# Generation and Comprehension of Unambiguous Object Descriptions

Junhua Mao[2*] Jonathan Huang[1] Alexander Toshev[1] Oana Camburu[3] Alan Yuille[2,4] Kevin Murphy[1]
[1]Google Inc. [2]University of California, Los Angeles [3]University of Oxford [4]Johns Hopkins University
{mjhustc@,yuille@stat.}ucla.edu, oana-maria.camburu@cs.ox.ac.uk
{jonathanhuang,toshev,kpmurphy}@google.com

## Abstract

*We propose a method that can generate an unambiguous description (known as a referring expression) of a specific object or region in an image, and which can also comprehend or interpret such an expression to infer which object is being described. We show that our method outperforms previous methods that generate descriptions of objects without taking into account other potentially ambiguous objects in the scene. Our model is inspired by recent successes of deep learning methods for image captioning, but while image captioning is difficult to evaluate, our task allows for easy objective evaluation. We also present a new large-scale dataset for referring expressions, based on MS-COCO. We have released the dataset and a toolbox for visualization and evaluation, see https://github.com/mjhucla/Google_Refexp_toolbox.*

## 1. Introduction

There has been a lot of recent interest in generating text descriptions of images (see e.g., [13, 53, 9, 5, 12, 26, 28, 40, 55, 8]). However, fundamentally this problem of image captioning is subjective and ill-posed. With so many valid ways to describe any given image, automatic captioning methods are thus notoriously difficult to evaluate. In particular, how can we decide that one sentence is a better description of an image than another?

In this paper, we focus on a special case of text generation given images, where the goal is to generate an unambiguous text description that applies to exactly one object or region in the image. Such a description is known as a "referring expression" [50, 52, 41, 42, 14, 19, 27]. This approach has a major advantage over generic image captioning, since there is a well-defined performance metric: a referring expression is considered to be good if it uniquely describes the relevant object or region within its context, such that a listener can comprehend the description and then recover the location of the original object. In addition, because of the discriminative nature of the task, referring expressions tend to be more detailed (and therefore more useful) than image captions. Finally, it is easier to collect training data to "cover" the space of reasonable referring expressions for a given object than it is for a whole image.

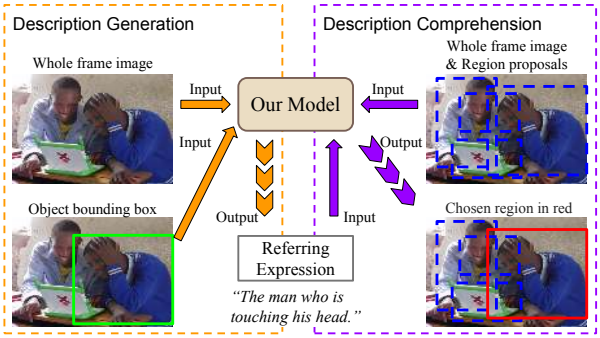


Figure 1. Illustration of our generation and comprehension system. On the left we see that the system is given an image and a region of interest; it describes it as "the man who is touching his head", which is unambiguous (unlike other possible expressions, such as "the man wearing blue", which would be unclear). On the right we see that the system is given an image, an expression, and a set of candidate regions (bounding boxes), and it selects the region that corresponds to the expression.

We consider two problems: (1) *description generation*, in which we must generate a text expression that uniquely pinpoints a highlighted object/region in the image and (2) *description comprehension*, in which we must automatically select an object given a text expression that refers to this object (see Figure 1). Most prior work in the literature has focused exclusively on description generation (e.g., [31, 27]). Golland *et al*. [19] consider generation and comprehension, but they do not process real world images.

In this paper, we jointly model both tasks of description generation and comprehension, using state-of-the-art deep learning approaches to handle real images and text. Specifically, our model is based upon recently developed methods that combine convolutional neural networks (CNNs) with recurrent neural networks (RNNs). We demonstrate that our model outperforms a baseline which generates referring expressions without regard to the listener who must comprehend the expression. We also show that our model can be trained in a semi-supervised fashion, by automatically generating descriptions for image regions.

Being able to generate and comprehend object descriptions is critical in a number of applications that use nat-

The major part of this work was done while J. Mao and O. Camburu were interns at Google Inc.

ural language interfaces, such as controlling a robot (e.g., "Rosie, please fetch me the beer from the top shelf of the fridge", cf. [4]), or interacting with photo editing software (e.g., "Picasa, please replace the third car behind the fence with a motorbike", cf. [6]). In addition, it is a good test bed for performing research in the area of vision and language systems because of the existence of a useful objective performance measure.

In order to train and evaluate our system, we have collected and released a new large scale referring expressions dataset based on the popular MS-COCO dataset [37].

To summarize, our main contributions are as follows. First, we present a new large scale dataset for referring expressions. Second, we evaluate how existing image captioning methods perform at the referring expression task. Third, we develop a new method for joint generation and comprehension that outperforms current methods.

## 2. Related Work

**Referring expressions.** Referring expression generation is a classic NLP problem (see e.g., [54, 31]). Important issues include understanding what types of attributes people typically use to describe visual objects (such as color and size) [42], usage of higher-order relationships (e.g., spatial comparison) [52], and the phenomena of over and under-specification, which is also related to speaker variance [14].

Context (sometimes called pragmatics [20]) plays a critical role in several ways [30]. First, the speaker must differentiate the target object from a collection of alternatives and must thus reason about how the object differs from its context. Second, the perception of the listener is also valuable. In particular, Golland *et al*. [19] recently proposed a game theoretic formulation of the referring expression problem showing that speakers that act optimally with respect to an explicit listener model naturally adhere to the Gricean Maxims of communication [22].

In most of this previous work, authors have focused on small datasets of computer generated objects (or photographs of simple objects) [50, 41] and have not connected their text generation systems to real vision systems. However there has been recent interest in understanding referring expressions in the context of complex real world images, for which humans tend to generate longer phrases [18]. [27] were the first to collect a large scale dataset of referring expressions for complex real world photos.

We likewise collect and evaluate against a large scale dataset. However we go beyond expression generation and jointly learn both generation and comprehension models. And where prior works have had to explicitly enumerate attribute categories such as size, color (e.g. [47]) or manually list all possible visual phrases (e.g. [46]), our deep learning-based models are able to learn to directly generate surface expressions from raw images without having to first convert to a formal object/attribute representation.

Concurrently, [24] propose a CNN-RNN based method that is similar to our baseline model and achieve state-of-the-art results on the ReferIt dataset [27]. But they did not use the discriminative training strategy proposed in our full model. [25, 32] investigate the task of generating dense descriptions in an image. But their descriptions are not required to be unambiguous.

**Image captioning.** Our methods are inspired by a long line of inquiry in joint models of images and text, primarily in the vision and learning communities [13, 23, 49, 43, 34, 56, 36]. From a modeling perspective, our approach is closest to recent works applying RNNs and CNNs to this problem domain [53, 9, 5, 12, 26, 28, 40, 55]. The main approach in these papers is to represent the image content using the hidden activations of a CNN, and then to feed this as input to an RNN, which is trained to generate a sequence of words.

Most papers on image captioning have focused on describing the full image, without any spatial localization. However, we are aware of two exceptions. [55] propose an attention model which is able to associate words to spatial regions within an image; however, they still focus on the full image captioning task. [26] propose a model for aligning words and short phrases within sentences to bounding boxes; they then train an model to generate these short snippets given features of the bounding box. Their model is similar to our baseline model, described in Section 5 (except we provide the alignment of phrases to boxes in the training set, similar to [45]). However, we show that this approach is not as good as our full model, which takes into account other potentially confusing regions in the image.

**Visual question answering.** Referring expressions is related to the task of VQA (see e.g., [2, 38, 39, 16, 15]). In particular, referring expression comprehension can be turned into a VQA task where the speaker asks a question such as "where in the image is the car in red?" and the system must return a bounding box (so the answer is numerical, not linguistic). However there are philosophical and practical differences between the two tasks. A referring expression (and language in general) is about *communication* — in our problem, the speaker is finding the optimal way to communicate to the listener, whereas VQA work typically focuses only on answering questions without regard to the listener's state of mind. Additionally, since questions tend to be more open ended in VQA, evaluating their answers can be as hard as with general image captioning, whereas evaluating the accuracy of a bounding box is easy.

## 3. Dataset Construction

The largest existing referring expressions dataset that we know of is the *ReferIt dataset*, which was collected by [27], and contains 130,525 expressions, referring to 96,654 distinct objects, in 19,894 photographs of natural scenes. Images in this dataset are from the segmented and annotated TC-12 expansion of the ImageCLEF IAPR dataset [11].

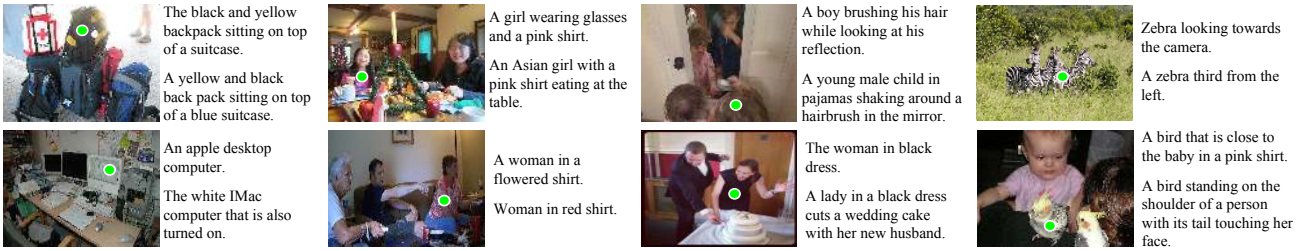


Figure 2. Some sample images from our Google Refexp (G-Ref) dataset. We use a green dot to indicate the object that the descriptions refer to. Since the dataset is based on MS COCO, we have access to the original annotations such as the object mask and category. Some of the objects are hard to describe, e.g., in the third image in the first row, we need to distinguish the boy from his reflection in the mirror.

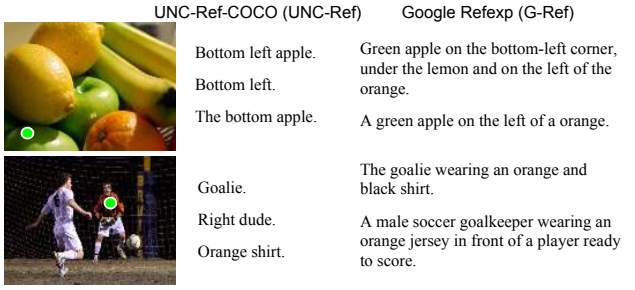


Figure 3. Comparison between the G-Ref and UNC-Ref dataset.

Two drawbacks of this dataset, however, are that (1) the images sometimes only contain one object of a given class, allowing speakers to use short descriptions without risking ambiguity, and (2) the ImageCLEF dataset focuses mostly on "stuff" (i.e. context) rather than "things" (i.e. objects).

In this paper, we use a similar methodology to that of [27], but building instead on top of the MSCOCO dataset [37], which contains more than 300,000 images, with 80 categories of objects segmented at the instance level.

For each image, we selected objects if (1) there are between 2 and 4 instances of the same object type within the same image, and (2) if their bounding boxes occupy at least 5% of image area. This resulted in selecting 54,822 objects from 26,711 images. We constructed a Mechanical Turk task in which we presented each object in each image (by highlighting the object mask) to a worker whose task was to generate a unique text description of this object. We then used a second task in which a different worker was presented with the image and description, and was asked to click inside the object being referred to. If the selected point was inside the original object's segmentation mask, we considered the description as valid, and kept it, otherwise we discarded it and re-annotated it by another worker. We repeated these description generation and verification tasks on Mechanical Turk iteratively up to three times. In this way, we selected 104,560 expressions. Each object has on average 1.91 expressions, and each image has on average 3.91 expressions. This dataset (released) is denoted as Google Refexp dataset and some samples are shown in Figure 2.

While we were collecting our dataset, we learned that Tamara Berg had independently applied her ReferIt game [27] to the MSCOCO dataset to generate expressions for 50,000 objects from 19,994 images. She kindly shared her data (named as UNC-Ref-COCO dataset) with us. For brevity, we call our Google Refexp dataset as *G-Ref* and the UNC-Ref-COCO as *UNC-ref*. We report results on both datasets in this paper. However, due to differences in our collection methodologies, we have found that the descriptions in the two overlapped datasets exhibit significant qualitative differences, with descriptions in the UNC-Ref dataset tending to be more concise and to contain less flowery language than our descriptions. [1] More specifically, the average lengths of expressions from our dataset and UNC-Ref are 8.43 and 3.61 respectively. And the size of the word dictionaries (keeping only words appearing more than 3 times) from our dataset and UNC-Ref are 4849 and 2890 respectively. See Figure 3 for some visual comparisons.

## 4. Tasks

In this section, we describe at a high level how we solve the two main tasks of description and generation. We will describe the model details and training in the next section.

### 4.1. Generation

In the *description generation task*, the system is given a full image and a target object (specified via a bounding box), and it must generate a referring expression for the target object. Formally, the task is to compute $\text{argmax}_S p(S|R, I)$, where $S$ is a sentence, $R$ is a region, and $I$ is an image.

Since we will use RNNs to represent $p(S|R, I)$, we can generate $S$ one word at a time until we generate an end of sentence symbol. Computing the globally most probable sentence is hard, but we can use beam search to approximately find the most probable sentences (we use a beam size of 3). This is very similar to a standard image captioning task, except the input is a region instead of a full image. The main difference is that we will train our model to generate descriptions that distinguish the input region from other candidate regions.

[1] According to our personal communication with the authors of the UNC-Ref dataset, the instruction and reward rule of UNC-Ref encourages the annotators to give a concise description in a limited time, while in our G-Ref dataset, we encourage the annotators to give rich and natural descriptions. This leads to different styles of annotations.

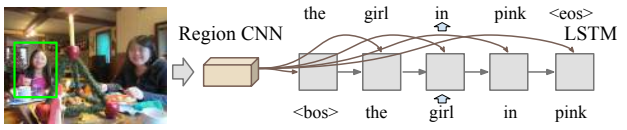


Figure 4. llustration of the baseline model architecture. ⟨bos⟩ and ⟨eos⟩ stand for beginning and end of sentence.

### 4.2. Comprehension

In the *description comprehension task*, we are given a full image and a referring expression and are asked to localize the the object being referred to within the image by returning a bounding box. One approach would be to train a model to directly predict the bounding box location given the referring expression (and image). However, in this paper, we adopt a simpler, ranking-based approach. In particular, we first generate a set $\mathcal{C}$ of region proposals, and then ask the system to rank these by probability. Then we select the region using $R^* = \text{argmax}_{R \in \mathcal{C}} p(R|S, I)$, where, by Bayes' rule, we have

$$p(R|S, I) = \frac{p(S|R, I)p(R|I)}{\sum_{R' \in \mathcal{C}} p(S|R', I)p(R'|I)}. \quad (1)$$

If we assume a uniform prior for $p(R|I)$,[2] we can select the region using $R^* = \text{argmax}_{R \in \mathcal{C}} p(S|R, I)$. This strategy is similar to image retrieval methods such as [29, 40], where the regions play the role of images.

At test time, we use the multibox method of [10] to generate objects proposals. This generates a large number of class agnostic bounding boxes. We then classify each box into one of the 80 MS-COCO categories, and discard those with low scores. We use the resulting post-classification boxes as the proposal set $\mathcal{C}$. To get an upper bound on performance, we also use the ground truth bounding boxes for all the objects in the image. In both cases, we do not use the label for the object of interest when ranking proposals.

## 5. The Baseline Method

In this section we explain our baseline method for computing $p(S|R, I)$.

### 5.1. Model Architecture

Our baseline model is similar to other image captioning models that use a CNN to represent the image, followed by an LSTM to generate the text (see e.g., [40, 9, 53]). The main difference is that we augment the CNN representation of the whole image with a CNN representation of the region of interest, in addition to location information. See Figure 4 for an illustration of our baseline model.

In more detail, we use VGGNet [48] as our CNN, pretrained on the ImageNet dataset [7, 33]. The last 1000 dimensional layer of VGGNet is used as our representation of the object region. In addition, we compute features for the whole image, to serve as context. In experiments, we only fine-tuned the weights for the last layer of the CNN and fixed all other layers. To feed a region to the CNN, we keep the aspect ratio of the region fixed and scale it to $224 \times 224$ resolution, padding the margins with the mean pixel value (similar to the region warping strategy in [17]). This gives us a 2000-dimensional feature, for the region and image.

[2] This implies that we are equally likely to choose any region to describe. This is approximately true by virtue of the way we constructed the dataset. However, in real applications, region saliency $p(R|I)$ should be taken into account.

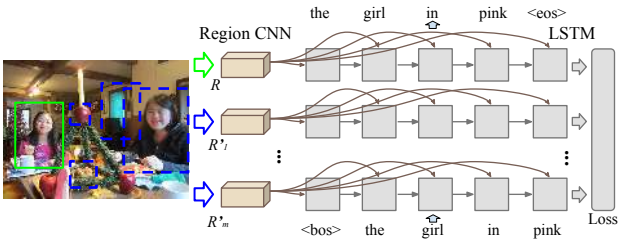


Figure 5. Illustration of how we train the full model using the softmax loss function. $R$ (green) is the target region, $R'$ are the incorrect regions. The weights of the LSTMs and CNNs are shared for $R$ and $R'$s. (Best viewed in color)

We encode the relative location and size of the region using a 5 dimensional vector as follows: $[\frac{x_{tl}}{W}, \frac{y_{tl}}{H}, \frac{x_{br}}{W}, \frac{y_{br}}{H}, \frac{S_{bbox}}{S_{image}}]$, where $(x_{tl}, y_{tl})$ and $(x_{br}, y_{br})$ are the coordinates of the top left and bottom right corners of the object bounding box, $H$ and $W$ are height and width of the image, and $S_{bbox}$ and $S_{image}$ are the sizes of the bounding box and image respectively.

Concatenating with the region, image, and location/size features, we obtain a 2005-dimensional vector which we feed as input into an LSTM model, which parameterizes the form of the distribution $p(S|R, I)$. For our LSTMs, we use a 1024-dimensional word-embedding space, and 1024-dimensional hidden state vector. We adopt the most commonly used vanilla LSTM structure [21] and feed the visual representation as input to the LSTM at each time step.

### 5.2. Maximum Likelihood Training

Our training data (discussed in Section 3) consists of observed triplets $(I, R, S)$, where $I$ is an image, $R$ denotes a region within $I$, and $S$ denotes a referring expression for $R$. To train the baseline model, we minimize the negative log probability of the referring expressions given their respective region and image:

$$J(\theta) = -\sum_{n=1}^{N} \log p(S_n|R_n, I_n, \theta), \quad (2)$$

where $\theta$ are the parameters of the RNN and CNN, and where we sum over the $N$ examples in the training set. We use ordinary stochastic gradient decent with a batch size of 16 and use an initial learning rate of 0.01 which is halved every 50,000 iterations. Gradient norms are clipped to a maximum value of 10. To combat overfitting, we regularize using dropout with a ratio of 0.5 for both the word-embedding and output layers of the LSTM.

## 6. The Full Method

The baseline method is to train the model to maximize $p(S|R, I)$, as is common for CNN-LSTM based image captioning models. However a strategy that directly generates an expression based only on the target object (which [19] calls the *reflex speaker* strategy) has the drawback that it may fail to generate discriminative sentences. For example, consider Figure 4: to generate a description of the girl highlighted by the green bounding box, generating the word "pink" is useful since it distinguishes this girl from the other girl on the right. To this end, we propose a modified training objective, described below.

### 6.1. Discriminative (MMI) Training

Section 5.2 proposed a way to train the model using maximum likelihood. We now propose the following alternative objective function:

$$J'(\theta) = -\sum_{n=1}^{N} \log p(R_n|S_n, I_n, \theta), \tag{3}$$

where

$$\log p(R_n|S_n, I_n, \theta) = \log \frac{p(S_n|R_n, I_n, \theta)}{\sum_{R' \in \mathcal{C}(I_n)} p(S_n|R', I_n, \theta)}. \tag{4}$$

We will call this the softmax loss. Note that this is the same as maximizing the mutual information between $S$ and $R$ (assuming a uniform prior for $p(R)$), since

$$\mathrm{MI}(S, R) = \log \frac{p(S, R)}{p(R)p(S)} = \log \frac{p(S|R)}{p(S)}. \tag{5}$$

where $p(S) = \sum_{R'} p(S|R')p(R') = \sum_{R'} p(S|R')$. Hence this approach is also called Maximum Mutual Information (MMI) training [3].

The main intuition behind MMI training is that we want to consider whether a listener would interpret the sentence unambiguously. We do this by penalizing the model if it thinks that a referring expression for a target object could also be plausibly generated by some other object within the same image. Thus given a training sample $(I, R, S)$, we train a model that outputs a high $p(S \mid R, I)$, while maintaining a low $p(S \mid R', I)$, whenever $R' \neq R$. Note that this stands in contrast to the Maximum Likelihood (ML) objective function in Equation 2 which directly maximizes $p(S|R)$ without considering other objects in the image.

There are several ways to select the region proposals $\mathcal{C}$. We could use all the true object bounding boxes, but this tends to waste time on objects that are visually very easy to discriminate from the target object (hence we call these *"easy ground truth negatives"*). An alternative is to select true object bounding boxes belonging to objects of the same class as the target object; these are more confusable (hence we call them *"hard ground truth negatives"*). Finally, we can use multibox proposals, the same as we use at test time, and select the ones with the same predicted object labels as $R$ (hence we call them *"hard multibox negatives"*). We will compare these different methods in Section 8.2. We use 5 random negatives at each step, so that all the data for a given image fits into GPU memory.

To optimize Equation 3, we must replicate the network (using tied weights) for each region $R' \in \mathcal{C}(I_n)$ (including the true region $R_n$), as shown in Figure 5. The resulting MMI trained model has exactly the same number of parameters as the ML trained model, and we use the same optimization and regularization strategy as in Section 5.2. Thus the only difference is the objective function.

For computational reasons, it is more convenient to use the following max-margin loss, which compares the target region $R$ against a single random negative region $R'$:

$$J''(\theta) = -\sum_{n=1}^{N} \{\log p(S_n|R_n, I_n, \theta) - \lambda \max(0,\ M - \log p(S_n|R_n, I_n, \theta) + \log p(S_n|R'_n, I_n, \theta))\} \tag{6}$$

This objective, which we call max-margin MMI (or MMI-MM) intuitively captures a similar effect as its softmax counterpart (MMI-SoftMax) and as we show in Section 8.2, yields similar results in practice. However, since it only compares two regions, the network must only be replicated twice. Consequently, less memory is used per sentence, allowing for more sentences to be loaded per minibatch which in turn helps in stabilizing the gradient.

## 7. Semi-supervised Training

Collecting referring expressions data can be expensive. In this section we discuss semi-supervised training of our full model by making use of bounding boxes that do not have descriptions, and thus are more ubiquitously available. Our main intuition for why a bounding box (region) $R$ can be useful even without an accompanying description is because it allows us to penalize our model during MMI training if it generates a sentence that it cannot itself decode to correctly recover $R$ (recall that MMI encourages $p(S|R, I)$ to be higher than $p(S|R', I)$, whenever $R' \neq R$).

In this semi-supervised setting, we consider a small dataset $D_{\mathrm{bb+txt}}$ of images with bounding boxes and descriptions, together with a larger dataset $D_{\mathrm{bb}}$ of images and bounding boxes, but without descriptions. We use $D_{\mathrm{bb+txt}}$ to train a model (which we call model $G$) to compute $p(S|R, I)$. We then use this model $G$ to generate a set of descriptions for the bounding boxes in $D_{\mathrm{bb}}$ (we

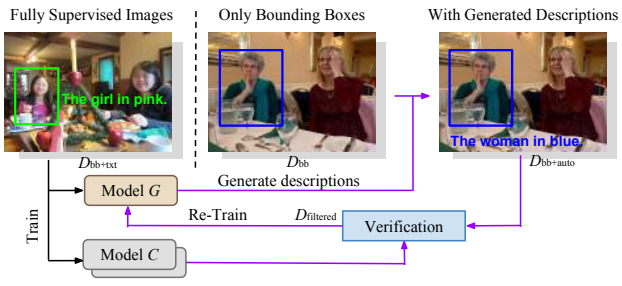


Figure 6. Ilustration of the semi-supervised training process.

call this new dataset $D_{\text{bb+auto}}$). We then retrain $G$ on $D_{\text{bb+txt}} \cup D_{\text{bb+auto}}$, in the spirit of bootstrap learning.

The above strategy suffers from the flaw that not all of the generated sentences are reliable, which may "pollute" the training set. To handle this, we train an ensemble of different models on $D_{\text{bb+txt}}$ (call them model $C$), and use these to determine which of the generated sentences for $D_{\text{bb+auto}}$ are trustworthy. In particular, we apply each model in the ensemble to decode each sentence in $D_{\text{bb+auto}}$, and only keep the sentence if every model maps it to the same correct object; we will call the resulting verified dataset $D_{\text{filtered}}$. This ensures that the generator creates referring expressions that can be understood by a variety of different models, thus minimizing overfitting. See Figure 6 for an illustration. In the experiments, we show that our model benefits from this semi-supervised training.

# 8. Experiments

We conducted experiments on both of the COCO referring expression datasets mentioned in Section 3: our G-Ref dataset and the UNC-Ref dataset. We randomly chose 5,000 objects as the validation set, 5,000 objects as the testing set and the remaining objects as the training set (44,822 for G-Ref and 40,000 for UNC-Ref).

## 8.1. Evaluation Metrics

In this section, we describe how we evaluate performance of the comprehension and generation tasks.

The comprehension task is easy to evaluate: we simply compute the Intersection over Union (IoU) ratio between the true and predicted bounding box. If IoU exceeds 0.5, we call the detection a true positive, otherwise it is a false positive (this is equivalent to computing the precision@1 measure). We then average this score over all images.

The generation task is more difficult — we can evaluate a generated description in the same way as an image description, using metrics such as CIDEr [51], BLEU [44] and METEOR [35]. However these metrics can be unreliable and do not account for semantic meaning. We rely instead on human evaluation, as was done in the most recent image captioning competition [1]. In particular, we asked Amazon Mechanical Turk (AMT) workers to compare an automatically generated object description to a human generated object description, when presented with an image and object of interest. The AMT workers do not know which sentences are human generated and which are computer generated (we do not even tell them that some sentences might be computer generated to reduce possible bias). We simply ask them to judge which sentence is a better description, or if they are equally good.

In addition to human evaluation, which does not scale, we evaluate our entire system by passing automatically generated descriptions to our comprehension system, and verifying that they get correctly decoded to the original object of interest. This end-to-end test is automatic and much more reliable than standard image captioning metrics.

| Proposals | GT | | Multibox | |
|---|---|---|---|---|
| Descriptions | GEN | GT | GEN | GT |
| ML (baseline) | 0.803 | 0.654 | 0.564 | 0.478 |
| MMI-MM-easy-GT-neg | 0.851 | 0.677 | 0.590 | 0.492 |
| MMI-MM-hard-GT-neg | **0.857** | **0.699** | 0.591 | 0.503 |
| MMI-MM-multibox-neg | 0.848 | 0.695 | **0.604** | **0.511** |
| MMI-SoftMax | 0.848 | 0.689 | 0.591 | 0.502 |

Table 1. We measure precision@1 on the UNC-Ref validation data. Each row is a different way of training the model. The columns show performance on ground truth or multibox proposals, and ground truth (human) or generated descriptions. Thus the columns with GT descriptions evaluate the performance of the comprehension system, and the columns with GEN descriptions evaluate (in an end-to-end way) the performance of the generation system.

## 8.2. Comparing different training methods

In this section, we compare different ways of training our model: maximum likelihood training (the baseline method); max-margin loss with easy ground truth negatives ("MMI-MM-easy-GT-neg"); max-margin loss with hard ground truth negatives ("MMI-MM-hard-GT-neg"); max-margin loss with hard multibox negatives ("MMI-MM-multibox-neg"); softmax/MMI loss with hard multibox negatives ("MMI-SoftMax"). For each method, we consider using either ground truth or multibox proposals at test time. In addition, we consider both ground truth descriptions and generated descriptions.

In this experiment we treat UNC-Ref as a validation set to explore various algorithmic options and hyperparameter settings for MMI. Only after having fixed these algorithmic options and hyperparameter settings did we do experiments on our G-Ref dataset (Section 8.3). This reduces the risk that we will have "overfit" our hyperparameters to each particular dataset. The results are summarized in Table 1 and we draw the following conclusions:

- All models perform better on generated descriptions than the groundtruth ones, possibly because the generated descriptions are shorter than the groundtruth (5.99 words on average vs 8.43), and/or because the generation and comprehension models share the same parameters, so that even if the generator uses a word incorrectly (e.g., describing a "dog" as a "cat"), the comprehension system can still decode it correctly. Intuitively, a model might "communicate" better with itself using its own language than with others.
- All the variants of the Full model (using MMI training) work better than the strong baseline using maximum likelihood training.
- The softmax version of MMI training is similar to the max-margin method, but slightly worse.
- MMI training benefits more from hard negatives than easy ones.

| Proposals | GT | | multibox | |
|---|---|---|---|---|
| Descriptions | GEN | GT | GEN | GT |
| G-Ref-Val | | | | |
| Baseline | 0.751 | 0.579 | 0.468 | 0.425 |
| Full Model | **0.799** | **0.607** | **0.500** | **0.445** |
| G-Ref-Test | | | | |
| Baseline | 0.769 | 0.545 | 0.485 | 0.406 |
| Full Model | **0.811** | **0.606** | **0.513** | **0.446** |
| UNC-Ref-Val | | | | |
| Baseline | 0.803 | 0.654 | 0.564 | 0.478 |
| Full Model | **0.848** | **0.695** | **0.604** | **0.511** |
| UNC-Ref-Test | | | | |
| Baseline | 0.834 | 0.643 | 0.596 | 0.477 |
| Full Model | **0.851** | **0.700** | **0.603** | **0.518** |

Table 2. Precision@1 for the baseline (ML) method and our full model with the max-margin objective function on various datasets.

- Training on ground truth negatives helps when using ground truth proposals, but when using multibox proposals (which is what we can use in practice), it is better to use multibox negatives.

Based on the above results, for the rest of the paper we will use max-margin training with hard multibox negatives as our Full Model.

### 8.3. Fully-supervised Training

In this section, we compare the strong baseline (maximum likelihood) with our max-margin MMI method on the validation and test sets from G-Ref and UNC-Ref. As before, we consider ground truth and multibox proposals at test time, and ground truth (human) or generated (automatic) descriptions. We see that MMI training outperforms ML training under every setting as shown in Table 2. [3]

In addition to the above end-to-end evaluation, we use human evaluators to judge generated sentence quality. In particular, we selected 1000 objects at random from our test set, and showed them to Amazon Mechanical Turk workers. The percentage of descriptions that are evaluated as better or equal to a human caption for the baseline and the full model are 15.9% and 20.4% respectively. This shows that MMI training is much better (4.5% absolute improvement, and 28.5% relative) than ML training.

### 8.4. Semi-supervised Training

To conduct the semi-supervised experiment, we separate the training set of our G-Ref dataset and the UNC-Ref dataset into two parts with the same number of objects. The first part (denoted by $D_{\text{bb+txt}}$) has the object description annotations while the second part (denoted by $D_{\text{bb}}$) only has object bounding boxes. Table 3 shows the results of semi-supervised training on the validation set of our dataset and UNC-Ref. We see that we get some improvement by training on $D_{\text{bb+txt}} \cup D_{\text{bb}}$ over just using $D_{\text{bb+txt}}$.

[3] We also train our baseline and full model on a random train, val, and test split w.r.t. to the images of our G-Ref dataset. The results are consistent with those in Table 2. With multibox proposals and GT descriptions, the Precision@1 of the baseline and full model are 0.404 and 0.444 on val set, and 0.407 and 0.451 on test set respectively.

| Proposals | GT | | multibox | |
|---|---|---|---|---|
| Descriptions | GEN | GT | GEN | GT |
| G-Ref | | | | |
| $D_{\text{bb+txt}}$ | 0.791 | 0.561 | **0.489** | 0.417 |
| $D_{\text{bb+txt}} \cup D_{\text{bb}}$ | **0.793** | **0.577** | **0.489** | **0.424** |
| UNC-Ref | | | | |
| $D_{\text{bb+txt}}$ | 0.826 | 0.655 | 0.588 | 0.483 |
| $D_{\text{bb+txt}} \cup D_{\text{bb}}$ | **0.833** | **0.660** | **0.591** | **0.486** |

Table 3. Performance of our full model when trained on a small strongly labeled dataset vs training on a larger dataset with automatically labeled data.

### 8.5. Qualitative Results

In Figure 7 we show qualitative results of our full generation model (above the dashed line) and the baseline generation model (below the dashed line) on some of our test images. We see that the descriptions generated by our full model are typically longer and more discriminative than the baseline model. In the second image, for example, the baseline describes one of the cats as "a cat laying on a bed", which is not sufficiently unambiguous for a listener to understand which cat is being described. Our full model, on the other hand, describes the same cat as "a cat laying on the left" which is completely unambiguous.

Figure 8 shows some qualitative results of our full comprehension model on our test dataset. The first and second columns show the original image and the multibox proposals respectively. The last four columns show the bounding boxes (denoted as a red bounding box in the figure) selected by our full model in response to different input sentences (both ground truth sentences and ones we created to probe the comprehension abilities of the model). To better interpret these results, we also show the bounding boxes that are within the margin of the model (see Eqn. 6) with dashed blue bounding boxes. Their bounding boxes are considered as "possible candidates" but their scores (i.e. $p(S|R, I)$) are not as high as the chosen one.

In general, we see that the comprehension model does quite well from short two word phrases to longer descriptions. It is able to respond correctly to single word changes in a referring expression (e.g., "the man in black" to "the man in red"). It also correctly identifies that the horse is the referent of the expression "a dark horse carrying a woman" whereas the woman is the referent in "a woman on the dark horse" — note that methods that average word embeddings would most likely fail on this example. However, there are also failure cases. E.g., in the fifth row, "the woman in white" selects a woman in black; this is because our model cannot handle the case where the object is not present, although it makes a reasonable guess. Also, in the fifth row, "the controller in the woman's hand" selects the woman, the orange juice and the controller, since this particular kind of

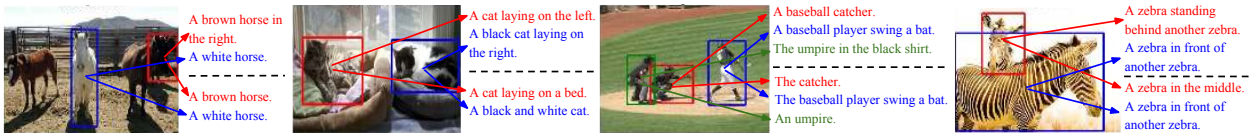


Figure 7. The sample results of the description generation using our full model (above the dashed line) and the strong baseline (below the dashed line). The descriptions generated by our full model are more discriminative than those generated by the baseline.

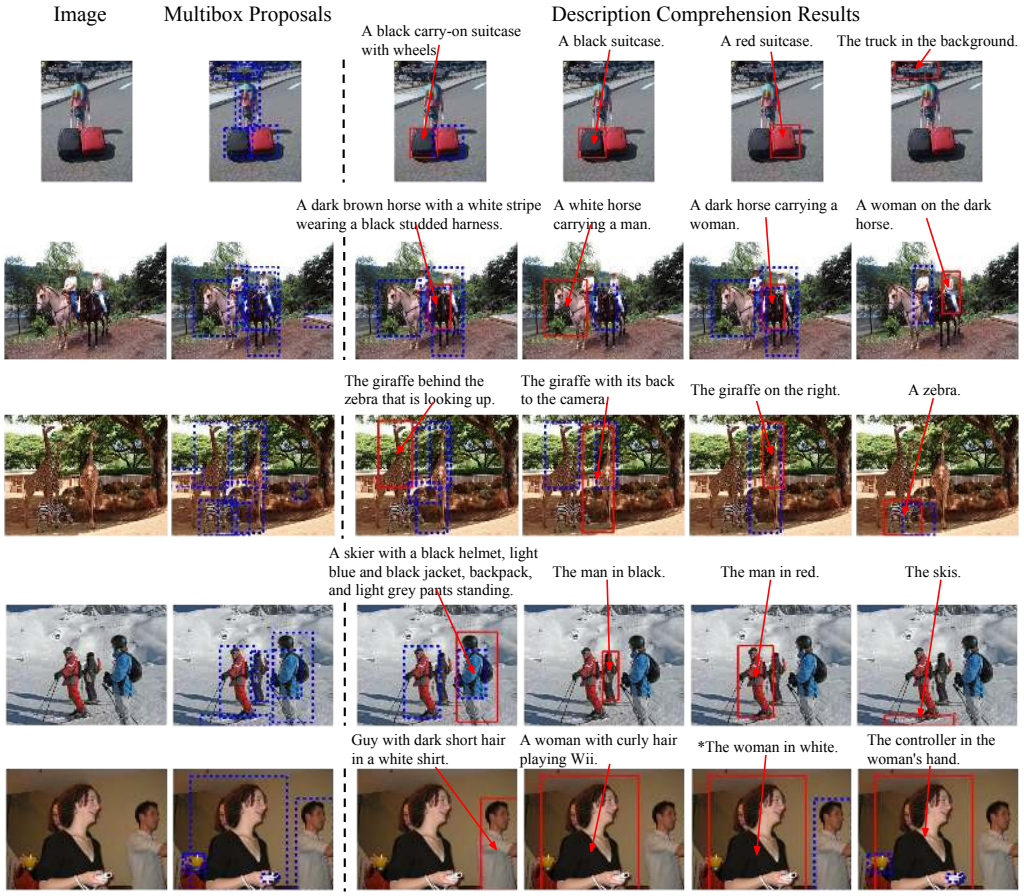


Figure 8. Sample results of the description comprehension task using our full model. The first and second column shows the original image and the multibox proposals. The third to sixth columns show the results of our model when input an arbitrary description of an object in the image. The red bounding box denotes the most probable object predicted by the model while the blue dashed ones denote the bounding boxes within the margin of the most probable one. The descriptions can be the groundtruth ones in the dataset (third column) or an customized descriptions (fourth to sixth columns). (Best viewed in color)

object is too small to detect, and lacks enough training data.

## 9. Conclusions

To conclude, we leave the reader with two simple points. First, referring expressions have been studied for decades, but in light of the recent burst of interest in image captioning, referring expressions take on new importance. Where image captioning itself is difficult to evaluate, referring expressions have an objective performance metric, and require the same semantic understanding of language and vision. Thus success on datasets such as the one contributed in this paper is more meaningful than success by standard image captioning metrics.

Second, to be successful at generating descriptions, we must *consider the listener*. Our experiments show that modeling a listener that must correctly decode a generated description consistently outperforms a model that simply emits captions based on region features. We hope that in addition to our dataset, these insights will spur further progress on joint models of vision and language.

**Acknowledgement** We are grateful to Tamara Berg for sharing the UNC-Ref-COCO dataset. We also thank Sergio Guadarrama, Vivek Rathod, Vignesh Ramanathan, Nando de Freitas, Rahul Sukthankar, Oriol Vinyals and Samy Bengio for early discussions and feedback on drafts. This work was partly supported by ARO 62250-CS, the NSF Center for Brains, Minds, and Machines, and NSF STC award CCF-1231216.

## References


[1] Ms coco captioning challenge. http://mscoco.org/dataset/#captions-challenge2015. 6

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

[3] L. Bahl, P. Brown, P. V. de Souza, and R. Mercer. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *ICASSP*, volume 11, pages 49–52, Apr. 1986. 5

[4] D. P. Barrett, S. A. Bronikowski, H. Yu, and J. M. Siskind. Robot language learning, generation, and comprehension. *arXiv preprint arXiv:1508.06161*, 2015. 2

[5] X. Chen and C. L. Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *CVPR*, 2015. 1, 2

[6] M.-M. Cheng, S. Zheng, W.-Y. Lin, V. Vineet, P. Sturgess, N. Crook, N. J. Mitra, and P. Torr. ImageSpirit: Verbal guided image parsing. *ACM Trans. Graphics*, 2014. 2

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 4

[8] J. Devlin, S. Gupta, R. Girshick, M. Mitchell, and C. L. Zitnick. Exploring nearest neighbor approaches for image captioning. *arXiv preprint arXiv:1505.04467*, 2015. 1

[9] J. Donahue, L. A. 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. 1, 2, 4

[10] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *CVPR*, pages 2155–2162, 2014. 4

[11] H. J. Escalante, C. A. Hernandez, J. A. Gonzalez, A. Lopez-Lopez, M. Montes, E. F. Morales, L. E. Sucar, L. Villasenor, and M. Grubinger. The segmented and annotated IAPR TC-12 benchmark. *CVIU*, 2010. 2

[12] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. Platt, et al. From captions to visual concepts and back. In *CVPR*, 2015. 1, 2

[13] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*, pages 15–29. 2010. 1, 2

[14] N. FitzGerald, Y. Artzi, and L. S. Zettlemoyer. Learning distributions over logical forms for referring expression generation. In *EMNLP*, pages 1914–1925, 2013. 1, 2

[15] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu. Are you talking to a machine? dataset and methods for multilingual image question answering. In *NIPS*, 2015. 2

[16] D. Geman, S. Geman, N. Hallonquist, and L. Younes. Visual turing test for computer vision systems. *PANS*, 112(12):3618–3623, 2015. 2

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 4

[18] D. Gkatzia, V. Rieser, P. Bartie, and W. Mackaness. From the virtual to the real world: Referring to objects in Real-World spatial scenes. In *EMNLP*, 2015. 2

[19] D. Golland, P. Liang, and D. Klein. A game-theoretic approach to generating spatial descriptions. In *EMNLP*, pages 410–419, 2010. 1, 2, 5

[20] N. D. Goodman and D. Lassiter. Probabilistic semantics and pragmatics: Uncertainty in language and thought. *Handbook of Contemporary Semantic Theory. Wiley-Blackwell*, 2014. 2

[21] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. Lstm: A search space odyssey. In *ICML*, 2015. 4

[22] H. P. Grice. *Logic and conversation*. na, 1970. 2

[23] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *JAIR*, 47:853–899, 2013. 2

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

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

[26] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *arXiv preprint arXiv:1412.2306*, 2014. 1, 2

[27] 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, 3

[28] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014. 1, 2

[29] R. Kiros, R. Zemel, and R. Salakhutdinov. Multimodal neural language models. In *ICML*, 2014. 4

[30] E. Krahmer and M. Theune. Efficient context-sensitive generation of referring expressions. *Information sharing: Reference and presupposition in language generation and interpretation*, 143:223–263, 2002. 2

[31] E. Krahmer and K. van Deemter. Computational generation of referring expressions: A survey. *Comp. Linguistics*, 38, 2012. 1, 2

[32] 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

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 4

[34] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating image descriptions. In *CVPR*, 2011. 2

[35] A. Lavie and A. Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgements. In *Workshop on Statistical Machine Translation*, pages 228–231, 2007. 6

[36] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi. Composing simple image descriptions using web-scale n-grams. In *CoNLL*, pages 220–228, 2011. 2

[37] 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. 2, 3

[38] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, pages 1682–1690, 2014. 2

[39] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *NIPS*, 2015. 2

[40] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). In *ICLR*, 2015. 1, 2, 4

[41] M. Mitchell, K. van Deemter, and E. Reiter. Natural reference to objects in a visual domain. In *INLG*, pages 95–104, 2010. 1, 2

[42] M. Mitchell, K. van Deemter, and E. Reiter. Generating expressions that refer to visible objects. In *HLT-NAACL*, pages 1174–1184, 2013. 1, 2

[43] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011. 2

[44] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002. 6

[45] 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

[46] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011. 2

[47] A. Sadovnik, Y.-I. Chiu, N. Snavely, S. Edelman, and T. Chen. Image description with a goal: Building efficient discriminating expressions for images. In *CVPR*, 2012. 2

[48] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4

[49] R. Socher, Q. Le, C. Manning, and A. Ng. Grounded compositional semantics for finding and describing images with sentences. In *TACL*, 2014. 2

[50] K. van Deemter, I. van der Sluis, and A. Gatt. Building a semantically transparent corpus for the generation of referring expressions. In *INLG*, pages 130–132, 2006. 1, 2

[51] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. CIDEr: Consensus-based image description evaluation. In *CVPR*, 2015. 6

[52] J. Viethen and R. Dale. The use of spatial relations in referring expression generation. In *INLG*, pages 59–67. Association for Computational Linguistics, 2008. 1, 2

[53] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 1, 2, 4

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

[55] K. Xu, J. Ba, R. Kiros, C. A. Cho, Kyunghyun, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 1, 2

[56] Y. Yang, C. L. Teo, H. Daumé III, and Y. Aloimonos. Corpus-guided sentence generation of natural images. In *EMNLP*, pages 444–454, 2011. 2