Image Caption Generation with Hierarchical Contextual Visual Spatial Attention

Mahmoud Khademi and Oliver Schulte
Simon Fraser University
Burnaby, BC, Canada
mkhademi@sfu.ca, oschulte@cs.sfu.ca

Abstract

We present a novel context-aware attention-based deep architecture for image caption generation. Our architecture employs a Bidirectional Grid LSTM, which takes visual features of an image as input and learns complex spatial patterns based on two-dimensional context, by selecting or ignoring its input. The Grid LSTM has not been applied to image caption generation task before. Another novel aspect is that we leverage a set of local region-grounded texts obtained by transfer learning. The region-grounded texts often describe the properties of the objects and their relationships in an image. To generate a global caption for the image, we integrate the spatial features from the Grid LSTM with the local region-grounded texts, using a two-layer Bidirectional LSTM. The first layer models the global scene context such as object presence. The second layer utilizes a novel dynamic spatial attention mechanism, based on another Grid LSTM, to generate the global caption word-by-word, while considering the caption context around a word in both directions. Unlike recent models that use a soft attention mechanism, our dynamic spatial attention mechanism considers the spatial context of the image regions. Experimental results on MS-COCO dataset show that our architecture outperforms the state-of-the-art.

1. Introduction

Automatically generating a description for an image is a fundamental problem in computer vision and scene understanding. This task is very challenging since not only a trained model must recognize objects in an image, but also it must represent the properties of the objects and their relationships in natural language. Recently, several research groups have developed visual recognition models based on Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) that significantly improved the quality of the generated captions [3, 18, 23, 6, 5, 16, 29, 28, 24, 27]. However, there is still a long way to go for an intelligent system to match the accuracy of human image descriptions.

One of the most important aspects of our visual recognition system is incorporating *contextual information* such as scene context (e.g. a farm), object presence, and object co-occurrence to describe a scene. For example, if we are asked to predict a caption for an image knowing only that it includes a `farm`, a `horse`, and a `human`, we will probably produce a sentence such as `a boy is riding the horse`, or `a man is next to the horse`. If we also consider the spatial relationship between the horse and the human, we can describe the image more accurately. Another crucial capability of our visual system is *visual spatial attention* [21], which directs attention to a particular location in a scene or image. Spatial attention enables us to give priority to a region within our visual field. The visual spatial attention is most important when the scene contains background clutter; humans do not describe everything in a scene, but instead look at the important regions and objects.

Inspired by these remarkable capabilities of the human visual system, we propose a hierarchical contextual attention-based deep architecture for image caption generation. The major components of our architecture function as follows. (1) Learn complex spatial patterns in a scene that aggregate local information from regions in both spatial directions. This component utilizes a recent model, called Grid LSTM, adding the advantages of a two-dimensional LSTM to a deep CNN. (2) Generate region-grounded texts for image regions using transfer learning from a region-grounded caption dataset. The region-grounded texts often describe the properties of the objects and their relationships in an image. (3) The Grid LSTM and region-grounded text generator provide informative spatial and textual features. Our main component integrates these two input modalities, to generate a caption using a novel dynamic spatial attention mechanism. This component utilizes a Deep Bidirectional LSTM. The Deep Bidirectional LSTM incorporates a new attention mechanism that selects relevant regions dynamically while generating a caption. This attention mechanism is implemented by (another) Grid LSTM. The Deep Bidirectional LSTM has a hierarchical

structure: the first layer models the global scene context such as scene class (e.g. a farm), presence of objects, and co-occurrence of the words which facilitates the image caption generation, while the second layer generates a caption which describes the image. Unlike recent attention-based models, which learn a simple fixed weight for each region of the image, the Grid LSTM allows our attention mechanism to take into account the two-dimensional spatial context and order of the image regions. Our components are carefully designed and connected to be effective for the image caption generation task. Our model outperforms the state-of-the-art performance on MS-COCO dataset. Lesion studies show the value added by the separate components. We will make the code and trained models available at https://github.com/khademi/Automatic-Image-Caption-Generation.

2. Related Work

Since image caption generation requires a comprehensive understanding of an image and capability to communicate that information via natural language, it is related to different areas in computer vision, machine learning, and natural language processing. On the language side, models based on RNNs have been shown to produce state-of-the-art results on various tasks. RNNs are suitable for sequential data of varying lengths and can learn complicated temporal dynamics. But, because of the vanishing gradient problem [9], they have difficulties in learning long-term dependencies. This drawback has been overcome by introducing LSTMs [9]. On the image side, CNNs such as ResNet, GoogLeNet and VGGNet have recently shown great success for visual recognition tasks such as image classification and object detection. These models are pre-trained on large image datasets such as ImageNet and are widely accessible. Recently, [11] proposed an extension to LSTM, called Grid LSTM, which can encode 2D signals such as images. They used Grid LSTM to classify digits. Our architecture introduces an extension of Grid LSTM which takes into account spatial context around an image region in all directions.

Several recent papers leverage the power of CNNs and RNNs for image caption generation problem [6, 2, 3, 23, 5, 16, 29, 18, 17, 1]. Most of these works represent an image using a feature vector at the very top layer of a pre-trained CNN. This approach may lose spatial information relevant to the caption. Moreover, since these models represent a whole image with a single feature vector, they are not robust to background clutter. [12] instead proposed a model based on a bidirectional RNN which scores the similarity between snippets of the caption, and the image regions generated by Region CNN object detectors [7]. However, they found that feeding the detected region features, instead of full image features, to their model deteriorates the caption generation performance. The main challenge is that some of the regions are not participating in the target caption. Also, it is difficult to find the right order to feed the image regions to the model at the test time. In this work, we provide a solution to these problems by representing an image with 2D feature maps, and introducing a new attention mechanism which can learn to focus on important regions of the image and ignore the other parts.

Closely related to our work, [26] proposed a model which can learn to fix its gaze on salient objects while generating the corresponding words. They computed a positive attention weight for each location in the input image using a multilayer perceptron conditioned on the previous generated word. In another related work, [29] employed an attention model to combine visual features and visual concepts such as words and objects in an RNN that generates the caption. Unlike these works, instead of applying a simple soft attention, we leverage the power of Grid LSTMs to dynamically attend to the important regions of an image. Also, our model considers spatial context along both vertical and horizontal directions in its attention mechanism.

3. Proposed Model

Our model has three components (see Figure 1): (1) A deep CNN for extracting image features. (2) A Bidirectional Grid LSTM (BiGrid LSTM) finds complex spatial patterns. This component considers only the image, not the words that already have been generated. (3) A caption generator which includes a Deep Bidirectional LSTM with a dynamic spatial attention mechanism, a word detector to represent global scene context, a region-grounded caption encoder, and a softmax layer to generate the next word in the caption. This component dynamically attends to different regions of the image while it generates the caption. We firstly describe the details of the model components. Then, we discuss training details.

3.1. Deep CNN for Extracting 2D Feature Maps

We apply a CNN to extract visual features. We experiment with the 16-layer VGGNet [22] and ResNet [8]. For VGGNet, we extract 512 $R \times S$ $(R, S = 7)$ feature maps of the last max pooling layer. The input to this CNN is an image of size 224×224. Thus, each $(224/R) \times (224/S)$ region (r, s) $(r = 1, ..., R)$ and $(s = 1, ..., S)$ is represented by a feature vector of size 512. For ResNet, we extract 1024 $R \times S$ $(R, S = 14)$ feature maps of layer *res4b35x*. Then, we project the feature vector that the CNN extracts from region (r, s) to obtain two vectors $\mathbf{x}_{r,s}$ and $\mathbf{x}'_{r,s}$ of size m, where m is 256 for VGGNet, and 512 for ResNet. These vectors are used as input to our Bidirectional Grid LSTM.

3.2. Bidirectional Grid LSTM

The outputs of the CNN, i.e. $\mathbf{x}_{r,s}$ and $\mathbf{x}'_{r,s}$, only describe location (r, s) of an input image, without considering the

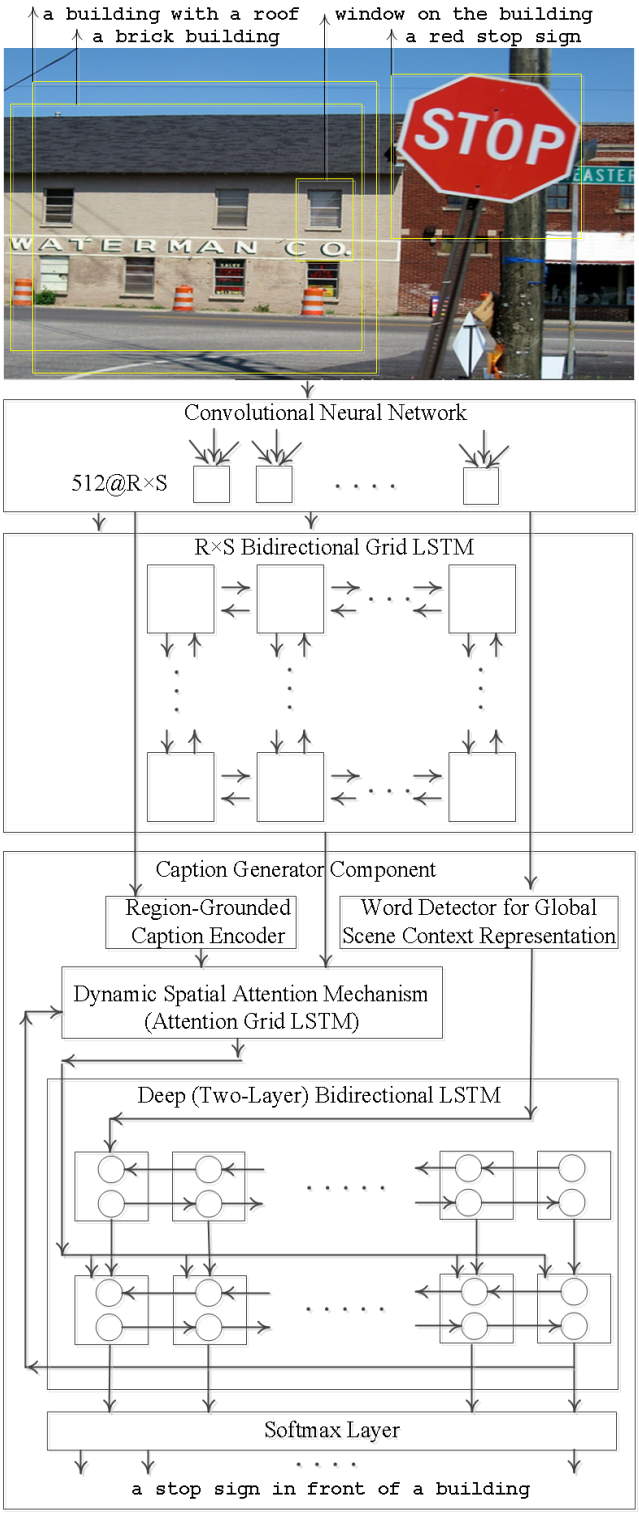

Figure 1. The architecture of our model

global information of the image in two spatial directions. Also, some of the image regions are not important for generating the caption. To address these issues, we propose to apply a Grid LSTM to an $R \times S$ grid, corresponding to the regions of the input image. A Grid LSTM is a grid of LSTM cells that can be applied to encode an image. The cells on the grid share the same trainable parameters. Figure 2 shows the basic architecture of a Grid LSTM cell. The cell executes computation with two LSTM cells along two spatial directions. The Grid LSTM with m' hidden states gets as input two input feature vectors $\mathbf{x}_{r,s}$ and $\mathbf{x}'_{r,s} \in \mathbb{R}^m$, two hidden vectors $\mathbf{h}_{r,s-1}$, $\mathbf{h}'_{r-1,s} \in \mathbb{R}^{m'}$, and two memory vectors $\mathbf{c}_{r,s-1}$, $\mathbf{c}'_{r-1,s} \in \mathbb{R}^{m'}$. It gets $\mathbf{h}_{r,s-1}$ and $\mathbf{c}_{r,s-1}$ from the previous Grid LSTM cell in horizontal direction. Also, $\mathbf{h}'_{r-1,s}$ and $\mathbf{c}'_{r-1,s}$ are coming from the previous Grid LSTM cell in vertical direction. A Grid LSTM cell consists of two input gates $\mathbf{i}_{r,s}$,$\mathbf{i}'_{r,s}$, two forget gates $\mathbf{f}_{r,s}$, $\mathbf{f}'_{r,s}$, two output gates $\mathbf{o}_{r,s}$,$\mathbf{o}'_{r,s}$, two input modulation gates $\mathbf{g}_{r,s}$, $\mathbf{g}'_{r,s}$, and two memory cells $\mathbf{c}_{r,s}$,$\mathbf{c}'_{r,s}$, corresponding to horizontal and vertical directions, respectively. The forget gates learn how much of the previous memory should be kept, and the input gates controls how much of the input should be read. Similarly, the output gates control how much of the memory cell should be carried to the hidden states. These

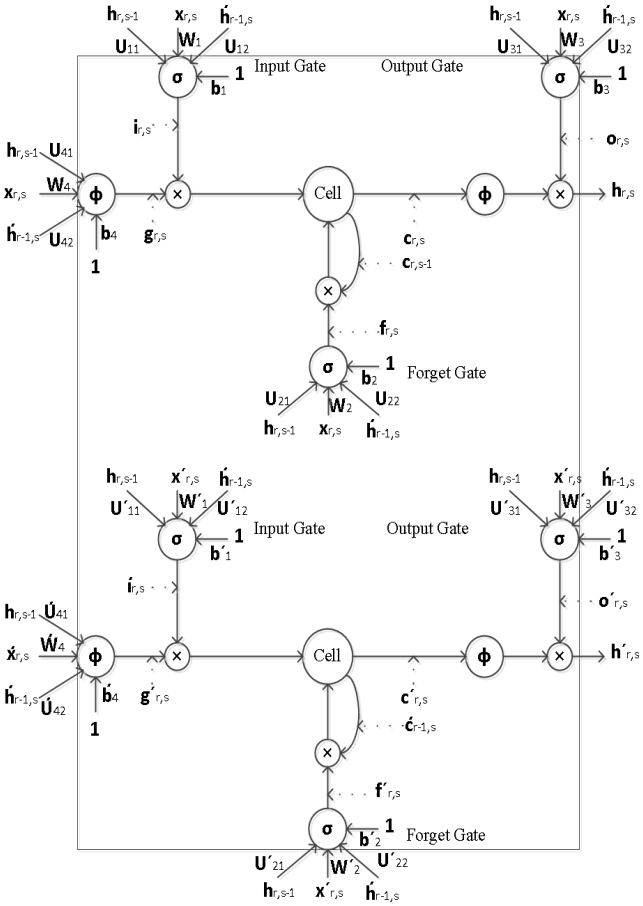

Figure 2. The architecture of a Grid LSTM cell.

gates enable the Grid LSTM to learn complicated distant spatial dynamics and attend to the important regions of the image by reading, writing and erasing the information from the memory cells.

Formally, let $\sigma : \mathbb{R} \mapsto (0, 1)$, $\sigma(x) = (1 + \exp(-x))^{-1}$ and $\phi : \mathbb{R} \mapsto (-1, 1)$, $\phi(x) = 2\sigma(2x) - 1$ be the *sigmoid* and *hyperbolic tangent* nonlinearity, respectively. At each step, the Grid LSTM outputs two new hidden vectors $\mathbf{h}_{r,s}$, $\mathbf{h}'_{r,s}$, and two new memory vectors $\mathbf{c}_{r,s}$, $\mathbf{c}'_{r,s}$ as follows

$$\mathbf{i}_{r,s} = \sigma\big(\mathbf{W}_1\mathbf{x}_{r,s} + \mathbf{U}_{11}\mathbf{h}_{r,s-1} + \mathbf{U}_{12}\mathbf{h}'_{r-1,s} + \mathbf{b}_1\big) \quad (1)$$

$$\mathbf{f}_{r,s} = \sigma\big(\mathbf{W}_2\mathbf{x}_{r,s} + \mathbf{U}_{21}\mathbf{h}_{r,s-1} + \mathbf{U}_{22}\mathbf{h}'_{r-1,s} + \mathbf{b}_2\big) \quad (2)$$

$$\mathbf{g}_{r,s} = \phi\big(\mathbf{W}_4\mathbf{x}_{r,s} + \mathbf{U}_{41}\mathbf{h}_{r,s-1} + \mathbf{U}_{42}\mathbf{h}'_{r-1,s} + \mathbf{b}_4\big) \quad (3)$$

$$\mathbf{c}_{r,s} = \mathbf{f}_{r,s} \odot \mathbf{c}_{r,s-1} + \mathbf{i}_{r,s} \odot \mathbf{g}_{r,s} \quad (4)$$

$$\mathbf{i}'_{r,s} = \sigma\big(\mathbf{W}'_1\mathbf{x}'_{r,s} + \mathbf{U}'_{11}\mathbf{h}_{r,s-1} + \mathbf{U}'_{12}\mathbf{h}'_{r-1,s} + \mathbf{b}'_1\big) \quad (5)$$

$$\mathbf{f}'_{r,s} = \sigma\big(\mathbf{W}'_2\mathbf{x}'_{r,s} + \mathbf{U}'_{21}\mathbf{h}_{r,s-1} + \mathbf{U}'_{22}\mathbf{h}'_{r-1,s} + \mathbf{b}'_2\big) \quad (6)$$

$$\mathbf{g}'_{r,s} = \phi\big(\mathbf{W}'_4\mathbf{x}'_{r,s} + \mathbf{U}'_{41}\mathbf{h}_{r,s-1} + \mathbf{U}'_{42}\mathbf{h}'_{r-1,s} + \mathbf{b}'_4\big) \quad (7)$$

$$\mathbf{c}'_{r,s} = \mathbf{f}'_{r,s} \odot \mathbf{c}'_{r-1,s} + \mathbf{i}'_{r,s} \odot \mathbf{g}'_{r,s} \quad (8)$$

$$\mathbf{o}_{r,s} = \sigma\big(\mathbf{W}_3\mathbf{x}_{r,s} + \mathbf{U}_{31}\mathbf{h}_{r,s-1} + \mathbf{U}_{32}\mathbf{h}'_{r-1,s} + \mathbf{b}_3\big) \quad (9)$$

$$\mathbf{h}_{r,s} = \mathbf{o}_{r,s} \odot \phi(\mathbf{c}_{r,s}) \quad (10)$$

$$\mathbf{o}'_{r,s} = \sigma\big(\mathbf{W}'_3\mathbf{x}'_{r,s} + \mathbf{U}'_{31}\mathbf{h}_{r,s-1} + \mathbf{U}'_{32}\mathbf{h}'_{r-1,s} + \mathbf{b}'_3\big) \quad (11)$$

$$\mathbf{h}'_{r,s} = \mathbf{o}'_{r,s} \odot \phi(\mathbf{c}'_{r,s}) \quad (12)$$

where, $\mathbf{h}_{1,0} = \mathbf{h}'_{0,1} = \mathbf{c}_{1,0} = \mathbf{c}'_{0,1} = \mathbf{0}$, $\odot$ denotes element-wise multiplication, $\mathbf{W}_i$,$\mathbf{W}'_i$,$\mathbf{U}_{ij}$, $\mathbf{U}'_{ij}$, $\mathbf{b}_i$, and $\mathbf{b}'_i$ ($i = 1, 2, 3, 4$, $j = 1, 2, 3$) are trainable parameters. The computation begins at the upper-left region of the input image, continues along horizontal and vertical spatial directions and ends at the bottom-right region.

One restriction of a Grid LSTM is that it can only benefit from the spatial context on upper left side of the current region. But, the spatial context in all directions around an image area is crucial to represent the visual meaning of that area. To resolve this problem, we introduce a Bidirectional Grid LSTM (BiGrid LSTM) by processing the input image in four directions (from top-left to bottom-right and vice versa, and from top-right to bottom-left and vice versa) with four separate Grid LSTMs. The equations for other Grid LSTMs are similar. For example, the equations for the Grid LSTM which its computation starts from bottom-right to top-left are as before, except that they are applied to the successor hidden states $\mathbf{h}_{r,s+1}$ and $\mathbf{h}'_{r+1,s}$ instead of the predecessor hidden states $\mathbf{h}_{r,s-1}$ and $\mathbf{h}'_{r-1,s}$. The initial hidden states for the bottom-right to top-left traversal are $\mathbf{h}_{R,S+1} = \mathbf{h}'_{R+1,S} = \mathbf{0}$. Since the visual meaning of a region in all Grid LSTMs must be the same, we share the weight matrices $\mathbf{W}_i$ and $\mathbf{W}'_i$, across all Grid LSTMs. This technique will reduce the number of the trainable parameters properly. We abbreviate the computation of the Grid LSTM which starts at top-left corner as

$$(\overset{\ulcorner}{\mathbf{c}}_{r,s}, \overset{\ulcorner}{\mathbf{c}}'_{r,s}, \overset{\ulcorner}{\mathbf{h}}_{r,s}, \overset{\ulcorner}{\mathbf{h}}'_{r,s}) = \texttt{GrLSTM}\,(\mathbf{x}_{r,s}, \mathbf{x}'_{r,s}, \overset{\ulcorner}{\mathbf{h}}_{r,s-1}, \overset{\ulcorner}{\mathbf{h}}'_{r-1,s}, \overset{\ulcorner}{\mathbf{c}}_{r,s-1}, \overset{\ulcorner}{\mathbf{c}}'_{r-1,s}).$$

With similar notations for other Grid LSTMs, the BiGrid LSTM computes two new hidden states and two new memory states as

$$\overset{\square}{\mathbf{h}}_{r,s} = \overset{\ulcorner}{\mathbf{h}}_{r,s} + \overset{\lrcorner}{\mathbf{h}}_{r,s} + \overset{\urcorner}{\mathbf{h}}_{r,s} + \overset{\llcorner}{\mathbf{h}}_{r,s} \quad (13)$$

$$\overset{\square}{\mathbf{h}}'_{r,s} = \overset{\ulcorner}{\mathbf{h}}'_{r,s} + \overset{\lrcorner}{\mathbf{h}}'_{r,s} + \overset{\urcorner}{\mathbf{h}}'_{r,s} + \overset{\llcorner}{\mathbf{h}}'_{r,s} \quad (14)$$

$$\overset{\square}{\mathbf{c}}_{r,s} = \overset{\ulcorner}{\mathbf{c}}_{r,s} + \overset{\lrcorner}{\mathbf{c}}_{r,s} + \overset{\urcorner}{\mathbf{c}}_{r,s} + \overset{\llcorner}{\mathbf{c}}_{r,s} \quad (15)$$

$$\overset{\square}{\mathbf{c}}'_{r,s} = \overset{\ulcorner}{\mathbf{c}}'_{r,s} + \overset{\lrcorner}{\mathbf{c}}'_{r,s} + \overset{\urcorner}{\mathbf{c}}'_{r,s} + \overset{\llcorner}{\mathbf{c}}'_{r,s} \quad (16)$$

We use $\mathbf{v}_{r,s} = (\overset{\square}{\mathbf{h}}_{r,s} \,||\, \overset{\square}{\mathbf{c}}_{r,s} \,||\, \overset{\square}{\mathbf{h}}'_{r,s} \,||\, \overset{\square}{\mathbf{c}}'_{r,s})$ as a feature vector extracted from region (r, s) of the image, where $||$ denotes concatenation. Note that unlike the CNN features, $\mathbf{v}_{r,s}$ considers the global scene context around region (r, s).

3.3. Deep (Two-Layer) Bidirectional LSTM with a Dynamic Visual Spatial Attention

We propose a Deep Bidirectional LSTM with a novel spatial attention mechanism to predict the t-th word of the caption, after it has seen the image and all preceding words. Intuitively, the first layer provides some contextual information about the image such as scene context (e.g. a farm), presence of objects, and co-occurrence of the words which facilitates the image caption generation. Then, the second layer generates the caption which describes the image in more detail. The motivation for a Bidirectional LSTM is that it can exploit the previous context of the input signal. The context around a word in both directions plays an important role in natural language description of the scene, e.g. the word `watching` is more likely to follow `TV` than `studying`.

Word Embedding. We map the *one hot* representation of word i, denoted by $\mathbf{e}_i$, to a semantic space of dimensionality d via $\mathbf{u}^i = \mathbf{L}\mathbf{e}_i$, where $\mathbf{L}$ denotes a $d \times n$ word embedding matrix, and $\mathbf{u}^i$ is the semantic representation of word i. We randomly initialize $\mathbf{L}$ and fine-tune during training our model. We denote by $Y = (Y_1, ..., Y_T)$, the caption of training image I, where Y_t $(t = 1, ..., T)$, is the word at time step t, Y_1 is a special start word, and Y_T is a special end word which define the start and end of the caption respectively. We represent Y_t by a vector of size d denoted by $\mathbf{u}_t$ using the word embedding matrix $\mathbf{L}$.

Global Scene Context. We initialize the Bidirectional LSTM with contextual information from the image. The information is represented by a feature vector denoted by $\mathbf{z}$ of

size 1000, where $\mathbf{z}_i$ is the probability of word i (in a dictionary of 1000 most common words) occurring in the caption of image I. The feature vector is computed by a word detector code from https://github.com/s-gupta/visual-concepts. The words are detected by applying the VGGNet to image regions and integrating the information with a Multiple Instance Learning framework [6]. The word detector is trained only using captions, not word bounding-boxes. We feed $\mathbf{Az} + \mathbf{a}$ to the first layer of the Deep Bidirectional LSTM at $t = 0$, where $\mathbf{A}$ is a $d \times 1000$ matrix, and $\mathbf{a}$ is a trainable bias. This layer also makes our model more robust to background clutter, since the word detector detect the words based on small local regions, not the whole image. After informing the Bidirectional LSTM about the image context, we ignore the output at time $t = 0$. Then, the embedded words $(\mathbf{u}_1, ..., \mathbf{u}_{T-1})$ are fed into the first layer.

Dynamic Spatial Attention Mechanism. For the second layer, we use a dynamic representation of the relevant patch of the image at time t denoted by $\mathbf{v}_t$. To produce $\mathbf{v}_t$, we first compute a weight for each encoded visual feature $\mathbf{v}_{r,s}$ using a two-layer feed forward neural network as follows

$$\hat{\upsilon}^t_{r,s} = \phi\big(\mathbf{M}^{(1)}\phi(\mathbf{M}^{(2)}\hat{\mathbf{v}}^t_{r,s} + \mathbf{b}^{(1)}) + b^{(2)}\big) \tag{17}$$

$$\upsilon^t_{r,s} = \frac{\exp\left(\hat{\upsilon}^t_{r,s}\right)}{\sum_{r=1}^{R}\sum_{s=1}^{S}\exp\left(\hat{\upsilon}^t_{r,s}\right)} \tag{18}$$

where, $\mathbf{M}^{(1)}$, $\mathbf{M}^{(2)}$, $\mathbf{b}^{(1)}$, $b^{(2)}$ are trainable parameters, and $\hat{\mathbf{v}}^t_{r,s}$ is the concatenation of the visual feature at location (r, s) and hidden state of the second layer of the Deep Bidirectional LSTM at time $t-1$. That is, $\hat{\mathbf{v}}^t_{r,s} = \mathbf{v}_{r,s} \,||\, \mathbf{h}^{(2)}_{t-1}$. Intuitively, $\upsilon^t_{r,s}$ is a positive weight for the location (r, s) which can be interpreted as the relative importance to give to location (r, s) at time t. The weight $\upsilon^t_{r,s}$ depends on the visual features and the previous word that has already been generated.

After computing the attention weights, we need an attention mechanism to compute $\mathbf{v}_t$. A straightforward attention mechanism is soft attention mechanism which is used by recent attention-based caption generation models [26]. It computes $\mathbf{v}_t$ as a weighted summation of visual features $\mathbf{v}_{r,s}$, $(r = 1, ..., R)$ and $(s = 1, ..., S)$. That is, $\mathbf{v}_t = \sum_{r=1}^{R}\sum_{s=1}^{S}\upsilon^t_{r,s}\mathbf{v}_{r,s}$. However, a disadvantage of this method is that it does not consider the spatial context and order of the visual features. To resolve this issue, [25, 15] proposed an attention mechanism based on a Gated Recurrent Unit. By extending this idea, we introduce a new 2D spatial attention mechanism based on a Grid LSTM. In a Grid LSTM, the forget gates learn how much of the previous memory should be kept, and the input gates control how much of the input should be read. Therefore, we use another Grid LSTM, called *attention Grid LSTM*, whose input and forget gates are replaced with the attention weights that we computed.

More precisely, the equations for the attention Grid LSTM are as before, except that 4, and 8 are substituted by

$$\hat{\mathbf{c}}_{r,s} = (1 - \upsilon^k_{r,s}) \odot \hat{\mathbf{c}}_{r,s-1} + \upsilon^k_{r,s} \odot \hat{\mathbf{g}}_{r,s} \tag{19}$$

$$\hat{\mathbf{c}}'_{r,s} = (1 - \upsilon^k_{r,s}) \odot \hat{\mathbf{c}}'_{r-1,s} + \upsilon^k_{r,s} \odot \hat{\mathbf{g}}'_{r,s}, \tag{20}$$

where, $\hat{}$ denote the computation of the attention Grid LSTM. The inputs to the attention Grid LSTM at step (r, s) are $\mathbf{x}_{r,s} = (\hat{\square}_{r,s} \,||\, \hat{\square}_{r,s})$ and $\mathbf{x}'_{r,s} = (\hat{\square}'_{r,s} \,||\, \hat{\square}'_{r,s})$. The concatenation of the vertical and horizontal hidden and memory states of the attention Grid LSTM at the last spatial step is used as the relevant patch of the image at time t. That is, $\mathbf{v}_t = \Big(\hat{\mathbf{h}}^t_{R,S} \,||\, \hat{\mathbf{c}}^t_{R,S} \,||\, \hat{\mathbf{h}}'^t_{R,S} \,||\, \hat{\mathbf{c}}'^t_{R,S}\Big)$. Our experiments show utilizing the spatial attention mechanism rather than soft attention mechanism will improve the performance. With $\upsilon^0_{r,s} = 1/RS$, $\mathbf{u}^{(1)}_0 = \mathbf{Az} + \mathbf{a}$, $\mathbf{v}^{(1)}_t = \mathbf{0}$, $\mathbf{v}^{(2)}_t = \mathbf{v}_t$, $\mathbf{u}^{(1)}_t = \mathbf{u}_t$, $\mathbf{u}^{(2)}_t = \mathbf{h}^{(1)}_t$ $(t = 0, ..., T)$, the equations for cell update and the output of the left-to-right layer l $(l = 1, 2)$ of the Deep Bidirectional LSTM are as follows:

$$\overrightarrow{\mathbf{i}}^{(l)}_t = \sigma\big(\overrightarrow{\mathbf{W}}^{(l)}_1\,\mathbf{u}^{(l)}_t + \overrightarrow{\mathbf{U}}^{(l)}_1\,\mathbf{h}^{(l)}_{t-1} + \mathbf{V}_1\mathbf{v}^{(l)}_t + \overrightarrow{\mathbf{b}}^{(l)}_1\big) \tag{21}$$

$$\overrightarrow{\mathbf{f}}^{(l)}_t = \sigma\big(\overrightarrow{\mathbf{W}}^{(l)}_2\,\mathbf{u}^{(l)}_t + \overrightarrow{\mathbf{U}}^{(l)}_2\,\mathbf{h}^{(l)}_{t-1} + \mathbf{V}_2\mathbf{v}^{(l)}_t + \overrightarrow{\mathbf{b}}^{(l)}_2\big) \tag{22}$$

$$\overrightarrow{\mathbf{o}}^{(l)}_t = \sigma\big(\overrightarrow{\mathbf{W}}^{(l)}_3\,\mathbf{u}^{(l)}_t + \overrightarrow{\mathbf{U}}^{(l)}_3\,\mathbf{h}^{(l)}_{t-1} + \mathbf{V}_3\mathbf{v}^{(l)}_t + \overrightarrow{\mathbf{b}}^{(l)}_3\big) \tag{23}$$

$$\overrightarrow{\mathbf{g}}^{(l)}_t = \phi\big(\overrightarrow{\mathbf{W}}^{(l)}_4\,\mathbf{u}^{(l)}_t + \overrightarrow{\mathbf{U}}^{(l)}_4\,\mathbf{h}^{(l)}_{t-1} + \mathbf{V}_4\mathbf{v}^{(l)}_t + \overrightarrow{\mathbf{b}}^{(l)}_4\big) \tag{24}$$

$$\overrightarrow{\mathbf{c}}^{(l)}_t = \overrightarrow{\mathbf{f}}^{(l)}_t \odot \overrightarrow{\mathbf{c}}^{(l)}_{t-1} + \overrightarrow{\mathbf{i}}^{(l)}_t \odot \overrightarrow{\mathbf{g}}^{(l)}_t \tag{25}$$

$$\overrightarrow{\mathbf{h}}^{(l)}_t = \overrightarrow{\mathbf{o}}^{(l)}_t \odot \phi(\overrightarrow{\mathbf{c}}^{(l)}_t) \tag{26}$$

where, $\overrightarrow{\mathbf{h}}^{(l)}_{-1} = \mathbf{0}$, $\mathbf{V}_i$, $\overrightarrow{\mathbf{W}}^{(l)}_i$, $\overrightarrow{\mathbf{U}}^{(l)}_i$, and $\overrightarrow{\mathbf{b}}^{(l)}_i$ $(i = 1, 2, 3, 4)$ are trainable parameters. The equations for the right-to-left computation are similar except that $t-1$ is replaced by $t+1$ and $\overleftarrow{\mathbf{h}}^{(l)}_{T+1} = \mathbf{0}$. Intuitively, $\overrightarrow{\mathbf{U}}^{(l)}_i$ models grammar and left-to-right context, while $\overrightarrow{\mathbf{W}}^{(l)}_i$ encodes the words. Since the meaning of a word form left-to-right or right-to-left is the same, we set $\overrightarrow{\mathbf{W}}^{(l)}_i = \overleftarrow{\mathbf{W}}^{(l)}_i$. This technique also helps to prevent overfitting. The output of the l-th layer of the Deep Bidirectional LSTM is computed as $\mathbf{h}^{(l)}_t = \overrightarrow{\mathbf{h}}^{(l)}_t + \overleftarrow{\mathbf{h}}^{(l)}_t$. The final output of the Deep Bidirectional LSTM at time t is fed to a softmax layer to produce a probability distribution $\mathbf{p}_t$ over the dictionary words

$$\mathbf{p}_t(i) = \frac{\exp\left(\mathbf{m}_i^\top \mathbf{h}^{(2)}_t + b_i\right)}{\sum_{j=1}^{n}\exp\left(\mathbf{m}_j^\top \mathbf{h}^{(2)}_t + b_j\right)} \tag{27}$$

where, $\mathbf{p}_t(i)$ is the probability of Y_{t+1} being i-th word in the dictionary given $\mathbf{h}_t^{(2)}$, b_i is i-th entry of a trainable bias $\mathbf{b}$, and $\mathbf{m}_i$ specifies i-th row of a trainable $n \times d$ matrix $\mathbf{M}$. Intuitively, this matrix decodes the dense word representation into a pseudo one-hot word representation which is the inverse function of the word embedding matrix. Thus, matrix $\mathbf{M}$ is shared with the transpose of the word embedding matrix $\mathbf{L}$. This technique will effectively reduce the number of parameters of the model [19].

To generate a caption for a new image at the test time, ideally we need to find a caption $\hat{Y}$ such that

$$\hat{Y} = \arg\max_{Y} \sum_{t=1}^{T-1} \log\left(\mathbf{p}_{t+1}(Y_{t+1})\right). \tag{28}$$

However, since the exhaustive search is intractable, we use *beam search* with size $k = 20$ to find $\hat{Y}$. The beam search algorithm iteratively considers the k best captions up to time t as candidates to generate new captions of size $t + 1$.

3.4. Integrating with Region-Grounded Texts

In this section, we propose a transfer learning technique which incorporates region-grounded texts of the input image, e.g. `a red stop sign`, `a cloudy sky`, `boy on horse`, to boost the performance. For this purpose, we use a dense captioning model from https://github.com/jcjohnson/densecap to extract a set of descriptions for the input image [10]. This model has been trained on Visual Genome region caption dataset [14]. This enables our model to transfer *learning* from a region-grounded caption dataset and produce more precise captions, since the grounded textual information often describes the properties of the objects and their relationships in an image which may not be represented properly by visual features on small datasets. Each region-grounded text has a bounding-box and a confidence score. Our goal is to summarize local region-grounded texts into a single global caption by attending to the important bounding-boxes.

For each image, we select all region-grounded texts with a confidence score greater than 1.0. The words are encoded by the same embedding matrix $\mathbf{L}$. We applied a simple Bag of Words to encode a region-grounded text. LSTMs may be also applied, but they need more computations and parameters. Each region (r, s) is then represented by a textual feature vector of size d by taking the average of the encoded region-grounded texts whose bounding-box covers region (r, s). Then, we obtain vertical and horizontal textual features $\bar{\mathbf{x}}_{r,s}$ and $\bar{\mathbf{x}}'_{r,s}$ by projecting the resulting feature vector to $d/2$ dimensions. The inputs to the attention Grid LSTM are now computed as $\mathbf{x}_{r,s} = (\square_{r,s} \,||\, \square_{r,s}) \,||\, \bar{\mathbf{x}}_{r,s}$ and $\mathbf{x}'_{r,s} = (\square'_{r,s} \,||\, \square'_{r,s}) \,||\, \bar{\mathbf{x}}'_{r,s}$, and $\hat{\mathbf{v}}^t_{r,s}$ is computed as $\hat{\mathbf{v}}^t_{r,s} = (\mathbf{v}_{r,s} \,||\, \mathbf{h}^{(2)}_{t-1}) \,||\, (\bar{\mathbf{x}}_{r,s} \,||\, \bar{\mathbf{x}}'_{r,s})$.

3.5. Training Details

The sum of the negative log likelihood of the correct word at each time step is chosen as the loss, that is $-\sum_{t=1}^{T-1} \log\left(\mathbf{p}_{t+1}(\texttt{idx}(Y_{t+1}))\right)$, where $\texttt{idx}(Y_{t+1})$ is the index of word Y_{t+1} in the dictionary. This loss is minimized using RMSprop with minibatches of size 50 and learning rate 0.0001. To prevent overfitting, dropout with probability 0.6 and early-stopping are used. During training, all parameters are tuned except for the weights of the word detector and CNN components which we keep fixed to prevent overfitting. In VGGNet experiments, we use $m' = 128$ hidden states for the BiGrid LSTM and 256 hidden states for the Deep Bidirectional LSTM, respectively. In ResNet experiments, we use $m' = 256$ hidden states for the BiGrid LSTM and 512 hidden states for the Deep Bidirectional LSTM, respectively. The word embedding size is set to $d = 256$ for VGGNet, and $d = 512$ for ResNet.

Since at the test time a caption must be generated word by word from left to right, the backward LSTM needs to see the reverse of the sub-captions during training. To resolve this problem we train our model with all sub-captions of a training sample, that is $Y = (Y_1, ..., Y_{t'})$ $(t' = 1, ...T)$. We reset the Bidirectional LSTM after feeding each sub-caption. This increases the number of training samples and the training time by the average of the caption lengths in the training data, which is 10 for MS-COCO dataset. However, we find fewer epochs are required for convergence. Our model took around four days to train on two NVIDIA Titan X GPUs.

4. Experiments

In this section, we firstly introduce the dataset and evaluation metrics that we use in our experiments. Then, we explain our experimental set up and methodology. Finally, the experimental results are presented and discussed.

4.1. Data, Metrics and Experimental Setup

Our results are reported on the MS-COCO dataset. MS-COCO dataset contains 82,783 training images, 40,504 validation images, and 40,775 test images. The dataset is annotated with 5 sentences using Amazon Mechanical Turk. The captions for the test set are not publicly available. We follow [12] to preprocess the captions with basic tokenization by converting all sentences to lower case, throwing away non-alphanumeric characters, and filtering the words to those that occur at least 5 times in the training set. This results in a dictionary of size 8,791. We use METEOR [4] and BLEU [20] as evaluation metrics, which are popular in the machine translation literature and used in recent image caption generation papers. The BLEU score is based on n-gram precision of the generated caption with respect to the references. The METEOR is based on the har-

Model	B-1	B-2	B-3	B-4	MET
BiRNN† [12]	62.5	45.0	32.1	23.0	19.50
mRNN† [18]	67.0	49.0	35.0	25.0	-
LRCN† [5]	62.8	44.2	30.4	21.0	-
Google NIC† [23]	66.6	46.1	32.9	24.6	-
Log Bilinear† [13]	70.8	48.9	34.4	24.3	20.03
Soft-Attention† [26]	70.7	49.2	34.4	24.3	23.90
Hard-Attention† [26]	71.8	50.4	35.7	25.0	23.04
ATT-FCN† [29]	70.9	53.7	40.2	30.4	24.30
Review Networks [27]	-	-	-	29.0	23.20
SCA-CNN° [1]	71.9	54.8	41.1	31.1	25.00
ACVT [24]	73.0	56.0	41.0	31.0	25.00
Boosting with Attributes° [28]	73.0	56.5	42.9	32.5	25.10
Adaptive Attention° [17]	74.2	58.0	43.9	33.2	26.60
Visual Concepts [6]+LSTM	65.1	48.3	35.0	24.6	21.00
CNN+LSTM	67.0	50.0	36.2	26.0	21.90
CNN+LSTM2	68.7	52.1	38.2	28.0	22.90
Gr+LSTM2+soft	73.2	56.1	41.1	31.3	25.14
Gr+LSTM2+sp	73.6	56.4	41.5	31.7	25.25
BiGr+LSTM2+sp	74.2	56.8	41.9	32.0	25.48
BiGr+BiLSTM2+sp	74.7	57.5	42.1	32.3	25.71
BiGr+BiLSTM2+sp°	74.9	58.9	44.0	33.9	26.25
BiGr+BiLSTM2+sp+rg°	**76.2**	**60.1**	**45.1**	**35.0**	**27.02**

Table 1. BLEU-1,2,3,4 and METEOR scores on MS-COCO test set. We only compare with the results that have been officially published. For fairness, we only compare with the results which employ comparable CNNs such as GoogLeNet, VGGNet and ResNet. The - denotes that the result is not reported. Models that use ResNet are denoted by °. The † denotes the results are reported on validation set, since the results on the test set are not available.

monic mean of unigram precision and recall, and produces a good correlation with human judgment.

To analyze the effect of various improvements, we implemented a few lesion models. In Visual Concepts+LSTM model, we feed word probabilities $\mathbf{z}$ to a single layer LSTM. The CNN+LSTM uses a feature vector of size $4,096$ at the very top layer of VGGNet. This vector is mapped to a space of dimensionality 256. Then, the resulting vector is fed to a single layer LSTM at time $t = 0$. The CNN+LSTM2 stacks two single layers of unidirectional LSTM in the first and second lesion models.

These lesion models do not apply the Grid LSTM. Thus, they cannot exploit spatial information efficiently. The Gr+LSTM2+soft uses a Grid LSTM and a deep LSTM for caption generation with a soft attention mechanism. The Gr+LSTM2+sp is similar to Gr+LSTM2+soft but it utilizes our dynamic spatial attention mechanism. The BiGr+LSTM2+sp employs a BiGrid LSTM and a deep LSTM for caption generation with a dynamic spatial attention mechanism. The BiGr+BiLSTM2+sp uses a BiGrid LSTM and a Deep Bidirectionl LSTM for caption generation with dynamic spatial attention mechanism. We also implemented this last model with a more powerful CNN, ResNet [8], instead of VGGNet. Finally, BiGr+BiLSTM2+sp+rg is the same as BiGr+BiLSTM2+sp with ResNet but it uses region-grounded texts.

4.2. Results and Discussion

Our results are reported in Table 1. The full model outperforms the baseline and previous works in all cases. The Visual Concepts+LSTM model does not use CNN features of the image. It only uses the word detector to generate a caption. Conversely, CNN+LSTM model only uses the CNN features. CNN+LSTM2 outperforms both of these models by exploiting both CNN features and word detector. Models which use a Grid LSTM outperform others that use global image features, which is due to the power of Grid LSTM to represent spatial information of an image. Gr+LSTM2+sp outperforms Gr+LSTM2+soft which shows dynamic spatial attention is more efficient than soft attention. BiGr+LSTM2+sp outperforms Gr+LSTM2+sp since it can scan an image from all directions (from top-left to bottom-right and vice versa, and from top-right to bottom-left and vice versa). Also, BiGr+BiLSTM2+sp outperforms BiGr+LSTM2+sp since it can consider the surrounding context of a word from both sides. Finally, BiGr+BiLSTM2+sp+rg with ResNet outperforms BiGr+BiLSTM2+sp with ResNet which shows incorporating region-grounded texts boost the performance. Since our model has more parameters than the other state-of-the-art models, it is extremely data driven. As a result, we believe that even with these promising results, the capability of our model will grow in the future, as the image captioning datasets extend.

Figure 3 shows example captions generated by our model. We also visualize the attention weights in Figure 4. The 7×7 heatmaps represents the value of the visual attention weights $\upsilon^t_{r,s}$ for each generated word. The examples shows our visual spatial attention model can attend to the right concept, even in the presence of background clutter, especially for the words which have a well-defined bounding box such as *bear*. Moreover, our model generates captions that are grammatically correct. This shows the power of our Deep Bidirectional LSTM to model the context from both sides, and thereby generate grammatically correct sentences. Finally, Figure 5 shows two example of mistakes of our model which are probably due to small size data.

5. Conclusions

We have presented a novel attention-based contextual deep architecture for image caption generation. Experimental results on MS-COCO dataset show the robustness of our model in terms of quantitative evaluations and qualitative results. The visualization results show that our model can attend to the right concept during the image caption generation. We believe that, by leveraging the power of BiGrid LSTM, our architecture can generate attention maps which are more compatible with the human attention maps than other state-of-the-art models.

Figure 3. Example captions generated by our model.

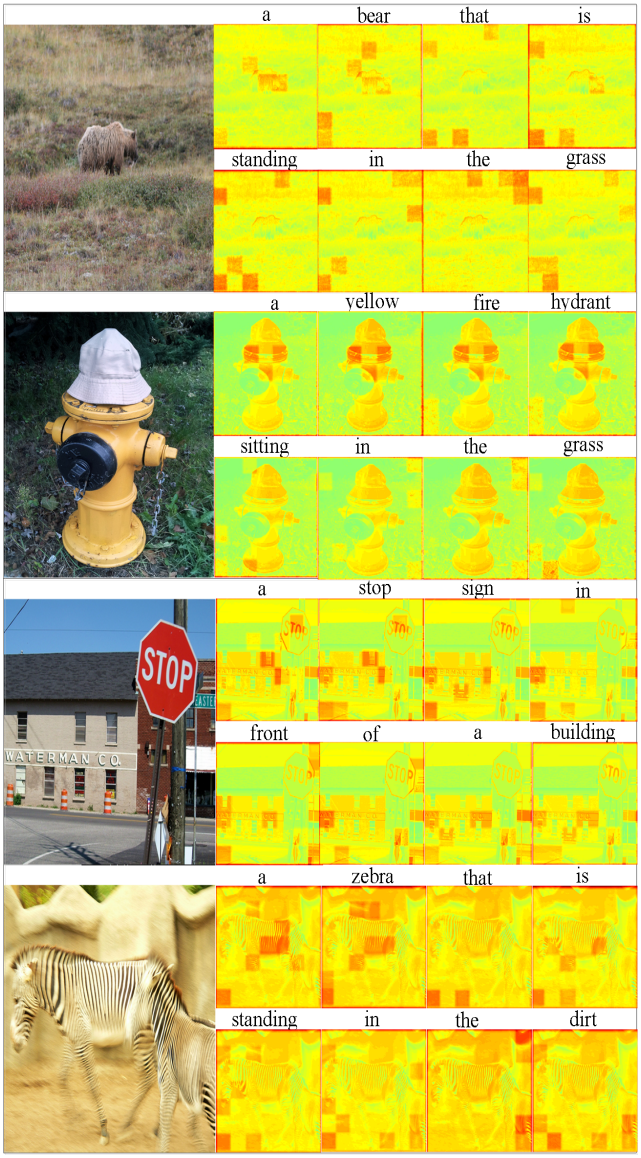

Figure 4. Visualization of the attention. The 7×7 heatmaps represents value of weight $\upsilon_{r,s}^{t}$ for t-th word in the generated caption. Each value corresponds to a specific region in the image.

Figure 5. Two example of mistakes of our model.

References

[1] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 6298–6306. IEEE, 2017. 2, 7

[2] X. Chen and C. Lawrence Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2422–2431, 2015. 2

[3] X. Chen and C. L. Zitnick. Learning a recurrent visual representation for image caption generation. *arXiv preprint arXiv:1411.5654*, 2014. 1, 2

[4] M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *In Proceedings of the Ninth Workshop on Statistical Machine Translation*. Citeseer, 2014. 6

[5] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015. 1, 2, 7

[6] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1473–1482, 2015. 1, 2, 5, 7

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 2, 7

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997. 2

[10] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. *arXiv preprint arXiv:1511.07571*, 2015. 6

[11] N. Kalchbrenner, I. Danihelka, and A. Graves. Grid long short-term memory. *arXiv preprint arXiv:1507.01526*, 2015. 2

[12] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015. 2, 6, 7

[13] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In T. Jebara and E. P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 595–603. JMLR Workshop and Conference Proceedings, 2014. 7

[14] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016. 6

[15] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387, 2016. 5

[16] R. Lebret, P. O. Pinheiro, and R. Collobert. Phrase-based image captioning. *arXiv preprint arXiv:1502.03671*, 2015. 1, 2

[17] J. Lu, C. Xiong, D. Parikh, and R. Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 6, 2017. 2, 7

[18] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*, 2014. 1, 2, 7

[19] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. L. Yuille. Learning like a child: Fast novel visual concept learning from sentence descriptions of images. *CoRR*, abs/1504.06692, 2015. 6

[20] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002. 6

[21] M. I. Posner. Orienting of attention. *Quarterly journal of experimental psychology*, 32(1):3–25, 1980. 1

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[23] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555, 2014. 1, 2, 7

[24] Q. Wu, C. Shen, L. Liu, A. Dick, and A. van den Hengel. What value do explicit high level concepts have in vision to language problems? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 203–212, 2016. 1, 7

[25] C. Xiong, S. Merity, and R. Socher. Dynamic memory networks for visual and textual question answering. *arXiv preprint arXiv:1603.01417*, 2016. 5

[26] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015. 2, 5, 7

[27] Z. Yang, Y. Yuan, Y. Wu, W. W. Cohen, and R. R. Salakhutdinov. Review networks for caption generation. In *Advances in Neural Information Processing Systems*, pages 2361–2369, 2016. 1, 7

[28] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei. Boosting image captioning with attributes. *arXiv preprint arXiv:1611.01646*, 2016. 1, 7

[29] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. *CoRR*, abs/1603.03925, 2016. 1, 2, 7