Referring Expression Generation and Comprehension via Attributes

Jingyu Liu[1,3] Liang Wang[1,2,3] Ming-Hsuan Yang[4]
[1]Center for Research on Intelligent Perception and Computing (CRIPAC),
National Laboratory of Pattern Recognition (NLPR)
[2]Center for Excellence in Brain Science and Intelligence Technology (CEBSIT),
Institute of Automation, Chinese Academy of Sciences (CASIA)
[3]University of Chinese Academy of Sciences (UCAS)
[4]University of California, Merced
{jingyu.liu, wangliang}@nlpr.ia.ac.cn mhyang@ucmerced.edu

Abstract

Referring expression is a kind of language expression that used for referring to particular objects. To make the expression without ambiguation, people often use attributes to describe the particular object. In this paper, we explore the role of attributes by incorporating them into both referring expression generation and comprehension. We first train an attribute learning model from visual objects and their paired descriptions. Then in the generation task, we take the learned attributes as the input into the generation model, thus the expressions are generated driven by both attributes and the previous words. For comprehension, we embed the learned attributes with visual features and semantics into the common space model, then the target object is retrieved based on its ranking distance in the common space. Experimental results on the three standard datasets, RefCOCO, RefCOCO+, and RefCOCOg show significant improvements over the baseline model, demonstrating that our method is effective for both tasks.

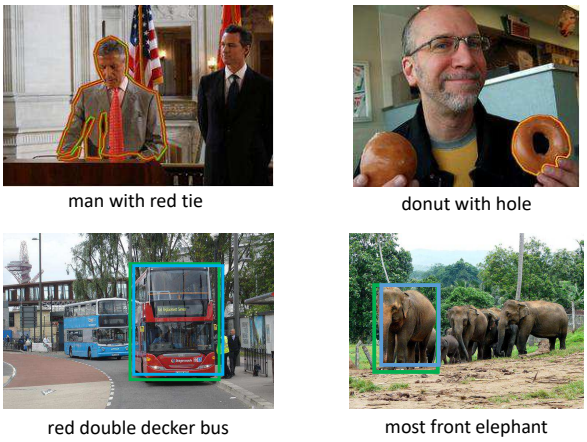

Figure 1. Some examples of our results. The first row are two examples of referring expression generation. The target object is shown in contour. The second row are two examples of referring expression comprehension. The green boxes are ground truth and the blue boxes are retrieved boxes.

1. Introduction

Referring expression [27] is a particular kind of human expressions that focuses on effectively describing the unique object in some environments. The interactive process of referring expression generation and comprehension exists in our everyday life. Given the target object and its distractions, one often uses the target's unique attributes to describe it. For instance, "a girl in a red skirt" contains the attributes "girl", "red" and "skirt" to differentiate the girl from other people in a particular scenario. The more accurate and rich the attributes are, the easier for the listener to comprehend which object is referred. This is where the difference lies between referring expression generation and natural language generation. On the other side, the task of comprehension requires the listener to prove his comprehension by pointing to the target object's location, which is based on the quality of the listener's interpretation of the expression. Figure 1 illustrates the two tasks. The top row and bottom row are examples of generation and comprehension respectively. In this paper, we mainly focus on the usage of visual attributes in both referring expression generation and comprehension, and the approach to effectively incorporate them into both tasks.

Modern approaches of referring expression [7, 16, 31] rely heavily on the encoding of both the image and the language. For referring expression generation, discriminative features from different modalities like appearance and loca-

tion are encoded by a neural model like convolutional neural network (CNN). Then a long short term memory (LSTM) model is adopted to decode the visual encoding into a referring expression. For referring expression comprehension, there are two approaches generally used. The first one is based on the generation model, wherein the probability $P(r|o)$ of the referring expression r given the object o can be obtained. By Bayes's rule, the most likely object can be obtained with the maximum posterior probability $P(o|r)$. Another approach is to embed the target object and its expression into a common space. Then the task can be addressed in a retrieval manner, wherein the target object is selected with the minimum distance to the expression in the common space. To make the comprehension process an automatic system, object detection systems can be utilized to obtain a group of candidate objects in the first place.

Our approach explores the role of attributes in referring expression. A target object is unique because it has unique attributes or a unique combination of attributes. For instance, the expressions of "A man" and "A woman" are unambiguous when there are only a man and a woman in a scenario, but are ambiguous when there are two men and a woman. To address the problem, an expression with more attributes like "A man with a hat" has to be used. In this paper, we address the tasks of referring expression generation and comprehension in two separate models. In generation, we extend the traditional CNN-LSTM model to take the learned attributes as the extra input to the LSTM model, so that the generated expression bears more accurate attributes correlated with the input attributes. For the comprehension task, we frame the problem in the retrieval approach, wherein the attributes and the expressions are embedded into a common space. The target object is retrieved based on its ranking distance to the queried expression. To model difference between objects, the hinge loss based MMI [16] is used in both models, wherein we dynamically alter the margin to let the model be aware of the categories of the distracted objects. Finally, we discuss the effective way to construct the attribute learning model for referring expression. Though there have been some works [30, 28] using semantic attributes in image caption and other tasks, we are the first to embed it into both models in referring expression. We also analyze on the successful and failure cases of our model in both tasks, and point out the correctness and defects of modern attribute learning models.

Figure 2 illustrates our framework, which is composed of the attribute learning model, the expression generation model, and the expression comprehension model. The attribute learning model outputs the attributes of both the target object (green solid box) and its distraction (green dashed box). Then the attributes are embedded into both the generation and comprehension model. To focus on the attributes in the figure, we omit the display of other visual features. Generative loss and hinge loss are computed on the two models respectively.

2. Related Work

With the development of powerful neural models for vision and language, the intersection of vision and language has witnessed the emergence of more and more tasks. From early applications like image caption [2, 24, 9, 29, 19] and image/text retrieval [4, 15], to image question answering [1] and text based grounding object localization [21].

Image Caption: Modern approaches of image caption are based on the CNN-RNN architecture [5, 2, 24]. The CNN feature extracted from the image is taken as input to the LSTM network, wherein both the visual signal and previous words guide the generation of next words. Attention models first proposed in natural language processing have also drawn inspiration on image caption. The attention mechanism shifts attention either at spatial level [29] or semantic level [30]. A more specific variation of image caption is to exploit the caption at a region(object) level, *e.g.*, structured alignment of words/phrases in sentences and regions in images [11], and provide a dense group of region captions [10]. These tasks do not focus on the unambiguity of the caption, so that differ from referring expression generation/comprehension.

Image/Text Retrieval: Different from image caption model that generates the sentence, image/text retrieval addresses the problem in a data-driven approach. Multi-modal embedding has been studied a lot recently [26, 8]. Traditional approaches adopt CNN and CNN/LSTM to encode the image and texts/sentences in their feature space, then neural models like MLP embed them into a common space. The retrieval can be formulated as a classification or a ranking based framework. The method in [8] includes a multimodal context-modulated attention scheme so that it can selectively attend to pairwise instances of image and sentence, and then dynamically aggregate measured similarity to obtain a global matching score for image and text.

Referring Expression Generation/Comprehension: Referring expression generation and comprehension are two complementary tasks which are always jointly addressed within the field. The generation task requires the system to generate an unambiguous expression that describes a particular object. Both visual features and location features are commonly used as the representation of target objects. To focus more on the uniqueness of the target object, recent works of [31, 18] also model the contexts of the target object. The context can either be used as the target's associated attributes or its contrast. Early works use rule-based model [17, 3] to generate the expression, while recent works rely on the LSTM model [6]. To model the difference of objects Max-margin Maximum Mutual Information(MMI)

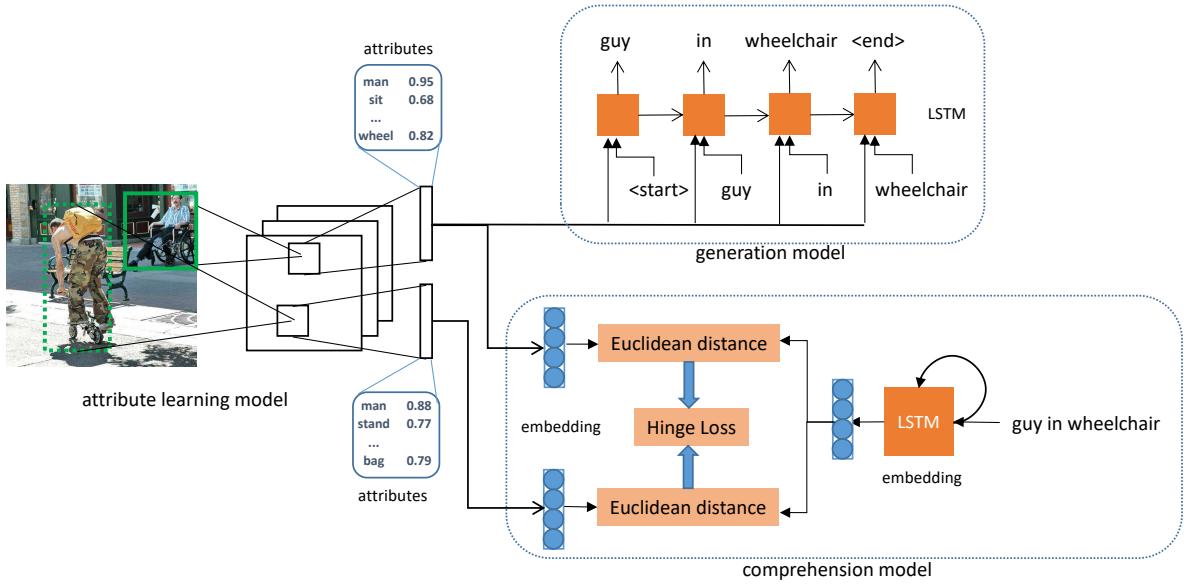

Figure 2. Illustration of the attribute embedded model. The framework is composed of the attribute learning model, the expression generation model and the expression comprehension model. The attribute learning model outputs the attributes of the target object (solid green box) and its distraction (dashed green box). The attributes are embedded into the generation model and the comprehension model. The generative loss and hinge loss are computed on the two models respectively.

is adopted in [16]. The generated expressions can be reranked by an offline comprehension model in the postprocess. For the comprehension task, early methods of [7, 16] base on the trained generation model, selecting the object with the maximum posterior probability, where more recent approach [21] applies embedding model to retrieve the target object.

3. Model

We address referring expression generation and comprehension in two separate models. In Section 3.1, we review the basic framework of the generation model, and the attribute embedded framework. In Section 3.2, we introduce the common space embedding model with attribute embedded. In Section 3.3, we discuss how to effectively train an attribute learning model.

3.1. Attribute Embedded Generation Model

The input of the generation model is an image I and a target object o, and the output is the referring expression r. The generation model is trained to maximize the likelihood of the correct expression by using the following formulation:

$$\theta^* = \arg\max_{\theta} \sum_i \log p(r_i|I_i, o_i) \tag{1}$$

where θ are the parameters of the model. In this paper, we use the CNN-LSTM framework commonly used in previous works [7, 16] as our generation model. Human often use features from different modalities, *e.g.*, appearance and location/size descriptions to refer to the target. To encode the visual feature, activations from VGG-fc7 are extracted from the object region as in [16]. For the location/size feature l_i, a 5-dimension vector $\left[\frac{x_l}{W}, \frac{y_t}{H}, \frac{x_r}{W}, \frac{y_b}{H}, \frac{w \cdot h}{W \cdot H}\right]$ is often used to encode the information, where x_l, y_t, x_r, y_b are coordinates of the object region and w, h, W, H are widths and heights of the region and the image. Other features, *e.g.*, global features and comparison features [16, 31] are also used to improve the performance. The final visual representation v_i of the target object is a concatenation of above features followed by a fully-connected layer of them.

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

On top of the above visual features, we define the attributes of a target as $a_i = [a_{i1}, a_{i2}, \ldots, a_{in}]$. a_i is a vector of dimension n. n is the number of attributes and each dimension of a_i denotes the likelihood of a particular kind of attribute. The attributes are composed of various kinds, *e.g.*, the name of an entity, color, action, *etc*. In Section 3.3 we will discuss how to train an attribute learning model to obtain a_i in detail. To effectively use the attributes a_i, we extend the LSTM module to embed a_i as the input:

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

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

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

$$\widetilde{c_t} = \tanh\left(W_{cx}x_t + W_{ch}h_{t-1} + W_{ca}a_i + W_{cv}v_i\right) \quad (6)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t} \quad (7)$$

$$h_t = o_t \odot \tanh\left(c_t\right) \quad (8)$$

where x_t is the input token word at each time step and the various W matrices are the training parameters. The attribute a_i influences on the input gate i_t, forget gate f_T, output gate o_t and the state c_t. The whole model can then be trained by minimizing the cross entropy loss, or equivalently the negative log-likelihood:

$$\begin{aligned} L_1(\theta) = &- \sum_i \log P(r_i|o_i;\theta) \\ &- \sum_i \sum_{t=1}^{T} \log P(r_{i,t}|r_{i,<t}, o_i;\theta) \end{aligned} \quad (9)$$

The property of referring expression is that no two objects in the same image should be described by the same sentence. Following the paradigm in [31], we apply the triplet hinge loss to encourage the target object to have a larger probability than other objects towards its descriptions. Considering the fact that objects from different categories normally have larger variance in appearance than those from the same category, we dynamically assign different margins during training according to the sampled objects' categories:

$$\begin{aligned} L_2(\theta) = &- \sum_i \max(0, M_1 \mathbb{1}_{C(o_i)=C(o_k)} + M_2 \mathbb{1}_{C(o_i)\neq C(o_k)} \\ &+ \log P(r_i|o_k) - \log P(r_i|o_i)) \end{aligned} \quad (10)$$

where M_1 and M_2 are margins and $C(o_i)$ indicates the category of object o_i.

3.2. Attribute Embedded Comprehension Model

Referring expression comprehension requires the listener to interpret the semantic meaning in the sentence. The listener should prove its understanding by pointing to the correct object. The input to the problem is an image I, a set of candidate regions (objects) $\{o\}$ and a referring expression r. All previous methods address the problem in a ranking based retrieval approach, either using the trained generation model or learning a common space embedding model of the region and the expression. In this paper, we use the latter one since it performs better in practice.

The common space embedding model requires both the visual object and the referring expression have an effective representation in the first place. Then neural networks are commonly used to project different-modality features into the same space so that metric can be calculated. The training encourages paired object and expression to be close in the common space, and unpaired ones to be apart. The embedding of language has been studied a lot in recent years [4, 15, 25, 26, 23]. CNN and LSTM are commonly used to encode the words/phrases or sentences, either at a character level or a word level. In this paper, we use a unidirectional LSTM to encode it, and the hidden state h of the last time step is extracted as its final representation. For the encoding of the visual object o_i, we follow the same setting in the generation model, *i.e*. the VGG-fc7 and the attributes entry vector a_i. They are scaled to the same scale, concatenated and followed by a fully-connected layer to the final visual feature v_i.

The next step is to embed features from both modalities to the common space, layers of MLPs are adopted to project them to have the same dimension sizes. After that, either similarity or distance functions can be used to compute the loss. In this paper, we use the Euclidean distance as the measurement.

$$d(f(v), g(h)) = \|f(v) - g(h)\|_2 \quad (11)$$

where $f(v)$ and $g(h)$ are the encoded visual and semantic features. So there are two commonly used approaches to frame the problem, a binary classification one which decides r and o is a pair or not, and a multi-classification one that assigns each r/o to a o/r from a candidate group. While referring expression focuses more on the difference between the target object and its distractions, we follow the paradigm used in the generation model. For each pair of r_i and o_i, we sample a negative pair of r_i and o_j, and another negative pair of r_k and o_i, then formulate the function of two triplet hinge losses:

$$\begin{aligned} L_2(\theta) = &- \sum_i [\lambda_1 \max(0, M_1 \mathbb{1}_{C(o_i)=C(o_j)} + M_2 \mathbb{1}_{C(o_i)\neq C(o_j)} \\ &- d(r_i, o_j) + d(r_i, o_i)) \\ &+ \lambda_2 \max(0, M_1 \mathbb{1}_{C(r_i)=C(r_k)} + M_2 \mathbb{1}_{C(r_i)\neq C(r_k)} \\ &- d(r_k, o_i) + d(r_i, o_i))] \end{aligned} \quad (12)$$

where λ_1 and λ_2 are the weights of the two losses. Following the paradigm for generation, we also dynamically alter the margin according to whether two regions or expressions are from the same category.

3.3. Attribute Learning

The first step of attribute learning is to construct an attribute set, which has been studied in other tasks like image caption and visual question answering [30, 28]. Like [28], we define attributes in various forms, *e.g*. name entities, properties (color, material, *etc*.) and motions. We first use NLP toolbox to exclude stopping words like "a", "the", "of", *etc*., then we exclude the words with low frequency. To make the attribute set more accurate and concise, we treat synonyms, *e.g*. "bike" and "bicycle" as the same attribute.

After the construction of the attribute set, we can extract attributes from the referring expression of each object region. Unlike attributes describing different parts of the image in the image caption model, attributes in referring expression have already been bounded in the bounding box. Therefore we do not need to frame it as a Multiple Instance Learning problem that used in image caption. Instead, we directly formulate it as a multi-label classification problem. Multi-label classification is a traditional problem and has been studied a lot. We test the performance of hinge loss, the margin ranking loss and the binary sigmoid cross-entropy loss. Unexpectedly, we find that the simple binary sigmoid cross-entropy loss works best in practice. We think it is due to the inconsistent annotation in the dataset. Therefore the cost function to minimize is:

$$E = \frac{1}{n}\sum_{i}^{n}\sum_{j}^{m}[y_{ij}\log(p_{ij}) + (1-y_{ij})\log(1-p_{ij})] \tag{13}$$

where m is the size of the attribute set. $y_i = [y_{i1}, y_{i2}, \ldots, y_{im}]$ is the label vector of the i^{th} image. $y_{ij} = 1\ or\ 0$ denotes whether the object has the attribute or not. $p_i = [p_{i1}, p_{i2}, \ldots, p_{im}]$ denotes the probability vector.

4. Experiments

Our experiments are conducted on the three standard datasets, RefCOCO, RefCOCO+ and RefCOCOg within the field of referring expression.

RefCOCO(UNC RefExp) [31] 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 [12]. Since people are much more frequent than other objects in the dataset, the split is person vs. objects: images containing multiple people are in Test A and images containing multiple objects from other categories are in Test B.

RefCOCO+ [31] has 141,564 expressions for 49,856 objects in 19,992 images from COCO. This dataset is also collected using ReferitGame, but this time players are disallowed to use location words to describe the object. Therefore this dataset focuses more on the purely appearance based description. The split in RefCOCO+ follows the same rule used in RefCOCO.

RefCOCOg(Google RefExp) [16] consists of 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 sentences. The split of this dataset is on a per-object basis. Objects are randomly partitioned into training and validation splits.

Table 2. Human evaluation results on RefCOCO and RefCOCO+.

	RefCOCO		RefCOCO+	
	TestA	TestB	TestA	TestB
baseline [16]	66%	65%	34%	34%
attr	73%	69%	39%	38%
attr+MMI	76%	72%	**43%**	38%
attr+visdif	77%	74%	39%	**44%**
attr+MMI+visdif	**78%**	**83%**	41%	43%

4.1. Parameter Settings

For attribute learning, the size of the attribute set is 600. The training model is fine-tuned on the VGG-19 network [22] pretrained on the Imagenet. The softmax loss layer is replaced with the binary sigmoid cross-entropy layer. Object regions are extracted from the training set of the three standard datasets. The regions' aspect ratios are kept and padded with zero values to resize to 256×256. Then 224×224 patches are randomly cropped as input to the network. The model is optimized using SGD in 10,000 iterations, with an initial learning rate of 0.0001, decreasing to 0.00001 after 5,000 iterations. The batch size is 32.

For referring expression generation and comprehension, the basic visual encoding features are VGG-fc7 and the location feature mentioned before. Global or context features are not used in our experiments. Additionally, we test our model with the visdif [31] since it complements our method well. Adam [13] is adopted as the optimization tool. The initial learning rate is 0.0005, halved every 8,000 iterations with a batch size of 32. The visual feature embedding size and the hidden state size of LSTM are both 512. For the task of generation, both visual features and attributes are taken as input at each time step of LSTM. For the task of comprehension, two layers of MLPs are followed by the initial encoding visual and language features. We set $M_1 = 0.2$ and $M_2 = 0.4$ in Eq. 10, and $\lambda_1 = 1$ and $\lambda_2 = 1$ in Eq. 12.

4.2. Generation

For the expression generation task, we use beam search to select our sampled expressions. We evaluate our generated expressions using automatic caption generation metrics, including BLEU1, BLEU2, ROUGE and METEOR. Due to space limitation we only display ROUGE and METEOR in Table 1. Complete results are shown in our supplementary material. MMI is the max mutual information method [16] re-implemented in [31]. "Visdif" and "tie" are two techniques used in [31], and are orthogonal to our methods. We denote our methods as "attr" in the table. The results show that our method can consistently improve the performance over the baseline method in all datasets. In most experiment settings, the attribute embedded generation model has better performances than

Figure 3. Some referring expression generation results on the three datasets. The generated expressions from the top to the bottom row are from the methods of baseline, attr and attr+visdif respectively.

Table 1. Referring expression generation results evaluated by automated metrics on RefCOCO, RefCOCO+ and RefCOCOg.

	RefCOCO				RefCOCO+				RefCOCOg	
	TestA		TestB		TestA		TestB		Val	
	Rouge	Meteor	Rouge	Meteor	Rouge	Meteor	Rouge	Meteor	Rouge	Meteor
baseline [16]	0.413	0.173	0.499	0.228	0.356	0.140	0.322	0.135	0.363	0.149
MMI [16]	0.418	0.175	0.497	0.228	0.346	0.136	0.320	0.133	0.354	0.144
visdif+MMI [31]	0.441	0.185	0.531	0.247	0.360	0.142	0.325	0.135	0.370	0.151
visdif+tie [31]	0.446	0.189	0.533	0.249	0.372	0.150	0.328	0.143	-	-
attr	0.472	0.208	0.532	0.247	0.362	0.150	0.345	0.149	**0.389**	**0.163**
attr+visdif	**0.494**	**0.222**	**0.546**	**0.258**	**0.374**	**0.155**	**0.355**	**0.155**	0.378	0.160

previous methods. We also test our model combined with visdif [31], which models the visual difference among objects of the same category. We find it especially effective in modeling the relative locations among objects and thus complement our method. Except for the results in RefCOCOg, where expressions have more descriptions of attributes and no location words, the results are further improved in other datasets. We also perform human evaluation on attr, attr+MMI, attr+visdif and attr+MMI+visdif of each split in RefCOCO and RefCOCO+. We first randomly select 100 target objects from each split. Then we ask two human "listeners" to click on the box after showing them the generated expression. The candidate boxes are all the annotated objects in MSCOCO. If both listeners click on the correct box of the target, then the result is counted as a correct expression. Table 2 shows the comparison results, where attributes contribute consistently to the unambiguity of the generated expression.

Figure 3 shows some expression generation results on the three datasets. From the top to the bottom row are the results based on the baseline, attr and attr+visdif methods respectively. The results show that attribute embedded model can detect more accurate properties of the objects, while the baseline model has difficulty in distinguishing the sim-

Table 3. Referring expression comprehension results. Top half shows results given ground truth bounding boxes for all objects in the image. Bottom half shows results using automatic object detectors to provide a candidate group of objects.

	RefCOCO		RefCOCO+		RefCOCOg
	Test A	Test B	Test A	Test B	Val
baseline [16]	63.15%	64.21%	48.73%	42.13%	55.16%
MMI [16]	71.72%	71.09%	58.42%	51.23%	62.14%
visdif+MMI [31]	73.98%	76.59%	59.17%	55.62%	64.02%
Neg Bag [18]	75.6%	78.0%	-	-	68.4
attr+MMI	78.12%	75.89%	60.76%	54.97%	67.43%
attr+MMI+visdif	**78.85%**	**78.07%**	**61.47%**	**57.22%**	**69.83%**
	RefCOCO(det)		RefCOCO+(det)		RefCOCOg(det)
	Test A	Test B	Test A	Test B	Val
baseline [16]	58.32%	48.48%	46.86%	34.04%	40.75%
MMI [16]	64.90%	54.51%	54.03%	42.81%	45.85%
visdif+MMI [31]	67.64%	55.16%	55.81%	43.43%	46.86%
Neg Bag [18]	58.6%	56.4%	-	-	39.5%
attr+MMI	70.55%	54.80%	56.38%	43.14%	50.02%
attr+MMI+visdif	**72.08%**	**57.29**	**57.97%**	**46.20%**	**52.35%**

Figure 4. Some results of referring expression comprehension. The top two rows are correct hits. The bottom row shows some failure examples.

ilar visual concepts, *e.g*. "tv" towards "screen", "bride" towards "woman", "bottle" towards "cup" *etc*. Another interesting observation is that the attr+visdif method sometimes ignores the visual attributes, which we think is due to the computation of visual difference would obliterate the common attributes among objects.

4.3. Comprehension

We evaluate our comprehension results on the datasets of RefCOCO, RefCOCO+ and RefCOCOg. In RefCOCO and RefCOCO+, we follow the split of people/non-people in TestA and TestB. For RefCOCOg, we evaluate the result on the validation set since the test set is not released. Additionally, there are two experimental settings for the task of comprehension. The first setting assumes the observer has already known what an object is, so the input region set $\{r\}$ consists of all the ground truth objects labeled in the COCO dataset. The model is required to select the target object from those ground truth objects. The second setting assumes the whole process as an automatic system. The

Table 4. Recall of some representative predicted attributes with $p(a) \geqslant 0.1$ in RefCOCO.

human		color		food		animal		pose		shape	
man	0.97	white	0.84	pizza	0.88	dog	0.74	run	0.43	round	0.37
woman	0.92	black	0.78	donut	0.92	cat	0.87	sit	0.58	square	0.25
boy	0.78	red	0.85	apple	0.88	horse	0.86	stand	0.49	hole	0.20
girl	0.88	green	0.82	cake	0.64	sheep	0.93	eat	0.34	stripe	0.69
baby	0.71	blue	0.87	broccoli	0.97	bird	0.76	jump	0.48	check	0.46

model needs to detect potential objects from the image in the first place, so the quality of object detectors will influence the final result. In this paper, we adopt Faster-RCNN [20] as our object detector. The metric follows the one used in object detection: a hit is counted if the model chooses a bounding box whose overlap is above 0.5 with the ground truth.

Table 3 shows the comprehension performance on the three datasets. The top half shows results given ground truth bounding boxes for all objects in the image. The bottom half shows results using Faster-RCNN to provide a candidate group of objects. We directly evaluate our model combined with MMI, which has been the standard approach adopted in comprehension [16, 31, 18]. The results show that the attribute embedded model has consistently better performance than the MMI [31]. The attributes contribute more performance improvements in the "people" split of Test A than in the "non-people" split of Test B. The reason is that people often show in different visual appearances of cloths, poses and motions, while other objects more often show in similar appearances. The most illustrative examples are categories in animals and food. For instance, if excluding location words, it often requires more subtle words to clearly describe an elephant from other elephants. Additionally, we also test our method combined with visdif. The results in the last row demonstrate it complements well to the attribute embedded model.

In Figure 4 we show some qualitative comprehension results of "attr" on the three datasets based on the first problem setting. The top two rows are correct hits and the bottom row shows some failure examples. The failure examples in the last row include three representative cases: First, objects with similar visual attributes shown in the first column. Second, expressions rely on complex location based words shown in the second column. Third, objects have discriminative but abstractive visual attributes, like "looking at phone" and "blank" in the third and forth column. In the following, we provide a more comprehensive analysis on attributes learning in referring expression.

4.4. Analysis on Attributes

To evaluate which attributes are correctly predicted in the attribute learning process, we compute the recall of attributes with the requirement $p(a) \geqslant 0.1$, where $p(a)$ is the predicted likelihood of an attribute. We manually select attributes from some representative categories, which can be grouped into several classes of human, color, food, animal, pose and shape. The experiment is conducted on RefCOCO and the recall is evaluated on the training set. Table 4 shows the result. As expected, entity attributes like human, food, animal are at high recall. Color is also easy to predict as it is visually discriminative. More abstractive attributes of pose and shape are much harder for the model to learn. We attribute the reason to mainly two aspects. First, they are much more implicit, *e.g.* the attribute "stand" can be both used to describe human and animals, thus confuse the classifier especially if pretrained on solid object entities like ImageNet. Second, they are much less frequent in expressions. Since people describe objects from different aspects, therefore making the annotation inconsistent.

5. Conclusion

In this paper we demonstrate the effectiveness of the visual attributes in referring expression generation and comprehension. We first train an attribute learning model from visual objects and their paired descriptions. Then the learned attributes are embedded into both the generation model and the comprehension model. Experimental results demonstrate that our model significantly improves the baseline method and is competitive to the state-of-the-art results. We also analyze the correctness and defects of attributes with deeper studies. We believe these studies would provide future directions to the researchers who want to continue along this approach.

Acknowledgement

The authors are grateful to Licheng Yu for helpful discussions. This work is supported by 1) the NSF CAREER Grant #1149783, gifts from Adobe and NVIDIA. 2) National Key Research and Development Program of China (2016YFB1001000), National Natural Science Foundation of China (61525306, 61633021, 61572504, 61420106015), Strategic Priority Research Program of the CAS (XDB02070001), and Beijing Natural Science Foundation (4162058). 3) grants from NVIDIA and the NVIDIA DGX-1 AI Supercomputer.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Zitnick, and D. Parikh. Vqa: Visual question answering. *In ICCV*, 2015.

[2] J. Donahue, L. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *In CVPR*, 2015.

[3] N. FitzGerald, Y. Artzi, and L. S. Zettlemoyer. Learning distributions over logical forms for referring expression generation. *In EMNLP*, 2013.

[4] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, and T. Mikolov. Devise: A deep visual semantic embedding model. *In NIPS*, 2013.

[5] A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. *In ICASSP*, 2013.

[6] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735C1780.

[7] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. *In CVPR*, 2016.

[8] Y. Huang, W. Wang, and L. Wang. Instance-aware image and sentence matching with selective multimodal lstm. *In CVPR*, 2017.

[9] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars. Guiding the long-short term memory model for image caption generation. *In ICCV*, 2015.

[10] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. *In CVPR*, 2016.

[11] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *In CVPR*, 2015.

[12] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referitgame: Referring to objects in photographs of natural scenes. *In EMNLP*, 2014.

[13] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint*, arXiv:1412.6980, 2014.

[14] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. Zitnick. Microsoft coco: Common objects in context. *In ECCV*, 2014.

[15] L. Ma, Z. Lu, L. Shang, and H. Li. Multimodal convolutional neural networks for matching image and sentence. *In ICCV*, 2016.

[16] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. *In CVPR*, 2016.

[17] M. Mitchell, K. V. Deemter, and E. Reiter. Generating expressions that refer to visible objects. *In HLT-NAACL*, 2013.

[18] V. K. Nagaraja, V. I. Morariu, and L. S. Davis. Modeling context between objects for referring expression understanding. *In ECCV*, 2016.

[19] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. *In CVPR*, 2016.

[20] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detec-tion with region proposal networks. *In NIPS*, 2015.

[21] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of textual phrases in images by reconstruction. *In ECCV*, 2016.

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, arXiv:1409.1556, 2014.

[23] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun. Order-embeddings of images and language. *In ICLR*, 2016.

[24] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *In CVPR*, 2015.

[25] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure preserving image-text embeddings. *In CVPR*, 2016.

[26] M. Wang, M. Azab, N. Kojima, R. Mihalcea, and J. Deng. Structured matching for phrase localization. *In ECCV*, 2016.

[27] T. Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1C191, 1972.

[28] Q. Wu, C. Shen, L. Liu, A. Dick, and A. Hengel. What value do explicit high level concepts have in vision to language problems? *In CVPR*, 2016.

[29] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *In ICML*, 2015.

[30] Q. You, H. Jin, and Z. Wang. Image captioning with semantic attention. *In CVPR*, 2016.

[31] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. *In ECCV*, 2016.