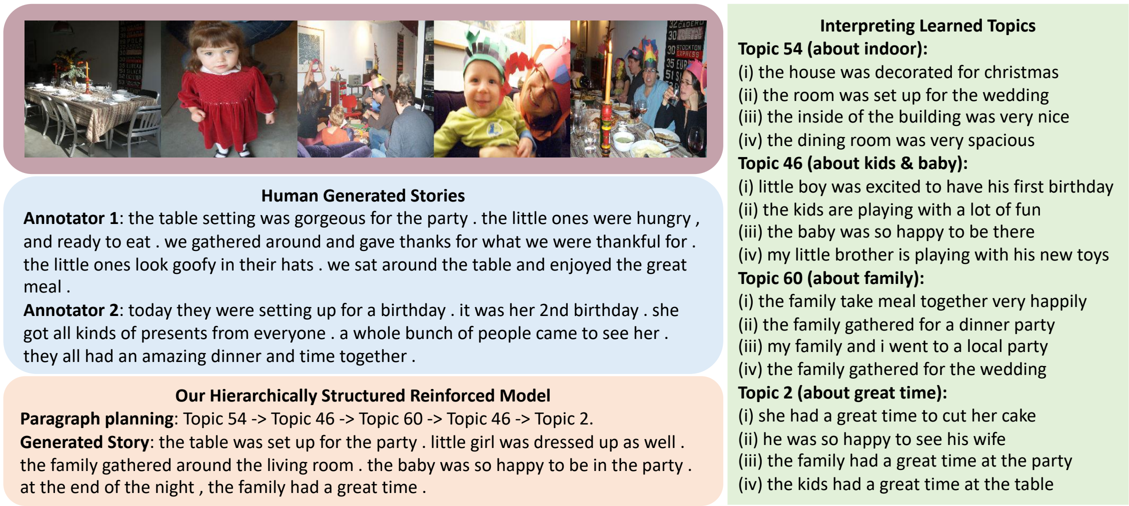
Figure 1: Overview of the Hierarchically Structured Reinforcement Learning, consisting of a Manager, a high-level decoder that generates topic (subgoal) sequences g_1,g_2,.., and a Worker, a low-level decoder which generates word sequences $a_{l,1}$..$a_{l,T}$ conditioned on the selected topic g_l.

Although the maximum likelihood estimation (MLE) is commonly used as the training loss for encoder-decoder RNNs, it may not be an appropriate surrogate for coherent long span generation task such as story generation. By only maximizing the ground truth probability, MLE can easily fail to exploit the wealth of information offered by the task specific losses. Most recent work in image captioning use reinforcement learning (RL) by providing sentence-level evaluation metrics for RNN training using global reward signals (Rennie et al. 2017; Ren et al. 2017; Liu et al. 2017a) (e.g, BLEU score). Motivated by the success of these work, we use RL to train our hierarchically structured model.

More specifically, we propose the hierarchically structured reinforcement learning (HSRL) model to realize a two-level generation mechanism (schematic overview is shown in Fig. 1). Our model consists of a high-level decoder (*i.e.*, Manager) and a low-level decoder (*i.e.*, Worker). The Manager aims to produce a sequence of semantically coherent topics for an image stream so that the overall theme is distributed among each sentence in the generated story. The topics assigned by the Manager are then supplied to the Worker for performing the task of sentence generation, given the surrounding image and textual context. Our Manager is a long short-term memory (LSTM) network (Hochreiter and Schmidhuber 1997), while the Worker is a semantic compositional network (SCN) (Gan et al. 2017), which effectively incorporates the topical information into the sentence generation process. The Manager and Worker are trained end-to-end jointly using a mixed MLE and self-critical reinforcement learning loss (Rennie et al. 2017) to generate focused and coherent stories.

Empirical results on the VIST dataset from both automatic and human evaluations demonstrate that the two-level

[*]Equal Contribution.



Human Generated Stories

Annotator 1: the table setting was gorgeous for the party . the little ones were hungry , and ready to eat . we gathered around and gave thanks for what we were thankful for . the little ones look goofy in their hats . we sat around the table and enjoyed the great meal .

Annotator 2: today they were setting up for a birthday . it was her 2nd birthday . she got all kinds of presents from everyone . a whole bunch of people came to see her . they all had an amazing dinner and time together .

Our Hierarchically Structured Reinforced Model

Paragraph planning: Topic 54 -> Topic 46 -> Topic 60 -> Topic 46 -> Topic 2.

Generated Story: the table was set up for the party . little girl was dressed up as well . the family gathered around the living room . the baby was so happy to be in the party . at the end of the night , the family had a great time .

Interpreting Learned Topics

Topic 54 (about indoor):
(i) the house was decorated for christmas
(ii) the room was set up for the wedding
(iii) the inside of the building was very nice
(iv) the dining room was very spacious

Topic 46 (about kids & baby):
(i) little boy was excited to have his first birthday
(ii) the kids are playing with a lot of fun
(iii) the baby was so happy to be there
(iv) my little brother is playing with his new toys

Topic 60 (about family):
(i) the family take meal together very happily
(ii) the family gathered for a dinner party
(iii) my family and i went to a local party
(iv) the family gathered for the wedding

Topic 2 (about great time):
(i) she had a great time to cut her cake
(ii) he was so happy to see his wife
(iii) the family had a great time at the party
(iv) the kids had a great time at the table

Figure 2: Example of hierarchically structured reinforcement learning for visual storytelling. Our model generates coherent stories by paragraph planning, *i.e.*, predicting a sequence of topics. In order to visualize learned topics, we present sentences generated from the corresponding topics in the test set. We manually assigned the topic names in this example for visual clarity.

decoder structure helps generate stories with improved narrative quality due to the control of overall thread on the generation of each sentence. Our benchmark analysis show that the hierarchically structured RL model significantly outperforms a strong flat deep reinforcement learning baseline, showing the effectiveness of paragraph planning and reinforcing the storytelling task at different levels.

Related work

Early work on generating descriptions for images have shaped the field of image/video captioning. A typical captioning model extracts a visual feature vector via a CNN, and then sends it to a language model for generation of a single sentence caption. Most notable work includes (Vinyals et al. 2015; Xu et al. 2015; Fang et al. 2015; Donahue et al. 2015; Karpathy and Fei-Fei 2015) for image captioning, and (Venugopalan et al. 2015; Pan et al. 2016a; 2016b; Yu et al. 2016; Pu et al. 2018) for video captioning.

More recently, the field has emerged into generation of long form text with the introduction of image paragraph generation (Krause et al. 2017) and dense video captioning (Krishna et al. 2017) tasks. In this work, we focus on visual storytelling, investigating the generation of narrative paragraph for a photo stream. First initial models used sequence-to-sequence framework (Huang et al. 2016; Liu et al. 2017b), while a joint embedding model was further developed in (Liu et al. 2017b) to overcome the large visual variance issue in image streams. Later in (Yu, Bansal, and Berg 2017), the task of album summarization and visual storytelling are jointly considered. While we share the same motivation as the above previous work, all of them rely on MLE training leaving out the fundamental problems, *e.g.*, exposure bias (Bengio et al. 2015) or not optimizing for the desired objective, which we tackle in this paper.

Most recent work on training captioning and story generation has used sequence scores such as BLEU (Ranzato et al. 2016) or CIDERr (Rennie et al. 2017) as a global reward to train a policy with the REINFORCE algorithm (Williams 1992). These work mainly focus on single sentence generation using *flat* RL approaches. In contrast, our work uses a hierarchically structured RL framework for capturing higher level semantics of the story generation task.

Our hierarchically structured model is related to the HRL work (Wang et al. 2018c) for video captioning; however, they have not explored the discovery or the usage of interpretable subgoals. The main novelty of our work when compared to them is the usage of explicit topics as subgoal representations. This yields significant improvements over the baselines, as well as provides a clear semantic subgoal for the sentences to be generated. In addition, one of the other novelty of our work is the usage of the SCN as the Worker, rather than a flat LSTM. Further, we introduce new approaches for training the high- and low-level decoders together, rather than in an iterative manner.

Our work is also related to (Wang et al. 2018a; 2018b), which uses adversarial training and inverse RL, respectively, for storytelling. However, neither of them has explored modeling of an explicit paragraph planning procedure. We introduced a "plan-ahead" strategy by using learned topics and proposing a hierarchically structured RL approach.

Hierarchically Reinforced Generation

Recent work in image captioning (Rennie et al. 2017; Pasunuru and Bansal 2017), machine translation (Wu et al. 2016), and summarization (Paulus, Xiong, and Socher 2018; Celikyilmaz et al. 2018) describe the benefits of fine-tuning neural generation systems with policy gradient methods on sentence-level scores (e.g. BLEU or CIDEr). While these approaches are able to learn a rough approximation of language when producing short sequences, they struggle on tasks involving long sequences (e.g., summaries, image paragraphs, stories, etc.). Preliminary work in introduc-

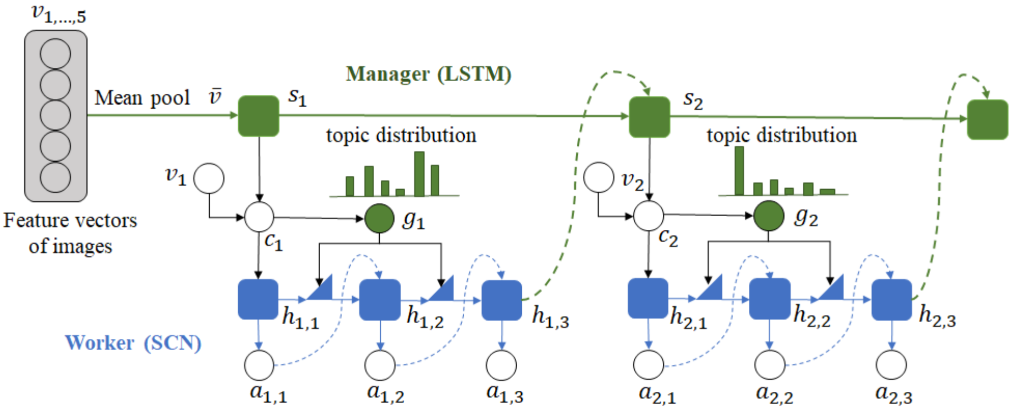


Figure 3: Proposed Manager-Worker framework. For each sequence of images, the Manager LSTM (top) generates a topic distribution g_ℓ, and the Worker SCN (Semantic Compositional Network) (bottom) generates sentences word by word $a_{\ell,t}$ conditioned on the topic distribution. Dash lines indicate copy operation where the output of one node is copied as input to the next node.

ing planning modules in a hierarchical reinforcement learning (HRL) framework has shown a promising new direction for long form generation, specifically in video captioning (Wang et al. 2018c). With hierarchies, they were able to generate coherent expressions, offering richer semantic structures and better syntactic consistency. However, they train the Manager and Worker policies alternately causing slow convergence issues for the HRL framework under the wake-sleep mode (Dayan and Neil 1996).

In visual storytelling, a latent world state is modeled by the words being generated. In this world, the scene in images, the players and the objects in it interact forming a coherent story of events. A suitable generated story must be coherent at both the *word-level*, linking words and phrases in a fluent way, and the *world-level*, describing events that fulfill the underlying topics. Motivated by this observation, we design a network that generates a plan of sequence of topics for long-form generation. These topics are used to seed low-level text generator that produce words, conditioned on the intent of the topic vector. Rather than using an alternating policy training, we train the Manager and Worker *jointly*, eliminating the problems of objective function balancing and slow convergence.

Building a sentence generator with a simple LSTM would not be sufficient to capture the linguistic features that are induced by the generated high level topics. A sentence generator that can provide an explicit semantic control over the sentence being generated can be better implemented with the semantic compositional network (SCN) (Gan et al. 2017), which we adopt in this work. Specifically, SCN adopts a mixture-of-expert design, which consists of a set of “expert networks”. Each expert is itself an LSTM with specific parameters relating to the topics and combination of all expert output yields globally coherent sentence generation.

In the following, we first describe the encoder and the structure of the two-level decoders. All $\mathbf{W}$ matrices are projection matrices. For simplicity we omit the bias vectors from the formulation.

Encoder

In the visual story generation task, we are given a sequence of images $\{\mathbf{i}_1, \ldots, \mathbf{i}_n\}$ and a corresponding multi-sentence story in training. The image sequences are first embedded into image features $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ via a pre-trained CNN (He et al. 2016) and mean pooling is applied to generate an image-sequence content vector $\bar{\boldsymbol{v}}$, which provides the whole model a global overview of the image-sequence content. This feature vector is then fed as initial input to the decoder. The words in the stories are embedded into word embedding vectors. At test time, the embeddings of the generated words are used.

Two-Level Decoder

Our two level Manager-Worker decoder is composed of two variants of LSTMs specifically designed for our topically structured network.

Manager As shown in Fig. 3, the Manager is implemented as an LSTM network, which uses the image-sequence content vector $\bar{\boldsymbol{v}}$ as the initial input to the LSTM. The input to any time step of the ℓ-th LSTM cell is the previous decoding output $\boldsymbol{s}_{\ell-1}$, and the last hidden state $\boldsymbol{h}_{\ell-1,T}$ (T-th output state) from the Worker LSTM (explained in detail later) after completing decoding of $(\ell-1)$-th sentence:

$$\boldsymbol{s}_\ell = \mathrm{LSTM}(\boldsymbol{s}_{\ell-1}, \boldsymbol{h}_{\ell-1,T}) \tag{1}$$

$$\boldsymbol{c}_\ell = [\boldsymbol{v}_\ell, \boldsymbol{s}_\ell] \tag{2}$$

$$g_\ell = \mathrm{softmax}(\mathrm{MLP}(\mathbf{W}_1 \boldsymbol{c}_\ell)), \tag{3}$$

where MLP$[\cdot]$ denotes the multi-layer perception. At each ℓ-th time step of the Manager, the corresponding image feature $\boldsymbol{v}_\ell$ is concatenated with the decoder output $\boldsymbol{s}_\ell$, which acts as the context vector $\boldsymbol{c}_\ell$ for the generation of sentence ℓ to fully describe the image content. Further, a new topic distribution g_ℓ is emitted via passing the resulting context vector $\boldsymbol{c}_\ell$ through a softmax layer. The Manager decoder generates a topic distribution g_ℓ at each step of the decoder.

Worker The worker generates sentences given the context vector $\boldsymbol{c}_\ell$ and the subgoal $\boldsymbol{g}_\ell$ using semantic compositional network (SCN) (Gan et al. 2017), as illustrated in Fig. 3.

Specifically, we define two weight tensors $\mathbf{W}_3 \in \mathbb{R}^{n_h \times n_x \times K}$ and $\mathbf{W}_4 \in \mathbb{R}^{n_h \times n_h \times K}$, where n_h is the number of hidden units, n_x is the dimension of word embedding and K is the number of topics. The k-th 2D "slice" of $\mathbf{W}_3[k]$ and $\mathbf{W}_4[k]$ represents parameters of the k-th expert. All K experts work cooperatively to generate an output $a_{\ell,t}$:

$$a_{\ell,t} = \text{softmax}(\text{MLP}(\mathbf{W}_2 \boldsymbol{h}_{\ell,t})) \tag{4}$$

$$\boldsymbol{h}_{\ell,t} = \sigma(\mathbf{W}_3(\boldsymbol{g}_\ell)\boldsymbol{x}_{\ell,t-1} + \mathbf{W}_4(\boldsymbol{g}_\ell)\boldsymbol{h}_{\ell,t-1}) \,, \tag{5}$$

$$\mathbf{W}_3(\boldsymbol{g}_\ell) = \sum_{k=1}^{K} \boldsymbol{g}_\ell[k]\mathbf{W}_3[k], \; \mathbf{W}_4(\boldsymbol{g}_\ell) = \sum_{k=1}^{K} \boldsymbol{g}_\ell[k]\mathbf{W}_4[k] \,, \tag{6}$$

where $\boldsymbol{x}_{\ell,t-1}$ is the embedding of word $a_{\ell,t-1}$, and $\boldsymbol{g}_\ell$ is a distributed vector with topic probabilities, rather than a one-hot vector. In order to reduce the number of model parameters, instead of implementing a tensor as in Eq. (6), we decompose $\mathbf{W}_3(\boldsymbol{g}_\ell)$:

$$\mathbf{W}_3(\boldsymbol{g}_\ell) = \mathbf{W}_{3a} \cdot \text{diag}(\mathbf{W}_{3b}\boldsymbol{g}_\ell) \cdot \mathbf{W}_{3c} \tag{7}$$

into a multiplication of three terms $\mathbf{W}_{3a} \in \mathbb{R}^{n_h \times n_f}$, $\mathbf{W}_{3b} \in \mathbb{R}^{n_f \times T}$ and $\mathbf{W}_{3c} \in \mathbb{R}^{n_f \times n_x}$, where n_f is the number of factors. The same factorization is also applied to $\mathbf{W}_4(\boldsymbol{g}_\ell)$. $\mathbf{W}_{3a}$ and $\mathbf{W}_{3c}$ are shared by all the topics, while the diagonal term, diag($\mathbf{W}_{3b}\boldsymbol{g}_\ell$), depends on the learned topics. Therefore, Eq. (7) *implicitly* defines K LSTMs. Our Worker model can be considered as training an ensemble of up to K LSTMs simultaneously. When the Manager emits a subgoal to the Worker, it also *implicitly* selects the corresponding LSTMs according to topics' probabilities to generate the current sentence.

After a special end-of-sentence token is generated, the last hidden state $\boldsymbol{h}_{\ell,T}$ of the Worker is sent to the Manager to emit the next subgoal $g_{\ell+1}$ distribution. This decoder defines the Worker policy, which maps the current state to the vocabulary distribution given the goal distribution to sample the next word.

Loss Functions

Training a story generator using MLE produces stories that are locally coherent, but lack the topical content of the image thread. Using a scoring function that rewards the model for capturing story semantics, the model learns to produce generations that better represents the world state. We design loss functions to train the policies based on this goal.

Manager Loss The Manager constructs a plan by generating a semantic concept (*i.e.*, topic) for each image in sequence and is trained using MLE objective conditioned on the previous output and the current state information from the Worker $\boldsymbol{h}_{\ell-1,T}$. It minimizes the negative log likelihood of predicting the next topic in the story given ground-truth topics g^*_ℓ:

$$\mathcal{L}^M_{\text{mle}}(\theta_m) = -\sum_{\ell=1}^{n} \log p_{\theta_m}(g^*_\ell | g^*_1, \ldots, g^*_{\ell-1}, h_{l-1,T}) \,, \tag{8}$$

where θ_m is the parameter vector of the Manager. This high-level semantic concept constrains the Worker's sentence generation policy. In the experiments, we define how we extract the ground-truth topic sequences $g^*_{1\ldots n}$ from the stories.

Worker Loss The Worker is responsible for generating sentences word by word. We define two different loss functions for the Worker training. The first is the MLE loss, which corresponds to the maximum-likelihood training:

$$\mathcal{L}^W_{\text{mle}}(\theta_w) = -\sum_{\ell=1}^{n}\sum_{t=1}^{T} \log p_{\theta_w}(y^*_{\ell,t} | y^*_{\ell,1}, \ldots, y^*_{\ell,t-1}, g_\ell, \boldsymbol{c}_\ell) \,, \tag{9}$$

where θ_w is the parameter vector of the Worker; y^*_ℓ is the ground-truth sentence and $y^*_{\ell,t}$ denotes the t-th word in sentence y^*_ℓ. The generation is conditioned on the goal distribution g_ℓ from the Manager and the context vector $\boldsymbol{c}_\ell$.

For the second loss, we assume the worker policy is stochastic and we learn it using the self-critical approach of (Rennie et al. 2017). In self critical training, the model learns to gather more rewards from its generations by randomly sampling sequences that achieve higher reward than its best greedy samples. Two separate output sequences are sampled at each training iteration t: The first $\hat{y}$ is generated by randomly sampling from the model's distribution $p_{\theta_w}(\hat{y}_{\ell,t} | \hat{y}_{\ell,1}, \ldots, \hat{y}_{\ell,t-1}, g_\ell, \boldsymbol{c}_\ell)$. The model's own outputs are the inputs at the next time step, acting similarly to test time conditions. Once the sentence is generated, a reward $\hat{r}$ is obtained. A second sequence $y^\star$, is generated by greedily sampling from the model's distribution $p_{\theta_w}(y^\star_{\ell,t} | y^\star_{\ell,1}, \ldots, y^\star_{\ell,t-1}, g_\ell, \boldsymbol{c}_\ell)$ at each time step t and a reward $r^\star$ is obtained. The following loss is used to train the self-critical RL method using the generated sequences and both rewards:

$$\mathcal{L}^W_{\text{rl}}(\theta_w) = -(r^\star - \hat{r}) \cdot \sum_{\ell=1}^{n}\sum_{t=1}^{T} \log p_{\theta_w}(\hat{y}_{\ell,t} | \hat{y}_{\ell,1}, \ldots, \hat{y}_{\ell,t-1}, g_\ell, \boldsymbol{c}_\ell) \,. \tag{10}$$

The model encourages generating sequences that receive more reward than the best sequence that can be greedily sampled from the current policy. This way, self-critical training allows the model to explore sequences that yield higher reward than the current best policy.

Mixed Worker Loss Minimizing the RL loss in Eq. (10) alone does not ensure the readability and fluency of the generated sentences. The model quickly learns to generate simple sequences that *exploit* the teacher for higher rewards despite producing nonsensical sentences. To remedy this, a better way is to optimize a mixed objective (Wu et al. 2016;

Pasunuru and Bansal 2017) that balances learning for generating coherent story sentences with maintaining generator's language model:

$$\mathcal{L}^{W}_{\text{mix}} = \gamma\mathcal{L}^{W}_{\text{rl}} + (1-\gamma)\mathcal{L}^{W}_{\text{mle}}\,, \tag{11}$$

where $\gamma \in [0, 1]$ is a scaling factor balancing the importance of $\mathcal{L}^{W}_{\text{rl}}$ and $\mathcal{L}^{W}_{\text{mle}}$. For annealing and faster convergence, we start with $\gamma = 0$ (*i.e.*, minimizing the cross-entropy loss), and gradually increase γ to a maximum value $\gamma_{max} < 1$.

Policy Learning

We investigate 3 objectives to combine the Manager and Worker policies in an end-to-end learning framework.

Cascaded Training The Manager and the Worker are trained independently. The manager is trained using the MLE loss $\mathcal{L}^{M}_{\text{mle}} = -\sum_{\ell=1}^{n} \log p(g^*_\ell | g^*_1, \ldots, g^*_{\ell-1})$ without any input from the Worker. Once the Manager training is converged, the Worker uses the trained Manager model to generate a topic sequence given each image sequence. The Worker is trained using the mixed loss $\mathcal{L}^{W}_{\text{mix}}$ with the ground-truth topic sequence.

Iterative Training (Wake-Sleep Mode) The Manager and Worker are trained iteratively in a wake-sleep mode similar to HRL training of (Wang et al. 2018c). The Manager observes the current state $\boldsymbol{h}_{\ell-1,T}$ after the Worker generates sentence $\ell - 1$, and produces a topic distribution g_ℓ for the generation of sentence ℓ. The Worker takes as input a state $\boldsymbol{h}_{\ell,t-1}$ and a topic distribution g_ℓ, and predicts its next action $a_{\ell,t}$, *i.e.*, the t-th word in sentence ℓ. Note that the the topic distribution g_ℓ at Manager decoder time ℓ serves as a guidance and remains a constant input to the Worker decoder during the whole process of generating sentence ℓ.

In the early stage of training, we set $\gamma = 0$ to pretrain the Worker policy with $\mathcal{L}^{W}_{\text{mle}}$. This ensures that our Worker RL agent training starts at a good initialization point. After the warm-up pre-training, the Worker policy and the Manager policy are trained iteratively while keeping the other one fixed, using the losses $\mathcal{L}^{M}_{\text{mle}}$ and $\mathcal{L}^{W}_{\text{mix}}$, respectively. In training the Worker, a sentence-level CIDEr score (Vedantam, Lawrence Zitnick, and Parikh 2015) is used as the intermediate reward, which measures how well a local sentence matches the ground-truth.

Joint Training Iterative training may yield instability and slow convergence issues in objective function optimization. An alternative is to use a joint training scheme to enable the Manager and Worker to backpropogate from each other's losses and optimize globally. We introduce a new joint training objective combining the Manager and the Worker losses:

$$\mathcal{L}_{\text{joint}} = (1-\gamma_1)\mathcal{L}^{M}_{\text{mle}} + \gamma_1(\gamma_2\mathcal{L}^{W}_{\text{rl}} + (1-\gamma_2)\mathcal{L}^{W}_{\text{mle}})\,. \tag{12}$$

Similar to iterative training, in the early stages of training, we set $\gamma_2 = 0$ to pretrain the Worker policy with MLE loss. After the warm-up pre-training, the Worker policy and the Manager policy are trained jointly using Eq. (12). Here, we also use sentence-level CIDEr score as the intermediate reward. While the Worker aims to learn a better sentence generator, the Manager aims to learn a better paragraph planner by optimizing for high-level topic semantics. The Manager's parameters are updated based on the rewards the Worker receives upon generating a story.

Sentence Level Credit Assignment The Worker takes actions by generating words until a special end-of-sentence token is reached, then it obtains a reward. We evaluate the model on story level until all sentences are generated. In multi-sentence generation tasks, the final reward after full story is generated can be a weak signal. To alleviate that, we also use *intermediate* rewards by assigning sentence-level rewards, by evaluating how well the current generated sentence matches the ground-truth. This intermediate reward helps alleviate the reward sparsity problem, and in experiments, we found that it also helps reduce sentence and word repetition issues, yielding more diverse stories.

Experimental Results

Dataset For learning and evaluation we use the VIST dataset (Huang et al. 2016), which are collected from Flickr albums and then annotated by Amazon's Mechanical Turk (AMT). Each story has 5 images and 5 corresponding descriptions. After filtering out broken images, we obtain 19,828 image sequences with 49,629 stories in total. On average, each image sequence is annotated with roughly 2.5 stories. The 19,828 image sequences are partitioned into three parts, 15,851 for training, 1,976 for validation and 2,001 for testing, respectively. Correspondingly, the 49,629 stories are also split into three parts, 39,676 for training, 4,943 for validation and 5,010 for testing, respectively. The vocabulary consists of 12,977 words.

Training We extract the image features with ResNet-152 (He et al. 2016) pretrained on the ImageNet dataset. The resulting image feature vector $\boldsymbol{v}$ has 2,048 dimensions. We use the GLove embedding vectors of (Pennington, Socher, and Manning 2014) for word embedding initialization.

Since the VIST dataset (Huang et al. 2016) is originally not annotated with topic sequences, we use clustering to generate golden topic sequences. Specifically, we use a simple k-means algorithm to cluster the ResNet-152 image features into K clusters, where each cluster implicitly defines a topic, and the sentences are then considered as belonging to the same cluster as the corresponding images.

Results

Scores We compute BLEU-4 (Papineni et al. 2002), METEOR (Banerjee and Lavie 2005), CIDEr (Vedantam, Lawrence Zitnick, and Parikh 2015), ROUGE-L (Lin 2004), and SPICE (Anderson et al. 2016) metrics for evaluation.

Baselines We provide results reported in previous methods: (Huang et al. 2016) adds a decoder-time heuristic method to alleviate the repetition issue when generating stories, (Liu et al. 2017b) uses an additional cross-modality

Table 1: Evaluation results for generated stories by models and baselines. **bold** the top performing result. The Worker+Random topics and Worker+GTT are the lower and upper bound scores for our Hierarchically Structured RL (HSRL) model.

Methods	BLEU-4	ROUGE-L	CIDEr-D	METEOR-v1	METEOR-v2	SPICE
seq2seq + heuristics (Huang et al. 2016)	3.50	–	6.84	10.25	31.4	–
BARNN (Liu et al. 2017b)	–	–	–	–	33.3	–
h-attn-rank (Yu, Bansal, and Berg 2017)	–	29.8	7.38	–	33.9	–
AREL (Wang et al. 2018b)	**14.1**	29.6	9.5	–	35.2	–
Show, Reward & Tell (Wang et al. 2018a)	5.16	–	**11.35**	12.32	–	–
Our Baselines						
Baseline LSTM (MLE)	7.32	27.34	7.52	8.04	31.43	7.03
Baseline LSTM (RL)	8.16	27.52	7.64	8.31	31.52	7.57
HRL (Wang et al. 2018c)	8.94	27.90	8.72	11.4	32.67	8.73
16 Topics						
Worker+Random topics	4.70	23.04	3.90	5.43	27.11	5.54
HSRL w/ Cascaded Training	10.38	30.14	9.65	12.32	34.73	9.62
HSRL w/ Iterative Training	11.23	30.32	9.68	12.83	34.82	9.87
HSRL w/ Joint Training	**11.64**	**30.61**	**9.73**	**13.27**	**34.95**	**10.25**
Worker+GTT	13.41	31.53	10.82	14.27	35.48	12.83
64 Topics						
HSRL w/ Cascaded Training	11.95	30.06	10.03	13.34	34.81	12.42
HSRL w/ Iterative Training	12.04	30.65	10.34	13.42	35.21	12.66
HSRL w/ Joint Training	**12.32**	**30.84**	**10.71**	**13.53**	**35.23**	**12.97**
Worker+GTT	14.68	32.73	12.63	16.32	36.22	14.34

embedding model to regularize the story generation model, while (Yu, Bansal, and Berg 2017) uses a hierarchical model, and considers performing album summarization and storytelling simultaneously. All these models are trained using the MLE loss. Recently, (Wang et al. 2018b) and (Wang et al. 2018a) proposes the usage of a learned reward in an RL setup for improving the performance. Our model also uses RL, but with a different focus on using learned topics for effective paragraph planning.

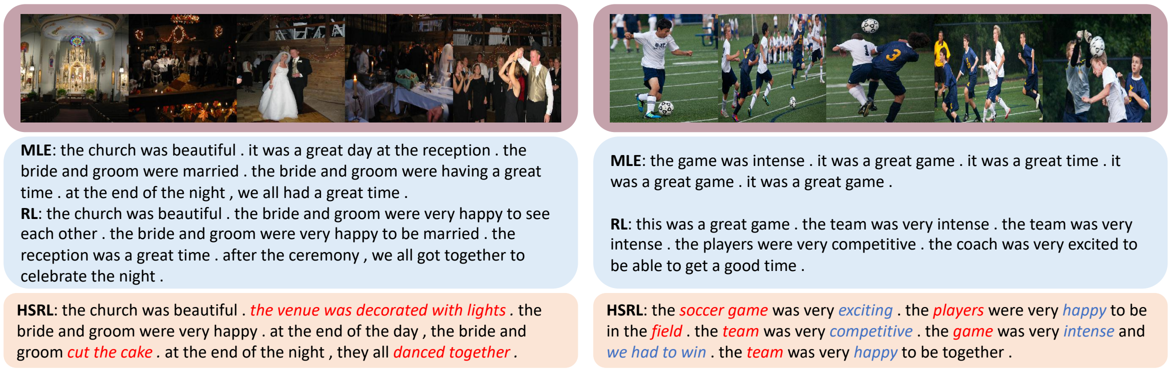
Note that we observe some discrepancy in the reported results of the related work (see Table 1). Specifically comparing to closest works to ours, in (Wang et al. 2018a), a high CIDEr-D and a low BLEU-4 score is reported, while in (Wang et al. 2018b) even though a much higher BLEU-4 score is reported, their CIDEr-D score is much lower. These discrepancies are possibly due to differences in data preprocessing and evaluation scripts.

Therefore, for fair comparison, we mainly focus on comparing our results with the following re-implemented baselines: Baseline LSTM (MLE) is trained with cross-entropy loss using Eq. (9), while Baseline LSTM (RL) is trained with self-critical REINFORCE loss using Eq. (10). We reimplemented the HRL approach in (Wang et al. 2018c) for our task, and also implemented a variant of our model, Worker+Random topics, which learns a Worker decoder without the Manager decoder but randomly samples a sequence of topics from an interval of [1,K], where K is the total number of topics. We use this baseline as a lower-bound of our model. Similarly, we also implemented a variant of our model, Worker+GTT, again with only a Worker decoder, but this time we used the ground-truth topics (GTT) as input to the Worker. We use this baseline as an upper-bound of our model. We experimented with K=16 and K=64. All the metrics are computed by using the code released by the COCO evaluation server (Chen et al. 2015).

Quantitative Results Our results[1] in Table 1 show that models optimized with joint training achieve the greatest improvement for all the scores. All hierarchically structured reinforced (HSRL) story generation models outperform the baseline LSTM (MLE) and LSTM (RL) training models by a margin. HRL only improves flat RL marginally in this task, while our HSRL achieved much better performance than HRL. This indicates the efficiency of using explicit topics as subgoals, rather than a latent continuous vector as used in (Wang et al. 2018c). Additionally, HSRL models consistently achieve high improvements across different number of topics against the lower-bound Worker+Random model, which was trained using random topic sequences. The joint training results are close to the upper bound Worker+GTT model, indicating the stronger performance of joint training.

Careful analysis of these three end-to-end training methods yields that the model optimized with cascaded training method performs worse than the rest of the training methods. Iterative training with HSRL improves the results over the cascaded training method across all scores, indicating the impact of sentence planning with higher-level decoder for representing higher level semantics of the story generation. The model trained jointly with HSRL achieves even higher scores, across different numbers of topics, showing the benefits of training a two-level decoder network jointly rather than iteratively in an alternating training mode. The

[1] METEOR-v1 represents the version used in the COCO evaluation server, and METEOR-v2 represents version 1.5 with HTER weights.



MLE: the church was beautiful . it was a great day at the reception . the bride and groom were married . the bride and groom were having a great time . at the end of the night , we all had a great time .
RL: the church was beautiful . the bride and groom were very happy to see each other . the bride and groom were very happy to be married . the reception was a great time . after the ceremony , we all got together to celebrate the night .

HSRL: the church was beautiful . *the venue was decorated with lights* . the bride and groom were very happy . at the end of the day , the bride and groom *cut the cake* . at the end of the night , they all *danced together* .

MLE: the game was intense . it was a great game . it was a great time . it was a great game . it was a great game .

RL: this was a great game . the team was very intense . the team was very intense . the players were very competitive . the coach was very excited to be able to get a good time .

HSRL: the *soccer game* was very *exciting* . the *players* were very *happy* to be in the *field* . the *team* was very *competitive* . the *game* was very *intense* and *we had to win* . the *team* was very *happy* to be together .

Figure 4: Example stories generated by three storytelling models. Compared to baseline models, the hierarchically structured model generates more coherent, detailed and expressive paragraphs.

success of joint training can also be attributed to the fact that at training time, both Manager and Worker have access to the ground truth topic and word sequences through reward functions and teacher forcing (Lamb et al. 2016) in MLE training, while in iterative training the Manager and Worker has access to either topics or word sequences.

Human Evaluation We perform two human evaluation tasks using Amazon Mechanical Turk: pairwise comparison and the closeness to the ground-truth story. For both tasks, we use all the 2001 test image sequences. Each image sequence is presented to the worker and the worker is requested to judge which generated story is better in terms of relevance, expressiveness and concreteness. A neutral option is provided if the worker cannot tell which one is better. The compared approaches include MLE, RL and our HSRL model. Results are summarized in Table 2. It is clearly shown that our HSRL approach is better than MLE and RL.

The second task requires the AMT-worker to judge the closeness of each story to the ground-truth story comparing against the other two generated stories. HSRL wins 55.87% tasks, RL (the second best) wins 34.53%, while MLE only wins 9.60%. More details about human evaluation are provided in the supplementary material.

Qualitative Analysis In Fig. 2 we illustrate the paragraph generation process by explicitly showing the generated topic sequences. A high-level plan is first constructed by generating a sequence of topics, based on which a sequence of sentences are then generated. Specifically in this example, our model constructs a plan as follow: (*i*) describe the story background; (*ii*) describe the little girl, family and baby sequentially; (*iii*) end the story with “everyone had a great time”.

In Fig. 4, we show two additional examples that are sampled from the test set. We compare our HSRL model with two of our baseline models and obtain the following observations. First, our HSRL model is globally coherent. It can describe images in connection to previous images more coherently. For instance, in Fig. 4(right), the model identifies that the setting of the story is *soccer game* and follows to explain that there were *players*, *field* and *team* in the scene, in accordance of their appearance in image sequences.

Table 2: Results of pairwise human comparison.

	MLE vs HSRL			RL vs HSRL		
Choice (%)	MLE	HSRL	Tie	RL	HSRL	Tie
Relevance	27.53	63.93	8.53	34.87	56.40	8.73
Expressiveness	24.87	62.53	12.60	31.60	55.67	12.73
Concreteness	25.87	62.47	11.66	33.93	54.73	11.33

The generated stories also indicate that the sentences generated by HSRL is more diverse compared to the baselines. For instance each sentence generated by HSRL in Fig. 4(right) is different from others, which can be attributed to the fact that sentence generation is nicely controlled by the topics generated for the story. In contrast, the stories generated by two baseline models are less diverse and contains repetitions. As a by-product, it was exciting to observe that the stories generated by the HSRL are more vibrant and engaging. Just like a human-created story that touches on human emotions, we found that the story generated by the HSRL is more emotional. For example, the words “*happy, exciting/excited, fun/funny, intense, tired*” have been used 1137, 678, 316, 51, and 58 times in HSRL generated stories, and 410, 138, 291, 38 and 3 times in the RL baseline.

Finally, we also observe that the HSRL model was able to learn to exploit the reward function to include more details in the generated text. For instance, in Fig. 4(left), though stories generated by all the 3 models are reasonable, we observe less details in the stories generated by the baselines, while the salient facts like “decorated with lights, cut the cake, danced together” are captured by our model. More examples are provided in the supplementary material.

Conclusion

We investigated the problem of generating topically coherent visual stories given an image stream and demonstrated that the use of hierarchically structured reinforcement learning can improve the generation. Analysis demonstrates that this improvement is due to the joint training of two hierarchically structured decoders, where the higher decoder is optimized for better learning high-level topical semantics, and the lower decoder optimizes to obtain more rewards for generating topically coherent sentences.

References

Anderson, P.; Fernando, B.; Johnson, M.; and Gould, S. 2016. Spice: Semantic propositional image caption evaluation. In *ECCV*.

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

Bengio, S.; Vinyals, O.; Jaitly, N.; and Shazeer, N. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *NIPS*.

Celikyilmaz, A.; Bosselut, A.; He, X.; and Choi, Y. 2018. Deep communicating agents for abstractive summarization. In *NAACL*.

Chen, X.; Fang, H.; Lin, T.-Y.; Vedantam, R.; Gupta, S.; Dollár, P.; and Zitnick, C. L. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

Dayan, P., and Neil, R. M. 1996. Factor analysis using delta-rule wake-sleep learning. *Technical Report, Department of Statistics, University of Toronto*.

Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*.

Fang, H.; Gupta, S.; Iandola, F.; Srivastava, R. K.; Deng, L.; Dollár, P.; Gao, J.; He, X.; Mitchell, M.; Platt, J. C.; et al. 2015. From captions to visual concepts and back. In *CVPR*.

Gan, Z.; Gan, C.; He, X.; Pu, Y.; Tran, K.; Gao, J.; Carin, L.; and Deng, L. 2017. Semantic compositional networks for visual captioning. In *CVPR*.

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

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*.

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*.

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

Krause, J.; Johnson, J.; Krishna, R.; and Fei-Fei, L. 2017. A hierarchical approach for generating descriptive image paragraphs. In *CVPR*.

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Niebles, J. C. 2017. Dense-captioning events in videos. In *ICCV*.

Lamb, A. M.; GOYAL, A. G. A. P.; Zhang, Y.; Zhang, S.; Courville, A. C.; and Bengio, Y. 2016. Professor forcing: A new algorithm for training recurrent networks. In *NIPS*.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *ACL Workshop*.

Liu, S.; Zhu, Z.; Ye, N.; Guadarrama, S.; and Murphy, K. 2017a. Improved image captioning via policy gradient optimization of spider. In *ICCV*.

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

Pan, P.; Xu, Z.; Yang, Y.; Wu, F.; and Zhuang, Y. 2016a. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*.

Pan, Y.; Mei, T.; Yao, T.; Li, H.; and Rui, Y. 2016b. Jointly modeling embedding and translation to bridge video and language. In *CVPR*.

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

Park, C. C., and Kim, G. 2015. Expressing an image stream with a sequence of natural sentences. In *NIPS*.

Pasunuru, R., and Bansal, M. 2017. Reinforced video captioning with entailment rewards. In *EMNLP*.

Paulus, R.; Xiong, C.; and Socher, R. 2018. A deep reinforced model for abstractive summarization. In *ICLR*.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Pu, Y.; Min, M. R.; Gan, Z.; and Carin, L. 2018. Adaptive feature abstraction for translating video to text. In *AAAI*.

Ranzato, M.; Chopra, S.; Auli, M.; and Zaremba, W. 2016. Sequence level training with recurrent neural networks. In *ICLR*.

Ren, Z.; Wang, X.; Zhang, N.; Lv, X.; and Li, L.-J. 2017. Deep reinforcement learning-based image captioning with embedding reward. In *CVPR*.

Rennie, S. J.; Marcheret, E.; Mroueh, Y.; Ross, J.; and Goel, V. 2017. Self-critical sequence training for image captioning. In *CVPR*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

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

Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R.; Darrell, T.; and Saenko, K. 2015. Sequence to sequence-video to text. In *ICCV*.

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *CVPR*.

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*.

Wang, X.; Chen, W.; Wang, Y.-F.; and Wang, W. Y. 2018b. No metrics are perfect: Adversarial reward learning for visual storytelling. In *ACL*.

Wang, X.; Chen, W.; Wu, J.; Wang, Y.-F.; and Wang, W. Y. 2018c. Video captioning via hierarchical reinforcement learning. In *CVPR*.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement Learning*.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R. S.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.

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

Yu, H.; Wang, J.; Huang, Z.; Yang, Y.; and Xu, W. 2016. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*.