This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Hierarchy Parsing for Image Captioning

Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei
JD AI Research, Beijing, China
{tingyao.ustc, panyw.ustc, yehaoli.sysu}@gmail.com, tmei@jd.com

Abstract

It is always well believed that parsing an image into constituent visual patterns would be helpful for understanding and representing an image. Nevertheless, there has not been evidence in support of the idea on describing an image with a natural-language utterance. In this paper, we introduce a new design to model a hierarchy from instance level (segmentation), region level (detection) to the whole image to delve into a thorough image understanding for captioning. Specifically, we present a HIerarchy Parsing (HIP) architecture that novelly integrates hierarchical structure into image encoder. Technically, an image decomposes into a set of regions and some of the regions are resolved into finer ones. Each region then regresses to an instance, i.e., foreground of the region. Such process naturally builds a hierarchal tree. A tree-structured Long Short-Term Memory (Tree-LSTM) network is then employed to interpret the hierarchal structure and enhance all the instance-level, region-level and image-level features. Our HIP is appealing in view that it is pluggable to any neural captioning models. Extensive experiments on COCO image captioning dataset demonstrate the superiority of HIP. More remarkably, HIP plus a top-down attention-based LSTM decoder increases CIDEr-D performance from 120.1% to 127.2% on COCO Karpathy test split. When further endowing instance-level and region-level features from HIP with semantic relation learnt through Graph Convolutional Networks (GCN), CIDEr-D is boosted up to 130.6%.

1. Introduction

Automatic image captioning is the task of generating a natural sentence that correctly reflects the visual content of an image. Practical automatic image description systems have a great potential impact for instance on robotic vision, or helping visually impaired people by transforming visual signals into information that can be communicated via text-to-speech technology. The recent state-of-the-art image captioning methods use to perform "encoder-decoder" translation [6, 14, 29]. A Convolutional Neural Network (CNN) first encodes an image into a feature vector, and a caption is then decoded from this vector, one word at each time step using a Long Short-Term Memory (LSTM) Network. There are variants of approaches arisen from this methodology, for instance, conducting attention on the feature map [32] or leveraging attributes to augment image features [35]. Regardless of these different versions, today's neural captioning models tend to leverage correlations from training data and produce generic plausible captions, but lack visual understanding on the compositional patterns in images.

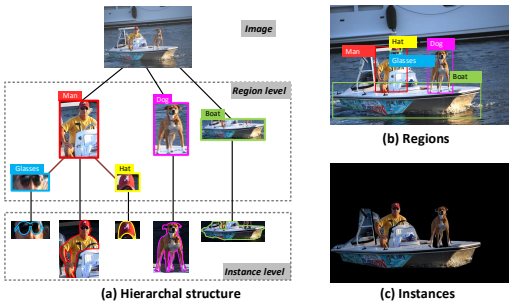


Figure 1. Examples of (a) the hierarchal tree structure in an image, (b) regions and (c) instances in the image.

We propose to mitigate the problem from the viewpoint of parsing an image into a hierarchical structure of constituent visual patterns to better represent the image. The key idea is to build a top-down hierarchical tree from the root of the whole image to the middle layers of regions and the leaf layer of instances. Each instance in the leaf layer emphasizes the discriminative foreground of a region. Figure 1 (a) conceptualizes the typical development of a tree structure on an image. Figure 1 (b) and (c) illustrates the regions and foreground/instances in the image, respectively. In this case, we could strengthen visual interpretations of image structure in a bottom-up manner and the learning of all the image-level, region-level and instance-level features does benefit from tree-structured topologies. We expect our design to be a feature refiner or bank, that outputs rich and multi-level representations of the image. Such multi-level representations could be utilized separately or jointly, depending on the particular task. It is also flexible to further integrate the reasoning of semantic relation between regions or instances to further endow region-level or instance-level features with more power.

By consolidating the exploitation of structure hierarchy in an image, we present a new HIerarchy Parsing (HIP) architecture for boosting image encoder in captioning. Specifically, we devise a three-level hierarchy, in which an image is first decomposed into a set of regions and one region is represented either at that level or by further split into finer ones. Each region then corresponds to an instance, that models foreground of the region. A tree-structured LSTM is then executed on the hierarchy from the bottom up to upgrade the features throughout all the levels. After that, a hybrid of features on three levels output by HIP could be easily fed into a general attention-based LSTM decoder to produce the sentence, one word at each time step. Moreover, our HIP, as a feature optimizer, would further augment the features by propagating and aggregating semantic relation. On the hierarchy, we build semantic graph with directed edges on region or instance level, where the vertex represents each region or instance and the edge denotes the relation between each pair of regions or instances. Graph Convolutional Networks (GCN) are exploited to enrich region/instance features with visual relation in the semantic graph. The enhanced features eventually improve image captioning. Please also note that HIP is flexible to generalize to other vision tasks, e.g., recognition.

The main contribution of this work is the parse of hierarchical structure in an image for captioning. The solution also leads to the elegant view of how to build and interpret the hierarchy of an image, and how to nicely integrate such hierarchy into typical neural captioning frameworks, which are problems not yet fully understood in the literature. Our design is viewed as a feature refiner in general and is readily pluggable to any neural captioning models.

2. Related Work

Image Captioning. The recent works for image captioning [3, 6, 29, 32, 35, 36] are mainly sequence learning based methods which utilize CNN plus RNN to generate sentences word-by-word, enabling the flexible modeling of syntactical structure within sentence. Specifically, [29] is one of the early attempts to cast the sentence generation task as a sequence learning problem and leverages LSTM to model the dependency among words for sentence generation conditioning on the input image. [32] further extends [29] by integrating soft and hard attention mechanism into LSTM-based decoder, which learns to focus on image regions to facilitate the generation of the corresponding word at each decoding stage. [31, 35, 36] demonstrate the effectiveness of semantic attributes in image captioning, where the attributes are taken as additional inputs of CNN plus RNN to emphasize them in output sentence. Later on, [24] develops a self-critical sequence training strategy to amend the discrepancy between training and inference for sequence modeling and thus boost image captioning. Furthermore, instead of measuring attention over a pre-defined uniform grid of image regions as in [32], [3] especially devises the bottom-up mechanism to enable the measurement of attention at object level, and the top-down mechanism to associate the salient image regions and the output words for sentence generation. Most recently, [34] models the relations between objects in the context of image captioning, which will be further incorporated into the top-down attention model [3] to enhance captions. In addition, image captioning could be extended to novel object captioning [17, 33] which leverages unpaired image/text data to describe novel objects or image paragraph generation [30] which produces a coherent paragraph to depict an image.

In our work, we exploit the hierarchal structure in images from instance level, region level, to the whole image, to facilitate a thorough image understanding for captioning. To do this, we design a novel hierarchy parsing architecture to integrate hierarchical structure into image encoder, which is pluggable to any neural captioning models.

Structured Scene Parsing. The task of structured scene parsing goes beyond the general recognition of scene type (classification) or localization of objects in a scene (semantic labeling or segmentation) and considers a deeper and structured understanding on scene. An early pioneering work [27] devises a Bayesian framework for parsing images into the constituent visual patterns over a hierarchical parsing graph. Later on, Han *et al.* leverage an attributed grammar model to hierarchically parse the man-made indoor scene [8]. A connected segmentation tree is proposed in [1] to capture canonical characteristics of the object in terms of photometric and geometric properties, and containment and neighbor relationships between its constituent image regions. [37] designs a hierarchical image model for image parsing, which represents image structures with different levels of contextual information. In [20], a hierarchical shape parsing strategy is proposed to partition and organize image components into a hierarchical structure in the scale space. Sharma *et al.* devise the recursive context propagation network [25] for semantic scene labeling by recursively aggregating contextual information from local neighborhoods up to the entire image and then disseminating the aggregated information back to individual local features over a binary parse tree.

The hierarchy parsing architecture in our method is also a type of structured scene parsing for images. Unlike the aforementioned methods that are developed for image parsing or semantic scene labeling, our hierarchy parsing architecture acts as an image encoder to interpret the hierarchal structure in images and is applicable to image captioning task. As such, all the instance-level, region-level and image-level features are enhanced with tree-structured topologies, which will be injected into captioning model to further boost sentence generation.

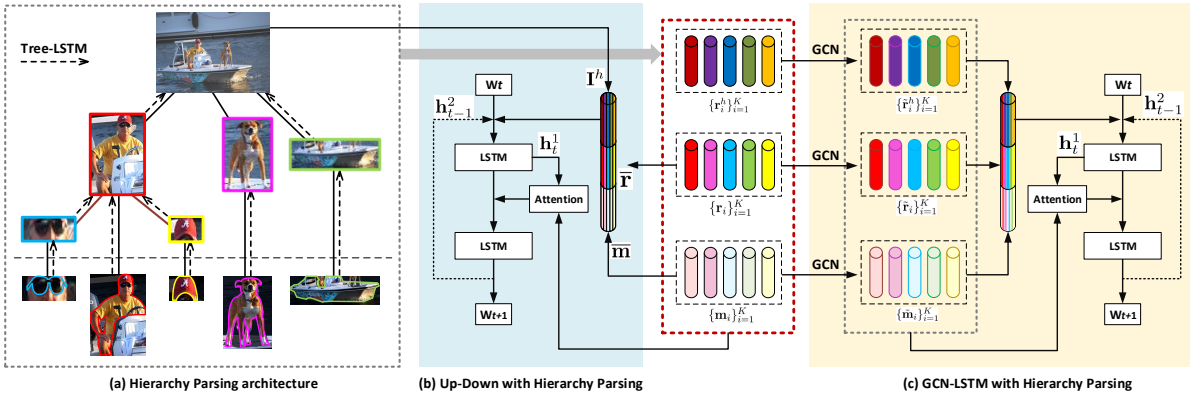


Figure 2. An overview of (a) our HIerarchy Parsing (HIP) architecture for integrating hierarchical structure into image encoder, and its applications for image captioning task by plugging HIP in (b) Up-Down [3] and (c) GCN-LSTM [34]. For HIP, Faster R-CNN and Mask R-CNN are first leveraged to detect and segment the set of object regions and instances, respectively. Next, we construct a three-level hierarchy, where the whole image is first decomposed into a set of regions and one region is represented either at that level or by further split into finer ones. Each region in the middle layers is naturally associated with the corresponding instance in the leaf layer. After that, a Tree-LSTM is executed on the hierarchy from the bottom up with enhanced region/instance features and the outputs are image-level features. A hybrid of features on three levels output by HIP could be easily fed into a general attention-based LSTM decoder in Up-Down for sentence generation. Moreover, it is also flexible to plug our HIP into GCN-LSTM by further enriching the multi-level features with semantic relations in semantic graph via Graph Convolutional Networks (GCN).

3. Our Approach

In this paper, we devise a HIerarchy Parsing (HIP) architecture to integrate hierarchical structure into image encoder, pursuing a thorough image understanding to facilitate image captioning. HIP firstly constructs a three-level hierarchy from the root of the whole image to the middle layers of regions and the leaf layer of instances, leading to a deep and structured modeling of image. Tree-LSTM is then leveraged to contextually enhance features at instance level, region level, and image level. In this sense, HIP acts as a feature refiner, that outputs rich and multi-level representations of an image. Hence HIP is pluggable to any neural captioning models, including the general attention-based LSTM decoder or a specific relation-augmented decoder. An overview of our HIP architecture and its applications in two different captioning models is illustrated in Figure 2.

3.1. Overview

Notation. The target of image captioning task is to describe the given image I with a textual sentence $\mathcal{S}$. Note that the textual sentence $\mathcal{S} = \{w_1, w_2, ..., w_{N_s}\}$ is a word sequence containing N_s words. Each word in sentence is represented as a D_s-dimensional textual feature, e.g., $\mathbf{w}_t \in \mathbb{R}^{D_s}$, which denotes the feature of t-th word in sentence $\mathcal{S}$. Since our ultimate hierarchy consists of compositional patterns at instance level, region level and image level, we firstly leverage object detection method (Faster R-CNN [23]) and instance segmentation approach (Mask R-CNN [9]) to produce the basic elements (i.e., regions and instances) of the hierarchy. The set of regions and instances in image I is denoted as $\mathcal{R} = \{r_i\}_{i=1}^K$ and $\mathcal{M} = \{m_i\}_{i=1}^K$ respectively, which corresponds to the regions and foregrounds/instances of detected K objects. Each image region and its instance are denoted as the D_r-dimensional features $\mathbf{r}_i \in \mathbb{R}^{D_r}$ and $\mathbf{m}_i \in \mathbb{R}^{D_r}$. Accordingly, the image I is parsed into a hierarchal tree $\mathcal{T} = (I, \mathcal{R}, \mathcal{M}, \mathcal{E}_{tree})$ consisting of layers at three levels: the root layer corresponds to entire image I, the middle layers of regions $\mathcal{R}$, and the leaf layer of instances $\mathcal{M}$. $\mathcal{E}_{tree}$ represents the connections. More details about how we represent regions & instances and construct the hierarchal tree will be elaborated in Section 3.2.

Problem Formulation. The key idea behind our formulation is to frame the hierarchical structure modeling of image in the context of image captioning task. To start, given the sets of regions and instances decomposed from input image, we holistically characterize an image with a three-level hierarchy. Derived from the idea of tree-structured LSTM [26], we further leverage a Tree-LSTM module to contextually refine the representation of each instance/region in a bottom-up manner along the hierarchical tree and finally acquire the image-level feature. As such, by taking the hierarchy parsing architecture (i.e., the construction of hierarchical tree and the feature enhancement via a Tree-LSTM) as a process of image encoding, the output multi-level representations are endowed with more power. After that, the contextually refined features on three levels from the hierarchy parsing architecture are fed into a general attention-based LSTM decoder [3] to facilitate sentence generation. Therefore, the image captioning problem here is generally formulated as the minimization of energy loss function:

$$E(\mathcal{T}, \mathcal{S}) = -\log \Pr(\mathcal{S}|\mathcal{T}), \tag{1}$$

which is the negative log probability of the correct sentence $\mathcal{S}$ given the hierarchal tree $\mathcal{T}$. Moreover, since the hierarchy parsing architecture is designed to be a feature refin-

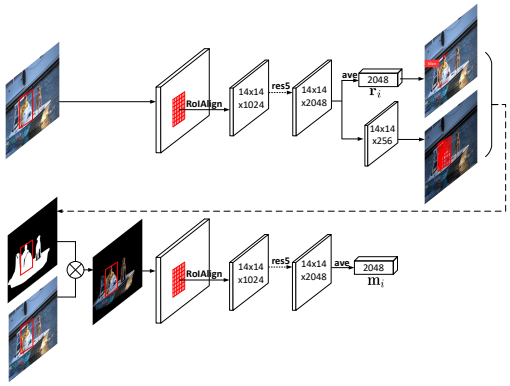


Figure 3. Feature extraction of regions and instances. Mask R-CNN augments the pre-trained Faster R-CNN with an additional mask branch and is adopted to extract region feature $\mathbf{r}_i$ and predict the instance-level foreground mask of each region. Next, a blend of each region and its binary via element-wise multiplication is fed into another Faster R-CNN to produce instance feature $\mathbf{m}_i$.

er or bank, we could further augment the output enhanced features by propagating and aggregating semantic relations. That is, our hierarchy parsing architecture can be applied in the relation-based captioning model [34] which endows the region-level or instance-level features with relations.

3.2. Hierarchy Parsing in Images

Next we describe the details of our HIerarchy Parsing (HIP) architecture, which strengthens all the instance-level, region-level and image-level features with tree-structured topologies as a feature optimizer. In particular, we begin this section by presenting the extraction of regions and instances within images. Then, we provide how to construct the three-level hierarchy by associating all regions and instances in an image. Finally, an image encoder equipped with Tree-LSTM for interpreting the hierarchal structure and enhancing multi-level features is presented.

Regions and Instances of Image. Given an input image, we firstly apply Faster R-CNN trained on Visual Genome [16] to detect the image regions of objects. Note that only the top $K = 36$ regions with highest confidences $\mathcal{R} = \{r_i\}_{i=1}^{K}$ are selected to represent the image. We represent each region as the 2,048-dimensional output ($\mathbf{r}_i$) of pool5 layer after RoI pooling from the feature map of Faster R-CNN (backbone: ResNet-101 [10]). In addition, to emphasize the discriminative knowledge of objects implicit in each image region, we separate the foreground & background of each region and take the foreground of region as the associated instance. Specifically, Mask R-CNN augments the pre-trained Faster R-CNN with an additional mask branch and predicts the instance-level foreground mask of each region. As such, the foreground/instance is obtained by blending each region and its binary mask via element-wise multiplication, leading to the set of instances $\mathcal{M} = \{m_i\}_{i=1}^{K}$. Next, we train another Faster R-CNN over the foreground images and the 2,048-dimensional output

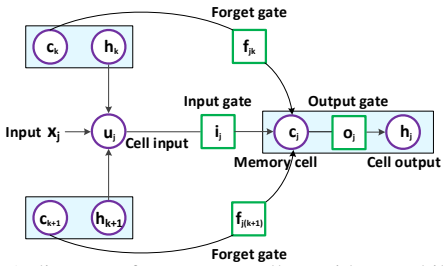


Figure 4. A diagram of a memory cell c_j with two children (subscripts k and $k+1$) in Tree-LSTM. We omit the dependencies of four gates for compactness.

($\mathbf{m}_i$) of this Faster R-CNN is taken as the representation of each instance m_i. Note that the inputs to the two Faster R-CNN models are different (one is original image and the other is foreground image) and the two models don't share any parameters. Figure 3 details the pipeline for the feature extraction of regions and instances.

Hierarchy Structure of an Image. Recent advances on visual relationship [34] have demonstrated that modeling the structure in an image (e.g., a semantic graph built based on relations between regions) does enhance image captioning. Our work takes a step forward and constructs a hierarchical structure of constituent visual patterns, i.e., hierarchal tree, to fully exploit a hierarchy from instance level, region level to the whole image and learn the connections across each level for image captioning. Specifically, the hierarchal tree $\mathcal{T} = (I, \mathcal{R}, \mathcal{M}, \mathcal{E}_{tree})$ organizes all the regions and instances of input image I in a top-down three-level hierarchy, including the root layer, the middle layers of regions, and the leaf layer of instances.

Concretely, only one root node is initially established in the upper root layer, which denotes the entire image I. Such image-level root node is decomposed into a set of regions, which in turn resolve into a number of finer regions, resulting in multiple middle layers of regions. Here the existence of dependency between pairs of root node and regions is assigned depending on their Intersection over Union (IoU). More precisely, given the K image regions $\mathcal{R}$, we firstly rank all the regions in order of descending area of region and then integrate each region into the hierarchal tree in turn. For each region r_i, we measure IoU between r_i and each object node in current hierarchal tree. If the maximum IoU is larger than a threshold ϵ, r_i is incorporated into the hierarchal tree as a child of the existing region node with maximum IoU, which indicates that r_i can be treated as the finer region within the region of its parent. Otherwise, r_i is directly taken as the child node of the image-level root node. Once the construction of middle layers of regions completes, we attach each instance to the corresponding region node as a child in the bottom leaf layer.

Image Encoder with Tree-LSTM. One natural way to model the contextual relationship across samples in a set/sequence is to adopt LSTM based models as in [4]. However, such kind of chain-structured LSTM is typical-

ly order-insensitive and thus insufficient to fully capture the differences in order or dependency structure. Taking the inspiration from the success of Tree-LSTM [26] for modeling tree-structured topologies in several NLP tasks, we leverage Tree-LSTM in image encoder to facilitate contextual information mining within hierarchy and thus enrich image-level features with holistic hierarchical structure.

A diagram of the Tree-LSTM unit is illustrated in Figure 4. Similar to the standard LSTM, Tree-LSTM unit consists of a memory cell $\mathbf{c}_j$ indexed by j, hidden state $\mathbf{h}_j$, input gate $\mathbf{i}_j$, and output gate $\mathbf{o}_j$. Unlike LSTM updates memory cell depending on only previous hidden state, the updating of Tree-LSTM unit relies on the multiple hidden states of its children. Moreover, Tree-LSTM unit includes forget gate $\mathbf{f}_{jk}$ for each child (indexed by k). In particular, the vector formulas for a Tree-LSTM unit forward pass are given below. For the node indexed by j in a tree, $\mathbf{x}_j$ and $\mathbf{h}_j$ denote the input and output vector, respectively. The set of children of this node is denoted as $C(j)$. $\mathbf{W}$ are input weights matrices, $\mathbf{U}$ are recurrent weight matrices and $\mathbf{b}$ are biases. Sigmoid σ and hyperbolic tangent ϕ are element-wise nonlinear activation functions. $\odot$ represents the dot product of two vectors. Hence the Tree-LSTM unit updates are:

$$\begin{aligned}
\widetilde{\mathbf{h}}_j &= \sum_{k \in C(j)} \mathbf{h}_k \\
\mathbf{u}_j &= \phi(\mathbf{W}_u \mathbf{x}_j + \mathbf{U}_u \widetilde{\mathbf{h}}_j + \mathbf{b}_u) && \textit{cell input} \\
\mathbf{i}_j &= \sigma(\mathbf{W}_i \mathbf{x}_j + \mathbf{U}_i \widetilde{\mathbf{h}}_j + \mathbf{b}_i) && \textit{input gate} \\
\mathbf{f}_{jk} &= \sigma(\mathbf{W}_f \mathbf{x}_j + \mathbf{U}_f \mathbf{h}_k + \mathbf{b}_f) && \textit{forget gate} \\
\mathbf{o}_j &= \sigma(\mathbf{W}_o \mathbf{x}_j + \mathbf{U}_o \widetilde{\mathbf{h}}_j + \mathbf{b}_o) && \textit{output gate} \\
\mathbf{c}_j &= \mathbf{u}_j \odot \mathbf{i}_j + \sum_{k \in C(j)} \mathbf{c}_k \odot \mathbf{f}_{jk} && \textit{cell state} \\
\mathbf{h}_j &= \phi(\mathbf{c}_j) \odot \mathbf{o}_j && \textit{cell output}
\end{aligned} . \quad (2)$$

Specifically, for our hierarchal tree $\mathcal{T}$, we take the original extracted region/instance features ($\{\mathbf{r}_i\}_{i=1}^K$ and $\{\mathbf{m}_i\}_{i=1}^K$) as the input vectors of region nodes in the middle layers and instance nodes in the leaf layer. The input vector of image-level root node is set as the linear fusion of image-level mean-pooled features of regions ($\overline{\mathbf{r}} = \frac{1}{K}\sum_{i=1}^K \mathbf{r}_i$) and instances ($\overline{\mathbf{m}} = \frac{1}{K}\sum_{i=1}^K \mathbf{m}_i$): $\mathbf{I} = \mathbf{W}_r \overline{\mathbf{r}} + \mathbf{W}_m \overline{\mathbf{m}}$. Accordingly, by operating Tree-LSTM over our hierarchal tree in a bottom-up manner, the region-level features of each region node are further strengthened with the contextual information mined from its instance and even finer regions, which are denoted as $\{\mathbf{r}_i^h\}_{i=1}^K$. In addition, the outputs of root node in hierarchal tree are treated as the image-level features $\mathbf{I}^h$, which are endowed with the inherent hierarchal structure from instance level, region level to the whole image.

3.3. Image Captioning with Hierarchy Parsing

Since we design our HIP architecture to be a feature refiner or bank that outputs rich and multi-level representations of the image, it is feasible to plug HIP into any neural captioning models. We next discuss how to integrate hierarchy parsing into a general attention-based LSTM decoder in Up-Down [3] or a specific relation-augmented decoder in GCN-LSTM [34]. Please also note that our HIP is flexible to generalize to other vision tasks, e.g., recognition.

Up-Down with Hierarchy Parsing. Given a hybrid of features on three levels output by HIP (i.e., the image-level features ($\mathbf{I}^h$, $\overline{\mathbf{r}}$, $\overline{\mathbf{m}}$) and region-level/instance-level features ($\{\mathbf{r}_i^h\}_{i=1}^K$, $\{\mathbf{r}_i\}_{i=1}^K$, $\{\mathbf{m}_i\}_{i=1}^K$), we directly feed them into a general attention-based decoder with two-layer LSTM in Up-Down, as depicted in Figure 2 (b). Specifically, at each time step t, the input of the first-layer LSTM unit is set as the concatenation of the input word w_t, the previous output of the second-layer LSTM unit $\mathbf{h}_{t-1}^2$ and all image-level features ($\mathbf{I}^h$, $\overline{\mathbf{r}}$, $\overline{\mathbf{m}}$). Such design can collect the maximum contextual information for the first-layer LSTM to model dependency among words. After that, we represent each image region by concatenating all region-level and instance-level features belonging to it, denoted as $\mathbf{v}_i = \left[\mathbf{r}_i^h, \mathbf{r}_i, \mathbf{m}_i\right]$. Next, a normalized attention distribution $\lambda_t \in \mathbb{R}^K$ over all regions $\{\mathbf{v}_i\}_{i=1}^K$ is calculated conditioning on the output $\mathbf{h}_t^1$ of the first-layer LSTM unit, resulting in the attended image feature $\hat{\mathbf{v}}_t = \sum_{i=1}^K \lambda_{t,i}\mathbf{v}_i$. Note that $\lambda_{t,i}$ is the i-th element in λ_t which represents the attention probability of i-th region. Therefore, we feed the concatenation of the attended image feature $\hat{\mathbf{v}}_t$ and $\mathbf{h}_t^1$ into the second-layer LSTM unit, aiming to trigger the generation of next word w_{t+1}.

GCN-LSTM with Hierarchy Parsing. When applying the hierarchy parsing into GCN-LSTM [34], the instance-level and region-level features from HIP are further enhanced with visual relation learnt through GCN and thus improve the captions, as shown in Figure 2 (c). In particular, we firstly build semantic graph with directed edges on region or instance level of the hierarchy. GCN is then leveraged to enrich the region-level/instance-level features ($\{\mathbf{r}_i^h\}_{i=1}^K$, $\{\mathbf{r}_i\}_{i=1}^K$, $\{\mathbf{m}_i\}_{i=1}^K$) with visual relations in the semantic graph. All of the enhanced region-level/instance-level features ($\{\tilde{\mathbf{r}}_i^h\}_{i=1}^K$, $\{\tilde{\mathbf{r}}_i\}_{i=1}^K$, $\{\tilde{\mathbf{m}}_i\}_{i=1}^K$) from GCN are further fed into a two-layer LSTM for sentence generation.

Extension to Recognition Task. The image-level features from our HIP can be further utilized to other vision tasks, e.g., recognition. The spirit behind follows the philosophy that the hierarchy parsing integrates hierarchical structure of image into encoder, making the learnt image-level features more representative and discriminative.

4. Experiments

We empirically verify the merit of our HIP by conducting experiments on COCO [19] for image captioning task.

4.1. Datasets and Settings

COCO is a standard benchmark in the field of image captioning. The dataset contains 123,287 images (82,783 for training and 40,504 for validation) and each image is annotated with 5 descriptions. Given the fact that the human-

Table 1. Performance (%) of our HIP and other methods on COCO Karpathy test split.

	Cross-Entropy Loss					CIDEr-D Score Optimization				
	BLEU@4	METEOR	ROUGE-L	CIDEr-D	SPICE	BLEU@4	METEOR	ROUGE-L	CIDEr-D	SPICE
LSTM [29]	29.6	25.2	52.6	94.0	-	31.9	25.5	54.3	106.3	-
SCST [24]	30.0	25.9	53.4	99.4	-	34.2	26.7	55.7	114.0	-
ADP-ATT [21]	33.2	26.6	-	108.5	-	-	-	-	-	-
LSTM-A [35]	35.2	26.9	55.8	108.8	20.0	35.5	27.3	56.8	118.3	20.8
RFNet [13]	37.0	27.9	57.3	116.3	20.8	37.9	28.3	58.3	125.7	21.7
Up-Down [3]	36.2	27.0	56.4	113.5	20.3	36.3	27.7	56.9	120.1	21.4
Up-Down+HIP	37.0	28.1	57.1	116.6	21.2	38.2	28.4	58.3	127.2	21.9
GCN-LSTM [34]	37.1	28.1	57.2	117.1	21.1	38.3	28.6	58.5	128.7	22.1
GCN-LSTM+HIP	**38.0**	**28.6**	**57.8**	**120.3**	**21.4**	**39.1**	**28.9**	**59.2**	**130.6**	**22.3**

annotated descriptions of the official testing set are not provided, we utilize Karpathy split (113,287 for training, 5,000 for validation and 5,000 for testing) as in [3]. Following [14], all the training sentences are converted to lower case and we omit rare words which occur less than 5 times. As such, the final vocabulary includes 10,201 unique words.

Visual Genome is adopted to train Faster R-CNN for object detection. Here we follow the setting in [3, 34] and take 98,077 images for training, 5,000 for validation, and 5,000 for testing. As in [3], 1,600 objects and 400 attributes are selected from Visual Genome for training Faster R-CNN with two branches for predicting object and attribute classes.

COCO-detect is a popular benchmark for instance segmentation, containing the same images with COCO from 80 object categories. All object instances are annotated with a detailed segmentation mask. Here we utilize the partially supervised training paradigm [12] to train Mask R-CNN, enabling instance segmentation over the entire 1,600 objects. In particular, the detection branch in Mask R-CNN is initialized with the weights of learnt Faster R-CNN from Visual Genome. Next, the mask branch and weight transfer function in Mask R-CNN are further trained on COCO-detect. Note that we adopt the same split of COCO for training Mask R-CNN on COCO-detect.

Implementation Details. We represent each word as "one-hot" vector. The threshold ϵ for constructing hierarchy is set as 0.1. The hidden layer size in Tree-LSTM and LSTM-based decoder is set as 500 and 1,000, respectively. The captioning models with our HIP are mainly implemented with PyTorch, optimized with Adam [15]. For the training with cross-entropy loss, we set the learning rate as 5×10^{-4} and the mini-batch size as 50. The maximum iteration is set as 30 epoches. For the training with self-critical training strategy, we follow [24] and select the model which is trained with cross-entropy loss and achieves best CIDEr-D score on validation set, as initialization. Next the captioning model is further optimized with CIDEr-D reward. Here the learning rate is set as 5×10^{-5} and the maximum iteration is 30 epoches. At inference, beam search strategy is adopted and we set the beam size as 3. Five popular metrics, i.e., BLEU@N [22], METEOR [5], ROUGE-L [18], CIDEr-D [28] and SPICE [2], are leveraged for evaluation.

Compared Methods. (1) **LSTM** [29] only feeds image into LSTM-based decoder at the initial time step for triggering sentence generation. The reported results are directly drawn from [24]. (2) **SCST** [24] devises a self-critical sequence training strategy to train a modified attention-based captioning model in [32]. (3) **ADP-ATT** [21] designs an adaptive attention mechanism to decide whether to attend to the image and which image regions to focus, for image captioning. (4) **LSTM-A** [35] extends the common encoder-decoder captioning model by additionally injecting semantic attributes into LSTM-based decoder. (5) **RFNet** [13] devises a recurrent fusion network to fuse multiple encoders and generate new informative features for decoder with attention. (6) **Up-Down** [3] devises a bottom-up attention mechanism to calculate attention at object level to boost image captioning. (7) **GCN-LSTM** [34] extends [3] by exploiting visual relationships between objects. (8) **Up-Down+HIP** and **GCN-LSTM+HIP** are our proposals by plugging the devised HIP into Up-Down and GCN-LSTM, respectively. Please note that all the state-of-the-art methods and our models utilize ResNet-101 as the backbone of image encoder, for fair comparison. Besides, we report the results of each model trained with both cross-entropy loss or CIDEr-D reward in self-critical strategy.

4.2. Performance Comparison and Analysis

Performance on COCO. The performances of different models on COCO for image captioning task are summarized in Table 1. Overall, the results across all metrics and two optimization methods (Cross-Entropy Loss and CIDEr-D Score Optimization) consistently indicate that our GCN-LSTM+HIP exhibits better performances than other approaches, including non-attention models (LSTM, LSTM-A) and attention-based approaches (SCST, ADP-ATT, RFNet, Up-Down, and GCN-LSTM). Up-Down+HIP and GCN-LSTM+HIP by integrating hierarchy parsing architecture makes the absolute improvement over Up-Down and GCN-LSTM by 3.1% and 3.2% in terms of CIDEr-D respectively, optimized with cross-entropy loss. The results generally highlight the key advantage of exploiting the hierarchal structure in an image from instance level, region level, to the whole image, pursuing a thorough image un-

Table 2. Performance (%) of the top ranking published state-of-the-art image captioning models on the online COCO test server.

Model	BLEU@1		BLEU@2		BLEU@3		BLEU@4		METEOR		ROUGE-L		CIDEr-D	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
GCN-LSTM+HIP	**81.6**	**95.9**	**66.2**	**90.4**	**51.5**	**81.6**	**39.3**	**71.0**	**28.8**	**38.1**	**59.0**	**74.1**	**127.9**	**130.2**
GCN-LSTM [34]	80.8	95.2	65.5	89.3	50.8	80.3	38.7	69.7	28.5	37.6	58.5	73.4	125.3	126.5
RFNet [13]	80.4	95.0	64.9	89.3	50.1	80.1	38.0	69.2	28.2	37.2	58.2	73.1	122.9	125.1
Up-Down [3]	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
LSTM-A [35]	78.7	93.7	62.7	86.7	47.6	76.5	35.6	65.2	27.0	35.4	56.4	70.5	116.0	118.0
SCST [24]	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.7

derstanding for captioning. Specifically, by injecting the high-level semantic attributes into LSTM-based decoder, LSTM-A outperforms LSTM that trains decoder only depending on the input image. Nevertheless, the attention-based methods (SCST, ADP-ATT, Up-Down, and RFNet) exhibit better performance than LSTM-A, which verifies the merit of attention mechanism that dynamically focuses on image regions for sentence generation. Furthermore, GCN-LSTM by exploring the relations between objects to enrich region-level features, improves SCST, ADP-ATT, Up-Down, and RFNet. However, the performances of GCN-LSTM are lower than GCN-LSTM+HIP that additionally exploits hierarchical structure in an image for enhancing all the instance-level, region-level and image-level features and eventually boosting image captioning. In addition, by optimizing the captioning models with CIDEr-D score instead of cross-entropy loss, the CIDEr-D score of GCN-LSTM+HIP is further boosted up to 130.6%. This confirms that self-critical training strategy is an effective way to amend the discrepancy between training and inference, and improve sentence generation regardless of image captioning approaches. Similar to the observations on the optimization with cross-entropy loss, Up-Down+HIP and GCN-LSTM+HIP lead to better performances than Up-Down and GCN-LSTM when optimized with CIDEr-D score.

Ablation Study. Next, we examine how captioning performance is affected when capitalizing on different features. Table 3 details the performances by exploiting different features in Up-Down sentence decoder. The use of original region-level features in general achieves a good performance. As expected, only utilizing original instance-level features is inferior to region-level features. The result indicates that the context in the background of a region is still a complement to the foreground. On a throughout hierarchy parsing of an image, the finer features produced by Tree-LSTM lead to better performance. The concatenation of every two features constantly outperforms the individual one. The integration of all three features, i.e., our HIP, reaches the highest performance for captioning. The results basically demonstrate the complementarity in between.

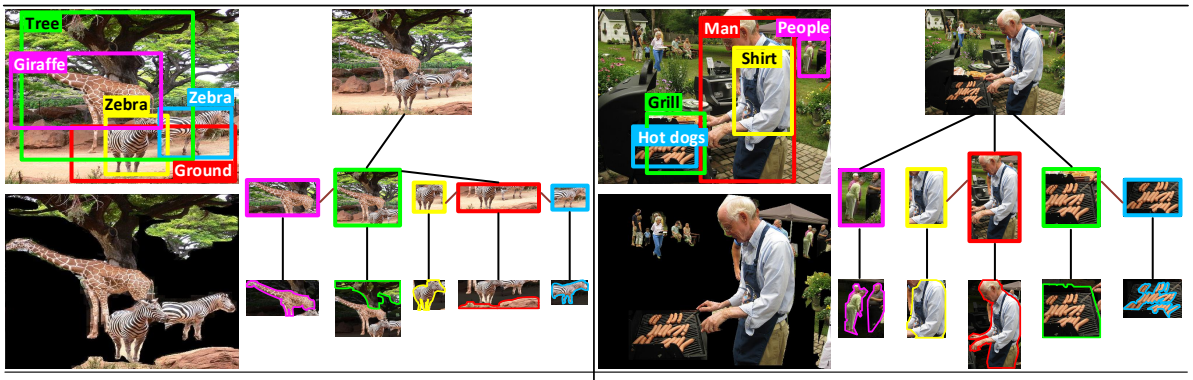
Qualitative Analysis. Figure 5 showcases a few image examples with instances, regions, the hierarchy structure, ground truth sentences and captions produced by LSTM, GCN-LSTM and GCN-LSTM+HIP, respectively. As illustrated in the exemplar results, the sentences output by GCN-LSTM+HIP are more descriptive. For example, compared to the phrase "a group of zebras" in the captions produced by LSTM and GCN-LSTM for the first image, the words of "two zebras" depict the image content more accurately in GCN-LSTM+HIP. We speculate that the results are benefited from the segmentation of two "zebra" instances and the integration of such information into hierarchy structure. The results again indicate the advantage of guiding sentence generation through holistically interpreting and parsing the structure of an image in our HIP.

Table 3. An ablation study on the use of different features.

Regions	Instances	Tree-LSTM	BLEU@4	METEOR	CIDEr-D
✓			36.2	27.0	113.5
	✓		36.1	27.0	113.3
		✓	36.3	27.4	113.7
✓	✓		36.6	27.5	114.9
✓		✓	36.8	27.9	115.5
	✓	✓	36.7	27.9	115.2
✓	✓	✓	**37.0**	**28.1**	**116.6**

Performance on COCO Online Testing Server. We also submitted the run of GCN-LSTM+HIP optimized with CIDEr-D score to online COCO testing server. Table 2 shows the performance Leaderboard on official testing image set with 5 reference captions (c5) and 40 reference captions (c40). Note that here we utilize SENet-154 [11] as the backbone of Faster R-CNN and Mask R-CNN in our final submission. The latest top-5 performing systems which have been officially published are included in the table. Our GCN-LSTM+HIP leads to performance boost against all the other top-performing systems on the Leaderboard.

Human Evaluation. As the automatic sentence evaluation metrics do not necessarily correlate with human judgement, we additionally conducted a human study to evaluate GCN-LSTM+HIP against two baselines, i.e., LSTM and GCN-LSTM. We invite 12 labelers and randomly select 1K images from testing set for human evaluation. All the labelers are grouped into two teams. We show the first team each image with three auto-generated sentences plus three human-annotated captions and ask the labelers: Do the systems produce human-like sentences? Instead, we show the second team only one sentence at a time, which could be generated by captioning methods or human annotation (Human). The labelers are asked: Can you distinguish human



GT: two zebras and a giraffe standing by a tree
LSTM: a group of zebras are standing next to a giraffe
GCN-LSTM: a group of zebras and a giraffe standing next to a tree
GCN-LSTM+HIP: two zebras and a giraffe standing next to a tree

GT: a man cooking hot dogs on a grill
LSTM: a man and a woman sitting at a table with a grill
GCN-LSTM: a man is cooking food on a grill
GCN-LSTM+HIP: a man is cooking hot dogs on a grill

Figure 5. Two image examples from COCO dataset with instances, regions, hierarchy structure, and sentence generation results. The output sentences are generated by 1) Ground Truth (GT): One ground truth sentence, 2) LSTM, 3) GCN-LSTM and 4) GCN-LSTM+HIP.

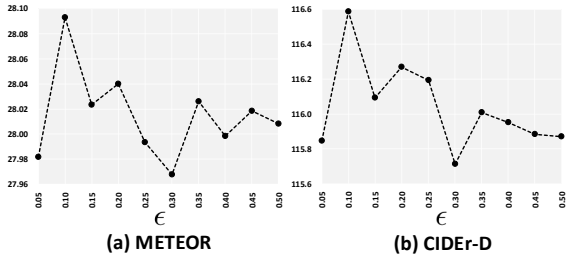


Figure 6. The effect of the threshold parameter ϵ for constructing hierarchy in Up-Down+HIP with cross-entropy loss over (a) METEOR (%) and (b) CIDEr-D (%) on COCO.

annotation from that by a system? Based on labelers' feedback, we calculate two metrics: 1) M1: percentage of captions that are as well as or even better than human annotation; 2) M2: percentage of captions that pass the Turing Test. The M1 scores of GCN-LSTM+HIP, GCN-LSTM and LSTM are 76.5%, 73.9% and 50.7%, respectively. In terms of M2, Human, GCN-LSTM+HIP, GCN-LSTM, and LSTM achieve 91.4%, 85.2%, 81.5%, and 57.1%. Apparently, our GCN-LSTM+HIP is the winner on both criteria.

Effect of the threshold ϵ. To clarify the effect of the threshold parameter ϵ for constructing hierarchy, we illustrate the performance curves over METEOR and CIDEr-D with different threshold parameters in Figure 6. As shown in the figure, we can see that both performance curves are generally like the "$\wedge$" shapes when ϵ varies in a range from 0.05 to 0.5. Hence we set the threshold parameter ϵ as 0.1 in our experiments, which can achieves the best performance.

Extension to Recognition. As a feature refiner, here we test the generalizability of our HIP on recognition task. We also experiment with COCO dataset on 80 object categories and utilize object annotations as multiple labels of an image. Multi-label softmax loss [7] is exploited for classification. For each image, we predict top-3 ranked labels. Then, we compute the precision and recall for each label separately, and report per-class precision (C-P) and pre-class recall (C-R). Furthermore, to alleviate the bias towards infrequent labels, we also compute overall precision (O-P) and overall recall (O-R). As the harmonic average of precision and recall, F1 (C-F1 and O-F1) scores are given as well. Table 4 details the performances of different features on recognition task. We take image-level features from our HIP and compare to mean-pooled image features in Up-Down [3]. By delving into hierarchy parsing, image-level features from HIP lead to 2% and 3.2% performance gain in C-F1 and O-F1 over the features in Up-Down. The results basically verify the generalizability of HIP on recognition task.

Table 4. Performance comparisons on recognition task when employing different features.

	C-P	C-R	C-F1	O-P	O-R	O-F1
Up-Down [3]	65.32	62.24	63.74	64.37	66.48	65.41
HIP	**66.18**	**65.30**	**65.74**	**67.53**	**69.74**	**68.61**

5. Conclusions

We have presented HIerarchy Parsing (HIP) architecture, which integrates hierarchical structure into image encoder to boost captioning. Particularly, we study the problem from the viewpoint of interpreting the hierarchy in tree-structured topologies from the root of the whole image to the middle layers of regions and finally to the leaf layer of instances. To verify our claim, we have built a three-level hierarchal structure of constituent visual patterns (i.e., instances, regions and the whole image) in an image. A Tree-LSTM is employed on the hierarchy to enrich the features throughout all the three levels. Extensive experiments conducted on COCO image captioning dataset demonstrate the efficacy of HIP in both cases of feeding a blend of features directly from HIP or further enhanced version with relation into an attention-based sentence decoder. More remarkably, we achieve new state-of-the-art performances on this captioning dataset. The evaluations on HIP also validate its potential of generalizing to recognition task.

References

[1] Narendra Ahuja and Sinisa Todorovic. Connected segmentation tree – a joint representation of region layout and hierarchy. In *CVPR*, 2008.

[2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016.

[3] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[5] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop*, 2005.

[6] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[7] Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek, and Cordelia Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *ICCV*, 2009.

[8] Feng Han and Song-Chun Zhu. Bottom-up/top-down image parsing with attribute grammar. *IEEE Trans. on PAMI*, 2009.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[12] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *CVPR*, 2018.

[13] Wenhao Jiang, Lin Ma, Yu-Gang Jiang, Wei Liu, and Tong Zhang. Recurrent fusion network for image captioning. In *ECCV*, 2018.

[14] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[15] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[16] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017.

[17] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. Pointing novel objects in image captioning. In *CVPR*, 2019.

[18] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *ACL Workshop*, 2004.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[20] Xian-Ming Liu, Rongrong Ji, Changhu Wang, Wei Liu, Bineng Zhong, and Thomas S Huang. Understanding image structure via hierarchical shape parsing. In *CVPR*, 2015.

[21] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*, 2017.

[22] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[24] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017.

[25] Abhishek Sharma, Oncel Tuzel, and Ming-Yu Liu. Recursive context propagation network for semantic scene labeling. In *NIPS*, 2014.

[26] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*, 2015.

[27] Zhuowen Tu, Xiangrong Chen, Alan L Yuille, and Song-Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. *IJCV*, 2005.

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

[29] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[30] Jing Wang, Yingwei Pan, Ting Yao, Jinhui Tang, and Tao Mei. Convolutional auto-encoding of sentence topics for image paragraph generation. In *IJCAI*, 2019.

[31] Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, and Anton van den Hengel. What value do explicit high level concepts have in vision to language problems? In *CVPR*, 2016.

[32] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[33] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Incorporating copying mechanism in image captioning for learning novel objects. In *CVPR*, 2017.

[34] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *ECCV*, 2018.

[35] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. Boosting image captioning with attributes. In *ICCV*, 2017.

[36] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *CVPR*, 2016.

[37] Long Zhu, Yuanhao Chen, Yuan Lin, Chenxi Lin, and Alan Yuille. Recursive segmentation and recognition templates for image parsing. *IEEE Trans. on PAMI*, 2012.