Modeling Context in Referring Expressions

Licheng Yu(✉), Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg

Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, USA
{licheng,poirson,alexyang,aberg,tlberg}@cs.unc.edu

Abstract. Humans refer to objects in their environments all the time, especially in dialogue with other people. We explore generating and comprehending natural language referring expressions for objects in images. In particular, we focus on incorporating better measures of visual context into referring expression models and find that visual comparison to other objects within an image helps improve performance significantly. We also develop methods to tie the language generation process together, so that we generate expressions for all objects of a particular category jointly. Evaluation on three recent datasets - RefCOCO, RefCOCO+, and RefCOCOg (Datasets and toolbox can be downloaded from https://github.com/lichengunc/refer), shows the advantages of our methods for both referring expression generation and comprehension.

Keywords: Language · Language and vision · Generation · Referring expression generation

1 Introduction

In this paper, we look at the dual-tasks of generating and comprehending natural language expressions referring to particular objects within an image. Referring to objects is a natural and common experience. For example, one often uses referring expressions in everyday speech to indicate a particular person or object to a co-observer, e.g., "the man in the red hat" or "the book on the table". Computational models to generate and comprehend such expressions would have applicability to human-computer interactions, especially for agents such as robots, interacting with humans in the physical world.

Successful models will have to connect both recognition of visual attributes of objects and effective natural language generation to compose useful expressions for dialogue. A broader version of this latter goal was considered in 1975 by Paul Grice who introduced maxims describing cooperative conversation between people [9]. These maxims, called the Gricean Maxims, describe a set of rational

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46475-6_5) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part II, LNCS 9906, pp. 69–85, 2016.
DOI: 10.1007/978-3-319-46475-6_5

principles for natural language dialogue interactions. The 4 maxims are: quality (try to be truthful), quantity (make your contribution as informative as you can, giving as much information as is needed but no more), relevance (be relevant and pertinent to the discussion), and manner (be as clear, brief, and orderly as possible, avoiding obscurity and ambiguity).

For the purpose of referring to objects in complex real world scenes these maxims suggest that a well formed expression should be informative, succinct, and unambiguous. The last point is especially necessary for referring to objects in the real world since we often find multiple objects of a particular category situated together in a scene. For example, consider the image in Fig. 1 which contains three giraffes. We should not refer to the target (outlined in green) as "the spotted giraffe" since all of the giraffes are spotted and this would create an ambiguous reference. More reasonably we should refer to the target as "the giraffe with lowered head" to differentiate this giraffe from the other two.

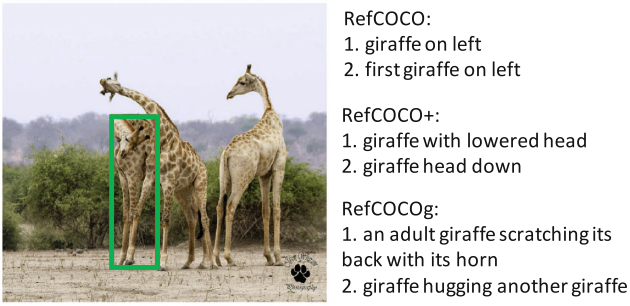

Fig. 1. Example referring expressions for the giraffe outlined in green from three referring expression datasets (described in Sect. 4). (Color figure online)

The task of referring expression generation (REG) has been studied since the 1970s [5,18,25,34], with most work focused on studying particular aspects of the problem in some relatively constrained datasets. Recent approaches have pushed this work toward more realistic scenarios. Kazemzadeh et al. [15] introduced the first large-scale dataset of referring expressions for objects in real-world natural images, collected in a two-player game. This dataset was originally collected on top of the 20,000 image ImageCleft dataset, but has recently been extended to images from the MSCOCO collection. We make use of the RefCOCO and RefCOCO+ datasets in our work along with another recently collected referring expression dataset, released by Google, denoted in our paper as RefCOCOg [21].

The most relevant work to ours is Mao et al. [21] which introduced the first deep learning approach to REG. In this model, the authors use a Convolutional Neural Network (CNN) [30] model pre-trained on ImageNet [28] to extract visual features from a bounding box around the target object and from the entire image. They use these features plus 5 features encoding the target object location and size as input to a Long Short-term Memory (LSTM) [8] network that generates

expressions. Additionally, they apply the same model to the inverse problem of referring expression comprehension where the input is a natural language expression and the goal is to localize the referred object in the image.

Similar to these recent methods, we also take a deep learning approach to referring expression generation and comprehension. However, while they use a generic model for object context – CNN features for the entire image containing the target object – we take a more focused approach to encode object comparisons. These object comparisons are critical for producing an unambiguous referring expression since one must consider visual characteristics of similar objects during generation in order to select the most distinct aspects for description. This mimics the process that a human would use to compose a good referring expression for an object, e.g. look at the object, look at other relevant objects, and generate an expression that could be used by a co-observer to unambiguously pick out the target object.

In addition, for the referring expression generation task, we introduce a method to tie the language generation process together for all depicted objects of the same type. This helps generate a good set of expressions such that the expressions differentiate between objects but are also complementary. For example, we never want to generate the exact same expression for two objects in an image. Alternatively, if we call one object "the red ball" then we may desire the expression for the other object to follow the same generation pattern, i.e., "the blue ball". Our experimental evaluations show that these visual and linguistic comparisons improve performance over previous state of the art.

In the rest of our paper, we first describe related work (Sect. 2). We then describe our improvements to models for referring expression generation and comprehension (Sect. 3), describe 3 referring expression datasets (Sect. 4), and perform experimental evaluations on several model variations (Sect. 5). Finally we present our conclusions (Sect. 6).

2 Related Work

Referring expressions are closely related to the more general problem of modeling the connection between images and descriptive language. In recent years, this has been studied in the **image captioning** task [4,10,19,26,31]. There, the aim is to condition the generation of language on the visual information from an image. The wide range of aspects of an image that could be described, and the variety of words that could be chosen for a particular description complicate studying image captioning. Our study of referring expressions is partially motivated by focusing on description for a specific, and more easily evaluated, communication goal. Although our task is somewhat different, we borrow machinery from state of the art caption generation [2,3,14,17,22,33,35] using LSTM to generate captions based on CNN features computed on an input image. Three recent approaches for referring expression generation [21] and comprehension [11,27] also take a deep learning approach. However, we add visual object comparisons and tie together language generation for multiple objects.

Referring expression generation has been studied for many years [18, 25,34] in linguistics and natural language processing. These works were limited by data collection and insufficient computer vision algorithms. Together Amazon Mechanical Turk and CNNs have somewhat mitigated these limitations, allowing us to revisit these ideas on large-scale datasets. We still use such work to motivate the architecture of our pipeline. For instance, Mitchell and Jordan et al. [13,25] show the importance of using attributes, Funakoshi et al. [6] show the importance of relative relations between objects in the same perceptual group, and Kelleher et al. [16] show the importance of spatial relationships. These provide motivation for our modeling choices: when considering a referring expression for an object, the model takes into account the relative spatial location of other objects of the same type and visual comparisons to objects in the same perceptual group.

The REG datasets of the past were sometimes limited to using computer generated images [32], or relatively small collections of natural objects [5,23,24]. Recently, a large-scale referring expression dataset was collected by Kazemzadeh et al. [15] featuring natural objects in the real world. Since then, another three REG datasets based on the object labels in MSCOCO have been collected [15, 21]. The availability of large-scale referring expression datasets allows us to train deep learning models. Additionally, our analysis of these datasets motivates our incorporation of visual comparisons between same-type objects, and the need to tie together choices for referring expression generation between objects.

3 Models

We implement several model variations for referring expression generation and comprehension. The first set of models are recent state of the art deep learning approaches from Mao et al. [21]. We use these as our baselines (Sect. 3.1). Next, we investigate incorporating better visual context features into the models (Sect. 3.2). Finally, we explore methods to jointly produce an entire set of referring expressions for all depicted objects of the same category (Sect. 3.3).

3.1 Baselines

For comparison, we implement both the baseline and strong model of Mao et al. [21]. Both models utilize a pre-trained CNN network to model the target object and its context within the image, and then use a LSTM for generation. In particular, object and context are modeled as features from a CNN trained to recognize 1,000 object categories [30] from ImageNet [28]. Specifically, the visual representation is composed of:

- Target object representation, o_i. The object is modeled as features extracted from the VGG-fc7 layer by forwarding its bounding box through the network.
- Global context representation, g_i. Context is modeled as features extracted from the VGG-fc7 layer for the entire image.

– Location/size representation, l_i, for the target object. Location and size are modeled as a 5-d vector encoding the x and y locations of the top left and bottom right corners of the target object bounding box, as well as the bounding box size with respect to the image, i.e., $l_i = [\frac{x_{tl}}{W}, \frac{y_{tl}}{H}, \frac{x_{br}}{W}, \frac{y_{br}}{H}, \frac{w \cdot h}{W \cdot H}]$.

Language generation is handled by a long short-term memory network (LSTM) [8] where inputs are the above visual features and the network is trained to generate natural language referring expressions. In Mao et al.'s baseline [21], the model uses maximum likelihood training and outputs the most likely referring expression given the target object, context, and location/size features. In addition, they also propose a stronger model that uses maximum mutual information (MMI) training to consider whether a listener would interpret a referring expression unambiguously. They impose this by penalizing the model if a generated referring expression could also be generated by some other object within the image. We implement both their original model and MMI model in our experiments. We subsequently refer to these two models as Baseline and MMI, respectively.

3.2 Visual Comparison

Previous works [1,25] have shown that objects in an image, of the same type as the target object, are most important for influencing what attributes people use to describe the target. One drawback of considering a general feature over the entire image to encode context (as in the baseline models) is that it may not specifically focus on visual comparisons to the most relevant objects – the other objects of the same object category within the image.

In this paper, we propose a more explicit encoding of the visual difference between objects of the same category within an image. This helps for generating referring expressions which best discriminate the target object from the surrounding objects. For example, in an image with three cars, two blue and one red, visual appearance comparisons could help generate "the red car" as an expression for the latter object.

Given the referred object and its surrounding objects, we compute two types of features for visual comparison. The first type encodes the similarities and differences in *visual appearance* between the target object and other objects of the same cateogry depicted in the image. Inspired by Sadeghi et al. [29], we compute the difference in visual CNN features as our representation of relative appearance. Because there may be many surrounding objects of the same type in the image, and not every object will provide useful information about how to describe the target object, we need to first select which objects to compare and aggregate their visual differences. In Sect. 5, we experiment with selecting different subsets of comparison objects: objects of the same category, objects of different category, or all other depicted objects. For each selected comparison object, we compute the appearance difference as the subtraction of the target object and comparison object CNN representations. We experiment with three different strategies for computing an aggregate vector to represent the visual

difference between the target object and the surrounding objects: minimum, maximum, and average over each feature dimension. In our experiments, pooling the average difference between the target object and surrounding objects seems to work best. Therefore, we use this pooling in all experiments.

– Visual appearance difference representation, $\delta v_i = \frac{1}{n}\sum_{j\neq i}\frac{o_i - o_j}{\|o_i - o_j\|}$, where n is the number of objects chosen for comparisons and we use average pooling to aggregate the differences.

The second type of comparison feature encodes the *relative location and size* differences between the target object and surrounding objects of the same object category. People often use comparative size or location terms in referring expressions, e.g. "the second giraffe from the left" or "the smaller monkey" [32]. To address the dynamic number of nearby objects, we choose up to five comparison objects of the same category as the target object, sorted by distance to the target. When fewer than five objects of the same category are depicted, this 25-d vector (5-d x 5 surrounding objects) is padded with zeros.

– Location difference representation, δl_i, where each 5-d difference is computed as $\delta l_{ij} = [\frac{[\triangle x_{tl}]_{ij}}{w_i}, \frac{[\triangle y_{tl}]_{ij}}{h_i}, \frac{[\triangle x_{br}]_{ij}}{w_i}, \frac{[\triangle y_{br}]_{ij}}{h_i}, \frac{w_j h_j}{w_i h_i}]$.

In summary, our final visual representation for a target object is:

$$r_i = W_m[o_i, g_i, l_i, \delta v_i, \delta l_i] + b_m \tag{1}$$

where o_i, g_i, l_i are the target object, global context, and location/size features from the baseline model, δv_i and δl_i encodes visual appearance difference and location difference. W_m and b_m project the concatenation of the five types of features to be the final representation.

3.3 Joint Language Generation

For the referring expression generation task, rather than generating sentences for each object in an image separately [12,21], we consider tying the generation process together into a single task to jointly generate expressions for all objects of the same object category depicted in an image. This makes sense intuitively – when a person attempts to generate a referring expression for an object in an image they inherently compose that expression while keeping in mind expressions for the other objects in the picture. This can be observed in the fact that the expressions people generate for objects in an image tend to share similar patterns of expression. If you say "the man on the left" for one object then you tend to say "the man on the right" for the other object. We would like our algorithms to mimic these behaviors. Additionally, the algorithm should also be able to push generated expressions away from each other to create less ambiguous references. For example, if we use the word "red" to describe one object, then we probably shouldn't use the same word to describe another object.

To model this joint generation process, we model generation using an LSTM model where in addition to the usual connections between time steps within

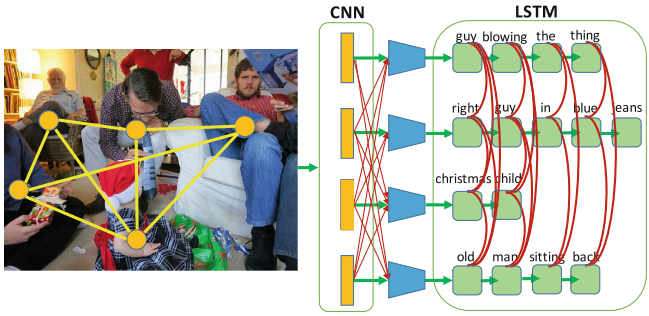

Fig. 2. Framework: we extract VGG-fc7 and location features for each object of the same type, then compute visual differences. These features and differences are then fed into LSTM. For sentence generation, the LSTMs are tied together, incorporating the hidden output difference as additional information for predicting words.

an expression we also add connections between expressions for different objects. This architecture is illustrated in Fig. 2.

Specifically, we use LSTM to generate multiple referring expressions, $\{r_i\}$, given depicted objects of the same type, $\{o_j\}$.

$$\begin{aligned} P(R|O) &= \prod_i P(r_i|o_i, \{o_{j\neq i}\}, \{r_{j\neq i}\}), \\ &= \prod_i \prod_t P(w_{i_t}|w_{i_{t-1}}, ..., w_{i_1}, v_i, \{h_{j_t, j\neq i}\}) \end{aligned} \tag{2}$$

where w_{i_t} are words at time t, v_i visual representations, and h_{j_t} is the hidden output of j-th object at time step t that encodes the visual and sentence information for the j-th object. As visual comparison, we aggregate the difference of hidden outputs to push away ambiguous information. $h_{dif_{i_t}} = \frac{1}{n}\sum_{j\neq i} \frac{h_{i_t} - h_{j_t}}{\|h_{i_t} - h_{j_t}\|}$. There, n is the the number of other objects of the same type. The hidden difference is jointly embedded with the target object's hidden output, and forwarded to the softmax layer for predicting the word.

$$P(w_{i_t}|w_{i_{t-1}}, ..., w_{i_1}, v_i, \{h_{j_t, j\neq i}\}) = \text{softmax}(W_h[h_{i_t}, h_{dif_{i_t}}] + b_h) \tag{3}$$

4 Data

We make use of 3 referring expression datasets in our work, all collected on top of the Microsoft COCO image collection [20]. One dataset, RefCOCOg [21] is collected in a non-interactive setting, while the other two datasets, RefCOCO and RefCOCO+, are collected interactively in a two-player game [15]. In the following, we describe each dataset and provide some analysis of their similarities and differences, and then discuss splits of the datasets used in our experiments.

4.1 Datasets and Analysis

Images for each dataset were selected to contain multiple objects of the same category (object categories depicted cover the 80 common objects from MSCOCO with ground-truth segmentation). These images provide useful cases for referring expression generation since the referrer needs to compose a referring expression that uniquely singles out one object from other relevant objects.

RefCOCOg: This dataset was collected on Amazon Mechanical Turk in a non-interactive setting. One set of workers were asked to write natural language referring expressions for objects in MSCOCO images then another set of workers were asked to click on the indicated object given a referring expression. If the click overlapped with the correct object then the referring expression was considered valid and added to the dataset. If not, another referring expression was collected for the object. This dataset consists of 85,474 referring expressions for 54,822 objects in 26,711 images. Images were selected to contain between 2 and 4 objects of the same object category.

RefCOCO & RefCOCO+: These datasets were collected using the ReferitGame [15]. In this two-player game, the first player is shown an image with a segmented target object and asked to write a natural language expression referring to the target object. The second player is shown only the image and the referring expression and asked to click on the corresponding object. If the players do their job correctly, they receive points and swap roles. If not, they are presented with a new object and image for description. Images in these collections were selected to contain two or more objects of the same object category. In the RefCOCO dataset, no restrictions are placed on the type of language used in the referring expressions while in the RefCOCO+ dataset players are disallowed from using location words in their referring expressions by adding "taboo" words to the ReferItGame. This dataset was collected to obtain a referring expression dataset focsed on purely appearance based description, e.g., "the man in the yellow polka-dotted shirt" rather than "the second man from the left", which tend to be more interesting from a computer vision based perspective and are independent of viewer perspective. RefCOCO consists of 142,209 refer expressions for 50,000 objects in 19,994 images, and RefCOCO+ has 141,564 expressions for 49,856 objects in 19,992 images.

Dataset Comparisons: As shown in Fig. 1, the languages used in RefCOCO and RefCOCO+ datasets tend to be more concise and less flowery than the languages used in the RefCOCOg. RefCOCO expressions have an average length of 3.61 while RefCOCO+ have an average length of 3.53, and RefCOCOg contain an average of 8.43 words. This is most likely due to the differences in collection strategy. RefCOCO and RefCOCO+ were collected in a game scenario where players are trying to efficiently provide enough information to indicate the correct object to the other player. RefCOCOg was collected in independent rounds of Mechanical Turk without any interactive time constraints and therefore tend to provide more complex expressions, often entire sentences rather than phrases.

In addition, RefCOCO and RefCOCO+ do not limit the number of objects of the same type to 4 and thus contain some images with many objects of the same type. Both RefCOCO and RefCOCO+ contain an average of 3.9 same-type objects per image, while RefCOCOg contains an average of 1.63 same-type objects per image. The large number of same-type objects per image in RefCOCO and RefCOCO+ suggests that incorporating visual comparisons to same-type objecs will be useful.

Dataset Splits: There are two types of splits of the data into train/test sets: a per-object split and a people-vs-objects split.

The first type is **per-object split**. In this split, the dataset is divided by randomly partitioning objects into training and testing sets. This means that each object will only appear either in training or testing set, but that one object from an image may appear in the training set while another object from the same image may appear in the test set. We use this split for RefCOCOg since same division was used in the previous state-of-the-art approach [21].

The second type is **people-vs-objects splits**. One thing we observe from analyzing the datasets is that about half of the referred objects are people. Therefore, we create a split for RefCOCO and RefCOCO+ datasets that evaluates images containing multiple people (testA) vs images containing multiple instances of all other objects (testB). In this split all objects from an image will appear either in the training or testing sets, but not both. This split creates a more meaningfully separated division between training and testing, allowing us to evaluate the usefulness of context more fairly.

5 Experiments

We first perform some experiments to analyze the use of context in referring expressions (Sect. 5.1). Given these findings, we then perform experiments evaluating the usefulness of our proposed visual and language innovations on the comprehension (Sect. 5.2) and generation tasks (Sect. 5.3).

In experiments for the referring expression comprehension task, we use the same evaluation as Mao et al. [21], namely we first predict the region referred by the given expression, then we compute the intersection over union (IOU) ratio between the true and predicted bounding box. If the IOU is larger than 0.5 we count it as a true positive. Otherwise, we count it as a false positive. We average this score over all images. For the referring expression generation task we use automatic evaluation metrics, BLEU, ROUGE, and METEOR developed for evaluating machine translation results, commonly used to evaluate language generation results [3,14,19,22,33,35]. We further perform human evaluations, and propose a new metric evaluating the duplicate rate of generated expressions. For both tasks, we compare our models with "Baseline" and "MMI" [21]. Specifically, we denote "visdif" as our visual comparison model, and "tie" as the LSTM tying model. We also perform an ablation study, evaluating the combinations.

5.1 Analysis Experiments

Context Representation: As previously discussed, we suggest that the approaches proposed in recent referring expression works [11,21] make use of relatively weak contextual information, by only considering a single global image context for all objects. To verify this intuition, we implemented both the baseline and strong MMI models from Mao et al. [21], and compare the results for referring expression comprehension task with and without global context on RefCOCO and Refcoco+ in Table 1. Surprisingly we find that the global context does not improve the performance of the model. In fact, adding context even decreases performance slightly. This may be due to the fact that the global context for each object in an image would be the same, introducing some ambiguity into the referring expression comprehension task. Given these findings, we implemented a simple modification to the global context, computing the same visual representation, but on a somewhat scaled window centered around the target object. We found this to improve performance, suggesting room for improving the visual context feature. This motivate our development of a better context feature.

Table 1. Expression comprehension accuracies on RefCOCO and RefCOCO+ of the Baseline model with differenct context source. Scale n indicates the size of the cropped window centered by the target object.

	RefCOCO		RefCOCO+	
	Test A	Test B	Test A	Test B
No context	63.91 %	66.31 %	50.09 %	45.05 %
Global context	63.15 %	64.21 %	48.73 %	42.13 %
Scale 2	65.57 %	67.13 %	50.38 %	44.89 %
Scale 3	66.14 %	68.07 %	50.25 %	45.40 %
Scale 4	**66.68 %**	**68.56 %**	**50.34 %**	**45.48 %**

Visual Comparison: For our visual appearance comparison feature, we have some choice regarding which objects from the image should be compared to the target object. We experiment with three sets of reference objects: (a) objects of the same-category, (b) objects of different categories, and (c) all objects appearing in the image. We refer the readers to the supplementary file for details. The results show that visual appearance comparisons to objects of the same category are most useful for comprehension task. Therefore, we use this subset of objects for visual comparisons in all of the remaining experiments.

5.2 Referring Expression Comprehension

We evaluate performance on the referring expression comprehension task on RefCOCO, RefCOCO+ and RefCOCOg datasets. For RefCOCO and RefCOCO+,

we evaluate on the two subsets of people (testA) and all other objects (testB). For RefCOCOg, we evaluate on the per-object split as previous work [21]. Since the authors haven't released their testing set, we show the performance on their validation set only, using the optimized hyper-parameters on RefCOCO. Table 2 shows the comprehension accuracies. We observe that our implementation of Mao et al. [21] achieves comparable performance to the numbers reported in their paper. We also find that adding visual comparison features to the Baseline model improves performance across all datasets and splits. Similar improvements are also observed on top of the MMI model.

Table 2. Referring expression comprehension results on the RefCOCO, RefCOCO+, and RefCOCOg datasets. Rows of "method(det)" are the results of automatic system built on Fast-RCNN detections.

	RefCOCO		RefCOCO+		RefCOCOg
	Test A	Test B	Test A	Test B	Validation
Baseline [21]	63.15 %	64.21 %	48.73 %	42.13 %	55.16 %
visdif	67.57 %	71.19 %	52.44 %	47.51 %	59.25 %
MMI [21]	71.72 %	71.09 %	58.42 %	51.23 %	62.14 %
visdif+MMI	**73.98 %**	**76.59 %**	**59.17 %**	**55.62 %**	**64.02 %**
Baseline(det) [21]	58.32 %	48.48 %	46.86 %	34.04 %	40.75 %
visdif(det)	62.50 %	50.80 %	50.10 %	37.48 %	41.85 %
MMI(det) [21]	64.90 %	54.51 %	54.03 %	42.81 %	45.85 %
visdif+MMI(det)	**67.64%**	**55.16%**	**55.81%**	**43.43%**	**46.86%**

In order to make a fully automatic referring system, we also train a Fast-RCNN [7] detector and build our system on top of the detections[1]. Results on shown in the bottom half of Table 2. Although all comprehension accuracies drop due to imperfect detections, the improvements of our models over Baseline and MMI are still observed. One weakness of our automatic system is that it highly depends on detection performance, especially for general objects (testB). Note, our detector was trained on MSCOCO validation only, thus we believe such weakness may be alleviated with more training data and stronger detection techniques. Some examples for referring expression comprehension are shown in Fig. 3, where top 2 rows show correct comprehensions (object correctly localized) and bottom two rows show incorrect comprehensions (wrong object localized).

[1] We train Fast-RCNN on the validation portion only as the RefCOCO and RefCOCO+ are collected using MSCOCO training data. For RefCOCOg, we use the detection results provided by [21].

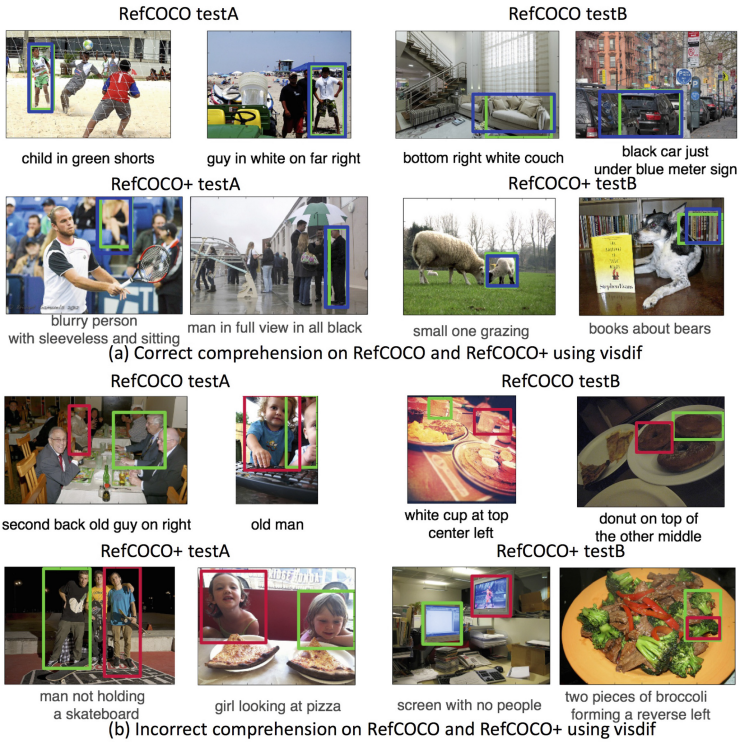

Fig. 3. Some examples showing comprehension results of visdif+MMI based on detection. The blue and red bounding boxes are correct and incorrect comprehension respectively, while the green boxes indicate the ground-truth regions. (Color figure online)

5.3 Referring Expression Generation

For the referring expression generation task, we evaluate the usefulness of our visual comparison features as well as our joint language generation model. These serve to tie the generation process together so that the model considers other objects of the same type both visually and linguistically during generation. On the visual side, comparisons are used to judge similarity of the target object to other objects of the same type in terms of appearance, size and location. On the language side, the joint LSTM model serves to both differentiate and mimic language patterns in the referring expressions for the entire set of depicted objects. Figure 4 shows some comparison between our model with other methods.

Our full results are shown in Table 3. We find that incorporating our visual comparison features into the Baseline model improves generation quality (compare row "Baseline" to row "visdif"). It also improves the performance of MMI model (compare row "MMI" to row "visdif+MMI"). We also observe that tying the language generation together across all objects consistently improves the performance (compare the bottom three "+tie" rows with the above). Especially for

Fig. 4. Referring expression generation by different methods.

Table 3. Referring expression generation results: Bleu, Rouge, Meteor evaluations for RefCOCO, RefCOCO+ and RefCOCOg.

RefCOCO

	Test A				Test B			
	Bleu 1	Bleu 2	Rouge	Meteor	Bleu 1	Bleu 2	Rouge	Meteor
Baseline [21]	0.477	0.290	0.413	0.173	0.553	0.343	0.499	0.228
MMI [21]	0.478	0.295	0.418	0.175	0.547	0.341	0.497	0.228
visdif	0.505	**0.322**	0.441	0.184	0.583	0.382	0.530	0.245
visdif+MMI	0.494	0.307	0.441	0.185	0.578	0.375	0.531	0.247
Baseline+tie	0.490	0.308	0.431	0.181	0.561	0.352	0.505	0.234
visdif+tie	**0.510**	0.318	**0.446**	**0.189**	**0.593**	**0.386**	**0.533**	**0.249**
visdif+MMI+tie	0.506	0.312	0.445	0.188	0.579	0.370	0.525	0.246

RefCOCO+

	Test A				Test B			
	Bleu 1	Bleu 2	Rouge	Meteor	Bleu 1	Bleu 2	Rouge	Meteor
Baseline [21]	0.391	0.218	0.356	0.140	0.331	0.174	0.322	0.135
MMI [21]	0.370	0.203	0.346	0.136	0.324	0.167	0.320	0.133
visdif	0.407	0.235	0.363	0.145	0.339	0.177	0.325	0.145
visdif+MMI	0.386	0.221	0.360	0.142	0.327	0.172	0.325	0.135
Baseline+tie	0.392	0.219	0.361	0.143	0.336	0.177	0.325	0.140
visdif+tie	**0.409**	**0.232**	**0.372**	**0.150**	**0.340**	**0.178**	**0.328**	**0.143**
visdif+MMI+tie	0.393	0.220	0.360	0.142	0.327	0.175	0.321	0.137

RefCOCOg

	validation			
	Bleu 1	Bleu 2	Rouge	Meteor
Baseline [21]	0.437	0.273	0.363	0.149
MMI [21]	0.428	0.263	0.354	0.144
visdif	**0.442**	**0.277**	**0.370**	**0.151**
visdif+MMI	0.430	0.262	0.356	0.145

method "visdif+tie", it achieves the highest score under almost every measurement. We do not perform language tying on RefCOCOg since here some objects from an image may appear in training while others may appear in testing.

We observe in Table 3 that models incoporating "+MMI" are worse than without "+MMI" under the automatic scoring metrics. To verify whether these metrics really reflect performance, we performed human evaluations on the expression generation task. Three Turkers were asked to click on the referred object given the image and the generated expression. If more than two clicked on the true target object, we consider this expression to be correct. Table 4 shows the human evaluation results, indicating that models with "+MMI" are consistently higher performance. We also find "+tie" methods perform the best, indicating that tying language together is able to produce less ambiguous referring expressions. Figure 5 shows examples of tied generations.

Table 4. Human evaluations on referring expression generation.

	RefCOCO		RefCOCO+	
	Test A	Test B	Test A	Test B
Baseline [21]	62.42 %	64.99 %	49.18 %	42.03 %
MMI	65.76 %	68.25 %	49.84 %	45.38 %
visdif	68.27 %	74.92 %	55.20 %	43.65 %
visdif+MMI	70.25 %	75.47 %	53.56 %	47.58 %
Baseline+tie	64.51 %	68.34 %	52.06 %	43.53 %
visdif+tie	**71.40 %**	76.14 %	**57.17 %**	47.92 %
visdif+MMI+tie	70.01 %	**76.31 %**	55.64 %	**48.04%**

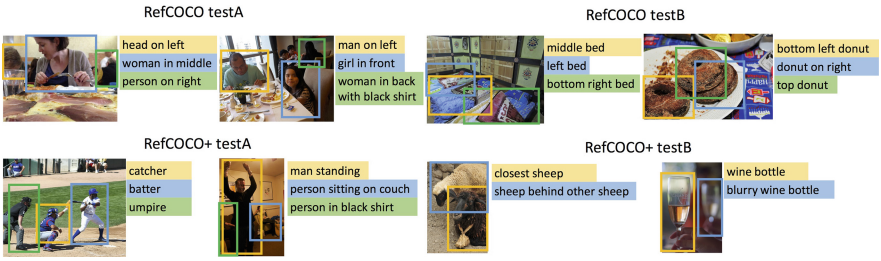

Fig. 5. Joint referring expression generation using the tied language model of "visdif+MMI+tie".

Finally, we introduce another evaluation metric which measures the fraction of images for which an algorithm produces the same generated referring expression for multiple objects within the image. Obviously, a good referring expression generator should never produce the same expressions for two objects within the same image. Thus we would like this number to be as small as possible. The

Table 5. Fraction of images for which the algorithm generates the same referring expression for multiple objects. Smaller is better.

	RefCOCO		RefCOCO+	
	Test A	Test B	Test A	Test B
Baseline [21]	15.60 %	16.40 %	28.67 %	46.27 %
MMI	11.60 %	11.73 %	21.07 %	26.40 %
visdif	9.20 %	8.80 %	19.60 %	31.07 %
visdif+MMI	5.07 %	6.13 %	12.13 %	16.00 %
Baseline+tie	11.20 %	14.93 %	22.00 %	32.13 %
visdif+tie	**4.27 %**	5.33 %	11.73 %	16.27 %
visdif+MMI+tie	6.53 %	**4.53 %**	**10.13 %**	**13.33 %**

evaluation results under such metric are shown in Table 5. We find "+MMI" produces smaller number of duplicated expressions on both RefCOCO and RefCOCO+, while "+tie" helps generating even more different expressions. Our combined model "visdif+MMI+tie" performs the best under this metric.

6 Conclusion

In this paper, we have developed a new model for incorporating detailed context into referring expression models. With this visual comparison based context we have improved performance over previous state of the art for referring expression generation and comprehension. In addition, for the referring expression generation task, we explore methods for joint generation over all relevant objects. Experiments verify that this joint generation improves results over previous attempts to reduce ambiguity during generation.

References

1. Brown-Schmidt, S., Tanenhaus, M.K.: Watching the eyes when talking about size: an investigation of message formulation and utterance planning. J. Mem. Lang. **54**(4), 592–609 (2006)
2. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: CVPR (2015)
3. Fang, H., Gupta, S., Iandola, F., Srivastava, R.K., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Platt, J.C., et al.: From captions to visual concepts and back. In: CVPR (2015)
4. Farhadi, A., Hejrati, M., Sadeghi, M.A., Young, P., Rashtchian, C., Hockenmaier, J., Forsyth, D.: Every picture tells a story: generating sentences from images. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 15–29. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15561-1_2

5. FitzGerald, N., Artzi, Y., Zettlemoyer, L.S.: Learning distributions over logical forms for referring expression generation. In: EMNLP, pp. 1914–1925 (2013)
6. Funakoshi, K., Watanabe, S., Kuriyama, N., Tokunaga, T.: Generating referring expressions using perceptual groups. In: Belz, A., Evans, R., Piwek, P. (eds.) INLG 2004. LNCS, vol. 3123, pp. 51–60. Springer, Heidelberg (2004)
7. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
8. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: a search space odyssey (2015). arXiv preprint arXiv:1503.04069
9. Grice, H.P.: Logic and conversation. In: Cole, P., Morgan, J.L. (eds.) Syntax and Semantics: Speech Acts, vol. 3, pp. 41–58. Academic Press, San Diego (1975)
10. Hodosh, M., Young, P., Hockenmaier, J.: Framing image description as a ranking task: data, models and evaluation metrics. J. Artif. Intell. Res. **47**, 853–899 (2013)
11. Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., Darrell, T.: Natural language object retrieval. In: CVPR (2016)
12. Johnson, J., Karpathy, A., Fei-Fei, L.: Densecap: fully convolutional localization networks for dense captioning (2015). arXiv preprint arXiv:1511.07571
13. Jordan, P., Walker, M.: Learning attribute selections for non-pronominal expressions. In: ACL (2000)
14. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: CVPR (2015)
15. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.L.: ReferitGame: referring to objects in photographs of natural scenes. In: EMNLP, pp. 787–798 (2014)
16. Kelleher, J.D., Kruijff, G.J.M.: Incremental generation of spatial referring expressions in situated dialog. In: ACL (2006)
17. Kiros, R., Salakhutdinov, R., Zemel, R.S.: Unifying visual-semantic embeddings with multimodal neural language models. In: TACL (2015)
18. Krahmer, E., Van Deemter, K.: Computational generation of referring expressions: a survey. Comput. Linguist. **38**(1), 173–218 (2012)
19. Kulkarni, G., Premraj, V., Ordonez, V., Dhar, S., Li, S., Choi, Y., Berg, A.C., Berg, T.: Babytalk: understanding and generating simple image descriptions. IEEE Trans. Pattern Anal. Mach. Intell. **35**, 2891–2903 (2013)
20. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10602-1_48
21. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: CVPR (2016)
22. Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., Yuille, A.: Deep captioning with multimodal recurrent neural networks (m-RNN). In: ICLR (2015)
23. Mitchell, M., van Deemter, K., Reiter, E.: Natural reference to objects in a visual domain. In: Proceedings of the 6th International Natural Language Generation Conference, pp. 95–104. Association for Computational Linguistics (2010)
24. Mitchell, M., Reiter, E., van Deemter, K.: Typicality and object reference. Cognitive Science (CogSci) (2013)
25. Mitchell, M., Van Deemter, K., Reiter, E.: Generating expressions that refer to visible objects. In: HLT-NAACL, pp. 1174–1184 (2013)
26. Ordonez, V., Kulkarni, G., Berg, T.L.: Im2Text: describing images using 1 million captioned photographs. In: Advances in Neural Information Processing Systems (2011)

27. Rohrbach, A., Rohrbach, M., Hu, R., Darrell, T., Schiele, B.: Grounding of textual phrases in images by reconstruction (2015). arXiv preprint arXiv:1511.03745
28. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
29. Sadeghi, F., Zitnick, C.L., Farhadi, A.: Visalogy: answering visual analogy questions. In: NIPS (2015)
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). arXiv preprint arXiv:1409.1556
31. Socher, R., Karpathy, A., Le, Q.V., Manning, C.D., Ng, A.Y.: Grounded compositional semantics for finding and describing images with sentences. Trans. Assoc. Comput. Linguist. **2**, 207–218 (2014)
32. Viethen, J., Dale, R.: The use of spatial relations in referring expression generation. In: Proceedings of the Fifth International Natural Language Generation Conference, pp. 59–67. Association for Computational Linguistics (2008)
33. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: a neural image caption generator. In: CVPR (2015)
34. Winograd, T.: Understanding natural language. Cogn. Psychol. **3**(1), 1–191 (1972)
35. Xu, K., Ba, J., Kiros, R., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: neural image caption generation with visual attention. In: ICML (2015)