Neural image caption generator on WikiArt dataset

Dishani Lahiri[1], Srishti Sardana[2]

[1]*Student, Electronics and Communication Engineering, Delhi Technological University, Delhi-110039, India*

lahiri.dishani367@gmail.com

[2]*Student, Electronics and Communication Engineering, Delhi Technological University, Delhi-110039, India*

sardana.srishti@gmail.com

Abstract: **With the motivation to design and develop more intelligent systems, the idea of neural image caption generator was born. The model described in this paper generates captions automatically for a given image. The images from the dataset are fed into the AlexNet deep CNN model to be classified based on the label. This encoded image along with the word count file are fed into the LSTM Units. Besides the encoded image and word count file, LSTM is also fed with the word embeddings which help predict the next word in the sequence. Finally, the log likelihood of the summation over the probabilities of the predicted words is minimised to reach an optimum solution. Instead of selecting the most probable word, top K candidates are selected which finally leads to top K captions. In this paper, we have taken K as 3.**

Keywords- Caption Generation, Convolutional Neural Networks, Deep Learning, AlexNet Architecture and LSTM.

I. INTRODUCTION

There is an increasing concoction of the two major fields of Artificial Intelligence, Deep Learning and Natural Language Processing. This amiable trend is leading to prime innovations. A few of them being text classification, language modelling, speech recognition, machine translation, document summarization, sentiment analysis and caption generation.

The introduction of deep learning in NLP has revolutionized the field majorly because it is able to condition the models on long sequences. Also as is the property of Neural Networks that the model extracts features itself and learns from them over time makes the incorporation to NLP more appealing. The implementation of deep neural network in the field of NLP also ensures continuous improvement in results.

This monumental leap from classical NLP has been illustrated well in the following figure.

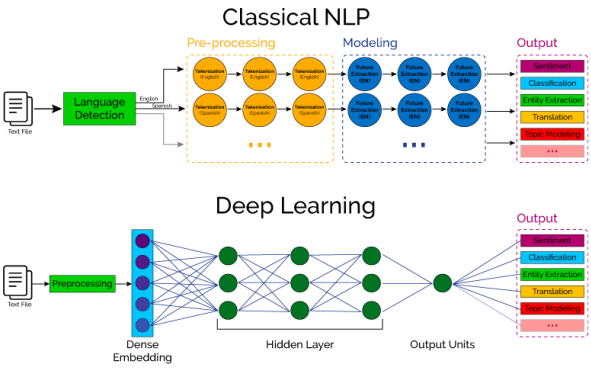

Fig. 1: Flow Diagrams of Classical Natural Language Processing Algorithms and Deep Learning Algorithms

Out of the eclectic applications of neural networks in Natural Language Processing, automatic caption generation given an image is of prime importance. It can act as an aid in scene description for blind citizens and help them lead independent and comfortable lives. Automatic caption generation can be implemented in understanding the scene better and assist in fundamental activities like driving a car by analysing the road conditions and traffic. This can improve overall road safety for the pedestrians and also the vehicle-owners.

In our paper, the model implemented for caption generation involves a deep Convolutional Neural Network (CNN) implemented through AlexNet architecture followed by Long-Short Term Memory (LSTM) Units for generation of the captions. The LSTM is fed with images encoded by the AlexNet and the word count file. The word count file contains the frequency of all unique words present in the corpus. The corpus is the collection of all documents, which here is the collection of all original image titles. The

collection of all the unique words in the corpus forms the dictionary or vocabulary. Later on, the word embeddings are formed to input to the LSTM cells at each time step 't'. The word embeddings are vectors of the size of the dictionary with 1 at the index of the current word in consideration and rest 0.This form of encoding is known as one-hot encoding.

Another key contribution of this paper is that we have used the WikiArt dataset which is unlike the usual datasets (like MSCOCO) which only consist images of common objects and scenes. The WikiArt dataset on the other hand contains complex images forming a sundry collection. Paintings from the Renaissance, baroque, rococo era and also modern art is present in the dataset which add to the complexity of the classification task of the CNN. Despite the challenges, our model produces vivacious results.

II. RELATED WORK

The problem of generating natural language descriptions from visual data has long been studied in computer vision, but mainly for video [7, 32]. Such systems are heavily hand-designed, relatively rudimentary and have been demonstrated only on limited domains, e.g. traffic scenes or sports. The problem of still image description with natural text has gained interest more recently. Leveraging recent advances in recognition of objects, their attributes and locations, allows us to drive natural language generation systems, though these are limited in their scope. Farhadi et al. [6] use detections to infer a triplet of scene elements which is converted to text using templates. Similarly, Li et al. [19] start off with detections and piece together a final description using phrases containing detected objects and relationships. A more complex graph of detections beyond triplets is used by Kulkani et al. [16], but with template-based text generation. More powerful language models based on language parsing have been used as well [23, 1, 17, 18, 5]. The above approaches have been able to describe images "in the wild", but they are heavily handdesigned and rigid when it comes to text generation. A large body of work has addressed the problem of ranking descriptions for a given image [11, 8, 24]. Such approaches are based on the idea of co-embedding of images and text in the same vector space. For an image query, descriptions are retrieved which lie close to the image in the embedding space. Most closely, neural networks are used to co-embed images and sentences together [29] or even image crops and subsentences [13] but do not attempt to generate novel descriptions. In general, the above approaches cannot describe previously unseen compositions of objects, even though the individual objects might have been observed in the training data. Moreover, they avoid addressing the problem of evaluating how good a generated description is. In this work we combine deep convolutional nets for image classification [12] with recurrent networks for sequence modeling [10], to create a single network that generates descriptions of images. The RNN is trained in the context of this single "end-to-end" network. The closest works are by Kiros et al. [15] who use a neural net, but a feedforward one, to predict the next word given the image and previous words. A recent work by Mao et al. [21] uses a recurrent NN for the same prediction task. This is very similar to the present proposal but there are a number of important differences: we use a more powerful RNN model, and provide the visual input to the RNN model directly, which makes it possible for the RNN to keep track of the objects that have been explained by the text. As a result of these subtle differences, we achieve better results.

III. METHODOLOGY IMPLEMENTED

The methodology implemented in this paper for generation of caption is captured in the following model architecture:

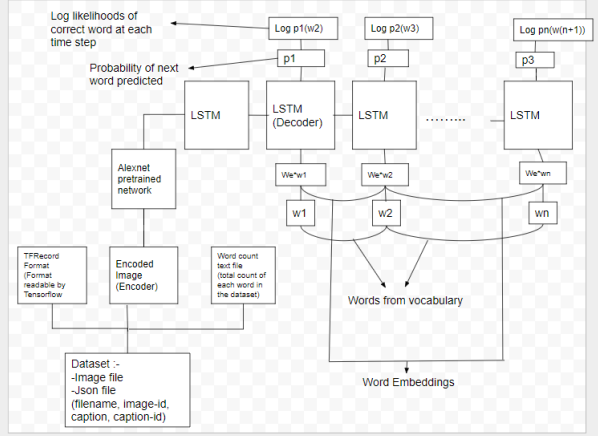

Fig. 2: Block Diagram of Neural Caption Generator

A. *Image Encoding Using AlexNet Architecture*

The various images from the dataset are fed to the open source pre-trained AlexNet image recognition model to be classified on the basis of styles or labels. This classification on the basis of styles/labels is referred to as image encoding.

The AlexNet Architecture, developed by Alex Krizhevsky, had been a breakthrough in the field of Computer Vision. It consists of 5 Convolutional layers, Max-Pooling layers, Dropout layers and 3 Fully Connected layers. The model uses Rectified Linear Units (ReLU) as the activation function which

reduces the training time with respect to their equivalents using conventional tanh function.
In the original model, the output of the final fully connected layer is connected to 1000-way Softmax which would produce a distribution over 1000 classification styles/labels. In our project, we have 21 categories of images, hence 21-way Softmax has been used.

The AlexNet Architecture has been illustrated below:

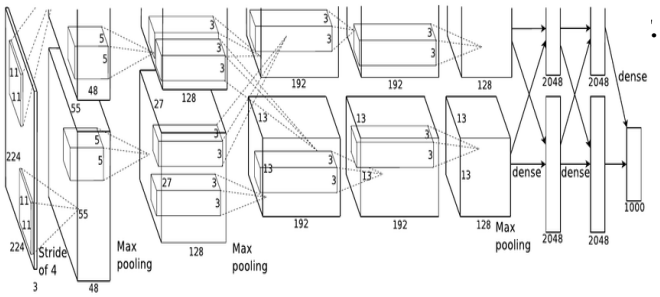

Fig. 3: AlexNet Architecture

B. *Generation of the Word Count File*

During the training of the model, the word count file is also generated. The word count file essentially contains the frequency corresponding to each unique word or token in the corpus.

The corpus C is a collection of the 'N' documents [A_1, A_2… A_N] having 'P' unique words or tokens. The collection of the entire word set forms the Dictionary D. In the word count file, the count of the number of times each of the 'P' words appears in C is saved.

The probability of a word being predicted in the caption over other similar words for the same image is directly proportional to its frequency in the word count file. Therefore the high frequency words will have higher importance, hence higher weight over others. For example, consider the word 'boat' has a frequency numerous times over the word 'canoe' in the word count file. The captions generated for images corresponding to boat-like objects will have a higher tendency to be 'boat' rather than a 'canoe'.

Corresponding to the dictionary D formed, the Word Embeddings generated in a later stage are fed as input to the LSTM for generation of captions.

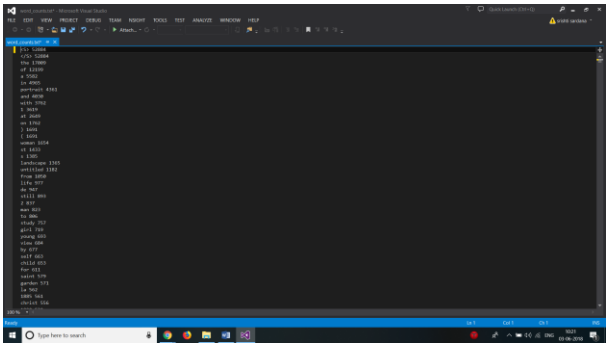

Fig. 4: Word Count File

The LSTM Unit

The LSTM cells form the nucleus of the entire image caption generator model. The LSTM is a variation of the Recurrent Neural Network (RNN). An LSTM cell has the unique property of being able to block and pass as much information as desired through the input gate, forget gate and output gate. Just like in any RNN, an LSTM output at time step 't' will be a function of input at time step 't' and the previous output at time 't-1'. Thus, it is often said that RNNs have memory.

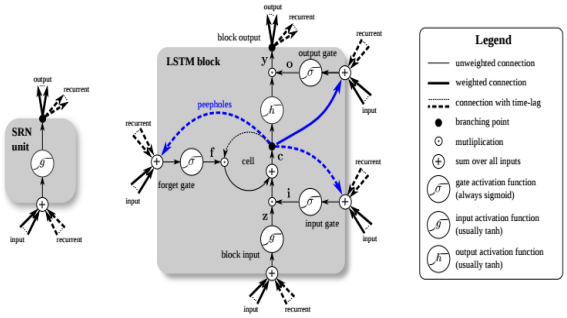

Fig. 5: Simple RNN Unit(left) and an LSTM Cell(Right)

In this model, the output from each LSTM cell at time step 't' is a function of the word embeddings received as input at time 't' and the output from the previous LSTM cell at time step 't-1'. After encoding the image into a fixed-length vector representation, the dictionary of words is generated. Each word from the dictionary is one-hot encoded, hence the word embedding corresponding to each word. Each word embedding is of a dimension equal to that of the Dictionary. The presence or absence of a word in a document is marked by '1' and '0' respectively. These word embeddings at each time step 't' is fed as input to corresponding LSTM cell. The LSTM Unit is trained as a language model conditioned to the encoded training images.

When captions have to be generated for the test images, initially the images encoded by the AlexNet model along with the word embeddings weighed by the Weight vector W are fed to the LSTM Unit. The

LSTM Unit henceforth takes the word embedding at time step ‘t’ and the previous output at ‘t-1’ as inputs to output the probabilities of the ‘K’ next possible words. Probability associated with each word in the dictionary is a function of the frequency of that word in the Word Count file previously formed.

The key role of LSTM here is the utilization of the previous outputs to generate the present output which is a fundamental in sentence formation. Each word in a meaningful sentence has relations with every other word in the sentence.

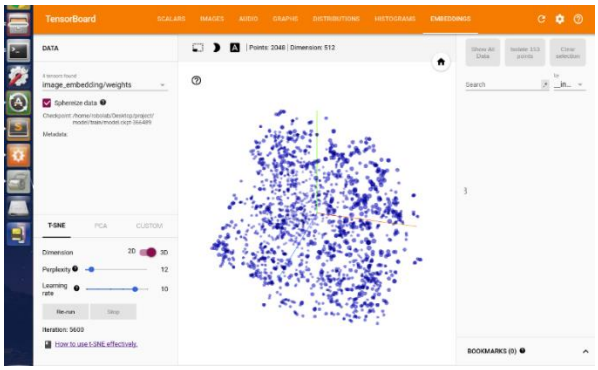

Fig. 6: Representation of data on TensorBoard

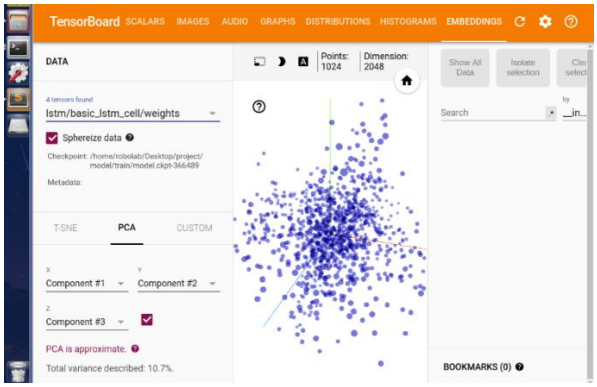

Fig. 7: Representation of data after applying Principal Component Analysis(PCA) on TensorBoard

D. LSTM Output

Corresponding to the word embeddings and encoded image, the trained LSTM outputs the probabilities of the next possible words. The maximum number of words in a sentence is a hyperparameter. To reach an optimum solution, we aim at minimising the negative sum of the log probabilities of the succeeding words.

Through a greedy search procedure, one can select the words with the highest probabilities at each time step ‘t’. By making greedy choices at each step, we wish to reach the global optimum. Nevertheless, this method is often flawed and we use Beam Search to solve this problem.

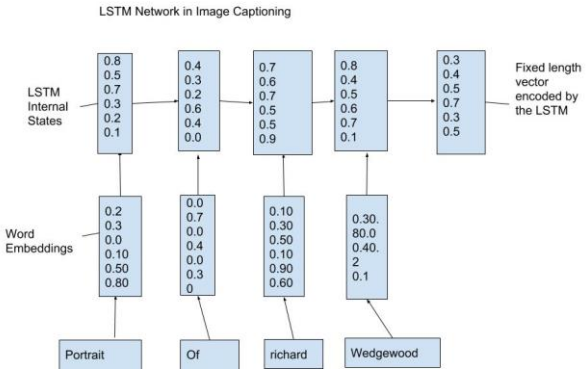

Fig. 8: Example of a trained Model Architecture

- $CNN(I) = W(I)\, g(I) + b\,(I)$

(converts into 512*1 input dimensions of the image, which is fed into the LSTM network)

- x−1 = CNN(I) **(image representation fed into lstm)**
- xt = WeSt for t = 0...N − 1

(‘We’ is the word embedding matrix shown in figure, Si is a one hot vector representation of the word (|V|*1), S0 and Sn are start and end of the sentence respectively).

- a) it = σ(Wixxt + Wimmt−1) **(input gate)**
  b) ft = σ(Wfxxt + Wfmmt−1) **(forget gate)**
  c) ot = σ(Woxxt + Wommt−1) **(output gate)**
  d) ct = ft* ct−1 + it *tanh(Wcxxt + Wcmmt−1)
  e) mt=ot*ct
  f) pt+1=softmax(mt)
  (Returns pt+1, and (mt,ct) is transferred as current state to the next state)
- pt+1 = LSTM(xt) for t = 0...N – 1 **(hidden state of LSTM as shown in the figure predicts the next word in the sentence)**

- J(S|I; θ) = − summation(t=1 to N) log pt(St|I; θ)

Minimizing the loss function , log of probability of predicting the correct word St at time t) (Loss function to tune the parameters)

E. Beam Search

In order to ensure that the final caption generated is the best that the model could have generated, we escape to Beam Search. First, we define the beam size 'K' as 3. Now, top K candidate words are selected at time 't-1'. These are used to predict the candidate words at time 't'. Corresponding to K candidate words at each time step, there are K possible captions corresponding to each word. Thus we are able to generate meaningful captions given an image with very high accuracy.

IV. DATASET USED

As opposed to the Microsoft COCO (Common Objects in Context) dataset, the WikiArt dataset consists of a diverse and mostly complex categories of images. The MS COCO Dataset consists of category labels which are commonly used by humans in day –to-day life. It consists of 330K images and 80 object categories. On the other hand, the WikiArt, dataset formerly known as WikiPaintings, is an online encyclopedia of 52,000 paintings from 195 artists.

Thus, the WikiArt dataset consists of 52,000 images and paintings over 21 categories. The classification of images from the dataset is made more difficult due to the presence of Modern Art paintings too. Modern Art in its true sense expresses the philosophy of the artist, casting away all traditional forms of painting.

There is also a category of paintings from the Renaissance era. The presence of unusual and complicated paintings makes the caption generation for these images a colossal task. Despite the challenges faced, the model generates captions which are of supreme accuracy and able to define the scene with high preciseness.

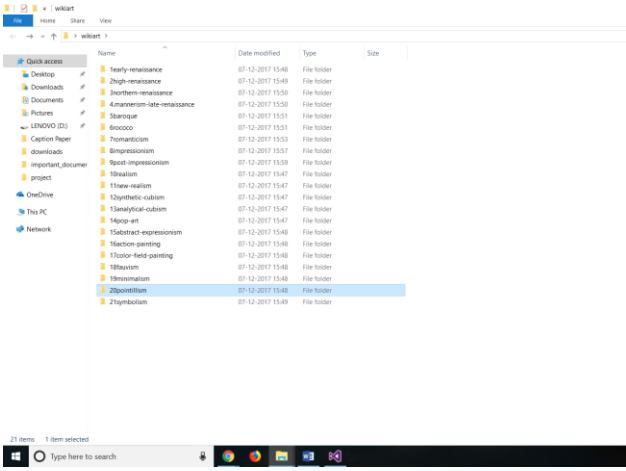

Fig. 9: Categories of Images

Fig. 10:Examples of images from the WikiArt Dataset

V. EVALUATION METRICS

A. BLEU

Bilingual Evaluation Understudy (BLEU) is a machine translation evaluation metric which is used for assessing the similarity between the generated captions and the original title of the image. The range of BLEU score is between 0 and 1. Greater the similarity between the texts, closer the BLEU score is to 1 and vice versa.

BLEU algorithm is basically a modified n-gram precision as each unigram in the candidate text can be considered as retrieved elements. The BLEU score does not evaluate semantic and syntactic correctness of a sentence.

It takes into account the maximum number of times each word in the candidate appears in the reference text. This count for each word in the candidate is summed and divided by the number of words in the candidate text. Hence, it can be induced that the BLEU score tends to favour shorter sentences.

$$P = \frac{\sum \frac{m_{max}}{m_w}}{w_t}$$

B. ROUGE

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is a modified n-gram recall machine translation evaluation metric. It takes into account the number of words which are matching between the generated caption and actual image title.

The final ROUGE score is calculated by dividing the number of matching words to the total count of words in the reference.

The ROGUE score ensures to penalize unnecessarily long candidate text generation by the machine.

$$P = \frac{m_o}{m_t}$$

C. METEOR

Metric for Evaluation of Translation with Explicit ORdering (METEOR) score is yet another machine translation evaluation metric. It helps circumvent many problems present in the BLEU score.

METEOR score takes into account n- gram precision and n-gram recall. Other than simply exact word matching, it also utilises synonym matching to make it more alike to human evaluation.

The score is calculated after selection of the final alignment. The alignment that has the minimum intersections in its mappings between the candidate text and reference text is selected.

$$P = \frac{m}{w_t}$$

Where

$$R = \frac{m}{w_r}$$

$$F_{mean} = \frac{10PR}{R + 9P}$$

D. CIDEr

Consensus-based Image Description Evaluation (CIDEr) is a novel machine translation metric which evaluates TF-IDF (Term Frequency Inverse Document Frequency). This TF-IDF value shows the similarity between the generated caption and original image title.

Followed by this, the cosine similarity between the reference text and the candidate text is evaluated which is the final CIDEr score.

The CIDEr score as an evaluation metric is superior to the previous evaluation metrics.

$$CIDEr(c_i, S_i) = \frac{1}{m}\sum_j \frac{g^n(c_i).g^n(s_{ij})}{\|g^n(c_i)\|\|g^n(s_{ij})\|}$$

E. SPICE

Semantic Propositional Image Caption Evaluation is the most accurate machine translation evaluation metric. It first generates a scene graph on the basis of object, attribute and relationship on the generated captions. In order to find the final SPICE score, it finds the semantic relationship between the caption and the reference text which is the original image title.

SPICE algorithm can be applied on both large as well as small datasets unlike CIDEr. SPICE scores are easy to understand as they are bound between 0 and 1.

$$SPICE(c, S) = \frac{2 \cdot P(c, S) \cdot R(c, S)}{P(c, S) + R(c, S)}$$

VI. EXPERIMENTAL RESULTS

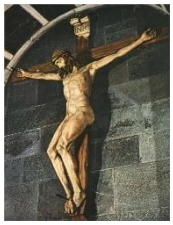

Fig. 11

Fig.12

Fig.13

Fig. 14

Fig. 15

The Neural Caption Generator generates three most probable captions for the image presented to it. The three most

probable captions generated for a few test cases which are the above figures have been provided along with their corresponding SPICE, CIDEr, METEOR, ROUGE and BLEU scores.

Captions for image 3431274821_**crucifixion**.jpg(Fig.11)

1. **crucifixion** (p=0.031974)
2. crucifixion (p=0.020127)
3. christ on the cross adored by two donors(p=0.011800)

Captions for image **madonna.**jpg(Fig. 12)

1. **madonna with child** (p=0.015410)
2. st augustine polyptych detail (p=0.007938)
3. madonna enthroned with saints (p=0.005588)

Captions for image josette.jpg(Fig. 13)

1. **painting by Josette** (p=0.045931)
2. untitled (p=0.005119)
3. untitled (p=0.004973)

Captions for image **flag**.jpg (Fig. 14)

1. **flag** (p=0.013674)
2. untitled (p=0.008948)
3. untitled (p=0.006213)

Captions for image **modern_art.**jpg(Fig. 15)

1. **modern art** (p=0.028295)
2. the pact of culture 1931 (p=0.008227)
3. the packet of coffee 1914 (p=0.004459)

EVALUATION SCORES:

SCORES	Fig.11	Fig.12	Fig.13	Fig. 14	Fig.15
SPICE	1.000	0.803	1.000	0.827	1.000
CIDEr	1.000	0.449	1.000	0.336	1.000
METEOR	1.000	0.346	1.000	0.320	1.000
ROUGE	1.000	0.338	1.000	0.311	1.000
BLEU	1.000	0.327	1.000	0.302	1.000

The Neural Caption Generator has also been aided by an audio generation application which reads out the generated caption. This will expand the horizons for visually impaired individuals who are interested in arts. They can also get involved in various research in works of art.

VII. CONCLUSION AND FUTURE WORK

The Neural Caption Generator thus produces self-explanatory and accurate captions for the images which provide better understanding and also reduce physical work. As the captions are also generated by audio, blind people can also be a part of art or sculptures. The evaluation scores and the accuracy of the produced captions prove the elegance of this algorithm.

The proposed algorithm can hence be applied to other datasets of greater use. It can help generate captions of real-time environment by treating it as an image and hence read out sign boards and traffic lights to aid blind as well as general drivers and pedestrians to avoid accidents.

REFERENCES

[1] A. Aker and R. Gaizauskas. Generating image descriptions using dependency relational patterns. In ACL, 2010.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. arXiv:1409.0473, 2014.

[3] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In EMNLP, 2014.

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

[5] D. Elliott and F. Keller. Image description using visual dependency representations. In EMNLP, 2013.

[6] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In ECCV, 2010.

[7] R. Gerber and H.-H. Nagel. Knowledge representation for the generation of quantified natural language descriptions of vehicle traffic in image sequences. In ICIP. IEEE, 1996.

[8] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In ECCV, 2014. [9] A. Graves. Generating sequences with recurrent neural networks. arXiv:1308.0850, 2013.

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

[11] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. JAIR, 47, 2013.

[12] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In arXiv:1502.03167, 2015.

[13] A. Karpathy, A. Joulin, and L. Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. NIPS, 2014.

[14] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. In arXiv:1411.2539, 2014.

[15] R. Kiros and R. Z. R. Salakhutdinov. Multimodal neural language models. In NIPS Deep Learning Workshop, 2013.

[16] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating simple image descriptions. In CVPR, 2011. [17] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi. Collective generation of natural image descriptions. In ACL, 2012.
[18] P. Kuznetsova, V. Ordonez, T. Berg, and Y. Choi. Treetalk: Composition and compression of trees for image descriptions. ACL, 2(10), 2014.
[19] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi. Composing

simple image descriptions using web-scale n-grams. In Conference on Computational Natural Language Learning, 2011.

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Com- ´ mon objects in context. arXiv:1405.0312, 2014. [21] J. Mao, W. Xu, Y. Yang, J. Wang, and A. Yuille. Explain images with multimodal recurrent neural networks. In arXiv:1410.1090, 2014.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In ICLR, 2013.

[23] M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. C. Berg, K. Yamaguchi, T. L. Berg, K. Stratos, and H. D. III. Midge: Generating image descriptions from computer vision detections. In EACL, 2012.

[24] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In NIPS, 2011.

[25] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. BLEU: A method for automatic evaluation of machine translation. In ACL, 2002.

[26] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. Collecting image annotations using amazon's mechanical turk. In NAACL HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pages 139– 147, 2010.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014.

[28] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229, 2013.

[29] R. Socher, A. Karpathy, Q. V. Le, C. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. In ACL, 2014.

[30] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In NIPS, 2014. [31] R. Vedantam, C. L. Zitnick, and D. Parikh. CIDEr: Consensus-based image description evaluation. In arXiv:1411.5726, 2015.

[32] B. Z. Yao, X. Yang, L. Lin, M. W. Lee, and S.-C. Zhu. I2t: Image parsing to text description. Proceedings of the IEEE, 98(8), 2010.

[33] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In ACL, 2014.

[34] W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. In arXiv:1409.2329, 2014.

[35] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan. Show and Tell : A Neural Image Caption Generator. arXiv : 1411.4555, 2014