This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

A Joint Speaker-Listener-Reinforcer Model for Referring Expressions

Licheng Yu, Hao Tan, Mohit Bansal, Tamara L. Berg
Department of Computer Science
University of North Carolina at Chapel Hill
{licheng, airsplay, mbansal, tlberg}@cs.unc.edu

Abstract

Referring expressions are natural language constructions used to identify particular objects within a scene. In this paper, we propose a unified framework for the tasks of referring expression comprehension and generation. Our model is composed of three modules: speaker, listener, and reinforcer. The speaker generates referring expressions, the listener comprehends referring expressions, and the reinforcer introduces a reward function to guide sampling of more discriminative expressions. The listener-speaker modules are trained jointly in an end-to-end learning framework, allowing the modules to be aware of one another during learning while also benefiting from the discriminative reinforcer's feedback. We demonstrate that this unified framework and training achieves state-of-the-art results for both comprehension and generation on three referring expression datasets. [1]

1. Introduction

People often use referring expressions in their everyday discourse to unambiguously identify or indicate particular objects within their physical environment. For example, one might point out a person in the crowd by referring to them as "the man in the blue shirt" or you might ask someone to "pass me the red pen on the table." In both of these examples, we have a pragmatic interaction between two people (or between a person and an intelligent agent such as a robot). First, we have a speaker who must generate an expression given a target object and its surrounding world context. Second, we have a listener who must interpret and comprehend the expression and map it to an object in the environment. Therefore, in this paper we propose an end-to-end trained listener-speaker framework that models these behaviors jointly.

In addition to the listener and speaker, we also introduce a new reinforcer module that learns a discriminative reward model to help generate less ambiguous expressions (expressions that apply to the target object but not to other objects in the image). This goal corresponds to the Gricean Maxim [8] of manner, where one tries to be as clear, brief, and orderly as possible while avoiding obscurity and ambiguity. Avoiding ambiguity is important because the generated expression should be easily and uniquely mapped to the target object. For example, if there were two pens on the table one "long and red" and the other "short and red", asking for the "red pen" would be ambiguous while asking for the "long pen" would be better. The reinforcer module is incorporated using reinforcement learning, inspired by behavioral psychology that says that agents operating in an environment should take actions that maximize their expected cumulative reward. In our case, the reward takes the form of a discriminative classifier trained to reward the speaker for generating less ambiguous expressions.

Within the realm of referring expressions, there are two tasks that can be computationally modeled, mimicking the listener and speaker roles. Referring Expression Generation (speaker) requires an algorithm to generate a referring expression for a given target object in a visual scene, as in Fig. 2. Referring Expression Comprehension (listener) requires an algorithm to localize the object/region in an image described by a given referring expression, as in Fig. 3.

The Referring Expression Generation (REG) task has been studied since the 1970s [29]. Many of the early works in this space focused on relatively limited datasets, using synthesized images of objects in artificial scenes or limited sets of real-world objects in simplified environments [20, 7, 15]. Recently, the research focus has shifted to more complex natural image datasets and has expanded to include the Referring Expression Comprehension task [13, 19, 31] as well as to real-world interactions with robotics [4, 3]. One reason this has become feasible is that several large-scale REG datasets [13, 31, 19] have been collected where deep learning models can be applied.

Recent neural approaches to the referring expression generation and comprehension tasks can be roughly split into two types. The first type uses a CNN-LSTM encoder-decoder generative model [25] to generate (decode) sentences given the encoded target object. With careful de-

[1] Project and demo page: https://vision.cs.unc.edu/refer.

sign of the visual representation of target object, this model can generate unambiguous expressions [19, 31]. Here, the CNN-LSTM models $P(r|o)$, where r is the referring expression and o is the target object, which can be easily converted to $P(o|r)$ via Bayes' rule and used to address the comprehension task [10, 19, 31, 21] by selecting the o with the largest posterior probability. The second type of approach uses a joint-embedding model that projects both a visual representation of the target object and a semantic representation of the expression into a common space and learns a distance metric. Generation and comprehension can be performed by embedding a target object (or expression) into the embedding space and retrieving the closest expression (or object) in this space. This type of approach typically achieves better comprehension performance than the CNN-LSTM model as in [23, 26], but previously was only applied to the referring expression comprehension task. Recent work [1] has also used both an encoder-decoder model (speaker) and an embedding model (listener) for referring expression generation in abstract images, where the offline listener reranks the speaker's output.

In this paper, we propose a unified model that *jointly* learns both the CNN-LSTM speaker and embedding-based listener models, for both the generation and comprehension tasks. Additionally, we add a discriminative reward-based *reinforcer* to guide the sampling of more discriminative expressions and further improve our final system. Instead of working independently, we let the speaker, listener, and reinforcer interact with each other, resulting in improved performance on both generation and comprehension tasks. Results evaluated on three standard, large-scale datasets verify that our proposed listener-speaker-reinforcer model significantly outperforms the state-of-the-art on both the comprehension task (Tables 1 and 2) and the generation task (evaluated using human judgements in Table 4, and automatic metrics in Table 3).

2. Related work

Recent years have witnessed a rise in multimodal research related to vision and language. Given the individual success in each area and the need for models with more advanced cognition capabilities, several tasks have emerged as evaluation applications, including image captioning, visual question answering, and referring expression generation/comprehension.

Image Captioning: The aim of image captioning is to generate a sentence describing the general content of an image. Most recent approaches use deep learning to address this problem. Perhaps the most common architecture is a CNN-LSTM model [25], which generates a sentence conditioned on visual information from the image. One paper related to our work is gLSTM [11] which uses CCA semantics to guide the caption generation. A further step beyond image captioning is to locate the regions being described in captions [27, 22, 26]. The Visual Genome [16] collected captions for dense regions in an image that have been used for dense-captioning tasks [12]. There also has been a movement toward more focused tasks, such as visual question answering [2, 30], and referring expression generation and comprehension which involve specific regions/objects within an image (discussed below).

Referring Expression Datasets: REG has been studied for many years [29, 15, 20] in linguistics and natural language processing, but mainly focused on small or artificial datasets. In 2014, Kazemzadeh et al [13] introduced the first large-scale dataset RefCLEF using 20,000 real-world natural images [9]. This dataset was collected in a two-player game, where the first player writes a referring expression given an indicated target object. The second player is shown only the image and expression and has to click on the correct object described by the expression. If the click lies within the target object region, both sides get points and their roles switch. Using the same game interace, the authors further collected RefCOCO and RefCOCO+ datasets on MSCOCO images [31]. The RefCOCO and RefCOCO+ datasets each contain 50,000 referred objects with 3 referring expressions on average. The main difference between RefCOCO and RefCOCO+ is that in RefCOCO+, players were forbidden from using absolute location words, e.g. left dog, therefore focusing the referring expression to purely appearance-based descriptions. In addition, Mao et al [19] also collected a referring expression dataset - RefCOCOg, using MSCOCO images, but in a non-interactive framework. These expressions are more similar to the MSCOCO captions in that they are longer and more complex as their was no time constraint in the non-interactive data collection setting. This dataset has 96,654 objects with 1.3 expressions per object on average.

Referring Expression Comprehension and Generation: Referring expressions are associated with two tasks, comprehension and generation. The comprehension task requires a system to select the region being described by a given referring expression. To address this problem, [19, 31, 21, 10] model $P(r|o)$ and looks for the object o maximizing the probability. People also try modeling $P(o, r)$ directly using embedding model [23, 26], which learns to minimize the distance between paired object and sentence in the embedding space. The generation task asks a system to compose an expression for a specified object within an image. While some previous work used rule-based approaches to generate expressions with fixed grammar pattern [20, 5, 13], recent work has followed the CNN-LSTM structure to generate expressions [19, 31].

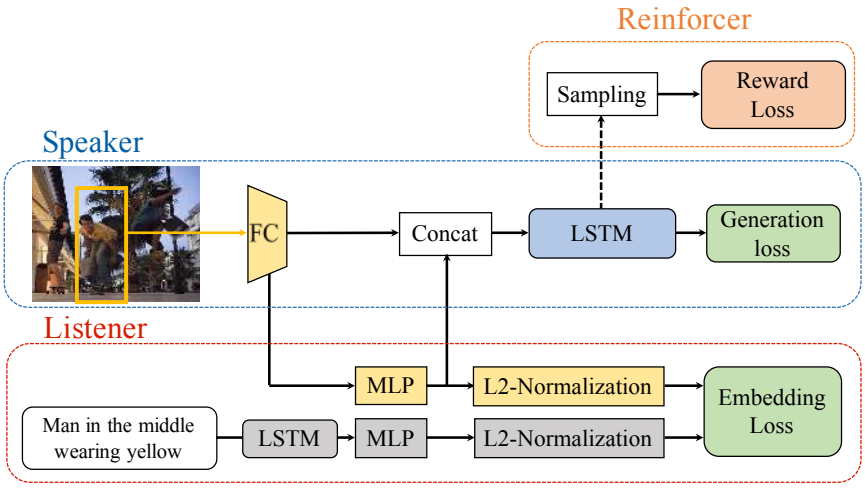

Figure 1: Framework: The Speaker is a CNN-LSTM model, which generates a referring expression for the target object. The Listener is a joint-embedding model learned to minimize the distance between paired object and expression representations. In addition, a Reinforcer module helps improve the speaker by sampling more discriminative expressions for training.

3. Model

Our model is composed of three modules: speaker (Sec 3.1), listener (Sec 3.2), and reinforcer (Sec 3.3). During training, the speaker and listener are trained jointly so that they can benefit from each other and from the reinforcer. As the reward function for the reinforcer is not differentiable, it is incorporated using reinforcement learning policy gradient algorithm.

3.1. Speaker

For our speaker module, we follow the previous state-of-the-art [19, 31], and use a CNN-LSTM framework. Here, a pre-trained CNN model is used to define a visual representation for the target object and other visual context. Then, a Long-short term memory (LSTM) is used to generate the most likely expression given the visual representation.

Because of the improved quantitative performance over [19], we use the visual comparison model of [31] as our speaker (to encode the target object). Here, the visual representation includes the target object, context, location/size features, and two visual comparison features. Specifically, the target object representation o_i is modeled as the fc7 features from a pre-trained VGG network [24]. Global context, g_i, is modeled as features extracted from the VGG-fc7 layer for the entire image. The location/size representation, l_i, for the target object is modeled as a 5-dimensional 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., $l_i = [\frac{x_{tl}}{W}, \frac{y_{tl}}{H}, \frac{x_{br}}{W}, \frac{y_{br}}{H}, \frac{w \cdot h}{W \cdot H}]$.

As referring expressions often relate an object to other objects of the same type within the image ("the red ball" vs "the blue ball" or "the larger elephant"), comparisons tend to be quite important for differentiation. The comparison features are composed of two parts: a) appearance similarity $\delta v_i = \frac{1}{n} \sum_{j \neq i} \frac{o_i - o_j}{\|o_i - o_j\|}$, where n is the number of objects chosen for comparisons, b) location and size similarity δl_i, concatenating the 5-d difference on each compared object $\delta l_{ij} = [\frac{[\triangle x_{tl}]_{ij}}{w_i}, \frac{[\triangle y_{tl}]_{ij}}{h_i}, \frac{[\triangle x_{br}]_{ij}}{w_i}, \frac{[\triangle y_{br}]_{ij}}{h_i}, \frac{w_j h_j}{w_i h_i}]$.

The final visual representation for the target object is then a concatenation of the above features followed by a fully-connected layer fusing them together, $r_i = W_m[o_i, g_i, l_i, \delta v_i, \delta l_i] + b_m$. This joint feature is then fed into the LSTM for referring expression generation. During training we minimize the negative log-likelihood:

$$\begin{aligned} L_1^s(\theta) &= -\sum_i \log P(r_i | o_i; \theta) \\ &= -\sum_i \sum_t \log P(r_i^t | r_i^{t-1}, \ldots, r_i^1, o_i; \theta) \end{aligned} \tag{1}$$

Note that the speaker can be modeled using any form of CNN-LSTM structure.

In [19], Mao proposed to add a Maximum Mutual Information (MMI) constraint encouraging the generated expression to describe the target object better than the other objects within the image (i.e., a ranking loss on objects). We generalize this idea to incorporate two triplet hinge losses composed of a positive match and two negative matches. Given a positive match (r_i, o_i), we sample the contrastive pair (r_j, o_i) where r_j is the expression describing some other object and pair (r_i, o_k) where o_k is some other object in the same image, then we optimize the following max-margin

loss:

$$L_2^s(\theta) = \sum_i [\lambda_1^s \max(0, M + \log P(r_i|o_k) - \log P(r_i|o_i)) + \lambda_2^s \max(0, M + \log P(r_j|o_i) - \log P(r_i|o_i))] \quad (2)$$

The first term is from [19], while the second term encourages that the target object to be better described by the true expression compared to expressions describing other objects in the image (i.e., a ranking loss on expressions).

3.2. Listener

We use a joint-embedding model to mimick the listener's behaviour. The purpose of this embedding model is to encode the visual information from the target object and semantic information from the referring expression into a joint space that embeds vectors that are visually or semantically related closer together in the space. Here for the referring expression comprehension task, given a referring expression representation, the listener embeds it into the joint space, then selects the closest object in the embedding space for the predicted target object.

As illustrated in Fig. 1, for our listener joint-embedding model (outlined by a red dashline), we use an LSTM to encode the input referring expression and the same visual representation as the speaker to encode the target object (thus connecting the speaker to the listener). We then add two MLPs (multi-layer perceptions) and two L2 normalization layers following each view, the object and the expression. Each MLP is composed of two fully connected layers with ReLU nonlinearities between them, serving to transform the object view and the expression view into a common embedding space. The inner-product of the two normalized representations is computed as their similarity score $S(r, o)$ in the space. As a listener, we force the similarity on target object and referring expression pairs by applying a hinge loss over triplets, which consist of a positive match and two negative matches:

$$L^l(\theta) = \sum_i [\lambda_1^l \max(0, M + S(r_i, o_k) - S(r_i, o_i)) + \lambda_2^l \max(0, M + S(r_j, o_i) - S(r_i, o_i))] \quad (3)$$

where the negative matches are randomly chosen from the other objects and expressions in the same image.

Note that the listener model is not limited to this particular triplet-based model. For example, [23] computes a similarity score between every object for given referring expression, and minimizes the cross entropy of the SoftMax knowing the target object, which could also be applied here.

3.3. Reinforcer

Besides using the ground-truth pairs of target object and referring expression for training the speaker, we also use reinforcement learning to guide the speaker toward generating less ambiguous expressions (expressions that apply to the target object but not to other objects). This reinforcer module is composed of a discriminative reward function and performs a non-differentiable policy gradient update to the speaker.

Specifically, given the softmax output of the speaker's LSTM, we sample words according to the categorical distribution at each time step, resulting in a complete expression after sampling the <END> token. This sampling operation is non-differentiable as we do not know whether an expression is ambiguous or not until we feed it into a reward function. Therefore, we use policy gradient reinforcement learning to update the speaker's parameters. Here, the goal is to maximize the reward expectation $F(w_{1:T})$ under the distribution of $p(w_{1:T}; \theta)$ parameterized by the speaker, i.e., $J = E_{p(w_{1:T})}[F]$. According to the policy gradient algorithm [28], we have

$$\nabla_\theta J = E_{p(w_{1:T})}[F(w_{1:T}) \nabla_\theta \log p(w_{1:T}; \theta)], \quad (4)$$

Where $\log p(w_t)$ is defined by the softmax output. We then use this gradient to update our speaker model during training.

The only thing left is to choose a reward function that encourages the speaker to sample less ambiguous expressions. As illustrated in Fig. 1 (outlined in dashed orange), the reinforcer module learns a reward function using paired objects and expressions. We again use the same visual representation for the target object and use another LSTM to encode the expression representation. Rather than using two MLPs to encode each view as in the listener, here we concatenate the two views and feed them together into a MLP to learn a 1-d Logistic Regression score between 0 and 1. Trained with cross-entropy loss, the reward function computes a match score between an input object and expression. We use this score as the reward signal in Eqn. 4 for sampled expression and target object pairs. After training, the reward function is fixed to assist our joint speaker-listener system.

3.4. Joint Model

In this subsection, we describe some specifics of how our three modules (speaker, listener, reinforcer) are integrated into a joint framework (shown in Fig. 1). For the listener, we notice that the visual vector in the embedding space is learned to capture the neighbourhood vectors of referring expressions, thus making it aware of the listener's knowledge. Therefore, we take this MLP embedded vector as an additional input for the speaker, which encodes the listener based information. In Fig. 1, we use concatenation to jointly encode the standard visual representation of target object and this listener-aware representation and then feed them into speaker. Besides concatenation, the element-wise product or compact bilinear pooling can also be applied [6]. During training, we sample the same triplets for

both the speaker and listener, and make the word embedding of the speaker and listener shared to reduce the number of parameters. For the reinforcer module, we do sentence sampling using the speaker's LSTM as shown in the top right of Fig. 1. Within each mini-batch, the sampled expressions for the target objects are fed into the reward function to obtain reward values.

The overall loss function is formulated as a multi-task learning problem:

$$\theta = \arg\min L_1^s(\theta) + L_2^s(\theta) + L^l(\theta) - \lambda^r J(\theta), \quad (5)$$

where λ^r is the weight on reward loss. The weights on the loss of speaker and listener are already included in Eqn. 2 and Eqn. 3. We list all hyper-parameters settings in Sec. 4.1.

3.5. Comprehension and Generation

For the comprehension task, at test time, we can use either the speaker or listener to select the target object given an input expression. Using the listener, we would embed the input expression into the learned embedding space and select the closest object as the predicted target. Using the speaker, we would generate expressions for each object within the image and then select the object whose generated expression best matches the input expression. Therefore, we utilize both modules by ensembling the speaker and listener predictions together to pick the most probable object given a referring expression.

$$\hat{o} = \arg\max_o P(r|o)S(o,r)^\lambda \quad (6)$$

Surprisingly, using the speaker alone (setting λ to 0) already achieves state-of-art results due to our joint training. Adding the listener further improves performance more than 4% over previous state-of-art results.

For the generation task, we first let the speaker generate multiple expressions per object via beam search. We then use the listener to rerank these expressions and select the least ambiguous expression, which is similar to [1]. To fully utilize the listener's power in generation, we propose to consider cross comprehension as well as the diversity of expressions by minimizing the potential:

$$\begin{aligned} E(r) &= \sum_i \theta_i(r_i) + \sum_{i,j} \theta_{i,j}(r_i, r_j) \\ \theta_i(r_i) &= -\log P(r_i|o_i) - \lambda_1 \log S(r_i, o_i) \\ &\quad + \lambda_2 \max_{j \neq i} \log S(r_i, o_j) \\ \theta_{i,j}(r_i, r_j) &= \lambda_3 I(r_i = r_j) \end{aligned} \quad (7)$$

The first term and second term in the unary potential measure how well the target object and generated expression match using the speaker and listener modules respectively (also used in [1]). The third term in the unary potential measures the likelihood of the generated sentence of describing other objects in the same image. The pairwise potential penalizes the same sentence being generated for different objects (encouraging diversity in generation). In this way, the expressions for every object in an image are jointly generated. Compared with the previous model that attempted to tie language generation of referring expressions together [31], the constraints in Eqn. 7 are more explicit and overall this works better to reduce ambiguity in the generated expressions.

4. Experiments

4.1. Optimization

We optimize our model using Adam [14] with an initial learning rate of 0.0004, halved every 2,000 iterations, with a batch size of 32. The word embedding size and hidden state size of the LSTM are set to 512. To avoid overfitting, we apply dropout with a ratio of 0.2 after each linear transformation in the MLP layers. We also regularize the word-embedding and output layers of the speaker's LSTM using dropout with ratio of 0.5. For the constrastive pairs, we set $\lambda_1^l = 1$ and $\lambda_2^l = 1$ in listener (Eqn. 3), and set $\lambda_1^s = 1$ and $\lambda_2^s = 0.1$ in speaker (Eqn. 2). The weight on reward loss is set as $\lambda^r = 1$.

4.2. Datasets

We perform experiments on three referring expression datasets: RefCOCO, RefCOCO+ and RefCOCOg (described in Sec 2). All three datasets are collected on MSCOCO images [17], but with several differences: 1) RefCOCO and RefCOCO+ were collected using an interactive game interface while RefCOCOg was collected in a non-interactive setting and contains longer expressions, 2) RefCOCOg contains on average 1.63 objects of the same type per images, while RefCOCO and RefCOCO have 3.9 on average, 3) RefCOCO+ disallowed absolute location words in referring expressions. Overall, RefCOCO has 142,210 expressions for 50,000 objects in 19,994 images, RefCOCO+ has 141,565 expressions for 49,856 objects in 19,992 images, and RefCOCOg has 104,560 expressions for 54,822 objects in 26,711 images.

Additionally, each dataset is provided with dataset splits for evaluation. RefCOCO and RefCOCO+ provide person vs. object splits for evaluation. Images containing multiple people are in "TestA" while images containing multiple objects of other categories are in "TestB". For RefCOCOg, the authors divide their dataset by randomly partitioning objects into training and testing splits. Thus the same image may appear in both splits. As only training and validation splits have been released for this dataset, we use the hyperparamters cross-validated on RefCOCO to train models on RefCOCOg.

Figure 2: Example comprehension results based on detection. Green box shows the ground-truth region, blue box shows correct comprehension using our "speaker+listener+reinforcer+MMI" model, and red box shows incorrect comprehension. We use top two rows to show some correct comprehensions and bottom two rows to show some incorrect ones.

		RefCOCO			RefCOCO+			RefCOCOg
		val	TestA	TestB	val	TestA	TestB	val
1	listener	77.48%	76.58%	78.94%	60.50%	61.39%	58.11%	71.12%
2	previous state-of-art[21][31]	76.90%	75.60%	78.00%[21]	58.94%	61.29%	56.24%[31]	65.32%[31]
3	baseline+MMI[19]	72.28%	72.60%	73.39%	56.66%	60.01%	53.15%	63.31%
4	**speaker**+MMI[31]	76.18%	74.39%	77.30%	58.94%	61.29%	56.24%	65.32%
5	**speaker**+listener+MMI	79.22%	77.78%	79.90%	61.72%	64.41%	58.62%	71.77%
6	**speaker**+reinforcer+MMI	78.38%	77.13%	79.53%	61.32%	63.99%	58.25%	67.06%
7	**speaker**+listener+reinforcer+MMI	**79.56%**	**78.95%**	**80.22%**	**62.26%**	**64.60%**	**59.62%**	**72.63%**
		RefCOCO (detected)			RefCOCO+ (detected)			RefCOCOg (detected)
		val	TestA	TestB	val	TestA	TestB	val
1	listener	-	71.63%	61.47%	-	57.33%	47.21%	56.18%
2	previous state-of-art[31]	-	72.03%	63.08%	-	58.87%	47.70%	58.26%
3	baseline+MMI[19]	-	68.73%	59.56%	-	58.15%	46.63%	57.23%
4	**speaker**+MMI[31]	-	72.03%	63.08%	-	58.87%	47.70%	58.26%
5	**speaker**+listener+MMI	-	**72.95%**	63.10%	-	60.23%	48.11%	58.57%
6	**speaker**+reinforcer+MMI	-	72.34%	63.24%	-	59.36%	48.72%	58.70%
7	**speaker**+listener+reinforcer+MMI	-	72.88%	**63.43%**	-	**60.43%**	**48.74%**	**59.51%**

Table 1: Ablation study using the speaker module for the comprehension task (indicated in **bold**). Top half shows performance given ground truth bounding boxes for objects, bottom half performance using automatic object detectors to select potential objects. We find that adding listener and reinforcer modules to the speaker increases performance.

4.3. Comprehension Task

After training, we can use either the speaker or listener to perform the comprehension task. For the speaker that models $P(r|o)$, we feed every ground-truth object region within the given image to the speaker and select the most probable region for the expression as the comprehension result, i.e., $o^* = \text{argmax}_{o_i} p(r|o_i)$. For the listener, we directly compute the similarity score $S(r, o)$ between the

		RefCOCO			RefCOCO+			RefCOCOg
		val	TestA	TestB	val	TestA	TestB	val
1	listener	77.48%	76.58%	78.94%	60.50%	61.39%	58.11%	71.12%
2	previous state-of-art [21][31]	76.90%	75.60%	78.00% [21]	58.94%	61.29%	56.24% [31]	65.32% [31]
3	speaker+**listener**+MMI	78.42%	78.45%	79.94%	61.48%	62.14%	58.91%	72.13%
4	speaker+**listener**+reinforcer+MMI	78.36%	77.97%	79.86%	61.33%	63.10%	58.19%	72.02%
5	**speaker+listener**+reinforcer+MMI (ensemble)	**80.36%**	**80.08%**	**81.73%**	**63.83%**	**65.40%**	**60.73%**	**74.19%**
		RefCOCO (detected)			RefCOCO+ (detected)			RefCOCOg (detected)
		val	TestA	TestB	val	TestA	TestB	val
1	listener	-	71.63%	61.47%	-	57.33%	47.21%	56.18%
2	previous state-of-art[31]	-	72.03%	63.08%	-	58.87%	47.70%	58.26%
3	speaker+**listener**+MMI	-	72.95%	62.43%	-	58.68%	48.44%	57.34%
4	speaker+**listener**+reinforcer+MMI	-	72.94%	62.98%	-	58.68%	47.68%	57.72%
5	**speaker+listener**+reinforcer+MMI (ensemble)	-	**73.78%**	**63.83%**	-	**60.48%**	**49.36%**	**59.84%**

Table 2: Ablation study using listener or ensembled listener+speaker modules for the comprehension task (indicated in **bold**). Top half shows performance given ground truth bounding boxes for objects, bottom half performance using automatic object detectors to select potential objects. We find that jointly training with the speaker improves listener's performance and that adding the reinforcer module in an ensembled speaker+listener prediction performs the best.

	RefCOCO				RefCOCO+				RefCOCOg	
	Test A		Test B		Test A		Test B		val	
	Meteor	CIDEr	Meteor	CIDEr	Meteor	CIDEr	Meteor	CIDEr	Meteor	CIDEr
speaker+tie [31]	0.283	0.681	0.320	1.273	0.204	0.499	0.196	0.683	-	-
baseline+MMI	0.243	0.615	0.300	1.227	0.199	0.462	0.189	0.679	0.149	0.585
speaker+MMI	0.260	0.679	0.319	1.276	0.202	0.475	0.196	0.683	0.147	0.573
speaker+listener+MMI	0.268	0.704	0.327	1.303	0.208	0.496	0.201	0.697	0.150	0.589
speaker+reinforcer+MMI	0.266	0.702	0.323	1.291	0.204	0.482	0.197	0.692	0.151	0.602
speaker+listener+reinforcer+MMI	0.268	0.697	0.329	**1.323**	0.204	0.494	0.202	0.709	0.154	0.592
speaker+listener+reinforcer+MMI+rerank	**0.296**	**0.775**	**0.340**	1.320	**0.213**	**0.520**	**0.215**	**0.735**	**0.159**	**0.662**

Table 3: Ablation study for generation using automatic evaluation.

proposal/object and expression and pick the object with the highest probability. For evaluation, we compute the intersection-over-union (IoU) of the comprehended region with the ground-truth object. If the IoU score of the predicted region is greater than 0.5, we consider this a correct comprehension.

To demonstrate the benefits of each module, we run ablation studies in Lines 4-7 of Table 1 (for speaker as comprehender) and in Lines 3-5 of Table 2 (for listener as comprehender) on all three datasets. Each row shows the results after adding a module during training. For some models that have both speaker and listener, we highlight the module being used for comprehension in bold. For example, "**speaker**+listener" means we use the speaker module of the joint model to do the comprehension task, while "speaker+**listener**" means we use the listener module for this task. Note our speaker module is implemented using the "visdif" model in [31] as mentioned in Section 3.1[2]. As the model trained with MMI on speaker outperforms w/o MMI as [19][31][21], we only show results with MMI in Table 12. We compare our models with the "baseline+MMI" model [19] (Line 3 in Table 1), the pure listener model (Line 1), and previous state-of-art results (Line 2) achieved in [21][31].

First, we evaluate the performance of the speaker on the comprehension task (Table 1). We find that the speaker can be improved by joint training with the listener module and by incorporating the reinforcer module. Note the speaker (Line 7) trained with full model is able to outperform the pure listener by around 2% on all three datasets, which already achieves state-of-art performance on the comprehension task.

Second, we show evaluations using variations of the listener module or ensembled listener+speaker modules (indicated in **bold**) for the comprehension task in Table 2. We note that the listener generally works better than speaker for the comprehension task, indicating that the deterministic joint-embedding model is more suitable for this task than the speaker model – similar results were observed in [23]. While the benefits from reinforcer seems not as effective as on the speaker, we still find that the joint training always brings additional discriminative benefits to the listener module, resulting in improved performance (compare Lines 3-5 with Line 1 in Table 2). Ensembling the speaker and listener together (Line 5) achieves the best results overall.

Both of the above experiments analyze comprehension performance given ground truth bounding boxes for potential comprehension objects, where the algorithm must select

[2]The "visdif" model trained in this paper performs slightly better on comprehension task than the original one reported in [31].

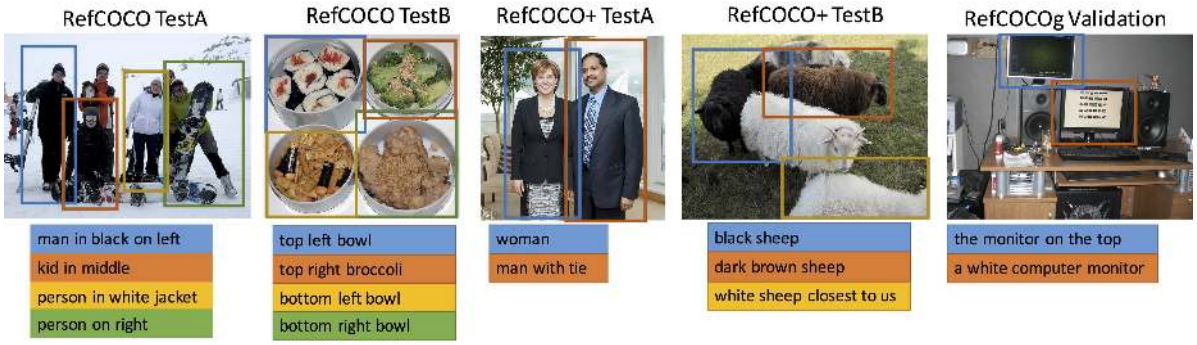

Figure 3: Joint generation examples using "speaker+listener+reinforcer+MMI+rerank". Each sentence shows the generated expression for one of the depicted objects (color coded to indicate correspondence)

	RefCOCO		RefCOCO+	
	Test A	Test B	Test A	Test B
speaker+tie[31]	71.40%	76.14%	57.17%	47.92%
speaker+MMI [31]	68.82%	75.50%	53.57%	47.88%
speaker+listener+MMI	73.23%	76.08%	53.83%	49.19%
speaker+reinforcer+MMI	71.08%	76.09%	55.16%	48.50%
speaker+listener+reinforcer+MMI	74.08%	76.44%	56.92%	53.23%
speaker+listener+reinforcer+MMI +rerank	**76.95%**	**78.10%**	**58.85%**	**58.20%**

Table 4: Human Evaluations on generation.

which of the objects is being described. This provides an analysis of comprehension performance independent of any particular object detection method. Additionally, we also show results using an object detector to automatically select regions for consideration during comprehension in the bottom half of each table (Tables 1 and 2). As our detection algorithm, we use current state of the art detector in effectiveness and speed, SSD [18], trained on a subset of the MS COCO train+val dataset, removing images that are in the test splits of RefCOCO or RefCOCO+ or in the validation split of RefCOCOg. We empirically select 0.30 as the confidence threshold for detection output. While performance drops somewhat due to the strong dependence of "visdif" model on detection [31], the overall improvements brought by each module are consistent with using ground-truth objects, showin the robustness of our joint model. Fig. 2 shows some comprehension results using our full model.

4.4. Generation Task

For the generation task, we evaluate variations on the speaker module. Evaluating the generation is not as simple as comprehension. In image captioning, BLEU, ROUGE, METEOR and CIDEr are common automatic metrics and have been widely used as standard evaluations. We show automatic evaluation using the METEOR and CIDEr metrics for generation in Table 3 where "+rerank" denotes models incorporating the reranking mechanism and global optimization over all objects (Eqn. 7). To computer CIDEr robustly, we collect more expressions for objects in the test sets for RefCOCO and RefCOCO+, obtaining 10.1 and 9.4 expressions respectively on average per object. For RefCOCOg we use the original expressions released with the dataset which may be limited, but we still show its performance for completeness. We choose the "speaker+tie" model in [31] as reference, which learns to tie the expression generation together and achieves state-of-art performance. Generally we find that the speaker in jointly learned models achieves higher scores than the single speaker under both metrics across datasets.

In addition, since previous work [31] has found that these metrics do not always agree well with human judgments for referring expressions, we also run a human evaluation on the same set of objects as [31] for RefCOCO and RefCOCO+. Here we ask Turkers to click on the referred object given a generated expression. These results are shown in Table. 4. Results indicate the ablated benefits brought by each module, and ultimately the "+rerank" of our joint model achieves the best performance.

We show the joint expression generation using our full model with "+rerank" in Fig. 3. As observed, the expressions of every target object are considered together, where each of them is meant to be relevant to the target object and irrelevant to the other objects.

5. Conclusion

We demonstrated the effectiveness of a unified framework for referring expression generation and comprehension. Our model consists of speaker and listener modules trained jointly to improve performance and a reinforcer module to help produce less ambiguous expressions. Experiments indicate that our model outperforms state of the art for both comprehension and generation on multiple datasets and evaluation metrics.

Acknowledgements: This research is supported by NSF Awards #1405822, 144234, 1562098, 1633295, NVidia, Google Research, and Microsoft Research.

References

[1] J. Andreas and D. Klein. Reasoning about pragmatics with neural listeners and speakers. *EMNLP*, 2016. 2, 5

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 2

[3] M. Eldon, D. Whitney, and S. Tellex. Interpreting multimodal referring expressions in real time. In *ICRA*, 2016. 1

[4] R. Fang, M. Doering, and J. Y. Chai. Embodied collaborative referring expression generation in situated human-robot interaction. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, 2015. 1

[5] N. FitzGerald, Y. Artzi, and L. S. Zettlemoyer. Learning distributions over logical forms for referring expression generation. In *EMNLP*, 2013. 2

[6] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, 2016. 4

[7] D. Golland, P. Liang, and D. Klein. A game-theoretic approach to generating spatial descriptions. In *EMNLP*, 2010. 1

[8] H. P. Grice. Logic and conversation. In P. Cole and J. L. Morgan, editors, *Syntax and Semantics: Vol. 3: Speech Acts*, pages 41–58. Academic Press, San Diego, CA, 1975. 1

[9] M. Grubinger, P. Clough, H. Müller, and T. Deselaers. The iapr tc-12 benchmark: A new evaluation resource for visual information systems. In *International Workshop OntoImage*, 2006. 2

[10] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. *CVPR*, 2016. 2

[11] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars. Guiding the long-short term memory model for image caption generation. In *ICCV*, 2015. 2

[12] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. *arXiv preprint arXiv:1511.07571*, 2015. 2

[13] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, pages 787–798, 2014. 1, 2

[14] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[15] E. Krahmer and K. Van Deemter. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218, 2012. 1, 2

[16] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016. 2

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. *ECCV*, 2016. 8

[19] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. *CVPR*, 2016. 1, 2, 3, 4, 6, 7

[20] M. Mitchell, K. Van Deemter, and E. Reiter. Generating expressions that refer to visible objects. In *HLT-NAACL*, 2013. 1, 2

[21] V. K. Nagaraja, V. I. Morariu, and L. S. Davis. Modeling context between objects for referring expression understanding. In *ECCV*, 2016. 2, 6, 7

[22] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015. 2

[23] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of textual phrases in images by reconstruction. *ECCV*, 2016. 2, 4, 7

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[25] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 1, 2

[26] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. *CVPR*, 2016. 2

[27] M. Wang, M. Azab, N. Kojima, R. Mihalcea, and J. Deng. Structured matching for phrase localization. *ECCV*, 2016. 2

[28] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992. 4

[29] T. Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972. 1, 2

[30] L. Yu, E. Park, A. C. Berg, and T. L. Berg. Visual madlibs: Fill in the blank description generation and question answering. In *ICCV*, 2015. 2

[31] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. *ECCV*, 2016. 1, 2, 3, 5, 6, 7, 8