Image captioning using deep learning models

Tuhin Kundu
University of Illinois at Chicago
Chicago, IL
tkundu2@uic.edu

Sriparna Ghosh
University of Illinois at Chicago
Chicago, IL
sghosh37@uic.edu

Abstract

With an immense amount of visual information being generated and aggregated from various sources, making sense of this information and organising this data is becoming increasingly important. Image captioning is generating a meaningful grammatically correct sentence to understand the scene holistically. It translates visual information to textual information by generating a description of an image using deep learning models. We use datasets such as Flickr8, Flickr30 and COCO 2014 to investigate the trade-off between using various combinations of encoder-decoder based models which comprise of convolutional and recurrent neural networks. We add embeddings obtained from language models such as Glove and BERT as weight initialization to our decoder unit to check for performance gains and to lower training time. We conclude that contextual embeddings obtained from BERT provide a significant performance in terms of BLEU score due to taking into account contextual information that may be present in captions or sentences. The code is available on Github. [1]

1. Introduction

Image Captioning has various practical applications which make it an important problem to solve in the current times. Image captioning leverages two important concepts namely computer vision and natural language processing with deep learning models and generates a sentence to describe an image or scene.

1.1. Background

As seen in computer vision theory, there is a lot of information that is contained in an image and various mathematical equations can help extract different information from the image such as shapes, colors, objects, and their relationships. Understanding the scene and to describe what the image consists of holistically is a challenging task and a popular research area in Artificial Intelligence. This task involves not only understanding the scene very well, which would require a lot of training data but also explain it effectively using a generated sentence in English, which would require a language model.

The first task in image captioning starts with identifyint the different objects in an image and then relate them to understand the context and then describe it using a syntactically and semantically correct sentence.

Deep learning based techniques are used to learn various complex features automatically from training data to handle a variety of data including images and videos. The most widely used technique for feature learning is using Convolution Neural Networks(CNN), with Softmax used to classify the images. The results from the CNN is then fed to a Recurrent Neural Network(RNN) to generate the captions.

1.2. Motivation of this project

The breadth of use cases for Image Captioning is a strong motivation to solve this problem. The following are some of the use cases that have direct application of Image Captioning:

- **Self Driving Cars** : Self driving cars need to constantly identify the scene in front of the car to detect objects, actions and their relationships to drive safely and perform various actions based on the scene in front of the car. This makes scene/image captioning a very important problem to solve with high accuracy to design Self Driving cars.
- **Aid to blind** : Image captioning can help visually impaired individuals understand the scene in front of their eyes by converting the text generated from image captioning into speech, which can be then be directed to an earphone. This can help a visually impaired individual navigate in unknown areas and virtually be their eyes at all times. Popular applications include, Google Lens and Air
- **Security** : Captioning scenes detected by CCTV cameras can help identify criminal activities and raise an

[1] https://github.com/TuhinKundu/image-captioning

alert to the police when a burglar or miscreant is identified on the camera.

- **Search** : Image captioning helps in Content-Based Image Retrieval(CBIR) to index images for search engines. This use case has variety of application in popular platforms such as Social media such as Facebook, Instagram as well as search engines like Bing, Google Search, and image galleries in Android or IOS phones, etc.

1.3. Contribution of this project

The image captioning model we use is inspired by [20]. Our contribution to this project includes the empirical analysis between three datasets and using the encoder decoder model with various types of initialised decoder weights such as randomized, Glove embeddings and the BERT model [4]. We search for performance gains with contextual word embedding based decoder weights initialization over generic word embedding initialization.

2. Related Work

Image captioning has been approached in two ways broadly [7], 1) Traditional machine learning techniques 2) Deep Learning techniques. Traditional machine learning approaches include feature extraction methods such as Local Binary Patterns (LBP), Scale-Invariant Feature Transform (SIFT), the Histogram of Oriented Gradients (HOG) and various combinations of features among these. These features are extracted from the input images and then classified using Support Vector Machine(SVM).

Deep learning is the better way to approach the problem and uses a combination of convolutional and recurrent neural networks to generate the captions from images. One of the popular approaches in this is the model by Xu et al. [20]. It uses CNN to build a dense representation of image and layers of LSTM for sequence modelling. Another paper by Karpathy et al (NeuralTalk) [10], have achieved benchmark results for Flickr8k, Flickr30k and MS COCO dataset in 2015, using alignment model with CNNs and bidirectional RNNs.

Hossain et al.[7] explains an overall taxonomy of deep learning-based image captioning with the help of the following diagram.

As can be seen in Figure 1 there are two kinds of architecture used to approach the image captioning problem. We are using an Encoder-Decoder based architecture.

In an encoder decoder architecture, hidden activations of a Convvolution Neural Network is used to extract global image features. These features are then fed to an LSTM model to generate a sequence of words.

A typical method of this category has the following general steps: [7] (1) In the first step, to identify the context of the scene, different objects constituting the scene and the relationship between then, a vanilla CNN is used. (2) The output from the previous step is then given to a language model to convert the information into words and combined phrases to generate image captions.

The other model as opposed to the encoder-decoder architecture we have used is called a compositional architecture.

In this architecture, the model is built from several independent functional building blocks. The input image is fed to a CNN to extract the semantic concepts from the image. The output from this layer is then fed to a language model, that generates a number of captions for the same image. After this to select the most suitable caption among these captions, these candidate captions are *re-ranked using a deep multi-modal similarity model.*

More recent approaches to image captioning include a novel reinforcement based image captioning model [15] based on an actor critic reinforcement learning model to train the model [11], another reinforcement learning based model using test time inference algorithm [16] while [22] proposed another actor-critic based image captioning model. Recently, Generative Adversarial Network (GAN) based approaches to image captioning have been also proposed [3, 17]. Widely popular but slightly older image captioning models also include DenseCap [9], where salient regional proposals are generated for every image and captions are generated for those proposals, and Neural Image Caption Generator [19], which uses a CNN for the encoder and a LSTM based decoder network to train the model.

The report has been split into the following sections: section 3 describes the datasets used in the experiments, section 4 describes the encoder and decoder models we use for training our image captioning model, section 5 talks about the experimental setup, section 6 includes results and conclusion and section 7 round up with the future scope of the project.

3. Datasets

The Flickr8k [6], Flickr30k [21] and COCO-2014 [12] are widely used datasets for a variety of computer vision tasks and also are widely benchmarked datasets for image captioning related problems. We also use these three datasets for our experiments. In the following section, we individually describe these three datasets.

3.1. Flickr8k

Flickr8k dataset includes images from Flickr website. There are different objects in all the images, and contain no famous person or place. There are approximately 8000 training images, 1000 images in validation set and 1000 testing images.

Figure 1. An overall taxonomy of deep learning-based image captioning[7]

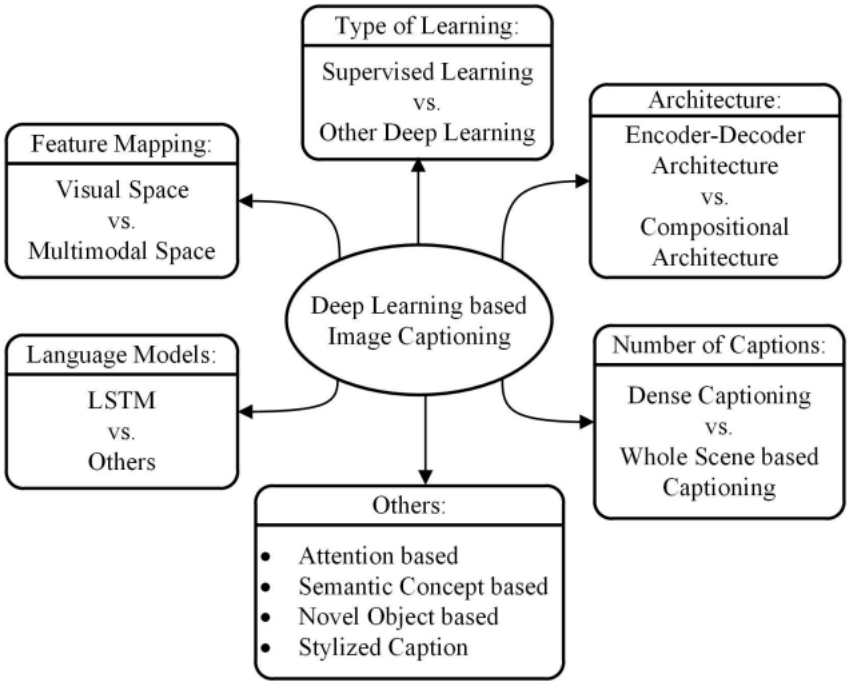


Figure 2. A block diagram of simple Encoder-Decoder architecture-based image captioning[7]

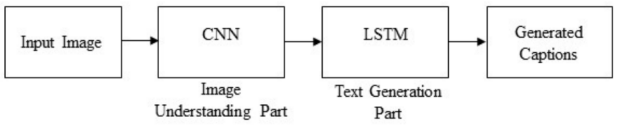
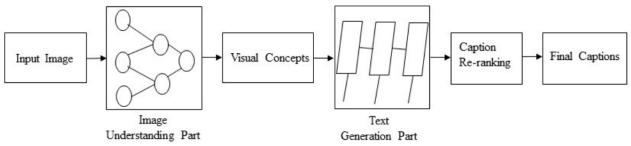


Figure 3. A block diagram of a compositional network-based captioning[7]



3.2. Flickr30k

The Flickr30k dataset is an extension to the Flickr8k dataset with 30000 images available for training and 1000 testing and validation images. The captions in this dataset have been created to maintain variance with 5 captions per image available in the annotations present in the dataset. The images also contain multiple objects, leading to the possibility of captions containing multiple subjects.

3.3. COCO-2014

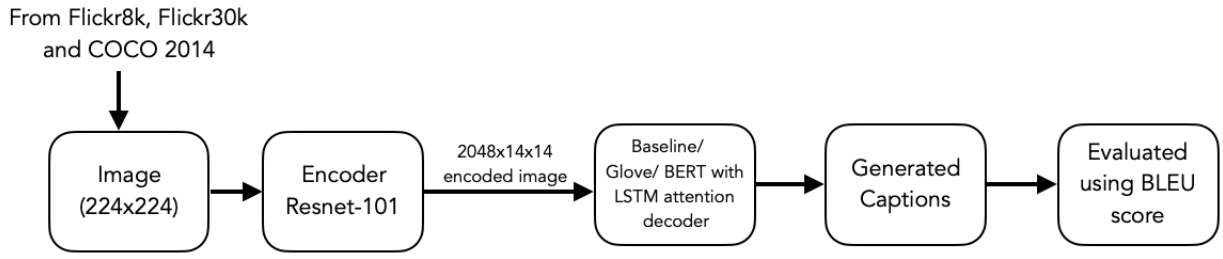
The original COCO 2014 dataset was created by Microsoft with 82000 training images approximately and 5000 testing and validation images. The images usually contain a primary object with background subjects present which is very useful to train models for image retrieval tasks. The dataset has been created for multiple tasks such as object detection, image segmentation, image captioning and image classification and is a widely benchmarked dataset for the aforementioned tasks.

4. Models

The model used for image captioning primarily consists of an encoder decoder architecture. The encoder here includes a CNN which takes an image as an input to extract features or useful information to train the model. These extracted features from the CNN are passed onto a decoder network which is composed of a recurrent network which trains on sequential information, which are the annotations for each image present in the dataset.

For the decoder RNN, we run experiments for three types initialised weights which is shared throughout the network to train using the sequentially data. We use a LSTM-attention based RNN for our decoder to predict the word in the next time step, and the three types of embeddings include randomized, Glove word vector representations and the BERT model, which is a type of transformer based

Figure 4. A block diagram of our encoder-decoder model for image captioning



contextual word embeddings. We use attention on top our LSTM network to focus over specific parts of the information flow and to better manage long term dependencies in the captions. We initialise the decoder RNN with these randomized or embeddings based weights and perform an empirical analysis to check for performance gains. A visual description of our model is shown in Figure 3.

In this section we describe in detail our encoder and decoder models that we used to perform the experiments for our image captioning model.

4.1. Encoder: Resnet-101

The encoder of the model comprises of a CNN similar to the one used in [20]. We could train our own CNN network, but we choose to focus more on the decoder portion of our model and training a CNN on big datasets usually is an extremely computationally expensive operation. So we move ahead and use one of the popular pre-trained models trained on Imagenet datasets, Resnet-101 [5].

Resnet-101 takes in an input image of dimension 224×224 and we discarded the final two layers of the architecture. The original model was used for object detection or classification tasks, hence includes a softmax layer for getting the resultant classes. Given we wish to use the Resnet-101 model to only extract features for our decoder to use, we remove the last two layers which include pooling and softmax layers. Removing these two layers, we obtain a encoded image of dimension $2048 \times 14 \times 14$ by performing adaptive pooling, which we pass onto our decoder model. We do not perform any fine tuning for our encoder model.

4.2. Decoder

4.2.1 Baseline - LSTM-attention model

We use a LSTM based RNN for our decoder unit to predict the captions at every time step. The implementation of this decoder module is based on the one given in [20]. The model unit predicts the word in the next time step on the basis of the hidden state, context vector and previously generated words. We also use soft attention as in [20] on top of our LSTM network to focus on specific parts while training or testing our network and map contextual information that may be present between image components and specific word representations.

$$\begin{bmatrix} i \\ f \\ o \\ g \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} .T_{D+m+n,n} \cdot \begin{bmatrix} E_{y-1} \\ h_{t-1} \\ z_{t-1} \end{bmatrix} \tag{1}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh c_t$$

In equation 1, i, f, o, g, h are the input, forget, memory, output and hidden states of the LSTM unit in our model, with c being the cell state. At timestep t, h_t is the hidden state with $T_{n,m}$ defining the affine trainsformation from dimension n to dimension m.

z_t in equation 1 denotes the context vectors of the relevant portions of the image at a particular timestep and E is the embedding matrix used in the recurrent network. The embedding matrix used in this decoder model is randomly sampled from a Gaussian distribution. So we are basically training the decoder module from scratch in this setup. In the Glove and BERT based models, we replace this randomized embedding matrix with word vector representations obtained from Glove and BERT models.

4.2.2 Glove - LSTM-attention model

Glove embeddings [14] are a type word vector representations obtained after training a model to incorporate global statistics and local contextual information and provides a word vector representation for every word with a fixed dimensionality. Glove embeddings were originally trained using a global log-bilinear regression model. These embeddings are useful for our experiments, as these embeddings already contain semantic or syntactic meaning for every word.

We use the Glove embeddings on the vocabulary of the annotations/captions of the images present in our dataset.

We initialise the embedding matrix of the Baseline-LSTM-attention model and use the decoder setup to train the model as in the previous setup. The difference particularly present here is that we fine tune our Glove representation based embedding matrix and we start our decoder training with some meaningful representation of the words.

We use Glove vectors of 200 dimensionality trained on Twitter dataset with 27 billion possible tokens.

4.2.3 BERT - LSTM-attention model

In the Glove embeddings used, each word had a specific and definite word vector representation of fixed dimensionality. While Glove had been sucessfully used in various tasks, a word may have different meaning while being used in a sentence. Hence the BERT model [4] is a bidirectional transformer based contextual word representation trained using masked language modelling, can be used to capture and encode contextual word information that may be present in a sentence. Masked language modelling is basically masking specific words in a sentence and making the transformer based recurrent network architecture try and predict those very masked words. For the BERT representation, the entire sentence needs to be tokenized together and sent to the pre-trained transformer based architecture along with signifying the [CLS] and [SEP] tokens to signify the starting and separation of sentences and obtain a representation of every word in that sentence of a fixed dimensionality, but with contextual information encoded. The BERT model itself was a breakthrough in language modelling and was devised in such a way, that it could be fine tuned to obtain state of the art performance on various NLP tasks such as question answering, language inference, machine reading comprehension, named entity recognition amongst various others.

BERT pre-trained models that have been released are of several types. The BERT pre-trained model we use for our experiments is the BERT-base-uncased version, which basically is the smaller model trained only on lowercase English language words. This model generates a word representation dimension of 768 for every word present in the sentence. Using all the words present in our annotation/caption vocabulary of a particular dataset, we form our embedding matrix using the BERT based contextual word representation to initialise our decoder weights in our decoder unit.

5. Experiments

We use all the three datasets, Flickr8k, Flickr30k and COCO 2014 dataset for our experiments using the encoder-decoder setup to train the image captioning model. We use all 8000 and 30000 images for training of the Flickr8k and Flickr30k datasets respectively, and performed testing experiments on their entire test sets of 1000 images each. Of the 82000 training images present in the COCO 2014 dataset, we use only 20000 images due to computational constraints in memory and GPU to train the models. We however test on the entire COCO 2014 testing set of 5000 images.

We resize all images we use for training and testing down to size 224×224 pixels, as its the default size used for the pre-trained Resnet-101 encoder model. The hyper parameters for the training the encoder-decoder unit include the baseline model having a randomized vector representation of 512, Glove representation of dimension 200, BERT model representation of dimension 768, the model being trained for 5 epochs (due to computational resource constraint), learning rate of the decoder unit set at 0.0004, dimensionality of encoder output tensor of size $2048 \times 14 \times 14$ and the attention head tensor of dimension 512. We use cross entropy loss for all our experiments.

5.1. Evaluation metric: BLEU

BLEU (Bilingual Evaluation Understudy) [13] is a widely used evaluation metric for machine generated textual information and to judge how close they are to the human readable form. BLEU is a quality metric for machine translation systems which tries to find the correspondence between a machine translation and human correspondence. However, BLEU focuses on strings generated rather than attempt to evaluate overall translation quality. We use BLEU-1, BLEU-2, BLEU-3 and BLEU-4 scores to evaluate model performance. BLEU-4 is basically a 4-gram BLEU score.

6. Results and Conclusion

Figure 5. Generated and annotation (reference) captions of the image from COCO 2014 dataset using our baseline decoder model.
Generated: woman sitting restaurant and of on table has on with food
Reference: woman in black tank top sits at table that is covered with dishes of food

Results of the experiments run have been showcased in table 1, 2 and 3 for the three datasets, Flickr8k, Flickr30k and COCO 2014 dataset respectively. We could see that the BLEU scores for our baseline LSTM model and the model with Glove initialised embeddings have similar scores in all three datasets, hence Glove representations do not directly translate into any performance related gain in comparison

Dataset	Encoder weights	Decoder weights	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Flickr8k	Resnet-101	LSTM	0.335	0.081	0.027	0.010
		Glove	0.332	0.083	0.027	0.009
		BERT	0.474	0.183	0.081	0.033

Table 1. Results for Flickr8k with Resnet-101 encoder

Dataset	Encoder weights	Decoder weights	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Flickr30k	Resnet-101	LSTM	0.430	0.128	0.043	0.018
		Glove	0.430	0.123	0.042	0.017
		BERT	0.674	0.415	0.269	0.174

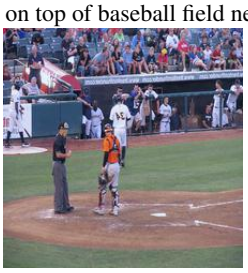
Table 2. Results for Flickr30k with Resnet-101 encoder

Dataset	Encoder weights	Decoder weights	BLEU-1	BLEU-2	BLEU-3	BLEU-4
COCO 2014	Resnet-101	LSTM	0.381	0.110	0.034	0.011
		Glove	0.385	0.114	0.036	0.010
		BERT	0.580	0.309	0.168	0.096

Table 3. Results for COCO-2014 with Resnet-101 encoder

Figure 6. Generated and annotation (reference) captions of the image from COCO 2014 dataset using our Glove based decoder model.
Generated: dogs are field in in water
Reference: two dogs on leash playing on the shore and in the water of lake

Figure 7. Generated and annotation (reference) captions of the image from COCO 2014 dataset using our BERT based decoder model.
Generated: standing on of baseball field next baseball
Reference: standing on top of baseball field next to an umpire



to randomized vector representations for words.

In comparison, BERT weight initialisation gives a significant performance gain over the randomized and Glove initialised embedding matrices. BERT based decoders units have performed the best across all the datasets and have significant gains in terms of BLEU scores over the other models. Hence we could make the assumption that contextual word information encoded in BERT vector representations provide significant gains in our encoder-decoder based image captioning model.

In figures 5,6 and 7, we see that the captions generated by the image captioning models on images from COCO 2014 dataset using baseline, Glove and BERT based decoder units respectively. We notice that, even though the captions generated are not perfect in terms of human understanding, the model maps the image to specific entities in the image and generates captions with objects that are actually present in the images. And this model has only been trained for 5 epochs which is less if compared to existing state of the art image captioning models. Also, its been trained on 20000 out of the 82000 images present in the COCO 2014 dataset. The model would probably generate better captions had it been trained for more epochs or trained on the entire training set.

Also while performing our experiments, we came to the conclusion that the training time of our encoder-decoder based image captioning models were training (and the loss converging) much faster than the time reported by various other CNN and RNN based models. We assume that the Glove and BERT based decoder weight representations and the Resnet-101 pre-trained weights gave a significant starting point for the image captioning to train on, as training randomized weight representations from scratch compromise on time efficiency of the models.

7. Future scope

Future scope of this project remains vast. Resnet-101 is not a recently published imagenet model and there are other various pre-trained models that can be used to boost encoder side performance of the model such as DenseNet [8] and Inception v4 [18]. The decoder weight initialization of the model could be replace with newer contextualized embedding models such as XLM-R [2] and ELECTRA

[1] that have recently outperformed BERT on various NLP tasks. Better computational resources could be used to train the encoder-decoder model for more than 5 epochs, as the BERT model can be seen performing very well in training using such a low number of epochs. Training for more number of epochs with better encoder and decoder weight initializations may result in the image captioning model outperforming various current state of the art techniques.

References

[1] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

[2] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

[3] Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. Towards diverse and natural image descriptions via a conditional gan. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2970–2979, 2017.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

[7] MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CSUR)*, 51(6):1–36, 2019.

[8] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[9] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4565–4574, 2016.

[10] Andrej Karpathy and Li 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.

[11] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing 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.

[14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[15] Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. Deep reinforcement learning-based image captioning with embedding reward. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 290–298, 2017.

[16] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017.

[17] Rakshith Shetty, Marcus Rohrbach, Lisa Anne Hendricks, Mario Fritz, and Bernt Schiele. Speaking the same language: Matching machine to human captions by adversarial training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4135–4144, 2017.

[18] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[19] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru 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.

[20] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[21] Peter Young, Alice Lai, Micah Hodosh, and Julia 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.

[22] Li Zhang, Flood Sung, Feng Liu, Tao Xiang, Shaogang Gong, Yongxin Yang, and Timothy M Hospedales. Actor-critic sequence training for image captioning. *arXiv preprint arXiv:1706.09601*, 2017.