Show, Discriminate, and Tell: A Discriminatory Image Captioning Model with Deep Neural Networks

Zelun Luo
Department of Computer Science
Stanford University
zelunluo@stanford.edu

Boya Peng
Department of Computer Science
Stanford University
boya@stanford.edu

Te-Lin Wu
Department of Electrical Engineering
Stanford University
telin@stanford.edu

Abstract

Caption generation has long been seen as a difficult problem in Computer Vision and Natural Language Processing. In this paper, we present an image captioning model based on a end-to-end neural framework that combines Convolutional Neural Network and Recurrent Neural Network. Critical to our approach is a ranking objective that attempts to add discriminatory power to the model. Experiments on MS COCO dataset shows that our model consistently outperforms its counterpart with no ranking objective, both quantitatively based on BLEU and CIDEr scores and qualitatively based on human evaluation.

1. Introduction

Several methods have been proposed for the task of image caption generation. Most of these methods are based on Recurrent Neural Networks, inspired by the successful use of sequence-to-sequence training with deep recurrent networks in machine translation [1, 2, 14].

The first deep learning method for image captioning was proposed by Kiros et al. [8]. The method utilizes a multimodal log-bilinear model that is biased by the features from the image. Kiros et al. [9] extends this work by proposing a method that allows both ranking and caption generation. Mao et al. [11] replaces the feed-forward neural network with a recurrent neural network. Vinyals et al. [15] used a LSTM (Long short-term memory) network, which is a refined version of a vanilla recurrent neural network. Unlike Mao et al.'s and Kiros et al.'s models, which feed in image features at every time step, in Vinyals et al.'s model the image is fed into the LSTM only at the first time step.

Unlike the above works that represent image as a single feature vector, Karpathy et al. [5] learn detectors for several visual concepts and train a model that generates natural language descriptions of images and their regions. Xu et al. [16] propose approaches to caption generation that attempt to incorporate a form of attention with either ”hard” or ”soft” attention mechanism.

Contributions Aiming to generate more discriminatory captions, we propose a novel ranking objective (elaborated in 3.2) on top of the end-to-end neural framework for image caption generation, which enforces alignments between images and generated captions.

2. Related Work

2.1. Previous Work

Several methods have been proposed for the task of image caption generation. Most of these methods are based on Recurrent Neural Networks, inspired by the successful use of sequence-to-sequence learning with deep recurrent neural networks in machine translation [1, 2, 14].

The first deep learning method for image captioning was proposed by Kiros et al. [8]. The method utilizes a multimodal log-bilinear model that is biased by the features from the image. Kiros et al. [9] extended this work by proposing a method that allows both ranking and caption generation. Mao et al. [11] replaces the feed-forward neural network with a recurrent neural network. Vinyals et al. [15] used a LSTM (Long short-term memory) network, which is a refined version of a vanilla recurrent neural network. Unlike Mao et al.'s and Kiros et al.'s models, which image features are fed in at every time step, in Vinyals et al.'s model, the image is fed into the LSTM once only at the first time step.

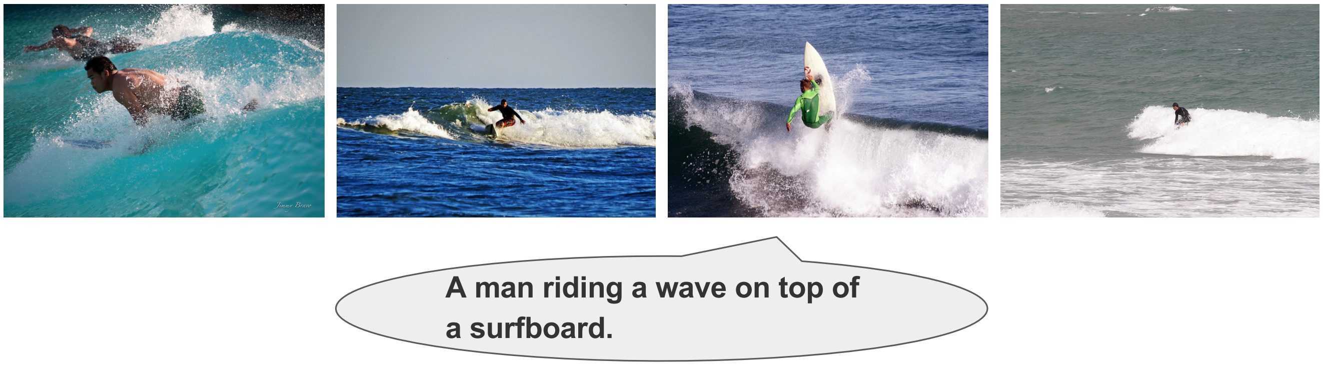


Figure 1: Examples of repetitive captions for different images.

2.2. Contributions

Aiming to generate more discriminatory and non overly general captions, we propose a novel ranking objective (elaborated in 3.2) on top of the sentence generator in an end-to-end fashion of neural framework for image caption generation, which enforces alignments between images and generated captions.

3. Technical Approach

3.1. Overview

In this project, we propose a novel ranking objective on top of the end-to-end neural framework for image caption generation. We leverage an encoder-decoder approach: The Convolutional Neural Network encoder transforms the images into some fixed-length image feature vectors, which is then fed into the Recurrent Neural Network decoder to generate the image captions. Aiming to generate more discriminatory captions, we introduce a ranking objective that enforces the alignments between images and generated captions and penalizes misaligned pairs. The overall architecture of our model is shown in Figure 2.

3.2. Model Architecture

Image Model We use a Convolutional Neural Network (CNN) to extract image features. The 16-layer VGGNet[13] pre-trained on ImageNet [3] is used as our image feature extractor. It was the state-of-the-art model in ImageNet Challenge 2014, featuring relatively small (3×3) convolutional filters and simple configurations. We changed the last 4096-dimensional fully connected layer into K-dimensional and then extract features from the last layer, where K is the size of word embeddings that are used as inputs to our language model. Each image is thus representing a K-dimensional feature vector $I_i \in R^K$.

Language Model We use a Long Short-Term Memory (LSTM) network [4] as the building block of our language model. As a particular form of Recurrent Neural Networks, LSTM is able to deal with vanishing and exploding gradients, which is the most common drawbacks for vanilla RNNs.

The core of the LSTM is a memory cell c that encodes knowledge at every time step of what inputs have been observed up to this step. The behavior of the cell is controlled by ”gates” – layers which are applied multiplicatively and thus can either keep a value from the gated layer if the gate is 1 or zeros this value if the gate is 0. More specifically, three gates are being used that control whether to forget the current cell value (forget gate f), if it should read its input (input gate i) and whether to output the new cell value (output gate o). The definition of the gates and cell update and output are as follows:

$$
\begin{aligned}
i^{(t)} &= \sigma(W^{(i)}x^{(t)} + U^{(i)}h^{(t-1)}) \\
f^{(t)} &= \sigma(W^{(f)}x^{(t)} + U^{(f)}h^{(t-1)}) \\
o^{(t)} &= \sigma(W^{(o)}x^{(t)} + U^{(o)}h^{(t-1)}) \\
\widetilde{c}^{(t)} &= \tanh(W^{(c)}x^{(t)} + U^{(c)}h^{(t-1)}) \\
c^{(t)} &= f^{(t)} \circ \widetilde{c}^{(t-1)} + i^{(t)} \circ \widetilde{c}^{(t)} \\
h^{(t)} &= o^{(t)} \circ \tanh(c^{(t)})
\end{aligned}
$$

where $\circ$ represents the product with a gate value, and $h^{(t)}$ is the output hidden state at time step t.

The LSTM takes the image feature vector I_i as its first hidden state and a sequence of input vectors $(x_1, ..., x_D)$. It outputs a sequence of log probabilities at each time step:

$$y = \{\vec{y_1}, \vec{y_2}, ..., \vec{y_D}\}, \vec{y_i} \in R^M$$

where M is the size of the vocabulary and D is the length of the sentence.

Ranking Objective During training, at each forward pass, our model takes a mini-batch of N image-sentence pairs. We use the dot product $I_i^T s_j$ to measure the similarity between the i-th image and the j-th sentence. Intuitively, $I_i^T s_i$

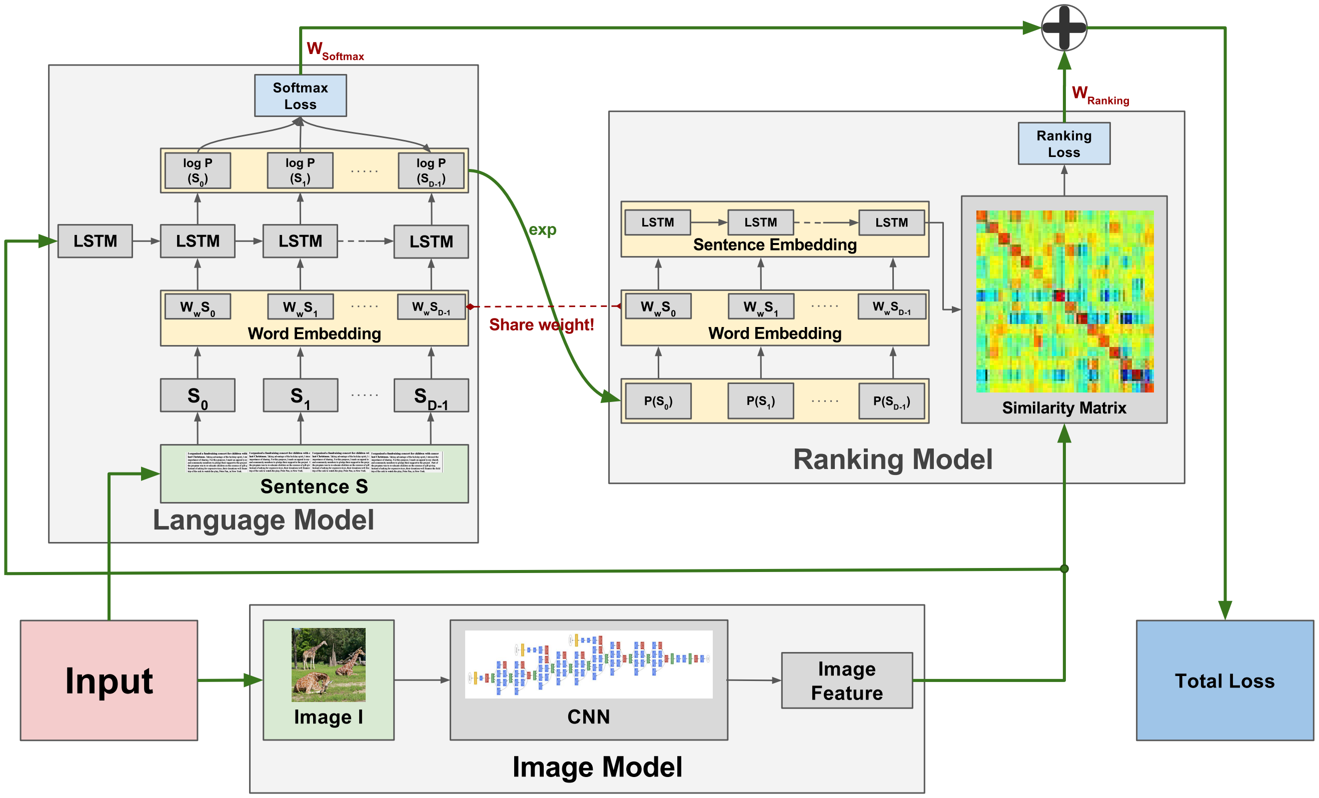


Figure 2: Diagram of our discriminatory image captioning model. It consists of three modules: image model, language model, and ranking model.

should be larger than any $I_i^T s_j (i \neq j)$ by a margin, as we want to ensure that the generated sentence 'uniquely' corresponds to the image, and thus add discriminatory power to our model. The ranking model takes a batch of image features $I \in R^{N \times K}$ and corresponding log probabilities $Y = \{\vec{Y_1}, \vec{Y_2}, ..., \vec{Y_D}\}, \vec{Y_i} \in R^{N \times M}$. We first transform log probabilities into probabilities, as probabilities naturally express distribution over outputs:

$$P = \exp(Y) \in R^{D \times N \times M}$$

We then use the probabilities as "soft indices" to index into the same word embedding table as in the language model to find each word embedding, and use another LSTM to learn corresponding sentence embeddings:

$$S = \{\vec{s_1}, ..., \vec{s_N}\}, \vec{s_i} \in R^{N \times K}$$

where the LSTM takes each word embedding at each time step, and the sentence embedding is represented as the output from the LSTM at the last time step (encoded all the temporal information). With a batch of image features and corresponding sentence embeddings, we compute the similarity matrix as follows:

$$Sim(I, S) = S \cdot I^T \in R^{N \times N}$$

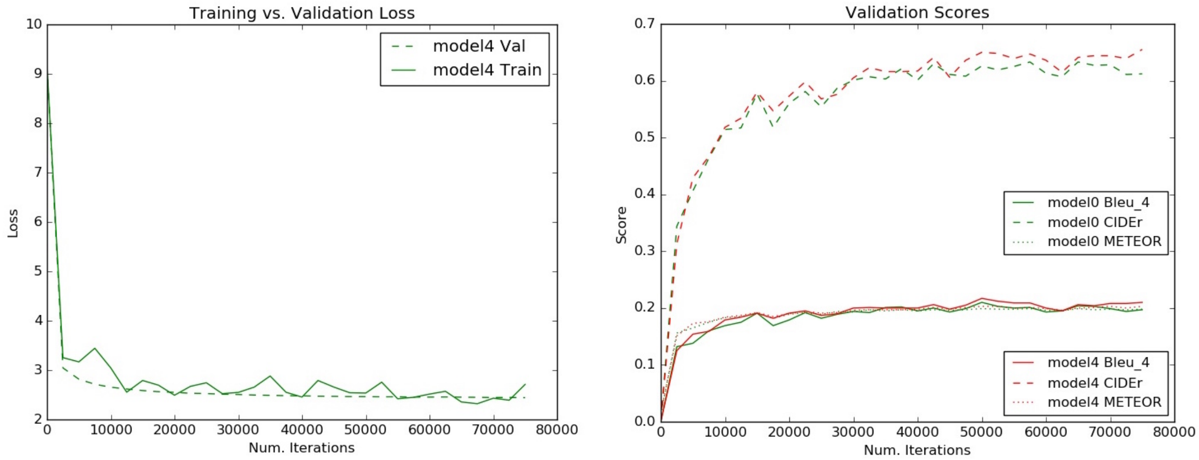
We then define the ranking objective over one mini-batch as the sum of max-margin loss over both columns and rows:

$$\begin{aligned} J(Sim(I,S)) = & \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\max(0, Sim[i,j] - Sim[i,i] + 1) + \\ & \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{N}\max(0, Sim[i,j] - Sim[i,i] + 1) \end{aligned}$$

This objective encourages aligned image-sentence pairs to have a higher score than misaligned pairs, by a margin.

Training Our language model is trained to combine a word embedding (x_t) and the previous hidden state (h_{t-1}) to predict the next word (y_t). We set h_0 to be the image feature vector and x_1 to a special START token. On the last step when x_D represents the last word, the target label is set to a special END token. The cost function is to minimize the negative log probability assigned to the target labels (Softmax classifier):

$$L(I, Y) = -\frac{1}{N}\sum_{i=1}^{N} Y_i$$



(a) Train vs. Validation Loss

(b) Validation Scores

Figure 3: Quantitative Results (model0 stands for baseline model, model4 stands for our model)

The total loss during training is defined as the weighted sum of the ranking objective and Softmax loss:

$$Loss = w_J J(Sim(I, S)) + w_L L(I, Y)$$

Test time At test time, we extract the image representation I, set h_0 to I, x_1 to the START token and compute the distribution over the first word y_1. We pick the argmax from the distribution, find its word embedding as x_2, and repeat this process until the END token is generated.

4. Experiments

4.1. Dataset

We train and test our model on the Microsoft COCO (Common Objects in Context) [10] dataset, a large image dataset designed for object detection, segmentation, and caption generation. There are 5 written caption descriptions for each image in MS COCO. For our task, we use the 2014 release of the dataset, which contains 82,783 training, 40,504 validation, and 40,775 testing images. All the words that occur less than 5 times are mapped to a special <UNK> token.

4.2. Evaluation Metric

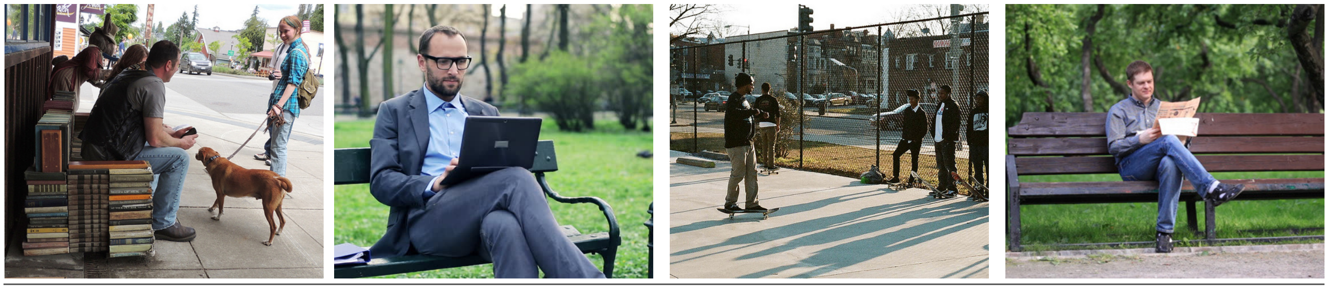
The most reliable metric for image captioning is based on human evaluations, which can take months to finish and involve human labor that cannot be reused. Moreover, choosing a human-like evaluation matric is a challenging problem for image captioning. In this work, we perform extensive experiments on our model with several metrics to evaluate the effectiveness of our model. The BLEU Score [12] is one of the most common metrics in image description tasks, which is a form of precision of word n-grams between generated and reference sentences. We report BLEU-4 as it is the standard in machine translation (note that BLEU-n is a geometric average of precision over 1- to n-grams). Besides BLEU, we also use METEOR and Cider, which are two popular metrics that are deemed to be appropriate for evaluating caption. [14].

4.3. Baseline Model

We use the model from NeuralTalk2 [6] as our baseline model. NeuralTalk2 is a Torch implementation of the "Show and Tell" model [15] which shares the same image model and language model as ours but does not apply the ranking objective. The pretrained 16-layer VGGNet [13] is used as the image model, with a learning rate of 1×10^{-5}. For the language model, both word embeddings and LSTM hidden states have 512 dimensions. The initial learning rate for the LSTM is 4×10^{-4}, which decays every 50000 iterations. We clip gradients when they exceed 0.1, and use a dropout of 0.5. For both the image model and the language model, the batch size is set to 16. We use the Adam optimizer [7] with $\alpha = 0.8$ and $\beta = 0.999$. The model is trained for 10 epochs (around 75000 iterations).

4.4. Our Model

We train our model for 10 epochs with the same set of hyperparameters for the image model and the language model. For the ranking model, we use a learning rate of 10^{-5} and the RMSProp optimizer with $\alpha = 0.8$. In particular, we initialize the weight w_J for ranking loss to 10^{-6} (Softmax loss weight is set to 1), and double w_J every 5000 iterations. Intuitively, captions generated at initial stages are mostly random. We make w_J larger and enforce the ranking loss more strongly when the generated captions start to make more sense.



Baseline: a man and a woman are sitting on a bench.

Our model: a man sitting on a bench **with a dog**.

Baseline: a man in a suit and tie standing in front of a building.

Our model: a man is sitting on a bench **with a laptop**.

Baseline: a group of people standing on top of a **snow covered slope**.

Our model: a man **riding a skateboard** down a street.

Baseline: a man is sitting on a bench.

Our model: a **woman** sitting on a bench **in a park**.

Figure 4: Qualitative results. Green text indicates discriminatory captions, and red text indicates errors.

4.5. Results

To show the effectiveness of the ranking model, we train our model and the baseline model (which does not include the ranking loss) using the same set of hyperparameters. We trained both models for 10 epochs (around 75,000 iterations). The loss and the validation scores have not fully converged due to the limitation of computing power. We also cross-validate these models with different set of hyperparameters, and our model outperforms the baseline model consistently.

Quantitative Results Most of the existing models fail to capture the subtle differences of similar images, and this is due to the lack of discriminatory power in evaluation metrics. Therefore, we do not expect a significant boost in validation scores on these flawed metrics. We visualize the results in the following graphs: figure 3a shows the training and validation cross entropy loss, and figure 3b shows BLEU/METEOR/Cider scores on validation results. Note that there is an 8% increase (from 0.6 to 0.65) in CIDEr score, which indicates that the ranking model not only helps generate more discriminatory captions, but also increases the overall performance.

Qualitative Results As seen in figure 4, our model generates more descriptive and differentiable captions compared to those from the baseline model. In particular, our model is able to capture less salient objects and context such as ”laptop”, ”skateboard”, and ”dog”.

5. Conclusion

From the qualitative results, we can see that our ranking objective does add discriminatory power to the model. However, our model doesn't show significant improvement quantitatively. Things we would like to explore in the future:

- In the ranking model, replace the LSTM net with a Bidirectional LSTM net for learning sentence embedding.
- Instead of having randomly selected images in each batch, we can put similar images in the same batch. The ranking objective should be more effective in this case because there is no need to further push down the misaligned image-sentence pairs if all the images are very different.
- Add an adversarial objective that enables the model to generate captions with a distribution closer to ground truth captions.

Code for this project can be found at `https://github.com/telin0411/CS231A_Project`.

6. Miscellaneous

This is a joint project with CS224D (Deep Learning for Natural Language Processing). The image model (image feature extraction with Convolutional Neural Networks) is more relevant to this class, while the language model (LSTM model) is more relevant to CS224D.

References

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

[2] 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.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

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

[5] 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.

[6] A. Karpathy and J. Johnson. NeuralTalk2. `https://github.com/karpathy/neuraltalk2/`, 2015. [Online; accessed 1-June-2016].

[7] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv*, 2014.

[8] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 595–603, 2014.

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

[10] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[11] 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.

[12] 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.

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

[14] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[15] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.

[16] 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.