MMT: A Multimodal Translator for Image Captioning

Chang Liu[1], Fuchun Sun[1], and Changhu Wang[2]

[1]Department of Computer Science, Tsinghua University,
[2] Tou Tiao AI Lab

Abstract. In this work, we formulate the problem of image captioning as a multimodal translation task. Analogous to machine translation, we present a sequence-to-sequence recurrent neural network (RNN) model for image caption generation. Different from most existing work where the whole image is represented by a convolutional neural network (CNN) feature, we propose to represent the input image as a sequence of detected objects to serve as the source sequence of the RNN model. In this way, the sequential representation of an image can be naturally translated into a sequence of words, as the target sequence of the RNN model. To obtain the sequential representation of an image, objects are first detected by well-trained detectors and then converted to a sequential representation by some ordering strategies. Extensive experiments are conducted to evaluate the proposed approach on benchmark dataset, i.e.,MSCOCO, and achieve the state-of-the-art performance. The proposed approach is also evaluated by the evaluation server of MS COCO captioning challenge and achieves very competitive results. [1]

Keywords: Image Captioning, Deep Learning, Natual Language Generation

1 Introduction

Image captioning is a challenging problem. Different from other computer vision tasks such as image classification and object detection, image captioning requires not only understanding the image, but also the knowledge of natural language. Early methods on image captioning either explore with template-based, e.g., [21, 6] or retrieval-based approaches, e.g., [12, 16]. However, the language models are usually heavily hand-designed, and suffer from the problem of generating novel sentences with new compositions.

Inspired by the success of sequence-to-sequence machine translation [25] based on recurrent neural networks (RNN), recent approaches on image captioning have brought new insights by using a two-stage 'encoding' and 'decoding' technique [5, 19, 20, 14, 13, 27]. The common idea of these approaches is to use the whole CNN feature of the image as the 'source' input, to replace the words of the 'source language' in the translation task. The caption is then generated by conditioning the output words on the CNN feature of the image. One problem these approaches might suffer from is the imbalance of the visual part (representation of the image) and the language part (representation of words), because in

[1] Acknowledgment: This paper is jointly supported by National Natural Science Foundation of China under with Grant No.61621136008, 61327809,61210013,91420302 & 91520201

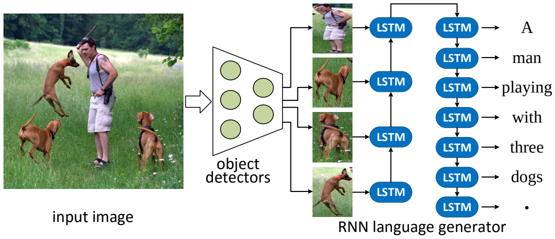


Fig. 1: The proposed framework. We represent the input image as a sequence of detected objects to serve as the source sequence of the RNN model. Then, this sequence of objects are further translated to a sequence of words, as the target sequence of the model.

RNN the image in the 'source language' only provides one single CNN feature at one time step, while the words in the 'target language' contribute at multiple time steps. Since the power of RNN lies in its capability in modeling the contextual information between each time step [11], such image representation weakens the RNN's memory of the visual information as it contains no temporal concept.

In this work, we follow the idea of CNN visual feature plus RNN model, but expand the image representation from one single CNN feature at a single time step in RNN, to a sequence of detected objects at multiple time steps. In this way, analogous to machine translation, image captioning is formulated as translating a visual language (a sequence of objects) to a natural language (a sequence of words). On the one hand, we can leverage object detection techniques to encode more visual information in the visual language; on the other hand, the sequential representation of the image accords more with the temporal concept of RNN models, and makes the two sides of translation more balanced.

We present a multimodal translation model for image captioning, to translate the visual language with a sequence of objects to a natural language with a sequence of words, as illustrated in Fig. 1. To detect the objects in an image, any existing well-trained object detector can be applied if it has a good category coverage. To convert the objects to a sequential order, several heuristic strategies are introduced and compared. Extensive experiments are conducted to show the effectiveness of the proposed solution on benchmark datasets, and achieve state-of-the-art performance.

2 Related Work

CNN+RNN based captioning A typical way is to combine CNN and RNN, where CNN is used to extract the feature of the whole image, and RNN to construct the language model. For example, Vinyals *et al.* [27] proposed an end-to-end model composed of a CNN and an RNN. The model is trained to maximize the likelihood of the target sentence given the CNN feature of the training image at the initial time step. Mao *et al.* [19] presented an m-RNN model, where the CNN feature of the image is fed into the multimodal layer after the recurrent layer rather than the initial time step. Similar work that utilizes CNN and RNN to generate descriptions includes [20, 3, 14]. However, most of above methods represent the image in a static form, such as a 4096-d CNN feature vector. Although the feature can well represent an image, it is insufficient for the sequential RNN model. That is because such a feature only provides the encoding phase of the RNN model with a single time-step data, leaving the rest of the model to the decoding phase where words in the caption are used.

Object based captioning Before the wide use of RNN, the methods of leveraging visual information of objects instead of the whole image had arisen [15, 21, 17]. They utilize object detectors [8] to obtain object names in the image, and construct tuples composed of the name and attribute description of the word. These tuples are then fed into a heavily designed language model, in which the visual features of objects are not leveraged. Some other approaches utilize the visual features of objects [14, 7]. Karpathy *et al.* [14] leveraged the features for word-region alignment rather than image captioning. Fang *et al.* [7] trained words detectors based on object features, and generated the sentence by feeding the detected words into a maximum-entropy language model. Xu *et al.* [28] incorporated visual attention to image captioning, where the model simultaneously reads the low-level visual patches and the words at each timestep to learn the correspondence between the two modalities. Our work differs from above approaches as we use the features of objects in a sequential form, and model the task as multimodal translation.

3 Proposed Method

3.1 Formulation and Overview

In previous work with CNN+RNN solutions, the core idea is usually to maximize the probability of the description given the input image:

$$\log p(S|I) = \sum_{t=0}^{N} \log p(S_t|I, S_{0:t-1}), \tag{1}$$

where I represents the image, S_i is the ith word in sentence S, and $p(S_t|I, S_{0:t-1})$ is the probability of generating word S_t given the image and previous words $S_{0:t-1}$. A common representation of the image is a CNN feature vector, and the recursive language part is usually modeled with recurrent neural networks (RNN), where an RNN unit considers the following two data as inputs: (1) input at current time step t, and (2) output from the previous time step $t-1$.

RNN has the capability of modeling sequential data in variant lengths, which works well for image captioning since the length of image description and the number of objects are not fixed.

In this work, we formulate image captioning as a multimodal translation task, i.e., translating a visual language (a sequence of objects) to a natural language (a sequence of words). The core idea is to use RNN to model the translation process, by feeding one object at a time to an RNN unit during encoding, and one word at a time during decoding.

Specifically, given an image I, we use $seq(I)$ to denote its sequential representation, which contains a list of objects in a specific order $seq(I) = \{O_{l_1}, O_{l_2}, ..., O_{l_m}\}$. Then the RNN takes in $seq(I)$ by encoding them into a hidden space, and is recursively activated at each time step, as shown in Fig. 2. The sentence is generated by conditioning the outputs given the last hidden representation h (Eqn. 2), where h is obtained from the source sequence of objects (Eqn. 3).

$$\log p(S|seq(I)) = \sum_{t=0}^{N} \log p(S_t|h, S_0, S_1, ..., S_{t-1}) \tag{2}$$

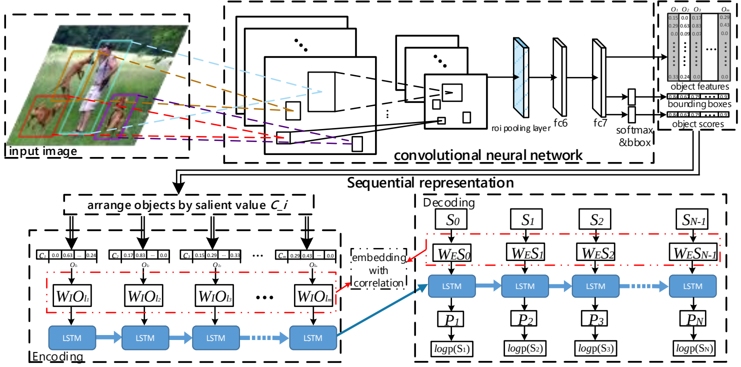


Fig. 2: Overview of the Model. We first detect objects in the image using Fast R-CNN [9], and arrange them in a specific order. The source sequence is represented with the embedding of object features in a hidden space; the target sequence is represented with the embedding of the words in the same space. The mapping from the source to the target is modeled with LSTM units in an unrolled version. All the units share the same parameters. The target word is generated by a softmax over the vocabulary given the output of the last hidden unit of the source sequence, as well as previous words in the target sequence.

$$h = \underset{t=1}{\overset{m}{RNN}}(seq(I)_t) \tag{3}$$

Besides the model itself, several issues still need to be solved: (1) how to detect and represent the objects in the image, and how to deal with the mismatch between detected objects and words in the caption; (2) how to determine the order of objects for the sequential representation.

3.2 Model Details

Below we introduce three components of the model: source sequence representation, target sequence representation, and RNN translation from source to target.

Source sequence representation The source sequence is represented with the visual information of the image. We first use object detectors to locate the objects and extract D_o-dimension CNN features, denoted as $\text{CNN}(O_i)$ for the i-th object O_i. The objects are arranged in an order denoted as l_i. Then object features are mapped into an H-dimension hidden space with embedding matrix W_I, with $H \times D_o$ dimension. The source sequence of the RNN is represented as:

$$x_t = W_I \text{CNN}(O_{l_t}), \quad t \in \{1, 2, ..., m\}, \tag{4}$$

where t is the time step of the network, and m is the total number of objects.

Target sequence representation The target sequence is represented with a set of words $S_0, S_1, ..., S_N$ in the sentence S, where S_N stands for a special symbol to denote the end of the sentence. Each word S_i is represented as a 'one-hot' vector, with a dimension D_s equal to the vocabulary size. Then the words are mapped to the same hidden space with word embedding matrix W_E, with a dimension of $H \times D_s$. Thus the target sequence is represented by:

$$x_t = W_E S_{t-m-1}, \quad t \in \{\text{m+1,m+2,...,m+N}\}, \tag{5}$$

where the subscript of S takes the length of the source sequence into account.

RNN translation from source to target To model the translation from the source sequence to the target sequence, we choose to adopt recurrent neural networks (RNN), as RNN has shown good capability on modeling temporal data in sequence in different research areas. Specifically, we leverage the long-short term memory (LSTM) to avoid the gradient exploding and vanishing problem of the network [11, 10].

LSTM takes in the output of the previous time step, as well as the input at the current time step, as the inputs of the current unit. To better illustrate the idea of this recursive process, we unroll the LSTM along the time dimension, by copying the LSTM unit at each time step, as shown in Fig. 2. The LSTM units thus share the same parameters. The core of an LSTM unit is a memory cell c, which is controlled by several gates. The activation of each gate determines whether the corresponding input is accepted or rejected. Thus the mapping from the source (Eqn. 4) to the target (Eqn. 5) is formulated by the following equations:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i), \quad f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f), \tag{6}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o), \quad g_t = \sigma(W_{xg}x_t + W_{hg}h_{t-1} + b_g), \tag{7}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t, \quad h_t = o_t \odot \phi(c_t), \quad p_{t+1} = Softmax(h_t), \tag{8}$$

where t ranges from the start of the source sequence to the end of the target sequence in all equations except the last one, where t starts from the decoding step; i_t, f_t, o_t, and g_t represent the input gate, forget gate, output gate, and input modulation gate at time step t, respectively; c_t and h_t are the memory cell and the hidden state; W_{ij} represents the connection matrix and b_j is the bias; σ is the sigmoid non-linearity operator, and ϕ is the hyperbolic tangent non-linearity; $\odot$ is the element-wise multiplication operator; and the distribution p_{t+1} is given by a softmax over the entire words in the vocabulary using the hidden output h_t.

3.3 Object Detection and Ordering

In the proposed approach, we represent the input image as a sequence of objects. Thus we need an object detector to locate salient objects in the image. We leverage the Fast R-CNN algorithm [9] to detect the objects, and the $fc7$ layer features of the VGG16 model [23] for representation.

To increase the object coverage, we can directly train detectors on the caption datasets, i.e., MSCOCO, based on additional object-level labels. The basic idea is to train semantic-related object detectors based on the referred objects mentioned in the image captions of the training set. The object detector is trained on the MSCOCO detection dataset, where the images and splits are the same as the caption dataset.

The order of the sequential representation is of importance because our model memorizes the relationships between the input data at time steps. The order of feeding the objects to the RNN model will influence the memory of the visual part of the model and further influence the performance. We define a ‘salient value’ for each object O_i, denoted by C_i, as the products of the detection score and the size of the bounding box of O_i, i.e., $C_i = score_i * \frac{S(O_i)}{S(I)}$, where $S(x)$ represents the area of x. The salient values of objects are leveraged to determine the order to feed to the proposed model. To explore the influence of the sequence order in image captioning, several variants are adopted in the experiments: 1) random order, denoted as MMT-rnd; 2) descending order, denoted as MMT-desc; and 3) ascending order, denoted as MMT-asc. MMT stands for Multimodal Translator.

3.4 Training and Inference

Training As shown in Fig. 2, during the encoding phase, the model reads the encoded object features one by one. After that, the model enters into the decoding phase and read the encoded words in the sentence. The loss is the generation loss (L_G)' in Eqn. 9, which is the negative of the sum of the log probabilities given the last hidden representation h and previous words.

$$L_G = -\sum_{t=1}^{N} \log p_t(S_t|h, S_0, S_1, ..., S_{t-1}). \tag{9}$$

The network is trained with SGD, with the minibatch size 100 and learning rate 1e-3. Both of object encoding and word encoding sizes are set to 256. The vocabulary is preprocessed with a word occurrence threshold of 5. To avoid overfitting, we also leverage regularization and dropout [24] (0.5).

Inference Given a testing image, the objects are first detected and represented by CNN features, after which these features are fed in to the encoding phase of the model. In decoding, the model generates a word at each time step by selecting the max probability given the objects and previous words.

4 Experimental Results

4.1 Datasets and Evaluation Measurements

MSCOCO [18] contains 82,783 training, 40,504 validation and 40,775 testing images. To compare with state-of-the-art methods, we follow previous work [14, 28, 13], where 82,783 training data are used for training, 5,000 validation images for validation and another 5,000 for testing. Moreover, we also validate our model on MSCOCO evaluation server, where 40,755 testing images are withheld for testing.

Evaluation Measurements: BLEU [22] is a widely used evaluation measurement for image captioning. We adopt BLEU-1,2,3,4, CIDEr [26], RougeL, and METEOR [1] based on the evaluation tool [2]. For the scores of all measures, the higher the better.

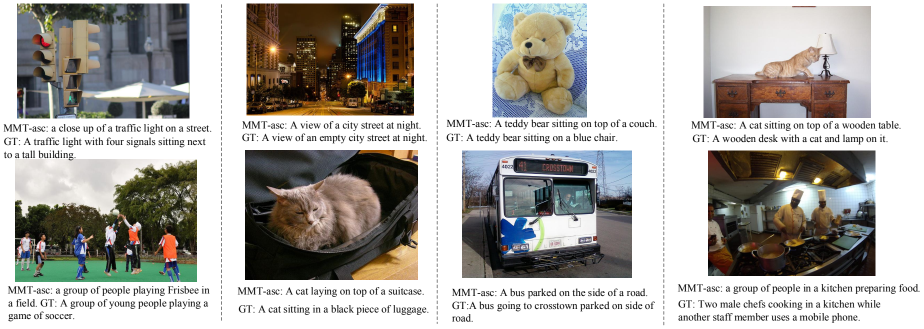
4.2 Overall Comparison with the State of the Arts

The performance of the proposed approach with three ordering strategies is shown, i.e., MMT-rnd, MMT-desc, MMT-asc. We also implemented a 'Baseline' algorithm, where the only difference is to represent the image using one CNN feature, instead of object features. **MSCOCO**: The results are shown in Table 1. It has parts: 1) upper part, the results on 5,000 validation images, which are presented in related papers; and 2) lower part, the results on 40,755 testing images, evaluated by the evaluation server of MS COCO captioning challenge.

From Table 1 we can see the superiority of our method over existing state-of-the-art algorithms, for we achieved the best results for most measures. Some randomly selected caption results on MSCOCO are shown in Fig. 3.

5 Conclusion

In this paper, we formulated the problem of image captioning as a multimodal translation task, that is, translating the visual language with a sequence of objects to a natural language with a sequence of words. Based on this formulation,



MMT-asc: a close up of a traffic light on a street. GT: A traffic light with four signals sitting next to a tall building.

MMT-asc: A view of a city street at night. GT: A view of an empty city street at night.

MMT-asc: A teddy bear sitting on top of a couch. GT: A teddy bear sitting on a blue chair.

MMT-asc: A cat sitting on top of a wooden table. GT: A wooden desk with a cat and lamp on it.

MMT-asc: a group of people playing Frisbee in a field. GT: A group of young people playing a game of soccer.

MMT-asc: A cat laying on top of a suitcase. GT: A cat sitting in a black piece of luggage.

MMT-asc: A bus parked on the side of a road. GT:A bus going to crosstown parked on side of road.

MMT-asc: a group of people in a kitchen preparing food. GT: Two male chefs cooking in a kitchen while another staff member uses a mobile phone.

Fig. 3: Randomly selected images from MSCOCO validation set with ground truth descriptions (GT) and MMT-asc results.

Table 1: Comparison results on MSCOCO. Upper part: the results on 5000 validation images; Lower part: the results on 40,755 testing images evaluated by the evaluation server of MS COCO. All methods with valid references are listed. The highest score is labeled as bold, and the 2nd is underlined.

	MS COCO, 5000 test images						
Methods	CIDEr	METEOR	RougeL	BLEU-1	BLEU-2	BLEU-3	BLEU-4
m-RNN(Baidu) [19]	90.9	23.5	51.9	71.8	55.0	40.9	30.5
Attention [28]	-	23.0	-	71.8	50.4	35.7	25.0
GLSTM [13]	81.3	22.7	-	67.0	49.1	35.8	26.4
DeepVS [14]	66.0	19.5	-	62.5	45.0	32.1	23.0
baseline	76.1	21.4	48.3	65.5	47.5	33.8	24.6
MMT-rnd	81.4	22.1	49.3	67.7	49.9	36.3	26.6
MMT-desc	89.6	23.5	51.9	70.5	53.6	40.0	30.0
MMT-asc	**93.6**	**24.0**	**52.6**	**72.6**	**55.9**	**41.7**	**31.1**
	MS COCO test server 2014, 40775 images C5						
Methods	CIDEr	METEOR	RougeL	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Google NIC [27]	**94.3**	**25.4**	53	71.3	54.2	40.7	30.9
MSR Captivator [4]	93.1	24.8	52.6	71.5	54.3	40.7	30.8
Berkeley LRCN [5]	92.1	24.7	52.8	71.8	54.8	40.9	30.6
m-RNN(Baidu) [19]	88.6	23.8	52.4	72.0	55.3	41.0	30.2
Attention [28]	86.5	24.1	51.6	70.5	52.8	38.3	27.7
DeepVS [14]	67.4	21.0	47.5	65.0	46.4	32.1	22.4
MMT-asc	93.2	24.6	**53.2**	**73.7**	**56.7**	**42.1**	**31.1**

we proposed a sequence-to-sequence RNN model for image captioning, in which the image was represented as a sequence of object features. We explored several important issues in the proposed method, and provided practical solutions. The proposed method was evaluated in extensive experiments, and achieved state-of-the-art performance on benchmark datasets.

References

1. Banerjee, S., Lavie, A.: Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In: ACL workshop (2005)
2. Chen, X., Fang, H., Lin, T., Vedantam, R., Gupta, S., Dollr, P., Zitnick, C.L.: Microsoft coco captions: Data collection and evaluation server. arXiv:1504.00325 (2015)
3. Chen, X., Lawrence Zitnick, C.: Mind's eye: A recurrent visual representation for image caption generation. In: CVPR (2015)
4. Devlin, J., Cheng, H., Fang, H., Gupta, S., Deng, L., He, X., Zweig, G., Mitchell, M.: Language models for image captioning: The quirks and what works. arXiv preprint arXiv:1505.01809 (2015)

5. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: CVPR (2015)
6. Elliott, D., Keller, F.: Image description using visual dependency representations. In: EMNLP. pp. 1292–1302 (2013)
7. Fang, H., Gupta, S., Iandola, F., Srivastava, R.K., Deng, L., Dollar, P., Gao, J., He, X., Mitchell, M., Platt, J.C., Lawrence Zitnick, C., Zweig, G.: From captions to visual concepts and back. In: CVPR (2015)
8. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. PAMI (2010)
9. Girshick, R.: Fast R-CNN. In: Proceedings of the International Conference on Computer Vision (ICCV) (2015)
10. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies (2001)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
12. Hodosh, M., Young, P., Hockenmaier, J.: Framing image description as a ranking task: Data, models and evaluation metrics. JAIR (2013)
13. Jia, X., Gavves, E., Fernando, B., Tuytelaars, T.: Guiding the long-short term memory model for image caption generation. In: ICCV (2015)
14. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: CVPR (2015)
15. Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A.C., Berg, T.L.: Baby talk: Understanding and generating image descriptions. In: CVPR (2011)
16. Kuznetsova, P., Ordonez, V., Berg, T.L., Choi, Y.: Treetalk: Composition and compression of trees for image descriptions. ACL (2014)
17. Li, S., Kulkarni, G., Berg, T.L., Berg, A.C., Choi, Y.: Composing simple image descriptions using web-scale n-grams. In: CoNLL (2011)
18. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
19. Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., Yuille, A.: Deep captioning with multimodal recurrent neural networks (m-rnn). ICLR (2015)
20. Mao, J., Xu, W., Yang, Y., Wang, J., Yuille, A.L.: Explain images with multimodal recurrent neural networks. NIPS Deep Learning Workshop (2014)
21. Mitchell, M., Han, X., Dodge, J., Mensch, A., Goyal, A., Berg, A., Yamaguchi, K., Berg, T., Stratos, K., Daumé III, H.: Midge: Generating image descriptions from computer vision detections. In: EACL (2012)
22. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: ACL (2002)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
24. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. JMLR (2014)
25. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS (2014)
26. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: Cider: Consensus-based image description evaluation. In: CVPR (2015)
27. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015)
28. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A.C., Salakhutdinov, R., Zemel, R.S., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: ICML (2015)