Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning

Jiasen Lu[2]*†, Caiming Xiong[1]†, Devi Parikh[3], Richard Socher[1]
[1]Salesforce Research, [2]Virginia Tech, [3]Georgia Institute of Technology
jiasenlu@vt.edu, parikh@gatech.edu, {cxiong, rsocher}@salesforce.com

Abstract

Attention-based neural encoder-decoder frameworks have been widely adopted for image captioning. Most methods force visual attention to be active for every generated word. However, the decoder likely requires little to no visual information from the image to predict non-visual words such as "the" and "of". Other words that may seem visual can often be predicted reliably just from the language model e.g., "sign" after "behind a red stop" or "phone" following "talking on a cell". In this paper, we propose a novel adaptive attention model with a visual sentinel. At each time step, our model decides whether to attend to the image (and if so, to which regions) or to the visual sentinel. The model decides whether to attend to the image and where, in order to extract meaningful information for sequential word generation. We test our method on the COCO image captioning 2015 challenge dataset and Flickr30K. Our approach sets the new state-of-the-art by a significant margin.

1. Introduction

Automatically generating captions for images has emerged as a prominent interdisciplinary research problem in both academia and industry. [8, 11, 18, 23, 27, 30]. It can aid visually impaired users, and make it easy for users to organize and navigate through large amounts of typically unstructured visual data. In order to generate high quality captions, the model needs to incorporate fine-grained visual clues from the image. Recently, visual attention-based neural encoder-decoder models [30, 11, 32] have been explored, where the attention mechanism typically produces a spatial map highlighting image regions relevant to each generated word.

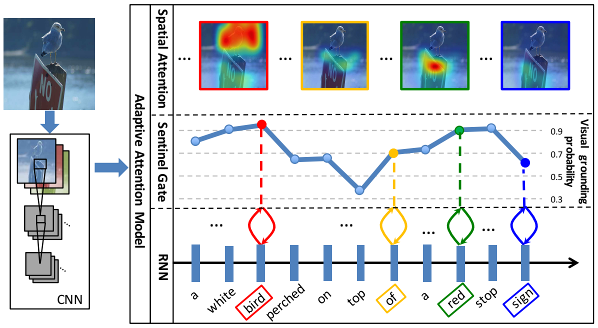


Figure 1: Our model learns an adaptive attention model that automatically determines when to look (**sentinel gate**) and where to look (**spatial attention**) for word generation, which are explained in section 2.2, 2.3 & 5.4.

Most attention models for image captioning and visual question answering attend to the image at every time step, irrespective of which word is going to be emitted next [31, 29, 17]. However, not all words in the caption have corresponding visual signals. Consider the example in Fig. 1 that shows an image and its generated caption "A white bird perched on top of a red stop sign". The words "a" and "of" do not have corresponding canonical visual signals. Moreover, language correlations make the visual signal unnecessary when generating words like "on" and "top" following "perched", and "sign" following "a red stop". In fact, gradients from non-visual words could mislead and diminish the overall effectiveness of the visual signal in guiding the caption generation process.

In this paper, we introduce an adaptive attention encoder-decoder framework which can automatically decide when to rely on visual signals and when to just rely on the language model. Of course, when relying on visual signals, the model also decides where – which image region – it should attend to. We first propose a novel spatial attention model for extracting spatial image features. Then as our proposed adaptive attention mechanism, we introduce a new Long Short Term Memory (LSTM) extension, which produces an additional "**visual sentinel**" vector instead of a single hidden state. The "visual sentinel", an additional latent representation of the decoder's memory, provides a fallback option to the decoder. We further design a new sentinel gate, which

[*] The major part of this work was done while J. Lu was an intern at Salesforce Research.

[†] Equal contribution

decides how much new information the decoder wants to get from the image as opposed to relying on the visual sentinel when generating the next word. For example, as illustrated in Fig. 1, our model learns to attend to the image more when generating words "white", "bird", "red" and "stop", and relies more on the visual sentinel when generating words "top", "of" and "sign".

Overall, the main contributions of this paper are:

- We introduce an adaptive encoder-decoder framework that automatically decides when to look at the image and when to rely on the language model to generate the next word.
- We first propose a new spatial attention model, and then build on it to design our novel adaptive attention model with "visual sentinel".
- Our model significantly outperforms other state-of-the-art methods on COCO and Flickr30k.
- We perform an extensive analysis of our adaptive attention model, including visual grounding probabilities of words and weakly supervised localization of generated attention maps.

2. Method

We first describe the generic neural encoder-decoder framework for image captioning in Sec. 2.1, then introduce our proposed attention-based image captioning models in Sec. 2.2 & 2.3.

2.1. Encoder-Decoder for Image Captioning

We start by briefly describing the encoder-decoder image captioning framework [27, 30]. Given an image and the corresponding caption, the encoder-decoder model directly maximizes the following objective:

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

where $\boldsymbol{\theta}$ are the parameters of the model, $\boldsymbol{I}$ is the image, and $\boldsymbol{y} = \{y_1, \ldots, y_t\}$ is the corresponding caption. Using the chain rule, the log likelihood of the joint probability distribution can be decomposed into ordered conditionals:

$$\log p(\boldsymbol{y}) = \sum_{t=1}^{T} \log p(y_t|y_1, \ldots, y_{t-1}, \boldsymbol{I}) \tag{2}$$

where we drop the dependency on model parameters for convenience.

In the encoder-decoder framework, with recurrent neural network (RNN), each conditional probability is modeled as:

$$\log p(y_t|y_1, \ldots, y_{t-1}, \boldsymbol{I}) = f(\boldsymbol{h}_t, \boldsymbol{c}_t) \tag{3}$$

where f is a nonlinear function that outputs the probability of y_t. $\boldsymbol{c}_t$ is the visual context vector at time t extracted from image $\boldsymbol{I}$. $\boldsymbol{h}_t$ is the hidden state of the RNN at time t. In this paper, we adopt Long-Short Term Memory (LSTM) instead of a vanilla RNN. The former have demonstrated state-of-the-art performance on a variety of sequence modeling tasks. $\boldsymbol{h}_t$ is modeled as:

$$\boldsymbol{h}_t = \text{LSTM}(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}, \boldsymbol{m}_{t-1}) \tag{4}$$

where $\boldsymbol{x}_t$ is the input vector. $\boldsymbol{m}_{t-1}$ is the memory cell vector at time $t-1$.

Commonly, context vector, $\boldsymbol{c}_t$ is an important factor in the neural encoder-decoder framework, which provides visual evidence for caption generation [18, 27, 30, 34]. These different ways of modeling the context vector fall into two categories: vanilla encoder-decoder and attention-based encoder-decoder frameworks:

- First, in the vanilla framework, $\boldsymbol{c}_t$ is only dependent on the encoder, a Convolutional Neural Network (CNN). The input image $\boldsymbol{I}$ is fed into the CNN, which extracts the last fully connected layer as a global image feature [18, 27]. Across generated words, the context vector $\boldsymbol{c}_t$ keeps constant, and does not depend on the hidden state of the decoder.
- Second, in the attention-based framework, $\boldsymbol{c}_t$ is dependent on both encoder and decoder. At time t, based on the hidden state, the decoder would attend to the specific regions of the image and compute $\boldsymbol{c}_t$ using the spatial image features from a convolution layer of a CNN. In [30, 34], they show that attention models can significantly improve the performance of image captioning.

To compute the context vector c_t, we first propose our spatial attention model in Sec. 2.2, then extend the model to an adaptive attention model in Sec. 2.3.

2.2. Spatial Attention Model

First, we propose a spatial attention model for computing the context vector $\boldsymbol{c}_t$ which is defined as:

$$\boldsymbol{c}_t = g(\boldsymbol{V}, \boldsymbol{h}_t) \tag{5}$$

where g is the attention function, $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k], \boldsymbol{v}_i \in \mathcal{R}^d$ is the spatial image features, each of which is a d dimensional representation corresponding to a part of the image. $\boldsymbol{h}_t$ is the hidden state of RNN at time t.

Given the spatial image feature $\boldsymbol{V} \in \mathcal{R}^{d\times k}$ and hidden state $\boldsymbol{h}_t \in \mathcal{R}^d$ of the LSTM, we feed them through a single layer neural network followed by a softmax function to generate the attention distribution over the k regions of the image:

$$\begin{aligned} \boldsymbol{z}_t &= \boldsymbol{w}_h^T \tanh(\boldsymbol{W}_v \boldsymbol{V} + (\boldsymbol{W}_g \boldsymbol{h}_t)\mathbb{1}^T) \qquad (6) \\ \boldsymbol{\alpha}_t &= \text{softmax}(\boldsymbol{z}_t) \qquad (7) \end{aligned}$$

where $\mathbb{1} \in \mathcal{R}^k$ is a vector with all elements set to 1. $\boldsymbol{W}_v, \boldsymbol{W}_g \in \mathcal{R}^{k\times d}$ and $\boldsymbol{w}_h \in \mathcal{R}^k$ are parameters to be

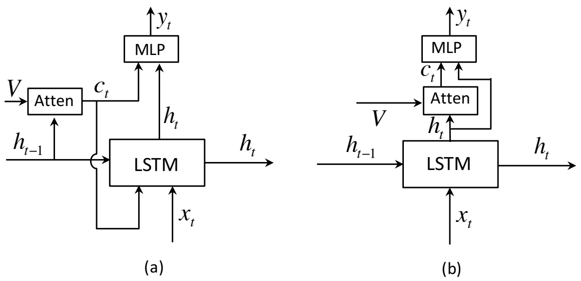


Figure 2: A illustration of soft attention model from [30] (a) and our proposed spatial attention model (b).

learnt. $\boldsymbol{\alpha} \in \mathcal{R}^k$ is the attention weight over features in $\boldsymbol{V}$. Based on the attention distribution, the context vector $\boldsymbol{c}_t$ can be obtained by:

$$\boldsymbol{c}_t = \sum_{i=1}^{k} \boldsymbol{\alpha}_{ti} \boldsymbol{v}_{ti} \tag{8}$$

where $\boldsymbol{c}_t$ and $\boldsymbol{h}_t$ are combined to predict next word y_{t+1} as in Equation 3.

Different from [30], shown in Fig. 2, we use the current hidden state $\boldsymbol{h}_t$ to analyze where to look (i.e., generating the context vector $\boldsymbol{c}_t$), then combine both sources of information to predict the next word. Our motivation stems from the superior performance of residual network [10]. The generated context vector $\boldsymbol{c}_t$ could be considered as the residual visual information of current hidden state $\boldsymbol{h}_t$, which diminishes the uncertainty or complements the informativeness of the current hidden state for next word prediction. We also empirically find our spatial attention model performs better, as illustrated in Table 1.

2.3. Adaptive Attention Model

While spatial attention based decoders have proven to be effective for image captioning, they cannot determine when to rely on visual signal and when to rely on the language model. In this section, motivated from Merity *et al.* [19], we introduce a new concept – "visual sentinel", which is a latent representation of what the decoder already knows. With the "visual sentinel", we extend our spatial attention model, and propose an adaptive model that is able to determine whether it needs to attend the image to predict next word.

What is visual sentinel? The decoder's memory stores both long and short term visual and linguistic information. Our model learns to extract a new component from this that the model can fall back on when it chooses to not attend to the image. This new component is called the visual sentinel. And the gate that decides whether to attend to the image or to the visual sentinel is the sentinel gate. When the decoder RNN is an LSTM, we consider those information preserved in its memory cell. Therefore, we extend the LSTM to obtain the "visual sentinel" vector $\boldsymbol{s}_t$ by:

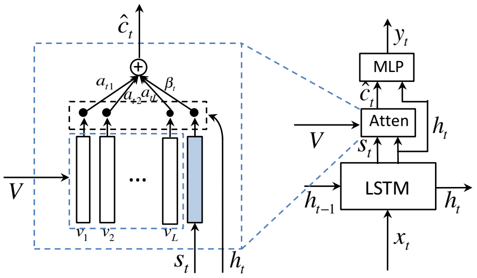


Figure 3: An illustration of the proposed model generating the t-th target word y_t given the image.

$$\boldsymbol{g}_t = \sigma\left(\boldsymbol{W}_x \boldsymbol{x}_t + \boldsymbol{W}_h \boldsymbol{h}_{t-1}\right) \tag{9}$$

$$\boldsymbol{s}_t = \boldsymbol{g}_t \odot \tanh\left(\boldsymbol{m}_t\right) \tag{10}$$

where $\boldsymbol{W}_x$ and $\boldsymbol{W}_h$ are weight parameters to be learned, $\boldsymbol{x}_t$ is the input to the LSTM at time step t, and $\boldsymbol{g}_t$ is the gate applied on the memory cell $\boldsymbol{m}_t$. $\odot$ represents the element-wise product and σ is the logistic sigmoid activation.

Based on the visual sentinel, we propose an adaptive attention model to compute the context vector. In our proposed architecture (see Fig. 3), our new adaptive context vector is defined as $\hat{\boldsymbol{c}}_t$, which is modeled as a mixture of the spatially attended image features (i.e. context vector of spatial attention model) and the visual sentinel vector. This trades off how much new information the network is considering from the image with what it already knows in the decoder memory (i.e., the visual sentinel). The mixture model is defined as follows:

$$\hat{\boldsymbol{c}}_t = \beta_t \boldsymbol{s}_t + (1 - \beta_t) \boldsymbol{c}_t \tag{11}$$

where β_t is the new sentinel gate at time t. In our mixture model, β_t produces a scalar in the range $[0, 1]$. A value of 1 implies that only the visual sentinel information is used and 0 means only spatial image information is used when generating the next word.

To compute the new sentinel gate β_t, we modified the spatial attention component. In particular, we add an additional element to z, the vector containing attention scores as defined in Equation 6. This element indicates how much "attention" the network is placing on the sentinel (as opposed to the image features). The addition of this extra element is summarized by converting Equation 7 to:

$$\hat{\boldsymbol{\alpha}}_t = \mathrm{softmax}([\boldsymbol{z}_t; \boldsymbol{w}_h^T \tanh(\boldsymbol{W}_s \boldsymbol{s}_t + (\boldsymbol{W}_g \boldsymbol{h}_t))]) \tag{12}$$

where $[\cdot;\cdot]$ indicates concatenation. $\boldsymbol{W}_s$ and $\boldsymbol{W}_g$ are weight parameters. Notably, $\boldsymbol{W}_g$ is the same weight parameter as in Equation 6. $\hat{\boldsymbol{\alpha}}_t \in \mathcal{R}^{k+1}$ is the attention distribution over both the spatial image feature as well as the visual sentinel vector. We interpret the last element of this vector to be the gate value: $\beta_t = \boldsymbol{\alpha}_t[k+1]$.

The probability over a vocabulary of possible words at time t can be calculated as:

$$\boldsymbol{p}_t = \mathrm{softmax}\left(\boldsymbol{W}_p(\hat{\boldsymbol{c}}_t + \boldsymbol{h}_t)\right) \tag{13}$$

where $\boldsymbol{W}_p$ is the weight parameters to be learnt.

This formulation encourages the model to adaptively attend to the image vs. the visual sentinel when generating the next word. The sentinel vector is updated at each time step. With this adaptive attention model, we call our framework the adaptive encoder-decoder image captioning framework.

3. Implementation Details

In this section, we describe the implementation details of our model and how we train our network.

Encoder-CNN. The encoder uses a CNN to get the representation of images. Specifically, the spatial feature outputs of the last convolutional layer of ResNet [10] are used, which have a dimension of 2048 × 7 × 7. We use $\boldsymbol{A} = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k\}, \boldsymbol{a}_i \in \mathcal{R}^{2048}$ to represent the spatial CNN features at each of the k grid locations. Following [10], the global image feature can be obtained by:

$$\boldsymbol{a}^g = \frac{1}{k}\sum_{i=1}^{k}\boldsymbol{a}_i \tag{14}$$

where $\boldsymbol{a}^g$ is the global image feature. For modeling convenience, we use a single layer perceptron with rectifier activation function to transform the image feature vector into new vectors with dimension d:

$$\boldsymbol{v}_i = \mathrm{ReLU}(\boldsymbol{W}_a\boldsymbol{a}_i) \tag{15}$$

$$\boldsymbol{v}^g = \mathrm{ReLU}(\boldsymbol{W}_b\boldsymbol{a}^g) \tag{16}$$

where $\boldsymbol{W}_a$ and $\boldsymbol{W}_g$ are the weight parameters. The transformed spatial image feature form $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k]$.

Decoder-RNN. We concatenate the word embedding vector $\boldsymbol{w}_t$ and global image feature vector $\boldsymbol{v}^g$ to get the input vector $\boldsymbol{x}_t = [\boldsymbol{w}_t; \boldsymbol{v}^g]$. We use a single layer neural network to transform the visual sentinel vector $\boldsymbol{s}_t$ and LSTM output vector $\boldsymbol{h}_t$ into new vectors that have the dimension d.

Training details. In our experiments, we use a single layer LSTM with hidden size of 512. We use the Adam optimizer with base learning rate of 5e-4 for the language model and 1e-5 for the CNN. The momentum and weight-decay are 0.8 and 0.999 respectively. We finetune the CNN network after 20 epochs. We set the batch size to be 80 and train for up to 50 epochs with early stopping if the validation CIDEr [26] score had not improved over the last 6 epochs. Our model can be trained within 30 hours on a single Titan X GPU. We use beam size of 3 when sampling the caption for both COCO and Flickr30k datasets.

4. Related Work

Image captioning has many important applications ranging from helping visually impaired users to human-robot interaction. As a result, many different models have been developed for image captioning. In general, those methods can be divided into two categories: template-based [9, 13, 14, 20] and neural-based [12, 18, 6, 3, 27, 7, 11, 30, 8, 34, 32, 33].

Template-based approaches generate caption templates whose slots are filled in based on outputs of object detection, attribute classification, and scene recognition. Farhadi *et al*. [9] infer a triplet of scene elements which is converted to text using templates. Kulkarni *et al*. [13] adopt a Conditional Random Field (CRF) to jointly reason across objects, attributes, and prepositions before filling the slots. [14, 20] use more powerful language templates such as a syntactically well-formed tree, and add descriptive information from the output of attribute detection.

Neural-based approaches are inspired by the success of sequence-to-sequence encoder-decoder frameworks in machine translation [4, 24, 2] with the view that image captioning is analogous to translating images to text. Kiros *et al*. [12] proposed a feed forward neural network with a multimodal log-bilinear model to predict the next word given the image and previous word. Other methods then replaced the feed forward neural network with a recurrent neural network [18, 3]. Vinyals *et al*. [27] use an LSTM instead of a vanilla RNN as the decoder. However, all these approaches represent the image with the last fully connected layer of a CNN. Karpathy *et al*. [11] adopt the result of object detection from R-CNN and output of a bidirectional RNN to learn a joint embedding space for caption ranking and generation.

Recently, attention mechanisms have been introduced to encoder-decoder neural frameworks in image captioning. Xu *et al*. [30] incorporate an attention mechanism to learn a latent alignment from scratch when generating corresponding words. [28, 34] utilize high-level concepts or attributes and inject them into a neural-based approach as semantic attention to enhance image captioning. Yang *et al*. [32] extend current attention encoder-decoder frameworks using a review network, which captures the global properties in a compact vector representation and are usable by the attention mechanism in the decoder. Yao *et al*. [33] present variants of architectures for augmenting high-level attributes from images to complement image representation for sentence generation.

To the best of our knowledge, ours is the first work to reason about when a model should attend to an image when

	Flickr30k						MS-COCO					
Method	B-1	B-2	B-3	B-4	METEOR	CIDEr	B-1	B-2	B-3	B-4	METEOR	CIDEr
DeepVS [11]	0.573	0.369	0.240	0.157	0.153	0.247	0.625	0.450	0.321	0.230	0.195	0.660
Hard-Attention [30]	0.669	0.439	0.296	0.199	0.185	-	0.718	0.504	0.357	0.250	0.230	-
ATT-FCN† [34]	0.647	0.460	0.324	0.230	0.189	-	0.709	0.537	0.402	0.304	0.243	-
ERD [32]	-	-	-	-	-	-	-	-	-	0.298	0.240	0.895
MSM† [33]	-	-	-	-	-	-	0.730	0.565	0.429	0.325	0.251	0.986
Ours-Spatial	0.644	0.462	0.327	0.231	0.202	0.493	0.734	0.566	0.418	0.304	0.257	1.029
Ours-Adaptive	**0.677**	**0.494**	**0.354**	**0.251**	**0.204**	**0.531**	**0.742**	**0.580**	**0.439**	**0.332**	**0.266**	**1.085**

Table 1: Performance on Flickr30k and COCO test splits. † indicates ensemble models. **B-n** is BLEU score that uses up to n-grams. Higher is better in all columns. For future comparisons, our ROUGE-L/SPICE Flickr30k scores are 0.467/0.145 and the COCO scores are 0.549/0.194.

	B-1		B-2		B-3		B-4		METEOR		ROUGE-L		CIDEr	
Method	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Google NIC [27]	0.713	0.895	0.542	0.802	0.407	0.694	0.309	0.587	0.254	0.346	0.530	0.682	0.943	0.946
MS Captivator [8]	0.715	0.907	0.543	0.819	0.407	0.710	0.308	0.601	0.248	0.339	0.526	0.680	0.931	0.937
m-RNN [18]	0.716	0.890	0.545	0.798	0.404	0.687	0.299	0.575	0.242	0.325	0.521	0.666	0.917	0.935
LRCN [7]	0.718	0.895	0.548	0.804	0.409	0.695	0.306	0.585	0.247	0.335	0.528	0.678	0.921	0.934
Hard-Attention [30]	0.705	0.881	0.528	0.779	0.383	0.658	0.277	0.537	0.241	0.322	0.516	0.654	0.865	0.893
ATT-FCN [34]	0.731	0.900	0.565	0.815	0.424	0.709	0.316	0.599	0.250	0.335	0.535	0.682	0.943	0.958
ERD [32]	0.720	0.900	0.550	0.812	0.414	0.705	0.313	0.597	0.256	0.347	0.533	0.686	0.965	0.969
MSM [33]	0.739	0.919	0.575	0.842	0.436	0.740	0.330	0.632	0.256	0.350	0.542	0.700	0.984	1.003
Ours-Adaptive	**0.748**	**0.920**	**0.584**	**0.845**	**0.444**	**0.744**	**0.336**	**0.637**	**0.264**	**0.359**	**0.550**	**0.705**	**1.042**	**1.059**

Table 2: Leaderboard of the published state-of-the-art image captioning models on the online COCO testing server. Our submission is a ensemble of 5 models trained with different initialization.

generating a sequence of words.

5. Results

5.1. Experiment Settings

We experiment with two datasets: Flickr30k [35] and COCO [16].

Flickr30k contains 31,783 images collected from Flickr. Most of these images depict humans performing various activities. Each image is paired with 5 crowd-sourced captions. We use the publicly available splits[1] containing 1,000 images for validation and test each.

COCO is the largest image captioning dataset, containing 82,783, 40,504 and 40,775 images for training, validation and test respectively. This dataset is more challenging, since most images contain multiple objects in the context of complex scenes. Each image has 5 human annotated captions. For offline evaluation, we use the same data split as in [32, 33, 34] containing 5000 images for validation and test each. For online evaluation on the COCO evaluation server, we reserve 2000 images from validation for development and the rest for training.

Pre-processing. We truncate captions longer than 18 words for COCO and 22 for Flickr30k. We then build a vocabulary of words that occur at least 5 and 3 times in the training set, resulting in 9567 and 7649 words for COCO and Flickr30k respectively.

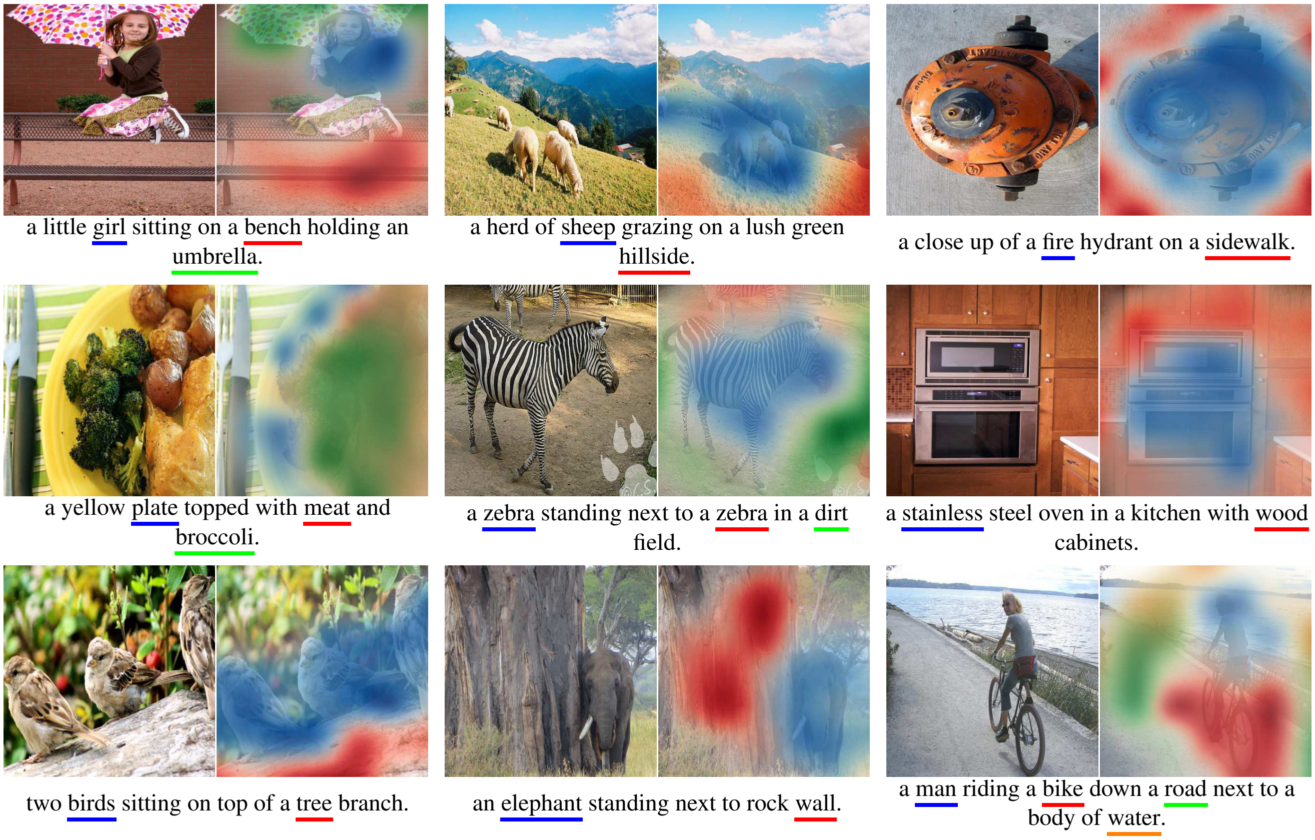
Compared Approaches: For offline evaluation on Flickr30k and COCO, we first compare our full model (**Ours-Adaptive**) with an ablated version (**Ours-Spatial**), which only performs the spatial attention. The goal of this comparison is to verify that our improvements are not the result of orthogonal contributions (e.g. better CNN features or better optimization). We further compare our method with **DeepVS** [11], **Hard-Attention** [30] and recently proposed **ATT** [34], **ERD** [32] and best performed method (LSTM-A_5) of **MSM** [33]. For online evaluation, we compare our method with **Google NIC** [27], **MS Captivator** [8], **m-RNN** [18], **LRCN** [7], **Hard-Attention** [30], **ATT-FCN** [34], **ERD** [32] and **MSM** [33].

5.2. Quantitative Analysis

We report results using the COCO captioning evaluation tool [16], which reports the following metrics: BLEU [21], Meteor [5], Rouge-L [15] and CIDEr [26]. We also report results using the new metric SPICE [1], which was found to better correlate with human judgments.

Table 1 shows results on the Flickr30k and COCO datasets. Comparing the full model w.r.t ablated versions without visual sentinel verifies the effectiveness of the pro-

[1] https://github.com/karpathy/neuraltalk



a little girl sitting on a bench holding an umbrella.

a herd of sheep grazing on a lush green hillside.

a close up of a fire hydrant on a sidewalk.

a yellow plate topped with meat and broccoli.

a zebra standing next to a zebra in a dirt field.

a stainless steel oven in a kitchen with wood cabinets.

two birds sitting on top of a tree branch.

an elephant standing next to rock wall.

a man riding a bike down a road next to a body of water.

Figure 4: Visualization of generated captions and image attention maps on the COCO dataset. Different colors show a correspondence between attended regions and underlined words. First 2 rows are success cases, last rows are failure examples. Best viewed in color.

posed framework. Our adaptive attention model significantly outperforms spatial attention model, which improves the CIDEr score from 0.493/1.029 to 0.531/1.085 on Flickr30k and COCO respectively. When comparing with previous methods, we can see that our single model significantly outperforms all previous methods in all metrics. On COCO, our approach improves the state-of-the-art on BLEU-4 from 0.325 (MSM†) to 0.332, METEOR from 0.251 (MSM†) to 0.266, and CIDEr from 0.986 (MSM†) to 1.085. Similarly, on Flickr30k, our model improves the state-of-the-art with a large margin.

We compare our model to state-of-the-art systems on the COCO evaluation server in Table 2. We can see that our approach achieves the best performance on all metrics among the published systems. Notably, Google NIC, ERD and MSM use Inception-v3 [25] as the encoder, which has similar or better classification performance compared to ResNet [10] (which is what our model uses).

5.3. Qualitative Analysis

To better understand our model, we first visualize the spatial attention weight $\boldsymbol{\alpha}$ for different words in the generated caption. We simply upsample the attention weight to the image size (224×224) using bilinear interpolation. Fig. 4 shows generated captions and the spatial attention maps for specific words in the caption. First two columns are success examples and the last one column shows failure examples. We see that our model learns alignments that correspond strongly with human intuition. Note that even in cases where the model produces inaccurate captions, we see that our model does look at reasonable regions in the image – it just seems to not be able to count or recognize texture and fine-grained categories. We provide a more extensive list of visualizations in supplementary material.

We further visualize the sentinel gate as a caption is generated. For each word, we use $1 - \beta$ as its visual grounding probability. In Fig. 5, we visualize the generated caption, the visual grounding probability and the spatial attention map generated by our model for each word. Our model successfully learns to attend to the image less when generating non-visual words such as “of” and “a”. For visual words like “red”, “rose”, “doughnuts”, “woman” and “snowboard”, our model assigns a high visual grounding probabilities (over 0.9). Note that the same word may be

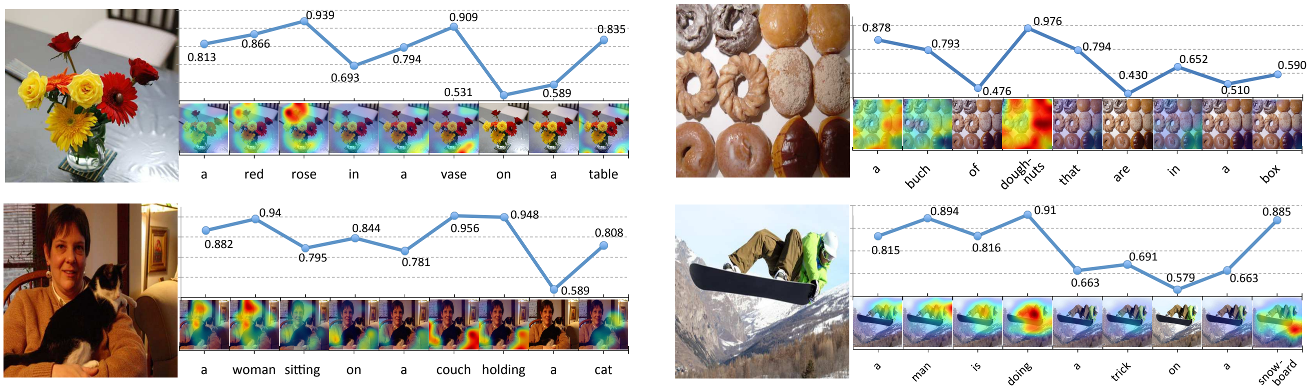


Figure 5: Visualization of generated captions, visual grounding probabilities of each generated word, and corresponding spatial attention maps produced by our model.

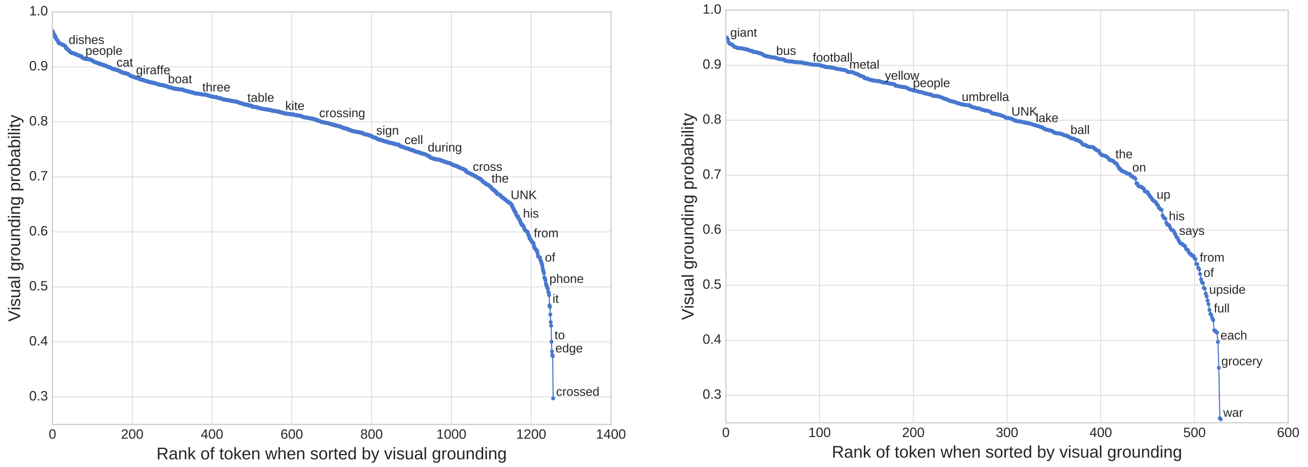


Figure 6: Rank-probability plots on COCO (left) and Flickr30k (right) indicating how likely a word is to be visually grounded when it is generated in a caption.

assigned different visual grounding probabilities when generated in different contexts. For example, the word “a” usually has a high visual grounding probability at the beginning of a sentence, since without any language context, the model needs the visual information to determine plurality (or not). On the other hand, the visual grounding probability of ”a” in the phrase “on a table” is much lower. Since it is unlikely for something to be on more than one table.

5.4. Adaptive Attention Analysis

In this section, we analysis the adaptive attention generated by our methods. We visualize the sentinel gate to understand “when” our model attends to the image as a caption is generated. We also perform a weakly-supervised localization on COCO categories by using the generated attention maps. This can help us to get an intuition of “where” our model attends, and whether it attends to the correct regions.

5.4.1 Learning “when” to attend

In order to assess whether our model learns to separate visual words in captions from non-visual words, we visualize the visual grounding probability. For each word in the vocabulary, we average the visual grounding probability over all the generated captions containing that word. Fig. 6 shows the rank-probability plot on COCO and Flickr30k.

We find that our model attends to the image more when generating object words like “dishes”, “people”, “cat”, “boat”; attribute words like “giant”, “metal”, “yellow” and number words like “three”. When the word is non-visual, our model learns to not attend to the image such as for “the”, “of”, “to” etc. For more abstract notions such as “crossing”, “during” etc., our model leans to attend less than the visual words and attend more than the non-visual words. Note that our model does not rely on any syntactic features or external knowledge. It discovers these trends automatically.

To quantify the visual grounding probability for the same

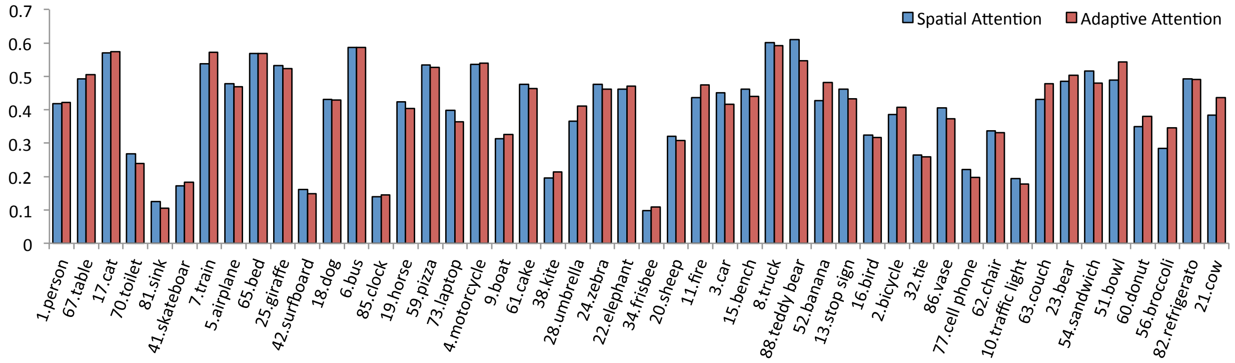


Figure 7: Localization accuracy over generated captions for top 45 most frequent COCO object categories. "Spatial Attention" and "Adaptive Attention" are our proposed spatial attention model and adaptive attention model, respectively. The COCO categories are ranked based on the align results of our adaptive attention, which cover 93.8% and 94.0% of total matched regions for spatial attention and adaptive attention, respectively.

words across COCO and Flickr30 datasets, we sort all common words between the two datasets by their visual grounding probabilities from both datasets. The rank correlation is 0.483. Words like "sheep" and "railing" have high visual grounding in COCO but not in Flickr30K, while "hair" and "run" are the reverse. Apart from different distributions of visual entities present in the dataset, some differences may be a consequence of different amounts of training data.

Our model cannot distinguish between words that are truly non-visual from the ones that are technically visual but have a high correlation with other words and hence chooses to not rely on the visual signal. For example, words such as "phone" get a relatively low visual grounding probability in our model. This is because it has a large language correlation with the word "cell".

5.4.2 Learning "where" to attend

We now assess whether our model attends to the correct spatial image regions. We perform weakly-supervised localization [22, 36] using the generated attention maps. To the best of our best knowledge, no previous works have used weakly supervised localization to evaluate spatial attention for image captioning. Given the word w_t and attention map $\boldsymbol{\alpha}_t$, we first segment the regions of of the image with attention values larger than th (after map is normalized to have the largest value be 1), where th is a per-class threshold estimated using the COCO validation split. Then we take the bounding box that covers the largest connected component in the segmentation map. We use intersection over union (IOU) of the generated and ground truth bounding box as the localization accuracy.

For each of the COCO object categories, we do a word-by-word match to align the generated words with the ground truth bounding box. For the object categories which has multiple words, such as "teddy bear", we take the maximum IOU score over the multiple words as its localization accuracy. We are able to align 5981 and 5924 regions for captions generated by the spatial and adaptive attention models respectively. The average localization accuracy for our spatial attention model is **0.362**, and **0.373** for our adaptive attention model. This demonstrates that as a byproduct, knowing when to attend also helps where to attend.

Fig. 7 shows the localization accuracy over the generated captions for top 45 most frequent COCO object categories. We can see that our spatial attention and adaptive attention models share similar trends. We observe that both models perform well on categories such as "cat", "bed", "bus" and "truck". On smaller objects, such as "sink", "surfboard", "clock" and "frisbee", both models perform relatively poorly. This is because our spatial attention maps are directly rescaled from a coarse 7 × 7 feature map, which looses a lot of spatial resolution and detail. Using a larger feature map may improve the performance.

6. Conclusion

In this paper, we present a novel adaptive attention encoder-decoder framework, which provides a fallback option to the decoder. We further introduce a new LSTM extension, which produces an additional "visual sentinel". Our model achieves state-of-the-art performance across standard benchmarks on image captioning. We perform extensive attention evaluation to analysis our adaptive attention. Though our model is evaluated on image captioning, it can have useful applications in other domains.

Acknowledgements This work was funded in part by an NSF CAREER award, ONR YIP award, Sloan Fellowship, ARO YIP award, Allen Distinguished Investigrator award from the Paul G. Allen Family Foundation, Google Faculty Research Award, Amazon Academic Research Award to DP

References

[1] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016. 5

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 4

[3] X. Chen and C. Lawrence Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *CVPR*, 2015. 4

[4] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 4

[5] M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *EACL 2014 Workshop on Statistical Machine Translation*, 2014. 5

[6] J. Devlin, S. Gupta, R. Girshick, M. Mitchell, and C. L. Zitnick. Exploring nearest neighbor approaches for image captioning. *arXiv preprint arXiv:1505.04467*, 2015. 4

[7] 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. 4, 5

[8] 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, 4, 5

[9] 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. 4

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 4, 6

[11] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 1, 4, 5

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

[13] 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. 4

[14] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi. Collective generation of natural image descriptions. In *ACL*, 2012. 4

[15] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *ACL 2004 Workshop*, 2004. 5

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

[17] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, 2016. 1

[18] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). In *ICLR*, 2015. 1, 2, 4, 5

[19] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. 3

[20] 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. 4

[21] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 5

[22] R. R.Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv:1611.01646*, 2016. 8

[23] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. 2014. 1

[24] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014. 4

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015. 6

[26] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 4, 5

[27] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 1, 2, 4, 5

[28] Q. Wu, C. Shen, L. Liu, A. Dick, and A. v. d. Hengel. What value do explicit high level concepts have in vision to language problems? *arXiv preprint arXiv:1506.01144*, 2015. 4

[29] C. Xiong, S. Merity, and R. Socher. Dynamic memory networks for visual and textual question answering. In *ICML*, 2016. 1

[30] 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, 2, 3, 4, 5

[31] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016. 1

[32] 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, 4, 5

[33] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei. Boosting image captioning with attributes. *arXiv preprint arXiv:1611.01646*, 2015. 4, 5

[34] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In *CVPR*, 2016. 2, 4, 5

[35] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *ACL*, 2014. 5

[36] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. *arXiv preprint arXiv:1512.04150*, 2015. 8