CNN LSTM Pipeline for Image Caption Generation

Nikhil Khatri
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
nkhatri2@illinois.edu

Pradeep Kumar Thiagu
School Of Information Sciences
University of Illinois at Urbana-Champaign
pthiagu2@illinois.edu

Abstract

In this project, we discuss an IOT system which involves Image caption generation.We obtain sensor data from cameras and analyze them.The analysis task can be broken into two parts, first, the analysis of images using feature extraction methods and second, generation of description by forming a semantic relationship between the objects in the image while maintaining language interdependencies.

For analysis of the image we make incorporate a deep convolution neural network whereas for analysis of text we use a Recurrent Neural Network. The two parts are then linked together by mapping the embeddings generated by the deep convolution network and the recurrent network.We test our model on the Flickr30k dataset and present the result.

1. Introduction and Background

Video and Image Analytics have emerged as one of the most important applications of IOT.This involves application of machine learning algorithms to images and video feeds.One of the most common uses for video analytics are enhancing public safety,increasing employee productivity and improving maintenance.

Image caption generation has emerged as a challenging and important research area following advances in statistical language modeling and image recognition.[5] The generation of captions from images has various practical benefits, ranging from aiding the visually impaired, to enabling the automatic and cost-saving labeling of the millions of images uploaded to the Internet every day. The field also brings together state-of-the-art models in Natural Language Processing and Computer Vision, two of the major fields in Artificial Intelligence.[5]

There are two main approaches to Image Captioning: bottom-up and top-down.[8] Bottom-up approaches generate items observed in an image, and then attempt to combine the items identified into a caption. Top-down approaches, attempt to generate a semantic representation of an image that is then decoded into a caption using various architectures, such as recurrent neural networks. The latter approach follows the footsteps of recent advances in statistical machine translation, and the state-of-the-art models mostly adopt the top-down approach.

Before we delve in the details, let us revisit an important paper that has been cited thousands of times in the past years.Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[3], this paper was turned about to be very important in the deep learning research community. They proposed a novel neural network based on Recurrent neural network Encoder Decoder. One RNN used to encode a sequence of symbols (variable) into a fixed length vector representation and the other decodes the representation into another sequence of symbols (variable). The encoder and decoder of the proposed model are jointly trained to maximize the conditional probability of a target sequence given a source sequence. They applied this model to the task of translating from English to French and found that the RNN Encoder Decoder is better at capturing the linguistic regularities. The author explains in short about RNNs. What we understand are that RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations. This introduces the concept of memory because we are taking the previous output in account to take the current decision. RNN read each symbol of the input sequence sequentially and as it does that, the hidden state of RNN changes. At the end of each sequence is the end of sequence symbol after which RNN is a summary of the whole input sequence. This is what today we refer to as embedding. In the decoding part, the output sequence is produced by predicting the next symbol given the hidden state.

The paper also talks about a new type of hidden unit which is very similar to LSTM but it has just two gates instead instead of four which they claim can capture short as well as long term dependencies. The advantage of being simpler and easy to compute. The next part of the paper talks about machine translation. The author define the goal of a Statistical Translation Model (SMT) that the goal of the system (decoder, specifically) is to find a translation f given a source sentence e. The performance of a statistical machine translation system is empirically found to improve by using the conditional probabilities of phrase pairs computed by the RNN Encoder-Decoder as an additional feature in the existing log-linear model. Qualitatively, they show that the proposed model learns a semantically and syntactically meaningful representation of linguistic phrases.[3]

2. IOT Architecture

We propose an IOT system to be a smart app or smart glass device which helps in captioning images.The proposed IOT system consists of device(Smart glass),sensor(Camera),Network(LTE) and Computing Element(Cloud processing).

3. Business Canvas

The customer segment is more niche and cost structure depends on the marketing,sales and service fees for the app or product.Software Development and problem solving are the key activities for this type of business.The value proposition depends on the design and accessibility of the system.As the key resources are physical and financial,the main revenue streams for this type of business are through sales or subscription fees.

4. Data Privacy

These are the data privacy laws for the proposed IOT system:

TRUST: Certifications can be obtained across service vendors. We adopt a zero trust approach in the network with respect to security compromises.

TRANSPARENCY: Organize Campaign to raise security awareness within the organization and customers. Privacy Disclosure forms are given to the customers

CONTROL: Customer can have control over their identity data. Customer can exercise their security and individual rights.

VALUES: Storing data to enhance prediction process. Surveys from customers can help improve the business model.

5. Method

5.1. MODEL

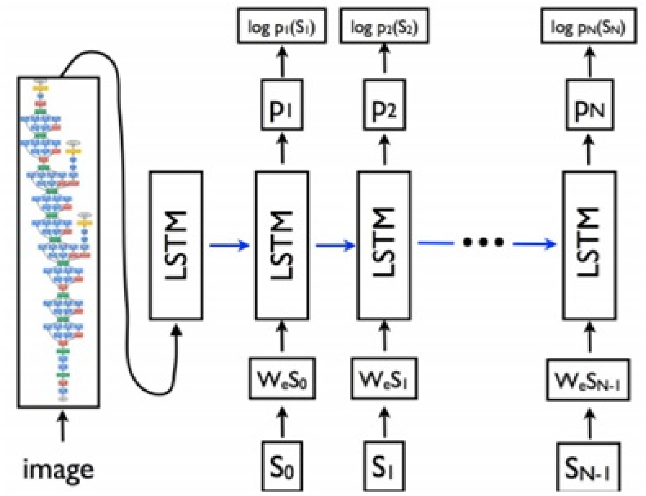

Figure 1. End to End framework

We explore a neural framework to generate descriptions from images. For a given sequence model, we maximize the probability of the correct translation given an input sentence in an " end-to-end" fashion - both for training and inference.

Similar to the machine translation problem where the model makes use of a recurrent neural network which encodes the variable length input into a fixed dimensional vector, and uses this representation to decode it to the desired output sentence., given an image, one applies the same principle of translating it into its description. Therefore, we maximize the probability of the correct description given the image by using the formulation as follows:

$$\theta^* = argmax_\theta \sum_{I,S} \log p(S|I;\theta) \qquad (1)$$

where θ are the parameters of our model, I is an image, and S its correct transcription.

Consider 'S' to depict a sentence where its length is unbounded. Thus we can apply chain rule to the model for joint probability over {S0, . . . , SN}, where N is the length of this particular example as

$$\log p(S|I) = \sum_{t=0}^{N} \log p(S_t|I, S_0, ..., S_{t-1}) \qquad (2)$$

where we dropped the dependency on θ for convenience.

During training, (S, I) is a training example pair and we optimize the sum of the log probabilities as described in equation (2) over the entire training set using stochastic gradient descent. As the input and output both comprise of a set of sequences, it is natural to model p(St—I, S0, . . . , St-1) with a Recurrent Neural Network (RNN), where the variable number of words we condition upon up to t - 1 is expressed by a fixed length hidden state or memory ht. This memory is updated after seeing a new input xt by using a non-linear function f:

$$h_{t+1} = f(h_t, x_t) \tag{3}$$

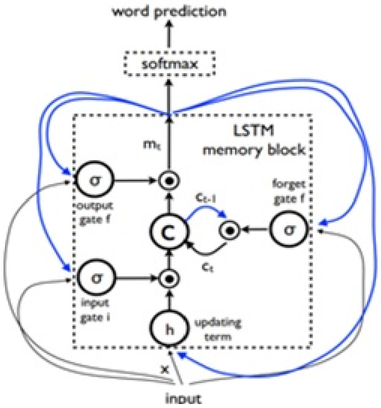

Figure 2. LSTM Cell

For f we use a Long-Short Term Memory net (to deal with the vanishing/exploding gradient problem), which has shown state-of-the art performance on sequence tasks such as translation.[9] For the representation of images, we use a Convolution Neural Network (CNN) which have been widely accepted for computer vision, and are currently state-of-the art for object recognition and detection. Particularly we use a VGG 16 (with batch normalization) model pre-trained on IMAGENET dataset from the ILSVRC 2014 classification competition.

5.2. TRAINING

The LSTM is trained to predict each word of the sentence and is defined by P($St|I$, S_0, . . . , S_{t1}). Thus, we consider the LSTM in an unrolled form i.e. a copy of the LSTM memory is created for the image and each sentence word such that all LSTMs share the same parameters and the output mt1 of the LSTM at time t $-$ 1 is fed to the LSTM at time t. All recurrent connections are transformed to feed-forward connections in the unrolled version.

To represent the previous word S_{t1} as input to the decoding LSTM producing S_t, we incorporate word embedding vectors, which have the advantage of being independent of the size of the dictionary (contrary to a simpler one hot-encoding approach).

The procedure can be summarized as follows:

$$x_{-1} = CNN(I) \tag{4}$$

$$x_t = W_e S_t, t \in 0....N-1 \tag{5}$$

$$p_{t+1} = LSTM(x_t), t \in 0....N-1 \tag{6}$$

where S_t represents one hot encoding of words and with a dimension equal to the size of the dictionary. The sentence is preceded by a special token known as the start token.

Both the image and the words are mapped together, the image by incorporating a vision CNN and the words by incorporating a word embedding We. Also we use the following loss function which is defined as the sum of the negative log likelihood of the correct word at each step as follows:

$$L(I,S) = -\sum_{t=1}^{N} \log p_t(S_t) \tag{7}$$

The above loss is minimized with respect to all the parameters using stochastic gradient descent.

5.3. Inference Techniques

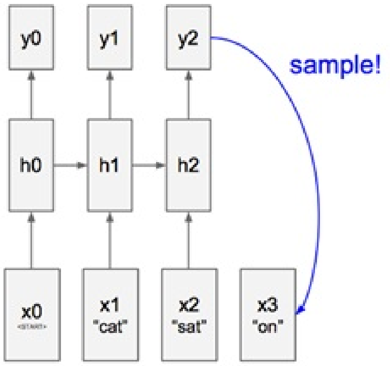

Figure 3. Sampling Inference

There are multiple approaches that can be used to generate a sentence given an image.

The first one is Sampling where we just sample the first word according to p1, then provide the corresponding embedding as input and sample p2, continuing like this until we sample the special end-of-sentence token or some maximum length. The second one is Beam Search[7]: iteratively consider the set of the k best sentences up to time t as candidates to generate sentences of size t + 1, and keep only the resulting best k of them.

5.4. Embeddings

To represent the previous word S_{t1} as input to the decoding LSTM producing S_t, we incorporate word embedding vectors, which have the advantage of being independent of

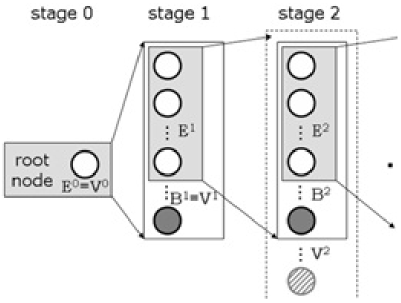

Figure 4. Beam Search

the size of the dictionary (contrary to a simpler one hot-encoding approach). Moreover, these embeddings can be jointly trained with the rest of the model.

6. Experiment

6.1. Datasets

For evaluating our model, we used the publicly available Flikr30 dataset(sensor data) which consists of thirty- thousand images and sentences in English describing these images. The dataset is readily distributed by the University of Illinois at Urbana Champaign. The dataset contains a set of 5 annotations for each image that are relatively visual and unbiased. The table below demonstrates the train-test split.

6.2. Implementation

In this section we include the details about our implementation and our training process. First of we start by using the weights of the pretrained VGG 16 model and we use these along with image embeddings of the dataset on VGG16. Before passing the inputs to the LSTM we preprocess the input and randomly shuffle our features. We then update our image and word embeddings by multiplying the vector obtained from passing the input image and the image embedding. For the first iteration, the image embedding is passed directly to the word embedding as a start token. Subsequently, we use an embedding lookup function such that the word embedding at time t=i looks for the word from t= i-1 iterating over the number of hidden layers in our LSTM. The one hot encoding of the word vector is taken and fed to a cross entropy loss function. It was noticed that Adam Optimizer along with an exponential decaying learning rate worked best among all the optimizers and gave us a stable result. The model was trained on NCSAs Bluewater supercomputer for 150 epochs

6.3. Results

Given below are a few results obtained after training the model for 150 epochs. We observe that although our captions are simplistic in nature and often wrong (in the failure cases) they still managed to capture the objects depicted in the image. Most of the failure cases demonstrate sufficient capacity to detect images but fail in expressing the semantic relationships of these objects.

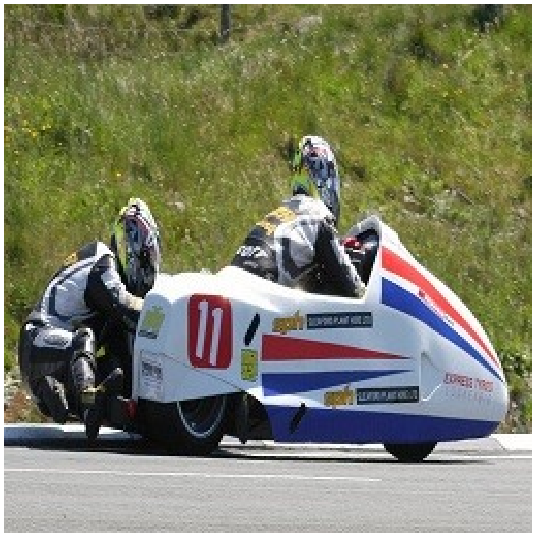

Figure 5. A man in a red shirt and a helmet is riding a bike

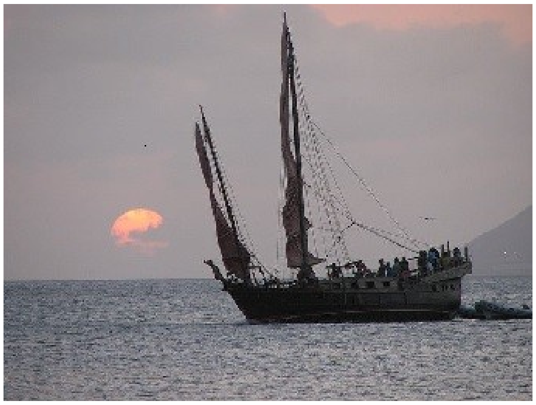

Figure 6. a man is standing on a boat in the ocean

7. Conclusion

The paper describes about the implementation of an IOT system ,the business process and its associated data privacy policies.We implemented a model which involves training a CNN-LSTM model on flickr-30k dataset(sensor data) and successfully demonstrated image captioning.

7.1. Evaluation Metrics

The problem with generating captions from images is that the text generated cannot be quantified or measured as a quantitatively, i.e. there is no way to measure the accuracy of these generated text. One way is to evaluate them by humans, but humans bring their own bias and subjectivity and

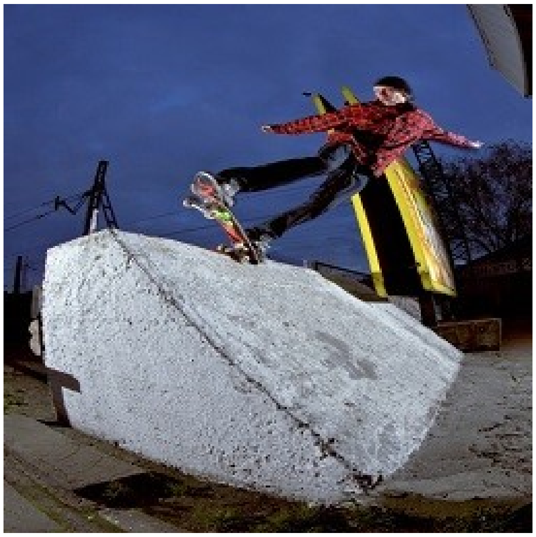

Figure 7. A man is standing on a rock

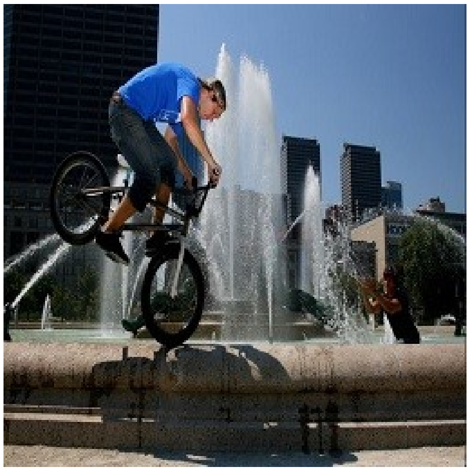

Figure 8. A man in a blue shirt is sleeping on the bench

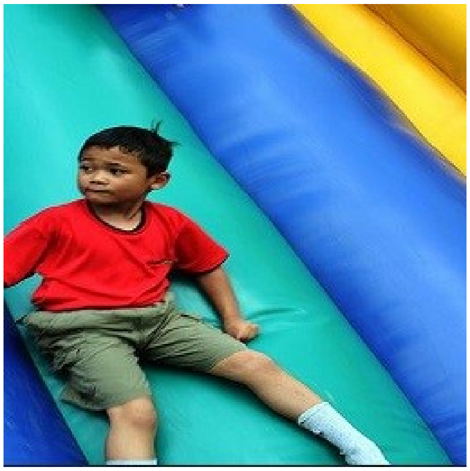

Figure 9. A man in a red shirt is riding a motorcycle

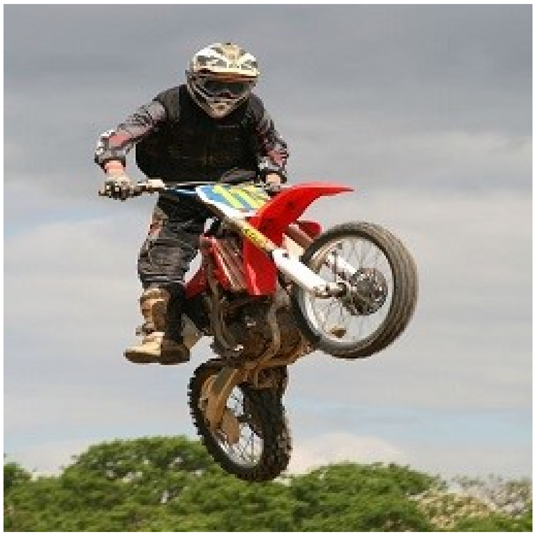

Figure 10. a young girl in a pink shirt is sliding down a slide

often fail to agree on details. Thus a lot of research has been going on improving these metrics so that we can effectively measure the performance of our models.

1. BLEU [4] This evaluation metric was proposed by Papineni et al. which analyses the co-occurrences of n-grams between candidate and reference sentences. However, this model suffers from large variations of scores in sentences and has thus been shown as an insufficient evaluation metric.

2. METEOR [2] This was proposed by Banerjee & Lavie which involves a more flexible MT metric that calculates sentence-level similarity scores as a harmonic mean of unigram precision based on Stemmed tokens and Paraphrasing.

3. CIDEr [6] This was proposed by Vedantam et. al which automatically evaluates for an Image Ii how well a sentence ci matches the consensus of a set of image descriptions S. Intuitively, the measure should encode how often n-grams in the candidate sentence are present in the reference sentences. Also, n-grams that commonly occur across all images in the dataset should be given lower weight (as they are less likely to be informative). In practice, this method performs a Term Frequency Inverse Document Frequency.

4. SPICE [1] The previous evaluation metrics are all sensitive to n-gram overlap and this metric proposes that semantic propositional content should be an important component of evaluation.

References

[1] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. 2016.

[2] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. 2005.

[3] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[4] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. 2002.

[5] M. Soh. Learning cnn-lstm architectures for image caption generation.

[6] R. Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. 2015.

[7] A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee1, D. Crandall, and D. Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. 2016.

[8] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555, 2014.

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