Video Storytelling: Textual Summaries for Events

Junnan Li, Yongkang Wong, Member, IEEE, Qi Zhao, Member, IEEE, Mohan S. Kankanhalli, Fellow, IEEE

Abstract—Bridging vision and natural language is a longstanding goal in computer vision and multimedia research. While earlier works focus on generating a single-sentence description for visual content, recent works have studied paragraph generation. In this work, we introduce the problem of video storytelling, which aims at generating coherent and succinct stories for long videos. Video storytelling introduces new challenges, mainly due to the diversity of the story and the length and complexity of the video. We propose novel methods to address the challenges. First, we propose a context-aware framework for multimodal embedding learning, where we design a Residual Bidirectional Recurrent Neural Network to leverage contextual information from past and future. The multimodal embedding is then used to retrieve sentences for video clips. Second, we propose a Narrator model to select clips that are representative of the underlying storyline. The Narrator is formulated as a reinforcement learning agent which is trained by directly optimizing the textual metric of the generated story. We evaluate our method on the Video Story dataset, a new dataset that we have collected to enable the study. We compare our method with multiple state-of-the-art baselines and show that our method achieves better performance, in terms of quantitative measures and user study.

Index Terms—Video Storytelling, Video Captioning, Sentence Retrieval, Multimodal Embedding Learning

I. INTRODUCTION

Generating natural language descriptions for visual content has become a major research problem for computer vision and multimedia research. Driven by the advent of large datasets pairing images and videos with natural language descriptions, encouraging progress has been made in both image and video captioning task. Building on the earlier works that describe images/videos with a single sentence, a recent work focuses on visual paragraph generation which aims to provide detailed descriptions for images/videos [1], [2], [3], [4], [5], [6], [7], [8], [9], [10].

Existing literature on visual paragraph generation can be divided into two categories: a) fine-grained dense description of images or short videos [1], [2], [3], [4], [5], [6], [7], [8] and b) storytelling of photo streams [9], [10]. However, other than images and short videos clips, consumers often record long videos for important events, such as birthday party or wedding. In order to assist users to access long videos, previous works on video summarization focused on generating a shorter version of the video [11], [12], [13], [14], [15], [16], [17]. We argue that compared to visual summary, a textual story generated for a long video is more compact, semantically meaningful, and easier to search and organize. In this paper, we introduce a new problem: visual storytelling of long videos. Specifically, we aim to compose coherent and succinct stories for videos that cover entire events.

The proposed video storytelling problem is different from the aforementioned visual paragraph generation works. First, compared to dense-captioning [1], [2], [3], [4], [5], we focus on long videos with more complex event dynamics while not aiming to describe every detail presented in the video. Instead, we focus on extracting the important scenes and compose a story. Second, video storytelling is a more visually-grounded problem. This is in contrast with the storytelling of photo streams [9], [10], where the datasets [9], [18] only consist of a few photos per story. As a result, the annotated stories may depend on annotators’ imagination and prior knowledge, and the challenge is to fill in the visual gap between photos. On the other hand, the challenge for video storytelling is not lack of visual data, but to compose a coherent and succinct story from abundant and complex visual data.

The task of video storytelling introduces two major challenges. First, compared to single-sentence descriptions, long stories contain more diverse sentences, where similar visual contents can be described very differently depending on the context. However, the widely-used Recurrent Neural Network (RNN) based sentence generation model tends to produce generic, repetitive and high-level descriptions [13], [19], [20]. Second, long video usually contains multiple actors, multiple locations and multiple activities. Hence it is difficult to discover the major storyline that is both coherent and succinct.

In this work, we address these two challenges with the following contributions. First, we propose a context-aware framework for multimodal semantic embedding learning. The embedding is learned through a two-step local-to-global process. The first step models individual clip-sentence pairs to learn a local embedding. The second step models the entire video as a sequence of clips. We design a Residual Bidirectional RNN (ResBRNN) that captures the temporal dynamics of the video and incorporates contextual information from past and future into the multimodal embedding space. The proposed ResBRNN preserves temporal coherence and increases diversity of the corresponding embeddings.

Second, we propose a Narrator model to generate stories. Given an input video, the Narrator extracts from it a sequence of important clips. Then a story is generated by retrieving a sequence of sentences that best matches the clips in the multimodal embedding space, and concatenating the retrieved sentences. Fig. 1 shows an example story using the proposed method. Discovering the important clips is difficult because there is no clear definition as to what visual aspects are important in terms of forming a good story. To this end, we formulate the Narrator as a reinforcement learning agent...
 Guests begin to walk into a church for the ceremony. 
The wedding party makes their way down the aisle followed by the bride and her father. 
The bride and groom face each other to recite their vows in front of the priest. 
Everyone gathers for the reception as the bride and groom take the first dance. 
The bride and groom cut the wedding cake together and take the traditional first bite.
domain. Sah \textit{et al.} \cite{47} and Chen \textit{et al.} \cite{48} further propose methods to generate a textual summary for a video. However, both methods generate disconnected captions for individual clips while ignoring the storyline of an event.

In this work, we propose a novel method to select the important video clips, which circumvents the need for predefined criteria. In contrast, we directly leverage the reference stories written by human for supervision, and learn a clip selection policy that maximizes the language metric of the stories written by human for supervision, and learn a clip selection policy that maximizes the language metric of the generated stories. The advantage of our method is that we can train with a more well-defined and objective goal.

\textbf{C. Learning Task-specific Policies}

We draw inspiration from recent works that use REINFORCE \cite{49} to learn task-specific policies. Ba \textit{et al.} \cite{50} and Mnih \textit{et al.} \cite{51} learn spatial attention policies for image classification, whereas other works apply REINFORCE for image captioning \cite{22}, \cite{52}, \cite{53}, \cite{54}. In video domain, Yeung \textit{et al.} \cite{55} learn a policy for the task of action detection. Lan \textit{et al.} \cite{56} propose a reinforcement learning agent for video fast-forwarding.

\section{Method}

The proposed video storytelling method involves two sub-tasks: (a) discover the important clips from a long video, and (b) generate a story for the selected video clips with sentence retrieval. To address this, the proposed method includes two parts. In the first part, we propose a context-aware multimodal embedding learning framework where the learned embedding captures the event dynamics. In the second part, we propose a \textit{Narrator model} that learns to discover important clips to form a succinct and diverse story.

\subsection{A. Context-Aware Multimodal Embedding}

We propose a two-step local-to-global multimodal embedding learning framework. An overview of the framework is shown in Fig. \ref{fig:framework}. In the first step, we follow the image-sentence ranking model proposed by \cite{57} to learn a local clip-sentence embedding. In the second step, we propose ResBRNN that leverages video-story pairs to model the temporal dynamics of a video, and incorporates global contextual information into the embedding. We will first describe the clip-sentence embedding.

\textbf{Encoder} The model by \cite{57} consists of two branches: a Long Short-Term Memory (LSTM) network to encode a sentence into a fixed-length vector, and a Convolutional Neural Network (CNN) followed by linear mapping to encode an image. Similarly, we construct two encoders for sentence and video, respectively. The sentence encoder is a RNN that takes the sentence with its last hidden state as input, and represents the sentence with its last hidden state \(v\). For the video encoder, inspired by the encoder-decoder model \cite{28}, \cite{29}, we use a RNN that take in the output of a CNN applied to each input frame, and encodes the video as its last hidden state \(x\). In terms of the recurrent unit, we use Gated Recurrent Unit (GRU) \cite{59} instead of LSTM, because GRUs have been

![Fig. 2: Local-to-global multimodal embedding learning framework.](image-url)
shown to achieve comparable performance to LSTM on several sequential modeling tasks while being simpler [60]. The GRU uses gating units to modulate the flow of information, by the following operation:

\[
\begin{align*}
    r_t &= \sigma(W_{rx}x_t + W_{rh}h_{t-1} + b_r) \\
    z_t &= \sigma(W_{zx}x_t + W_{zh}h_{t-1} + b_z) \\
    \tilde{h} &= \text{tanh}(W_{hz}x_t + W_{hh}r_t \odot h_{t-1} + b_h) \\
    h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}
\end{align*}
\]

(1)

where \( \odot \) denotes element-wise multiplication, \( \sigma \) are Sigmoid functions, \( t \) is the current time, \( x_t \) is the input, \( \tilde{h} \) is the current hidden state, \( h_t \) is the output. \( r_t \) and \( z_t \) are reset gate and update gate, respectively.

In this work, the dimensionality of the hidden state of both encoder RNNs are set to be 300. The Word2Vecs are 300-dimensional vectors. The CNN we use on video frame is the 101-layer ResNet [61] that outputs 2048-dimensional vectors. Following [28], [27], a video is sampled on every tenth frame.

**Clip-Sentence Ranking Loss** We aim to learn a joint embedding space where distance between embeddings reflects semantic relation. We assume that paired video clip and sentence share the same semantics, hence they should be closer in the embedding space. We define a similarity scoring function \( s(v, x) = v \cdot x \) for sentence embedding \( v \) and clip embedding \( x \), where \( v \) and \( x \) are first scaled to have unit norm (making \( s \) equivalent to cosine similarity).

We then minimize the following pairwise ranking loss:

\[
\begin{align*}
    \min_{\theta_{en}} \sum_v \sum_k \max\{0, \alpha - s(x, v) + s(x, v_k)\} \\
    + \lambda \sum_v \sum_k \max\{0, \alpha - s(v, x) + s(v, x_k)\}
\end{align*}
\]

(2)

where \( \theta_{en} \) denote all the parameters to be learned (weights of the two encoder RNNs), \( v_k \) is the negative paired (non-descriptive) sentence for clip embedding \( x \), and vice-versa with \( x_k \). The negative samples are randomly chosen from training set and re-sampled every epoch. \( \alpha \) denotes the margin and is set to 0.1 in our experiment. The weight \( \lambda \) balances the strengths of the two ranking terms. We found \( \lambda = 0.5 \) produces the best results in our experiment.

**Temporal Dynamics with Residual Bidirectional RNN**

One major challenge for video storytelling is that visually similar video clips could have different semantic meanings depending on the global context. For example, in Fig. 2, the second and the penultimate clip both show some people around a tent. However, depending on the sequence of events that happen before and after, their descriptions differ from ‘set up their tent’ to ‘disassemble their tent’. It is therefore very important to model the entire flow of the story. To this end, we propose a ResBRNN model, that builds upon the embedding learned in the first step, and leverages video-story pairs to incorporate global contextual information into the embedding.

ResBRNN takes a sequence of clip embedding vectors \( X = \{x_1, x_2, \ldots, x_N\} \), and outputs a sequence of context-aware clip embedding vectors \( M = \{m_1, m_2, \ldots, m_N\} \) of the same size. The role of ResBRNN is to refine the embeddings \( X \) by incorporating past and future information. Inspired by the residual mapping [61], we add a shortcut connection from the input directly to the output. Intuitively, to refine the embedding, it is much easier to add in fine-scale details via residual \( M - X \), where the shortcut connection enables an identity mapping from \( X \) to \( M \). Compared with regular BRNN, the identity mapping provides a good initialization in the output space that is much more likely to be closer to the optimal solution. Therefore learning would be more effective. Our experiment in Section IV proves that ResBRNN achieves significant performance improvement. To enable this, we re-write the GRU formula in [1] into a compact form:

\[
\begin{align*}
    h_t &= GRU(x_t, h_{t-1}; W) \\
    m_t &= x_t + h_t^f + h_t^b
\end{align*}
\]

where \( f \) denotes the forward pass and \( b \) denotes the backward pass. The parameters \( W \) for the two passes are shared to reduce the number of parameters.

During training, we minimize a video-story pairwise ranking loss:

\[
\begin{align*}
    \min_{\theta} \sum_v \sum_k \max\{0, \beta - s(m, v) + s(m, v_k)\} \\
    + \lambda \sum_v \sum_k \max\{0, \beta - s(v, m) + s(v, v_k)\}
\end{align*}
\]

(3)

\( v \) and \( m \) are paired sentence and clip embeddings from the same story (or video), while \( v_k \) (or \( m_k \)) are the negative paired sample to \( m \) (or \( v \)). \( \theta \) include the parameters of ResBRNN and the parameters of the encoders \( \theta_{en} \). \( \beta \) denotes the margin. We set this margin to be larger than \( \alpha (\beta = 0.2) \) to apply harder constraints on the embeddings.

**Optimization** We first optimize the clip-sentence ranking loss (2) until validation error stops decreasing. Then we optimize the video-story ranking loss (4). We use ADAM [62] optimizer with a first momentum coefficient of 0.8 and a second momentum coefficient of 0.999.

**B. Narrator Network**

The goal of the Narrator is to take in a long video and output a sequence of important video clips that form a story. The model is formulated as a reinforcement learning agent that interacts with a video over time. The video is first processed by the video encoder CNN+RNN that has been trained following the method described in Section III-A. The video encoder produces a sequence of features \( X = \{x_1, x_2, \ldots, x_N\} \) (300-d hidden state) for \( N \) frames, where the frames are sampled on every tenth frame from the video. At timestep \( n \), \( x_n \) contains semantic information of the current and all previous frames. The Narrator sequentially observes \( X \), and decides both when to sample a clip and the length of the sampled clip. An overview of the model is shown in Fig. 3. We now describe the Narrator in detail.
**Candidate Gate** At each timestep $n$, a binary gate $R_n$ is used to decide whether the current position is a candidate to sample a clip. The gate is defined as

$$R_n(x_n, x_p) = \begin{cases} 
1 & \text{if } s(x_n, x_p) > \tau \\
0 & \text{otherwise} 
\end{cases}$$

(5)

where $s(x_n, x_p) = x_n \cdot x_p$ is the similarity score for the normalized frame embeddings of the current position $n$ and the previous sample position $p$. The candidate gate serves as an attention mechanism. It rejects the current frame if it is semantically similar with the previous sampled frame. It enforces succinctness and diversity to the story, and also makes training easier by reducing the size of the agent’s possible action space. In this work, we set the threshold $\tau$ to be 0.7.

**Clip Indicator** If $R_n = 1$ at timestep $n$, the current frame embedding $x_n$ is processed by a function $f_c(x_n; \theta_c)$ to get the clip indicator $c_n$, which is a binary value that signals whether a clip should be sampled at the current position. $f_c(x_n; \theta_c)$ is a function learned from data that decides how important the current frame is for a story. During training, $c_n$ is sampled from a Bernoulli distribution parameterized by $f_c(x_n; \theta_c)$ defined as

$$f_c(x_n; \theta_c) = \max(0, \sigma(W_c x_n) + b_c)$$

(6)

where $\theta_c = \{W_c, b_c\}$, $\sigma$ denotes the Sigmoid function, $W_c$ denotes the weights for a fully-connected layer, and $b_c$ is a scalar value to offset the probability distribution. Since our goal is to generate succinct rather than dense descriptions, the clips should be sampled sparsely from the video. Hence we initialize $b_c$ to be $-0.4$, so that $f_c(x_n; \theta_c)$ is more biased towards zero.

At test time, $c_n$ is computed as

$$c_n = \begin{cases} 
1 & \text{if } f_c(x_n; \theta_c) > \epsilon \\
0 & \text{otherwise} 
\end{cases}$$

(7)

where $\epsilon$ is a threshold that controls the succinctness of the story. In our experiment, we set $\epsilon = 0.2$ based on performance on the validation set.

**Clip Length** If $c_n$ is set to 1, the agent will sample a clip centered at the current position with length $l_n$, which refers to the number of frames (note that the frames are sampled on every tenth frame from the video). $l_n$ is computed by a function $f_l(x_n; \theta_l) = \kappa \cdot \sigma(W_l x_n + b_l)$, where $\kappa$ is a scaling constant and is set to 40. During training, $l_n$ is stochastically sampled from a Gaussian distribution with a mean of $f_l(x_n; \theta_l)$ and a fixed variance. At test time, $l_n = f_l(x_n; \theta_l)$.

**Storytelling** As shown in Fig. 3, we take all the clips sampled from a video and compute their context-aware semantic embeddings $m$ following the method in Section III-A. Meanwhile, we compute the sentence semantic embeddings $v$ for all candidate sentences. Then we generate the story by retrieving a set of sentences that best match the sampled clips in the embedding space. Note that the candidate pool consists of all sentences from the training set.
### Table I: Comparison of Video Story with existing video description visual story datasets. Note that SIND [18] is an image dataset, and * marks the number of albums

| Dataset               | Domain     | Num. videos | Avg. video length | Avg. text length | Avg. sent length | Vocabulary |
|-----------------------|------------|-------------|-------------------|------------------|------------------|------------|
| MSR-VTT [63]          | open       | 7180        | 20.66s            | 9.6              | 9.6              | 24,549     |
| TaCos M-L [3]         | cooking    | 185         | 5m 5s             | 115.9            | 8.2              | 2,833      |
| ActivityNet Captions [5] | open    | 20k         | 180s              | 52.5             | 13.5             |            |
| SIND [18]             | Flickr album | 10,117* | 5 images | 51.0            | 10.2             | 18,200     |
| VideoSet [40]         | TV & egocentric | 11     | 3h 4f1m        | 376.2            | 8.6              | 1,195      |
| Video Story           | open       | 105         | 12m 35s          | 162.6            | 12.1             | 4,045      |

### C. Narrator Training

There are two challenges for learning effective clip selection policy. First, it is difficult to visually evaluate the clip proposals [46]. Second, the sentences retrieved from training set can introduce bias to the story. To address those challenges, we learn the clip selection policy by directly optimizing the generated story in text domain. Inspired by [53], [54], we evaluate the story with NLP metrics such as BLEU [64], METEOR [65] or CIDEr [66]. Optimizing on NLP metrics has two advantages: (1) it is more practical, well-defined and semantically-meaningful compared with optimizing on the visual content. (2) the Narrator takes account of the bias from the candidate sentences.

The training of Narrator is a non-differentiable process because (1) the NLP metrics are discrete and non-differentiable, and (2) the clip indicator and clip length outputs are non-differentiable components. We take advantage of the REINFORCE algorithm [49] that enables learning of non-differentiable models. We first briefly describe REINFORCE below. Then we introduce a variance-reduced reward to learn effective clip selection policy.

**REINFORCE** Given a space of possible action sequences $A$, the policy of the agent, in our case $f$, and $f_1$ induces a distribution $p_\theta(a)$ over $a \in A$ parameterized by $\theta = \{\theta, \theta_1\}$. The objective of REINFORCE is to maximize the expected reward $J(\theta)$ defined as

$$J(\theta) = \sum_{a \in A} p_\theta(a) r(a)$$  \hspace{1cm} (8)

where $r(a)$ is a reward assigned to each individual action sequence. The gradient of the objective is

$$\nabla J(\theta) = \sum_{a \in A} p_\theta(a) \nabla \log p_\theta(a) r(a)$$  \hspace{1cm} (9)

Maximizing $J$ is non-trivial due to the high-dimensional space of possible action sequences. REINFORCE addresses this by approximating the gradient equation with Monte Carlo sampling:

$$\nabla J(\theta) \approx \frac{1}{M} \sum_{i=1}^{M} \sum_{n=1}^{N} \nabla \log p_\theta(a_{1:n}^i, x_{1:n}^i, a_{1:n-1}^i) R^i$$  \hspace{1cm} (10)

where $\pi_\theta$ is the agent’s current policy. At time step $n$, $a_n$ is the policy’s current action (i.e. clip indicator $c_n$ and clip length $l_n$), $x_{1:n}$ are the past states of the environment including the current (frame embeddings), and $a_{1:n-1}$ are the past actions. $R^i$ is the reward received by running current action sequence to the end. The approximate gradient is computed by running the agent’s current policy for $M$ episodes.

In this work, we use the CIDEr score for the generated story as the reward. Empirically, we find that optimizing over CIDEr achieves the best overall performance compared with other metrics, which has also been observed by [52], [54].

**Variance-reduced reward** The gradient estimate in (10) may have high variance. Therefore it is common to use a baseline reward $b$ to reduce the variance, and so the gradient equation becomes:

$$\nabla J(\theta) \approx \frac{1}{M} \sum_{i=1}^{M} \sum_{n=1}^{N} \nabla \log p_\theta(a_{1:n}^i, x_{1:n}^i, a_{1:n-1}^i)(R^i - b)$$  \hspace{1cm} (11)

The baseline reward is often estimated with a separate network [51], [52], [53], [55]. For our model, there are two drawbacks of using a separate network for baseline estimation. First, it introduces extra parameters to be learned. Second, a baseline estimator that only operates on the visual input $x$ does not account for the bias introduced by the candidate sentences. Therefore, we propose a greedy approach that computes a different baseline for each video. For each video, we first randomly sample a set of clips, and retrieval a story using those clips. Then we repeat the above procedure $K$ times, and take the average of the $K$ stories’ CIDEr score as the baseline reward for that video ($K = 10$). In this way, a policy receives a positive reward if it performs better than a random policy, and negative otherwise.

### IV. Dataset

Since video storytelling is a new problem, we have collected a Video Story dataset to enable research in this direction. We choose four types of common and complex events (i.e. birthday, camping, Christmas and wedding), and use keyword search to download videos from Youtube. Then we manually selected 105 videos with sufficient inter-event and intra-event variation. The stories are collected via Amazon Mechanical Turk. For each video we asked crowd workers to write their stories following three rules: (1) The story should have at least 8 sentences. (2) Each sentence should have at least 6 words. (3) The story should be coherent and relevant to the event. We then asked workers to label the start and end time in the video where each sentence occurred. Each video is annotated by at least 5 different workers. In total, we collected 529 stories.

Table II shows the statistics of the Video Story dataset in comparison with existing datasets. Compared with video captioning datasets (i.e. MSR-VTT [63], TaCos M-L [3], ActivityNet Captions [5]) and album story dataset (i.e. SIND [18]),
Video Story dataset has longer videos (12 min 35 sec in average) and longer descriptions (162.6 words in average). Moreover, the sentences in Video Story are more sparsely distributed across the video (55.77 sec per sentence). Compared with VideoSet [46] for video summarization study that only has 11 videos, Video Story dataset has more videos in open domain, the sentences are more diverse, and the vocabulary size is also larger.

V. EXPERIMENT

We compare the proposed approach with state-of-the-art methods and variations of the proposed model using quantitative measures and user study. Experiments are performed on the Video Story dataset. We randomly split 70% (73 videos) as training set, 15% (16 videos) as validation set and the others (16 videos with 4 per event category) as test set. We perform different tasks to validate the proposed multimodal embedding learning framework and the Narrator model.

A. Multimodal Embedding Evaluation

1) Sentence retrieval task: The goal of this task is to retrieve a sequence of sentences given a sequence of clips as query. We consider a withheld test set, and evaluate the median rank of the closest ground truth (GT) sentence and Recall@K, which measures the fraction of times a GT sentence was found among the top K results. Since the temporal location of clips are sparse in the Video Story dataset, overlapping clips usually share similar description. Therefore for a clip, we also include the paired sentences for its overlapping clips into the GT sentences. In average, each clip has 4.44 GT sentences.

In this work, we compare with the following baseline methods to fully evaluate our model.

- **Random**: A naive baseline where sentences are randomly ranked.
- **Category Random**: Only sentences from the same category as the test clip are considered candidates in the random ranking.
- **CCA**: Canonical Correlation Analysis (CCA) has been used to build an image-text space [67]. We use CCA to learn a joint video-sentence space. The video feature is the average of frame-level ResNet outputs, and the sentence is represented by the average of Word2Vecs. Then we project clips and sentences into the joint space for ranking.
- **m-RNN** [21]: m-RNN consists of two sub-networks that encode sentence and image respectively, which interact with each other in a multimodal layer. To apply m-RNN for video data, we replace the original CNN image encoder with mean-pooling of frame-level ResNet outputs.
- **Xu et al.** [35]: This is a sentence-video joint embedding model, where sentences are embedded with a dependency-tree structure model. We replace the original CNN with ResNet.
- **EMB**: This is the first local step of our embedding learning framework. Sentences are ranked based on their distance from the video in the embedding space. Note that EMB is similar to the image-sentence ranking model in [10], [57] and the movie-text ranking model in [36].
- **BRNN**: We use a BRNN in the second step to model the context, which makes it a variant to our model without the residual mapping.
- **ResBRNN**: The proposed context-aware multimodal embedding learning framework.

The results are shown in Table II. Our local embedding learning method (EMB) is a strong baseline that outperforms existing methods. The proposed context-aware embedding learning framework (ResBRNN) further improves the performance by a large margin. The improvement suggests that by incorporating information from contextual clips, the model can learn temporal coherent embeddings that are more representative of the underlying storyline. BRNN without residual mapping performs worse than ResBRNN and EMB, which demonstrates the importance of residual mapping.

| Method       | R@1 | R@5 | R@10 | Medr |
|--------------|-----|-----|------|------|
| Random       | 0.31| 1.38| 3.98 | 215  |
| Category Random | 1.18| 5.47| 16.03| 57   |
| CCA          | 2.37| 12.83| 23.25| 46   |
| Xu et al. [35]| 4.72| 19.85| 35.46| 31   |
| m-RNN [21]   | 5.34| 21.23| 39.02| 29   |
| EMB          | 5.50| 22.02| 40.52| 27   |
| BRNN         | 5.50| 20.26| 36.39| 29   |
| ResBRNN      | 7.44| 25.77| 46.41| 22   |

2) Story generation task: In this task, we generate stories for test videos with different methods. In order to evaluate our embedding learning framework, we need to first fix on a good sequence of clips to represent the story visually. We employ a ‘Pseudo-GT’ clip selection scheme. For each test video, we compare a human annotator’s selected clips with other annotators’, and choose the sequence of clips that have the largest overlap with others, where overlap is computed as intersection-over-union (IoU). For quantitative measures, we compare the generated story with reference stories and compute NLP metrics of language similarity (i.e. BLEU-N [64], ROUGE-L [68], METEOR [65] and CIDEr [66]) using MSCOCO evaluation code [69]. We would like to first clarify that each of the above metrics has its own strengths and weaknesses [70]. For example, BLEU-N metric computes an n-gram based precision, thus it favors short and generic descriptions, and may not be a good measure for fully evaluating the semantic similarity of long paragraphs. CIDEr and METEOR have shown reasonable correlation with human judgments in image captioning [65], [70]. Therefore, we consider those two metrics as primary indicators of a model’s performance. Nonetheless, we report results for all metrics.

For this task, we compare against the baselines in sentence retrieval task as well as two state-of-the-art generative models for video captioning and video paragraph generation. The additional baselines are as follow.

- **S2VT** [28]: This model encodes each clip with a CNN+RNN model and decodes the sentence with another
| Method         | Model Type | CIDEr | METEOR | ROUGE-L | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 |
|---------------|------------|-------|--------|---------|--------|--------|--------|--------|
| S2VT*         | generative | 64.0  | 14.3   | 28.6    | 63.3   | 40.6   | 24.6   | 15.4   |
| H-RNN*        | generative | 64.6  | 15.5   | 28.8    | 61.6   | 41.4   | 26.3   | 16.1   |
| Random        | retrieval  | 30.2  | 13.1   | 21.4    | 43.1   | 23.1   | 10.0   | 4.8    |
| CCA           | retrieval  | 71.8  | 16.5   | 26.7    | 60.1   | 34.7   | 11.8   | 10.1   |
| Xu et al.     | retrieval  | 79.5  | 17.7   | 28.0    | 61.7   | 36.4   | 20.2   | 11.5   |
| m-RNN         | retrieval  | 81.2  | 18.0   | 28.5    | 61.9   | 37.0   | 21.1   | 11.8   |
| EMB           | retrieval  | 88.8  | 19.1   | 28.9    | 64.5   | 39.3   | 22.7   | 13.4   |
| BRNN          | retrieval  | 81.0  | 18.1   | 28.3    | 61.4   | 36.6   | 20.3   | 11.3   |
| ResBRNN       | retrieval  | 94.3  | 19.6   | 29.7    | 66.0   | 41.7   | 24.3   | 14.7   |
| ResBRNN-kNN   | retrieval  | 103.6 | 20.1   | 29.9    | 69.1   | 43.5   | 26.1   | 15.6   |

**Fig. 4:** Example stories generated by the proposed method (ResBRNN-kNN+Narrator), two baselines (H-RNN and EMD), and a ground truth. The frames are handpicked from the set of clips proposed by Narrator and the ground truth example. **Green** boxes highlight the frames from GT, **orange** boxes highlight the frames from Narrator proposals, while **red** boxes highlight frames shared by both.
TABLE IV: Evaluation of story generation on Video Story test set with different clip selection methods and fixed story retrieval method

| Method               | CIDEr | METEOR | ROUGE-L | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 |
|----------------------|-------|--------|---------|--------|--------|--------|--------|
| Uniform              | 89.9  | 18.3   | 28.0    | 65.7   | 40.6   | 23.2   | 13.2   |
| SeqDPP [71]          | 91.6  | 18.3   | 28.3    | 66.3   | 41.0   | 23.6   | 13.1   |
| Submodular [13]      | 92.0  | 18.4   | 28.1    | 66.4   | 41.0   | 23.8   | 13.3   |
| vsLSTM [72]          | 92.4  | 18.2   | 28.2    | 66.6   | 41.5   | 24.1   | 13.6   |
| Narrator IoU         | 93.7  | 18.6   | 28.2    | 67.9   | 42.1   | 24.7   | 14.1   |
| Narrator w/o x       | 93.4  | 18.5   | 28.5    | 67.3   | 41.6   | 24.5   | 14.2   |
| Narrator w/o f1      | 96.1  | 19.0   | 29.1    | 68.6   | 42.9   | 25.2   | 14.5   |
| Narrator             | 98.4  | 19.6   | 29.5    | 69.1   | 43.0   | 25.3   | 15.0   |

RNN.

- **H-RNN [4]**: This model is designed for generating multiple sentences, where it uses previous sentences to initialize the hidden state for next sentence generation.

- **ResBRNN-kNN**: ResBRNN retrieves the best sentence for each clip in a sequence, which may result in duplicate sentences. To increase diversity, we apply k-nearest search. For each clip, we first find its k-nearest sentences. Then we consider the entire sequence, and find the best sequence of non-duplicate sentences that have the minimum total distance with the clips. We use $k = 4$ in our experiment.

Table III shows the results for story generation with fixed clip selection. Retrieval based methods generally outperform generative methods, because generative methods tend to give short, repetitive and low-diversity descriptions. Among the retrieval methods, we observe similar trend of performance improvement as in Table II. ResBRNN-kNN further improves upon ResBRNN by increasing story diversity. H-RNN has higher BLEU-3 and BLEU-4 score because the BLEU metric favors generic descriptions. Qualitative examples are shown in Fig. 4.

### B. Narrator Evaluation

Here we evaluate the Narrator model for clip selection. Since our end goal is video storytelling, we directly evaluate the NLP metrics of the story retrieved from different clip proposals, which is a more well-defined and objective evaluation compared with evaluating the overlap of visual content. We fix the story retrieval method with ResBRNN-kNN, and compare Narrator with multiple baseline methods for clip selection, as described below.

- **Uniform**: As a naive baseline, we uniformly sample 12 clips per test video, while 12 approximates the average number of sentences per story.

- **SeqDPP [71]**: A probabilistic model for diverse subshot selection that learns from human-created summaries.

- **Submodular [13]**: A supervised approach for video summarization, which jointly optimizes for multiple objectives including Interestness, Representativeness and Uniformity.

- **vsLSTM [72]**: vsLSTM is a bi-directional LSTM network proposed for video summarization. The inputs are frame-level CNN features, in our case ResNet outputs. The outputs are frame-level importance scores, which can be used to select clips following the approach in [72]. We compute the GT frame-level importance scores using the same approach as [72], where a frame is considered more important if it has been selected by more human annotators.

In addition to the above baseline methods, we evaluate several variants of the proposed Narrator as ablation study to understand the efficacy of each components.

- **Narrator IoU**: Instead of using the CIDEr score of the story as reward, we compute the overlap (IoU) between the Narrator’s clip proposal and the annotators’ selected clips as reward.

- **Narrator w/o x**: A variant of the proposed method. Inputs are frame-level CNN features instead of the semantic embedding $x$.

- **Narrator w/o $f_1$**: A variant of the proposed method without the component $f_1$ to decide clip length. Each clip has a fixed length of 20.

Table IV shows the evaluation of the retrieved story with various clip selection methods. The proposed Narrator model outperforms all other methods. The improvement over Narrator w/o $x$ indicates the representative power of the video semantic embedding, and the improvement over Narrator w/o $f_1$ suggests that it is useful to have a parameterized clip length compared with fixed. The state-of-the-art video summarization methods (i.e. Submodular [13], SeqDPP [71] and vsLSTM [72]) only perform slightly better than uniform sampling. This is because the human-selected clips among different annotators have high variance, which makes frame-level supervision a weak supervision to learn from. On the other hand, the stories written by annotators are more congruent, because different set of clips can still convey similar stories, thus it is stronger supervision. For the same reason, using IoU as reward for the Narrator does not perform well.

Note that the performance gaps between clip selection methods are relatively small compared with that of Table III. This is because the ResBRNN-kNN method used in this task can already retrieve relevant and meaningful sentences for a video, hence the clip selection methods would make more subtle improvements on the quality of the story.

### C. User Study

We perform user studies using Amazon Mechanical Turk to observe general users’ preference on stories generated by four methods: (a) the proposed ResBRNN-kNN with Narrator
TABLE V: User study results. Numbers indicate the percentage of pairwise preference

| Compared methods          | Percentage of preference |
|---------------------------|-------------------------|
| EMB > H-RNN               | 70.0%                   |
| ResNarrator > H-RNN       | 88.8%                   |
| ResNarrator > EMB         | 80.0%                   |
| ResNarrator > GT          | 38.1%                   |

(ResNarrator), (b) EMB with Pseudo-GT clips , (c) H-RNN with Pseudo-GT clips and (d) GT story. The results are shown in Table V. Please see Fig. 3 for qualitative examples of the stories.

First, we compare stories generated by the ResNarrator with ones generated by the two baseline methods EMB and H-RNN. For each of the 16 videos in the test set, we ask 10 users to watch the video, read the three stories presented in random order, and rank the stories based on how well they describe the video. On average, 80.0% users prefer ResNarrator over EMB, and 88.8% prefer ResNarrator over H-RNN. which validate that the proposed method generate better stories compared with the baselines.

Then, we ask the user to do a pairwise selection between stories from ResNarrator and GT. 38.1% users prefer stories generated by ResNarrator over GT stories, which further shows its efficacy.

VI. CONCLUSION

In this work, we have studied the problem of video storytelling. To address the challenges posed by the diversity of the story and the length and complexity of the video, we propose a ResBRNN model for context-aware multimodal embedding learning, and a Narrator model directly optimized on the story for clip selection. We evaluate our method on a new Video Story dataset, and demonstrate the efficacy of the proposed method both quantitatively and qualitatively. One limitation of our method is that the story is limited by sentences in the training set. For future work, we intend to utilize sentences in-the-wild to further improve the diversity of the story. We also intend to explore NLP based methods to refine the story with smoother transition between sentences. Furthermore, we plan to incorporate recent advances in text representations and hard negatives into our framework.

ACKNOWLEDGMENT

This research is supported by the National Research Foundation, Prime Minister’s Office, Singapore under its Strategic Capability Research Centres Funding Initiative.

REFERENCES

[1] J. Krause, J. Johnson, R. Krishna, and L. Fei-Fei, “A hierarchical approach for generating descriptive image paragraphs,” in CVPR, 2017, pp. 317–325.
[2] X. Liang, Z. Hu, H. Zhang, C. Gan, and E. P. Xing, “Recurrent topic-transition GAN for visual paragraph generation,” in ICCV, 2017, pp. 3362–3371.
[3] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Scheibe, “Coherent multi-sentence video description with variable level of detail,” in German Conference on Pattern Recognition, 2014, pp. 184–195.
[4] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, “Video paragraph captioning using hierarchical recurrent neural networks,” in CVPR, 2016, pp. 4584–4593.
[5] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles, “Dense-captioning events in videos,” in ICCV, 2017, pp. 706–715.
[6] N. Xu, A. Liu, Y. Wong, Y. Zhang, W. N. Y. Su, and M. S. Kankanhalli, “Dual-stream recurrent neural network for video captioning,” IEEE Transactions on Circuits and Systems for Video Technology, 2018.
[7] A. Liu, Y. Qiu, Y. Wong, Y. Su, and M. S. Kankanhalli, “A fine-grained spatial-temporal attention model for video captioning,” IEEE Access, vol. 6, pp. 68463–68471, 2018.
[8] A. Liu, N. Xu, Y. Wong, J. Li, Y. Su, and M. S. Kankanhalli, “Hierarchical & multimodal video captioning: Discovering and transferring multimodal knowledge for vision to language,” Computer Vision and Image Understanding, vol. 163, pp. 113–125, 2017.
[9] C. C. Park and G. Kim, “Expressing an image stream with a sequence of natural sentences,” in Advances in Neural Information Processing Systems, 2015, pp. 73–81.
[10] Y. Liu, J. Fu, T. Mei, and C. W. Chen, “Let your photos talk: Generating narrative paragraph for photo stream via bidirectional attention recurrent neural networks,” in AAAI, 2017, pp. 1445–1452.
[11] F. Dufaux, “Key frame selection to represent a video,” in ICIP, 2000, pp. 275–278.
[12] D. B. Goldman, B. Curless, D. Salesin, and S. M. Seitz, “Schematic storyboarding for video visualization and editing,” ACM Transactions on Graphics, vol. 25, no. 3, pp. 862–871, 2006.
[13] M. Gygli, H. Grabner, and L. J. V. Gool, “Video summarization by learning submodular mixtures of objectives,” in CVPR, 2015, pp. 3090–3098.
[14] Z. Lu and K. Grauman, “Story-driven summarization for egocentric video,” in CVPR, 2013, pp. 2714–2721.
[15] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, “Category-specific video summarization,” in ECCV, ser. Lecture Notes in Computer Science, vol. 8694, 2014, pp. 540–555.
[16] K. Zhang, W. Chao, F. Sha, and K. Grauman, “Summary transfer: Exemplar-based subset selection for video summarization,” in CVPR, 2016, pp. 1059–1067.
[17] Y. Song, J. Vallmitjana, A. Stent, and A. James, “Tvsum: Summarizing web videos using titles,” in CVPR, 2015, pp. 5179–5187.
[18] F. Huang, Ting-Hao K.and Ferraro, N. Mostafazadeh, I. Misra, J. Devlin, A. Agrawal, R. Girshick, X. He, P. Kohli, D. Batra et al., “Visual storytelling,” in NAACL, 2016.
[19] B. Dai, D. Lin, R. Urtasun, and S. Fidler, “Towards diverse and natural image descriptions via a conditional GAN,” in ICCV, 2017, pp. 2970–2979.
[20] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, “A diversity-promoting objective function for neural conversation models,” in NAACL HLT, 2016, pp. 110–119.
[21] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille, “Deep captioning with multimodal recurrent neural networks (m-RNN),” in ICLR, 2015.
[22] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in ICML, 2015, pp. 2048–2057.
[23] A. Karpathy and F. Li, “Deep visual-semantic alignments for generating image descriptions,” in CVPR, 2015, pp. 3128–3137.
[24] J. Li, Y. Song, Q. Zhao, and M. S. Kankanhalli, “Attention transfer from web images for video recognition,” in ACM Multimedia, 2017, pp. 1–9.
[25] L. Yao, A. Torabi, K. Cho, N. Ballas, C. J. Pal, H. Larochelle, and A. C. Courville, “Describing videos by exploiting temporal structure,” in ICCV, 2015, pp. 4507–4515.
[26] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, “Dual-glance model for deciphering social relationships,” in ICCV, 2017, pp. 2669–2678.
[27] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. J. Mooney, and K. Saenko, “Translating videos to natural language using deep recurrent neural networks,” in NAACL HLT, 2015, pp. 1494–1504.
[28] S. Venugopalan, M. Rohrbach, J. Donahue, R. J. Mooney, T. Darrell, and K. Saenko, “Sequence to sequence - video to text,” in ICCV, 2015, pp. 4534–4542.
[29] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” IEEE Transactions on Pattern Analysis and Machine Learning, vol. 39, no. 4, pp. 677–691, 2017.
[30] J. Li, Y. Song, Q. Zhao, and M. S. Kankanhalli, “Unsupervised learning of view-invariant action representations,” in Advances in Neural Information Processing Systems, 2018, pp. 1260–1270.
