Neural Baby Talk

Jiasen Lu[1*] Jianwei Yang[1*] Dhruv Batra[1,2] Devi Parikh[1,2]
[1]Georgia Institute of Technology [2]Facebook AI Research
{jiasenlu, jw2yang, dbatra, parikh}@gatech.edu

Abstract

We introduce a novel framework for image captioning that can produce natural language explicitly grounded in entities that object detectors find in the image. Our approach reconciles classical slot filling approaches (that are generally better grounded in images) with modern neural captioning approaches (that are generally more natural sounding and accurate). Our approach first generates a sentence 'template' with slot locations explicitly tied to specific image regions. These slots are then filled in by visual concepts identified in the regions by object detectors. The entire architecture (sentence template generation and slot filling with object detectors) is end-to-end differentiable. We verify the effectiveness of our proposed model on different image captioning tasks. On standard image captioning and novel object captioning, our model reaches state-of-the-art on both COCO and Flickr30k datasets. We also demonstrate that our model has unique advantages when the train and test distributions of scene compositions – and hence language priors of associated captions – are different. Code has been made available at: https://github.com/jiasenlu/NeuralBabyTalk.

1. Introduction

Image captioning is a challenging problem that lies at the intersection of computer vision and natural language processing. It involves generating a natural language sentence that accurately summarizes the contents of an image. Image captioning is also an important first step towards real-world applications with significant practical impact, ranging from aiding visually impaired users to personal assistants to human-robot interaction [5, 9].

State-of-art image captioning models today tend to be monolithic neural models, essentially of the "encoder-decoder" paradigm. Images are encoded into a vector with a convolutional neural network (CNN), and captions are decoded from this vector using a Recurrent Neural Network (RNN), with the entire system trained end-to-end. While there are many recent extensions of this basic idea to include attention [45, 12, 49, 47, 27], it is well-understood that models still lack visual grounding (*i.e*., do not associate named concepts to pixels in the image). They often tend to 'look' at different regions than humans would and tend to copy captions from training data [8].

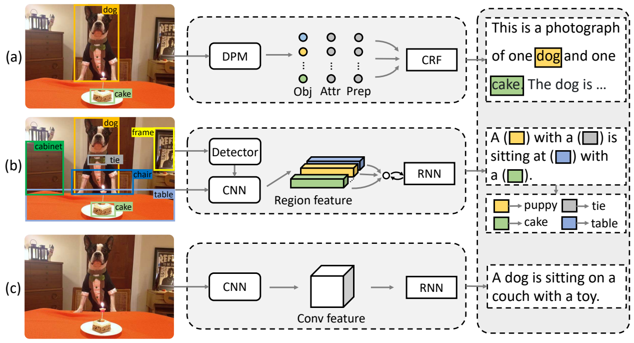

(a)
dog
cake
DPM
Obj
Attr
Prep
CRF
This is a photograph of one dog and one cake. The dog is ...
(b)
dog
frame
cabinet
tie
chair
table
cake
Detector
CNN
Region feature
RNN
A (🟨) with a (⬜) is sitting at (🟦) with a (🟩).
🟨→ puppy
⬜→ tie
🟩→ cake
🟦→ table
(c)
CNN
Conv feature
RNN
A dog is sitting on a couch with a toy.

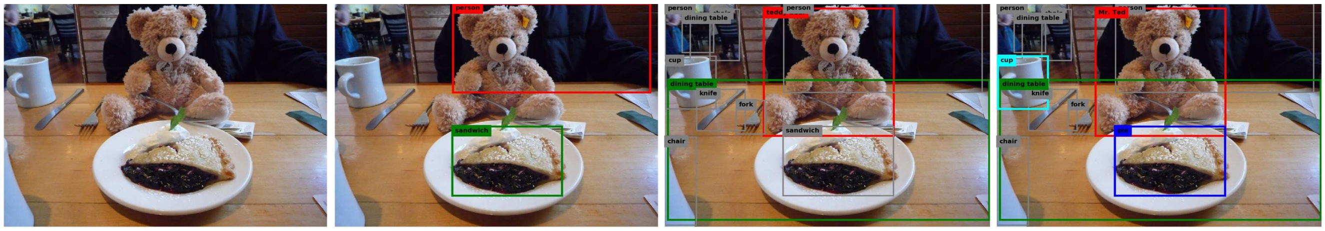
Figure 1. Example captions generated by (a) Baby Talk [24], (c) neural image captioning [20] and (b) our Neural Baby Talk approach. Our method generates the sentence "template" with slot locations (illustrated with filled boxes) explicitly tied to image regions (drawn in the image in corresponding colors). These slots are then filled by object detectors with concepts found in regions.

For instance, in Fig. 1 a neural image captioning approach [20] describes the image as "A dog is sitting on a couch with a toy." This is not quite accurate. But if one were to *really* squint at the image, it (arguably) does perhaps look like a scene where a dog *could* be sitting on a couch with a toy. It certainly is common to find dogs sitting on couches with toys. A-priori, the description is reasonable. That's exactly what today's neural captioning models tend to do – produce generic *plausible* captions based on the language model[1] that match a first-glance gist of the scene. While this may suffice for common scenes, images that differ from canonical scenes – given the diversity in our visual world, there are *plenty* of such images – tend to be underserved by these models.

If we take a step back – do we really need the language model to do the heavy lifting in image captioning? Given

[*] Equal contribution

[1] frequently, directly reproduced from a caption in the training data.



A close up of a stuffed animal on a plate.

A person is sitting at a table with a sandwich.

A teddy bear sitting on a table with a plate of food.

A Mr. Ted sitting at a table with a pie and a cup of coffee.

Figure 2. From left to right is the generated caption using the same captioning model but with different detectors: 1) No detector; 2) A weak detector that only detects "person" and "sandwich"; 3) A detector trained on COCO [26] categories (including "teddy bear"). 4) A detector that can detect novel concepts (e.g. "Mr. Ted" and "pie" that never occurred in the captioning training data). Different colors show a correspondence between the visual word and grounding regions.

the unprecedented progress we are seeing in object recognition[2] (e.g., object detection, semantic segmentation, instance segmentation, pose estimation), it seems like the vision pipeline can certainly do better than rely on just a first-glance gist of the scene. In fact, today's state-of-the-art object detectors can successfully detect the table and cake in the image in Fig. 1(c)! The caption ought to be able to talk about the table and cake *actually detected* as opposed to letting the language model hallucinate a couch and a toy simply because that sounds plausible.

Interestingly, some of the first attempts at image captioning [13, 24, 25, 33] – before the deep learning "revolution" – relied heavily on outputs of object detectors and attribute classifiers to describe images. For instance, consider the output of Baby Talk [24] in Fig. 1, that used a slot filling approach to talk about all the objects and attributes found in the scene via a templated caption. The language is unnatural but the caption is very much grounded in what the model sees in the image. Today's approaches fall at the other extreme on the spectrum – the language generated by modern neural image captioning approaches is much more natural but tends to be much less grounded in the image.

In this paper, we introduce Neural Baby Talk that reconciles these methodologies. It produces natural language *explicitly* grounded in entities found by object detectors. It is a neural approach that generates a sentence "template" with slot locations explicitly tied to image regions. These slots are then filled by object recognizers with concepts found in the regions. The entire approach is trained end-to-end. This results in natural sounding and grounded captions.

Our main technical contribution is a novel neural decoder for grounded image captioning. Specifically, at each time step, the model decides whether to generate a word from the textual vocabulary or generate a "visual" word. The visual word is essentially a token that will hold the slot for a word that is to describe a specific region in the image. For instance, for the image in Fig. 1, the generated sequence may be "A <region−17> is sitting at a <region−123> with a <region−3>." The visual words (<region−[.]>'s) are then filled in during a second stage that classifies each of the indicated regions (e.g., <region−17>→puppy, <region−123>→table), resulting in a final description of "A puppy is sitting at a table with a cake." – a free-form natural language description that is grounded in the image. One nice feature of our model is that it allows for different object detectors to be plugged in easily. As a result, a variety of captions can be produced for the same image using different detection backends. See Fig. 2 for an illustration.

Contributions: Our contributions are as follows:

- We present Neural Baby Talk – a novel framework for visually grounded image captioning that explicitly localizes objects in the image while generating free-form natural language descriptions.
- Ours is a two-stage approach that first generates a hybrid template that contains a mix of (text) words and slots explicitly associated with image regions, and then fills in the slots with (text) words by recognizing the content in the corresponding image regions.
- We propose a robust image captioning task to benchmark compositionality of image captioning algorithms where at test time the model encounters images containing known objects but in novel combinations (e.g., the model has seen dogs on couches and people at tables during training, but at test time encounters a dog at a table). Generalizing to such novel compositions is one way to demonstrate image grounding as opposed to simply leveraging correlations from training data.
- Our proposed method achieves state-of-the-art performance on COCO and Flickr30k datasets on the standard image captioning task, and significantly outperforms existing approaches on the robust image captioning and novel object captioning tasks.

2. Related Work

Some of the earlier approaches generated templated image captions via slot-filling. For instance, Kulkarni *et*

[2] e.g., 11% absolute increase in average precision in object detection in the COCO challenge in the last year.

al. [24] detect objects, attributes, and prepositions, jointly reason about these through a CRF, and finally fill appropriate slots in a template. Farhadi *et al.* [13] compute a triplet for a scene, and use this templated "meaning" representation to retrieve a caption from a database. [25, 33] use more powerful language templates such as a syntactically well-formed tree. These approaches tend to either produce captions that are relevant to the image but not natural sounding, or captions that are natural (*e.g.* retrieved from a database of captions) but may not be sufficiently grounded in the image.

Neural models for image captioning have been receiving increased attention in the last few years [23, 32, 7, 44, 11, 20]. State-of-the-art neural approaches include attention mechanisms [45, 12, 49, 47, 27, 38, 3] that identify regions in the image to "ground" emitted words. In practice, these attention regions tend to be quite blurry, and rarely correspond to semantically meaningful individual entities (e.g., objects instances) in the image. Our approach grounds words in object detections, which by design identify concrete semantic entities (object instances) in the image.

There has been some recent interest in grounding natural language in images. Dense Captioning [19] generates descriptions for specific image regions. In contrast, our model produces captions for the entire image, with words grounded in concrete entities in the image. Another related line of work is on resolving referring expressions [21] (or description-based object retrieval [36, 17, 18, 39] – given a description of an object in the image, identify which object is being referred to) or referring expression generation [21, 29, 31, 50] (given an object in the image, generate a discriminative description of the object). While the interest in grounded language is in common, our task is different.

One natural strength of our model is its ability to incorporate different object detectors, including the ability to generate captions with novel objects (never seen before in training captions). In that context, our work is related to prior works on novel object captioning [4, 42, 48, 2]. As we describe in Sec. 4.3, our method outperforms these approaches by 14.6% on the averaged F1 score.

3. Method

Given an image $\boldsymbol{I}$, the goal of our method is to generate visually grounded descriptions $\boldsymbol{y} = \{y_1, \ldots, y_T\}$. Let $\boldsymbol{r_I} = \{r_1, ..., r_N\}$ be the set of N images regions extracted from $\boldsymbol{I}$. When generating an entity word in the caption, we want to ground it in a specific image region $r \in \boldsymbol{r_I}$. Following the standard supervised learning paradigm, we learn parameters $\boldsymbol{\theta}$ of our model by maximizing the likelihood of the correct caption:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \sum_{(\boldsymbol{I},\boldsymbol{y})} \log p(\boldsymbol{y}|\boldsymbol{I};\boldsymbol{\theta}) \tag{1}$$

Using chain rule, the joint probability distribution can be decomposed over a sequence of tokens:

$$p(\boldsymbol{y}|\boldsymbol{I}) = \prod_{t=1}^{T} p(y_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{I}) \tag{2}$$

where we drop the dependency on model parameters to avoid notational clutter. We introduce a latent variable r_t to denote a specific image region so that y_t can explicitly ground in it. Thus the probability of y_t is decomposed to:

$$p(y_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{I}) = p(y_t|r_t, \boldsymbol{y}_{1:t-1}, \boldsymbol{I})p(r_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{I}) \tag{3}$$

In our framework, y_t can be of one of two types: a visual word or a textual word, denoted as y^{vis} and y^{txt} respectively. A visual word y^{vis} is a type of word that is grounded in a specific image region drawn from $\boldsymbol{r_I}$. A textual word y^{txt} is a word from the remainder of the caption. It is drawn from the language model , which is associated with a "default" sentinel "region" $\tilde{r}$ obtained from the language model [27] (discussed in Sec. 3.1). For example, as illustrated in Fig. 1, "puppy" and "cake" grounded in the bounding box of category "dog" and "cake" respectively, are visual words. While "with" and "sitting" are not associated with any image regions and thus are textual words.

With this, Eq. 1 can be decomposed into two cascaded objectives. First, maximizing the probability of generating the sentence "template". A sequence of grounding regions associated with the visual words interspersed with the textual words can be viewed as a sentence "template", where the grounding regions are slots to be filled in with visual words.[3] An example template (Fig. 3) is "A <region−2> is laying on the <region−4> near a <region−7>. Second, maximizing the probability of visual words y^{vis}_t conditioned on the grounding regions and object detection information, e.g., categories recognized by detector. In the template example above, the model will fill the slots with 'cat', 'laptop' and 'chair' respectively.

In the following, we first describe how we generate the slotted caption template (Sec. 3.1), and then how the slots are filled in to obtain the final image description (Sec. 3.2). The overall objective function is described in Sec. 3.3 and the implementation details in Sec. 3.4.

3.1. "Slotted" Caption Template Generation

Given an image $\boldsymbol{I}$, and the corresponding caption $\boldsymbol{y}$, the candidate grounding regions are obtained by using a pre-trained Faster-RCNN network [37]. To generate the caption "template", we use a recurrent neural network, which is commonly used as the decoder for image captioning [32, 44]. At each time step, we compute the RNN hidden state $\boldsymbol{h}_t$ according to the previous hidden state $\boldsymbol{h}_{t-1}$ and the input $\boldsymbol{x}_t$ such that $\boldsymbol{h}_t = \text{RNN}(\boldsymbol{x}_t, \boldsymbol{h}_{t-1})$. At training time,

[3] Our approach is not limited to any pre-specified bank of templates. Rather, our approach automatically generates a template (with placeholders – slots – for visually grounded words), which may be any one of the exponentially many possible templates.

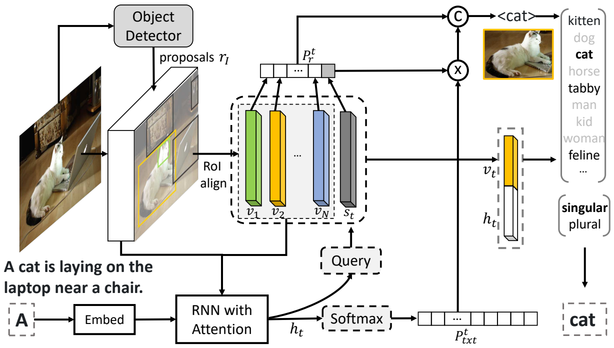


Figure 3. One block of the proposed approach. Given an image, proposals from any object detector and current word "A", the figure shows the process to predict the next visual word "cat".

x_t is the ground truth token (teacher forcing) and at test time is the sampled token y_{t-1}. Our decoder consists of an attention based LSTM layer [38] that takes convolution feature maps as input. Details can be found in Sec. 3.4. To generate the "slot" for visual words, we use a pointer network [43] that modulates a content-based attention mechanism over the grounding regions. Let $\boldsymbol{v_t} \in \mathcal{R}^{d\times 1}$ be the region feature of r_t, which is calculated based on Faster R-CNN. We compute the pointing vector with:

$$u_i^t = \boldsymbol{w}_h^T \tanh(\boldsymbol{W}_v \boldsymbol{v}_t + \boldsymbol{W}_z \boldsymbol{h}_t) \tag{4}$$

$$\boldsymbol{P}_{\boldsymbol{r_I}}^t = \text{softmax}(\boldsymbol{u}^t) \tag{5}$$

where $\boldsymbol{W}_v \in \mathbb{R}^{m\times d}$, $\boldsymbol{W}_z \in \mathbb{R}^{d\times d}$ and $\boldsymbol{w}_h \in \mathbb{R}^{d\times 1}$ are parameters to be learned. The softmax normalizes the vector $\boldsymbol{u}^t$ to be a distribution over grounding regions $\boldsymbol{r_I}$.

Since textual words y_t^{txt} are not tied to specific regions in the image, inspired by [27], we add a "visual sentinel" $\tilde{r}$ as a latent variable to serve as dummy grounding for the textual word. The visual sentinel can be thought of as a latent representation of what the decoder already knows about the image. The probability of a textual word y_t^{txt} then is:

$$p(y_t^{txt}|\boldsymbol{y}_{1:t-1}) = p(y_t^{txt}|\tilde{r}, \boldsymbol{y}_{1:t-1})p(\tilde{r}|\boldsymbol{y}_{1:t-1}) \tag{6}$$

where we drop the dependency on $\boldsymbol{I}$ to avoid clutter.

We first describe how the visual sentinel is computed, and then how the textual words are determined based on the visual sentinel. Following [27], when the decoder RNN is an LSTM [16], the representation for visual sentinel $\boldsymbol{s}_t$ can be obtained by:

$$\boldsymbol{g}_t = \sigma\left(\boldsymbol{W}_x \boldsymbol{x}_t + \boldsymbol{W}_h \boldsymbol{h}_{t-1}\right) \tag{7}$$

$$\boldsymbol{s}_t = \boldsymbol{g}_t \odot \tanh\left(\boldsymbol{c}_t\right) \tag{8}$$

where $\boldsymbol{W}_x \in \mathbb{R}^{d\times d}$, $\boldsymbol{W}_h \in \mathbb{R}^{d\times d}$. $\boldsymbol{x}_t$ is the LSTM input at time step t, and $\boldsymbol{g}_t$ is the gate applied on the cell state $\boldsymbol{c}_t$. $\odot$ represents element-wise product, σ the logistic sigmoid activation. Modifying Eq. 5, the probability over the grounding regions including the visual sentinel is:

$$\boldsymbol{P}_{\boldsymbol{r}}^t = \text{softmax}([\boldsymbol{u}^t; \boldsymbol{w}_h^T \tanh(\boldsymbol{W}_s \boldsymbol{s}_t + \boldsymbol{W}_z \boldsymbol{h}_t)]) \tag{9}$$

where $\boldsymbol{W}_s \in \mathbb{R}^{d\times d}$ and $\boldsymbol{W}_z \in \mathbb{R}^{d\times d}$ are the parameters. Notably, $\boldsymbol{W}_z$ and $\boldsymbol{w}_h$ are the same parameters as in Eq. 4. $\boldsymbol{P}_{\boldsymbol{r}}^t$ is the probability distribution over grounding regions $\boldsymbol{r_I}$ and visual sentinel $\tilde{r}$. The last element of the vector in Eq. 9 captures $p(\tilde{r}|\boldsymbol{y}_{1:t-1})$.

We feed the hidden state $\boldsymbol{h}_t$ into a softmax layer to obtain the probability over textual words conditioned on the image, all previous words, and the visual sentinel:

$$\boldsymbol{P}_{txt}^t = \text{softmax}\left(\boldsymbol{W}_q \boldsymbol{h}_t\right) \tag{10}$$

where $\boldsymbol{W}_q \in \mathbb{R}^{V\times d}$, d is hidden state size, and V is textual vocabulary size. Plugging in Eq. 10 and $p(\tilde{r}|\boldsymbol{y}_{1:t-1})$ from the last element of the vector in Eq. 9 into Eq. 6 gives us the probability of generating a textual word in the template.

3.2. Caption Refinement: Filling in The Slots

To fill the slots in the generated template with visual words grounded in image regions, we leverage the outputs of an object detection network. Given a grounding region, the category can be obtained through any detection framework [37]. But outputs of detection networks are typically singular coarse labels *e.g.* "dog". Captions often refer to these entities in a fine-grained fashion *e.g.* "puppy" or in the plural form "dogs". In order to accommodate for these linguistic variations, the visual word y^{vis} in our model is a refinement of the category name by considering the following two factors: First, determine the plurality – whether it should be singular or plural. Second, determine the fine-grained class (if any). Using two single layer MLPs with ReLU activation $f(\cdot)$, we compute them with:

$$\boldsymbol{P}_b^t = \text{softmax}\left(\boldsymbol{W}_b f_b\left([\boldsymbol{v}_t; \boldsymbol{h}_t]\right)\right) \tag{11}$$

$$\boldsymbol{P}_g^t = \text{softmax}\left(\boldsymbol{U}^T \boldsymbol{W}_g f_g\left([\boldsymbol{v}_t; \boldsymbol{h}_t]\right)\right) \tag{12}$$

$\boldsymbol{W}_b \in \mathbb{R}^{2\times d}$, $\boldsymbol{W}_g \in \mathbb{R}^{300\times d}$ are the weight parameters. $\boldsymbol{U} \in \mathbb{R}^{300\times k}$ is the glove vector embeddings [35] for k fine-grained words associated with the category name. The visual word y_t^{vis} is then determined by plurality and fine-grained class (*e.g.*, if plurality is plural, and the fine-grained class is "puppy", the visual word will be "puppies").

3.3. Objective

Most standard image captioning datasets (*e.g.* COCO [26]) do not contain phrase grounding annotations, while some datasets do (*e.g.* Flickr30k [36]). Our training objective (presented next) can incorporate different kinds of supervision – be it strong annotations indicating which words in the caption are grounded in which boxes in the image, or weak supervision where objects are annotated in the image but are not aligned to words in the caption. Given the target ground truth caption $\boldsymbol{y}_{1:T}^*$ and a image captioning model with parameters $\boldsymbol{\theta}$, we minimize the cross entropy loss:

$$L(\boldsymbol{\theta}) = -\sum_{t=1}^{T} \log \Big(\overbrace{p(y_t^*|\tilde{r}, \boldsymbol{y}_{1:t-1}^*) p(\tilde{r}|\boldsymbol{y}_{1:t-1}^*) \mathbb{1}_{(y_t^* = y^{\text{txt}})}}^{\text{Textual word probability}} + \underbrace{p\left(b_t^*, s_t^* | \boldsymbol{r}_t, \boldsymbol{y}_{1:t-1}^*\right)}_{\text{Caption refinement}} \underbrace{\Big(\frac{1}{m} \sum_{i=1}^{m} p\left(r_t^i | \boldsymbol{y}_{1:t-1}^*\right) \Big) \mathbb{1}_{(y_t^* = y^{\text{vis}})}}_{\text{Averaged target region probability}} \Big) \tag{13}$$

where y_t^* is the word from the ground truth caption at time t. $\mathbb{1}_{(y_t^* = y^{\text{txt}})}$ is the indicator function which equals to 1 if y_t^* is textual word and 0 otherwise. b_t^* and s_t^* are the target ground truth plurality and find-grained class. $\{r_t^i\}_{i=1}^m \in \boldsymbol{r_I}$ are the target grounding regions of the visual word at time t. We maximize the averaged log probability of the target grounding regions.

Visual word extraction. During training, visual words in a caption are dynamically identified by matching the base form of each word (using the Stanford lemmatization toolbox [30]) against a vocabulary of visual words (details of how to get visual word can be found in dataset Sec. 4). The grounding regions $\{r_t^i\}_{i=1}^m$ for a visual word y_t is identified by computing the IoU of all boxes detected by the object detection network with the ground truth bounding box associated with the category corresponding to y_t. If the score exceeds a threshold of 0.5 and the grounding region label matches the visual word, the bounding boxes are selected as the grounding regions. E.g., given a target visual word "cat", if there are no proposals that match the target bounding box, the model predicts the textual word "cat" instead.

3.4. Implementation Details

Detection model. We use Faster R-CNN [37] with ResNet-101 [15] to obtain region proposals for the image. We use an IoU threshold of 0.7 for region proposal suppression and 0.3 for class suppressions. A class detection confidence threshold of 0.5 is used to select regions.

Region feature. We use a pre-trained ResNet-101 [15] in our model. The image is first resized to 576 × 576 and we random crop 512 × 512 as the input to the CNN network. Given proposals from the pre-trained detection model, the feature $\boldsymbol{v}_i$ for region i is a concatenation of 3 different features $\boldsymbol{v}_i = [\boldsymbol{v}_i^p; \boldsymbol{v}_i^l; \boldsymbol{v}_i^g]$ where $\boldsymbol{v}_i^p$ is the pooling feature of RoI align layer [14] given the proposal coordinates, $\boldsymbol{v}_i^l$ is the location feature and $\boldsymbol{v}_i^g$ is the glove vector embedding of the class label for region i. Let $x_{\text{min}}, y_{\text{min}}, x_{\text{max}}, y_{\text{max}}$ be the bounding box coordinates of the region b; W_I and H_I be the width and height of the image I. Then the location feature $\boldsymbol{v}_i^l$ can be obtained by projecting the normalized location $[\frac{x_{\text{min}}}{W_I}, \frac{y_{\text{min}}}{H_I}, \frac{x_{\text{max}}}{W_I}, \frac{y_{\text{max}}}{H_I}]$ into another embedding space.

Language model. We use an attention model with two LSTM layers [3] as our base attention model. Given N region features from detection proposals $\boldsymbol{V} = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_N\}$ and CNN features from the last convolution layer at K grids $\hat{\boldsymbol{V}} = \{\hat{\boldsymbol{v}}_1, \dots, \hat{\boldsymbol{v}}_K\}$, the language model has two separate attention layers shown in Fig 4. The attention distribution over the image features for detection proposals is:

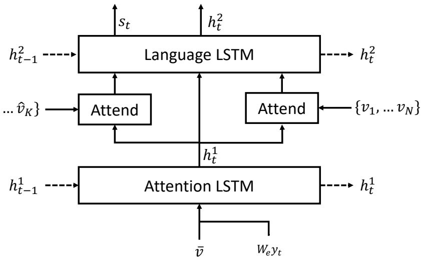


Figure 4. Language model used in our approach.

$$\begin{aligned} \boldsymbol{z}_t &= \boldsymbol{w}_z^T \tanh\left(\boldsymbol{W}_v \boldsymbol{V} + (\boldsymbol{W_g h_t}) \mathbb{1}^T\right) \\ \boldsymbol{\alpha}_t &= \text{softmax}(\boldsymbol{z}_t) \end{aligned} \tag{14}$$

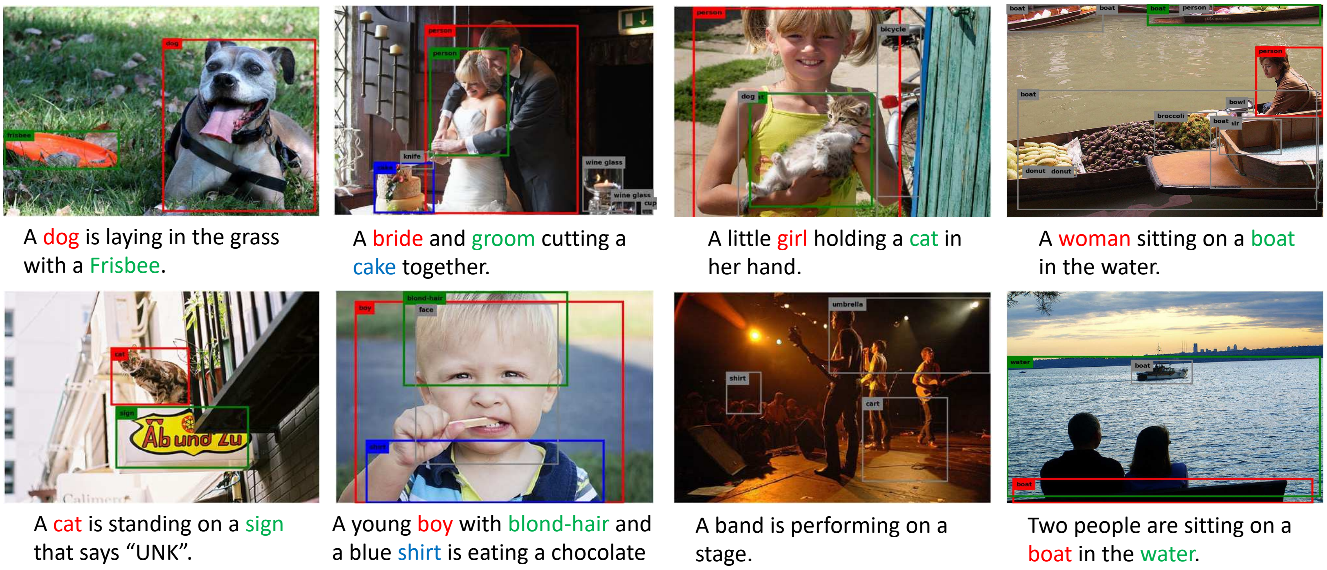
where $\boldsymbol{W}_v \in \mathbb{R}^{m \times d}$, $\boldsymbol{W}_g \in \mathbb{R}^{d \times d}$ and $\boldsymbol{w} \in \mathbb{R}^{d \times 1}$. $\mathbb{1} \in \mathbb{R}^N$ is a vector with all elements set to 1. $\boldsymbol{\alpha}_t$ is the attention weight over N image location features.

Training details. In our experiments, we use a two layer LSTM with hidden size 1024. The number of hidden units in the attention layer and the size of the input word embedding are 512. We use the Adam [22] optimizer with an initial learning rate of 5×10^{-4} and anneal the learning rate by a factor of 0.8 every three epochs. We train the model up to 50 epochs with early stopping. Note that we do not finetune the CNN network during training. We set the batch size to be 100 for COCO [26] and 50 for Flickr30k [36].

4. Experimental Results

Datasets. We experiment with two datasets. Flickr30k Entities [36] contains 275,755 bounding boxes from 31,783 images associated with natural language phrases. Each image is annotated with 5 crowdsourced captions. For each annotated phrase in the caption, we identify visual words by selecting the inner most NP (noun phrase) tag from the Stanford part-of-speech tagger [6]. We use Stanford Lemmatization Toolbox [30] to get the base form of the entity words resulting in 2,567 unique words.

COCO [26] contains 82,783, 40,504 and 40,775 images for training, validation and testing respectively. Each image has around 5 crowdsourced captions. Unlike Flickr30k Entities, COCO does not have bounding box annotations associated with specific phrases or entities in the caption. To identify visual words, we manually constructed an object category to word mapping that maps object categories like <person> to a list of potential fine-grained labels like ["child", "baker", ...]. This results in 80 categories with a total of 413 fine-grained classes. See supp. for details.



A dog is laying in the grass with a Frisbee.

A bride and groom cutting a cake together.

A little girl holding a cat in her hand.

A woman sitting on a boat in the water.

A cat is standing on a sign that says "UNK".

A young boy with blond-hair and a blue shirt is eating a chocolate

A band is performing on a stage.

Two people are sitting on a boat in the water.

Figure 5. Generated captions and corresponding visual grounding regions on the standard image captioning task (Top: COCO, Bottom: Flickr30k). Different colors show a correspondence between the visual words and grounding regions. Grey regions are the proposals not selected in the caption. First 3 columns show success and last column shows failure cases (words are grounded in the wrong region).

Method	BLEU1	BLEU4	METEOR	CIDEr	SPICE
Hard-Attention [45]	66.9	19.9	18.5	-	-
ATT-FCN [49]	64.7	23.0	18.9	-	-
Adaptive [27]	67.7	25.1	20.4	53.1	14.5
NBT	**69.0**	**27.1**	**21.7**	**57.5**	**15.6**
NBT$^{\text{oracle}}$	72.0	28.5	23.1	64.8	19.6

Table 1. Performance on the test portion of Karpathy *et al.* [20]'s splits on Flickr30k Entities dataset.

Detector pre-training. We use open an source implementation [46] of Faster-RCNN [37] to train the detector. For Flickr30K Entities, we use visual words that occur at least 100 times as detection labels, resulting in a total of 460 detection labels. Since detection labels and visual words have a one-to-one mapping, we do not have fine-grained classes for the Flickr30K Entities dataset – the caption refinement process only determines the plurality of detection labels. For COCO, ground truth detection annotations are used to train the object detector.

Caption pre-processing. We truncate captions longer than 16 words for both COCO and Flickr30k Entities dataset. We then build a vocabulary of words that occur at least 5 times in the training set, resulting in 9,587 and 6,864 words for COCO and Flickr30k Entities, respectively.

4.1. Standard Image Captioning

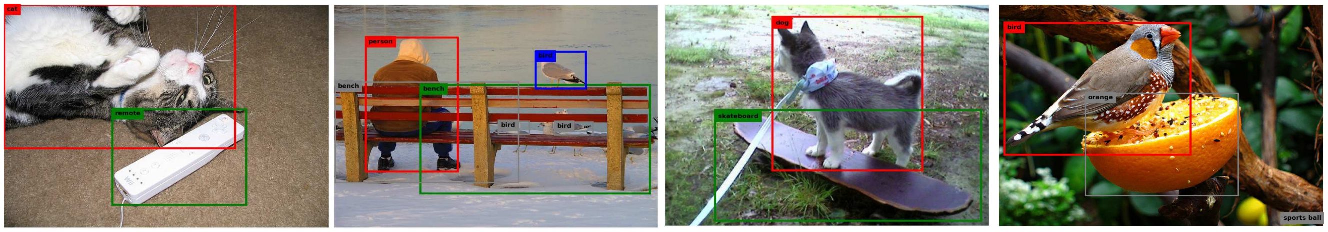
For standard image captioning, we use splits from Karpathy *et al.* [20] on COCO/Flickr30k. We report results using the COCO captioning evaluation toolkit [26], which reports the widely used automatic evaluation metrics, BLEU [34], METEOR [10], CIDEr [41] and SPICE [1].

Method	BLEU1	BLEU4	METEOR	CIDEr	SPICE
Adaptive [27]	74.2	32.5	26.6	**108.5**	19.5
Att2in [38]	-	31.3	26.0	101.3	-
Up-Down [3]	74.5	33.4	26.1	105.4	19.2
Att2in* [38]	-	33.3	26.3	111.4	-
Up-Down† [3]	79.8	36.3	27.7	120.1	21.4
NBT	**75.5**	**34.7**	**27.1**	107.2	**20.1**
NBT$^{\text{oracle}}$	75.9	34.9	27.4	108.9	20.4

Table 2. Performance on the test portion of Karpathy *et al.* [20]'s splits on COCO dataset. $*$ directly optimizes the CIDEr Metric, † uses better image features, and are thus not directly comparable.

We present our methods trained on different object detectors: Flickr and COCO. We compare our approach (referred to as NBT) to recently proposed Hard-Attention [45], ATT-FCN [49] and Adaptive [27] on Flickr30k, and Att2in [38], Up-Down [3] on COCO. Since object detectors have not yet achieved near-perfect accuracies on these datasets, we also report the performance of our model under an oracle setting, where the ground truth object region and category is also provided during test time. (referred to as NBT$^{\text{oracle}}$) This can be viewed as the upper bound of our method when we have perfect object detectors.

Table 1 shows results on the Flickr30k dataset. We see that our method achieves state of the art on all automatic evaluation metrics, outperforming the previous state-of-art model Adaptive [27] by 2.0 and 4.4 on BLEU4 and CIDEr. When using ground truth proposals, NBT$^{\text{oracle}}$ significantly outperforms previous methods, improving 5.1 on SPICE, which implies that our method could further benefit from improved object detectors.



A cat laying on the floor next to a remote control.

A man sitting on a bench next to a bird.

A dog is standing on a skateboard in the grass.

A bird sitting on a branch in a tree.

Figure 6. Generated captions and corresponding visual grounding regions for the robust image captioning task. “cat-remote”, “man-bird”, “dog-skateboard” and “orange-bird” are co-occurring categories excluded in the training split. First 3 columns show success and last column shows failure case (orange was not mentioned).

Method	BLEU4	METEOR	CIDEr	SPICE	Accuracy
Att2in [38]	31.5	24.6	90.6	17.7	39.0
Up-Down [3]	31.6	25.0	92.0	18.1	39.7
NBT	**31.7**	**25.2**	**94.1**	**18.3**	**42.4**
NBT$^{\text{oracle}}$	31.9	25.5	95.5	18.7	45.7

Table 3. Performance on the test portion of the robust image captioning split on COCO dataset.

Table 2 shows results on the COCO dataset. Our method outperforms 4 out of 5 automatic evaluation metrics compared to the state of the art [38, 27, 3] without using better visual features or directly optimizing the CIDEr metric. Interestingly, the NBT$^{\text{oracle}}$ has little improvement over NBT. We suspect the reason is that explicit ground truth annotation is absent for visual words. Our model can be further improved with explicit co-reference supervision where the ground truth location annotation of the visual word is provided. Fig. 5 shows qualitative results on both datasets. We see that our model learns to correctly identify the visual word, and ground it in image regions even under weak supervision (COCO). Our model is also robust to erroneous detections and produces correct captions (3rd column).

4.2. Robust Image Captioning

To quantitatively evaluate image captioning models for novel scene compositions, we present a new split of the COCO dataset, called the robust-COCO split. This new split is created by re-organizing the train and val splits of the COCO dataset such that the distribution of co-occurring objects in train is different from test. We also present a new metric to evaluate grounding.

Robust split. To create the new split, we first identify entity words that belong to the 80 COCO object categories by following the same pre-processing procedure. For each image, we get a list of object categories that are mentioned in the caption. We then calculate the co-occurrence statistics for these 80 object categories. Starting from the least co-occurring category pairs, we greedily add them to the test set and ensure that for each category, at least half the instances of each category are in the train set. As a result, there are sufficient examples from each category in train, but at test time we see novel compositions (pairs) of categories. Remaining images are assigned to the training set. The final split has 110,234/3,915/9,138 images in train/val/test respectively.

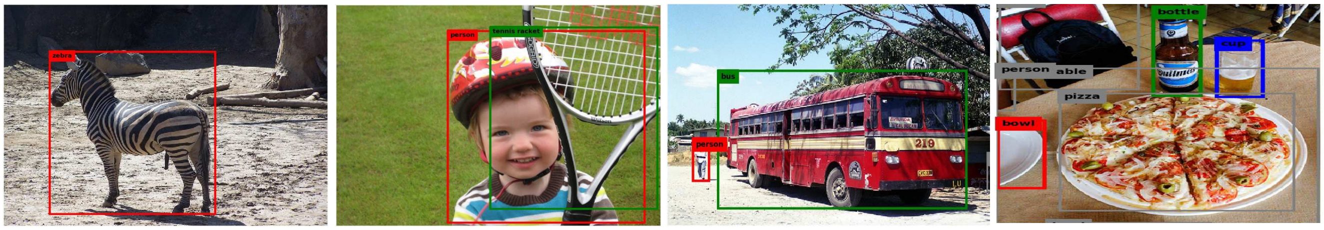
Evaluation metric. To evaluate visual grounding on the robust-COCO split, we want a metric that indicates whether or not a generated caption includes the new object combination. Common automatic evaluation metrics such as BLEU [34] and CIDEr [41] measure the overall sentence fluency. We also measure whether the generated caption contains the novel co-occurring categories that exist in the ground truth caption. A generated caption is deemed 100% accurate if it contains at least one mention of the *compositionally novel* category-pairs in any ground truth annotation that describe the image.

Results and analysis. We compare our method with state of the art Att2in [38] and Up-Down [3]. These are implemented using the open source implementation from [28] that can replicate results on Karpathy’s split. We follow the experimental setting from [38] and train the model using the robust-COCO train set. Table 3 shows the results on the robust-COCO split. As we can see, all models perform worse on the robust-COCO split than the Karpathy’s split by 2$\sim$3 points in general. Our method outperforms the previous state of the art methods on all metrics, outperforming Up-Down [3] by 2.7 on the proposed metric. The oracle setting (NBT$^{\text{oracle}}$) has consistent improvements on all metrics, improving 3.3 on the proposed metric.

Fig. 6 shows qualitative results on the robust image captioning task. Our model successfully produces a caption with novel compositions, such as “cat-remote”, “man-bird” and “dog-skateboard” to describe the image. The last column shows failure cases where our model didn’t select “orange” in the caption. We can force our model to produce a caption containing “orange” and “bird” using constrained beam search [2], further illustrated in Sec. 4.3.

4.3. Novel Object Captioning

Since our model directly fills the “slotted” caption template with the concept, it can seamlessly generate descrip-



A zebra that is standing in the dirt.

A little girl wearing a helmet and holding a tennis racket.

A woman standing in front of a red bus.

A plate of food with a bottle and a cup of beer.

Figure 7. Generated captions and corresponding visual grounding regions for the novel object captioning task. “zebra”, “tennis racket”, “bus” and “pizza” are categories excluded in the training split. First 3 columns show success and last column shows a failure case.

	Out-of-Domain Test Data												In-Domain Test Data		
Method	bottle	bus	couch	microwave	pizza	racket	suitcase	zebra	Avg	SPICE	METEOR	CIDEr	SPICE	METEOR	CIDER
DCC [4]	4.6	29.8	45.9	28.1	64.6	52.2	13.2	79.9	39.8	13.4	21.0	59.1	15.9	23.0	77.2
NOC [42]	17.8	68.8	25.6	24.7	69.3	68.1	39.9	89.0	49.1	-	21.4	-	-	-	-
C-LSTM [48]	29.7	74.4	38.8	27.8	68.2	70.3	44.8	91.4	55.7	-	23.0	-	-	-	-
Base+T4 [2]	16.3	67.8	48.2	29.7	77.2	57.1	49.9	85.7	54.0	15.9	23.3	77.9	18.0	24.5	86.3
NBT*+G	7.1	73.7	34.4	61.9	59.9	20.2	42.3	88.5	48.5	15.7	22.8	77.0	17.5	24.3	87.4
NBT†+G	14.0	74.8	42.8	63.7	74.4	19.0	44.5	92.0	53.2	16.6	23.9	84.0	**18.4**	25.3	94.0
NBT†+T1	36.2	77.7	43.9	65.8	70.3	19.8	51.2	93.7	57.3	16.7	23.9	85.7	**18.4**	**25.5**	**95.2**
NBT†+T2	**38.3**	**80.0**	**54.0**	**70.3**	**81.1**	**74.8**	**67.8**	**96.6**	**70.3**	**17.4**	**24.1**	**86.0**	18.0	25.0	92.1

Table 4. Evaluation of captions generated using the proposed method. G means greedy decoding, and T1$-$2 means using constrained beam search [2] with 1$-$2 top detected concepts. $*$ is the result using VGG-16 [40] and $\dagger$ is the result using ResNet-101.

tions for out-of-domain images. We replicated an existing experimental design [4] on COCO which excludes all the image-sentence pairs that contain at least one of eight objects in COCO. The excluded objects are ‘bottle’, “bus”, “couch”, “microwave”, “pizza”, “racket”, “suitcase” and “zebra”. We follow the same splits for training, validation, and testing as in prior work [4]. We use Faster R-CNN in conjunction with ResNet-101 which is pre-trained on COCO train split as the detection model. Note that we do not pre-train the language model using COCO captions as in [4, 42, 48], and simply replace the novel object’s word embedding with an existing one which belongs to the same super-category in COCO (e.g., bus ← car).

Following [2], the test set is split into in-domain and out-of-domain subsets. We report F1 as in [4], which checks if the specific excluded object is mentioned in the generated caption. To evaluate the quality of the generated caption, we use SPICE, METEOR and CIDEr metrics and the scores on out-of-domain test data are macro-averaged across eight excluded categories. For consistency with previous work [3], the inverse document frequency statistics used by CIDEr are determined across the entire test set.

As illustrated in Table 4.1, simply using greedy decoding, our model (NBT*+G) can successfully caption novel concepts with minimum changes to the model. When using ResNet-101 and constrained beam search [2], our model significantly outperforms prior works under F1 scores, SPICE, METEOR, and CIDEr, across both out-of-domain and in-domain test data. Specifically, NBT†+T2 outperforms the previous state-of-art model C-LSTM by 14.6% on average F1 scores. From the category F1 scores, we can see that our model is less likely to select small objects, e.g. “bottle”, “racket” when only using the greedy decoding. Since the visual words are grounded at the object-level, by using [2], our model was able to significantly boost the captioning performance on out-of-domain images. Fig. 7 shows qualitative novel object captioning results. Also see rightmost example in Fig. 2.

5. Conclusion

In this paper, we introduce Neural Baby Talk, a novel image captioning framework that produces natural language explicitly grounded in entities object detectors find in images. Our approach is a two-stage approach that first generates a hybrid template that contains a mix of words from a text vocabulary as well as slots corresponding to image regions. It then fills the slots based on categories recognized by object detectors in the image regions. We also introduce a robust image captioning split by re-organizing the train and val splits of the COCO dataset. Experimental results on standard, robust, and novel object image captioning tasks validate the effectiveness of our proposed approach.

Acknowledgements This work was funded in part by: NSF CAREER awards to DB, DP; ONR YIP awards to DP, DB; ONR Grants N00014-14-1-{0679,2713}; PGA Family Foundation award to DP; Google FRAs to DP, DB; and Amazon ARAs to DP, DB; DARPA XAI grant to DB, DP.

References

[1] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016. 6

[2] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Guided open vocabulary image captioning with constrained beam search. *EMNLP*, 2017. 3, 7, 8

[3] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 3, 5, 6, 7, 8

[4] L. Anne Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, and T. Darrell. Deep compositional captioning: Describing novel object categories without paired training data. In *CVPR*, 2016. 3, 8

[5] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *ICCV*, 2015. 1

[6] D. Chen and C. Manning. A fast and accurate dependency parser using neural networks. In *EMNLP*, 2014. 5

[7] X. Chen and C. Lawrence Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *CVPR*, 2015. 3

[8] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra. Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions? In *EMNLP*, 2016. 1

[9] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra. Visual Dialog. In *CVPR*, 2017. 1

[10] M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *EACL 2014 Workshop on Statistical Machine Translation*, 2014. 6

[11] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 3

[12] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, et al. From captions to visual concepts and back. In *CVPR*, 2015. 1, 3

[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*, 2010. 2, 3

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *ICCV*, 2017. 5

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4

[17] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko. Modeling relationships in referential expressions with compositional modular networks. *arXiv preprint arXiv:1611.09978*, 2016. 3

[18] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *CVPR*, 2016. 3

[19] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *CVPR*, 2016. 3

[20] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 1, 3, 6

[21] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 3

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

[23] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Multimodal neural language models. In *ICML*, 2014. 3

[24] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Babytalk: Understanding and generating simple image descriptions. In *CVPR*, 2011. 1, 2, 3

[25] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi. Collective generation of natural image descriptions. In *ACL*, 2012. 2, 3

[26] 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, 4, 5, 6

[27] J. Lu, C. Xiong, D. Parikh, and R. Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*, 2017. 1, 3, 4, 6, 7

[28] R. Luo. Unofficial pytorch implementation for self-critical sequence training for image captioning. https://github.com/ruotianluo/self-critical.pytorch, 2017. 7

[29] R. Luo and G. Shakhnarovich. Comprehension-guided referring expressions. In *CVPR*, 2017. 3

[30] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *ACL*, 2014. 5

[31] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 3

[32] 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. 3

[33] M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. Berg, K. Yamaguchi, T. Berg, K. Stratos, and H. Daumé III. Midge: Generating image descriptions from computer vision detections. In *EACL*, 2012. 2, 3

[34] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 6, 7

[35] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 4

[36] 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. 3, 4, 5

[37] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 3, 4, 5, 6

[38] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017. 3, 4, 6, 7

[39] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of textual phrases in images by reconstruction. In *ECCV*, 2016. 3

[40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 8

[41] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 6, 7

[42] S. Venugopalan, L. A. Hendricks, M. Rohrbach, R. Mooney, T. Darrell, and K. Saenko. Captioning images with diverse objects. In *CVPR*, 2017. 3, 8

[43] O. Vinyals, M. Fortunato, and N. Jaitly. Pointer networks. In *NIPS*, 2015. 4

[44] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 3

[45] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 1, 3, 6

[46] J. Yang, J. Lu, D. Batra, and D. Parikh. A faster pytorch implementation of faster r-cnn. `https://github.com/jwyang/faster-rcnn.pytorch`, 2017. 6

[47] Z. Yang, Y. Yuan, Y. Wu, R. Salakhutdinov, and W. W. Cohen. Encode, review, and decode: Reviewer module for caption generation. In *NIPS*, 2016. 1, 3

[48] T. Yao, Y. Pan, Y. Li, and T. Mei. Incorporating copying mechanism in image captioning for learning novel objects. In *CVPR*, 2017. 3, 8

[49] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In *CVPR*, 2016. 1, 3, 6

[50] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *ECCV*, 2016. 3