Multi-modal Dependency Tree for Video Captioning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Generating fluent and relevant language to describe visual content is critical for the video captioning task. Many existing methods generate captions using sequence models that predict words in a left-to-right order. In this paper, we investigate a graph structured model by explicitly modeling the hierarchical structure in the sentences to further improve the fluency and relevance of the generated captions. To this end, we propose a novel video captioning method that generates a sentence by first constructing a multi-modal dependency tree and then traversing the constructed tree, where the syntactic structure and semantic relationship in the sentence are represented by the tree topology. To take full advantage of the information from both vision and language, both the visual and textual representation features are encoded into each tree node. Different from existing dependency parsing methods that generate uni-modal dependency trees for language understanding, our method constructs multi-modal dependency trees for language generation of videos. We also propose a tree-structured reinforcement learning algorithm to effectively optimize the captioning model, where a novel reward is designed by evaluating the semantic consistency between the generated sub-trees and the ground-truth tree. Extensive experiments on several video captioning datasets demonstrate the effectiveness of the proposed method.

1 Introduction

Video captioning has received extensive attention from researchers in both computer vision and natural language processing. It is a challenging task since it requires not only understanding the visual content but also generating fluent and relevant sentences. With the success of deep neural networks in natural language processing, many existing video captioning methods [1, 2] use recurrent neural network (RNN) to generate sentences, which processes sequences by updating hidden states. Several recent studies [3–6] apply Transformer [7], which relates the words in the sequences using a self-attention mechanism, to video captioning. All these methods treat each sentence as a word sequence and generate words in a predefined left-to-right order by capturing the relatively close contextual relationship between words in the sentence.

In this paper, we investigate a *graph structured model* for video captioning to explicitly model the *hierarchical structure in the sentence* and capture the *long-range dependency between words*. With this in mind, we propose to generate a sentence by first constructing a multi-modal dependency tree in a top-down and depth-first order, and then traversing the tree in a recursive manner, as shown in Figure 1. Each node of the tree is represented by a multi-modal embedding representation that integrates the information from both visual and textual modalities. During the tree construction, each newly generated node receives the multi-modal embeddings from its parent and sibling nodes, and these multi-modal embeddings are used to predict the attention weights of the input features and the word. The attended visual feature and the predicted word are fused to generate the multi-modal embedding of the new node.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

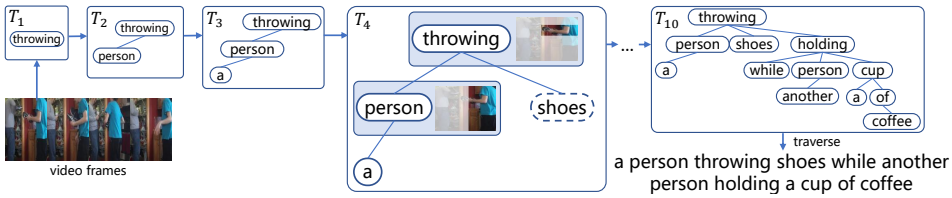


Figure 1: The sentence generation process of the proposed method. T_i denotes the dependency tree generated in the i-th step, and the colored box denotes the multi-modal embedding of each node.

Different from existing sequence models that tend to focus on the dependency between each word and its close preceding words, our tree model sufficiently captures the global dependency structure in the sentence to further improve the fluency of the generated sentences. In contrast to the uni-modal dependency tree widely used in dependency parsing for language analysis, our multi-modal dependency tree effectively integrates both the visual and linguistic information, improving the relevance of the output sentence to the visual input.

To effectively optimize the captioning model, we propose a tree-structured reinforcement learning algorithm and a novel node-level reward tailor-made for the tree construction process. By evaluating the consistency between the parent-child node pairs in the constructed tree and those in the ground truth trees, the tree node-level reward enables the model to capture the topological structure of the ground-truth dependency trees. Compared to the sequence-level rewards in existing methods, our node-level reward recognizes the contribution of each node to the multi-modal dependency tree and avoids the reward ambiguity problem [8].

To evaluate the effectiveness of the proposed method, we conduct experiments on two difficult video captioning datasets, the ActivityNet Captions dataset and Charades Captions dataset. We also perform experiments on two most widely-used datasets, namely the MSVD dataset and MSR-VTT dataset. Compared to MSVD and MSR-VTT, the sentences in the ActivityNet Captions dataset and Charades Captions dataset are typically longer and describe more complex activities in the videos. The experiments on these datasets demonstrate that our method not only generates long and complex sentences with high fluency and relevancy, but is also effective for relatively simple sentences.

The main contributions of this paper are as follows:

- We propose a multi-modal dependency tree construction method for video captioning. With the help of tree topology, our model generates more fluent and relevant sentences by effectively capturing the syntactic and semantic dependencies in natural language, especially when generating long and complex sentences.
- We develop a novel tree-structured reinforcement learning algorithm that optimizes the captioning model using a node-level reward, which facilitates learning the topology of the dependency trees and alleviates the reward ambiguity problem.

2 Related Work

2.1 Video Captioning

Thanks to the recent advances in computer vision and natural language processing, many video captioning methods based on the encoder-decoder framework have been proposed. The pioneers of video captioning methods [9] employ a convolutional neural network (CNN) to extract visual features and a recurrent neural network (RNN) to generate the words in the sentences in a sequential manner. To further improve the performance of video captioning, some methods [10] applys the attention mechanism in video captioning, which enables the model to focus on different temporal segments when generating the sentences. Inspired by the advances in natural language processing, some methods [11, 12] propose to learn better syntax representations with the help of part-of-speech tags. In order to provide richer semantic information to video captioning models, Rahman *et al.* [13] attempt to utilize the audio information.

Motivated by the advances in neural machine translation, several recent methods [3, 4, 6] investigate the application of Transformer [7] to video captioning, which uses multi-head self-attention

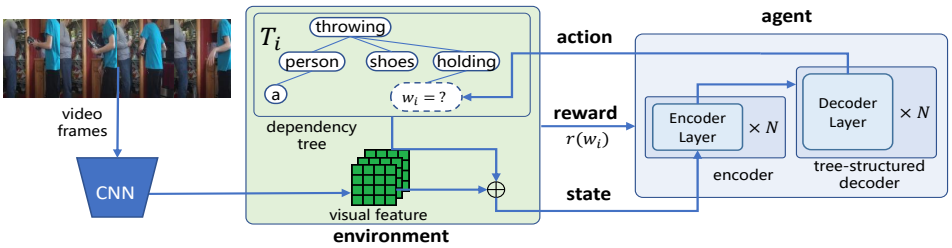


Figure 2: Overview of the proposed multi-modal dependency tree generation framework and the tree-structured reinforcement learning training algorithm. T_i and w_i denote the multi-modal dependency tree and the predicted word in the i-th step, respectively. The green-colored box in the middle stands for the *environment*, and the *state* of the environment is comprised of the perviously generated tree and the visual features. The blue-colored box on the right denotes the *agent*, namely the captioning model. During training, the agent executes an *action* by predicting a word w_i in the i-th step, and receives a reward $r(w_i)$ from the environment.

mechanism to yield more expressive representations of the input. Some methods [14, 15] focus on optimizing video captioning models using reinforcement learning, where the rewards are calculated based on the n-gram statistics.

Instead of directly generating a word sequence, our method generates a sentence by first constructing a dependency tree and then traversing the tree. Compared to the sequence generation models, our method explicitly models the grammatical structure with tree topology and better preserves the semantic relationship by incorporating both visual and linguistic information into the dependency tree.

2.2 Tree-structured Language Generation

In recent years, tree-structured language generation methods are proposed for various natural language processing tasks, including program generation [16], generating math equation [17] and machine translation [18]. Alvarez *et al.* [16] generate the tree nodes in depth-first search order using doubly-recurrent neural network, which combines the hidden states from the parent node and the last sibling node when predicting the label for a new node. A dependency tree generation method is proposed by [18] to generate target sentences in machine translation. Instead of directly constructing the tree structure, this method generates canonizalized ternary trees where each node has a fixed number of child nodes. Liu *et al.* [17] propose to generate abstract syntax trees of math expressions to solve math problems described in natural language.

The work closest to our method is [19] that generates image captions by constructing canonicalized dependency trees. Different from this method that generates trees with only linguistic information, our method constructs multi-modal dependency trees that contain both visual and linguistic information to capture richer contextual information.

3 Our Method

3.1 Overview

In this section, we introduce the proposed multi-modal dependency tree generation framework. The training data is formulated as $D = \{(x_i, T_i)|_i\}$, where x_i denotes the i-th or video, and T_i denotes the dependency tree of the corresponding sentence. Our model follows the encoder-decoder paradigm, where both the encoder and the decoder are stacks of N self-attention layers. Given an input video, the encoder of our model takes a set of d-dimensional visual features $V = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_M\}$ as input, where $\boldsymbol{v}_i \in \mathbb{R}^d$. The decoder firstly generates a multi-modal dependency tree $T = \{U, E\}$, where U and E denote the node set and edge set, respectively. Each node $u_i \in U$ corresponds to a word in the sentence, and a directed edge $\langle u_j, u_i \rangle \in E$ indicates that the node u_i is dependent on its parent node u_j. Then we use a recursive algorithm to traverse the tree T and recover the order of the words in the output sentence $y = \{w_1, w_2, ..., w_L\}$, where L denotes the length of the sentence and w_i denotes the i-th word. The overview of our framework is illustrated in Figure 2.

3.2 Tree Generation Process

Our tree generation model follows a step-by-step generation process. In the i-th step, a node u_i is added to the tree, and the corresponding word w_i is predicted. Formally, given an a video x, the probability of generating a tree T is given by

$$p(T|x) = \prod_{u_i \in T} p(u_i|x, T_{i-1}), \tag{1}$$

where T_{i-1} denotes the sub-tree that has been generated in the $(i-1)$-th step.

Since the structure of the dependency tree is unknown in the tree generation process, our model also predicts the topological information related to each node. For each node u_i, we define three topological labels: $t_i^s \in \{0, 1\}$, $t_i^c \in \{0, 1\}$ and $t_i^e \in \{0, 1\}$. t_i^s indicates whether node u_i has another sibling node. If $t_i^s = 1$, another sibling node of u_i will be added to the tree in the following steps. t_i^c decides whether u_i have child nodes, and least one child node of u_i will be added when $t_i^c = 1$. $t_i^e = 0$ indicates that the node u_i is the left child of its parent node u_j, namely the word w_i appears on the left of w_j in the output sentence, and $t_i^e = 1$ indicates that u_i is the right child of u_j.

3.3 Encoder

Given a set of visual features $V = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_N\}$, the encoder uses multiple self-attention operations to capture the relationships between video frames represented by the visual features. We first recall the scaled dot-product attention used by the self-attention layers, which operates on three sets of vectors $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d}})\boldsymbol{V}, \tag{2}$$

where $\boldsymbol{Q} \in \mathbb{R}^{n_q \times d}$ is a matrix consisting of n_q query vectors, $\boldsymbol{K} \in \mathbb{R}^{n_k \times d}$ and $\boldsymbol{V} \in \mathbb{R}^{n_k \times d}$ are matrices containing n_k key vectors and n_k value vectors, respectively. The vectors in Q, K and V are with the same dimensionality d. In addition to the self-attention layer, each encoder layer contains a feed-forward network (FFN) consisting of two fully connected layers, which can be formulated as

$$\text{FFN}(\boldsymbol{X}) = \text{ReLU}(\boldsymbol{X}\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2, \tag{3}$$

where $\boldsymbol{W_1}, \boldsymbol{W_2} \in \mathbb{R}^{d \times d}$ and $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathbb{R}^d$ are learnable parameters.

In each layer of the encoder, the queries, keys and values are obtained by linearly mapping the input features, and the output of the n-th encoding layer $\boldsymbol{X}_n$ is given by

$$\boldsymbol{X}'_n = \text{Attention}(\boldsymbol{W}_n^q \boldsymbol{X}_{n-1}, \boldsymbol{W}_n^k \boldsymbol{X}_{n-1}, \boldsymbol{W}_n^v \boldsymbol{X}_{n-1}), \boldsymbol{X}_n = \text{FFN}(\boldsymbol{X}_n), \tag{4}$$

where $\boldsymbol{W}_n^q \in \mathbb{R}^{d \times d}$, $\boldsymbol{W}_n^k \in \mathbb{R}^{d \times d}$ and $\boldsymbol{W}_n^v \in \mathbb{R}^{d \times d}$ are learnable parameters, and $\boldsymbol{X}_{n-1}$ is the output of the previous encoding layer. FFN represents the feed-forward network defined in Eq. 3. The visual features V are used as the input of the first encoding layer.

3.4 Tree Structured Decoder

The decoder is conditioned on both the output of the encoder and the previously generated sub-tree. In the i-th step, the decoder takes the encoded visual feature $\boldsymbol{X}_N$, the words in the sub-tree T_{i-1} and the multi-modal embeddings of the nodes in T_{i-1} as input, and predicts the word w_i together with the topological labels t_i^s, t_i^c and t_i^e. The tree generation process terminates if the predicted word w_i is a special token $\langle eos \rangle$ indicating the end of the sentence. Since the order of words may affect the semantics of the sentence, in each step, we traverse the tree T_{i-1} using the algorithm that will be described in Section 3.5 and obtain a node sequence $u_{k_1}, u_{k_2}, ..., u_{k_{i-1}}$ before feeding the words that have been generated so far to the decoder, where k_i indicates the index of the nodes.

Let $\boldsymbol{E}_{i-1} \in \mathbb{R}^{(i-1) \times d}$ denote the embeddings of the previously generated $(i-1)$ words. The decoder acquires the multi-modal embedding of the current node $\boldsymbol{a}_i$ by firstly attending to the visual features and the embeddings of the previously generated words using self-attention layers, and then fusing the output of the self-attention. The n-th decoding layer is formulated as:

$$\begin{aligned} \boldsymbol{Y}_n &= \text{FFN}(\boldsymbol{Y}_n^{vis} + \boldsymbol{Y}_n^{word}) \\ &= \text{FFN}(\text{Attention}(\boldsymbol{Y}_{n-1}, \boldsymbol{X}_N, \boldsymbol{X}_N) + \text{Attention}(\boldsymbol{Y}_{n-1}, \boldsymbol{E}_{i-1}, \boldsymbol{E}_{i-1})). \end{aligned} \tag{5}$$

$\boldsymbol{Y}_N \in \mathbb{R}^d$ represents the output of the last decoding layer and is used as the multi-modal embedding of the current node u_i, i.e. $\boldsymbol{a}_i = \boldsymbol{Y}_N$ in the i-th step. The probability distribution of the word corresponding to u_i is calculated using the multi-modal embedding:

$$\boldsymbol{p}_{w_i} = \boldsymbol{W}_{word}\boldsymbol{a}_i, \tag{6}$$

where $\boldsymbol{W}_{word} \in \mathbb{R}^{|D|\times d}$ is a learnable parameter, and $|D|$ denotes the vocabulary size. After the word w_i is determined by a sampling strategy (e.g. greedy sampling or beam search), the topological labels of the current node are predicted by

$$\begin{aligned} p(t_i^s = 1) &= \text{sigmoid}(\boldsymbol{w}_s^\top[\boldsymbol{a}_i; \boldsymbol{E}_{w_i}]), \\ p(t_i^c = 1) &= \text{sigmoid}(\boldsymbol{w}_c^\top[\boldsymbol{a}_i; \boldsymbol{E}_{w_i}]), \\ p(t_i^e = 1) &= \text{sigmoid}(\boldsymbol{w}_e^\top[\boldsymbol{a}_i; \boldsymbol{E}_{w_i}]), \end{aligned} \tag{7}$$

where $\boldsymbol{w}_s, \boldsymbol{w}_c, \boldsymbol{w}_e \in \mathbb{R}^{2d}$ are learnable parameters, $\boldsymbol{E}_{w_i} \in \mathbb{R}^d$ denotes the embedding of word w_i, respectively. The operator $[;]$ denotes the vector concatenation.

3.5 Tree Traversal Algorithm

We restore the word order and convert the tree to human-readable word sequence by traversing the tree. Given a generated dependency tree T, our tree traversal algorithm recursively processes each node in the tree, and restores the word order of the phrase represented by the node and its sub-trees. Since the tree is generated following the depth-first search, the generation order of the child nodes of each node is the same as the order of the corresponding words in the sentence. Thus, for an arbitary node u_i in the dependency tree T, we sequentially traverse its left child nodes, the node u_i and its right child nodes.

3.6 Model Training

3.6.1 Pre-Training Stage

The training process of the multi-modal dependency generation model involves a pre-training stage and a fine-tuning stage. Since the model predicts both the word and the topology labels, we use a word loss $\mathcal{L}_w$ and a topological loss $\mathcal{L}_t$ in the pre-training stage. Formally, the word loss is defined as

$$\mathcal{L}_w = -\sum_{i=1}^{|V|} \log p(w_i|T_{i-1}, x), \tag{8}$$

where $p(w_i|T_{i-1}, x)$ denotes the probability of predicting word w_i given the previously generated tree T_{i-1} and the visual input x. The topological loss is defined as

$$\begin{aligned} \mathcal{L}_t &= \mathcal{L}_{bce}(t_i^s, \hat{t}_i^s), + \mathcal{L}_{bce}(t_i^c, \hat{t}_i^c) + \mathcal{L}_{bce}(t_i^e, \hat{t}_i^e), \\ \mathcal{L}_{bce}(t, \hat{t}) &= -\sum_{i=1}^{|V|} \hat{t} \cdot \log(p(t)) + (1 - \hat{t}) \cdot \log(1 - p(t)), \end{aligned} \tag{9}$$

where $\hat{t}_i^s$, $\hat{t}_i^c$ and $\hat{t}_i^e$ are ground-truth topological labels and $\mathcal{L}_{bce}$ denotes the binary cross-entropy loss. The overall loss function in the pre-training stage is formulated by

$$\mathcal{L}_{pretrain} = \mathcal{L}_w + \mathcal{L}_t \tag{10}$$

3.6.2 Fine-tuning Stage

To effectively optimize the proposed model, we formulate the multi-modal dependency tree generation as a decision-making process and fine-tune the model with reinforcement learning. Specifically, the captioning model is regarded as the *agent*, the input video x and the generated dependency tree T_{i-1} are regarded as the *state* of the environment, and the prediction of the word w_i is regarded as the *action*. Instead of using the same sequence-level reward for all the actions, we design a novel node-level reward that estimates the contribution of each individual action. The reward $r(w_i)$ for the word w_i is given by

$$r(w_i) = \lambda_1(\text{CIDEr}(T_i) - \text{CIDEr}(T_{i-1})) + \lambda_2 \mathbb{1}(\langle u_i, u_p \rangle \in E), \tag{11}$$

where $\mathrm{CIDEr}(T)$ denotes the CIDEr score of the word sequence represented by the nodes in T, u_p denotes the parent node of u_i and $\mathbb{1}(\langle u_i, u_p \rangle \in E)$ indicates whether the edge $\langle u_i, u_p \rangle$ is present in the ground-truth dependency tree. λ_1 and λ_2 are tunable hyper parameters. The discounted future reward for the agent is calculated by

$$R(w_i) = r(w_i) + \sum_{k=1}^{\infty} \gamma^k r(w_{i+k}), \tag{12}$$

where γ denotes the discount factor. Let θ denote the parameters of the captioning model, and then the loss function in the fine-tuning stage is given by

$$\mathcal{L}_{rl} = -\mathbb{E}_{w_i \sim \pi(\theta)}(R(w_i)), \tag{13}$$

where $\pi(\theta)$ denotes the policy defined by θ.

4 Experiments

4.1 Datasets

ActivityNet Captions is a dense video captioning dataset that contains 10,030 training videos, 4,926 validation videos and 5,044 test videos. Each video is annotated with an average of 3.65 sentences together with temporal annotation and the average length of the sentences is 13.7 words. We conduct the experiments on ActivityNet Captions with ground-truth proposals and the results are reported on the validation set.

Charades Captions is composed of 9,223 videos of indoor activities. Each video is annotated with 2-5 sentences, and the average length of the sentences is 24.13 words. Following [14], we split this dataset into 6,963 training videos, 500 validation videos and 1,760 test videos.

MSVD consists of 1,970 video clips collected from YouTube, each of which annotated with about 41 sentences and the average length of the sentences is 7.10 words. We follow [9] to split the dataset into 1,200 training videos, 100 validation videos and 670 testing videos.

MSR-VTT is a dataset collected for open-domain video captioning, which contains 10,000 video clips in total. Each video in the MSR-VTT dataset contains 20 human annotated captions. We use the splits provided by [14], where the training split, validatation split and test split contains 6,513 videos, 497 videos and 2,990 videos, respectively. The average length of the sentences in MSR-VTT is 9.28 words.

4.2 Evaluation Metrics

We report several widely-used automatic evaluation metrics for video captioning, including Bleu-n [20], METEOR [21], ROUGE-L [22] and CIDEr [23]. Nevertheless, these n-gram based evaluation metrics have some limitations, e.g., they penalize the phrases that are semantically correct but differ from ground-truth in the specific word choices [24]. To evaluate the quality of the sentences more effectively, we also report Improved BERTScore metric proposed by [25]. Compared with the n-gram based metrics, Improved BERTScore is computed using pre-trained BERT embeddings and has better correlation with human experts.

4.3 Implementation Details

In the training process, in order to convert the sentences to dependency trees, we use the dependency parser in the SpaCy toolkit [26]. For the ActivityNet Captions dataset, we extract the visual features of the 16-frame video segments using the pre-trained C3D network. To compare our method with [3], we also extract the frame features using ResNet-200 [27], and extract the optical flow features using BN-Inception [28]. For the Charades Captions dataset, we sample the video frames at 3 fps and extract the feature of each frame using the pre-trained ResNet-152 network [27] following [14].

The hyper parameters λ_1 and λ_2 defined in Eq. 11 are set to 0.7 and 0.3, respectively. We use the Adam optimizer [29] in both pre-training and fine-tuning stages. The learning rate is set to 0.0002 in pre-training stage. During fine-tuning, the initial learning rate is 0.0002 and decays 0.8 times for every 10 epochs. The experiments are conducted using one NVIDIA RTX 2080Ti GPU.

Table 1: Comparison with state-of-the-art methods and the results of ablation studies on the ActivityNet Captions dataset using ground-truth proposals. B@n, M, C and BS are abbreviations for Bleu-n, METEOR, CIDEr and Improved BERTScore, respectively. *Note that the last two rows are compared for different video features.*

Feature	Model	B@1	B@2	B@3	B@4	M	C	BS
C3D	DCE [30]	18.13	8.43	4.09	1.60	8.88	25.12	-
	DVC [31]	19.57	9.90	4.55	1.62	10.33	25.24	-
	SDVC [32]	28.02	12.05	4.41	1.28	13.07	43.48	-
	w/o tree	27.01	11.12	4.10	1.47	12.40	42.30	35.57
	w/o visual embedding	25.30	10.90	3.74	1.36	11.09	35.10	32.69
	w/o RL	20.31	8.54	3.90	1.40	11.23	38.90	34.30
	sequence reward	27.35	11.18	4.15	1.62	11.70	41.39	35.23
	Ours	**28.53**	**12.12**	**4.49**	**1.75**	**13.24**	**44.13**	**36.35**
ResNet-200+ BN-Inception	Masked [3]	23.93	12.16	**5.76**	2.71	11.16	47.71	-
	Ours	**24.25**	**12.35**	5.54	**2.74**	**11.20**	**47.87**	-

Table 2: Experiment results and the results of ablation studies on the Charades Captions dataset. R is the abbreviation for ROUGE-L.

Model	B@1	B@2	B@3	B@4	M	R	C	BS
HRL [14]	64.40	44.30	29.40	18.80	19.50	41.40	23.20	-
w/o tree	62.10	43.50	28.50	17.70	18.70	40.20	23.50	34.13
w/o visual embedding	60.40	40.30	26.70	16.90	18.20	39.50	21.90	34.51
w/o RL	58.40	39.30	25.30	16.20	18.30	39.10	21.60	33.71
sequence reward	64.00	44.90	29.70	18.50	19.20	41.60	23.50	34.25
Ours	**65.30**	**45.60**	**29.80**	**18.90**	**19.80**	**41.70**	**24.20**	**34.97**

4.4 Comparison with State-of-the-Art Methods

The results on ActivityNet Captions are shown in Table 1. We compare our method with DCE [30], DVC [31], SDVC [32], and Masked (which uses different video features) [3]. All these methods generate sentences using sequence models. The captioning models in [31, 32, 13] are implemented by RNN and the model proposed by [3] is based on Transformer.

All the results of these compared methods are directly quoted from their original papers. Our method achieves the best performance on the ActivityNet Captions dataset, demonstrating that by explicitly modeling the hierarchical structure of the sentences, our proposed method can generate long and complex sentences with higher fluency and relevance.

We compare our method with HRL [14] on the Charades Captions dataset. This method involves a manager network that designs sub-goals and a worker network that fulfills each sub-goal by predicting words, both of which are trained end-to-end using hierarchical reinforcement learning. The results are shown in Table 2. We observe that our method outperforms HRL, which evaluates the effectiveness of our method in generating complex video descriptions.

The results on the MSVD dataset and the MSR-VTT dataset are shown in Table 3 and Table 3, respectively. We compare our method with HRL [14], MARN [33], POS-CG [12], Joint [11] and RMN [34].All these methods generate sentences using sequence models implemented by RNN. All the results of these compared methods are directly quoted from their original papers. For fair comparison, we use the same visual features as these methods. From the results on MSVD and MSR-VTT, we observe that our method achieves comparable performance with the state-of-the-art methods when using different visual features, which demonstrates that our method is also effective for generating relatively simple sentences. Note that in such simple cases our proposed method is not expected to make much difference or perform better.

4.5 Ablation Studies

To analyze the effect of different components, we conduct ablation studies on the ActivityNet Captions dataset and the Charades Captions dataset. The following variants of our method are evaluated:

Table 3: Comparison with state-of-the-art methods on MSVD and MSR-VTT using different visual features. I3D(RGB) and I3D(OF) indicate using I3D network [35] to extract the features of raw video frames and optical flow, respectively.

Feature	Model	MSVD				MSR-VTT			
		B@4	R	M	C	B@4	R	M	C
ResNet152	HRL [14]	-	-	-	-	41.3	61.7	28.7	48.0
	Joint [11]	52.1	69.8	33.7	80.6	41.4	**62.0**	28.9	48.1
	Ours	**52.2**	**70.0**	**34.0**	**81.2**	**41.6**	**62.0**	**29.1**	**48.4**
ResNet101+ ResNext101	MARN [33]	48.4	71.9	35.1	92.2	**40.4**	60.7	28.1	47.1
	Ours	**49.0**	**72.2**	**35.3**	**92.5**	40.2	**61.1**	**28.2**	**47.3**
InceptionResnetV2+ I3D(OF)	POS-CG[12]	**52.5**	71.3	34.1	92.0	**42.0**	**61.6**	28.2	48.7
	Ours	51.7	**71.6**	**34.9**	**92.4**	41.8	61.4	**28.3**	**49.0**
InceptionResNetV2+ I3D(RGB)	RMN [34]	**52.5**	72.7	36.1	**92.8**	42.5	61.6	**28.4**	49.6
	Ours	51.8	**72.8**	**36.7**	92.5	**42.6**	**61.8**	29.3	**49.8**

- **w/o tree**: To evaluate the advantage of the tree structure over the sequence structure, we replace the dependency trees with chain-structured trees, where the root node corresponds to the leftmost word in the sentence. For each non-leaf node, its only child node corresponds to the word on its right. This variant of our model is actually a sequence model that generates the words in the sentence in left-to-right order.
- **w/o visual embedding**: To evaluate the effectiveness of the visual embeddings of the tree nodes, we replace the attended visual feature $\boldsymbol{Y}_n^{vis}$ defined in Eq. 5 with an all-zero matrix, and the node representations contains only linguistic information.
- **w/o RL**: To validate the effectiveness of reinforcement learning, we only optimize the model with the cross-entropy loss in pre-training stage.
- **sequence reward**: To verify the contribution of the tree node-level reward, we replace it with the sequence-level reward, i.e. the reward of each node equals to the CIDEr score of the entire sentence.

The results of ablation studies on the ActivityNet Captions and Charades Captions datasets are shown in the middle part of Table 1 and Table 2, respectively. From these results, we make the following observations. First, by replacing the dependency tree with the chain-structured tree, the performance of our model degrades in terms of all the metrics, indicating that the hierarchical structure of the sentences is beneficial for generating sentences with high quality. Second, our full model outperforms "w/o visual embedding", validating the superiority of the multi-modal representation of the tree node by taking full advantage of both visual and linguistic information. Third, when the fine-tuning stage with reinforcement learning is removed, our model performs worse on all the metrics, which validates the contribution of our reinforcement learning algorithm. Our full model also outperforms "sequence reward", which verifies that the node-level reward effectively guides the training process.

4.6 Evaluation on Simple and Complex Subsets

To further evaluate the effect of the tree structure on generating simple sentences and complex sentences, we split the videos in the test sets of MSR-VTT and Charades Captions into a simple subset and a complex subset according to the average length of the ground-truth sentences. The data distributions of the two subsets and the evaluation results are shown in Table 4. From these results, we observe that by utilizing the tree structure, the performance of our model significantly increases on the complex subset, which demonstrates the proposed model's capability in terms of generating complex sentences.

4.7 Qualitative Results

We show some examples of the generated sentences on Charades Captions and MSR-VTT in Figure 3. As shown in the figure, by utilizing the tree structure, our method generates sentences that not only describes the objects and the human actions more accurately, but also possess correct grammatical structures.

Table 4: Data distribution and evaluation results on the simple subset and complex subset of Charades Captions and MSR-VTT.

Subset	Sentence length	Video number	w/o tree			Ours		
			B@4	R	C	B@4	R	C
Charades (simple)	≤ 20	550	18.8	36.7	28.4	17.9	40.9	25.8
Charades (complex)	> 20	1210	17.1	33.3	20.8	18.8	39.8	24.4
MSR-VTT (simple)	≤ 10	2005	38.3	62.3	54.8	45.8	66.2	59.8
MSR-VTT (complex)	> 10	985	31.0	48.6	25.3	35.1	50.8	27.7

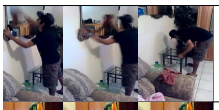
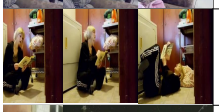
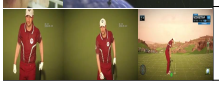
	**ours:** A person is standing in front of a mirror holding a towel. The person puts the towel on a shelf and leaves.	**w/o tree:** A person is holding a book and a phone. The person puts the book on the floor and begins tidying up the room.	**gt:** A person walks to a mirror and begins wiping it with a towel. The person then tosses the towel on a couch and adjusts a table.
	**ours:** A person is sitting on the floor reading a book. The person puts the book on a shelf picks up a book and leaves.	**w/o tree:** A person is sitting on the floor reading a book while reading a book.	**gt:** She is sitting laundry room and reading book, and now she is lying on floor while reading, and put the sheet under her head.
	ours: A man is talking about space.	**w/o tree:** There is a man talking about something.	**gt:** A man talks about the benefits of defensive satellites.
	**ours:** A person is playing a golf game.	**w/o tree:** A man is in a green shirt playing a baseball game.	**gt:** A golf player is trying to hit the ball into the pit.

Figure 3: Qualitative results on Charades Captions (the first two rows) and MSR-VTT (the last two rows). "gt", "Ours" and "w/o tree" denote the ground-truth, the sentence generated by our method and the sentence generated by the variant "w/o tree" of our model, respectively.

4.8 Human Evaluation

To intuitively evaluate the effect of the tree structure on the quality of the generated sentences, we conduct human evaluation on the test splits of ActivityNet Captions and MSR-VTT. We randomly choose 200 videos from the test splits of ActivityNet Captions and MSR-VTT, respectively. The workers are given the original video as well as two sentences generated by "Ours" and "w/o tree", and are asked to select the sentence that has better relevance to the video and the sentence with better fluency. For the two methods "Ours" and "w/o tree", we report the percentage of sentences that have better relevance and the percentage of sentences that have better fluency. From the results in Table 5, we observe that the tree structure remarkably improves both the relevance and the fluency of the generated sentences.

Table 5: The results of human evaluation.

Dataset	Relevance		Fluency	
	w/o tree	Ours	w/o tree	Ours
ActivityNet	44.63%	55.36%	48.02%	51.97%
MSR-VTT	45.65%	54.34%	46.20%	53.80%

5 Conclusions

We have presented a multi-modal dependency tree generation method for video captioning. Our model can explicitly model the hierarchical structure of natural language using tree topology to better capture both the syntactic structure and semantic relationship in sentences. Both visual and linguistic information are incorporated into the node embeddings to further improve the relevance of the generated sentence to the video. Moreover, we have developed a tree-structured reinforcement learning algorithm and designed a novel tree node-level reward to effectively optimize the captioning model. Our multi-modal dependency tree is capable of generating complex sentences with high fluency and relevance for videos. Extensive experiment on multiple challenging video captioning datasets have demonstrated the effectiveness of our method.

References

[1] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015.

[2] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018.

[3] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-end dense video captioning with masked transformer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018.

[4] L. Huang, W. Wang, J. Chen, and X. Wei, "Attention on attention for image captioning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019.

[5] G. Li, L. Zhu, P. Liu, and Y. Yang, "Entangled transformer for image captioning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019.

[6] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara, "Meshed-memory transformer for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017.

[8] A. Y. Ng, D. Harada, and S. J. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proc. 6th Int. Conf. Mach. Learn.* (I. Bratko and S. Dzeroski, eds.), 1999.

[9] S. Venugopalan, M. Rohrbach, J. Donahue, R. J. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence - video to text," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015.

[10] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, "Video captioning with attention-based LSTM and semantic consistency," *IEEE Trans. Multimedia*, vol. 19, no. 9, 2017.

[11] J. Hou, X. Wu, W. Zhao, J. Luo, and Y. Jia, "Joint syntax representation learning and visual cue translation for video captioning," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 8918–8927, 2019.

[12] B. Wang, L. Ma, W. Zhang, W. Jiang, J. Wang, and W. Liu, "Controllable video captioning with pos sequence guidance based on gated fusion network," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2641–2650, 2019.

[13] T. Rahman, B. Xu, and L. Sigal, "Watch, listen and tell: Multi-modal weakly supervised dense event captioning," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 8908–8917, 2019.

[14] X. Wang, W. Chen, J. Wu, Y. Wang, and W. Y. Wang, "Video captioning via hierarchical reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018.

[15] L. Li and B. Gong, "End-to-end video captioning with multitask reinforcement learning," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 339–348, 2019.

[16] D. Alvarez-Melis and T. S. Jaakkola, "Tree-structured decoding with doubly-recurrent neural networks," in *5th Int. Conf. Learn. Represent.*, 2017.

[17] Q. Liu, W. Guan, S. Li, and D. Kawahara, "Tree-structured decoding for solving math word problems," in *Proc. Conf. Empirical Methods in Natural Language Processing*, 2019.

[18] G. Zhou, P. Luo, R. Cao, Y. Xiao, F. Lin, B. Chen, and Q. He, "Tree-structured neural machine for linguistics-aware sentence generation," in *Proc. AAAI Conf. Artif. Intell.*, 2018.

[19] Z. Ma, C. Yuan, Y. Cheng, and X. Zhu, "Image-to-tree: A tree-structured decoder for image captioning," in *IEEE International Conference on Multimedia and Expo*, 2019.

[20] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

[21] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72, 2005.

[22] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, pp. 74–81, 2004.

[23] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575, 2015.

[24] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[25] Y. Yi, H. Deng, and J. Hu, "Improving image captioning evaluation by considering inter references variance," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 985–994, 2020.

[26] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing." To appear, 2017.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, PMLR, 2015.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd Int. Conf. Learn. Represent.*, 2015.

[30] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles, "Dense-captioning events in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017.

[31] Y. Li, T. Yao, Y. Pan, H. Chao, and T. Mei, "Jointly localizing and describing events for dense video captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018.

[32] J. Mun, L. Yang, Z. Ren, N. Xu, and B. Han, "Streamlined dense video captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019.

[33] W. Pei, J. Zhang, X. Wang, L. Ke, X. Shen, and Y.-W. Tai, "Memory-attended recurrent network for video captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 8347–8356, 2019.

[34] G. Tan, D. Liu, M. Wang, and Z.-J. Zha, "Learning to discretely compose reasoning module networks for video captioning," *arXiv preprint arXiv:2007.09049*, 2020.

[35] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 6299–6308, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Please refer to the supplementary material for potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes]

   (b) Did you mention the license of the assets? [No]

   (c) Did you include any new assets either in the supplemental material or as a URL? [No]

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes] Please refer to the supplementary materials for details of human evaluation.

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [Yes]