Image Captioning by Parts

Chintak Sheth
Stony Brook University, Stony Brook, NY 11794, USA
csheth@cs.stonybrook.edu

Abstract

In this paper, we implement and analyse the performance of Neural Image Captioning [15] model on the Flickr8k and Flickr30k datasets. We identify some of the biases of the Flickr8k dataset which allows captions for certain objects to be predicted easily but restricts its power to generalize. We analyze the learnt word embeddings and explain some of the discrepencies for the observed drop in performance.

1. Introduction

Automatically describing the content of an image is a very challenging task, but it can also have a great impact, for instance, bi-directional text-image retrieval can enable search over the content in the image rather than simple keyword based search. It can be assist the visually impaired people better understand the world around them and the vast content on the internet. This task is significantly more complex than the task of image classification or object recognition task which have received the majority of the focus from the computer vision community. A description of an image must capture the not only the objects contained in the image but also the interactions between these objects. Indeed, a description must be able to express how the objects relate to each other and also the activity they are involved in.

Generally, there are 4 components in the image captioning pipeline - extracting image features, generating a set of candidate words, generating a set of sentences/captions and finally, ranking the sentences according to an evaluation metric to choose the sentence with the highest score. Certain approaches combine two or more components into a single block to solve the problem.

The usual trend for representing image features is to use a convolutional neural network (CNN) pre-trained on the ImageNet Large Scale Visual Recognition Challenge [13] dataset. Recent successes in object recognition, image classification and other related tasks have demonstrated the versatility of these features. The VGGNet [14] are usually favoured for extracting discriminating features. Statistical

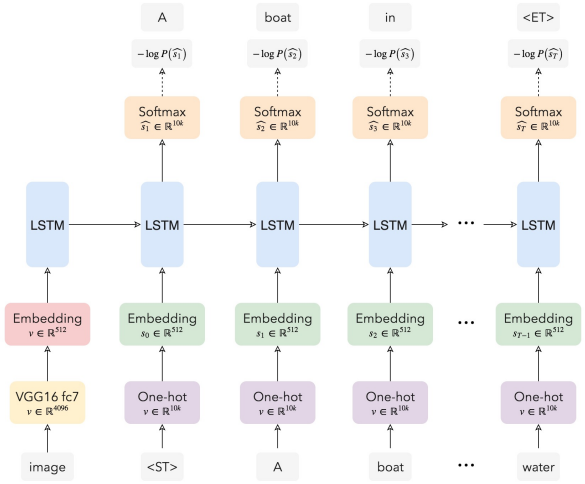


Figure 1: Example of caption. It is set in Roman so that mathematics (always set in Roman: $B \sin A = A \sin B$) may be included without an ugly clash.

natural language processing methods like the Maximum-Entropy model is used to build a language model. A left-to-right Beam search is used to generate a set of candidate sentences using the language model. Lastly, MERT [10] is used to rank the candidate sentences according to an evaluation metric. The output caption is the sentence with the highest score.

2. Related Work

Since the introduction of Microsoft COCO [7] in 2014, a lot of work involving deep neural networks has been published. Many of these approaches leverage the recent successes in recognition of objects, their attributes and locations, to generate natural language descriptions.

Fang *et al*. [3] use Multiple Instance Learning in conjunction with a CNN to output the set of likely words contained in an image. A Maximum-Entropy [1] model is used for modelling the likelihood of seeing a word at location l in a sentence from the dictionary given the sequence of words

prior to the current word. In addition, they condition the likelihood on the set of candidate words generated in the first block. Using beam search, they generate a set of candidate sentences/captions and lastly, ranking of captions is done using MERT.

On the other extreme [9, 15, 16] train a joint deep neural network which takes as input an image I and is trained to maximise the likelihood $p(S|I)$ of producing a target sequence of words $S = S_1, S_2, \ldots$ where each word S_t comes from a given dictionary. In this approach they combine all the 4 components into a single component and achieve state-of-the-art results, showcasing that there is some merit in mapping the features of the image and text to the same embedding space and trying to reduce the distance between image and it's captions. The inspiration for this approach comes from the task of machine translation where the input is a sequence of words in a source language and the output is a semantically equivalent sequence of words in a target language. Traditionally, this task was solved by combining solutions to the various sub-problems but recent work has shown that translation can be done in a much simpler was using Recurrent Neural Networks (RNN). Briefly, there is an *encoder* RNN which learns a fixed-high dimensional embedding for the input sequence of words. This embedding is then fed as an input for another *decoder* RNN which generates a sequence of words in the target language given the embedding for the words in the source language.

3. Model

Inspired by [15], we implement a model for image captioning which is trainable end-to-end. Recent progress in statistical machine translation have achieved state-of-the-art results by directly maximizing the probability of the correct translation in the target language given the input sentence in the source language. These models are based on utilizing the power of recurrent neural networks in learning representations which serve as a mapping between words in the source language and words in the target language while preserving syntactic and semantic meaning of the words.

Following this success, [15] propose a model to maximize the probability of generating a correct caption given the input image by using the following formulation:

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

where θ are the parameters of the model, I is the input image and S is its correct caption. Here, S represents a sequence of words and is usually written in terms of each word by using the product rule as,

$$\log p(S|I;\theta) = \sum_{t=0}^{T} \log p(s_t|I, s_0, s_1, \ldots, s_{t-1};\theta) \tag{2}$$

where, s_t corresponds to the embedding for the tth word in the input caption. We model $\log p(s_t|I, s_0, s_1, \ldots, s_{t-1};\theta)$ using a variant of the Recurrent Neural Network (RNN) called Long-Short Term Memory (LSTM) [4]. The prediction of a word at any time t is conditioned on the input image and the past predictions from time 0 to $t-1$. Next, we describe how we encode the image and each word in the caption.

3.1. Image Encoder

We use the 4096-dimensional `fc7` layer of the VGGNet [14] pre-trained on the ILSVRC 2014 classification datset [13]. We use the VGG model with a total of 16 layers. This model consists of 13 convolutional layers with 3×3 kernels. Deep convolutional layers allow the network to learn non-linear functions from the input to the output. The filters learnt by the model loosely correspond to the layered processing of the human visual cortex system. The filters learnt in the lower most layers correspond to edge and color blob filters, while in the middle layers they correspond to parts of objects such as eyes, nose, wheel, etc whereas in the uppermost layer the filters may be sensitive to objects such as humans, cats, dogs, etc. Furthermore, the features learnt by the model have been shown to generalize of various tasks including scene classification by means of transfer learning [2]. This shows the efficacy of the VGGNet in encoding the high dimensional 224×224 pixel images into discriminative and compact 4096-D vectors. We also add an intermediate fully connected layer with 512 units to further reduce the dimensionality of the image vectors, and to match the dimensionality with the word embedding vectors.

3.2. LSTM based Language Model

LSTM has been used successfully for modelling machine translation problem. It works by representing the words seen from time 0 to $t-1$ using a fixed length hidden state or memory m_t. It then, describes the mapping over the next time step given the word embedding for the current word and the hidden state for all the words seen so far. It was introduced in [4] to deal with the problem of vanishing and exploding gradients while training Recurrent Neural Networks.

In Figure 1, the LSTM is shown in the time-unrolled form. This form of representation allows us to interpret the LSTM as a feed-forward network which makes the analysis easier. In essence, this is just a type of Bayesian Network who's joint distribution can be given as the product of the conditional distributions. The core of the LSTM is a memory cell c which encodes the knowledge of the sequence of words it has seen at every time step. The memory cell is shown in Figure 2. Three gates - input gate i, forget gate f and output gate o - control the behaviour of the LSTM. These gates help LSTM deal with the vanishing and ex-

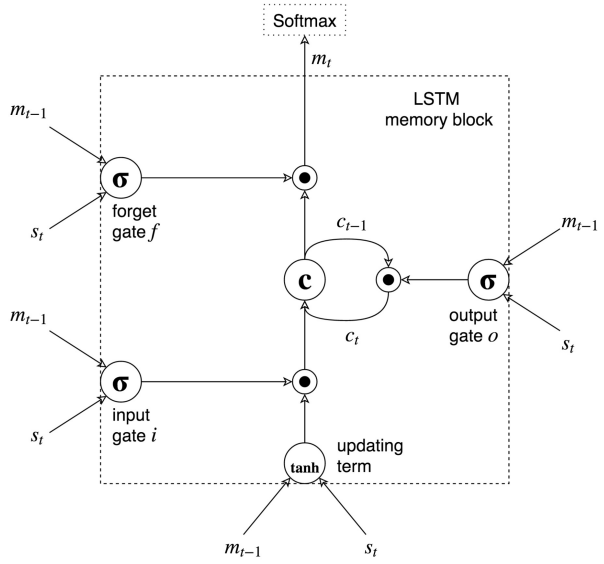


Figure 2: Memory cell c of a LSTM. This component is reponsible for "remembering" previous items. The input gate i controls the extent of new information entering the cell. The forget gate f enables the cell state c to persist, or erase itself. The output gate o controls the amount of information passed on from the cell state to the output of the cell block.

ploding gradients problem. The definition of the gates is as follows:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1}) \quad (3)$$
$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1}) \quad (4)$$
$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1}) \quad (5)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{cx}x_t + W_{cm}m_{t-1}) \quad (6)$$
$$m_t = o_t \odot c_t \quad (7)$$
$$p_{t+1} = \text{Softmax}(m_t) \quad (8)$$

where $\odot$ represents elementwise multiplication. The W matrices are the trainable parameters. The input gate i controls the amount of information which flows into the LSTM at any given time step. The forget gate f allows the cell to store the information for avery long time by reinforcing the signal or alternatively, it could also "forget" and reset the cell state. The output gate determines the proportion of information that flows from the current input embedding and the previous cell state. The W matrices are time-independent, that is, they are shared across time. This helps in reducing the number of parameters learnt by the model and acts as a natural check against overfitting.

Training The LSTM model is trained to predict each word of the sentence after it has seen the image and all the preceding words. This probability was defined in 2. The input to the model is an image-caption pair. The entire model can be described by the following set of equations:

$$x_{-1} = W_{ie}\text{CNN}(I) \quad (9)$$
$$x_t = W_{se}s_t, \quad t \in 0, \ldots N-1 \quad (10)$$
$$s_t = \text{OneHot}(w_t) \quad (11)$$
$$p_{t+1} = \text{LSTM}(x_t) \quad (12)$$

where w_t corresponds to the index of the word at time t in the vocabulary. CNN(I) corresponds to the output of the VGG `fc7` layer. OneHot(w_t) corresponds to the one-hot vector for representing a word in the vocabulary. LSTM(x_t) represents the output of the LSTM for seeing t words and input image I. The W matrices have equal number of rows and are the trainable parameters of the model. W_{ie} tranforms the `fc7` layer features to an embedding space of 512-D. W_{se} transforms the one-hot vector of a word to an embedding space of 512-D. This operation corresponds to mapping the image and words to a common embedding so that they can be used together in the LSTM.

In more detail, we first extract the VGG `fc7` (CNN(I)) layer features for the input image I. By feed-forwarding these features through the image embedding layer we transform the image into a common embedding space as the words. We represent each word in the input caption as an one-hot vector which are transformed to a common embedding space using a fully connected word embedding layer. We use this image embedding x_{-1} to initialize the memory cell of the LSTM. After this the word embeddings are fed into the LSTM at different time steps. The output of the LSTM is connected to another fully connected layer which has a hidden unit corresponding to every word in the vocabulary. This can be viewed as a k class classification problem. It is natural to convert the scores to probability by applying a softmax on output of this layer and compute the cross-entropy loss for the word classification. We sum up the log losses for all the words in the sequence to calculate the loss for the entire sequence. Hence, the negative log likelihood is given as:

$$L(I, S) = -\sum_{t=1}^{N} \log p_t(s_t) \quad (13)$$

The above loss is minimized w.r.t. all the parameters of the model - W matrices of the LSTM, W_{ie} matrix for the image embedding layer and W_{se} matrix for the word embedding layer.

Inference During testing time, we only feed in the input image to the network and the model generates the corresponding caption for the image. **Sampling**: In order to

Metric	BLEU-1	BLEU-4	METEOR
MSCOCO			
NIC	–	27.7	23.7
Random	–	4.6	9.0
Nearest Neighbor	–	9.9	15.7
Human	–	21.7	25.2
Flickr8k			
NIC	63	–	–
Our	57.3	14.7	15.5
Flickr30k			
NIC	66	–	–
Our	59.1	16.7	17.9

Table 1: Evaluation metrics. Comparison with NIC [15], Nearest Neighbor, Random and Human evaluations as proposed in [15]. They have provided results for the MSCOCO [7] dataset. We evaluate our model for Flickr30k and Flickr8k datasets and compare the BLEU-1 score provided by NIC.

evaluate the performance of LSTM as a language model we sample the most likely word at each time step of the LSTM as our generated caption. There is an alternate method called **Beam Search** available for generating captions using a language model. In this approach, instead of generating a single caption we simultaneously maintain k best candidate captions and finally select the one with the highest score as our final caption. We only experiment with the Sampling method for generating captions.

4. Experiments

We evaluate our implementation on the Flickr8k [5] and Flickr30k [18] datasets. The Flickr8k dataset consists of 8091 images with 5 captions per image. The Flickr30k dataset consists of 31783 images with 5 captions per image.

4.1. Evaluation Metrics

We evaluate our model using two popular metrics which were originally developed for evaluating machine translation performance - BLEU [11] and METEOR [6]. BLEU computes the amount of N-gram overlap between the generated caption and the reference captions. It can be viewed as a form of precision of word N-grams between the hypothesis caption and the reference captions. Usually, BLEU scores with up to 4-grams are reported. In addition, we also report METEOR scores. METEOR scores the hypothesis caption by aligning it to one or more reference captions. The alignments are based on exact, stem, synonym or paraphrase match between words. In this sense, it is forgiving that BLEU scores which requires exact match to achieve a higher score. The scores are reported in Table 1.

4.2. Qualitative Analysis

In Table 2, we show some sample captions generated by our system and compare them with the ground truth captions provided. We note that our model tends to produce more generic captions in comparison to the ground truth captions given. Also on closer inspection, we find that images containing "dogs" tend to be annotated correctly by our model.

Table 3 shows some more examples of partially correct and incorrect captions generated by our model. We find that our model has difficulty identifying activities accurately. As an example, in the bottom left image in Table 3, our model claims that the man is doing "rock climbing", when in fact he is resting on the rock. It appears that our model is learning to associate activities with gross level image information such as "rock climbing" if there is a big rock and a person in the scene, or "running" when there is a dog in scene.

4.3. Failure Cases

Table 4 shows images for which our model generated incorrect captions. In the image on the left hand side, our model completely gets the color information wrong although, is able to correctly notice that the "person" is "playing with a dog". In the second image, our model incorrectly generates that the person is "surfing on a snowy hill".

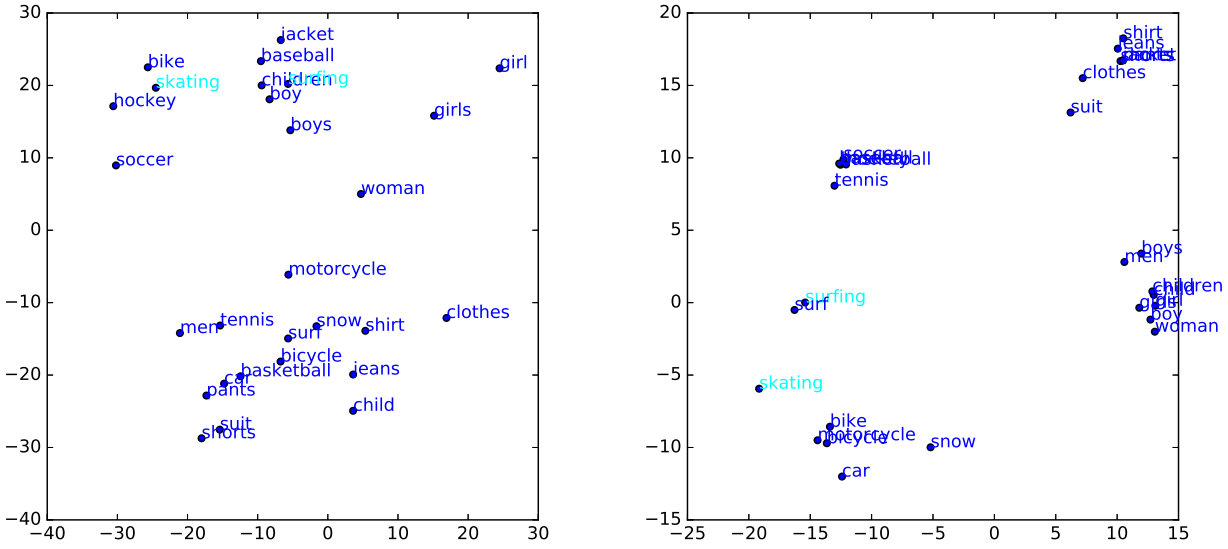
4.4. Further Analysis

In order to further inspect the behaviour and gain better intuition about the features that the model is learning we plot a word cloud for the words in the Flickr8k dataset. The word cloud is plotted in Figure 4. Thus, we identify that "dog" is the most frequently mentioned object in the captions and consequently most images in the dataset correspond to dogs. This explains the fact that our model is able to perform better when a dog is present in the scene.

The core task in image captioning is learning appropriate embeddings for image and word which enables us to correctly describe a given image. In our model we are learning the word embeddings from scratch. Hence, we can get a good estimate about the model by visualizing the word embeddings of the trained model. In Figure 3, we plot the word embeddings using t-SNE [8]. We compare this with the t-SNE visualization of the 300-D GloVe [12] vectors trained on 6 billion Wikipedia tokens. We observe that GloVe vectors for words like "tennis", "basketball", "baseball", "hockey" form a tight cluster in contrast to our vectors. This indicates that our model hasn't yet been able to learn proper discriminative embeddings. This can be attributed to the constraint posed by using just 30,000 captions (Flickr8k train set). This explains the discrepancies we noted above.

a black dog runs through the grass .	a black dog runs through water .
1. a black dog is running through a field of grass and dandilions .	1. a brown dog walks through a shallow stream .
2. a big thin dog runs fast in a field of grass .	2. the dog walks in the shallow water .
3. a dog runs through a field .	3. a black dog walking through splashing water in a rocky creek .
4. a very thin black dog running in a field .	4. a greenish dog runs through the water by several large rocks .
5. a black greyhound running in an open field .	5. a black dog splashes through greenish water .

Table 2: Success cases. First row: input image. Second row: generated caption. Third row: 5 ground truth captions from Flickr8k dataset.



(a) t-SNE for learnt embedding vectors.

(b) t-SNE for GloVe [12] vectors.

Figure 3: Comparison between the learnt language embedding vectors and GloVe [12] embedding vectors. Note how similar words are tightly clustered for GloVe vector embeddings but not yet tightly clustered for the learnt embeddings. (High resolution plot containing 1000 most popular words is provided as supplemental material.)

a black and white dog runs through the snow .

a baby is sitting in a stroller in a wooden room .

a man is riding a surfboard down a wave .

a man is rock climbing .

a man and woman sitting back on a rock .

a man and woman walk sunset through the water .

Table 3: More examples of generated captions. First-second row: images with partially correct captions. Third-fourth row: images with incorrect captions.

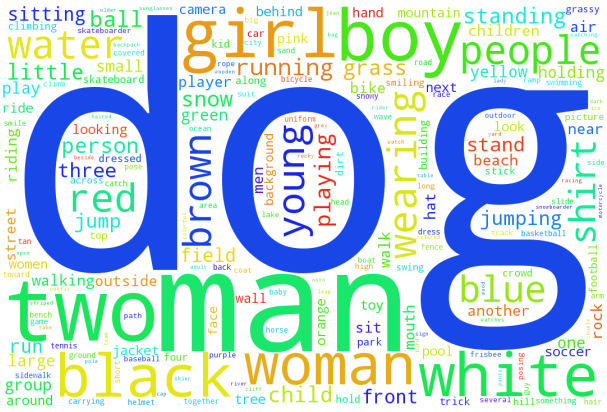


Figure 4: Word cloud visualizing the frequency of usage of words in Flick8k dataset. We observe that the most frequently occuring word in the dataset is ”dog”, hence, there is a high number of image-caption pairs containing dogs. This is the reason why we are able to perform well on images containing dogs.

5. Conclusions

In this project, we implemented the Neural Image Captioning [15] model and evaluated it on the Flickr8k and Flickr30k datasets. We also identified few of the biases implicit in the Flickr8k dataset which leads to good predictions for certain objects (*e.g.* dogs) but faces difficulties in generalizing. We analysed the performance of the model and noted the need of language model training to focus on finer-grained attributes of the image. We also identified the primary task in image captioning - learn a correspondence between the image embedding and the word embedding. The primary constraint seems to be the fact that learning the mapping from image embedding features to word embedding features requires a large number of image-captions pairs. A method for visualizing and understanding the efficacy of image embeddings in capturing fine-grained attributes of the image is another direction to gain better intuition about the task. Another direction is to follow attention modelling approaches such as [17] and [16]. Alternatively, initializing the word embeddings using GloVe vectors followed by fine-tuning could also lead to improvements.

a girl in a pink and white shirt is playing with a dog .

a man in a black jacket is surfing on a snowy hill .

a boy in a red shirt is jumping into a pool with a girl in his hand .

Table 4: Failure cases. First row: input image. Second row: generated caption. Third row: 5 ground truth captions from Flickr8k dataset.

References

[1] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71, 1996.

[2] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014.

[3] 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.

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

[5] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

[6] A. Lavie and A. Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231. Association for Computational Linguistics, 2007.

[7] 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.

[8] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[9] 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.

[10] F. J. Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics, 2003.

[11] 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.

[12] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43, 2014.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

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

[15] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[16] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2(3):5, 2015.

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

[18] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.