DeepFiction: Every image tells a story

Shreya Jain (sj2842)

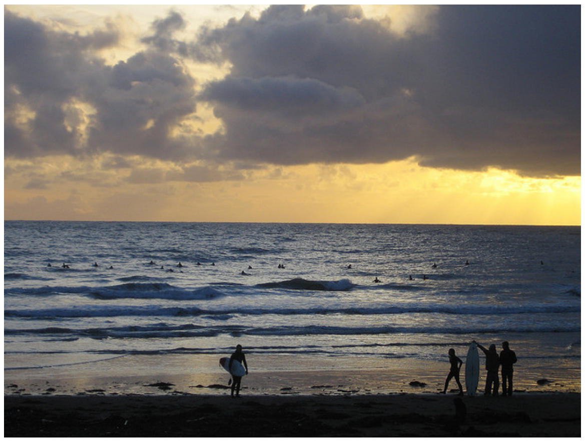
***Abstract*— We introduce the task of generating stories from images, which requires a computer vision system to describe salient regions in images in natural language and generate stories from these captions while preserving thoughts and adding the style of different genres.**

The two major tasks in the process are image captioning and learning how to write stories. To generate captions for new images, we have trained our image captioning model on MS COCO dataset. The model uses a LSTM to encode a joint image-sentence embedding and then natural language model, which focuses on structure and content is trained to generate new captions. To learn the book style, a RNN encoder-decoder is trained on romantic novels of BookCorpus Dataset. From a created mapping, we obtain the vector of book passage than needs to be decoded now to get our story. The resulting stories were noisy as we were using many captions. The captions are improved using two other models. The obtained results which is now using less captions and improved due to these models, are significantly better. The overall noise is less and other details of the image get captured.

I. Introduction

Stories are a fundamental human tool that we use to communicate thought. Creating a story about a image is a difficult task that many humans also struggle with. New deep learning techniques are enabling us to generate stories based on the content of images. The work aims to not just generate story but also specific to a genre. It will first generate descriptions that describe the salient regions of the image and then use these captions to not only capture the thought of the visual data but at the same time transform into a genre specific story.

First, we combine the work done by Kiros in papers [1],[2] to generate story. The work embedded image and sentences in the same multimodal space with the use of LSTM. A language model that focuses on structure and content is used to generate captions. The recommended setting of the project[12] was to retrieve hundred nearest captions to condition on and generate stories from them. These are also ranked using the pair wise loss minimization. This took many captions half of which were quite unrelated to the image and the impact could be seen in the generated story. To get over this flaw, we have improved the quality of captions that are used as input. We use Karpathys research[3] to generate one caption for an image and using this caption. This embedded object regions using CNN and words enriched by context in the same multimodal space. A RNN language model was used to generate captions. If the caption was used alone then because of very less content the story was very repetitive and not at all meaningful. When used along with 99 captions would not matter as the noise would still be there. It was seen using a total of twenty captions, was producing good results. So we used one of this caption and rest from the baseline model ranking the captions in the this order, we could see that the stories weren't perfect but it was also visible that they would do better with more quality captions.



(a) ”We were a few people at the beach, and it felt so beautiful that I could barely catch my breath as the sun slipped through the sky. She had no intention of falling in love with him, but she wasn't going to be able to walk away from the beach after the ceremony. The beach was beautiful, the sun on the horizon, and the memory of the ocean passed out on the beach. She had no idea what to do next. It was such a strange feeling, as if Alex and his friend had come up on the beach. Her body language betrayed her. No, he seemed to want people at the beach in record time.”

Fig. 1: Romantic Story for the given picture

To make better stories that incorporated more details and werent so repetitive, we must capture more details in the captions. This led us to use architecture given in [4] to generate captions for regions of the image but also a score was associated with them that helped us rank them for generation of story. After multiple runs, we see that ten captions from dense cap are quite helpful in correcting grammar and capturing some of the details. On multiple runs, a total of 20 captions were used, 1 good caption, 10 details and the rest from Baseline Model (in this ranking) and that produced the best results.

*This work was done as project for course Deep Learning for Computer Vision taught at Columbia University spring 2018 with Prof. Peter Belhumeur

[1]Alaap Sivaram is a student at Columbia University pursuing his Masters in Computer Science

[2]Maitri Shroff is a student at Columbia University pursuing her Masters in Computer Science

[3]Shreya Jain is a student at Columbia University pursuing her Masters in Computer Science

II. Overview of the Baseline Model

The model is composed of three parts -

- Image Sentence Embedding
- Skip Thought Vectors and RNN Encoder Decoder
- Style Transfer

III. Baseline Model

A. Sentence Image Embedding

Long short-term memory[8] are recurrent neural networks that incorporate a built in memory cell to store information and preserve the context of the language. LSTM (Long Short Term Memory) are used to encode sentences and CNN (Convolutional Neural Network) are used to encode features of images obtained in the hidden states of LSTM. A pairwise ranking loss is minimized in order to learn to rank images and their descriptions[1].

For language model, we use something that disentangles the string of a sentence to its content, conditioned on distance representations produced by encoder. Every description can be represented as S = w_1,..,w_N where w_1,..w_N are the words the description is made of. Along with this we are also given a sequence of word related variables T = t_1,..,t_N. For this model these variables t_i corresponds to the POS for word w_i. From the given embedding of context word, our aim is to model a probability distribution P(w_n = i$|w_{1:n1}$; $t_{n:n+k}$; u) from previous word context $w_{1:n1}$ and forward structure context $t_{n:n+k}$, where k is the forward context size[1]. Basically it can be thought of in such a way that these structure variables help the model during the generation phrase and also prevents the model from making grammatical mistakes.

The benefit of the approach is this language model as it uses structure and context, it can be trained on only descriptions as well. This would make it easier for it to be trained on large corpora as it does not need a image-description pair for training. Being trained on large texts, it has good caption quality. The architecture for the image captioning module of our baseline model is detailed out in Fig 2.

This can retrieve k number of captions where k is user specified. [12]

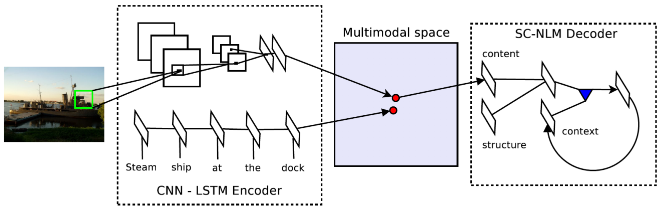


Fig. 2: Image Sentence Embedding

B. Skip Though Vectors and Conditional RNN

In natural language, the words around the current word are always the ones providing context as it is possible for previous as well next sentence could provide context. For example, "Mary was playing the garden. A bug bit her." We can see that in the second sentence, context from the previous sentence is being used in the next sentence. The word 'her' refers to Mary, but basic skip-grams would not be able to pick up on this as they operate on words. This is done using Skip Thought vectors.

Skip Thought Vectors are basically simple neural networks model for learning fixed length representations of sentences in any natural language. It is not a supervised learning task. We only pass the order of the sentences in a large natural language corpus[2]. Skip thought vectors work similar to Word2Vec representations of words in terms of representing variable length input to fixed length vectors. Fixed size representations of sentences make the processing, understanding and mathematical calculation much simpler. Word2Vec learns fixed length representations of individual words. But in the case of sentences, the order of words is also important for example, the sentences "Mary played with Alice" and " Alice played with Mary" are two different sentences. Just word representations would not be able to take care of the order. Basically skip thoughts have neural network take care of the order of the sentences. Our skip thought vectors are trained on contiguous sentences of romantic genre of Book Corpus. Given to us is a tuple (s_i, s_{i-1}, s_{i+1}) as input to the model. Skip-Thoughts model has three parts:-

- Encoder Network: Takes the sentence s(i) at index i and generates a fixed length representation z(i). This is a recurrent network (generally GRU) that takes the words in a sentence sequentially. Let w_t denote t^{th} word for sentence s_i. Sentence s_i contains words $w_i^1, w_i^2, ..., w_i^N$
- Previous Decoder Network: Takes the embedding z(i) and tries to generate the sentence s(i-1). This also is a recurrent network (GRU) that generates the sentence sequentially. In such sequence modeling tasks, the performance of the GRU is similar to that of an LSTM while being much simpler. GRU units have only 2 gates and don't require the use of a cell.
- Next Decoder Network: Takes the embedding z(i) and tries to generate the sentence s(i+1). Again a recurrent network similar to the Previous Decoder Network.
  The vocabulary of the encoder is small and it is possible to not come across all words in training, we can use the Words2Vec corpus for such words. Let V_{w2v} be the vocabulary for word2vec and V_{rnn} be our vocabulary. We know that V_{w2v} is much larger than V_{rnn} and hence a mapping from words in V_{w2v} to words in V_{rnn} is required. To do so, we can create a mapping f: $V_{w2v} \rightarrow V_{rnn}$ parameterized by vector W such that $v' = Wv$ where $v \in V_{w2v}$ and $v' \in V_{rnn}$. To find this W, we solve an un-regularized L2 linear regression Loss. This is helpful as now we can deal with unseen words while testing.

C. Style Transfer

Given the joint embedding of text and images and the RNN network models, we need to bridge the gap between the retrieved image captions and passages in novels. It means we would like to have a function F that maps a collection of image caption vectors x to a book passage vector F(x), so that we could then feed F(x) to the decoder to get our story.

$$F(x) = x - sx + sfx \tag{1}$$

where x is the caption obtained, sx is the style of the caption, $F(x)$ is the book passage vector and sfx is the style of the book passage. For our project we constructed c by taking the mean of the skip-thought vectors for Microsoft COCO training captions. Similarly for constructing b, we take the mean of the skip-thought vectors for romance novel passages that are of length greater than 100. We limit the length to be greater than 100 because any lesser than that is plain noise. This will give us an embedding for a sentence that by sense is identical to the original embedded sentence, but one that has the flavor or genre of the book style that the decoder has already been trained on.

IV. Improvements

Image captioning brings together two key areas in artificial intelligence: computer vision and natural language processing. For the computer vision side, researchers train their systems on a massive dataset of images, so they learn to identify objects in images. Language models can then be used to put these words together. These model are built with the goal to associated enriched sentences with images.

A. Better Captions

Here, we use a CNN pre-trained on ImageNet and fine-tuned on the 200 classes of the ImageNet Detection Challenge. CNN(I_b) transforms the pixels inside bounding box I_b into 4096-dimensional activations of the fully connected layer immediately before the classifier [3]. The matrix W_m has dimensions h $\times$ 4096, where h is the size of the multimodal embedding space. Every image is thus represented as a set of h-dimensional vectors. The baseline model gave the top 5 captions as shown in Fig 3. Fig 4 shows the single good caption that was generated.

We use Bidirectional RNN[10] takes a sequence of N words as input and outputs each one into an h-dimensional vector. Each word is enriched by context around it (could be variable in length). This has succesfully built transformations that map every image and sentence into a set of vectors in a common h-dimensional space.

We are interested in associating snippets of text instead of just single words we are required to align text sequences with bounding boxes. This can be solved by accessing true alignments as latent variables in a Markov Random Field, which encourages neighboring words align to the same region.Given to us is an input set of images and their textual descriptions. The key challenge left now to develop a model that can predict a variable-sized sequence of outputs corresponding the input image. The RNN is trained to combine a word (x_t), the previous context (h_{t-1}) to predict the next word (y_t). The image context vector is also fed to RNN's first iteration.

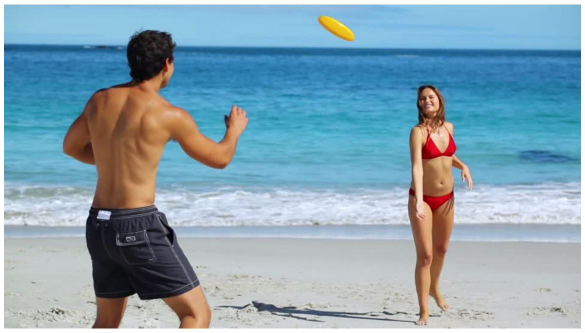


Fig. 3: Top 5 Captions from Baseline Model : Two young people are playing volleyball together on the beach, A couple of men playing frisbee on a sandy beach, Adults and children enjoying a game of beach volleyball, Men playing on the beach with a frisbee, A man catching a frisbee from a woman on a beach

Fig. 4: "Man and woman are playing frisbee on the beach"

The training is done by setting h_0 = 0, x_1 to a special START vector, the first word in the actual sequence is the desired label to START. Then x_2 is set to the word vector of first word and similarly we keep going. The last one is when x_T represents the last word and the target label is set to a special END token. The cost function is to maximize the log probability assigned to the target labels.

This produces one good caption for the entire image given. An example is given in the Fig .

B. Captions with Details

Convolutional Layer - An image of width W and height H is passed through a VGG-16 architecture (13 layers of 3x3 convolutions with 5 layers of 2x2 max pooling in the middle)[9]. The final pooling layer is removed and tensor of features of shape 512 x W' ($\frac{W}{16}$) x H' ($\frac{H}{16}$). This is the input to the localization layer.
Localization layer - It most essentially identifies spatial regions of interest and smoothly extracts a fixed sized representation from each region. Input : Tensor of activations of size $C \times H/16 \times W/16$ Internally selects B regions and returns three output tensors

- Region Coordinates : Matrix Bx4 giving bounding box coordinates for each output region
- Region Scores : Vector of length B with confidence score of each region.

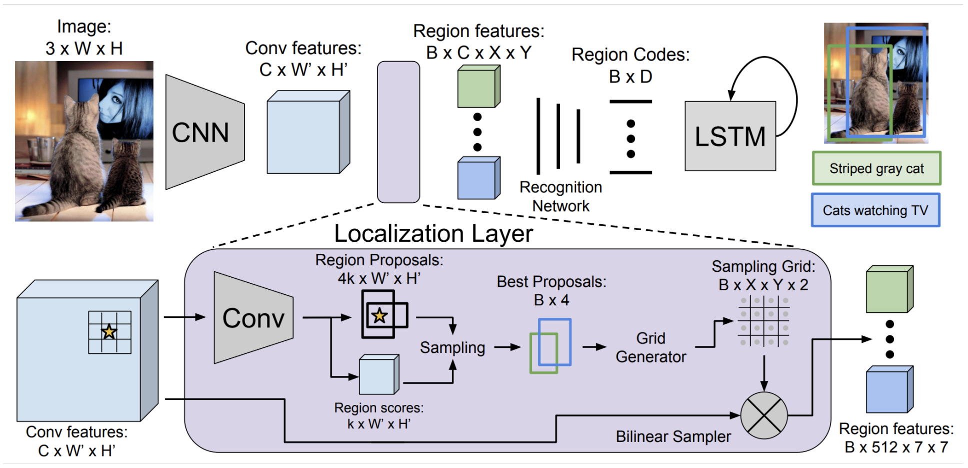


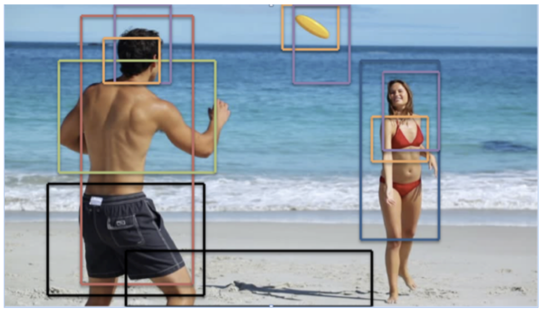
Fig. 5: Architecture for model 3

- Project each point in WxH grid of input back into WxH image plane Consider k anchor boxes of different sizes centered at this projected point. For each k box, a confidence score and four coordinates are predicted.
- The input feature map is passed through a 3x3 conv with 256 filters, then ReLU source of non linearity is introduced, further putting it through 1x1 with 5k filters to get $W/16 \times H/16 \times 5k$.

Recognition Layer - Features from each region are flattened into a vector and passed through 2 Fully Connected Layers, each with ReLu (source of nonlinearity) and regularized using Dropout. Each region produces a code of dimension D=4096 that compactly encodes its visual appearance.Codes for all positive regions is collected and put in matrix BxD which passed to RNN. Also refines the confidence and position of each proposed region.

RNN Language Model - Training sequence of tokens s_1, s_2, ., s_t feed the RNN + 2 word vectors x_{-1},x_0,x_1, x_2, . x_t where x_{-1}=CNN(I) and x_0 is start token. RNN computes a sequence of hidden states h_t and output vector y_t using recurrence formula $h_t, y_t = f(h_{t-1}, x_t)$ (LSTM is used). Output vector size is V+1 where V is the token vocabulary and one is special END token. Targets at time t=0,1,.T-1 are token indices for s_{t+1}. At test time x_{-1} info is fed to the RNN. At each step we sample the most likely next token and feed it too RNN in the next time step, repeating until END token is reached.

This produces one phrase for a region given. An example is given in the Fig 6.



'black shorts on man','man standing on the beach',
'woman in a bikini','man with no shirt','man has short hair'
'the frisbee is yellow','woman wearing a red top', 'man with short hair'
'a volleyball','woman with long hair','the man is surfing'

Fig. 6: Captions

V. Dataset

- MSCOCO - The publically available and most used dataset is made by Microsoft. It is a standardized dataset made for object detection and image understanding. It contains 82,783 images with 5 captions each [7].
- Visual Genome - It comprised of 94000 image and 4100000 captions for multiple of region of interests in the images. In this dataset, humans on Amazon Mechanical Turk generate all the captions given and the descriptions provided have a bounding box with the corresponding region. This image set is made from the combination of two datasets - MSCOCO and YFCC100M. Each image contains approximately 21 objects, 18 attributes and 18 pairwise connections between objects.
- BookCorpus - Dataset made by University of Toronto is only available for academic research and it contains 11,038 books with many genres[14].
- Children's Book - This publically available dataset was made available by Facebook Research and the purpose was for question answering and had fill in the blanks [13].

VI. Experiments

- We attempted to train the RNN encoder decoder on Children's Book dataset. It was cleaned and preprocessed to be in the format like that of Bookcorpus datasets.
- We tried to make our detailed captions longer by joining them with an appropriate pronoun. That did not seem

to help the model. Hence, it was dropped.

- We were trying to weight the good caption more and feed it to generate the stories but that resulted in repetitive sentences.

VII. Evaluation

BLEU which stands for Bilingual Evaluation Understudy[11], is a score for comparing a candidate translation of text to a list of reference translations. Essentially Although developed for machine translation, it can be used to evaluate text generated for a suite of natural language processing tasks.

$$B_N = \min(1, e^{(1-\frac{r}{c})}\frac{1}{N}e^{\sum_{(}{}^{n=1)^N}} \log(P_n)$$

where r,c are the length of reference sentence and generated sentence respectively and p_n is the modified n gram precisions. A perfect match results in a score of 1.0, whereas a perfect mismatch results in a score of 0.0. The score was developed for evaluating the predictions made by automatic machine translation systems. It is not perfect, but does offer compelling benefits:

- Calculations are easy and can be done quickly.
- Understandable metrics.
- It is not dependent on the language being used.
- It has been seen that this correlates with human evaluation.

The idea is simple ; it tries to match the N-grams between the candidate and reference translations and computes the scores on the basis of that. Here, we have calculated average 1-gram and 4-gram BLEU scores to compare the first two models and we can see that our model gives an improvement.

TABLE I: BLEU Metrics for Evaluation

Score	B@1	B@4
Model1	0.423	0.227
Model2	0.515	0.253

Apart from the BLEU score, we also made use of a package called language-check in python programming language which is able to detect the grammatical errors and small mistakes like using couldnt in place of dont, are also identified. It identifies if the pronouns are messed up or syntactic structure of given sentence is not correct. These average values are over a subset of our test dataset and as you can see that our final model makes the least mistakes. The following table contains our evaluation

TABLE II: Average of Grammatical Mistakes

Story from Model	Mistakes per story
Baseline Model	12.96
Model with better caption	13.42
Model with details	9.73

(a) ”Everyone had a few friends at the beach, and for the first time in my life she felt the urge to hug Nate. He had no intention of telling him what to do. She was acting like an injured kid, and now she rode to be the beach. The beach was on the beach, so it had nothing to do with the tattoo artist on his arm. In fact, it felt as if hed just pulled her into his arms and carried her out of the camp, leaving Chelsea and her friends and friends all over the beach on horses. Such a miracle. I could hardly hear what happened to Wes, since the beach girls were alive.”

(b) ”Everyone had a few friends at the beach, and for the first time in my life, She felt like she was going to jump up and fall asleep on the beach. She had no idea what he wanted to do with her, so he didn’t have a choice. She shook her head at Josh, his arm still wrapped around her waist, and he pulled her into his arms . There was nothing to talk about, so Im pretty sure it wont be the same. The only person on the beach , I wished Id joined them in the sand.”

(c) ”She was a girl on the beach, and she laughed at my touch. She felt the tension in her body, as if he had a few days to get back to the beach. She had not seen Nate for a while, so it would be the perfect time for she to discuss what happened with the other team. She loved Nate, and she had no intention of going back to New York. She felt like a fish in the sand, so he carried his arm around her waist and pulled out his T-shirt. The girl looked beautiful playing beach volleyball. She could hardly recall what happened between them on the beach.”

Fig. 7: Story from three models

VIII. Conclusion

We introduced a model that generates genre specific story based on the natural language descriptions of given image. First model, we embedded image-text as a joint embedding and got captions using a language model that used structure and context. This was developed using works [1] and [2]. This recommended to use 100 captions to condition on. This lead to noise creeping into the generated images. We can see in Fig. that the first story says beach in every sentence and words like horse, tatoo artist creep in. These are probably coming in as the training data was biased and picking 100 captions is not helping eliminate non existing objects.

Now, we use an image captioning model that embeds image-words into a common multimodal space. The captions are generated for this model using an RNN language model[3]. The second story is a slight improvement as we take only 20 captions. The word beach appears lesser and words like horse do not appear. The story is still very generic.

The model introduced now associates regions of images with phrases. It was seen the top ten pairs of regions and phrases were a sufficient number to specify details[4]. These captions help add detail as words like beach volleball, sand, pulled out his T-shirt. There is no volleyball game but that comes as it is very intuitive and also romantic novels might not have frisbee. Refer fig 7 for the stories generated by each model.

The generated stories have flavor of romance that can be understood as we read them. Also although grammatical mistakes exist, but our model has the least of them. Unlike humans, it is extremely difficult for computer systems to do perfectly well on grammar but it is improving woth better training sets being released.

Thus, our final model is not perfect but does better than all the other and must be developed further to give better results.

IX. Future Scope

Instead of taking captions separately from these three models, we will train an end to end image captioning model which would give some quality captions sufficient to produce good stories. Also we need to figure out a global scheme to rank the image descriptions.

The model will be trained on more genres to check much it can be generalised across corpuses. We have already started training it on Children's book corpus.

The existing model still shows errors in terms of pronouns messing up (he and she getting messed up), proper nouns creeping in, some noise because of biased dataset creeps in and context gets lost. All this needs to be dealt with to improve the quality of the generated stories. Also we aim to generate longer stories on our given input or using multiple related images.

We can also incorporate GIFs or videos as another input to generate longer stories with more context.

Acknowledgement

We thank Siddharth Prasad for helpful comments and discussions. We gratefully acknowledge the support of Google with the donation of the GPUs credits used for this research. We would also like to thank Prof. Peter Belhumour who provided us with the opportunity to pick such a project and independently work and develop it.

References

[1] Ryan Kiros, Ruslan Salakhutdinov, Richard S. Zemel. "Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models." arXiv preprint arXiv:1411.2539 (2014)

[2] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-Thought Vectors arXiv preprint arXiv:1506.06726 (2015)

[3] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. arXiv preprint arXiv:1412.2306, 2014

[4] Justin Johnson, Andrej Karpathy, Li Fei-Fei. DenseCap: Fully Convolutional Localization Networks for Dense Captioning arXiv:1511.07571 (2015)

[5] D. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012

[7] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollar, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325, 2015

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):17351780, 1997.

[9] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015

[10] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. Signal Processing, IEEE Transactions on, 1997

[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 311318. Association for Computational Linguistics, 2002

[12] https://github.com/ryankiros/neural-storyteller

[13] Felix Hill, Antoine Bordes, Sumit Chopra, Jason Weston. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. arXiv:1511.02301, 2015.

[14] Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, Sanja Fidler. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. arXiv preprint arXiv:1506.06724 (2015)