Image Description Generation with Disentangled Language and Description Models
Individual Project AI

Victor Milewski, 10529136[1] and Iacer Calixto[2]

[1] *Master Artificial Intelligence, University of Amsterdam, Science Park 904, Amsterdam, The Netherlands*

[2] *Institute for Logic, Language and Computation, University of Amsterdam, Science Park 107, Amsterdam, The Netherlands*

victor.milewski@student.uva.nl, iacer.calixto@uva.nl

Keywords: Image Captioning, Disentanglement, Language Model, Description Model, Flick8k, Flickr30k

Abstract: In this work, an attempt has been made to disentangle the two tasks of the decoder part of a caption generator: creating a grammaticaly correct sentence and describing an image. This is achieved by implementing three sub-models: the Language Model for creating the grammatically correct sentences, the Description Model for selecting descriptive words about the image, and the Binary Switch Model for deciding at each time-step which of the former two is used. Multiple feature vectors were tested for each of the different sub-models.

1 Introduction

Two of the main fields in Artificial Intelligence are Computer Vision (CV) and Natural Language Processing (NLP). Both fields get a lot of attention due to success in varying applications and tasks. CV has made some big advances using Convolutional Neural Networks (CNNs) (LeCun et al., 1990). These perform well when applied on visual data like images and videos, for tasks such as classifications (Szegedy et al., 2016), object detection (Redmon et al., 2016), and tracking (Nam and Han, 2016). Within NLP, the use of Recurrent Neural Networks (RNNs) with special Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) cells, has led to big advancements in tasks like neural machine translation (Cho et al., 2014) and text summarisation (Rush et al., 2015). In the last few years, the addition of attention mechanisms has shown even further improvements (Bahdanau et al., 2014). Furthermore, there have been successful attempts where the RNN is entirely replaced by a special attention model (Vaswani et al., 2017).

More complex tasks arise when multiple fields, like CV and NLP, are brought together. One of these tasks is the caption generation for images (Vinyals et al., 2015; Xu et al., 2015). The goal is when given an image, to generate a text describing the visual information in the image. This is achieved by first using a CNN to encode the images into vectors. Using these vectors, a decoder LSTM-RNN is initialised for generating the sentences (Vinyals et al., 2015). As mentioned before, attention can be used to improve the results of such recurrent models (Xu et al., 2015). For the captioning model, the encoder has to generate a set of vectors for regions of the image. The attention mechanism can be used during decoding. In each time-step, the decoder pays attention to one or more of the image regions to predict the next token.

A downside to such approaches is the use of a single model for decoding. This model essentially has to learn two tasks, generating a syntactically correct sentence, and generating a good description of the image. In the first approach, the decoder is only initialised with the single vector describing the image, which makes creating good descriptions difficult. For the attention approach, the model has to focus at regions of the image for generating a token, even if visual representations of such tokens do not exist (like determiners, some verbs, etc.) Lu et al. (2017). In this study, there will be investigated if a disentanglement of the tasks into two sub-models by training a separate language and description model performs well. A third sub-model, the binary switch model, will be trained to determine at each time-step which of the models will be used for predicting the following token. The link to this project is https://github.com/victormilewski1994/disentangled-caption-generator.

2 Related Work

A lot of research has been conducted on the task of Caption Generation. According to You et al. (2016), there are two main approaches in Caption Generation, the top-down and the bottom-up approach. With bottom-up approaches, a set of words is selected and these are used to generate a sentence. This is close to how it was done in earlier stages, where the captions only existed of a set of tokens which are likely present in the picture (Pan et al., 2004). This selection of tokens can be done using algorithms like expectation maximisation. Nowadays, after extracting tokens from the image or image regions, a Language Model (LM) is used to combine a subset of these tokens into sentences (Fang et al., 2015).

However, since the rise of deep learning, most studies tend to use the top-down approach. Often, an encoder-decoder network is used, with a CNN encoder, and an LSTM-RNN decoder (Vinyals et al., 2015)(Devlin et al., 2015). Many additions have been created to improve these models. The most noteworthy is the addition of attention mechanisms (Bahdanau et al., 2014)(Xu et al., 2015). Separate vectors for image regions are created and the decoder is trained to focus on a different region at each time-step. It has been shown that the correctness of such an attention mechanism is correlated with the quality of the captions (Liu et al., 2017).

However, these models are still not optimal. In the last couple of years, it has been investigated whether the attention can be improved, extended, or replaced by other mechanisms. One of these methods involves using reinforcement learning (Ren et al., 2017)(Rennie et al., 2017). An adaption of the transformer from Vaswani et al. (2017) in combination with an RNN has also been used to make these attentions even more informative (Pedersoli et al., 2016).

Finally, there have been approaches that improve the attention by making the decoder only look at the image when needed, which is similar to this study. Lu et al. (2017) made the model more complex such that the attention is only used when needed, whereas in this study two simple models are created that operate side by side.

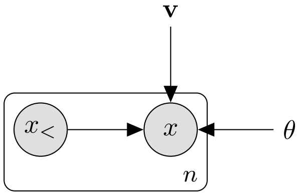

Figure 1: A representation of the fully supervised LSTM-RNN decoder model for image descriptions. Here x is the n'th token, $\mathbf{v}$ the image descriptor, $x_<$ the previously predicted tokens, and θ the collection of parameters.

3 Model

As a baseline the implementation from Vinyals et al. (2015) is used. This is an encoder-decoder model where an image v is encoded into a descriptor $\mathbf{v}$ using a CNN. Hereafter, it is decoded into a sequence x_1^n. The baseline decoder is an LM, which can be depicted as in Figure 1. The goal is to maximize $P_\theta(x_1^n|\mathbf{v})$ according to the Markov assumption as in:

$$\begin{aligned} P_\theta(x_1^n|\,\mathbf{v}) &= \prod_{i=1}^{n} P_\theta(x_i|\,\mathbf{v}, x_1^{i-1}) \\ &= \prod_{i=1}^{n} \mathrm{Cat}(x_i|f(\mathbf{v}, x_{<i}; \theta)). \end{aligned} \quad (1)$$

The categorical in (1) is a mapping over the vocabulary of the language. Function f is implemented as a softmax over an two-layer LSTM-RNN, which is initialised with the image descriptor.

3.1 Disentanglement

In the current work, a disentanglement in the decoder is proposed. An overview of the model and its sub-models is given in Figure 2. The disentangled sub-models will receive information about the image through topics. This topic modelling is described in Section 3.1.1.

The first disentanglement is the LM, where the main focus is to create grammatically correct sentences (see Section 3.1.3). The second is the Description Model (DM), which focusses on generating descriptive words about the image (see Section 3.1.4). At each time-step i from (1), there will be determined whether the LM or the DM will be used for function f. In Section 3.1.2 the Binary Switch Model (BM) is defined, which will make this decision.

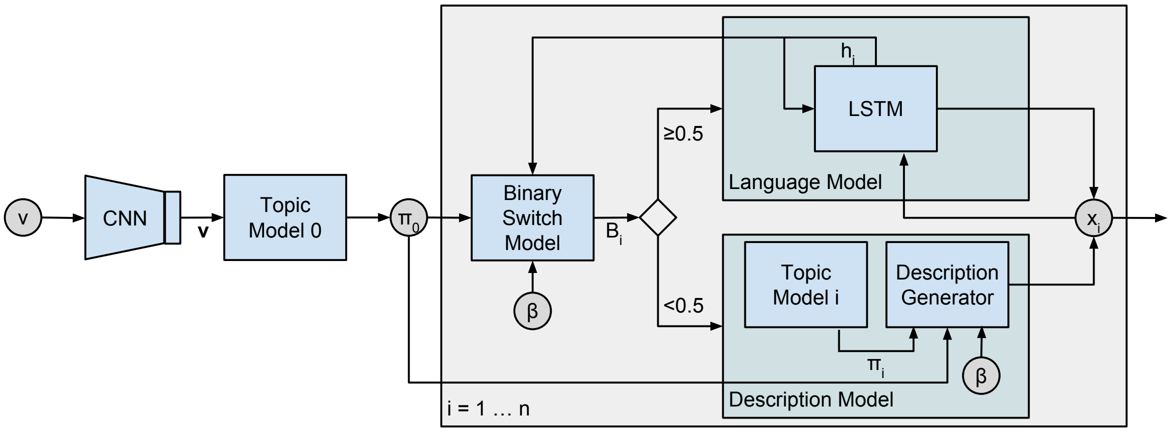

Figure 2: An overview of the basic flow for the disentanglement model.

3.1.1 Topic Modelling

The main features used for the sub-models will be obtained through topic modelling over the image. First, the topic embeddings are defined

$$\beta_k \in \mathbb{R}^d, \tag{2}$$

where $k \in < 1, \dots, K >$ with K the number of topics and d their dimensionality. These embeddings will be trained along with the model, so the topics will contain relevant information for the given dataset.

For each image, it must be determined which topics are present by computing a global topic distribution π_0 over the image descriptor, as in (3). Similarly, a positional topic distribution π_i can be computed at every time-step in a sequence

$$\pi_0 = \text{softmax}(\alpha_0(\mathbf{v})), \tag{3}$$

$$\pi_i = \text{softmax}(\alpha(\varphi)), \text{ with } i > 0. \tag{4}$$

where φ is the used feature vector, which will be described in Section 3.1.4. The two separate equations, (3) and (4), are defined, since for the zeroth time-step a global distribution for the image must be created that selects all the topics present, while at the other time-steps the model must select a subset from the topics present, needed to generate a single descriptive token.

Using the topic distribution and the embeddings, a topic embedding is computed as a weighted sum

$$z_i = \sum_{k=1}^{K} \pi_{ik}\beta_k. \text{ with } i \geq 0 \tag{5}$$

Finally, we create a mixing of past used topic embeddings $z_{1:t}$, to keep a memory of previously used topics in the DM

$$z_{1:t} = \sum_{k=1}^{K} \Big(\frac{\pi_{0k}}{t} \sum_{i=1}^{t} \pi_{ik}\beta_k\Big). \tag{6}$$

As can be seen, the mean of the π's is multiplied by the global topic distribution π_0. This ensures that there is no information present about topics that do not exist in the global topic distribution.

α_0 is implemented as a two-layer Multi-Layer Perceptron (MLP) with a Rectified Linear Unit (ReLU) after the first layer, whereas α is implemented as a four-layer MLP with a ReLU non-linearity after the first three layers.

3.1.2 Binary Switch Model

The BM is used to generate an output B_i for every time-step according to

$$B_i = \text{sigmoid}(s([z_0; x_{<i}; \chi])), \tag{7}$$

with $\chi \in \{\mathbf{v}, z_{1:t}\}$ and $0 \leq B_i \leq 1$. When $B_i \geq 0.5$, the LM as described in Section 3.1.3 is used, otherwise the DM from Section 3.1.4 is used.

The features used in function s in (7), are chosen so the switch has information about the predicted words so far and about which topics are represented in the image. With χ, the switch can be given additional information about the image or about the previously used topics. This can be beneficial since it might shift towards choosing the DM more when the topics lack in information, or less if all the topics are already used. In Section 4, it will be determined if these assumptions are true.

The function s from (7) is implemented using a two layer MLP with a ReLU non-linearity after the first layer.

3.1.3 Language Model

The LM is similar to function f in (1), and just as for the baseline implemented as a two-layer LSTM-RNN. However, to enforce the disentanglement, the hidden state is no longer initialised on the image. There can be opted to use $\vec{0}$, or to give some information about the context of the image by giving it the global topic embeddings z_0 from (5). Especially the topic embeddings can be beneficial, since this will cause the LM to backpropagate through the embeddings as well and aid in the convergence to descriptive embeddings. This will be tested in Section 4.

3.1.4 Description Model

When the BM opts to use the DM, the positional topic distribution and embedding are computed using (4) and (5). We define φ as a concatenation of z_0, $z_{1:t}$, and one of $\{\emptyset, \mathbf{v}, x_{<i}, [\mathbf{v}; x_{<i}]\}$. Using this, the model has knowledge about the global and the previously used topics, during the selection of the next topics. When chosen to give the model more information about the image it might be able to generate words describing the image better. On the other hand, when chosen to give more information about previously predicted words it can aid in not repeating itself. In Section 4 will be determined if this is the case.

Next, using both the global topic embedding and the positional embedding, we define function g instead of f from Equation 1, for predicting the next token using

$$P_\theta(x_i|\,\mathbf{v}) = \text{Cat}(x_i|g(z_0, z_i; \theta)). \qquad (8)$$

Function g is implemented as a two-layer MLP with a ReLU non-linearity after the first layer.

4 Experiments

First the used datasets are described in Section 4.1. Then some settings are defined in Section 4.2, followed by the experiments for tuning the Baseline Model in Section 4.3. Finally, in Section 4.4, the experiments on the disentangled model are discussed and evaluated.

4.1 Dataset

During the experiments, two datasets were used. The Flickr8k (Hodosh et al., 2013) and the Flickr30k (Young et al., 2014), to have a small and a large dataset respectively. Both datasets consist of images, with five captions. A vocabulary is created from the words in the training set that occurred more than five times. For the Flickr8k it comes with a training, development and test set division. The Flickr30k, a division was created manually by selecting 1000 images for both the test and development set and using the rest for training.

4.2 Training & Model Setup

Since the work from Vinyals et al. (2015) is used as a baseline, we tried to set as many of the hyperparameters identical to theirs. For the encoder part of the models, a pre-trained Inception CNN from Szegedy et al. (2016) is used. The final fully connected layer is replaced by a new one to create a mapping to our embedding size. The hidden size is set to 512 and the embedding size, for both the words and topics, was set to 128.

For training the disentangled model, all the sub-models are trained in parallel by weighing the output from f in (1) and g in (8) with B_i and $1-B_i$ from (7). The same weight of $1-B_i$ must be used for summing the past topic distributions from (6), since they are not used fully in the predictions. The updated equations look like

$$x_i = \text{Cat}\left(B_i f(x_{<i}; \theta) + (1 - B_i) g(z_0, z_i)\right). \quad (9)$$

$$z_{1:t} = \sum_{k=1}^{K} \big(\frac{\pi_{0k}}{t} \sum_{i=1}^{t} (1 - B_i)\pi_{ik}\beta_k\big). \qquad (10)$$

The model is trained using a cross-entropy loss. For validation BLEU (1 through 4), METEOR, ROGUE-l, and CIDer are computed, just as for Microsft COCO captions task (Chen et al., 2015). Furthermore, early stopping with a patience of 10 on the BLEU4 metric is used. The learning rate was set to $1e^{-4}$.

4.3 Baseline Tuning

While many of the parameters were fixed in accordance with Vinyals et al. (2015), a couple still required tuning. Thes are the dropout rate p, the optimiser, and the gradient clipping value. Each of them was tuned separately, by keeping all parameters fixed, except for one. For both datasets

Table 1: Results for tuning the optimiser.

Dataset	Optimizer	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	ROGUE-l	CIDer
Flickr8k	Adagrad	44.464	22.798	11.286	6.587	8.117	32.157	4.961
	Adam	56.464	38.249	24.689	16.038	17.656	41.979	36.262
	RMSProp	**56.701**	**38.824**	**25.66**	**16.949**	**18.05**	**42.511**	**38.763**
Flickr30k	Adagrad	47.128	26.396	14.907	9.409	8.599	33.047	4.67
	Adam	**58.804**	**40.26**	**27.023**	**18.629**	**17.237**	**41.741**	**31.994**
	RMSProp	57.122	38.994	26.144	17.862	16.997	41.38	30.07

Table 2: Results for tuning the dropout rate.

Dataset	Dropout P	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	ROGUE-l	CIDer
Flickr8k	0.1	57.067	38.363	24.748	16.148	17.239	41.488	33.94
	0.2	56.917	38.75	25.492	**16.849**	**17.673**	42.054	37.58
	0.3	**57.458**	38.378	24.661	15.794	16.882	41.207	33.702
	0.4	57.026	**38.857**	**25.54**	16.793	17.638	**42.352**	**38.089**
	0.5	56.464	38.249	24.689	16.038	17.656	41.979	36.262
Flickr30k	0.1	57.517	38.847	25.679	17.503	16.893	41.17	28.186
	0.2	**58.946**	**40.377**	**27.142**	**18.658**	**17.655**	**42.299**	**34.99**
	0.3	58.688	40.193	26.948	18.46	17.109	41.605	30.944
	0.4	57.781	39.066	25.82	17.615	16.753	41.306	28.234
	0.5	58.804	40.26	27.023	18.629	17.237	41.741	31.994

Table 3: Results for tuning the grad clipping value.

Dataset	Grad Clip Value	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	ROGUE-l	CIDer
Flickr8k	2.5	56.477	38.171	24.546	16.016	17.466	42.045	35.151
	5	56.464	38.249	24.689	16.038	17.656	41.979	36.262
	7.5	**57.536**	**39.537**	**26.127**	**17.194**	**18.229**	**42.998**	**39.464**
	10	56.508	37.653	24.042	15.617	17.102	41.159	33.231
Flickr30k	2.5	58.015	39.138	25.831	17.63	16.545	40.965	28.156
	5	58.804	40.26	27.023	**18.629**	17.237	41.741	**31.994**
	7.5	58.421	39.908	26.722	18.373	17.129	41.516	30.213
	10	**59.169**	**40.56**	**27.054**	18.617	**17.368**	**42.015**	31.141

Table 4: The results for tuning the disentanglement feature settings.

BM Features	H Init	DM Features	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	ROGUE-l	CIDer
IMAGE	TOPICS	BOTH	2.56	0	0	0	2.805	5.324	0
		IMAGE	2.56	0	0	0	2.805	5.324	0
		NEITHER	**51.481**	**31.158**	**17.316**	**10.421**	**13.731**	**36.895**	**15.842**
		PAST	2.56	0	0	0	2.805	5.324	0
	ZEROS	BOTH	2.56	0	0	0	2.805	5.324	0
		IMAGE	2.56	0	0	0	2.805	5.324	0
		NEITHER	2.56	0	0	0	2.805	5.324	0
		PAST	48.886	25.784	12.815	7.581	7.798	35.205	1.462
PAST TOPICS	TOPICS	BOTH	48.886	25.784	12.815	7.581	7.798	35.205	1.462
		IMAGE	47.782	26.834	13.254	7.417	11.156	34.111	4.983
		NEITHER	50.94	28.31	13.777	7.798	10.709	34.91	4.21
		PAST	48.886	25.784	12.815	7.581	7.798	35.205	1.462
	ZEROS	BOTH	46.56	23.34	11.86	6.916	10.094	33.27	3.437
		IMAGE	47.782	26.834	13.254	7.417	11.156	34.111	4.983
		NEITHER	43.67	23.155	11.416	6.319	9.915	34.305	4.799
		PAST	47.782	26.834	13.254	7.417	11.156	34.111	4.983

they were tuned separately, since the different sizes may require different settings.

The results for the optimiser, dropout rate, and grad clipping value are shown in Tables 1, 2, and 3 respectively. from these results, there can be concluded that the model behaves slightly different for the smaller dataset compared to the larger one. This is conform the findings by Xu et al. (2015). For the smaller Flickr8k dataset, RMSProp as optimiser with gradient clipping at 7.5 works best. For the Flickr30k, the Adam optimiser with gradient clipping at 10 works best.

The best settings for the dropout rate vary over the used metrics for the Flickr8k. However, for the Flickr30k a dropout rate of 0.2 is consistently the best for the different metrics. Furthermore, early stopping is used with the BLEU4 metric and this is therefore used as most important metric. Hence, the decision was made to use a dropout rate of 0.2 for the Flickr8k.

4.4 Disentangled Model

The assumption is made that the tuned parameters from the baseline model can be translated to the disentangled model. Therefore, only the settings discussed in Section 3.1 will be tuned. These are the features used in the BM and in the DM, and the initialisation for the hidden layers of the LM.

An overview of the results from tuning the settings can be seen in Table 4. There are two interesting points to notice here. First of all, for every combination of setting, the model performs worse than the baseline model. Secondly, the best results are achieved for a combination of using the image descriptor for the BM, no additional features for the DM and the global topic distribution for initialising the LM model. this is especially interesting, since using the image descriptor in the BM is always worse with most scores being zero. The predicted sentences using this optimal case are one of the following:

- a man in a red shirt is standing on a bench .
- a black dog is running through the snow .
- a dog is running through the snow .

For the other cases where the image descriptors in the BM features are used, the sentences look like this:

- a

For the last cases, the same identical sentence is predicted for every sample.

To investigate the origin of these predictions further, the decisions from the BM were printed. For every batch the min, max and mean value at every time-step were printed, as shown in Table 5.

Table 5: Outcome for BM for the first time-steps of sentence prediction for a batch.

Neither			Both		
Min	Max	Mean	Min	Max	Mean
0.19	0.19	0.19	0.0	0.0	0.0
0.8	0.81	0.8	0.0	0.0	0.0
0.97	0.98	0.98	0.0	0.0	0.0
1.0	1.0	1.0	0.0	0.0	0.0
1.0	1.0	1.0	0.0	0.0	0.0
...			...		

This shows that the BM has trouble deciding which model to use, since both are not capable of generating good sentences. When the LM gets enough information about the image in its features, it is able to generate the most likely sentence given the entire set of images. This is what you would see in earlier stages of training the baseline. The LM has not enough information about the image to converge further from this local optimum.

On the other hand, when the LM does not have enough information, the BM might tend to only use the DM. As shown in Table 5 and by looking at its output shown before, it was discovered that the DM always will predict the most probable token in the vocabulary. One of the conclusions that can be drawn from this is that the implementation of the DM is not complex enough. It might lack in layers, hidden size, or in the features it receives. The second point of improvement might be a reduction in complexity for the DM, by reducing its vocabulary. Since the goal of the DM is to predict descriptive words, the vocabulary only has to contain those words. By using Part of Speech tags, only the nouns, verbs and adjectives can be selected.

5 Conclusion

In this study, an attempt has been made to disentangle the decoder part of a caption generator. The two main tasks of the decoder, generating a grammatically correct sentence and describing the visual information in the image, are performed by two separate sub-models, the language model and the description model respectively. A binary switch model is used to decide at each time-step, which of the two is used.

To give the binary switch model and the description model enough information about the image, topic modelling over the image has been used. Different methods of using the topic distributions and creating feature vectors for the sub-models have been tested. Nonetheless, an optimal setup that results in close to par or better results than the baseline was not achieved.

The main issue is the complexity of the task for the description model, which causes the binary switch model to start favouring the language model. In the future, there must be determined if fixes regarding this complexity causes the description model to learn to predict correct tokens. One of these fixes, could be making the model more advanced. Another fix could be limiting the vocabulary for the description model. Since the goal for this sub-model is to generate descriptive information about the content, it mainly has to be able to predict nouns, verbs and adjectives. Reducing the probability of predicting non-descriptive words could aid in a correct disentanglement.

References

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

Chen, X., Fang, H., Lin, T., Vedantam, R., Gupta, S., Dollr, P., and Zitnick, C. L. (2015). Microsoft coco captions: Data collection and evaluation server. *arXiv:1504.00325*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Devlin, J., Cheng, H., Fang, H., Gupta, S., Deng, L., He, X., Zweig, G., and Mitchell, M. (2015). Language models for image captioning: The quirks and what works. *arXiv preprint arXiv:1505.01809*.

Fang, H., Gupta, S., Iandola, F., Srivastava, R. K., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Platt, J. C., et al. (2015). From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1473–1482.

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

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

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404.

Liu, C., Mao, J., Sha, F., and Yuille, A. L. (2017). Attention correctness in neural image captioning. In *AAAI*, pages 4176–4182.

Lu, J., Xiong, C., Parikh, D., and Socher, R. (2017). Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 6, page 2.

Nam, H. and Han, B. (2016). Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4293–4302.

Pan, J.-Y., Yang, H.-J., Duygulu, P., and Faloutsos, C. (2004). Automatic image captioning. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 3, pages 1987–1990. IEEE.

Pedersoli, M., Lucas, T., Schmid, C., and Verbeek, J. (2016). Areas of attention for image captioning. *arXiv preprint arXiv:1612.01033*.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Ren, Z., Wang, X., Zhang, N., Lv, X., and Li, L.-J. (2017). Deep reinforcement learning-based image captioning with embedding reward. *arXiv preprint arXiv:1704.03899*.

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. (2017). Self-critical sequence training for image captioning. In *CVPR*, volume 1, page 3.

Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057.

You, Q., Jin, H., Wang, Z., Fang, C., and Luo, J. (2016). Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659.

Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. (2014). 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.