Attribute-Guided Attention for Referring Expression Generation and Comprehension

Jingyu Liu^{iD}, Wei Wang, Liang Wang, *Fellow, IEEE*, and Ming-Hsuan Yang^{iD}, *Fellow, IEEE*

Abstract—**Referring expression is a special kind of verbal expression. The goal of referring expression is to refer to a particular object in some scenarios. Referring expression generation and comprehension are two inverse tasks within the field. Considering the critical role that visual attributes play in distinguishing the referred object from other objects, we propose an attribute-guided attention model to address the two tasks. In our proposed framework, attributes collected from referring expressions are used as explicit supervision signals on the generation and comprehension modules. The online predicted attributes of the visual object can benefit both tasks in two aspects: First, attributes can be directly embedded into the generation and comprehension modules, distinguishing the referred object as additional visual representations. Second, since attributes have their correspondence in both visual and textual space, an attribute-guided attention module is proposed as a bridging part to link the counterparts in visual representation and textual expression. Attention weights learned on both visual feature and word embeddings validate our motivation. We experiment on three standard datasets of RefCOCO, RefCOCO+ and RefCOCOg commonly used in this field. Both quantitative and qualitative results demonstrate the effectiveness of our proposed framework. The experimental results show significant improvements over baseline methods, and are favorably comparable to the state-of-the-art results. Further ablation study and analysis clearly demonstrate the contribution of each module, which could provide useful inspirations to the community.**

Index Terms—**Referring expression, generation, comprehension, attributes, attribute-guided attention.**

I. INTRODUCTION

REFERRING expression is often a noun phrase to identify an object in a discourse. It is frequently used in our daily conversation when a speaker needs to refer or indicate a
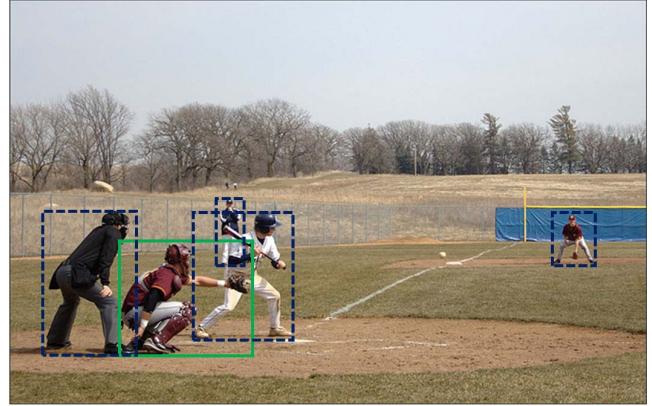
Fig. 1. Referring expression in everyday life to identify an object. The green box and blue boxes stand for the referring object and other objects respectively. Both the attributes "closer" and "red" make the target unambiguous.

particular object to a listener. Imagine a dialogue between two viewers before a crowd of people in Figure 1. The speaker can use the expression "The closer boy in red" to refer the target, then the listener can successfully comprehend which person is referred to by attributes of "closer" and "red". Note that lacking either attribute will make it ambiguous.

Regarding the two tasks in computer vision, referring expression generation and comprehension are mutually inverse. The task of generation requires the model to generate unambiguous expressions for a target object in the image. On the other side, comprehension requires the model to understand the received expression, accomplishing it by localizing the referred object in the image. Figure 2 illustrates referring expression comprehension and generation in two rows respectively. The green and blue boxes denote ground truths and comprehended objects respectively.

Referring expression comprehension is a newer task which outputs the object's location given the expression. Practical approaches often accomplish this task in two steps: First, generate a group of candidate objects via object detectors. Second, pick the referred object from the candidates. Recent approaches focus on how to design a ranking-based strategy to retrieve the referred object in the second step, and mainly formalizing it in two ways. The first one addresses the problem as the inverse process of the generation. By the generation model, the probability $P(r|o)$ of the referring expression r given the object o can be obtained. By Bayes's rule, given r, $P(o|r)$ can be obtained by converting $P(r|o)$. The second one addresses the problem in a image/text retrieval approach. The visual and textual representation of the target object are embedded into a common space, then a distance metric is
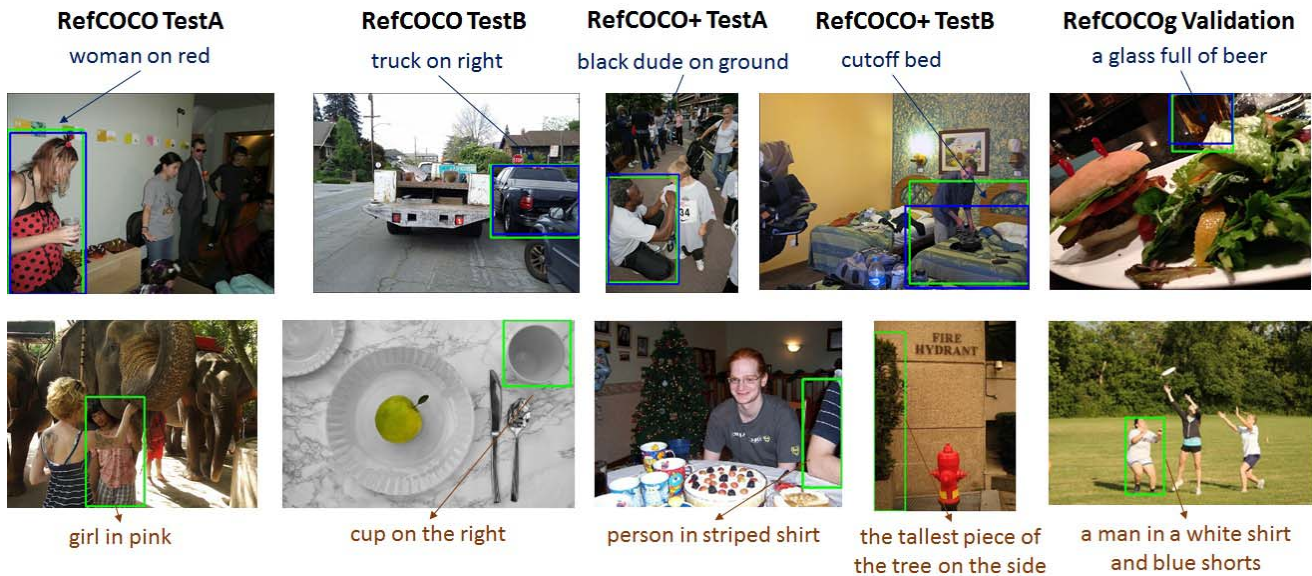
Fig. 2. Some examples of referring expression comprehension (first row) and generation (second row) based on our proposed attribute-guided attention model. The green box denotes the referring object and the blue box denotes the comprehended ones.

learned. The referred object can be retrieved by computing similarity with the embedded expression in the common space.

As a subtask of natural language generation (NLG), Referring expression generation (REG) has a longer history. REG has been studied since the 1970s [1]–[4], when ground truth attributes are given as building blocks of the expression. The major concern lies on how to select attributes to build an expression uniquely describing the object. The generation process is often accomplished by rule-based algorithms, leading to highly formatted expressions. REG based on vision is inherently a different task, wherein the attributes are not available. In this modern branch, expressions are generated on the referred object in the image. Modern vision-based REG methods benefit a lot from the powerful encoding of both the image and the language. Early works like Mao *et al.* [5] are the first to use deep neural networks, where Convolutional Neural Network (CNN) and Long Short-term Memory (LSTM) [6] are the encoding methods for the visual and language parts respectively.

Considering the core roles that attribute play in distinguishing the referred object from other distractions. Our approach aims to re-utilize the attributes once used in traditional referring expression generation. Kazemzadeh *et al.* [7] categorizes the attributes in referring expression into 7 types: category name, color, size, absolute location, relative location, relative object, and generic attribute. While in [7], attributes for each type are pre-defined manually and the generated expression is the selected attributes. Liu *et al.* [8] are the first to automatically learn and predict attributes in referring expression. In [8], attributes are defined in a broader sense of actions, patterns along with original ones, and collected from expressions without manual categorization. Then an off-line attribute learning model is learned via the one-to-many mapping of visual object and the associated attributes as labels. Finally, the obtained visual feature of attributes can be embedded into both generation and comprehension.

This paper is an extended version of our initial conference paper [8]. In this paper, we not only use attributes as additional features [8], but also adopt them to guide the attention learning on visual and textual spaces. Since attributes are the most prominent elements that distinguish the referred object from other objects, they are highly possibly reflected in both the visual feature and the textual description of the referred object. It is desired to let the attribute-corresponding parts in visual feature and expression words have more attention. To be more specific, we use the learned attributes as the guiding signal on both the visual feature and the expression. Inspired by the successful attention mechanism on both language tasks and vision tasks, we here use attributes to guide the visual and word attention to focus more on the distinguishing parts.

Early datasets [2], [3], [9] of referring expression mainly use synthesized images in artificial scenarios. Recently, larger and natural image datasets are built and the comprehension task is included, as well as real-world interactions with robotics [10], [11]. Kazemzadeh *et al.* [7] introduce the first large-scale REG dataset of 20k natural images from the ImageClef dataset [12], by a two-player interactive game. Later, Yu *et al.* [13] introduce the RefCOCO and RefCOCO+ dataset by collecting images from the MSCOCO dataset [14]. In addition, RefCOCOg from Mao *et al.* [5] is also based on MSCOCO, while the expressions are longer and more complex. We evaluate our attribute-guided attention model on the three standard benchmarks, RefCOCO, RefCOCO+ and RefCOCOg.

The remainder of the paper is organized as follows. In Section II, we discuss related works. In Section III, we describe the the proposed model and its analysis. In Section IV, we evaluate the performance on the datasets of RefCOCO, RefCOCO+ and RefCOCOg. In Section V, we discuss limitness of our work and future direction. In Section VI, we give the conclusion of our work.

II. RELATED WORKS

Referring expression generation and comprehension are at the intersection of vision and language, and involve several tasks, including image/region caption [8], [15]–[28], visual-semantic embedding [29]–[40] and object detection/localization [41]–[43]. Tasks of visual question answering (VQA) [44]–[46] and word/phrase grounding are also related. Besides, the requirement of unambiguity in referring expression drives researchers to design particular features and algorithms. We here review related works from the above respects.

A. Image/Region Caption

Image/region caption is similar to the task of referring expression generation, while the latter generates shorter noun phrases, and focuses more on the unambiguity. Modern approaches of image caption are based on the CNN-RNN architecture [15], [16]. The CNN feature extracted from the image is taken as the input to the RNN/LSTM network, either at the very beginning step or at all time steps. In the training stage, word tokens are taken as inputs to each time step. In the testing stage, word tokens sampled from previous step are input to the next step. Attention model originally proposed for natural language translation [47] is later successfully used in image caption. Xu *et al.* [18] are the first to introduce the attention model to image caption. Later, variations of attention models [19]–[22] are emerging. Attributes or high level semantics are also explored, either as embedded features or combined with attention. It should be noted that our method differs from them that we use attributes as the guidance for the attention on both visual and textual features. A more similar task to referring expression generation is region caption, like [26] providing a dense group of region captions, or [27] giving structured alignment of words/phrases and regions. These tasks do not focus on the unambiguity of the caption, therefore differ from referring expression generation.

B. Visual-Semantic Embedding

Some recent referring expression comprehension approaches rely on visual-semantic embedding algorithms. Frome *et al.* [29] propose the first visual-semantic embedding framework, where they use CNN and Skip-Gram to encode image and words, along with the ranking based training strategy. Kiros *et al.* [30] replace the Skip-Gram [31] with LSTM to encode the sentence in a similar framework. Vendrov *et al.* [32] consider the order structure of visual semantic hierarchy with a new objective. Wang *et al.* [33] add within-view constraints to preserve structure constraints. Yan and Mikolajczyk [34] use deep canonical correlation analysis as the objective, pushing the paired image-sentence closer with high correlation. Based on the similar framework, Klein *et al.* [35] use Fisher Vectors (FV) [36] to learn more discriminative representations for sentence. Lev *et al.* [37] alternatively use RNN to aggregate FV and Plummer *et al.* [38] explore the region-phrase correspondences.

C. Object Detection/Localization

The output of referring expression comprehension is the bounding box of the target object. This naturally accords with the task of object localization [48], wherein the output is also the bounding box of the single object in the image. The specialty of referring expression comprehension is that it has an extra input of the expression, and always contains multiple objects in the image. A direct thinking is to address the task in the approach of object localization, i.e., regressing the coordinates of the target object. But the method turns out not to be working well. One major reason is the limited data in this field. Therefore, current comprehension methods have to take two steps to accomplish the task in practice. The first step is to use modern object detectors to obtain a group of candidate objects. Then, the target object is retrieved from the group of objects. Thus, the final result is directly dependent on the quality of the object detector. There have been maturely developed detectors, like the two-stage RCNN series [41]–[43], or the one-stage SSD/YOLO series [49]–[51]. Further discussion is beyond the scope of this paper.

D. Referring Expression

Referring expression generation and comprehension are two inverse and complementary tasks. There are papers like [52]–[54] addressing single tasks of either generation or comprehension, as well as papers like [5], [13], [55] on both tasks. The connection between two tasks is interesting and they impact each other. For instance, the comprehension can be accomplished by using the trained generation model, wherein the probability $P(r|o)$ of the referring expression r given the object o can be obtained. By Bayes's rule, given r, $P(o|r)$ can be obtained by converting $P(r|o)$. Then we can pick the one with the maximum $P(o|r)$ as the target object. On the other side, a trained comprehension model can be either used as a guidance during the process of generation, or as a post-ranking tool to pick the expression with least ambiguity.

Deep learning based methods start from Mao *et al.* [5] and Hu *et al.* [52]. Both approaches extract VGG features from the object regions, and encode the location/size of the objects, then input them to the LSTM model to generate the expression. For the task of comprehension, they both use the retrieval based method following Bayes' rule, picking the region maximizing $P(r|o)$. In [5], the need for unambiguity in referring expression generation is considered, where the max-margin MMI (Maximum Mutual Information) strategy is used to maximize the mutual information between the target object and its expression. The MMI training later becomes a standard strategy in referring expression. Later works of [13] and [53] both model the context of the target object. To make the target unique, [13] computes the visual difference between it and its surrounding negative objects. A tie LSTM model is also proposed to generate the expression of multiple objects together. In [53], context is modeled as the supportive description, which can be found in the expressions like "flower on the left of TV". More recent works are considering the inner connection between generation and comprehension. Both [55] and [54] use a pre-trained comprehension model as a guiding tool supervised on the training of the expression generation module. In [54], a proxy training strategy is proposed to supervise the generation module to generate more unique expressions. In [55], similar idea is

implemented in a reinforcement learning approach, where the pre-trained comprehension model is used as the reward function towards the generation process. In addition, both [55] and [54] use the comprehension model as a post-ranking tool to pick the expression from a much wider range of candidates, which differs from traditional caption approaches like beam search. Yu *et al.* [55] are also the first to integrate generation and comprehension in a joint learning model. Experiments show that the two tasks can be benefited from each other.

III. ARCHITECTURE

In this section, we review the backbone frameworks for referring expression generation and comprehension respectively in Subsection III-A and Subsection III-B. Then in Subsection III-C, we introduce the proposed attribute-guided attention model. In the rest of the paper, we use the term "generation model" and "speaker" alternatively to denote the model for the task of generation, and the term "comprehension model" and "listener" to denote the model for the task of comprehension.

A. Referring Expression Generation

The task of referring expression generation is to generate an expression r, given the input image I and a target object o (in the form of a bounding box). Formally, the generation model is trained to maximize the likelihood of the correct expression by:

$$G^* = \arg\max_G \sum_i \log P_G(r_i | I_i, o_i) \tag{1}$$

where G is the generation model, i denotes the index of the training sample. In this paper, we use the CNN-LSTM framework as the backbone of the generation model. The CNN-LSTM model is also commonly used in image caption, where the input is the image itself. Moreover, the motivation of referring expression is to generate descriptions that distinguish the target object from other objects.

The CNN module is used to encode the visual features, and the LSTM is used to decode it and generate the sequence. Human in real world often use appearance and location word to refer to their target. Following previous works [5], [13], we use the VGG-fc7 feature extracted from the object region as the basic visual appearance feature o_i. For the location feature l_i we use the commonly used 5-dimension vector $\left[\frac{x_l}{W}, \frac{y_t}{H}, \frac{x_r}{W}, \frac{y_b}{H}, \frac{w \cdot h}{W \cdot H}\right]$ to encode it. x_l, y_t, x_r, y_b are the left, top, right and bottom coordinates of the object region and w, h, W, H are widths and heights of the region and the image. Additionally, other features e.g. global features g_i of the whole image and comparison features δ_i [13] are also used to improve the performance. In this paper, we use o_i and l_i as the basic components for the visual feature v_i, and optionally use δ_i. The final visual representation v_i of the target object is a concatenation of above features followed by a fully-connected layer W_t.

$$v_i = W_t([o_i, l_i, \delta_i]) + b_t \tag{2}$$

where δ_i is concatenation of δv_i and δl_i. $\delta v_i = \frac{1}{n} \sum_{j \neq i} \frac{o_i - o_j}{\|o_i - o_j\|}$ is the appearance difference feature and δl_i is the location/size difference feature. For the generation part, LSTM receives the word token at each time step and outputs the word for the next step. Similar to approaches in image caption, the visual feature v_i is taken as another input to the LSTM module. In this paper, we input v_i at all time steps since we find it works better in practice. Therefore the LSTM is formulated as follows:

$$i_t = \sigma\left(W_{ix}x_t + W_{ih}h_{t-1} + W_{iv}v_i + b_i\right) \tag{3}$$

$$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + W_{fv}v_i + b_f\right) \tag{4}$$

$$o_t = \sigma\left(W_{ox}x_t + W_{oh}h_{t-1} + W_{ov}v_i + b_o\right) \tag{5}$$

$$c_t = f_t \odot c_{t-1}$$
$$\quad + i_t \odot \tanh\left(W_{cx}x_t + W_{ch}h_{t-1} + W_{cv}v_i + b_c\right) \tag{6}$$

$$h_t = o_t \odot \tanh\left(c_t\right) \tag{7}$$

$$prob_t = softmax\left(W_p h_t + b_p\right) \tag{8}$$

where x_t is the input token word at each time step and the various W matrices and b are the training parameters. i_t, f_t, o_t, c_t and h_t are input gate, forget gate, output gate, memory cell and hidden states respectively. $prob_t$ is the probability of the output word tokens. The whole model can then be trained by minimizing the cross entropy loss, or equivalently the negative log-likelihood:

$$L_1(\theta_G) = -\sum_i \log P(r_i | o_i; \theta_G)$$

$$= -\sum_i \sum_{t=1}^{T} \log P(r_{i,t} | r_{i,<t}, o_i; \theta_G) \tag{9}$$

where θ_G is the parameter of the generation model G. To model the property that no two objects in the same image should be described by the same expression, we follow the paradigm of MMI training in [5]. We use the triplet hinge loss to encourage the target object to have a larger probability than other objects towards its descriptions. The margin ranking loss is formulated as follows:

$$L_2(\theta_G) = -\sum_i \max(0, M + \log P(r_i | o_k; \theta_G)$$
$$\quad - \log P(r_i | o_i; \theta_G)) \tag{10}$$

where M is the margin value.

B. Referring Expression Comprehension

The task of referring expression comprehension requires the comprehension model to be a good listener. Moreover, the listener needs to prove its understanding by pointing to the target object, specified by the bounding box of the object. Therefore the listener should also have a good sense of object-ness. We evaluate both situations when ground-truth candidate objects are available or not. The former case focuses on the comprehension model itself, while the latter can evaluate a more practical system using object detectors. To this end, the input of the task is an image I, a set C of candidate regions (objects) o and a referring expression r. As mentioned before, approaches addressing comprehension can be split into two types: speaker based approach and the common space embedding model.
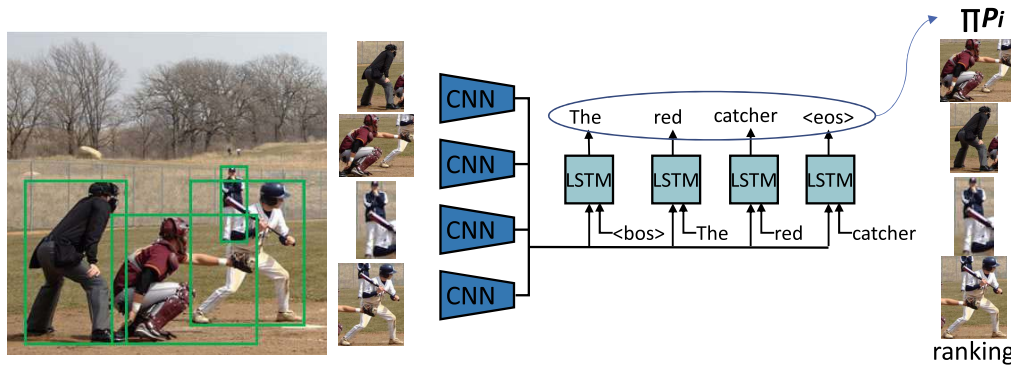
Fig. 3. Illustration of the speaker based comprehension approach.

1) Speaker Based Approach: The goal is to pick the object region with $o^* = \arg\max_{o \in C} P(o|r)$, but the value of $P(o|r)$ is not directly available. The method is to obtain $P(o|r)$ via $P(r|o)$, which is the product of output probability generated by the trained speaker model. Then by Bayes' rule, we have

$$P(o|r) = \frac{P(r|o)}{\sum_{r' \in C} P(o|r')} \quad (11)$$

Since the denominator is identical for all candidate regions, to select the object region $o^* = \arg\max_{o \in C} P(o|r)$, we can in turn select the one with $o^* = \arg\max_{o \in C} P(r|o)$.

Figure 3 illustrates the process of the speaker based approach of referring expression comprehension. The LSTM blocks are the trained speaker module. The left side shows the input candidate object regions. Finally, the object regions are ranked by the product of probability $\prod P_{ij}$, where i and j denotes the index of the object and the output word token respectively.

2) Common Space Embedding Model: The common space embedding model has been effectively used in the field of image/text retrieval. For the task of referring expression comprehension, it performs better than the speaker based model in practice. Effective representations in both visual and textual space are prerequisites for the common space embedding. The embedding of language has been studied a lot in the community of natural language processing. CNN and RNN based methods are commonly used to encode the words/phrases and sentences, either at a character level or a word level. In the baseline model, we use a unidirectional LSTM to encode it, and the hidden state h of the last time step is used as its final representation. For the encoding of the visual object o_i, we also use the same v_i used in the task of generation.

The next step is to project features from different modalities into the common space. MLPs with normalization are commonly adopted to do this task. Then similarity or distance metrics can be computed in the common space. In this paper, we use the inner product as the similarity function. Following the paradigm in generation, margin ranking loss can also be used here to make pairs of r_i and o_i close, and negative pairs far away. Therefore for each pair of r_i and o_i, two negative pairs of r_i and o_j, as well as r_k and o_i are sampled together, to formulate a loss function of a dual triplet margin ranking (hinge) loss. Considering the fact that objects from

different categories normally have larger variance than those from the same category, we dynamically assign different margins during training according to the sampled objects' categories

$$\begin{aligned}
L_3(\theta_C) = -\sum_i [&\lambda_1 \max(0, M_1 \mathbb{1}_{C(o_i)=C(o_j)} + M_2 \mathbb{1}_{C(o_i)\neq C(o_j)} \\
&- d(r_i, o_j; \theta_C) + d(r_i, o_i; \theta_C) \\
&+ \lambda_2 \max(0, M_1 \mathbb{1}_{C(r_i)=C(r_k)} + M_2 \mathbb{1}_{C(r_i)\neq C(r_k)} \\
&- d(r_k, o_i; \theta_C) + d(r_i, o_i; \theta_C)]
\end{aligned} \quad (12)$$

where λ_1 and λ_2 are the weights of the two losses. M_1 and M_2 are different margin values. $C(o_i)$ and $C(r_i)$ indicate the category of object o_i and r_i respectively.

Figure 4 illustrates the process of the common space embedding model. The object region of the red catcher and the expression "the red catcher" compose the positive pair of $P(o^+, r^+)$. r^- and o^- are "the white batter" and the object region respectively.

C. Attribute-Guide Attention for Referring Expression

1) Attribute Learning: In [7], attributes are categorized into 7 types: category name, color, size, absolute location, relative location, relative object, and generic attribute. In this paper, the definition of attributes is partially overlapped with that in [7], we include category name, color, material, actions, etc., in our attribute set.

So how to obtain the corresponding attributes of each referred object and the whole attribute set? Based on the above design, they are already reflected in forms of human words in referring expressions. We use stanford NLP parser 3.5 to parse referring expressions in the training set into part-of-speech tags. After that, nouns, verbs, and adjectives are preserved. Words of different tenses (mainly active and passive voices) and pluralities are unified to make the attribute set more accurate and concise. We also use GloVe to compute cosine distance among words, then unify synonyms, e.g. "bike" and "bicycle" if their distance is less than a threshold. Finally, we preserve the top m frequent attributes as our final attribute set. Besides, a mapping from words to attributes is also obtained, thus the corresponding attribute labels of each referred object is obtained.

Formally, the attribute set is denoted by $A = [a_0, a_1, \ldots, a_m]$. For a referred object o_i, its associated attribute labels
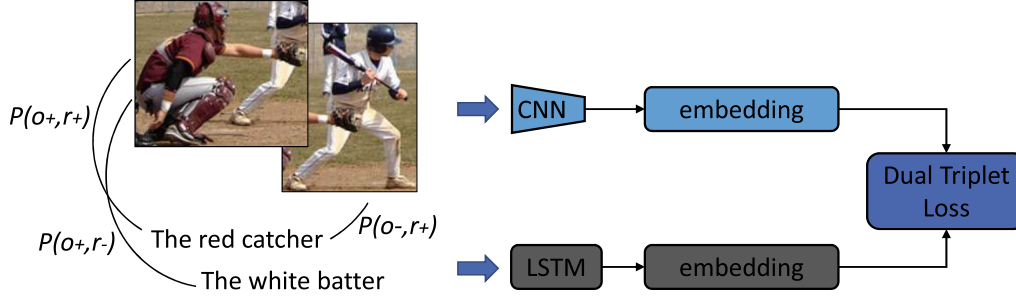
Fig. 4. Illustration of the common space embedding approach of comprehension.

are a set of attributes $a_i, a_j, \ldots, a_k$ of arbitrary number. To predict attributes for the referred object in the testing set, an attribute learning model needs to be trained. Therefore we directly formulate it as a multi-label multi-classification problem. To explore the most effective loss function for the problem, we experiment on two commonly used loss functions: binary cross-entropy loss and margin ranking loss. The function of binary cross-entropy loss to minimize is:

$$loss(p, y) = \sum_i^m \left[y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right] \quad (13)$$

The function of margin ranking loss to minimize is:

$$loss(p, y) = \sum_{i \neq y_j}^m \sum_{y_j \neq 0}^m [\max(0, 1 - (p_{y_j} - p_i))] \quad (14)$$

where $y_i \in \{0, 1\}$ denotes the existence of the ith attribute. p_i is the predicted probability. m is the number of attributes. Later in experimental part we will compare accuracy of the two losses. Here we use $L_4(\theta_A)$ to denote $loss(p, y)$.

Different from [8] using attribute learning model as an off-line tool to extract attributes, We here integrate it into the training of the generation and the comprehension modules in an end-to-end framework. The predicted attributes are embedded into the following modules and adopted as the guiding signal of attentions.

2) Attribute-Guided Attention Model: After the attribute learning model is trained, we can obtain a probability score list of attributes a_i for each object region, where $a_i = [a_{i1}, a_{i2}, \ldots, a_{im}]$ and m is the size of the attribute set. each $a_{ik} \in [0, 1]$ denotes the predicted probability of a particular attribute. There are two approaches to utilize the predicted attributes. The first approach is to directly use the probability list as a soft-coded list. In this approach, though extracted ground-truth attributes in the training expressions are available, we still use the predicted attributes. The second approach is to threshold the soft-coded list to obtain a hard-coded list. In this approach, we directly use the extracted ground-truth attributes in the training expressions. Therefore, attribute lists in training and testing are consistent in both approaches. We report results of both approaches in Table VII.

In our previous work [8], the attribute list a_i is directly used as an input feature to the generation and comprehension model, and the results demonstrate its effectiveness. In this paper, we try to exploit the potential of attributes into a deeper level. To be more specific, we use it as the guiding signal of attention on both visual objects and the referring expression.

Attention model has been successfully used in both natural language processing and computer vision. It is the mechanism to automatically learn which parts of the visual or the language representation should be more weighted. The guiding process of the attention model is various, e.g., language guided visual attention, vision guided language attention, co-attention and self-attention. In this paper, we propose the attribute-guided attention. The advantage of attribute-guided attention is that the attributes are directly reflected in both the visual object and the language description. So it is naturally the bridging part between two modalities. For instance, a girl in the image has the unique attributes of "hat" and "red shirt". So if the visual parts of hat and shirt, as well as the word encodings of "hat" and "red shirt" can have more attention, it makes the speaker and the listener easier to understand each other.

In particular, we apply visual attention in both spatial-wise and channel-wise approaches. That corresponds with that particular attributes only appear in some locations and some channels of the feature map. For the spatial-wise part, we use the conv5-3 feature map V as the visual feature of the original image. $V \in R^{H \times W \times C}$ is a tensor like spatial grids of channels, where H, W and C are height, width and channels respectively.

For the spatial-wise attention, with the soft attribute list a, we compute its attention on each grid:

$$S_a = tanh([W_s V + W_{a,s} a])$$
$$\alpha^v = softmax(w_s^T S_a)$$
$$\overline{V} = \sum_{i=1}^{W \times H} \alpha_i^v \overline{v}_i \quad (15)$$

where S_a and α_v are inner state and attention weights respectively. $\overline{v}_i \in R^C$ are tensors of each grid, and $\overline{V}$ is the weighted sum of $\overline{v}_i$.

For the channel-wise attention, we use the same feature map. Given the soft attribute list a, we compute its attention on each channel:

$$P_a = tanh([W_p V + W_{a,p} a])$$
$$\beta^v = softmax(w_p^T P_a)$$
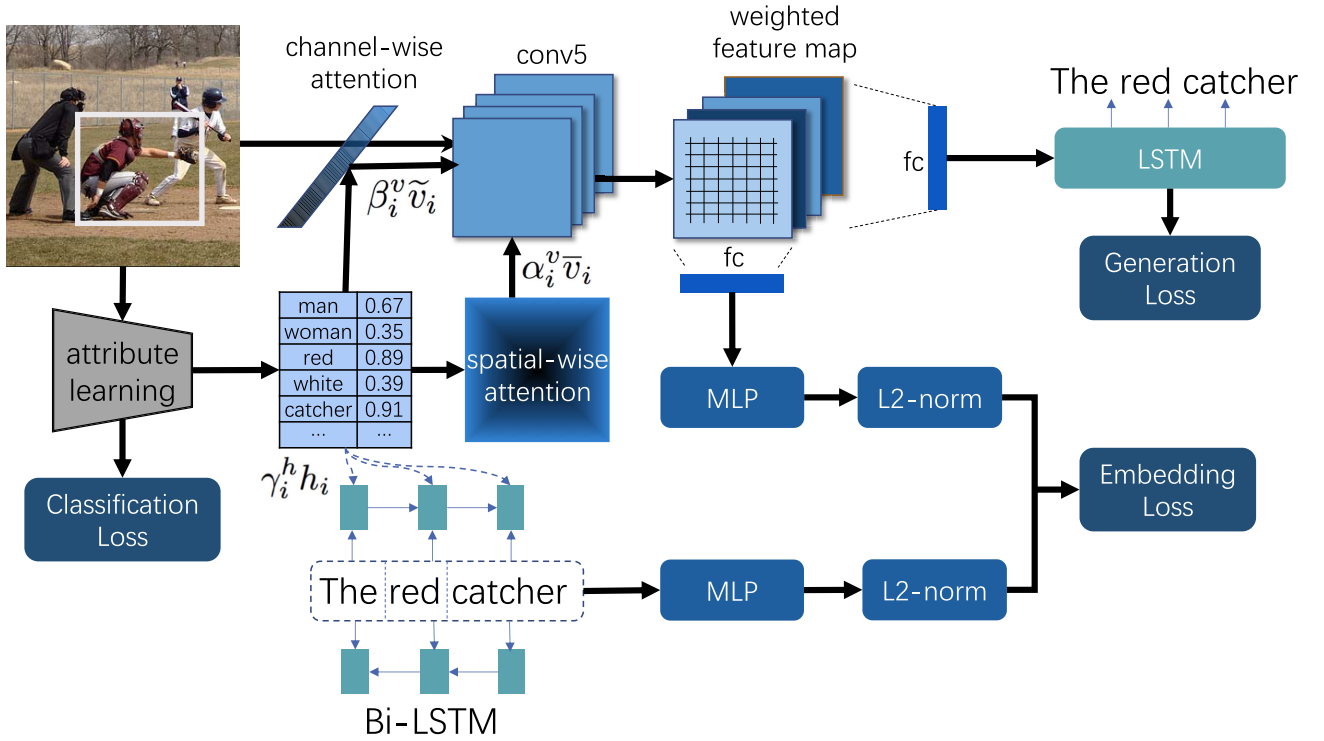$$\widetilde{V} = \sum_{i=1}^{C} \beta_i^v \widetilde{v}_i \quad (16)$$

Fig. 5. Joint generation-comprehension model with attribute-guided attention. The learned attributes can guide attention weights on both visual and texual parts. For the visual part, channel-wise and spatial-wise attention are learned on conv5 feature maps. For the texual part, token-wise attention is learned on hidden states of bidirectional LSTM of the expression. The weighted feature map then goes through fully connected layers to be the encoder for generation, and to the common space for comprehension. The output of Bi-LSTM is also embedded to the common space for comprehension. For simplicity, we omit the display of negative region/expression and the corresponding margin ranking loss.

where $\widetilde{v}_i \in R^{W \times H}$ are tensors of each single-channel feature map, and $\widetilde{V}$ is the weighted sum of $\widetilde{v}_i$.

Similarly, for the word attention part, a bidirectional LSTM is used as the backbone for the word level embedding. Additionally, the hidden states H at each step is treated as the embedded representation of each word. $H \in R^{2T \times d}$ where T and d are numbers of time steps and dimensions respectively. Given the soft attribute list a, we compute its attention on hidden states at all time steps:

$$Q_a = tanh([W_h H \otimes W_{a,h} a])$$
$$\alpha^h = softmax(w_q^T Q_a)$$
$$H_a = \sum_{i=1}^{T} \gamma_i^h h_i \qquad (17)$$

where $\otimes$ is the element-wise multiplication operator. Q_a and α_h are inner state and attention weights respectively. As the final representation of a referring expression, H_a is the weighted sum of H. After that, V_a and H_a from two modalities are embedded with FC and normalization layers into the common space.

3) Joint Speaker-Listener Model With Attribute-Guided Attention: Inspired by [55], we integrate the learning of attributes, the generation module, and the comprehension module into a joint learning framework. Figure 5 illustrates our proposed framework. The visual attention on the conv5 feature is both aware on the generation and the comprehension process. Besides, the parameters of the attribute learning module receive the back propagating gradients from both

streams of the speaker and listener. In conclusion, we believe that the attribute-guided attention model can be more effective in a joint learning process. The final goal is to minimize the following overall loss function:

$$\theta = \arg\min \beta_1 L_1(\theta_G) + \beta_2 L_2(\theta_G) + \beta_3 L_3(\theta_C) + \beta_4 L_4(\theta_A)$$
$$(18)$$

where β_1, β_2, β_3 and β_4 are weights of each part respectively.

IV. EXPERIMENTS

A. Datasets

We evaluate our results on the three datasets of RefCOCO, RefCOCO+, and RefCOCOg. Images of all datasets are collected from MS COCO images [14], but with different focuses.

RefCOCO [13] contains 142,209 referring expressions for 50,000 objects in 19,994 images from COCO [14]. The dataset is collected using an interactive interface called ReferitGame [7]. The average length of expressions is 3.61 words. Since people are much more frequent than other objects in the dataset, the test set is split into split A and split B, containing only persons and only other objects respectively.

RefCOCO+ [13] contains 141,564 referring expressions for 49,856 objects in 19,992 images from COCO. This dataset is also collected using ReferitGame, but annotators are forbidden to use explicit location words to describe the object. Therefore this dataset focuses more on the appearance-based description.

TABLE I

REFERRING EXPRESSION GENERATION RESULTS COMPARED WITH PREVIOUS METHODS EVALUATED BY AUTOMATED METRICS ON REFCOCO, REFCOCO+ AND REFCOCOG. THE RESULTS WITH * ARE EVALUATED ON THE ORIGINAL TEST SETS OF REFCOCO AND REFCOCO+

RefCOCO

		TestA					TestB				
		Bleu1	Bleu2	Rouge	Meteor	CIDEr	Bleu1	Bleu2	Rouge	Meteor	CIDEr
1	baseline+MMI, Mao [5]	0.478*	0.295*	0.418*	0.243	0.615	0.547*	0.341*	0.497*	0.300	1.227
2	baseline+MMI+visdif, Yu [13]	0.494*	0.307*	0.441*	0.260	0.679	0.578*	0.375*	0.531*	0.319	1.276
3	baseline+visdif+tie [13]	0.510*	0.318*	0.446*	0.283	0.681	0.593*	0.386*	0.533*	0.320	1.273
4	speaker+listener+MMI [55]	-	-	-	0.268	0.704	-	-	-	0.327	1.303
5	speaker+listener+reinforcer+MMI [55]	-	-	-	0.268	0.697	-	-	-	0.329	1.323
6	speaker+attr, Liu [8]	0.732	0.562	0.609	0.274	0.710	0.759	0.587	0.661	0.313	1.257
7	speaker+attr+visdif, Liu [8]	0.734	0.559	0.620	0.284	0.735	0.767	0.585	0.677	0.323	1.295
8	speaker+attr+attn	0.744	0.582	0.632	0.295	0.790	0.749	0.571	0.661	0.314	1.236
9	speaker+attn+attr+visdif	0.754	0.592	0.650	0.312	0.802	0.772	0.589	0.689	0.332	1.301

RefCOCO+

		TestA					TestB				
		Bleu1	Bleu2	Rouge	Meteor	CIDEr	Bleu1	Bleu2	Rouge	Meteor	CIDEr
1	baseline+MMI, Mao [5]	0.370*	0.202*	0.346*	0.199	0.462	0.324*	0.167*	0.320*	0.189	0.679
2	baseline+MMI+visdif, Yu [13]	0.386*	0.221*	0.360*	0.202	0.475	0.327*	0.172*	0.325*	0.196	0.683
3	speaker+visdif+tie [13]	0.409*	0.232*	0.372*	0.204	0.499	0.340*	0.178*	0.328*	0.196	0.683
4	speaker+listener+MMI [55]	-	-	-	0.208	0.496	-	-	-	0.201	0.697
5	speaker+listener+reinforcer+MMI [55]	-	-	-	0.204	0.494	-	-	-	0.202	0.709
6	speaker+attr, Liu [8]	0.574	0.379	0.499	0.219	0.512	0.463	0.285	0.447	0.203	0.704
7	speaker+attr+visdif, Liu [8]	0.575	0.380	0.507	0.223	0.514	0.471	0.285	0.459	0.211	0.717
7	speaker+attr+attn	0.612	0.428	0.556	0.264	0.651	0.489	0.335	0.491	0.243	0.744
8	speaker+attn+attr+visdif	0.603	0.414	0.529	0.236	0.585	0.461	0.284	0.449	0.206	0.692

RefCOCOg

		Val				
		Bleu1	Bleu2	Rouge	Meteor	CIDEr
1	baseline+MMI, Mao [5]	0.428*	0.263*	0.354*	0.149	0.585
2	baseline+visdif+MMI, Yu [13]	0.430*	0.262*	0.356*	0.147	0.573
3	baseline+visdif+tie, Yu [13]	-	-	-	-	-
4	speaker+listener+MMI, Yu [55]	-	-	-	0.150	0.589
5	speaker+listener+reinforcer+MMI, Yu [55]	-	-	-	0.154	0.592
6	speaker+attr, Liu [8]	0.428	0.265	0.370	0.157	0.639
7	speaker+attr+visdif, Liu [8]	0.417	0.254	0.366	0.153	0.617
8	speaker+attr+attn	0.442	0.289	0.392	0.181	0.671
9	speaker+attr+attn+visdif	0.430	0.268	0.374	0.163	0.645

TABLE II

HUMAN EVALUATION RESULTS ON REFCOCO AND REFCOCO+

		RefCOCO		RefCOCO+	
		TestA	TestB	TestA	TestB
1	speaker+attr, Liu [8]	76%	72%	43%	38%
2	speaker+attr+visdif, Liu [8]	78%	83%	41%	43%
3	speaker+attn+attr+visdif	83%	87%	49%	46%

The average length of expressions is 3.53 words. The split in RefCOCO+ follows the same rule used in RefCOCO.

RefCOCOg [5] contains 85,474 referring expressions for 54,822 objects in 26,711 images from COCO. Different from RefCOCO and RefCOCO+, this dataset is collected using a non-interactive setting and contains much longer expressions. The average length of expressions is 8.43 words. The split of this dataset is on a per-object basis, thus the same image could appear in both training and validation sets.

B. Parameter Setting and Optimization

We use the conv5-3 feature map after a 2×2 max-pooling layer of the VGG-19 model as the visual attention field. The feature map is $V \in R^{D \times c}$, where D and c is 7×7 and 512 respectively. The VGG-19 model is pre-trained on Imagenet of 1000 categories. The convolutional layers of VGG-19 are fixed during the training of our proposed model. For the semantic embedding of the expressions, the time length of LSTM is set to 10, and the encoding size of the hidden state is set to 512. Therefore the hidden states $H \in R^{T \times d}$ is of size 10×512. Both the generation and comprehension modules share the same word embedding weight, and the embedding size is set to 512. For FC layers used in the common space embedding module, we find a single layer of size 512 after both visual and expression embedding performs better than multiple layers. Finally, the embedding size in the common space is set to 512.

In Eq. 18, the weight of each loss module is simply set to be equal, thus β_1, β_2, β_3 and β_4 are all set to 1. The margin of M in Eq. 10 is set to 1. The margins of M_1 and M_2 in Eq. 12 are set to 0.1 and 0.2. The model is optimized using Adam [57] with an initial learning rate of 5×10^{-4}, halved every 5,000 iterations, with a batch size of 16.

1) The Choice of L_4: We experiment the two losses: binary cross-entropy loss and margin ranking loss on the validation sets of three datasets, with number of attributes set to 100. The precision, recall and F1 score are shown in Table V. From the classification results we can see that binary cross-entropy loss is slightly better than margin ranking loss. Therefore, in the following experiments, we use binary cross-entropy loss for L_4.
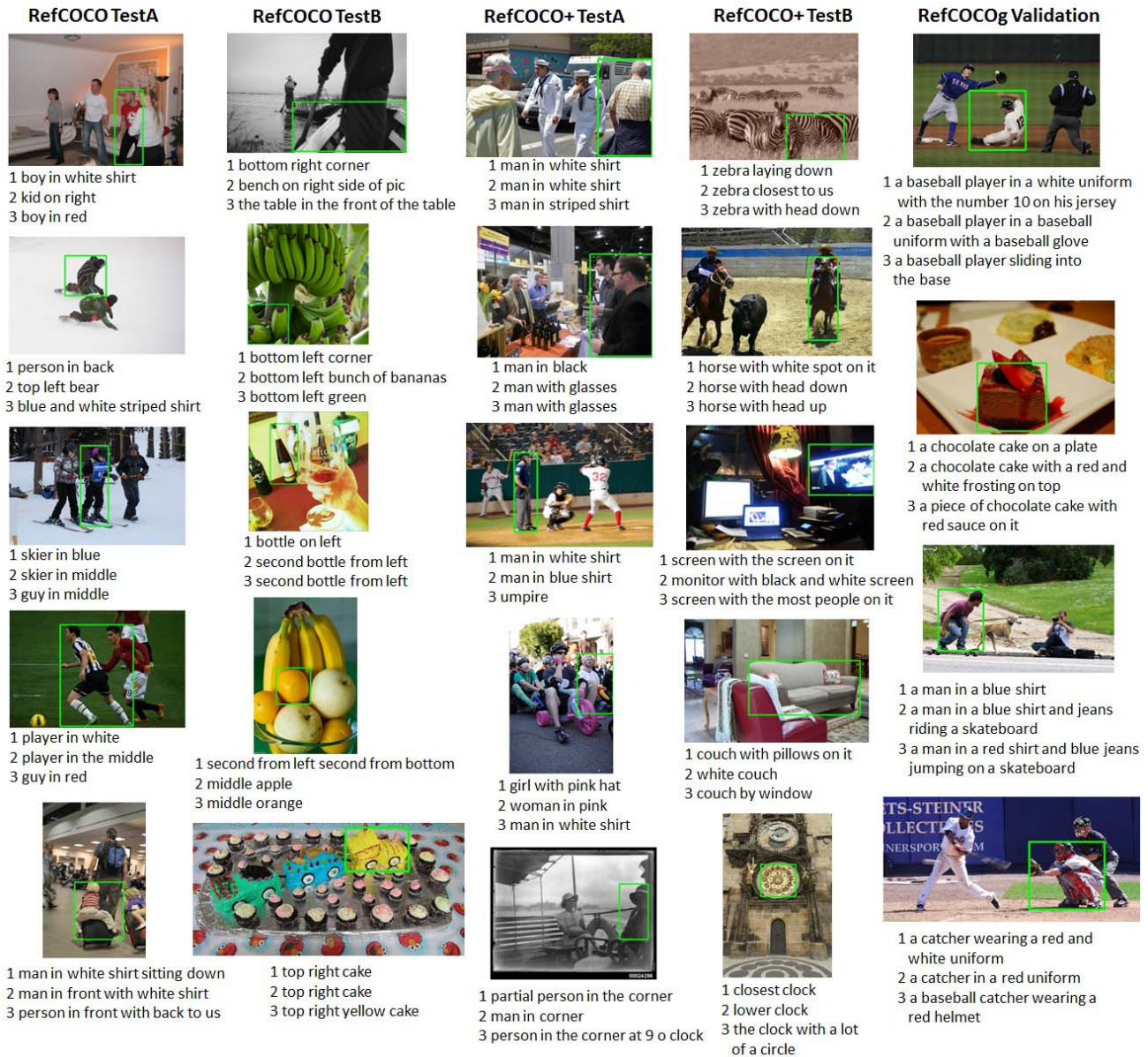
Fig. 6. Some results of referring expression generation based on 1 speaker+attr 2 speaker+attr+visdif and 3 speaker+attr+visdif+attn. The green boxes denote the referring objects.

2) The Choice of Number of Attributes: We experiment different number of attributes to find the best size of the attribute set. Specifically, we test the size of 50, 100 and 200, measured by attribute classification accuracy, generation results and comprehension results on RefCOCO. The results are shown in Table VI The results show that with the number of attributes increases, the classification accuracy decreases. But the generation and comprehension results are better of size 200 than 100 and 400. We believe that larger size of attributes can include more information, while too many attributes will largely decrease the classification accuracy, thus devastate the generation and comprehension performance. Therefore, in the following experiments, we choose to use the size of 200.

3) The Choice of Hard List and Soft List: We also test the usage on the hard list and soft list of attributes. For the hard list, we use the threshold value of 0.5 to decide the existence of the attribute. In Table, we show the comprehension and generation results on RefCOCO. The results show that the soft list is consistently better than the hard list. We believe that both accuracy and information are lose in the process of thresholding, thus devastate the generation and comprehension performance.

C. Results: Referring Expression Generation

We evaluate the referring expression generation results with two metrics. The first is the automatic evaluations commonly used in image caption. The second is human evaluation, which is more reliable regarding the objective of referring expressions.

1) Automatic Evaluation: Automatic evaluations of BLEU, ROUGE, METEOR, and CIDEr have already been commonly

TABLE III
COMPARISON WITH PREVIOUS METHODS ON REFERRING EXPRESSION COMPREHENSION WITH GROUND TRUTH OBJECT REGIONS

		RefCOCO			RefCOCO+			RefCOCOg
		Val	Test A	Test B	Val	Test A	Test B	Val
1	baseline (no MMI), Mao [5]	-	63.15%	64.21%	-	48.73%	42.13%	55.16%
2	speaker, Mao [5]	-	71.72%	71.09%	-	58.42%	51.23%	62.14%
3	speaker+neg bag, Nagaraja [53]	-	75.6%	78.0%	-	-	-	68.4%
4	softmax, Luo [54]	-	74.04%	73.43%	-	60.26%	55.03%	65.36%
5	speaker+visdif, Yu [13]	-	73.98%	76.59%	-	59.17%	55.62%	64.02%
6	listener+visdif, Yu [55]	77.48%	76.58%	78.94%	60.50%	61.39%	58.11%	71.12%
7	speaker+listener+reinforcer, Yu [55]	79.56%	78.95%	80.22%	62.26%	64.60%	59.62%	72.63%
8	speaker+listener+reinforcer, Yu [55]	78.36%	77.97%	79.86%	61.33%	63.10%	58.19%	72.02%
9	speaker+listener+reinforcer, Yu [55]	80.36%	80.08%	81.73%	63.83%	65.40%	60.73%	74.19%
10	MatNet, Yu [56]	80.94%	79.99%	82.30%	63.07%	65.04%	61.77%	73.08%
11	listener+attr, Liu [8]	79.03%	79.30%	78.96%	63.63%	65.23%	59.83%	71.22%
12	listener+attr+visdif, Liu [8]	80.35%	79.46%	80.20%	63.62%	65.05%	58.68%	74.38%
13	listener+attr+attn	81.22%	80.67%	81.86%	65.93%	69.41%	62.47%	76.31%
14	listener+attr+attn+visdif	80.63%	80.03%	80.24%	64.64%	68.34%	61.02%	74.85%

TABLE IV
COMPARISON WITH PREVIOUS METHODS ON AUTOMATIC REFERRING EXPRESSION COMPREHENSION

		RefCOCO(det)			RefCOCO+(det)			RefCOCOg(det)
		Val	Test A	Test B	Val	Test A	Test B	Val
1	baseline (no MMI), Mao [5]	-	58.32%	48.48%	-	46.86%	34.04%	40.75%
2	speaker, Mao [5]	-	64.90%	54.51%	-	54.03%	42.81%	45.85%
3	speaker+neg bag, Nagaraja [53]	-	58.6%	56.4%	-	-	-	39.5%
4	speaker+visdif, Yu [13]	-	67.64%	55.16%	-	55.81%	43.43%	46.86%
5	listener+visdif, Yu [55]	-	71.63%	61.47%	-	57.33%	47.21%	56.18%
6	speaker+listener+reinforcer, Yu [55]	-	72.88%	63.43%	-	60.43%	48.74%	59.51%
7	speaker+listener+reinforcer, Yu [55]	-	72.94%	62.98%	-	58.68%	47.68%	57.72%
8	speaker+listener+reinforcer, Yu [55]	-	73.78%	63.83%	-	60.48%	49.36%	59.84%
9	listener+attr, Liu [8]	-	70.55%	54.80%	-	56.38%	43.14%	50.02%
10	listener+attr+visdif, Liu [8]	-	73.25%	64.83%	-	61.14%	48.39%	61.64%
11	listener+attr+attn	-	74.81%	65.04%	-	63.66%	52.28%	62.71%
12	listener+attr+attn+visdif	-	74.30%	64.18%	-	63.31%	48.76%	60.77%

TABLE V
CLASSIFICATION ACCURACY OF 200 ATTRIBUTES WITH **BINARY CROSS-ENTROPY/MARGIN RANKING** LOSS ON THE VALIDATION SETS OF THE THREE DATASETS

	precision	recall	F1
RefCOCO	52.47/52.05	25.19/24.87	33.53/33.19
RefCOCO+	50.22/50.03	15.40/15.17	27.18/26.97
RefCOCOg	53.93/53.76	26.01/25.83	33.33/33.10

TABLE VI
RESULTS COMPARISONS OF DIFFERENT ATTRIBUTE SIZE ON REFCOCO

#attrs	F1 score	Bleu1 on gen	acc on comp
100	33.73	0.629	78.94%
200	33.53	0.732	79.03%
400	32.41	0.625	78.85%

TABLE VII
RESULTS COMPARISON OF HARD AND SOFT ATTRIBUTE LISTS ON THE VALIDATION SETS OF REFCOCO

	Bleu1 on generation	accuracy on comprehension
hard list	0.618	78.20%
soft list	0.625	78.85%

used in image caption. Such evaluations basically compute the word matching score in a direct or a more complex approach. The score is determined by the generated expression with the best matching one in several ground truths. One limitation is that the generated expressions can be diverse, so none of the ground truths might be matched well even it is clear and accurate. One way to alleviate this problem is to expand the ground truth set, hoping to have a broader coverage in the expression space. Such supplementary approach has been applied in [55], wherein more expressions are collected for

referred objects in the test sets of RefCOCO and RefCOCO+. In this paper, we also use the expanded test sets in RefCOCO and RefCOCO+, and here we report the updated results of speaker+attr and speaker+attr+visdif in Lines 6-7 of Table I. Line 8 shows the result of speaker+attn+attr+visdif, which adds the attribute-guided attention module. The results show that the attention module is more effective in TestA of RefCOCO and RefCOCO+, which means the attribute-guided attention are more accurately paid to human beings. We believe this is mainly due to two reasons: First, people in everyday life always bear more visual attributes, like clothes, hairstyles, etc. Second, training samples of humans are plentiful, while non-human samples are much less per category. Visdif [13] is especially useful in modeling the order information of a referred object, e.g. "the second zebra from left". In RefCOCOg, speaker+attn+attr works best in all settings, which is due to the reason that no location information is used in this dataset, thus visdif may deteriorate the performance.
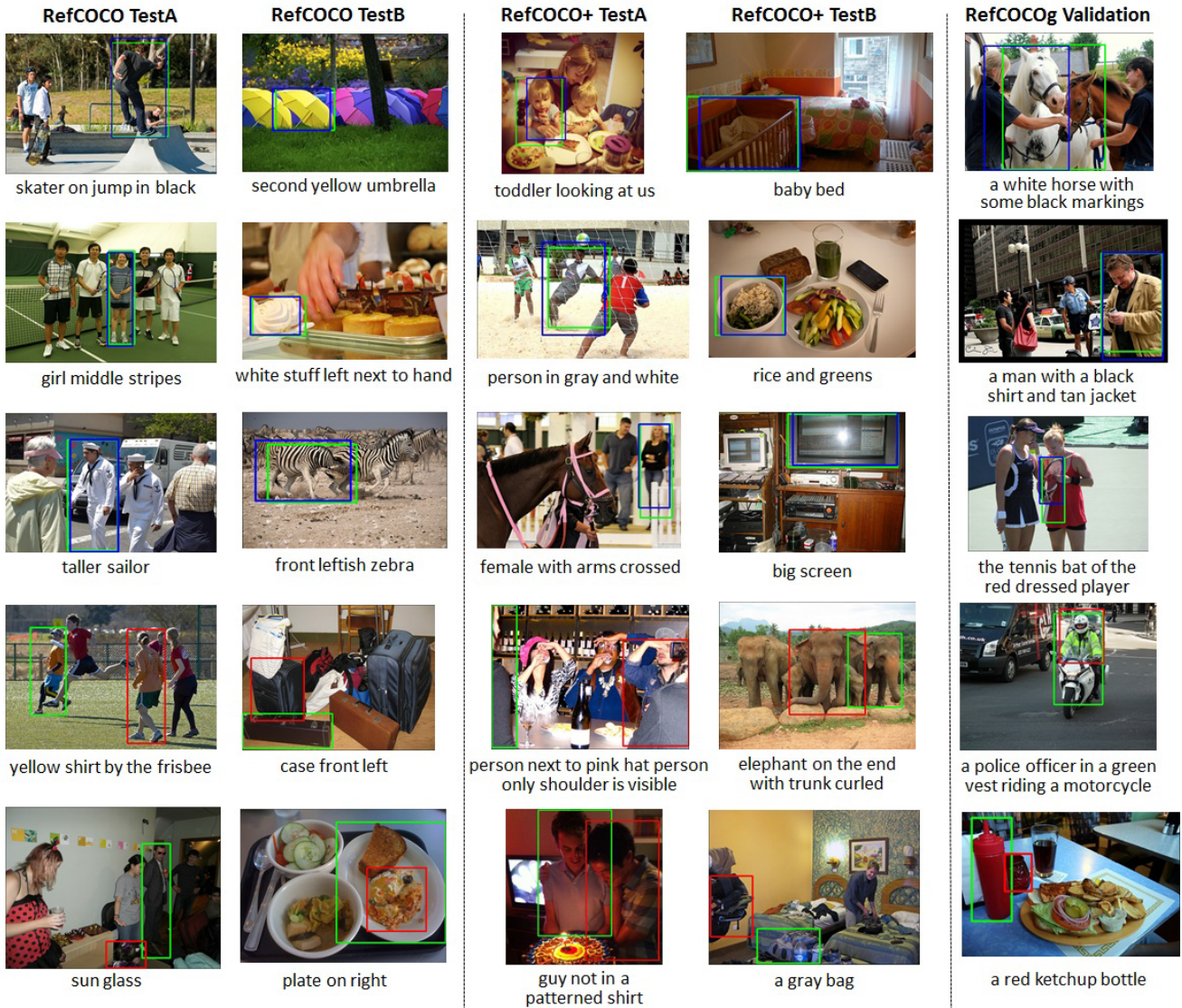
Fig. 7. Some results of automatic referring expression comprehension based on listener+attn+attn using SSD as the detector. The top three rows and the bottom two rows are successful and failure examples respectively. The green, blue and red bounding boxes are ground truths, correct hits, and wrong hits respectively.

Figure 6 shows some qualitative results in RefCOCO, RefCOCO+ and RefCOCOg. The green boxes denote the referred objects and expressions generated from 1 speaker+attr, 2 speaker+attr+visdif and 3 speaker+attr+visdif+attn are displayed. Some generated expressions like "blue and white stripped shirt" (row 2, column 1), "the clock with a lot of a circle" (row 5, column 4) demonstrate that the proposed attribute-guided attention model can generate expressions with more details.

2) Human Evaluation: One question remains open is that whether the automatic evaluation score can truly reflect the unambiguity of an expression. For instance, the expression "The man without glasses" has a high automatic evaluation score for a man with glasses, but would surely give the wrong information to the listener. Currently, the most reliable way to meet the core requirement of referring expression is to let the human do the evaluation job. In this paper, we randomly select 100 generated expressions in each split of all datasets.

Next, we ask two users to select the region box of the object, if both users click on the ground truth box then we determine the expression as correct or without ambiguity. Table II shows the accuracy by human evaluation. The results demonstrate the general tendency of the added modules, which are effective in reducing the ambiguity.

D. Results: Referring Expression Comprehension

As above mentioned, referring expression comprehension can be accomplished by either the speaker based approach or the listener (common space embedding) module. The previous approach of [55] ensemble outputs of the two modules in the testing stage, obtaining the state-of-the-art result. In this paper, the roles of speaker and listener are not our primary concern, so we test the comprehension performance based only on the listener module. We do not use the speaker module since it is empirically a little worse than the listener module and slower in practice.
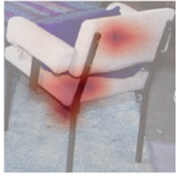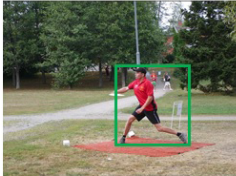
Fig. 8. Some examples of the learned attributes, and their guide attention learned on object regions and words.

1) Comprehension Results on the Ground Truth Boxes: To exclude the influence of the object detectors in the first place of the comprehension system, we first test performance directly on the ground truth boxes. This would indicate the upper limit of the comprehension module. In Table III we show the ablated combinations of attributes, visual difference, and attribute-guided attention. Results in Lines 10-14 show the contribution of each additional module or feature. Models added with attributes and the guide attention perform generally better. Compared with Line 9 of the ensemble model of speaker and listener, our approach based solely on listener are favorably competitive to the state-of-the-art results in most cases. Another interesting observation is still the difference between human and non-human which has been found in generation. Attributes and the guide attention generally contribute more when associated with referred objects of people. The most evident result is in Line 12 of TestA in RefCOCO+, with a 4 point improvement compared with those in Lines 10-11.

Figure 8 shows some attribute-guided attention learned on the referred objects and the words in expressions. We highlight the visual attention in the region box of the referred object. For the word attention, we also use different colors to roughly visualize the attention weights.

2) Automatic Comprehension Results on Detected Boxes: To make the comprehension process fully automatic, we need to replace the ground truth boxes with the detected objects. Here we follow the principles applied in [55], wherein SSD [51] is trained from MSCOCO images that do not exist in RefCOCO, RefCOCO+ and RefCOCOg. The metric is borrowed from the one commonly used in object localization,

wherein the localized region should have at least 0.5 IOU with the ground truth box. Table IV shows the same configurations used in Table III. The general trend maintains the same with that using ground truth boxes. The result also proves that our proposed model is more robust than previous methods towards the detected bounding boxes, as our models perform better than they are in the setting of ground truth object regions, and achieve the state-of-the-art results.

Figure 7 shows some qualitative automatic comprehension results from three datasets. The top three rows are correct comprehended results and the bottom two rows are wrongly comprehended ones. The green, blue and red bounding boxes stand for ground truth, correct comprehensions, and incorrect ones respectively. The reason for failure examples can be split into two types: The first type is due to the failure of the common space semantic embedding. The two failure examples from TestA of RefCOCO fall into this type. The second type is due to the failure of the detector, which cannot detect or accurately detect the target object. The last example from TestB of RefCOCO+ and the fourth example from RefCOCOg fall into this type.

E. Analysis on Attributes

Figure 9 shows the frequency of top 50 attributes collected from the datasets of RefCOCO and RefCOCO+. The attribute learning model is trained on the training sets, and we evaluate its attribute classification accuracy on the validation sets. To simplify the evaluation, we use 0.5 as the threshold value for the correct classification. Table VIII shows the precision,
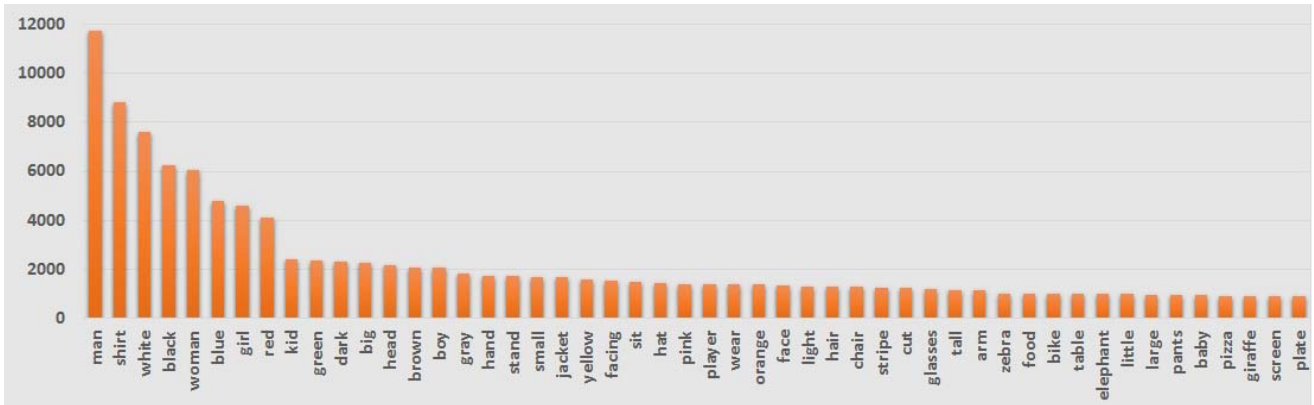
Fig. 9.   Top 50 frequent attributes collected from the datasets of RefCOCO, and RefCOCO+.

TABLE VIII

EVALUATION ON THE ATTRIBUTE LEARNING MODEL
ON THE VALIDATION SET OF THE THREE DATASETS

	precision	recall	F1
RefCOCO	52.47	25.19	33.53
RefCOCO+	50.22	15.40	27.18
RefCOCOg	53.93	26.01	33.33

recall and F1 measure on the three datasets. The major difficulty that decreases the accuracy of the attribute learning model is the quality of the labels. Since human expressions are subjective and various, thus the collected attribute words are inconsistent as labels.

V. DISCUSSION AND FUTURE WORK

Attributes play a key role in describing a unique object. In [7], the authors conclude with 7 major kinds of attributes: category name, color, size, absolute location, relative location, relative object, and generic attribute. For attributes of location, they are not collected in our attribute learning model since they have been already encoded as a part of visual feature. A further step can consider including location attributes in the attribute learning model as well. We believe this would predict locations more accurately especially for some expressions like "apple at 4 o'clock". Another attributes that are not considered in this paper are the ones from relative objects. It is not uncommon in both everyday life and the datasets to use relative objects to describe the target object. Moreover, the attention module can also be used to learn the attention weights to select the relative objects. An alternative approach can be found in [53]. In early works of referring expression generation, rule-based models are adopted to select the off-line ground truth attributes to build the expression. We believe it is worth retrospecting the traditional ways by combining modern deep learning methods. For referring expression comprehension, common space embedding approaches prove to be successful. But expressions with more complex logic like "man without glasses" would be very difficult to overcome with embedding models, since such expressions are very likely to have similar embeddings even with opposite meanings. There are papers [58] using deep module networks to model the structure of the expressions,

while the more complex logic in sentence still needs to be more explicitly modeled. We believe traditional natural language parsing methods are also good tools to use here.

VI. CONCLUSION

In this paper we present an attribute-guided attention model addressing the two tasks of referring expression generation and comprehension. The attribute-guided attention model is motivated by the key role of attributes in referring expression. Attributes are corresponded to both the visual and textual space of the referred object. An attribute-guided attention module is learned to attend to corresponding visual parts of the object and embedded words in expressions. Experimental results on three standard datasets of RefCOCO, RefCOCO+, and RefCOCOg demonstrate the effectiveness of the proposed model. The proposed model has a significant improvement on baseline methods and is favorably competitive to the state-of-the-art approach. Ablation study and analysis clearly show the contribution and shortcomings of each part, providing useful inspirations to researchers within this field.

REFERENCES

[1] T. Winograd, "Understanding natural language," *Cogn. Psychol.*, vol. 3, no. 1, pp. 1–191, 1972.

[2] E. Krahmer and K. van Deemter, "Computational generation of referring expressions: A survey," *Comput. Linguistics*, vol. 38, no. 1, pp. 173–218, Mar. 2012.

[3] M. Mitchell, K. V. Deemter, and E. Reiter, "Generating expressions that refer to visible objects," in *Proc. HLT-NAACL*, 2013, pp. 1–11.

[4] N. FitzGerald, Y. Artzi, and L. S. Zettlemoyer, "Learning distributions over logical forms for referring expression generation," in *Proc. EMNLP*, 2013, pp. 1–12.

[5] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy, "Generation and comprehension of unambiguous object descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 11–20.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, "ReferItGame: Referring to objects in photographs of natural scenes," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 787–798.

[8] J. Liu, L. Wang, and M. Yang, "Referring expression generation and comprehension via attributes," in *Proc. ICCV*, 2017.

[9] D. Golland, P. Liang, and D. Klein, "A game-theoretic approach to generating spatial descriptions," in *Proc. EMNLP*, 2010, pp. 410–419.

[10] R. Fang, M. Doering, and J. Y. Chai, "Embodied collaborative referring expression generation in situated human-robot interaction," in *Proc. 10th Annu. ACM/IEEE Int. Conf. Hum.-Robot Interact.*, Mar. 2015, pp. 271–278.

[11] D. Whitney, M. Eldon, J. Oberlin, and S. Tellex, "Interpreting multimodal referring expressions in real time," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 3331–3338.

[12] M. Grubinger, P. Clough, H. Müller, and T. Deselaers, "The IAPR TC-12 benchmark: A new evaluation resource for visual information systems," in *Proc. Int. Workshop OntoImage*, 2006, pp. 13–55.

[13] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg, "Modeling context in referring expressions," in *Proc. ECCV*, 2016, pp. 69–85.

[14] T. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.

[15] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. CVPR*, Jun. 2015, pp. 2625–2634.

[16] 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. (CVPR)*, Jun. 2015, pp. 3156–3164.

[17] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars, "Guiding the long-short term memory model for image caption generation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2407–2415.

[18] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. ICML*, 2015, pp. 1–10.

[19] Z. Yang, Y. Yuan, Y. Wu, R. Salakhutdinov, and W. W. Cohen, "Encode, review, and decode: Reviewer module for caption generation," in *Proc. NIPS*, 2016, pp. 1–9.

[20] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 21–29.

[21] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Proc. NIPS*, 2016, pp. 289–297.

[22] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 375–383.

[23] S. Reed, Z. Akata, H. Lee, and B. Schiele, "Learning deep representations of fine-grained visual descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 49–58.

[24] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4651–4659.

[25] Q. Wu, C. Shen, L. Liu, A. Dick, and A. V. D. Hengel, "What value do explicit high level concepts have in vision to language problems?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 203–212.

[26] J. Johnson, A. Karpathy, and L. Fei-Fei, "DenseCap: Fully convolutional localization networks for dense captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4565–4574.

[27] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proc. CVPR*, Jun. 2015, pp. 3128–3137.

[28] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele, "Grounding of textual phrases in images by reconstruction," in *Proc. ECCV*, 2016, pp. 817–834.

[29] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, and T. Mikolov, "DeVISE: A deep visual-semantic embedding model," in *Proc. NIPS*, 2013, pp. 2121–2129.

[30] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Visual-semantic embeddings with multimodal neural language models," in *Proc. TACL*, 2015, pp. 1–13.

[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. ICLR*, 2013, pp. 1–12.

[32] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun, "Order-embeddings of images and language," in *Proc. ICLR*, 2016, pp. 1–13.

[33] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Proc. CVPR*, Jun. 2016, pp. 5005–5013.

[34] F. Yan and K. Mikolajczyk, "Deep correlation for matching images and text," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3441–3450.

[35] B. Klein, G. Lev, G. Sadeh, and L. Wolf, "Associating neural word embeddings with deep image representations using Fisher vectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4437–4446.

[36] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[37] G. Lev, G. Sadeh, B. Klein, and L. Wolf, "RNN Fisher vectors for action recognition and image annotation," in *Proc. ECCV*, 2016, pp. 833–850.

[38] B. Plummer, L. Wang, C. Cervantes, J. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k Entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," in *Proc. ICCV*, Dec. 2015, pp. 2641–2649.

[39] M. Wang, M. Azab, N. Kojima, R. Mihalcea, and J. Deng, "Structured matching for phrase localization," in *Proc. ECCV*, 2016, pp. 696–711.

[40] L. Ma, Z. Lu, L. Shang, and H. Li, "Multimodal convolutional neural networks for matching image and sentence," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2623–2631.

[41] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[42] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[43] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. NIPS*, 2015, pp. 91–99.

[44] S. Antol *et al.*, "VQA: Visual question answering," in *Proc. ICCV*, Dec. 2015, pp. 2425–2433.

[45] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A neural-based approach to answering questions about images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1–9.

[46] M. Ren, R. Kiros, and R. Zemel, "Exploring models and data for image question answering," in *Proc. NIPS*, 2015, pp. 2953–2961.

[47] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2015, pp. 1–5.

[48] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," in *Proc. ICLR*, 2014, pp. 1–16.

[49] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[50] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[51] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. ECCV*, 2016, pp. 21–37.

[52] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell, "Natural language object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4555–4564.

[53] V. K. Nagaraja, V. I. Morariu, and L. S. Davis, "Modeling context between objects for referring expression understanding," in *Proc. ECCV*, 2016, pp. 792–807.

[54] R. Luo and G. Shakhnarovich, "Comprehension-guided referring expressions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7102–7111.

[55] L. Yu, H. Tan, M. Bansal, and T. L. Berg, "A joint Speaker-Listener-Reinforcer model for referring expressions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7282–7290.

[56] L. Yu *et al.*, "MAttNet: Modular attention network for referring expression comprehension," in *Proc. CVPR*, Jun. 2018, pp. 1307–1315.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[58] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko, "Modeling relationships in referential expressions with compositional modular networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1115–1124.

Jingyu Liu received the B.Eng. degree from Hunan University in 2011, the M.Eng. degree from Beijing Jiaotong University in 2014, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences (CAS) in 2018.

He is currently a Postdoctoral Researcher with the School of Electronics Engineering and Computer Science, Peking University. His research interests include object detection, vision and language understanding, and medical image analysis.

Wei Wang received the B.E. degree from the Department of Automation, Wuhan University, in 2005, and the Ph.D. degree from the University of Chinese Academy of Sciences in 2011. He is currently an Associate Professor with the National Lab of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA). He has published more than 30 articles in refereed international journals and conferences, such as TPAMI, TIP, CVPR, ICCV, and NIPS. His research interests focus on computer vision and machine learning, particularly on the computational modeling of visual attention and memory, vision, and language understanding.

Ming-Hsuan Yang (Fellow, IEEE) received the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign. He is currently a Professor of computer science and engineering with the University of California at Merced and an Adjunct Professor with Yonsei University. He received the NSF CAREER Award in 2012 and the Google Faculty Award in 2009. He has served as an Associate Editor for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 2007 to 2011. He is also an Associate Editor of the *International Journal of Computer Vision*, *Image and Vision Computing*, and the *Journal of Artificial Intelligence Research*.

Liang Wang (Fellow, IEEE) received the B.Eng. and M.Eng. degrees from Anhui University in 1997 and 2000, respectively, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences (CAS) in 2004.

From 2004 to 2010, he worked as a Research Assistant with Imperial College London, U.K., and Monash University, Australia, a Research Fellow with the University of Melbourne, Australia, and a Lecturer with the University of Bath, U.K., respectively. He is currently a Full Professor of Hundred Talents Program with the National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China. His major research interests include machine learning, pattern recognition, and computer vision. He has widely published at highly-ranked international journals, such as the IEEE TPAMI and the IEEE TIP, and leading international conferences, such as CVPR, ICCV, and AAAI. He has obtained several honors and awards, such as the Special Prize of the Presidential Scholarship of Chinese Academy of Sciences. He is a Fellow of IAPR and CIE, as well as a member of BMVA and ACM. He is also an Associate Editor of the IEEE TPAMI, the IEEE TIP, and Pattern Recognition.