Abstract

Generating multi-sentence descriptions for videos is one of the most challenging captioning tasks due to its high requirements for not only visual relevance but also discourse-based coherence across the sentences in the paragraph. Towards this goal, we propose a new approach called Memory-Augmented Recurrent Transformer (MART), which uses a memory module to augment the transformer architecture. The memory module generates a highly summarized memory state from the video segments and the sentence history so as to help better prediction of the next sentence (w.r.t. coreference and repetition aspects), thus encouraging coherent paragraph generation. Extensive experiments, human evaluations, and qualitative analyses on two popular datasets ActivityNet Captions and YouCookII show that MART generates more coherent and less repetitive paragraph captions than baseline methods, while maintaining relevance to the input video events.

1 Introduction

In video captioning, the task is to generate a natural language description capturing the content of a video. Recently, dense video captioning (Krishna et al., 2017) has emerged as an important task in this field, where systems first generate a list of temporal event segments from a video, then decode a coherent paragraph (multi-sentence) description from the generated segments. Park et al. (2019) simplify this task as generating a coherent paragraph from a provided list of segments, removing the requirements for generating the event segments, and focusing on decoding better paragraph captions from the segments. As noted by Xiong et al. (2018); Park et al. (2019), generating paragraph descriptions for videos can be very challenging due to the difficulties of having relevant, less redundant, as well as coherent generated sentences.

Towards this goal, Xiong et al. (2018) proposed a variant of the LSTM network (Hochreiter and Schmidhuber, 1997) that generates a new sentence conditioned on previously generated sentences by passing the LSTM hidden states throughout the entire decoding process. Park et al. (2019) further augmented the above LSTM caption generator with a set of three discriminators that score generated sentences based on defined metrics, i.e., relevance, linguistic diversity, and inter-sentence coherence. Though different, both these methods use LSTMs as the language decoder.

Recently, transformers (Vaswani et al., 2017) have proven to be more effective than RNNs (e.g., LSTM (Hochreiter and Schmidhuber, 1997), GRU (Chung et al., 2014), etc.), demonstrating superior performance in many sequential modeling tasks (Vaswani et al., 2017; Zhou et al., 2018; Devlin et al., 2019; Dai et al., 2019; Yang et al., 2019). Zhou et al. (2018) first introduced the transformer model to the video paragraph captioning task, with a transformer captioning module decoding natural language sentences from encoded video segment representations. This transformer captioning model is essentially the same as the original transformer (Vaswani et al., 2017) for machine translation, except that it takes a video representation rather than a source sentence representation as its encoder input. However, in such design, each video segment caption is decoded individually without knowing the context (i.e., previous video segments and the captions that have already been generated), thus often leading to inconsistent and redundant sentences w.r.t. previously generated sentences (see Figure 3 for examples). Dai et al. (2019) recognize this problem as context fragmentation in
the task of language modeling, where the transformers are operating on separated fixed-length segments, without any information flow across segments. Therefore, to generate more coherent video paragraphs, it is imperative to build a model that can span over multiple video segments and capture longer range dependencies.

Hence, in this work, we propose the Memory-Augmented Recurrent Transformer (MART) model (see Section 3 for details), a transformer-based model that uses a shared encoder-decoder architecture augmented with an external memory module to enable the modeling of the previous history of video segments and sentences. Compared to the vanilla transformer video paragraph captioning model (Zhou et al., 2018), our first architecture change is the unified encoder-decoder design, i.e., the encoder and decoder in MART use shared transformer layers rather than separated as in Zhou et al. (2018); Vaswani et al. (2017). This unified encoder-decoder design is inspired by recent transformer language models (Devlin et al., 2019; Dai et al., 2019; Sun et al., 2019) to prevent overfitting and reduce memory usage. Additionally, the memory module works as a memory updater that updates its memory state using both the current inputs and previous memory state. The memory state can be interpreted as a container of the highly summarized video segments and caption history information. At the encoding stage, the current video segment representation is enhanced with the memory state from the previous step using cross-attention (Vaswani et al., 2017). Hence, when generating a new sentence, MART is aware of the previous contextual information and can generate paragraph captions with higher coherence and lower repetition.

Transformer-XL (Dai et al., 2019) is a recently proposed transformer language model that also uses recurrence, and is able to resolve context fragmentation for language modeling (Dai et al., 2019). Different from MART that uses a highly-summarized memory to remember history information, Transformer-XL directly uses hidden states from previous segments. We modify the Transformer-XL framework for video paragraph captioning and present it as an additional comparison. We benchmark MART on two standard datasets: ActivityNet Captions (Krishna et al., 2017) and YouCookII (Zhou et al., 2017). Both automatic evaluation and human evaluation show that MART generates more satisfying results than previous LSTM-based approaches (Xiong et al., 2018; Zhou et al., 2019; Zhang et al., 2018) and transformer-based approaches (Zhou et al., 2018; Dai et al., 2019). In particular, MART can generate more coherent (e.g., coreference and order), less redundant paragraphs without losing paragraph accuracy (visual relevance).

2 Related Work

Video Captioning Recently, video captioning has attracted much attention from both the computer vision and the natural language processing community. Methods for the task share the same intrinsic nature of taking a video as the input and outputting a language description that can best describe the content, though they differ from each other on whether a single sentence (Wang et al., 2019; Xu et al., 2016; Chen and Dolan, 2011; Pasunuru and Bansal, 2017a) or multiple sentences (Rohrbach et al., 2014; Krishna et al., 2017; Xiong et al., 2018; Zhou et al., 2018; Gella et al., 2018; Park et al., 2019) are generated for the given video. In this paper, our goal falls into the category of generating a paragraph (multiple sentences) conditioned on an input video with several pre-defined event segments.

One line of work (Zhou et al., 2018, 2019) addresses the video paragraph captioning task by decoding each video event segment separately into a sentence. The final paragraph description is obtained by concatenating the generated single sentence descriptions. Though individual sentences may precisely describe the corresponding event segments, when put together the sentences often become inconsistent and redundant. Another line of works (Xiong et al., 2018; Gella et al., 2018) use the LSTM decoder’s last (word) hidden state from the previous sentence as the initial hidden state for the next sentence decoding, thus enabling information flow from previous sentences to subsequent sentences. While these methods have shown better performance than their single sentence counterpart, they are still undesirable as the sentence-level recurrence is achieved at word-level, and the context history information quickly decays due to vanishing gradients (Pascanu et al., 2013) problem. Additionally, these designs also have difficulty modeling long-term dependencies (Hochreiter et al., 2001). In comparison, the recurrence in MART resides in the sentence or segment level and is thus more robust to the aforementioned problems. AdvInf (Park...
et al., 2019) augments the above LSTM word-level recurrence methods with adversarial inference, using a set of separately trained discriminators to re-rank the generated sentences. The techniques in AdvInf can be viewed as an orthogonal way of generating captions with better quality.

Transformers Transformer (Vaswani et al., 2017) is used as the basis of our approach. Different from RNNs (e.g., LSTM (Hochreiter and Schmidhuber, 1997), GRU (Chung et al., 2014), etc) that use recurrent structure to model long-term dependencies, transformer relies on self-attention to learn the dependencies between input words. Transformers have proven to be more efficient and powerful than RNNs, with superior performance in many sequential modeling tasks, including machine translation (Vaswani et al., 2017), language modeling/pre-training (Devlin et al., 2019; Dai et al., 2019; Yang et al., 2019) and multi-modal representation learning (Tan and Bansal, 2019; Chen et al., 2019; Sun et al., 2019). Additionally, Zhou et al. (2018) have shown that a transformer model can generate better captions than the LSTM model. However, transformer architectures are still unable to model history information well. This problem is identified in the task of language modeling as context fragmentation (Dai et al., 2019), i.e., each language segment is modeled individually without knowing its surrounding context, leading to inefficient optimization and inferior performance. To resolve this issue, Transformer-XL (Dai et al., 2019) introduces the idea of recurrence to the transformer language model. Specifically, the modeling of a new language segment in Transformer-XL is conditioned on hidden states from previous language segments. Experimental results show Transformer-XL has stronger language modeling capability than the non-recurrent transformer. Transformer-XL directly uses all the hidden states from the previous segment to enable recurrence. In comparison, our MART uses highly summarized memory states, making it more efficient in passing useful semantic or linguistic cues to future sentences.

3 Methods

Though our method provides a general temporal multi-modal learning framework, we focus on the video paragraph captioning task in this paper. Given a video \( V \), with several temporally ordered event segments \([e_1, e_2, ..., e_T]\), the task is to generate a coherent paragraph consisting of multiple sentences \([s_1, s_2, ..., s_T]\) to describe the whole video, where sentence \( s_t \) should describe the content in the segment \( e_t \). In the following, we first describe the baseline transformer that generates sentences without recurrent architecture, then introduce our approach – Memory-Augmented Recurrent Transformer (MART). Besides, we also compare MART with the recently proposed Transformer-XL (Dai et al., 2019) in detail.

3.1 Background: Vanilla Transformer

We start by introducing the vanilla transformer video paragraph captioning model proposed by Zhou et al. (2018), which is an application of the original transformer (Vaswani et al., 2017) model for video paragraph captioning. An overview of the model is shown in Figure 1. The core of the architecture is the scaled dot-product attention. Given query matrix \( Q \in \mathbb{R}^{T_q \times d_k} \), key matrix \( K \in \mathbb{R}^{T_v \times d_k} \) and value matrix \( V \in \mathbb{R}^{T_v \times d_v} \), the attentional output is computed as:

\[
A(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}}, \text{dim}=1 \right) V,
\]

where \( \text{softmax}(\cdot, \text{dim}=1) \) denotes performing softmax at the second dimension of the input. Combining \( h \) paralleled scaled dot-product attention,
we obtain the multi-head attention (Vaswani et al., 2017), we denote it as MultiHeadAttr($Q$, $K$, $V$). The attention formulation discussed above is quite general. It can be used for various purposes, such as self-attention (Vaswani et al., 2017) where query, key, and value matrix are all the same, and cross-attention (Vaswani et al., 2017) where the query matrix is different from the key and value matrix. In this paper, we also use multi-head attention for memory aggregation and update, as discussed later.

The vanilla transformer video paragraph captioning model has $N$ encoder layers and $N$ decoder layers. At the $l$-th encoder layer, the multi-head attention module takes the last layer’s hidden states $H^{l-1}$ as inputs and performs self-attention. The attentional outputs are then projected by a feed-forward layer. At the $l$-th decoder layer, the model first encodes the last decoder layer’s hidden states using masked multi-head attention. It then uses multi-head attention, with the masked outputs as query matrix, and the hidden states $H^{l}$ from $l$-th encoder layer as key and value matrix to gather information from the encoder side. Similarly, a feed-forward layer is used to encode the sentences further. Residual connection (He et al., 2016) and layer-normalization (Ba et al., 2016) are applied for each layer, for both encoder and decoder.

### 3.2 Memory-Augmented Recurrent Transformer

The vanilla transformer captioning model follows the classical encoder-decoder architecture, where the encoder and decoder network are separated. In comparison, the encoder and decoder are shared in MART, as shown in Figure 2 (left). The video and text inputs are firstly separately encoded and normalized. We denote the encoded video and text embeddings as $H^{0}_{video} \in \mathbb{R}^{T_{video} \times d}$ and $H^{0}_{text} \in \mathbb{R}^{T_{text} \times d}$, where $T_{video}$ and $T_{text}$ are the lengths of video and text, respectively. $d$ denotes the hidden size. We then concatenate these two embeddings as input to the transformer layers: $H^{0}=[H^{0}_{video}; H^{0}_{text}] \in \mathbb{R}^{T \times d}$, where $[;]$ denotes concatenation, $T= T_{video} + T_{text}$. This unified encoder-decoder design is inspired by recent works on multi-modal representation learning (Chen et al., 2019; Sun et al., 2019). We also use two trainable...
token type embedding vectors to indicate whether an input token is from video or text, similar to Devlin et al. (2019) where the token type embeddings are added to indicate different input sequences. We ignore the video token positions and only consider the text token positions when calculating loss and generating words.

While the aforementioned vanilla transformer is a powerful method, it is less suitable for video paragraph captioning due to its inability to utilize video segments and sentences history information. Thus, given the unified encoder-decoder transformer, we augment it with an external memory module, which helps it to utilize video segments and the corresponding caption history to generate the next sentence. An overview of the memory module is shown in Figure 2 (left). At step $t$, i.e., decoding the $t$-th video segment, the $l$-th layer aggregates the information from both its intermediate hidden states $\bar{H}_t^l \in \mathbb{R}^{T_l \times d}$ and the memory states $M_{l-1}^t \in \mathbb{R}^{T_m \times d}$ ($T_m$ denotes memory state length or equivalently #slots in the memory) from the last step, using a multi-head attention. The input query matrix of the multi-head attention $Q=\bar{H}_t^l$, key and value matrices are $K, V = [M_{l-1}^t; \bar{H}_t^l] \in \mathbb{R}^{(T_m+T_l) \times d}$. The memory augmented hidden states are further encoded using a feed forward layer and then merged with the intermediate hidden states $\bar{H}_t^l$ using a residual connection and layer norm to form the hidden states output $H_t^l \in \mathbb{R}^{T_l \times d}$. The memory state $M_{l-1}^t$ is updated as $M_t^l$, using the intermediate hidden states $\bar{H}_t^l$. This process is conducted in the Memory Updater module, illustrated in Figure 2. We summarize the procedure below:

$$S_t^l = \text{MultiHeadAtt}(M_{l-1}^t, \bar{H}_t^l, \bar{H}_t^l),$$
$$C_t^l = \tanh(W_{mc}^l M_{l-1}^t + W_{sc}^l S_t^l + b_c^l),$$
$$Z_t^l = \text{sigmoid}(W_{mz}^l M_{l-1}^t + W_{sz}^l S_t^l + b_z^l),$$
$$M_t^l = (1-Z_t^l) \odot C_t^l + Z_t^l \odot M_{l-1}^t,$$

where $\odot$ denotes Hadamard product, $W_{mc}^l, W_{sc}^l, W_{mz}^l, \text{and } W_{sz}^l$ are trainable weights, $b_c^l$ and $b_z^l$ are trainable bias. $C_t^l \in \mathbb{R}^{T_m \times d}$ is the internal cell state. $Z_t^l \in \mathbb{R}^{T_m \times d}$ is the update gate that controls which information to retain from the previous memory state, and thus reducing redundancy and maintaining coherence in the generated paragraphs.

This update strategy is conceptually similar to LSTM (Hochreiter and Schmidhuber, 1997) and GRU (Chung et al., 2014). It differs in that multi-head attention is used to encode the memory state and thus multiple memory slots are supported instead of a single one in LSTM and GRU, which gives it a higher capacity of modeling complex relations. Recent works (Sukhbaatar et al., 2015; Graves et al., 2014; Xiong et al., 2016a) introduce a memory component into neural networks, where the memory is mainly designed to memorize facts in the input context to support downstream tasks, e.g., copy (Graves et al., 2014) or question answering (Sukhbaatar et al., 2015; Xiong et al., 2016a). In comparison, the memory in MART is designed to memorize the sequence generation history to support the coherent generation of the next sequence.

### 3.3 Comparison with Transformer-XL

Transformer-XL (Dai et al., 2019) is a recently proposed transformer-based language model that uses a segment-level recurrence mechanism to capture the long-term dependency in context. In Figure 2 (right) we show a modified version of Transformer-XL for video paragraph captioning. At step $t$, at its $l$-th layer, Transformer-XL takes as inputs the last layer’s hidden states from both the current step and the last step, which we denote as $H_{l-1}^t$ and $SG(H_{l-1}^t)$, where $SG(\cdot)$ stands for stop-gradient, and is used to save GPU memory and computation (Dai et al., 2019). The input query matrix of the multi-head attention $Q=H_t^l$, key and value matrices are $K, V = [SG(H_t^l); H_t^l]$. Note the multi-head attention here is integrated with relative positional encoding (Dai et al., 2019).

Both designed to leverage the long-term dependency in context, the recurrence in Transformer-XL is between $H_t^l$ and $H_{l-1}^t$, which shifts one layer downwards per step. This mismatch in representation granularity may potentially be harmful to the learning process and affect the model performance. In contrast, the recurrence in MART is between $\bar{H}_t^l$ and $M_{l-1}^t$ (updated using $\bar{H}_{l-1}^l$) of the same layer. Besides, Transformer-XL directly uses all the hidden states from the last step to enable recurrence, which might be less effective as less relevant and repetitive information is also passed along. In comparison, MART achieves recurrence by using memory states that are highly summarized from previous steps, which may help the model to reduce redundancy and only keep important information from previous steps.
4 Experiments

We conducted experiments on two popular benchmark datasets, ActivityNet Captions (Krishna et al., 2017) and YouCookII (Zhou et al., 2017). We evaluate our proposed MART and compare it with various baseline approaches.

4.1 Data and Evaluation Metrics

Datasets ActivityNet Captions (Krishna et al., 2017) contains 10,009 videos in train set, 4,917 videos in val set. Each video in train has a single reference paragraph while each video in val has two reference paragraphs. Park et al. (2019) uses the same set of videos (though different segments) in val for both validation and test. To allow better evaluation of the models, we use splits provided by Zhou et al. (2019), where the original val set is split into two subsets: ae-val with 2,460 videos for validation and ae-test with 2,457 videos for test. This setup makes sure the videos used for test will not be seen in validation. YouCookII (Zhou et al., 2017) contains 1,333 training videos and 457 validation videos. Each video has a single reference paragraph. Both datasets come with temporal event segments annotated with human written natural language sentences. On average, there are 3.65 event segments for each video in ActivityNet Captions, 7.7 segments for each video in YouCookII.

Data Preprocessing We use aligned appearance and optical flow features extracted at 2FPS to represent videos, provided by Zhou et al. (2018). Specifically, for appearance, 2048D feature vectors from the ‘Flatten-673’ layer in ResNet-200 (He et al., 2016) are used; for optical flow, 1024D feature vectors from the ‘global pool’ layer of BN-Inception (Ioffe and Szegedy, 2015) are used. Both networks are pre-trained on ActivityNet (Caba Heilbron et al., 2015) for action recognition, provided by (Xiong et al., 2016b). We truncate sequences longer than 100 for video and 20 for text and set the maximum number of video segments to 6 for ActivityNet Captions and 12 for YouCookII. Finally, we build vocabularies based on words that occur at least 5 times for ActivityNet Captions and 3 times for YouCookII. The resulting vocabulary contains 3,544 words for ActivityNet Captions and 992 words for YouCookII.

Evaluation Metrics (Automatic and Human) We evaluate the captioning performance at paragraph-level, following (Park et al., 2019; Xiong et al., 2018), reporting numbers on standard metrics, including BLEU@4 (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014), CIDEr-D (Vedantam et al., 2015). Since these metrics mainly focus on whether the generated paragraph matches the ground-truth paragraph, they fail to evaluate the redundancy of these multi-sentence paragraphs. Thus, we follow previous works (Park et al., 2019; Xiong et al., 2018) to evaluate repetition using R@4. It measures the degree of N-gram (N=4) repetition in the descriptions. Besides the automated metrics, we also conduct human evaluations to provide additional comparisons between the methods. We consider two aspects in human evaluation, relevance (i.e., how related is a generated paragraph caption to the content of the given video) and coherence (i.e., whether a generated paragraph caption reads fluently and is linguistically coherent over its multiple sentences).

4.2 Implementation Details

MART is implemented in PyTorch (Paszke et al., 2017). We set the hidden size to 768, the number of transformer layers to 2, and the number of attention heads to 12. For positional encoding, we follow Vaswani et al. (2017) to use the fixed scheme. For memory module, we set the length of recurrent memory state to 1, i.e., $T_m=1$. We optimize the model following the strategy used by Devlin et al. (2019). Specifically, we use Adam (Kingma and Ba, 2014) with an initial learning rate of 1e-4, $\beta_1=0.9$, $\beta_2=0.999$, L2 weight decay of 0.01, and learning rate warmup over the first 5 epochs. We train the model for at most 50 epochs with early stopping using CIDEr-D and batch size 16. We use greedy decoding as we did not observe better performance using beam search.

4.3 Baselines

Vanilla Transformer This model originates from the transformer (Vaswani et al., 2017), proposed by Zhou et al. (2018) (more details in Section 3.1). It takes a single video segment as input and independently generates a single sentence describing the given segment. Note that Zhou et al. (2018) also have a separate proposal generation module, but here we only focus on its captioning module. To obtain paragraph-level generation, the independently generated single sentence captions are concatenated as the output paragraph.
Table 1: Comparison with transformer baselines on ActivityNet Captions \textit{ae-test} split and YouCookII \textit{val} split. Re. indicates whether sentence-level recurrence is used. We report BLEU@4 (B@4), METEOR (M), CIDEr-D (C) and Repetition (R@4). \textit{VTransformer} denotes vanilla transformer.

<table>
<thead>
<tr>
<th>Model</th>
<th>Det.</th>
<th>Re.</th>
<th>ActivityNet Captions (\textit{ae-test})</th>
<th>YouCookII (\textit{val})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>B@4</td>
<td>M</td>
</tr>
<tr>
<td>MART</td>
<td>✓</td>
<td>✓</td>
<td>9.78</td>
<td>15.57</td>
</tr>
<tr>
<td>Human</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2: Comparison with baselines on ActivityNet Captions \textit{ae-val} split. Det. indicates whether the model uses detection feature. Models that use detection features are shown in gray background to indicate they are not in fair comparison with the others. Re. indicates whether sentence-level recurrence is used. \textit{VTransformer} denotes vanilla transformer.

<table>
<thead>
<tr>
<th>Model</th>
<th>Det.</th>
<th>Re.</th>
<th>ActivityNet Captions (\textit{ae-val})</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM based methods</td>
<td></td>
<td></td>
<td>B@4</td>
</tr>
<tr>
<td>MFT (2018)</td>
<td>✓</td>
<td>✓</td>
<td>10.29</td>
</tr>
<tr>
<td>HSE (2018)</td>
<td>✓</td>
<td>✓</td>
<td>9.84</td>
</tr>
</tbody>
</table>

LSTM based methods with detection feature

<table>
<thead>
<tr>
<th>Model</th>
<th>Det.</th>
<th>Re.</th>
<th>ActivityNet Captions (\textit{ae-val})</th>
</tr>
</thead>
<tbody>
<tr>
<td>GVD (2019)</td>
<td>✓</td>
<td>✓</td>
<td>11.04</td>
</tr>
<tr>
<td>GVDsup (2019)</td>
<td>✓</td>
<td>✓</td>
<td>11.30</td>
</tr>
<tr>
<td>AdvInf (2019)</td>
<td>✓</td>
<td>✓</td>
<td>10.04</td>
</tr>
</tbody>
</table>

Transformer based methods

<table>
<thead>
<tr>
<th>Model</th>
<th>Det.</th>
<th>Re.</th>
<th>ActivityNet Captions (\textit{ae-val})</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTransformer (2018)</td>
<td>✓</td>
<td>✓</td>
<td>9.75</td>
</tr>
<tr>
<td>Transformer-XLRG (Ours)</td>
<td>✓</td>
<td>✓</td>
<td>10.17</td>
</tr>
<tr>
<td>MART</td>
<td>✓</td>
<td>✓</td>
<td>10.33</td>
</tr>
<tr>
<td>Human</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Transformer-XL is proposed by Dai et al. (2019) for modeling long-term dependency in natural language. Here we adapt it for video paragraph captioning (more details in Section 3.3). The original design of Transformer-XL stops gradients from passing between different recurrent steps to save GPU memory and computation. To enable a more fair comparison with our model, we implemented a version that allows gradient flow through different steps, calling this \textit{Transformer-XLRG} (Transformer-XL with Recurrent Gradient).

AdvInf AdvInf (Park et al., 2019) uses a set of three discriminators to do adversarial inference on a strong LSTM captioning model. The input features of the LSTM model are the concatenation of image recognition, action recognition, and object detection features. To encourage temporal coherence between consecutive sentences, the last hidden state from the previous sentence is used as input to the decoder (Xiong et al., 2018; Gella et al., 2018). The three discriminators are trained adversarially and are specifically designed to reduce repetition and encourage fluency and relevance in the generated paragraph.

GVD An LSTM based model for grounded video description (Zhou et al., 2019). It uses densely detected object regions as inputs, with a grounding module that grounds generated words to the regions. Additionally, we also consider a GVD variant (GVDsup) that uses grounding supervision from Zhou et al. (2019).

MFT MFT (Xiong et al., 2018) uses an LSTM model with a similar sentence-level recurrence as in AdvInf (Park et al., 2019).

HSE HSE (Zhang et al., 2018) is a hierarchical model designed to learn both clip-sentence and paragraph-video correspondences. Given the learned contextualized video embedding, HSE uses a 2-layer LSTM to generate captions.

For AdvInf, MFT, HSE, GVD, and GVDsup, we obtain generated sentences from the authors. We only report their performance on ActivityNet Captions \textit{ae-val} split to enable a fair comparison, as (i) AdvInf, MFT and HSE have different settings as ours, where \textit{ae-test} videos are included as part of their validation set; (ii) we do not have access to the \textit{ae-test} predictions of GVD and GVDsup. For vanilla transformer, Transformer-XL and Transformer-XLRG, we borrow/modify the model implementations from the original authors and train them under the same settings as MART.

4.4 Results

**Automatic Evaluation** Table 1 shows the results of MART and several transformer baseline methods. We observe stronger or comparable performance for the language metrics (B@4, M, C) for
Table 3: Human evaluation on ActivityNet Captions ae-test set w.r.t. relevance and coherence. Top: MART vs. vanilla transformer (VTransformer). Bottom: MART vs. Transformer-XL.

Table 4: Model ablation on ActivityNet Captions ae-val split. Re. indicates whether sentence-level recurrence is used. mem. len. indicates the length of the memory state. MART w/o re. denotes a MART variant without recurrence. Top two scores are highlighted.

Human Evaluation

In addition to the automatic metrics, we also run human evaluation on Amazon Mechanical Turk (AMT) with 200 randomly sampled videos from ActivityNet Captions ae-test split, where each video was judged by three different AMT workers. We design a set of pairwise experiments (Pasunuru and Bansal, 2017b; Park et al., 2019), where we compare two models at a time. AMT workers are instructed to choose which caption is better or decide that the two captions are not distinguishable based on relevance and coherence, respectively. The models are anonymized, and the predictions are shuffled. In total, we have 54 workers participated the MART vs. vanilla transformer experiments, 47 workers participated the MART vs. Transformer-XL experiments. In Table 3 we show human evaluation results, where the scores are calculated as the percentage of workers that have voted a certain option. With its sentence-level recurrence mechanism, MART is substantially better than the vanilla transformer model for both relevance and coherence. Compared to the strong baseline approach Transformer-XL, MART has similar performance in terms of relevance, but still reasonably better performance in terms of coherence.

Model Ablation

We show model ablation in Table 4. MART models with recurrence have better overall performance than the variant without, suggesting the effectiveness of our recurrent memory design. We choose to use the model with 2 hidden layers and memory state length 1 as it shows a good balance between performance and computation.

Qualitative Examples

In Figure 3, we show paragraph captions generated by vanilla transformer, Transformer-XL, and our method MART. Compared to the two baselines, MART produces more coherent and less redundant paragraphs. In particular, we noticed that vanilla transformer often uses incoherent pronouns/person mentions, while MART and Transformer-XL is able to use suitable pronouns/person mentions across the sentences and thus improve the coherence of the paragraph. Compared with Transformer-XL, we found that the paragraphs generated by MART have much less cross-sentence repetitions. We attribute MART’s success to its recurrence design - the previous memory states are highly summarized, in which redundant information is removed. While there is less redun-
He is sitting down in a chair. He continues playing the harmonica and looking off into the distance. He stops playing and looks at the camera.

Transformer-XL
A man is seen speaking to the camera while holding a harmonica. He continues playing the harmonica while looking at the camera. He continues playing the instrument and looking off into the distance. He stops playing and stops playing.

MART (ours)
A man is sitting down talking to the camera while holding a harmonica. He takes a harmonica and begins playing his harmonica. He continues playing the harmonica as he continues playing. He stops and looks at the camera.

Ground-Truth
A young man wearing a Cuervo black shirt stares and speaks to the camera as he sits on his chair. He puts a harmonica to his mouth and begins playing. He plays on for about a minute and is very into his song. He then puts the harmonica down and looks into the camera as the video comes to an end.

Figure 3: Qualitative examples. Red/bold indicates pronoun errors (inappropriate use of pronouns), blue/italic indicates repetitive patterns, underline indicates content errors. Compared to baselines, our model generates more coherent, less repeated paragraphs while maintaining relevance.

Figure 4: Nearest neighbors retrieved using memory states. Top row shows the query, the 3 rows below it are the top-3 nearest neighbors.

To explore whether the learned memory state could store useful information about the videos and captions, we conducted a video retrieval experiment on ActivityNet Captions train split with 10K videos, where we extract the last step memory state in the first layer of a trained MART model for each video as its representation to perform nearest neighbor search with cosine similarity. Though not explicitly trained for the retrieval task, we observe some positive examples in the experiments. We show an example in Figure 4, the neighbors mostly show related activities.

5 Conclusion
In this work, we present a new approach – Memory-Augmented Recurrent Transformer (MART) for video paragraph captioning, where we designed an auxiliary memory module to enable recurrence in transformers. Experimental results on two standard datasets show that MART has better overall performance than the baseline methods. In particular, MART can generate more coherent, less redundant paragraphs without any degradation in relevance.

Acknowledgments
We thank the anonymous reviewers for their helpful comments and discussions. This work was performed while Jie Lei was an intern at Tencent AI Lab, Seattle, USA. It was later partially supported by NSF Awards CAREER-1846185, 1562098, DARPA KAIROS Grant FA8750-19-2-1004, and ARO-YIP Award W911NF-18-1-0336. The views contained in this article are those of the authors and not of the funding agency.

References
Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. Advances in
Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In CVPR.


Spandana Gella, Mike Lewis, and Marcus Rohrbach. 2018. A dataset for telling the stories of social media videos. In EMNLP.


Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In ICCV.


Ramakanth Pasunuru and Mohit Bansal. 2017b. Reinforced video captioning with entailment rewards. In EMNLP.


Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. 2014. Coherent multi-sentence video description with variable level of detail. In GCPR.


Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. Videobert: A joint model for video and language representation learning. In ICCV.


Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In CVPR.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In ICCV.

A Appendices

A.1 Additional Qualitative Examples

We show more caption examples in Figure 5. Overall, we see captions generated by models with sentence-level recurrence, i.e., MART and Transformer-XL, tend to be more coherent. Comparing with Transformer-XL, captions generated by MART are usually less repetitive. However, as shown in the two examples at the last row of Figure 5, all three models suffer from the content error, where the models are not able to recognize and describe the fine-grained details in the videos, e.g., gender and fine-grained objects/actions.
Ground

MART (ours)

Vanilla Transformer

A man is seen speaking to the camera while holding a violin while moving his hands up and down. The man continues playing the instrument while looking down at the camera. He continues playing the violin and then stops to speak to the camera.

Transformer-XL

A man is seen speaking to the camera while holding a violin. The man continues playing the instrument while moving his hands up and down. He continues to play and stops by moving his hands up and down.

MART (ours)

A man is seen looking to the camera while holding a violin. The man then begins playing the instrument while the camera zooms in on his fingers. The man continues to play and stops to speak to the camera.

Ground-Truth

A girl's face is shown in front of the camera. Then she holds up a bottle of mouthwash and talks to the camera. The man then puts lotion on her face and begins rubbing it down. The man then begins to blow dry her face and shows off the camera.

Vanilla Transformer

She continues moving around the room and leads into her speaking to the camera. She continues moving around on the step and ends by speaking to the camera.

Transformer-XL

A woman is standing in a gym. She begins to do a step.

MART (ours)

A woman is standing in a room talking. She starts working out on the equipment.

Ground-Truth

A woman is seen speaking to the camera and leads into her walking up and down the board. She then stands on top of the beam while speaking to the camera continuously.

Vanilla Transformer

The man then holds up a bottle of mouthwash and talks to the camera. The man then puts lotion on her face and begins rubbing it down. The man then begins to blow dry her face and shows off the camera.

Transformer-XL

A man is seen speaking to the camera while holding up a brush. He then rubs lotion all over his face and begins brushing his face. He then puts lotion on the face and rubs it on the wall.

MART (ours)

A man is seen speaking to the camera and leads into him holding up a bottle of water. The man then holds up a cup and begins to shave his face. He finishes putting the cup into the mirror and smiles to the camera.

Ground-Truth

A girl's face is shown in front of the camera. She shows an orange bottle, reads the label and squirts the orange content on her palms, showed the cream on the camera, then rubs the cream all over her face. She bends down and raise her face, when her face is visible on the camera her face is clear.

Figure 5: Additional qualitative examples. Red/bold indicates pronoun errors (inappropriate use of pronouns or person mentions), blue/italic indicates repetitive patterns, underline indicates content errors. Compared to baselines, our model generates more coherent, less repeated paragraphs while maintaining relevance.