Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Kelvin Xu KELVIN.XU@UMONTREAL.CA
Jimmy Lei Ba JIMMY@PSI.UTORONTO.CA
Ryan Kiros RKIROS@CS.TORONTO.EDU
Kyunghyun Cho KYUNGHYUN.CHO@UMONTREAL.CA
Aaron Courville AARON.COURVILLE@UMONTREAL.CA
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Richard S. Zemel ZEMEL@CS.TORONTO.EDU
Yoshua Bengio FIND-ME@THE.WEB

Abstract

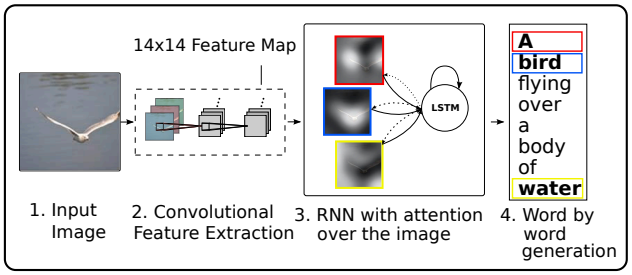
Inspired by recent work in machine translation and object detection, we introduce an attention based model that automatically learns to describe the content of images. We describe how we can train this model in a deterministic manner using standard backpropagation techniques and stochastically by maximizing a variational lower bound. We also show through visualization how the model is able to automatically learn to fix its gaze on salient objects while generating the corresponding words in the output sequence. We validate the use of attention with state-of-the-art performance on three benchmark datasets: Flickr8k, Flickr30k and MS COCO.

1. Introduction

Automatically generating captions of an image is a task very close to the heart of scene understanding — one of the primary goals of computer vision. Not only must caption generation models be powerful enough to solve the computer vision challenges of determining which objects are in an image, but they must also be capable of capturing and expressing their relationships in a natural language. For this reason, caption generation has long been viewed as a difficult problem. It is a very important challenge for machine learning algorithms, as it amounts to mimicking the remarkable human ability to compress huge amounts of salient visual infomation into descriptive language.

Despite the challenging nature of this task, there has been a recent surge of research interest in attacking the image caption generation problem. Aided by advances in training neural networks (Krizhevsky et al., 2012) and large classification datasets (Russakovsky et al., 2014), recent work has significantly improved the quality of caption generation using a combination of convolutional neural networks (convnets) to obtain vectorial representation of images and recurrent neural networks to decode those representations into natural language sentences (see Sec. 2).

Figure 1. Our model learns a words/image alignment. The visualized attentional maps (3) are explained in section 3.1 & 5.4



One of the most curious facets of the human visual system is the presence of attention (Rensink, 2000; Corbetta & Shulman, 2002). Rather than compress an entire image into a static representation, attention allows for salient features to dynamically come to the forefront as needed. This is especially important when there is a lot of clutter in an image. Using representations (such as those from the top layer of a convnet) that distill information in image down to the most salient objects is one effective solution that has been widely adopted in previous work. Unfortunately, this has one potential drawback of losing information which could be useful for richer, more descriptive captions. Using more low-level representation can help preserve this information. However working with these features necessitates a powerful mechanism to steer the model to information important to the task at hand.

In this paper, we describe approaches to caption generation that attempt to incorporate a form of attention with

Figure 2. Attention over time. As the model generates each word, its attention changes to reflect the relevant parts of the image. "soft" (top row) vs "hard" (bottom row) attention. (Note that both models generated the same captions in this example.)

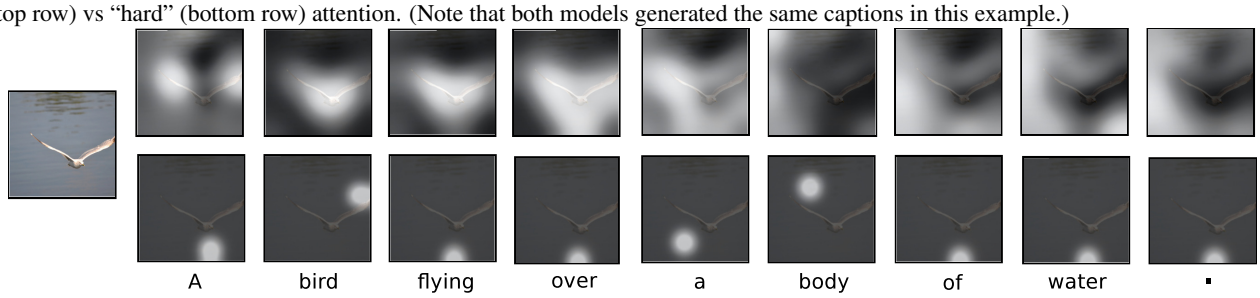
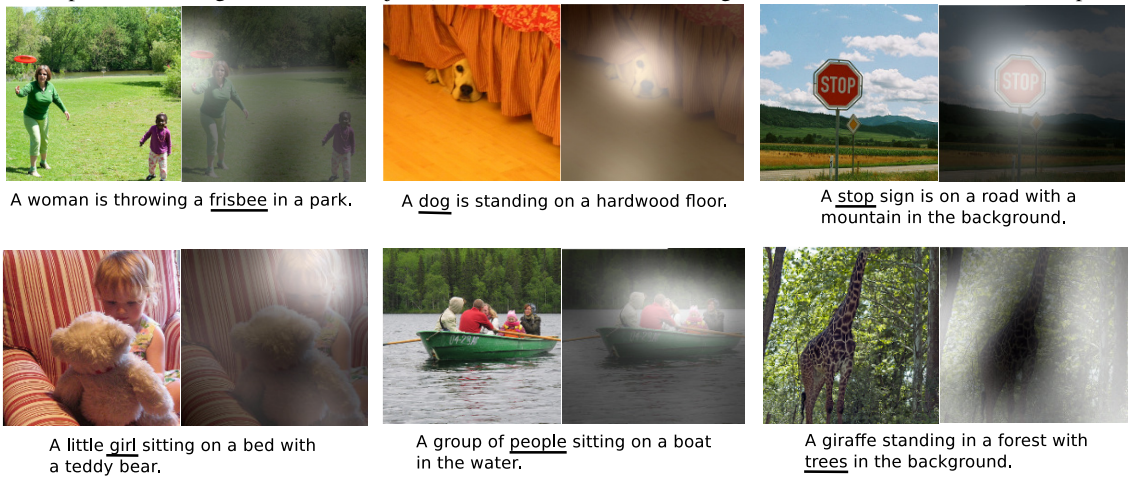


Figure 3. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)



two variants: a "hard" attention mechanism and a "soft" attention mechanism. We also show how one advantage of including attention is the ability to visualize what the model "sees". Encouraged by recent advances in caption generation and inspired by recent success in employing attention in machine translation (Bahdanau et al., 2014) and object recognition (Ba et al., 2014; Mnih et al., 2014), we investigate models that can attend to salient part of an image while generating its caption.

The contributions of this paper are the following:

- We introduce two attention-based image caption generators under a common framework (Sec. 3.1): 1) a "soft" deterministic attention mechanism trainable by standard back-propagation methods and 2) a "hard" stochastic attention mechanism trainable by maximizing an approximate variational lower bound or equivalently by REINFORCE (Williams, 1992).
- We show how we can gain insight and interpret the results of this framework by visualizing "where" and "what" the attention focused on. (see Sec. 5.4)
- Finally, we quantitatively validate the usefulness of attention in caption generation with state of the art performance (Sec. 5.3) on three benchmark datasets: Flickr8k (Hodosh et al., 2013) , Flickr30k (Young et al., 2014) and the MS COCO dataset (Lin et al., 2014).

2. Related Work

In this section we provide relevant background on previous work on image caption generation and attention. Recently, several methods have been proposed for generating image descriptions. Many of these methods are based on recurrent neural networks and inspired by the successful use of sequence to sequence training with neural networks for machine translation (Cho et al., 2014; Bahdanau et al., 2014; Sutskever et al., 2014). One major reason image caption generation is well suited to the encoder-decoder framework (Cho et al., 2014) of machine translation is because it is analogous to "translating" an image to a sentence.

The first approach to use neural networks for caption generation was Kiros et al. (2014a), who proposed a multimodal log-bilinear model that was biased by features from the image. This work was later followed by Kiros et al. (2014b) whose method was designed to explicitly allow a natural way of doing both ranking and generation. Mao et al. (2014) took a similar approach to generation but replaced a feed-forward neural language model with a recurrent one. Both Vinyals et al. (2014) and Donahue et al. (2014) use LSTM RNNs for their models. Unlike Kiros et al. (2014a) and Mao et al. (2014) whose models see the image at each time step of the output word sequence, Vinyals et al. (2014) only show the image to the RNN at the beginning. Along

with images, Donahue et al. (2014) also apply LSTMs to videos, allowing their model to generate video descriptions.

All of these works represent images as a single feature vector from the top layer of a pre-trained convolutional network. Karpathy & Li (2014) instead proposed to learn a joint embedding space for ranking and generation whose model learns to score sentence and image similarity as a function of R-CNN object detections with outputs of a bidirectional RNN. Fang et al. (2014) proposed a three-step pipeline for generation by incorporating object detections. Their model first learn detectors for several visual concepts based on a multi-instance learning framework. A language model trained on captions was then applied to the detector outputs, followed by rescoring from a joint image-text embedding space. Unlike these models, our proposed attention framework does not explicitly use object detectors but instead learns latent alignments from scratch. This allows our model to go beyond "objectness" and learn to attend to abstract concepts.

Prior to the use of neural networks for generating captions, two main approaches were dominant. The first involved generating caption templates which were filled in based on the results of object detections and attribute discovery (Kulkarni et al. (2013), Li et al. (2011), Yang et al. (2011), Mitchell et al. (2012), Elliott & Keller (2013)). The second approach was based on first retrieving similar captioned images from a large database then modifying these retrieved captions to fit the query (Kuznetsova et al., 2012; 2014). These approaches typically involved an intermediate "generalization" step to remove the specifics of a caption that are only relevant to the retrieved image, such as the name of a city. Both of these approaches have since fallen out of favour to the now dominant neural network methods.

There has been a long line of previous work incorporating attention into neural networks for vision related tasks. Some that share the same spirit as our work include Larochelle & Hinton (2010); Denil et al. (2012); Tang et al. (2014). In particular however, our work directly extends the work of Bahdanau et al. (2014); Mnih et al. (2014); Ba et al. (2014).

3. Image Caption Generation with Attention Mechanism

3.1. Model Details

In this section, we describe the two variants of our attention-based model by first describing their common framework. The main difference is the definition of the ϕ function which we describe in detail in Section 4. We denote vectors with bolded font and matrices with capital letters. In our description below, we suppress bias terms for readability.

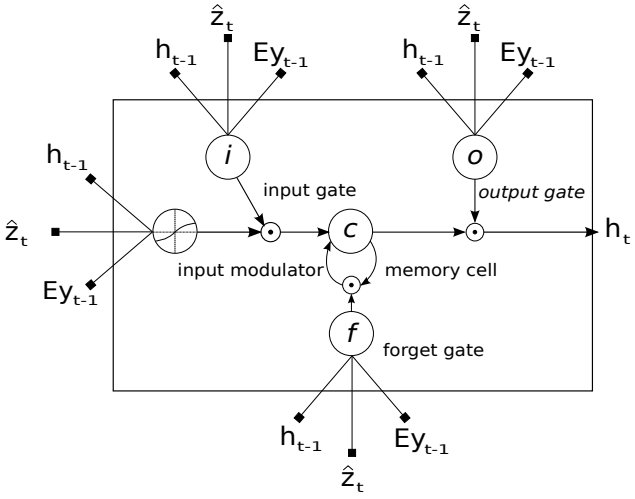


Figure 4. A LSTM cell, lines with bolded squares imply projections with a learnt weight vector. Each cell learns how to weigh its input components (input gate), while learning how to modulate that contribution to the memory (input modulator). It also learns weights which erase the memory cell (forget gate), and weights which control how this memory should be emitted (output gate).

3.1.1. Encoder: Convolutional Features

Our model takes a single raw image and generates a caption $\mathbf{y}$ encoded as a sequence of 1-of-K encoded words.

$$y = \{\mathbf{y}_1, \ldots, \mathbf{y}_C\}, \ \mathbf{y}_i \in \mathbb{R}^K$$

where K is the size of the vocabulary and C is the length of the caption.

We use a convolutional neural network in order to extract a set of feature vectors which we refer to as annotation vectors. The extractor produces L vectors, each of which is a D-dimensional representation corresponding to a part of the image.

$$a = \{\mathbf{a}_1, \ldots, \mathbf{a}_L\}, \ \mathbf{a}_i \in \mathbb{R}^D$$

In order to obtain a correspondence between the feature vectors and portions of the 2-D image, we extract features from a lower convolutional layer unlike previous work which instead used a fully connected layer. This allows the decoder to selectively focus on certain parts of an image by selecting a subset of all the feature vectors.

3.1.2. Decoder: Long Short-Term Memory Network

We use a long short-term memory (LSTM) network (Hochreiter & Schmidhuber, 1997) that produces a caption by generating one word at every time step conditioned on a context vector, the previous hidden state and the previously generated words. Our implementation of LSTM

closely follows the one used in Zaremba et al. (2014) (see Fig. 4). Using $T_{s,t} : \mathbb{R}^s \rightarrow \mathbb{R}^t$ to denote a simple affine transformation with parameters that are learned,

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{h}_{t-1} \\ \hat{\mathbf{z}}_t \end{pmatrix} \quad (1)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \quad (2)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t). \quad (3)$$

Here, $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{c}_t$, $\mathbf{o}_t$, $\mathbf{h}_t$ are the input, forget, memory, output and hidden state of the LSTM, respectively. The vector $\hat{\mathbf{z}} \in \mathbb{R}^D$ is the context vector, capturing the visual information associated with a particular input location, as explained below. $\mathbf{E} \in \mathbb{R}^{m \times K}$ is an embedding matrix. Let m and n denote the embedding and LSTM dimensionality respectively and σ and $\odot$ be the logistic sigmoid activation and element-wise multiplication respectively.

In simple terms, the context vector $\hat{\mathbf{z}}_t$ (equations (1)–(3)) is a dynamic representation of the relevant part of the image input at time t. We define a mechanism ϕ that computes $\hat{\mathbf{z}}_t$ from the annotation vectors $\mathbf{a}_i, i = 1, \ldots, L$ corresponding to the features extracted at different image locations. For each location i, the mechanism generates a positive weight α_i which can be interpreted either as the probability that location i is the right place to focus for producing the next word (the "hard" but stochastic attention mechanism), or as the relative importance to give to location i in blending the a_i's together. The weight α_i of each annotation vector a_i is computed by an *attention model* f_{att} for which we use a multilayer perceptron conditioned on the previous hidden state h_{t-1}. The soft version of this attention mechanism was introduced by Bahdanau et al. (2014). For emphasis, we note that the hidden state varies as the output RNN advances in its output sequence: "where" the network looks next depends on the sequence of words that has already been generated.

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1}) \quad (4)$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}. \quad (5)$$

Once the weights (which sum to one) are computed, the context vector $\hat{z}_t$ is computed by

$$\hat{\mathbf{z}}_t = \phi\left(\{\mathbf{a}_i\}, \{\alpha_i\}\right), \quad (6)$$

where ϕ is a function that returns a single vector given the set of annotation vectors and their corresponding weights. The details of ϕ function are discussed in Sec. 4.

The initial memory state and hidden state of the LSTM are predicted by an average of the annotation vectors fed through two separate MLPs (init,c and init,h):

$$\mathbf{c}_0 = f_{\text{init,c}}\left(\frac{1}{L}\sum_{i}^{L} \mathbf{a}_i\right)$$

$$\mathbf{h}_0 = f_{\text{init,h}}\left(\frac{1}{L}\sum_{i}^{L} \mathbf{a}_i\right)$$

In this work, we use a deep output layer (Pascanu et al., 2014) to compute the output word probability given the LSTM state, the context vector and the previous word:

$$p(\mathbf{y}_t|\mathbf{a}, \mathbf{y}_1^{t-1}) \propto \exp(\mathbf{L}_o(\mathbf{E}\mathbf{y}_{t-1} + \mathbf{L}_h\mathbf{h}_t + \mathbf{L}_z\hat{\mathbf{z}}_t)) \quad (7)$$

Where $\mathbf{L}_o \in \mathbb{R}^{K \times m}$, $\mathbf{L}_h \in \mathbb{R}^{m \times n}$, $\mathbf{L}_z \in \mathbb{R}^{m \times D}$, and $\mathbf{E}$ are learned parameters initialized randomly.

4. Learning Stochastic "Hard" vs Deterministic "Soft" Attention

In this section we discuss two alternative mechanisms for the attention model f_{att}: stochastic attention and deterministic attention.

4.1. Stochastic "Hard" Attention

We represent the location variable s_t as where the model decides to focus attention when generating the t^{th} word. $s_{t,i}$ is an indicator one-hot variable which is set to 1 if the i-th location (out of L) is the one used to extract visual features. By treating the attention locations as intermediate latent variables, we can assign a multinoulli distribution parametrized by $\{\alpha_i\}$, and view $\hat{z}_t$ as a random variable:

$$p(s_{t,i} = 1 \mid s_{j<t}, \mathbf{a}) = \alpha_{t,i} \quad (8)$$

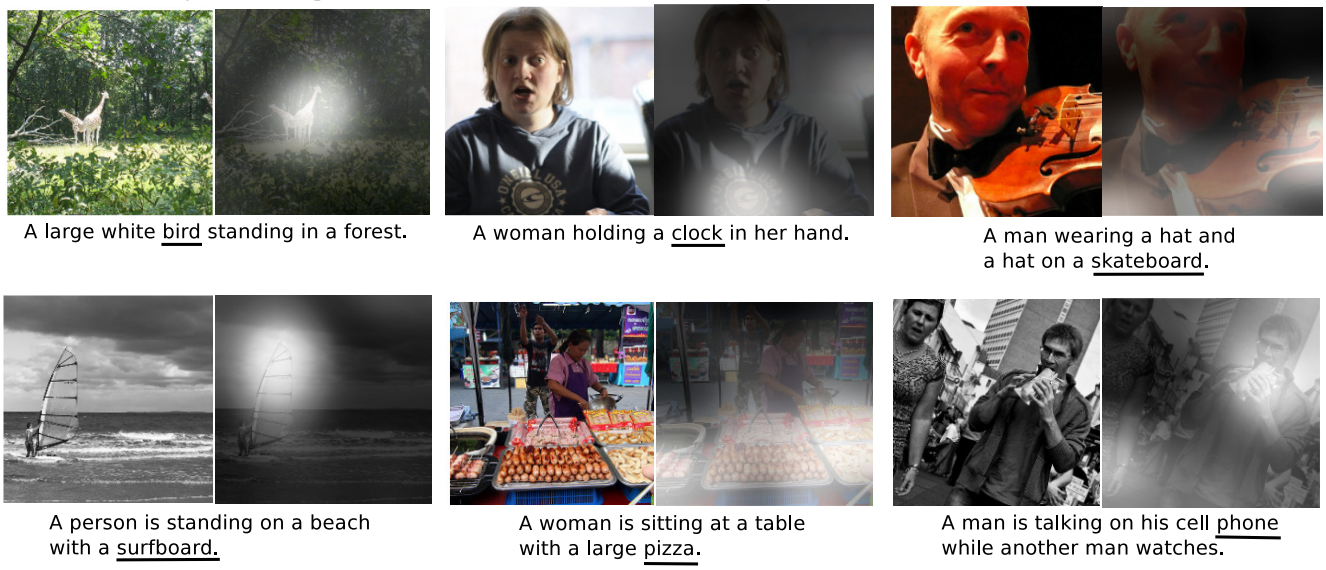
$$\hat{\mathbf{z}}_t = \sum_i s_{t,i}\mathbf{a}_i. \quad (9)$$

We define a new objective function L_s that is a variational lower bound on the marginal log-likelihood $\log p(\mathbf{y} \mid \mathbf{a})$ of observing the sequence of words $\mathbf{y}$ given image features $\mathbf{a}$. The learning algorithm for the parameters W of the models can be derived by directly optimizing L_s:

$$\begin{aligned} L_s &= \sum_s p(s \mid \mathbf{a}) \log p(\mathbf{y} \mid s, \mathbf{a}) \\ &\leq \log \sum_s p(s \mid \mathbf{a}) p(\mathbf{y} \mid s, \mathbf{a}) \\ &= \log p(\mathbf{y} \mid \mathbf{a}) \end{aligned} \quad (10)$$

$$\frac{\partial L_s}{\partial W} = \sum_s p(s \mid \mathbf{a}) \left[\frac{\partial \log p(\mathbf{y} \mid s, \mathbf{a})}{\partial W} + \log p(\mathbf{y} \mid s, \mathbf{a}) \frac{\partial \log p(s \mid \mathbf{a})}{\partial W} \right]. \quad (11)$$

Figure 5. Examples of mistakes where we can use attention to gain intuition into what the model saw.



Equation 11 suggests a Monte Carlo based sampling approximation of the gradient with respect to the model parameters. This can be done by sampling the location s_t from a multinouilli distribution defined by Equation 8.

$$\tilde{s_t} \sim \text{Multinoulli}_L(\{\alpha_i\})$$

$$\frac{\partial L_s}{\partial W} \approx \frac{1}{N}\sum_{n=1}^{N}\Big[\frac{\partial \log p(\mathbf{y} \mid \tilde{s}^n, \mathbf{a})}{\partial W} + \log p(\mathbf{y} \mid \tilde{s}^n, \mathbf{a})\frac{\partial \log p(\tilde{s}^n \mid \mathbf{a})}{\partial W}\Big] \quad (12)$$

A moving average baseline is used to reduce the variance in the Monte Carlo estimator of the gradient, following Weaver & Tao (2001). Similar, but more complicated variance reduction techniques have previously been used by Mnih et al. (2014) and Ba et al. (2014). Upon seeing the k^{th} mini-batch, the moving average baseline is estimated as an accumulated sum of the previous log likelihoods with exponential decay:

$$b_k = 0.9 \times b_{k-1} + 0.1 \times \log p(\mathbf{y} \mid \tilde{s}_k, \mathbf{a})$$

To further reduce the estimator variance, an entropy term on the multinouilli distribution $H[s]$ is added. Also, with probability 0.5 for a given image, we set the sampled attention location $\tilde{s}$ to its expected value α. Both techniques improve the robustness of the stochastic attention learning algorithm. The final learning rule for the model is then the following:

$$\frac{\partial L_s}{\partial W} \approx \frac{1}{N}\sum_{n=1}^{N}\Big[\frac{\partial \log p(\mathbf{y} \mid \tilde{s}^n, \mathbf{a})}{\partial W} + \lambda_r(\log p(\mathbf{y} \mid \tilde{s}^n, \mathbf{a}) - b)\frac{\partial \log p(\tilde{s}^n \mid \mathbf{a})}{\partial W} + \lambda_e\frac{\partial H[\tilde{s}^n]}{\partial W}\Big]$$

where, λ_r and λ_e are two hyper-parameters set by cross-validation. As pointed out and used in Ba et al. (2014) and Mnih et al. (2014), this is formulation is equivalent to the REINFORCE learning rule (Williams, 1992), where the reward for the attention choosing a sequence of actions is a real value proportional to the log likelihood of the target sentence under the sampled attention trajectory.

In making a hard choice at every point, $\phi(\{\mathbf{a}_i\}, \{\alpha_i\})$ from Equation 6 is a function that returns a sampled $\mathbf{a}_i$ at every point in time based upon a multinouilli distribution parameterized by α.

4.2. Deterministic "Soft" Attention

Learning stochastic attention requires sampling the attention location s_t each time, instead we can take the expectation of the context vector $\hat{\mathbf{z}}_t$ directly,

$$\mathbb{E}_{p(s_t|a)}[\hat{\mathbf{z}}_t] = \sum_{i=1}^{L}\alpha_{t,i}\mathbf{a}_i \quad (13)$$

and formulate a deterministic attention model by computing a soft attention weighted annotation vector $\phi(\{\mathbf{a}_i\}, \{\alpha_i\}) = \sum_i^L \alpha_i \mathbf{a}_i$ as introduced by Bahdanau et al. (2014). This corresponds to feeding in a soft α

weighted context into the system. The whole model is smooth and differentiable under the deterministic attention, so learning end-to-end is trivial by using standard back-propagation.

Learning the deterministic attention can also be understood as approximately optimizing the marginal likelihood in Equation 10 under the attention location random variable s_t from Sec. 4.1. The hidden activation of LSTM $\mathbf{h}_t$ is a linear projection of the stochastic context vector $\hat{\mathbf{z}}_t$ followed by $\tanh$ non-linearity. To the first order Taylor approximation, the expected value $\mathbb{E}_{p(s_t|a)}[\mathbf{h}_t]$ is equal to computing $\mathbf{h}_t$ using a single forward prop with the expected context vector $\mathbb{E}_{p(s_t|a)}[\hat{\mathbf{z}}_t]$. Considering Eq. 7, let $\mathbf{n}_t = \mathbf{L}_o(\mathbf{E}\mathbf{y}_{t-1}+\mathbf{L}_h\mathbf{h}_t+\mathbf{L}_z\hat{\mathbf{z}}_t)$, $\mathbf{n}_{t,i}$ denotes $\mathbf{n}_t$ computed by setting the random variable $\hat{\mathbf{z}}$ value to $\mathbf{a}_i$. We define the normalized weighted geometric mean for the softmax k^{th} word prediction:

$$NWGM[p(y_t = k \mid \mathbf{a})] = \frac{\prod_i \exp(n_{t,k,i})^{p(s_{t,i}=1|a)}}{\sum_j \prod_i \exp(n_{t,j,i})^{p(s_{t,i}=1|a)}} = \frac{\exp(\mathbb{E}_{p(s_t|a)}[n_{t,k}])}{\sum_j \exp(\mathbb{E}_{p(s_t|a)}[n_{t,j}])}$$

The equation above shows the normalized weighted geometric mean of the caption prediction can be approximated well by using the expected context vector, where $\mathbb{E}[\mathbf{n}_t] = \mathbf{L}_o(\mathbf{E}\mathbf{y}_{t-1} + \mathbf{L}_h\mathbb{E}[\mathbf{h}_t] + \mathbf{L}_z\mathbb{E}[\hat{\mathbf{z}}_t])$. It shows that the NWGM of a softmax unit is obtained by applying softmax to the expectations of the underlying linear projections. Also, from the results in (Baldi & Sadowski, 2014), $NWGM[p(y_t = k \mid \mathbf{a})] \approx \mathbb{E}[p(y_t = k \mid \mathbf{a})]$ under softmax activation. That means the expectation of the outputs over all possible attention locations induced by random variable s_t is computed by simple feedforward propagation with expected context vector $\mathbb{E}[\hat{\mathbf{z}}_t]$. In other words, the deterministic attention model is an approximation to the marginal likelihood over the attention locations.

4.2.1. Doubly Stochastic Attention

By construction, $\sum_i \alpha_{ti} = 1$ as they are the output of a softmax. In training the deterministic version of our model we introduce a form of doubly stochastic regularization, where we also encourage $\sum_t \alpha_{ti} \approx 1$. This can be interpreted as encouraging the model to pay equal attention to every part of the image over the course of generation. In our experiments, we observed that this penalty was important quantitatively to improving overall BLEU score and that qualitatively this leads to more rich and descriptive captions. In addition, the soft attention model predicts a gating scalar β from previous hidden state $\mathbf{h}_{t-1}$ at each time step t, such that, $\phi\left(\{\mathbf{a}_i\}, \{\alpha_i\}\right) = \beta\sum_i^L \alpha_i\mathbf{a}_i$, where $\beta_t = \sigma(f_\beta(\mathbf{h}_{t-1}))$. We notice our attention weights put more emphasis on the objects in the images by including the scalar β.

Concretely, the model is trained end-to-end by minimizing the following penalized negative log-likelihood:

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda\sum_i^L(1 - \sum_t^C \alpha_{ti})^2 \qquad (14)$$

4.3. Training Procedure

Both variants of our attention model were trained with stochastic gradient descent using adaptive learning rate algorithms. For the Flickr8k dataset, we found that RMSProp (Tieleman & Hinton, 2012) worked best, while for Flickr30k/MS COCO dataset we used the recently proposed Adam algorithm (Kingma & Ba, 2014) .

To create the annotations a_i used by our decoder, we used the Oxford VGGnet (Simonyan & Zisserman, 2014) pretrained on ImageNet without finetuning. In principle however, any encoding function could be used. In addition, with enough data, we could also train the encoder from scratch (or fine-tune) with the rest of the model. In our experiments we use the 14×14×512 feature map of the fourth convolutional layer before max pooling. This means our decoder operates on the flattened 196 × 512 (i.e $L \times D$) encoding.

As our implementation requires time proportional to the length of the longest sentence per update, we found training on a random group of captions to be computationally wasteful. To mitigate this problem, in preprocessing we build a dictionary mapping the length of a sentence to the corresponding subset of captions. Then, during training we randomly sample a length and retrieve a mini-batch of size 64 of that length. We found that this greatly improved convergence speed with no noticeable diminishment in performance. On our largest dataset (MS COCO), our soft attention model took less than 3 days to train on an NVIDIA Titan Black GPU.

In addition to dropout (Srivastava et al., 2014), the only other regularization strategy we used was early stopping on BLEU score. We observed a breakdown in correlation between the validation set log-likelihood and BLEU in the later stages of training during our experiments. Since BLEU is the most commonly reported metric, we used BLEU on our validation set for model selection.

In our experiments with soft attention, we also used Whetlab[1] (Snoek et al., 2012; 2014) in our Flickr8k experiments. Some of the intuitions we gained from hyperparameter regions it explored were especially important in our Flickr30k and COCO experiments.

We make our code for these models based in Theano

[1] https://www.whetlab.com/

Table 1. BLEU-1,2,3,4/METEOR metrics compared to other methods, † indicates a different split, (—) indicates an unknown metric, ∘ indicates the authors kindly provided missing metrics by personal communication, Σ indicates an ensemble, a indicates using AlexNet

		BLEU				
Dataset	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Flickr8k	Google NIC(Vinyals et al., 2014)$^{\dagger\Sigma}$	63	41	27	—	—
	Log Bilinear (Kiros et al., 2014a)$^{\circ}$	65.6	42.4	27.7	17.7	17.31
	Soft-Attention	**67**	44.8	29.9	19.5	18.93
	Hard-Attention	**67**	**45.7**	**31.4**	**21.3**	**20.30**
Flickr30k	Google NIC$^{\dagger\circ\Sigma}$	66.3	42.3	27.7	18.3	—
	Log Bilinear	60.0	38	25.4	17.1	16.88
	Soft-Attention	66.7	43.4	28.8	19.1	**18.49**
	Hard-Attention	**66.9**	**43.9**	**29.6**	**19.9**	18.46
COCO	CMU/MS Research (Chen & Zitnick, 2014)^{a}	—	—	—	—	20.41
	MS Research (Fang et al., 2014)$^{\dagger a}$	—	—	—	—	20.71
	BRNN (Karpathy & Li, 2014)$^{\circ}$	64.2	45.1	30.4	20.3	—
	Google NIC$^{\dagger\circ\Sigma}$	66.6	46.1	32.9	24.6	—
	Log Bilinear$^{\circ}$	70.8	48.9	34.4	24.3	20.03
	Soft-Attention	70.7	49.2	34.4	24.3	**23.90**
	Hard-Attention	**71.8**	**50.4**	**35.7**	**25.0**	23.04

(Bergstra et al., 2010) publicly available upon publication to encourage future research in this area.

5. Experiments

We describe our experimental methodology and quantitative results which validate the effectiveness of our model for caption generation.

5.1. Data

We report results on the popular Flickr8k and Flickr30k dataset which has 8,000 and 30,000 images respectively as well as the more challenging Microsoft COCO dataset which has 82,783 images. The Flickr8k/Flickr30k dataset both come with 5 reference sentences per image, but for the MS COCO dataset, some of the images have references in excess of 5 which for consistency across our datasets we discard. We applied only basic tokenization to MS COCO so that it is consistent with the tokenization present in Flickr8k and Flickr30k. For all our experiments, we used a fixed vocabulary size of 10,000.

Results for our attention-based architecture are reported in Table 4.2.1. We report results with the frequently used BLEU metric[2] which is the standard in the caption generation literature. We report BLEU from 1 to 4 without a brevity penalty. There has been, however, criticism of BLEU, so in addition we report another common metric METEOR (Denkowski & Lavie, 2014), and compare whenever possible.

5.2. Evaluation Procedures

A few challenges exist for comparison, which we explain here. The first is a difference in choice of convolutional feature extractor. For identical decoder architectures, using more recent architectures such as GoogLeNet or Oxford VGG Szegedy et al. (2014), Simonyan & Zisserman (2014) can give a boost in performance over using the AlexNet (Krizhevsky et al., 2012). In our evaluation, we compare directly only with results which use the comparable GoogLeNet/Oxford VGG features, but for METEOR comparison we note some results that use AlexNet.

The second challenge is a single model versus ensemble comparison. While other methods have reported performance boosts by using ensembling, in our results we report a single model performance.

Finally, there is challenge due to differences between dataset splits. In our reported results, we use the predefined splits of Flickr8k. However, one challenge for the Flickr30k and COCO datasets is the lack of standardized splits. As a result, we report with the publicly available splits[3] used in previous work (Karpathy & Li, 2014). In our experience, differences in splits do not make a substantial difference in overall performance, but we note the differ-

[2]We verified that our BLEU evaluation code matches the authors of Vinyals et al. (2014), Karpathy & Li (2014) and Kiros et al. (2014b). For fairness, we only compare against results for which we have verified that our BLEU evaluation code is the same. With the upcoming release of the COCO evaluation server, we will include comparison results with all other recent image captioning models.

[3]http://cs.stanford.edu/people/karpathy/deepimagesent/

ences where they exist.

5.3. Quantitative Analysis

In Table 4.2.1, we provide a summary of the experiment validating the quantitative effectiveness of attention. We obtain state of the art performance on the Flickr8k, Flickr30k and MS COCO. In addition, we note that in our experiments we are able to significantly improve the state of the art performance METEOR on MS COCO that we speculate is connected to some of the regularization techniques we used 4.2.1 and our lower level representation. Finally, we also note that we are able to obtain this performance using a single model without an ensemble.

5.4. Qualitative Analysis: Learning to attend

By visualizing the attention component learned by the model, we are able to add an extra layer of interpretability to the output of the model (see Fig. 1). Other systems that have done this rely on object detection systems to produce candidate alignment targets (Karpathy & Li, 2014). Our approach is much more flexible, since the model can attend to "non object" salient regions.

The 19-layer OxfordNet uses stacks of 3x3 filters meaning the only time the feature maps decrease in size are due to the max pooling layers. The input image is resized so that the shortest side is 256 dimensional with preserved aspect ratio. The input to the convolutional network is the center cropped 224x224 image. Consequently, with 4 max pooling layers we get an output dimension of the top convolutional layer of 14x14. Thus in order to visualize the attention weights for the soft model, we simply upsample the weights by a factor of $2^4 = 16$ and apply a Gaussian filter. We note that the receptive fields of each of the 14x14 units are highly overlapping.

As we can see in Figure 2 and 3, the model learns alignments that correspond very strongly with human intuition. Especially in the examples of mistakes, we see that it is possible to exploit such visualizations to get an intuition as to why those mistakes were made. We provide a more extensive list of visualizations in Appendix A for the reader.

6. Conclusion

We propose an attention based approach that gives state of the art performance on three benchmark datasets using the BLEU and METEOR metric. We also show how the learned attention can be exploited to give more interpretability into the models generation process, and demonstrate that the learned alignments correspond very well to human intuition. We hope that the results of this paper will encourage future work in using visual attention. We also expect that the modularity of the encoder-decoder approach combined with attention to have useful applications in other domains.

Acknowledgments

The authors would like to thank the developers of Theano (Bergstra et al., 2010; Bastien et al., 2012). We acknowledge the support of the following organizations for research funding and computing support: the Nuance Foundation, NSERC, Samsung, Calcul Québec, Compute Canada, the Canada Research Chairs and CIFAR. The authors would also like to thank Nitish Srivastava for assistance with his ConvNet package as well as preparing the Oxford convolutional network and Relu Patrascu for helping with numerous infrastructure related problems.

References

Ba, Jimmy Lei, Mnih, Volodymyr, and Kavukcuoglu, Koray. Multiple object recognition with visual attention. *arXiv:*1412.7755, December 2014.

Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua. Neural machine translation by jointly learning to align and translate. *arXiv:*1409.0473, September 2014.

Baldi, Pierre and Sadowski, Peter. The dropout learning algorithm. *Artificial intelligence*, 210:78–122, 2014.

Bastien, Frederic, Lamblin, Pascal, Pascanu, Razvan, Bergstra, James, Goodfellow, Ian, Bergeron, Arnaud, Bouchard, Nicolas, Warde-Farley, David, and Bengio, Yoshua. Theano: new features and speed improvements. Submited to the Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

Bergstra, James, Breuleux, Olivier, Bastien, Frédéric, Lamblin, Pascal, Pascanu, Razvan, Desjardins, Guillaume, Turian, Joseph, Warde-Farley, David, and Bengio, Yoshua. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2010.

Chen, Xinlei and Zitnick, C Lawrence. Learning a recurrent visual representation for image caption generation. *arXiv preprint arXiv:1411.5654*, 2014.

Cho, Kyunghyun, van Merrienboer, Bart, Gulcehre, Caglar, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, October 2014.

Corbetta, Maurizio and Shulman, Gordon L. Control of goal-directed and stimulus-driven attention in the brain. *Nature reviews neuroscience*, 3(3):201–215, 2002.

Denil, Misha, Bazzani, Loris, Larochelle, Hugo, and de Freitas, Nando. Learning where to attend with deep architectures for image tracking. *Neural Computation*, 2012.

Denkowski, Michael and Lavie, Alon. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014.

Donahue, Jeff, Hendrikcs, Lisa Anne, Guadarrama, Segio, Rohrbach, Marcus, Venugopalan, Subhashini, Saenko, Kate, and Darrell, Trevor. Long-term recurrent convolutional networks for visual recognition and description. *arXiv:*1411.4389v2, November 2014.

Elliott, Desmond and Keller, Frank. Image description using visual dependency representations. In *EMNLP*, 2013.

Fang, Hao, Gupta, Saurabh, Iandola, Forrest, Srivastava, Rupesh, Deng, Li, Dollár, Piotr, Gao, Jianfeng, He, Xiaodong, Mitchell, Margaret, Platt, John, et al. From captions to visual concepts and back. *arXiv:*1411.4952, November 2014.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Hodosh, Micah, Young, Peter, and Hockenmaier, Julia. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, pp. 853–899, 2013.

Karpathy, Andrej and Li, Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *arXiv:*1412.2306, December 2014.

Kingma, Diederik P. and Ba, Jimmy. Adam: A Method for Stochastic Optimization. *arXiv:*1412.6980, December 2014.

Kiros, Ryan, Salahutdinov, Ruslan, and Zemel, Richard. Multimodal neural language models. In *International Conference on Machine Learning*, pp. 595–603, 2014a.

Kiros, Ryan, Salakhutdinov, Ruslan, and Zemel, Richard. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv:*1411.2539, November 2014b.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey. ImageNet classification with deep convolutional neural networks. In *NIPS*. 2012.

Kulkarni, Girish, Premraj, Visruth, Ordonez, Vicente, Dhar, Sagnik, Li, Siming, Choi, Yejin, Berg, Alexander C, and Berg, Tamara L. Babytalk: Understanding and generating simple image descriptions. *PAMI, IEEE Transactions on*, 35(12):2891–2903, 2013.

Kuznetsova, Polina, Ordonez, Vicente, Berg, Alexander C, Berg, Tamara L, and Choi, Yejin. Collective generation of natural image descriptions. In *Association for Computational Linguistics*. ACL, 2012.

Kuznetsova, Polina, Ordonez, Vicente, Berg, Tamara L, and Choi, Yejin. Treetalk: Composition and compression of trees for image descriptions. *TACL*, 2(10):351–362, 2014.

Larochelle, Hugo and Hinton, Geoffrey E. Learning to combine foveal glimpses with a third-order boltzmann machine. In *NIPS*, pp. 1243–1251, 2010.

Li, Siming, Kulkarni, Girish, Berg, Tamara L, Berg, Alexander C, and Choi, Yejin. Composing simple image descriptions using web-scale n-grams. In *Computational Natural Language Learning*. ACL, 2011.

Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence. Microsoft coco: Common objects in context. In *ECCV*, pp. 740–755. 2014.

Mao, Junhua, Xu, Wei, Yang, Yi, Wang, Jiang, and Yuille, Alan. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv:*1412.6632, December 2014.

Mitchell, Margaret, Han, Xufeng, Dodge, Jesse, Mensch, Alyssa, Goyal, Amit, Berg, Alex, Yamaguchi, Kota, Berg, Tamara, Stratos, Karl, and Daumé III, Hal. Midge: Generating image descriptions from computer vision detections. In *European Chapter of the Association for Computational Linguistics*, pp. 747–756. ACL, 2012.

Mnih, Volodymyr, Hees, Nicolas, Graves, Alex, and Kavukcuoglu, Koray. Recurrent models of visual attention. In *NIPS*, 2014.

Pascanu, Razvan, Gulcehre, Caglar, Cho, Kyunghyun, and Bengio, Yoshua. How to construct deep recurrent neural networks. In *ICLR*, 2014.

Rensink, Ronald A. The dynamic representation of scenes. *Visual cognition*, 7(1-3):17–42, 2000.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, Berg, Alexander C., and Fei-Fei, Li. ImageNet Large Scale Visual Recognition Challenge, 2014.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

Snoek, Jasper, Larochelle, Hugo, and Adams, Ryan P. Practical bayesian optimization of machine learning algorithms. In *NIPS*, pp. 2951–2959, 2012.

Snoek, Jasper, Swersky, Kevin, Zemel, Richard S, and Adams, Ryan P. Input warping for bayesian optimization of non-stationary functions. *arXiv preprint arXiv:1402.0929*, 2014.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15, 2014.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc VV. Sequence to sequence learning with neural networks. In *NIPS*, pp. 3104–3112, 2014.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

Tang, Yichuan, Srivastava, Nitish, and Salakhutdinov, Ruslan R. Learning generative models with visual attention. In *NIPS*, pp. 1808–1816, 2014.

Tieleman, Tijmen and Hinton, Geoffrey. Lecture 6.5 - rmsprop. Technical report, 2012.

Vinyals, Oriol, Toshev, Alexander, Bengio, Samy, and Erhan, Dumitru. Show and tell: A neural image caption generator. *arXiv:*1411.4555, November 2014.

Weaver, Lex and Tao, Nigel. The optimal reward baseline for gradient-based reinforcement learning. In *Proc. UAI'2001*, pp. 538–545, 2001.

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

Yang, Yezhou, Teo, Ching Lik, Daumé III, Hal, and Aloimonos, Yiannis. Corpus-guided sentence generation of natural images. In *EMNLP*, pp. 444–454. ACL, 2011.

Young, Peter, Lai, Alice, Hodosh, Micah, and Hockenmaier, Julia. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2:67–78, 2014.

Zaremba, Wojciech, Sutskever, Ilya, and Vinyals, Oriol. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, September 2014.

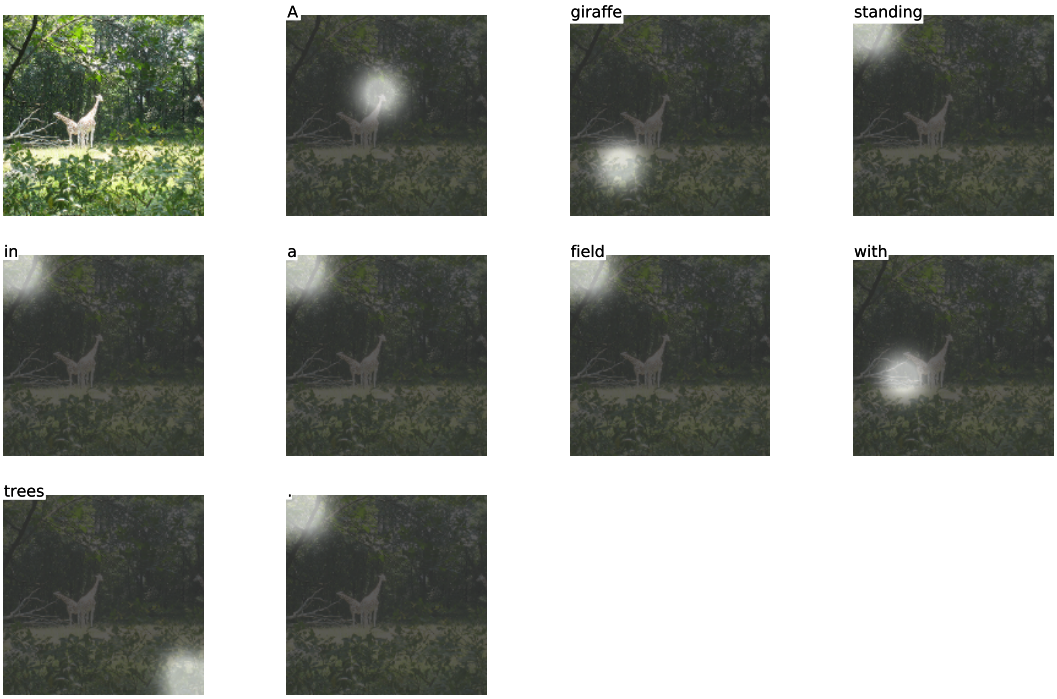
A. Appendix

Visualizations from our "hard" (a) and "soft" (b) attention model. *White* indicates the regions where the model roughly attends to (see section 5.4).

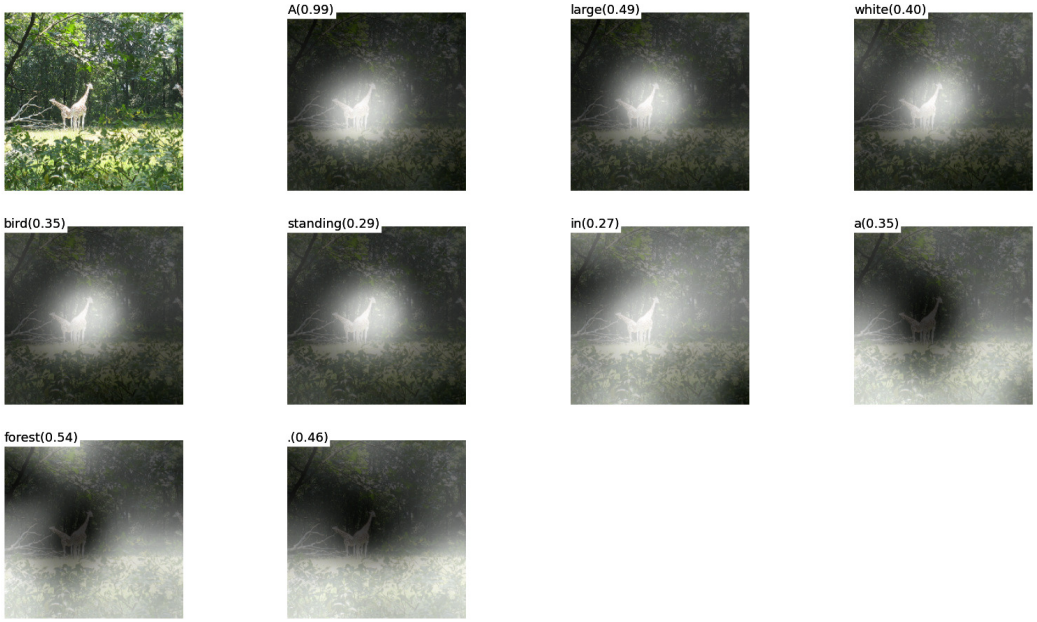
(a) A man and a woman playing frisbee in a field.

(b) A woman is throwing a frisbee in a park.

Figure 6.

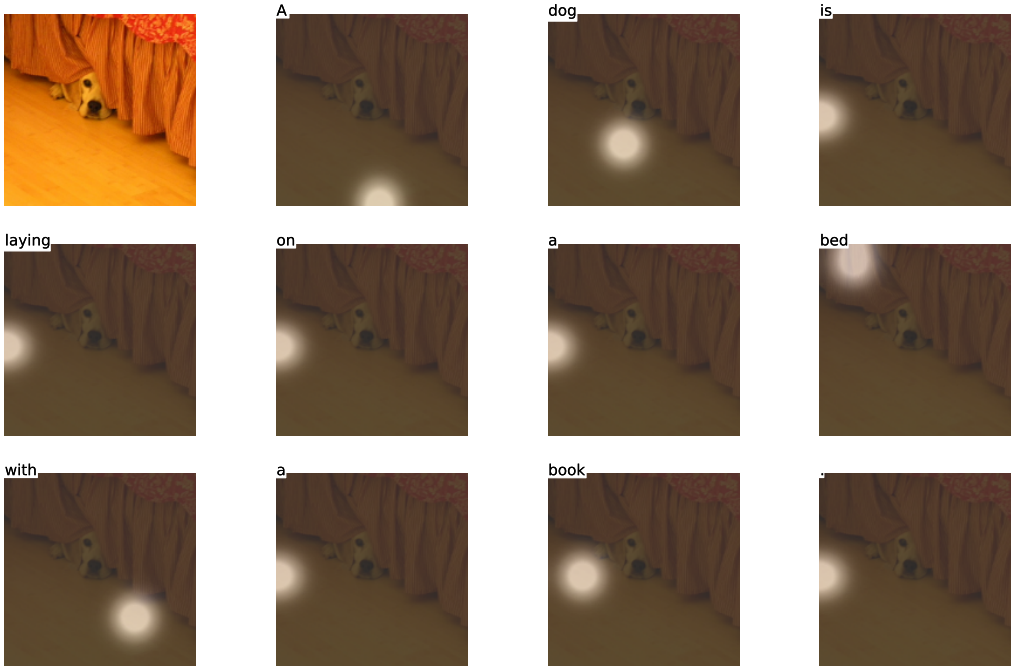


(a) A giraffe standing in the field with trees.

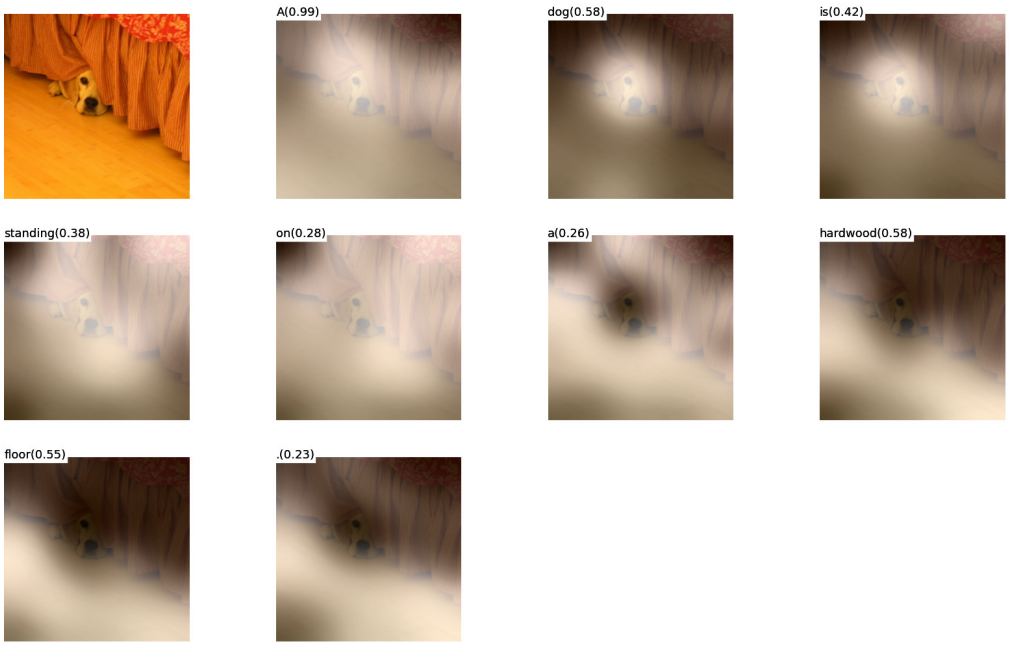


(b) A large white bird standing in a forest.

Figure 7.

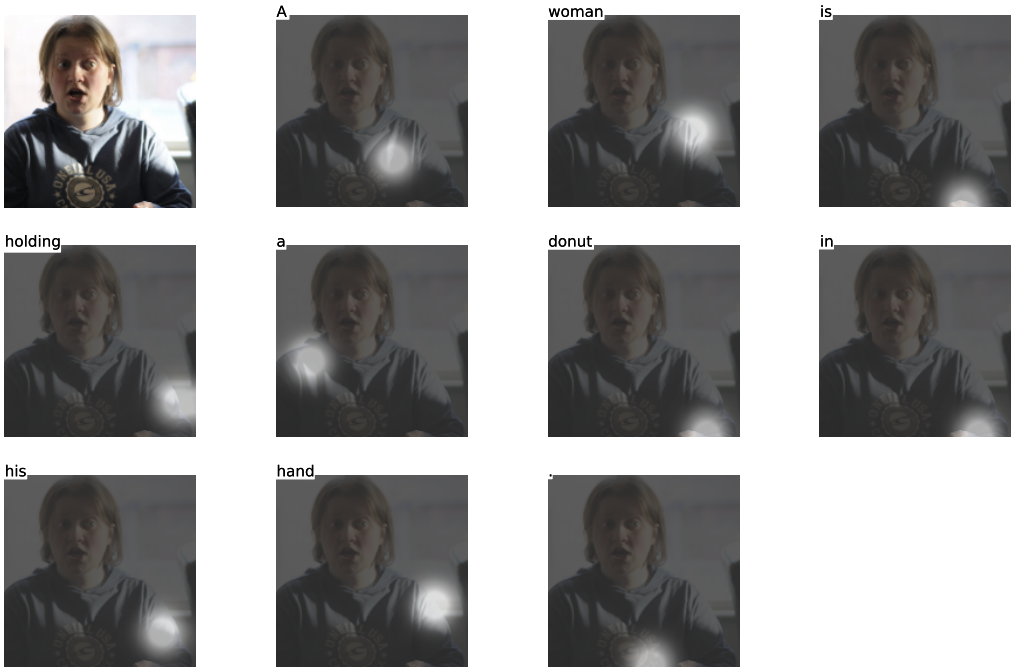


(a) A dog is laying on a bed with a book.



(b) A dog is standing on a hardwood floor.

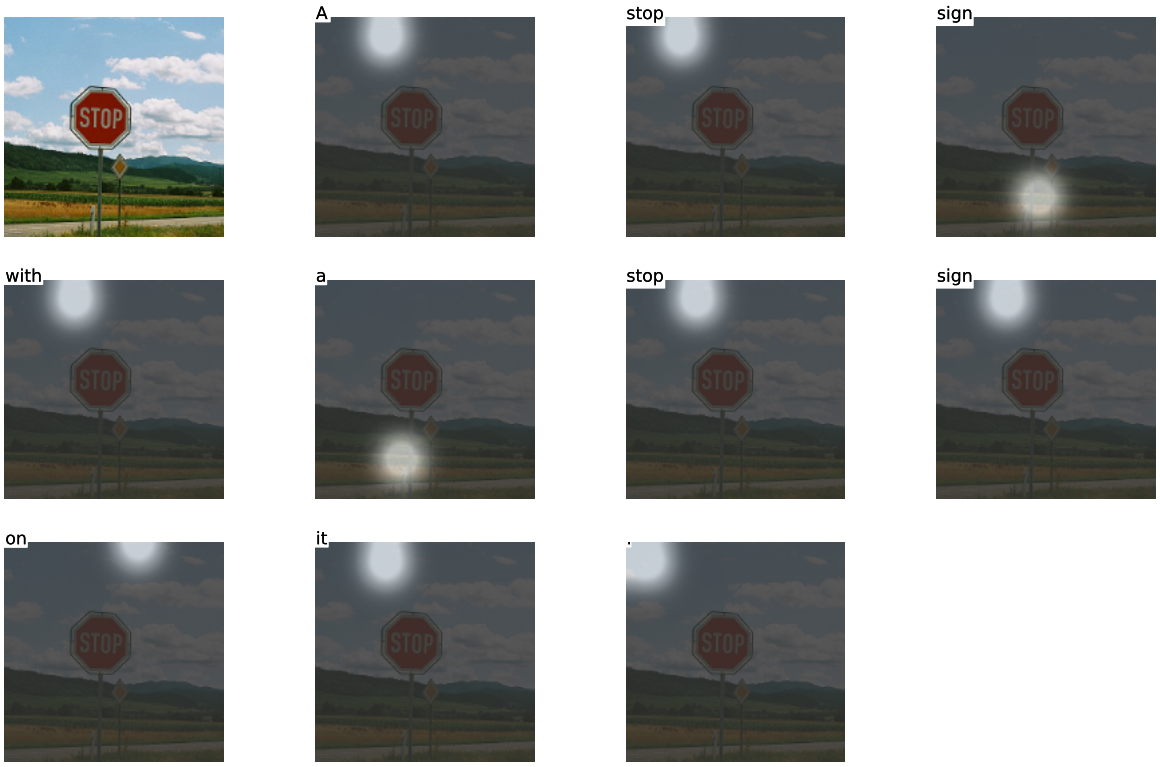
Figure 8.



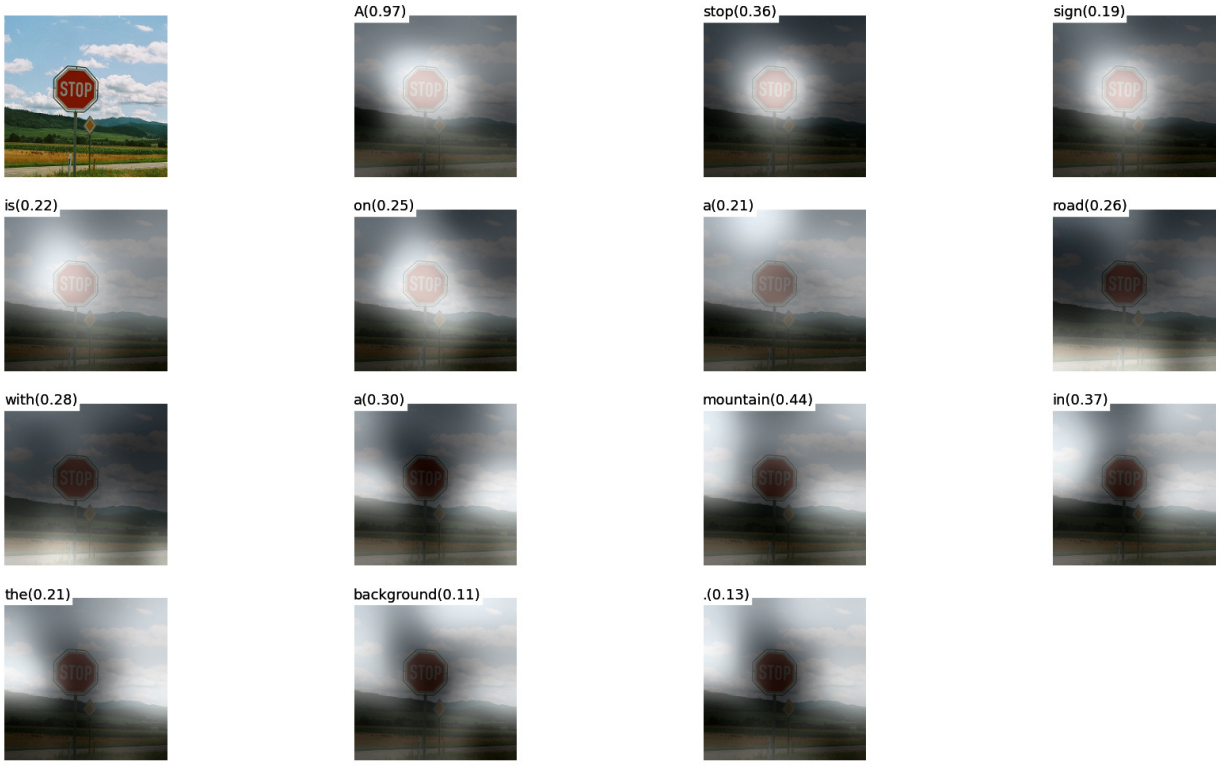
(a) A woman is holding a donut in his hand.

(b) A woman holding a clock in her hand.

Figure 9.

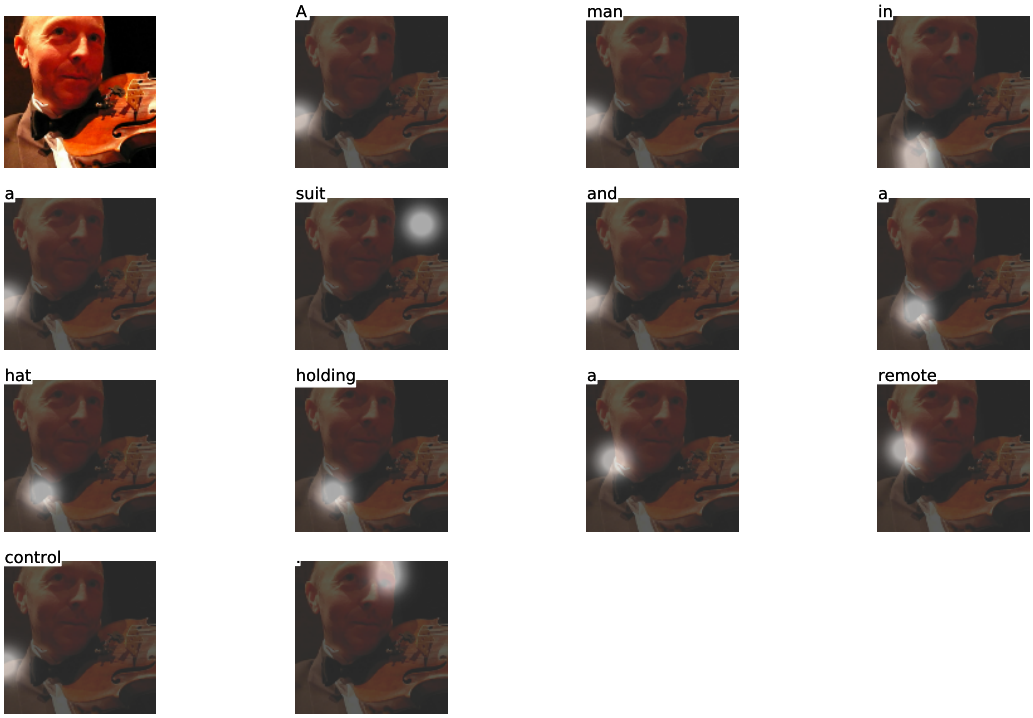


(a) A stop sign with a stop sign on it.

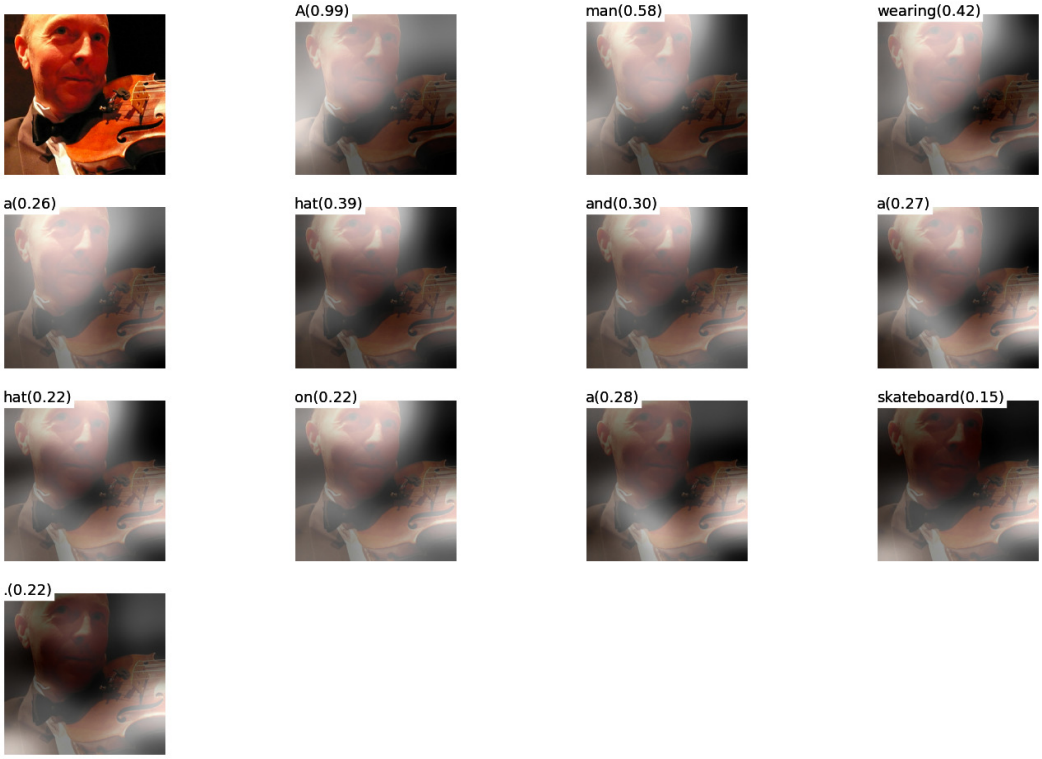


(b) A stop sign is on a road with a mountain in the background.

Figure 10.



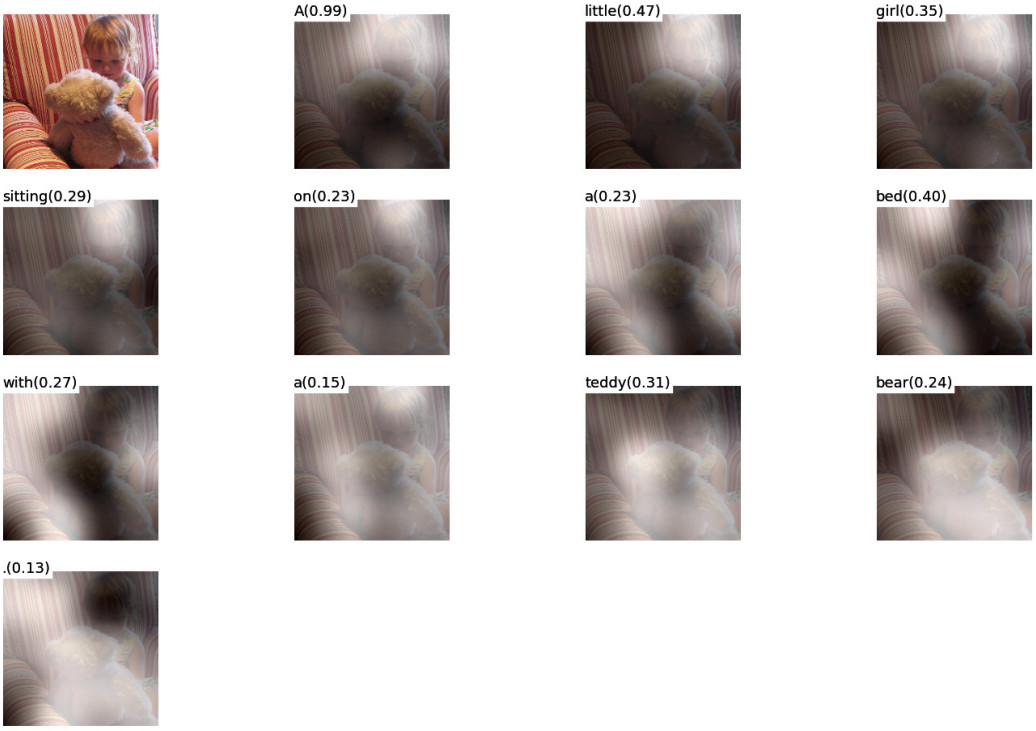
(a) A man in a suit and a hat holding a remote control.



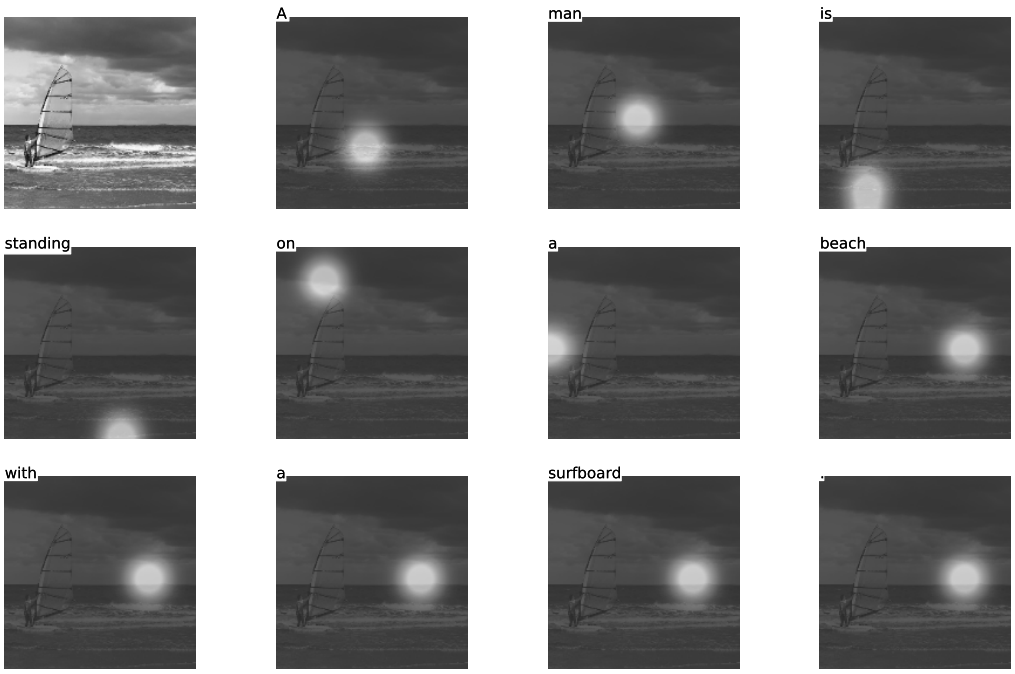
(b) A man wearing a hat and a hat on a skateboard.

Figure 11.

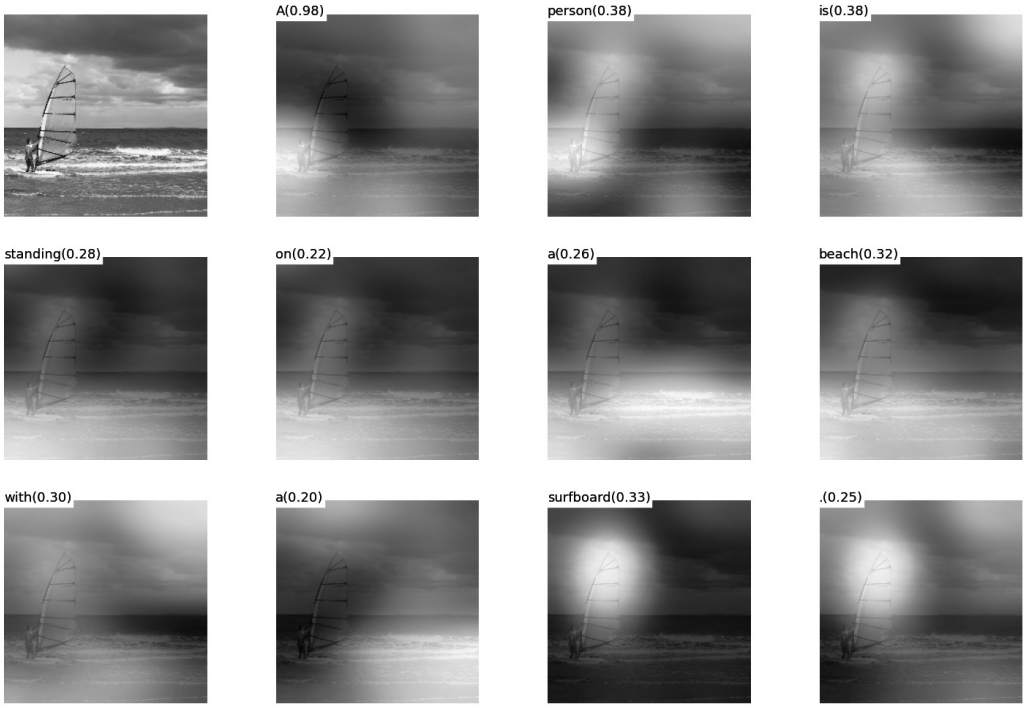
(a) A little girl sitting on a couch with a teddy bear.



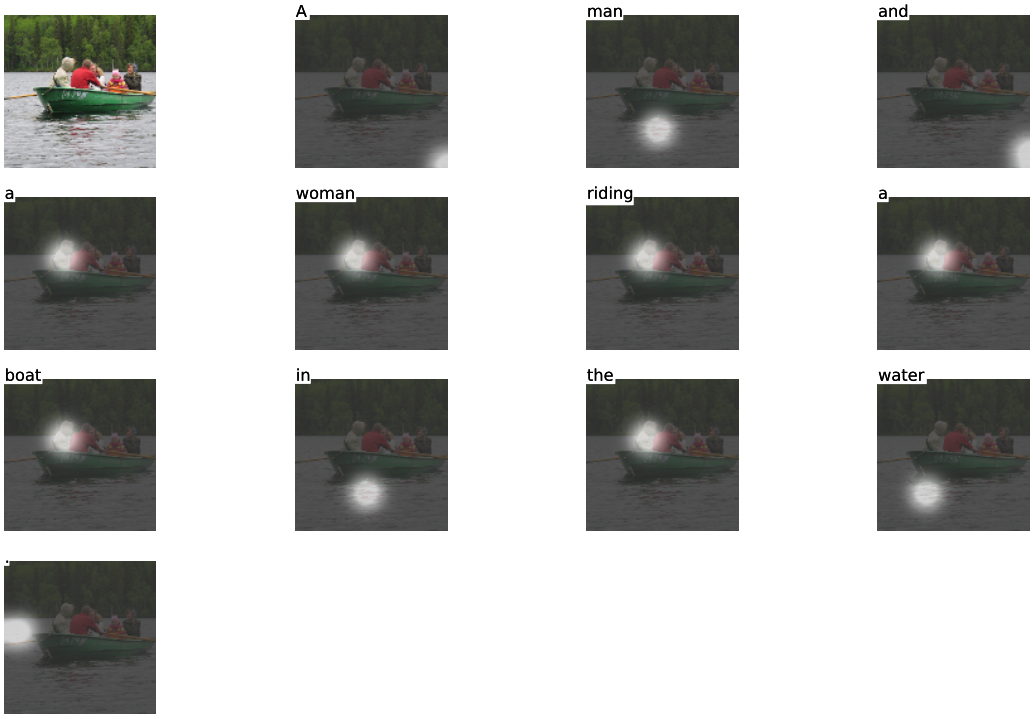
(b) A little girl sitting on a bed with a teddy bear.



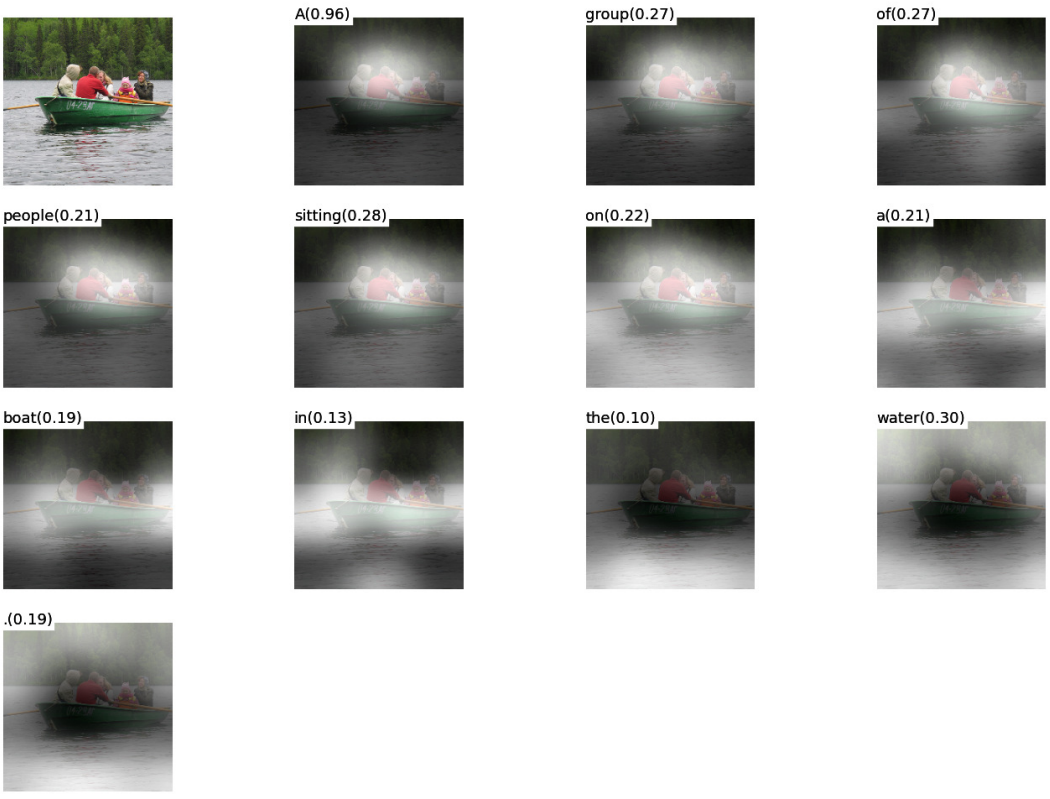
(a) A man is standing on a beach with a surfboard.



(b) A person is standing on a beach with a surfboard.

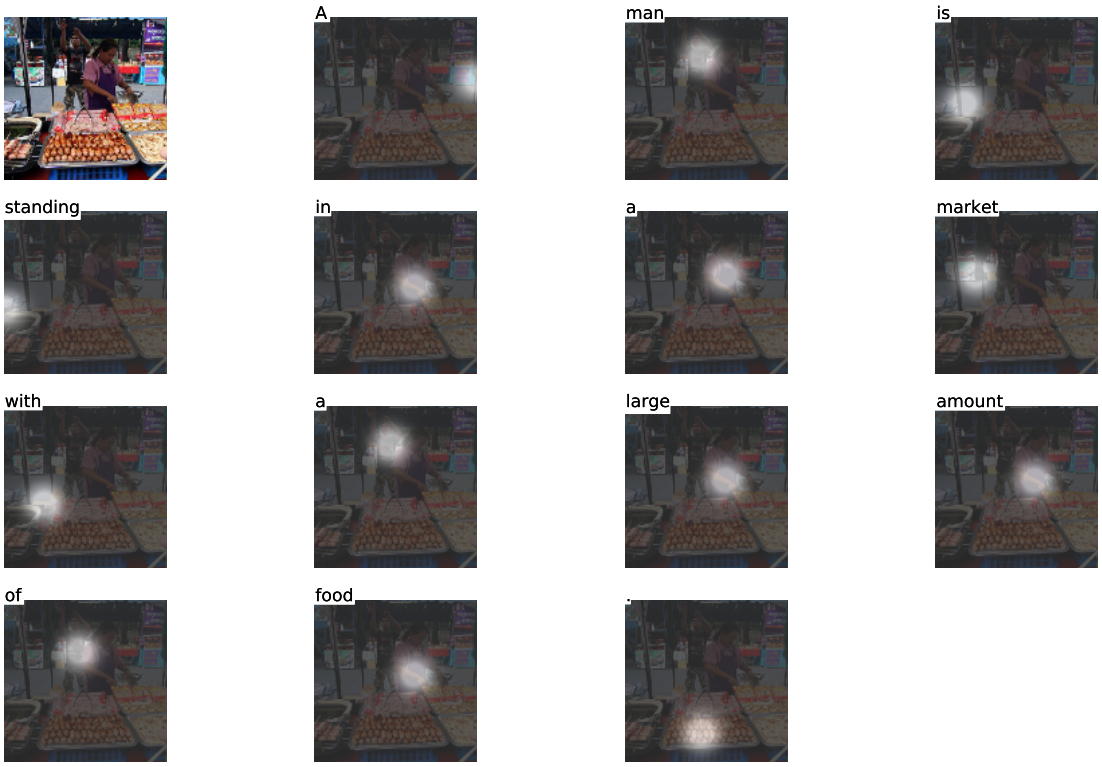


(a) A man and a woman riding a boat in the water.

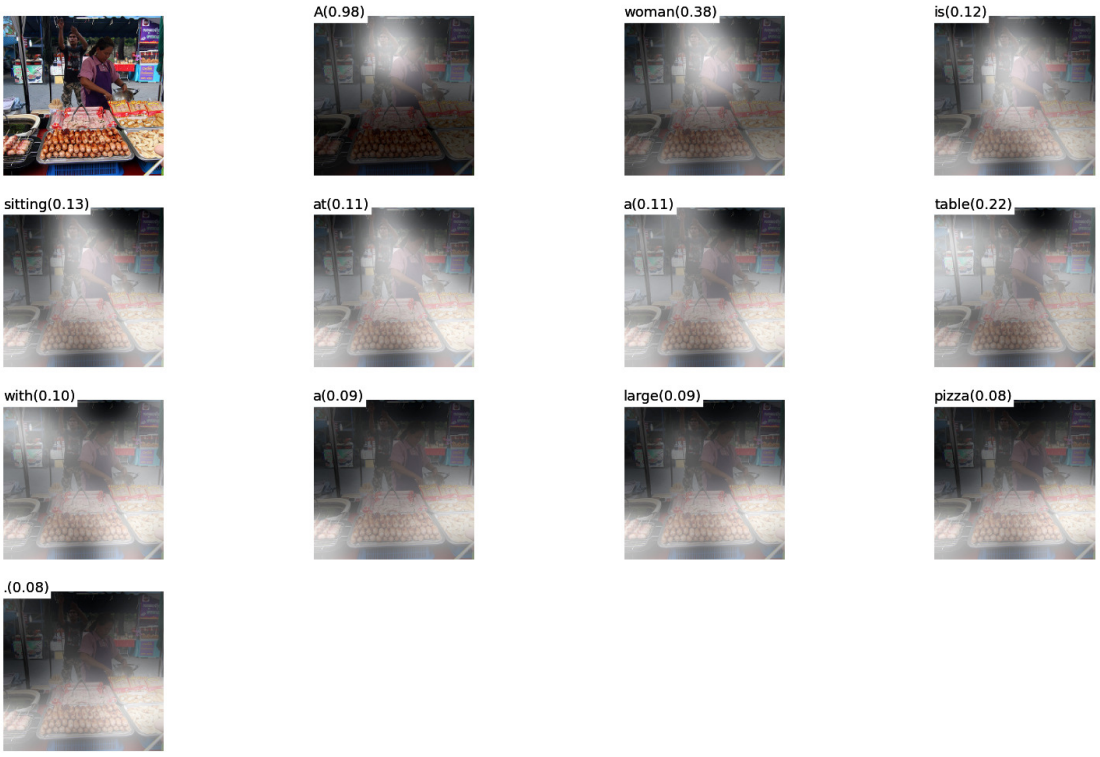


(b) A group of people sitting on a boat in the water.

Figure 12.

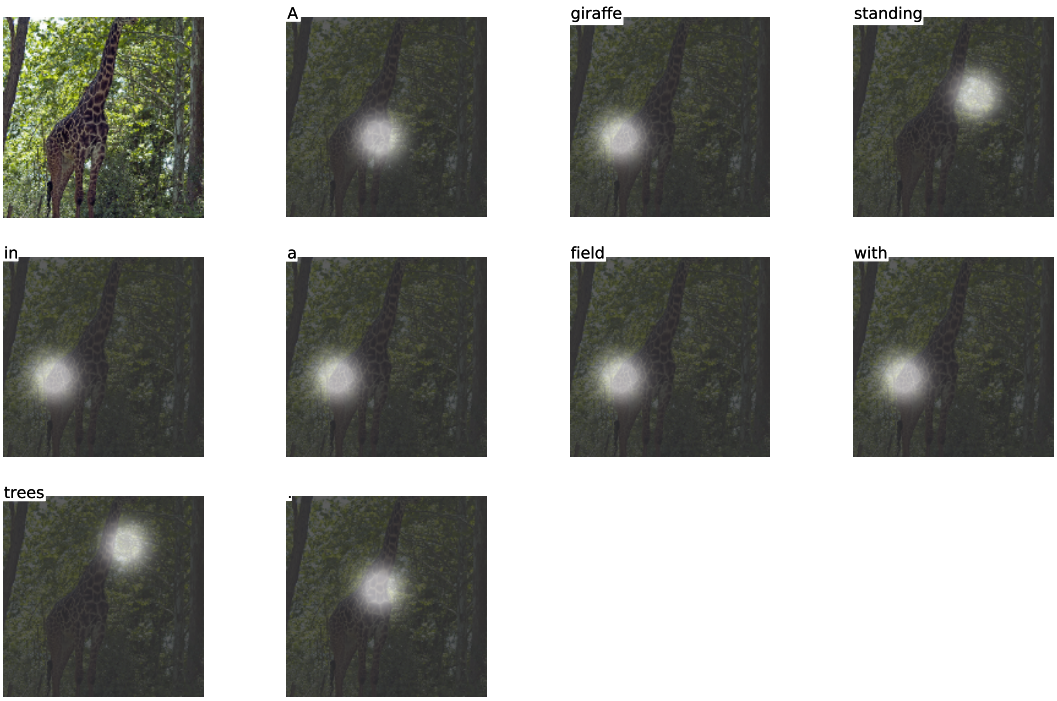


(a) A man is standing in a market with a large amount of food.

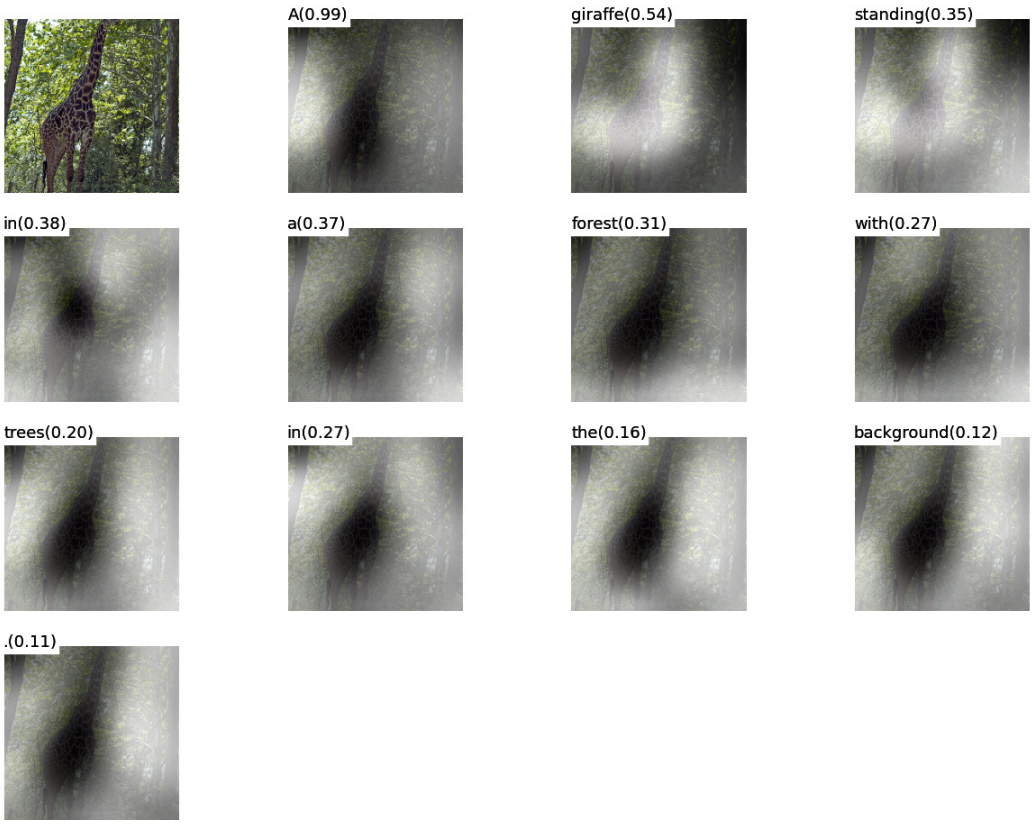


(b) A woman is sitting at a table with a large pizza.

Figure 13.

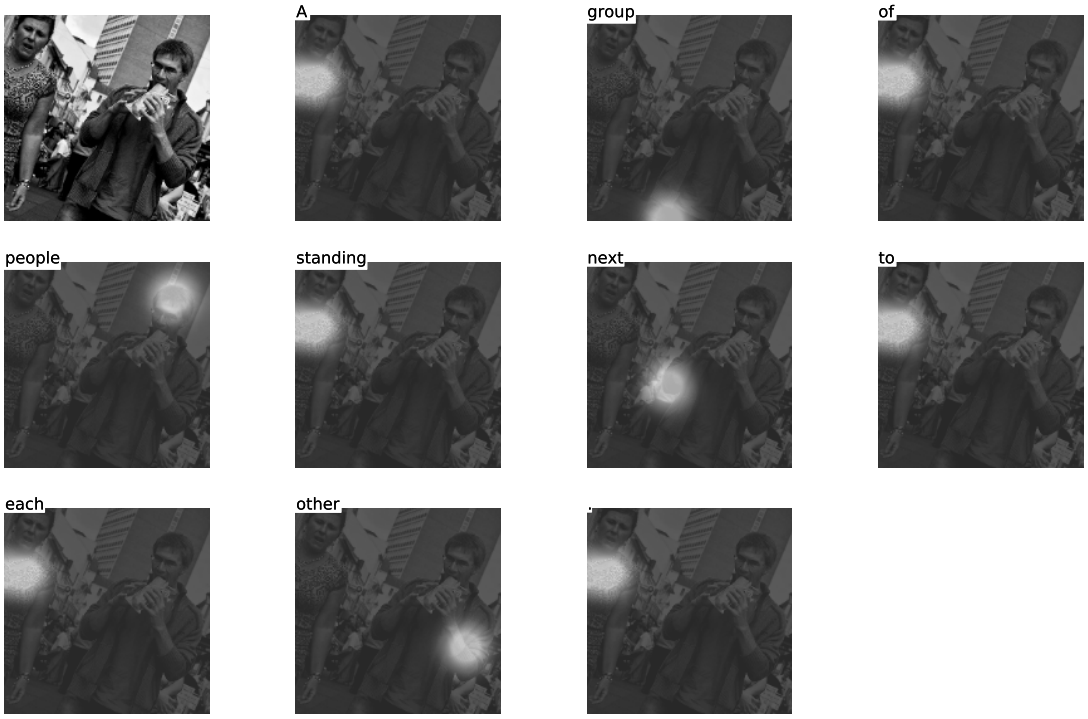


(a) A giraffe standing in a field with trees.

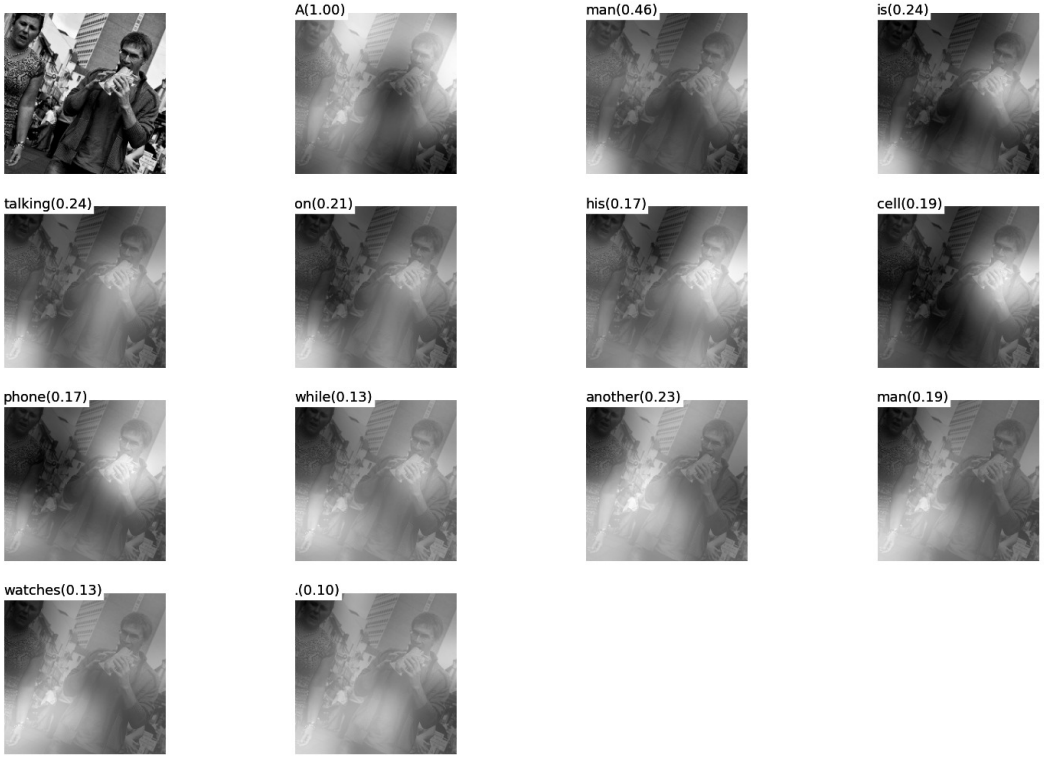


(b) A giraffe standing in a forest with trees in the background.

Figure 14.



(a) A group of people standing next to each other.



(b) A man is talking on his cell phone while another man watches.

Figure 15.