Image Captioning using Attention Models and BERT Embeddings

First Author
University of Massachusetts, Amherst
firstauthor@umass.edu

Second Author
University of Massachusetts, Amhesrt
secondauthor@umass.edu

Abstract

We present a template that produces explanations of the activities or the objects of the image in the natural language. Our approach uses image data-set and word definitions to learn about the inter-model relationship between language and visual information. Our model of alignment is based on a novel combination of Convolution Neural Networks over image regions, Recurrent Neural Networks over sentences, and a structured goal that aligns the two modalities with an embedded BERT. We will demonstrate that our model generates state-of-the-art results in MS-COCO data-set retrieval experiments. Then we show that the descriptions produced significantly outperform the baselines of retrieval on complete images.

1. Introduction

The automatic generation of image captions an essential task required to understand images captured — primary task of computer vision technology. Caption generation models have to be sophisticated enough to solve challenges of image classification, identification and also articulate their relationships in natural language. Thus, caption generation has been viewed as a difficult problem for a long time. The sample image output is shown in Figure 1.

Although being a very challenging task, the increase in research to address this issue aided by advances in training Neural Networks and large classification data-sets led to several research papers being published. The quality of the production of captions has significantly increased in recent works using a combination of Convolutional neural networks to get vector representation of pictures and recurrent neural networks to decode them for natural language phrases. Image captioning can be broken down into two major tasks: (1) Correctly classifying the object detected in computer vision, and (2) Create a language template to correctly create a word describing the objects identified. Figure 2 shows the simple methodology for this model.

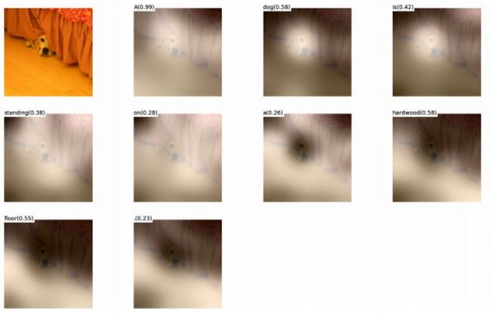

Figure 1. Attention over time. As the model generates each word, its attention changes to reflect the relevant parts of the image [9]

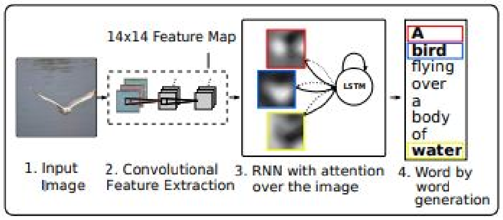
14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Figure 2. Methodology/ Architecture [9]

The baseline model for this project a model which uses CNNs for image classification and LSTMs for captioning of the images, and encoder-decoder models with attention in the task of image captioning. As an extension of this project, we have implemented the pre-trained BERT - embeddings on the task of image captioning by integrating BERT context vectors to enhance the models performance and accuracy[3]. Figure 3 shows the architecture of the transformer used for attention model.

The contributions of this paper includes the following:

- A pyTorch implementation of the soft attention model with an encoder-decoder architecture for improved image captioning.
- Enhance the performance of the implemented soft at-

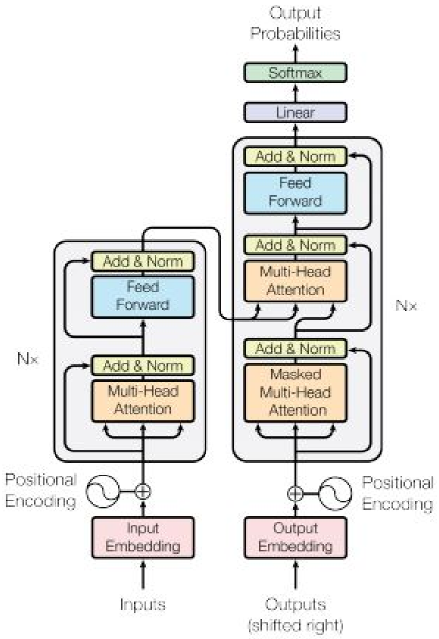

Figure 3. Methodology/ The Transformer - Modal Architecture [7]

tention model by integrating BERT context vectors as an extension to the project.

- To show that our BERT integration with soft attention model outperforms the baseline model, the results are visualized and quantitatively validated the models with MS COCO validation data-set.

2. Background/Related Work

[5] was the earliest to use deep neural networks for image captioning with multimodal log-bilinear model biased by image features. A similar approch by [6] to generate captions used a recurrent model inplace of feed-forward neaural language model. Later, methods to use LSTM RNN's evolved [8] and [4]. Also, methods to apply LSTM's for video description generation was developed and explored [4].

Karpathy Li developed methods for training a joint ranking and generation embedding space whose template learns to score sentence and image similarity as a function of R-CNN object detection with bidirectional RNN outputs rather than representing objects from a pre-trained convolution network as a single top-level feature vector.

Our project is based on [2], [1] and [9]. And the Implementation of BERT-embeddings on top of this model[9]

3. Approach

3.1. Model Details

To create captions, we use an encoder-decoder architecture. The encoder is a Convolution Neural Network (CNN) which takes a single image and generates a matrix which defines the objects observed. This object vector is then passed to the decoder, which is a Long Short-Term Memory Network (LSTM) that takes care of the image and produces a descriptive one-word caption at each step.

We define in this section the encoder used in all our models and two versions of [9] attention-based decoder. The first decoder is an exact replica of the model of soft attention described in [9]. This initial model is going to act as our foundation. The second decoder is a baseline model extension that integrates the pre-trained context vectors of BERT into the captions to improve the performance and accuracy of the model.

3.1.1 Problem Statement

Given a single raw image, our goal is to generate a caption **y** encoded as a one-hot vector corresponding to our vocabulary.

$$\mathbf{y} - \{y_1, y_2, ...y_c\}, y_i \epsilon R^V$$

where V is the size of the vocabulary and C is the length of the caption.

3.1.2 ENCODER: CONVOLUTIONAL FEATURES

Similar to the encoder described in [9], we use a CNN to extract feature vectors from the images. The Encoder provides L vectors, where each vector has D-dimensions that represent part of the image.

$$\mathbf{a} - \{a_1, a_2, ...y_L\}, a_i \epsilon R^D$$

The pretrained ResNet-101 CNN is used as a encoder to reduce our training time and focus on enhancing the performance of the decoder. In ResNet-101, the last two layers - pooling layer and linear layer are discarded as we only need the image encoding, rather than image classification. The output of the modified ResNet is passed onto an adaptive pooling layer to create a fixed size output vector - fixed L- that can be easily passed to the decoder.

3.1.3 DECODER: LONG SHORT-TERM MEMORY NETWORK

A long-term memory network is used to produce caption words one step at a time by conditioning on the hidden state of the previous step, the context vector, and the words previously generated. Implementation is based on the model described in [9]

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} \mathbf{Ey}_{t-1} \\ h_{t-1} \\ z_t \end{pmatrix}$$

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

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

Here,$\mathrm{i_t}$, $\mathrm{f_t}$, $\mathrm{o_t}$ and $\mathrm{g_t}$ are the input, forget, memory, and output states of the LSTM. h is the hidden state,c is the cell state (that keeps long term memory), $\mathrm{h_t}$ denotes the hidden state at timestamp t. Additionally, $\mathrm{T_{n,m}}$ defines an affine transformation from dimension n to m. $\odot$ is an element-wise multiplication.

E represents an embedding matrix that is used and z_t denotes the context vectors of the relevant part of the image at time step t that is generated through soft-attention. To generate z_t using ”soft” attention, we implement the soft-attention mechanism detailed at [9]. In this model, the caption embedding, **E**, is learned alongside training the model to generate captions.

In the next subsection, the method for optimizing **E** and enhancing the models performance are described.

3.1.4 DECODER : BERT Attention Model

Prior to BERT Attention Model, GloVe vector representations were used in which each word is represented by a single unique vector no matter the context of the word used in. This raised concerns for several researchers[3] as they realized that each word would have multiple meanings depending on the usage of the word. rather having one representation for each word, BERT uses a Transformer to generate a bi-directional contextualized word embeddings conditioned on the context of the word in a sentence.

The BERT-base and the BERT-large are the two models of BERT. The base version has 12 layers of encoders in Transformer, 768 hidden units in the feed forward-network, and 12 heads of attention. On the other hand, the large version has 24 layers of encoders in its Transformer, 1024 hidden units the forward-network feed of in it, and 16 heads of attention.We use BERT base in our implementation to produce the contextualized word vectors of the caption due to the increased training time provided by the broad template.

In the decoder, we take a caption as our input , where c_i is a full text representation of the caption. Then we iterate through each caption c_i and perform the following steps:

1. Tokenize each caption with the word tokenizer of BERT to allow BERT to digest the caption and add the special '[CLS]' BERT token at the start of the caption, and the '[SEP]' BERT token at the end of the caption.

2. The segment ids for each caption is obtained, it contains 1's of length equal to the number of words in the caption since we have only one sentence in each caption.

3. Pass the tokenized captions and the segment ids for each caption into the BERT base model. The segment ids for each sentence contains 1's for a length equal to the number of words in the caption since we have only one sentence in each caption.

4. Retrieve the 12th layer output tensor of the Bert Model(the output of BERT model at the last time step).

By following the above steps for each caption in the batch, caption embeddings $\mathrm{b_i}$ is obtained which is a tensor of size(1 x caption size x 768) as each word in the caption has a vector size of 768 as its contexualized embeddings. The $\mathrm{b_i}$ embeddings can then directly replace trained embeddings used in the attention model mentioned above.

4. Experiment

This section begins with what kind of experiments you're doing, what kind of dataset(s) you're using, and what is the way you measure or evaluate your results. It then shows in details the results of your experiments. By details, we mean both quantitative evaluations (show numbers, figures, tables, etc) as well as qualitative results (show images, example results, etc).

We describe our experimental methodology and quantitative results which validate the effectiveness of our model for caption generation.

4.1. Data

There are several easily accessible data sets for image captioning tasks such as MS COCO, Flickr8k, and Flickr30k to train and test models. In this paper, we use the MS-COCO 2014 dataset for both training and validation. After re-sizing and normalizing all the images to 256 x 256 pixels, captions are extracted and tokenized with the NLTK tokenizer.

4.2. Caption

Captions are both the target and the inputs of the Decoder as each word is used to generate the next word. However, the first word is a common word i.e, zeroth word, ¡start¿ in Attention model and '[CLS]'. At the end we should predict ¡end¿ in Attention model and ¡sep¿ in BERT model. the decoder must learn to predict the end of a caption.

Since we pass the captions as fixed size Tensors, we need to pad captions to the same length with ¡pad¿ tokens. From this a $word_map$ is created which is an index mapping for each word in the corpus, including the 'start', 'end', 'CLS', 'SEP', and 'pad' tokens. Therefore, captions fed to the model must be an Int tensor of dimension N,L where L is the padded length.

4.3. Training Procedure

The attention model and the BERT model were trained with stochastic gradient descent using adaptive learning rate algorithms. We used MS COCO data-set for training for which Adam algorithm worked best. The optimized hyper-parameters used were as follows: (1) gradient clip = 5(to avoid gradient explosion), (2) number of epochs = 3, (3) batch size = 1, (4) decoder laerning rate = 0.0004, (5) dropout = 0.5, (6) encoder dimension = 2048(based on nRESNET - 101's output size), (7) attention dimension = 512, and (9) all the weights were initialized using a uniform distribution with range = [-0.1, 0.1].

While implementing the embedding extensions, embedding dimension of 512 was used for the attention model and 768 for BERT model. All models are trained and validated with the same vocabulary on the same dataset splits to allow an accurate comparison of results. Each epoch for Attention model took around 5 hours to train on a m40 GPU, while the BERT model's epoch took around 7 hours.

To evaluate the model's performance on the validation set, we used the automated Bilingual Evaluation Understudy(BLEU) evaluation metric. This evaluates a generated caption against reference caption(s). For each generated caption, we will use all N_c captions available for that image as the reference captions.

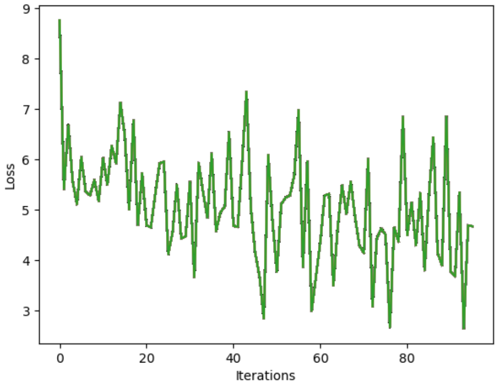

Figure 4. Validation Loss for Attention model

The validation loss for attention model is shown in Figure 5. The code for the graph was implemented later on after milestone feedback was given. Thus, we did not have enough time to train the model for large number of iterations. From the graph obtained, we observe that the loss is decreasing and eventually it would converge to a minimum point.

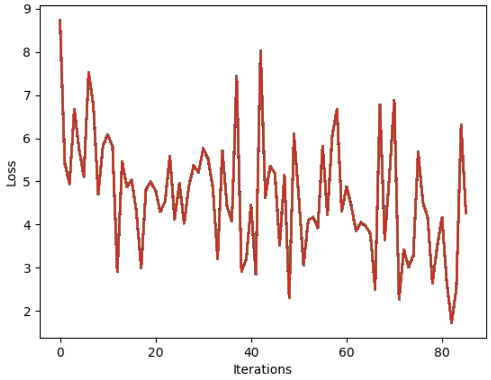

Figure 5. Validation Loss for Bert model

The validation loss for BERT model is shown in Figure 6. Same issue was faced even for this graph. From the graph obtained, we observe that the loss is decreasing and eventually it would converge to a minimum point.

4.4. Qualitative Analysis

Below figures shows the outputs for both the attention models and BERT models. Qualitatively analyzing the results, we see that the BERT model output image has an accurate, grammatically correct hypotheses than the Attention model.

4.4.1 Success Scenarios

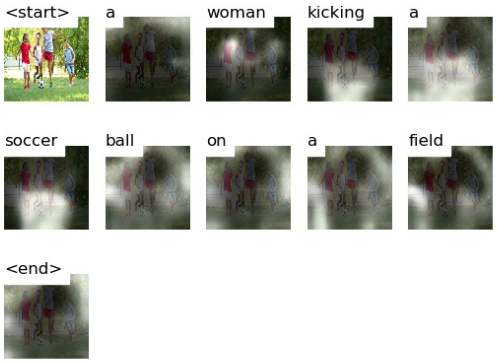

Figure 6. Attention Model Caption - Image 1

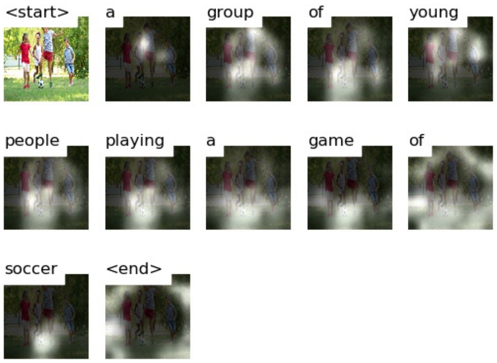

Figure 7. BERT Model Caption - Image 1

From Figure 6 and Figure 7, we see that for an image of children playing soccer, we get two different captions. For Attention model, the output produced is that women kicking a soccer ball whereas for BERT model it captions correctly stating that group of young people are playing soccer.

Two people riding a jet ski was given as input. From Figure 8 and Figure 9, we see that attention model got bad results stating that it was a group of people riding a boat whereas the BERT Model predicted accurately that 2 people were riding a boat.

From Figure 10 and Figure 11, a Chef preparing for meal is predicted as a man preparing food wheres in the BERT model, keywords ’chef’ and ’meal’ is produced.

From Figure 12 and Figure 13, a man riding a tractor is given as a input image. Both the predictions are almost same for this test case although small variations can be seen

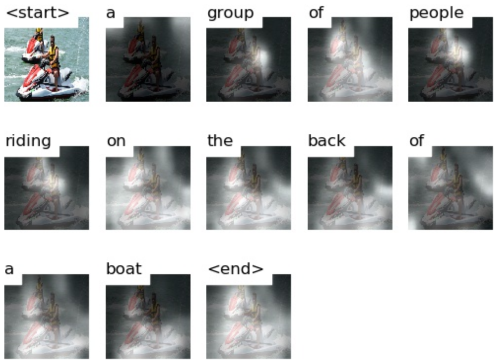

Figure 8. Attention Model Caption - Image 2

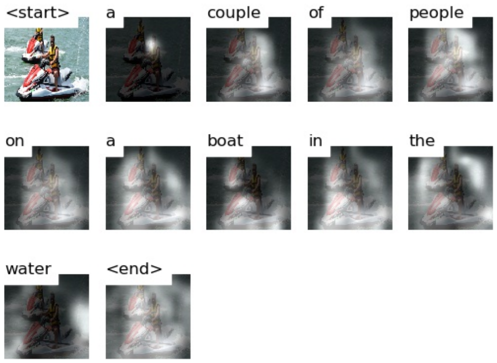

Figure 9. BERT Model Caption - Image 2

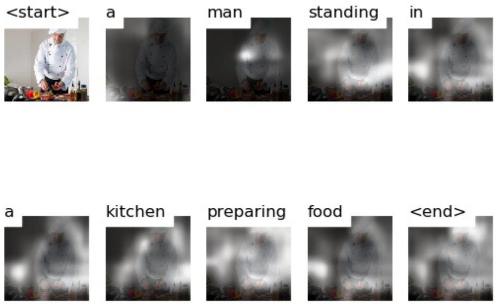

Figure 10. Attention Model Caption - Image 3

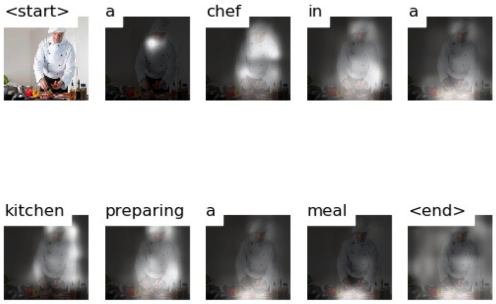

Figure 11. BERT Model Caption - Image 3

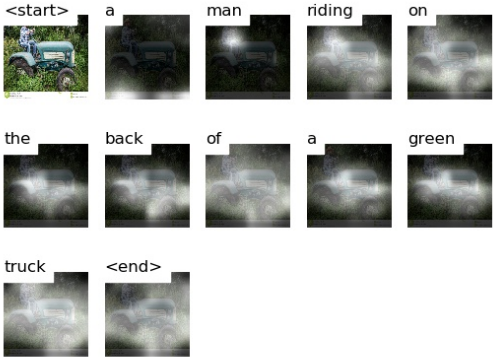

Figure 12. Attention Model Caption - Image 4

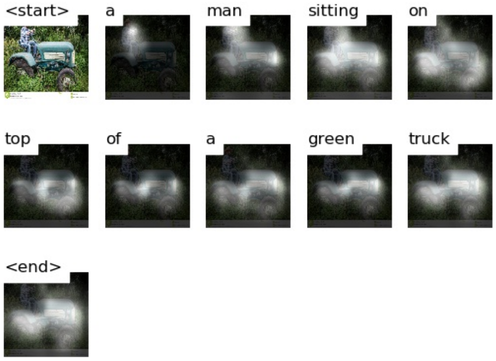

Figure 13. BERT Model Caption - Image 4

in the produced captions.

4.4.2 Failure Scenarios

In the above section, Success scenarios were shown. Here are few examples where the Bert model did not give better predictions than the attention model.

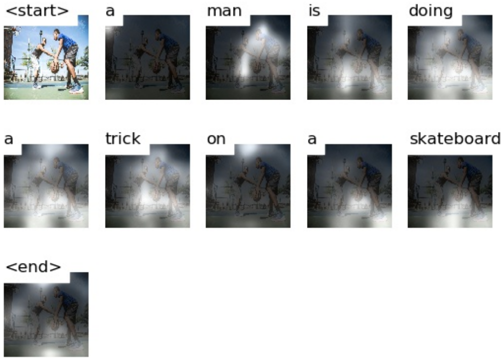

Figure 14. Attention Model Caption - Image 5

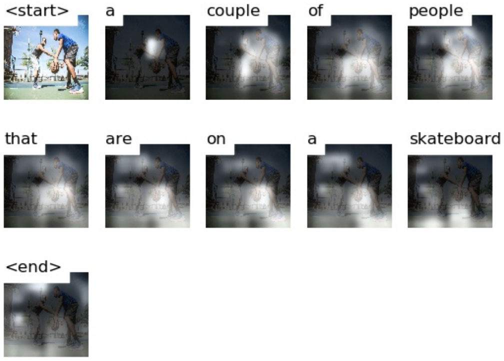

Figure 15. BERT Model Caption - Image 5

For Figures 14 and 15, two people playing basket ball was wrongly captioned as playing skateboards. Although wrong caption predictions are made, we see that the structure of the sentence of the two models are quite different.

For Figure 16 and Figure 17, an image of a Cheetah chasing a deer was given as a input. While Attention model predicted it was giraffe, the Bert model predicted slightly better as animals.

4.5. Quantitative Results

- BLEU-4 scores for MS-COCO Data-set.

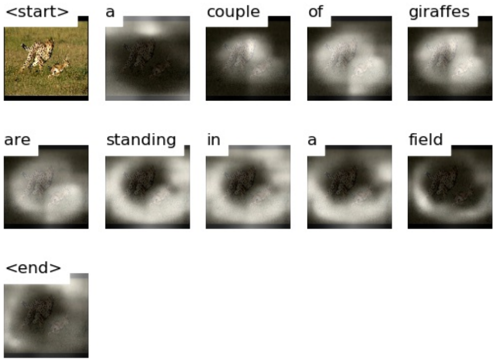

Figure 16. Attention Model Caption - Image 6

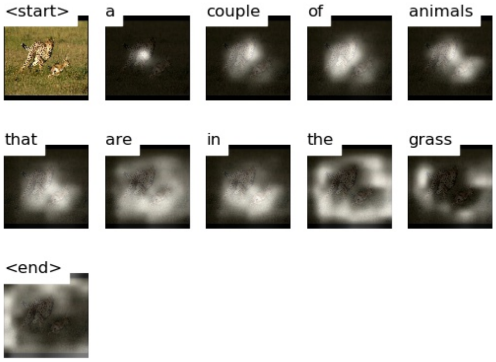

Figure 17. BERT Model Caption - Image 6

BLEU-4 Scores	
Attention Model	BERT + Attention Model
21.1	26.8

Table 1. BLEU-4 Scores

- Validation Loss for MS-COCO Data-set.

Validation Loss	
Attention Model	BERT + Attention Model
4.5	2.7

Table 2. Validation Loss

5. Conclusion

We propose addition to the baseline model of CNN and LSTM approach of image captioning using attention models that provides state of the art performance on the MS COCO dataset using BLEU metric. The BERT approach surpasses the MS COCO validation scores of the Attention model while being trained on fewer epochs with the same hyper-parameters. Our experiments outline the importance of word embedding in the processing of the language of nature and contextualizing word meanings, and also offer a new way of integrating BERT with already developed models to improve their performance.

Future ideas would include to train a new model with BERT large as apposed to the BERT base which was used here. A good idea will be to use image data-set with occlusions with a BERT-large embedding and check the performance. Although we could not train the model as we would have liked to, for large enough epochs and better hyper-parameter tuning given the heavy time constraints and a limited GPU access, the learning included in-detail understanding of the encoder-decoder architecture, attention models and the BERT embedding technique used in natural language processing. Other ideas would be to utilize beam search validation, train the models until the training loss converges, i.e, for a higher number of epochs.

References

[1] J. Ba, V. Mnih, and K. Kavukcuoglu. Multiple object recognition with visual attention, 2014.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate, 2014.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[4] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *CoRR*, abs/1411.4389, 2014.

[5] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–595–II–603. JMLR.org, 2014.

[6] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn), 2014.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.

[8] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator, 2014.

[9] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention, 2015.