Handwriting recognition

you

September 10, 2017

1 Problem statement

Handwriting recognition is a task of interpreting handwritten inputs from sources such as image of text written on a paper, touch screen, electronic whiteboard or other devices. This task is subdivided among two categories off-line recognition and on-line recognition. on-line recognition, off-line recognition, degraded text
https://www.overleaf.com/10991049vdhtwfqssvym#/41384505/

2 Introduction

- Study of various approaches for Image/video caption generation, multiview learning, Encoder-decoder framework, attention model, coverage
- Extending LSTM based approach by introducing CCA based cost function at the output layer.

3 Dataset

3.1 IAM

Movie Description dataset: Figure 1 shows a MPII dataset sample.

RIMES It contains video plus audio clips, i.e., short sequences from 32 movies. The video frames typically consist of 240 lines, the aspect ratios vary. Each clip is annotated according to 8 classes: AnswerPhone, GetOutCar, HandShake, HugPerson, Kiss, SitDown, SitUp, StandUp. Dataset is 2.2gb in size.

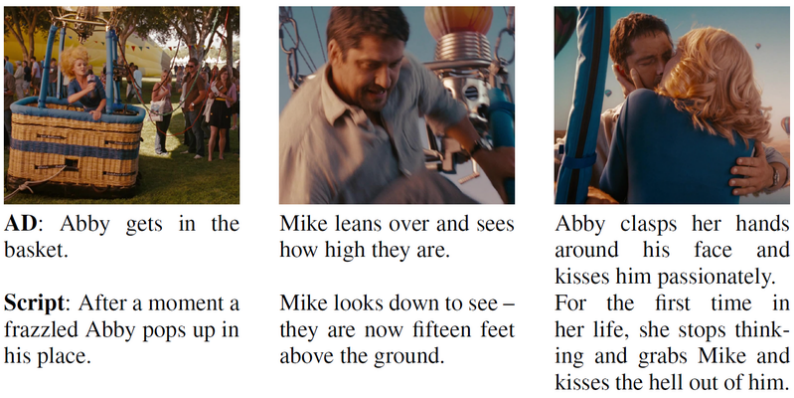

Figure 1: An example of MPII dataset showing audio description and script.

Dataset	Classes	Videos	Evaluation metric	Properties
Max Plank II movie description dataset		68k clips	BLEU, METEOR	video plus text
Montreal Video Annotation dataset		49k clips	BLEU, METEOR	video plus text
Hollywood: human action from movies	8	430	classification accuracy	video plus audio
Hollywood2: human action from movies	12	3669	classification accuracy	video plus audio

Table 1: Movie dataset

Figure 2: MSCOCO dataset. source mscoco.org/dataset

3.2 UW3 dataset

The flickr8k dataset contains 8000 images along with 5 captions per image. It takes 700Mb of disk space. The flickr30k dataset contains 30000 images along with 5 captions per image. Microsoft Common Objects in Context (COCO) dataset [10] is a new image recognition, segmentation, and captioning dataset. The MS COCO caption dataset contains human generated captions for the images. It comprises 82783 training images, 40504 validation images, and 40775 test images. Each image is accompanied by at least five captions of varying length. An image from MS-coco dataset is shown in figure2 and from the flicker8k dataset is shown in figure 3.

Dataset	Number of Images	Number of captions per Image
MSCOCO	160k	5
flickr30k	30k	5
flickr8k	8k	5

Table 2: Image captioning datasets

3.3 Qutar dataset

VidTIMIT dataset http://conradsanderson.id.au/vidtimit is a multiview dataset of people speaking consisting of both audio and video. This dataset has obvious temporal properties and thus we expect DCCA extendend with LSTMs should achieve strong results.

Bouncing MNIST dataset a variation of the MNIST dataset that adds a temporal component where the MNIST digits move and "bounce" off the boundary edges in a 64*64 square.

Wisconsin X-Ray Microbeam (XRMB) dataset of articulatory data and recorded speech, is that fixed-size input must be used.

4 Background

Below, we discuss a few concepts that are relevant to the idea we are working upon.

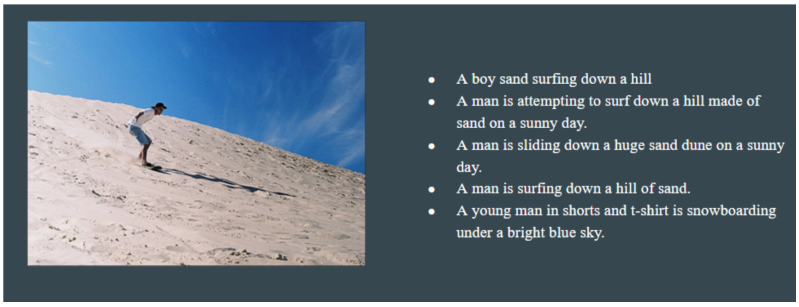

Figure 3: An example of flickr8k dataset showing an image along with its caption

4.1 RNN

Traditional Neural networks(NNs) assume all inputs or outputs are independent of each other. They perform quite well on some problems such as classification, regression but lack the ability to capture sequential information in the data. RNN is an extension of traditional NN for sequence. It can memorize sequential features of data [21]. In RNN the current output of RNN depends on the previous computations and current input. It is shown by equation 1 , where h_{i-1} is previous computation, x_i is current input and g_Q and f_Q are non-linear transformation functions.

$$\begin{aligned} h_i &= g_Q(h_{i-1}, x_i) \\ y_i &= f_Q(h_i, x_i) \end{aligned} \tag{1}$$

for caption generation, x_i represents previous generated word y_{i-1}, hence the equation becomes

$$y_i = f_Q(h_i, y_{i-1}) \tag{2}$$

One major disadvantage of RNN is that it have vanishing gradient or exploding gradient problem, it makes the gradient of weight from a far-away step hard to train. Compared to the typical RNN model, LSTMs have an additional cell state, c_t. It creates a path for the flow of gradient inside each time step. This cell state gives the ability to LSTM to selectively read input to update the hidden state. Hence an LSTM network is well-suited for long time series data. A general LSTM structure (figure 4) and equations are given below.

$$\begin{aligned} i_t &= g_{iQ}(h_{t-1}, x_t) \\ f_t &= g_{fQ}(h_{t-1}, x_t) \\ o_t &= g_{oQ}(h_{t-1}, x_t) \\ \overline{c_t} &= g_{\bar{c}Q}(h_{t-1}, x_t) \\ c_t &= g_{cQ}(\overline{c_t}, i_t, f_t, c_{t-1}) \\ h_t &= g_{hQ}(o_t, c_t) \end{aligned} \tag{3}$$

where i_t, f_t, o_t are input, forget and output gates respectively, input gate controls the amount of current information to consider for update, forget gate controls the amount of information to forget for previous cell state and output gate controls the amount of information to transfer for cell state to hidden state. $\overline{c_t}$ is intermediate cell state.

4.2 Encoder–Decoder Network

The RNN encoder-decoder is a neural network model that maximizes the conditional probability of correct output sequence given the input sequence [22] as shown in equation 3, where Q is a model parameter, TO and TI is output and input sequence length respectively.

$$agrmaxQ : p(y_0, y_1 ..., y_{TO} | x_0, x_1, ..., x_{TI}, Q) \tag{4}$$

For for video caption generation, the input is usually a sequence of video frame feature vector, **instead of video frame feature vector we can take video feature vector embedding** while the output is a sequence of word embedding.

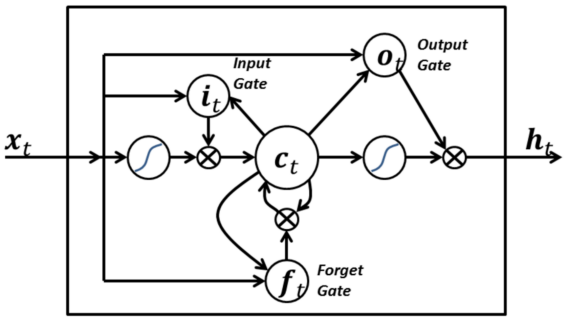

Figure 4: A LSTM block with input, output, and forget gates. source [30]

From chain rule, an equation can be decompose as show below.

$$p(w_1, w_2, \ldots, w_T) = p(w_1) * p(w_2/w_1) * p(w_3/w_1, w_2) * \cdots * p(w_T/w_1, w_2, ..., w_{T-1})$$
$$p(w_1, w_2, \ldots, w_T) = \prod_{i=1}^{T} p(w_i/w_{1:i}) \tag{5}$$

where

$$w_{1:i} = (w_1, w_2, ..., w_{i-1}) \tag{6}$$

Hence equation 3 can be decompose using chain rule into below form:

$$p(y_0, y_1..., y_{TO}|x_0, x_1, ..., x_{TI}, Q) = \prod_{i=0}^{TO} p(y_i/y_{1:i}, x_0, x_1, ..., x_{TI}, Q) \tag{7}$$

Generally, the encoder maps the input sequence into a fixed hidden length representation c, also known as context vector, as shown below:

$$c = E(x_0, x_1, ..., x_{TI}) \tag{8}$$

c is passed as input to the decoder, hence equation 7 simplifies to equation 9

$$p(y_0, y_1..., y_{TO}|x_0, x_1, ..., x_{TI}, Q) = \prod_{i=0}^{TO} p(y_i/y_{1:i}, c, Q) \tag{9}$$

In the encoder-decoder framework, with recurrent neural network (RNN), conditional probability is modeled as below, where h_i denotes the output of a recurrent hidden layer f($\cdot$) with inputs y_{i-1} and h_{i-1}. g($\cdot$) is a non-linear function with inputs y_{i-1}, h_i, and c. This equation shows that generation of target word depends on both source context(c) and target context (h_{i-1}, y_{i-1}).

$$p(y_i/y_{1:i}, c, Q) = g_Q(h_i, y_{i-1}, c)$$
$$h_i = f_Q(h_{i-1}, y_{i-1}) \tag{10}$$

For example, in the case of image caption generation, the decoder generates one word y_i at each time step conditioned on the context vector c, decoder hidden states h_i and previously generated word y_{i-1}. An encoder-decoder framework is illustrated in figure 5.

4.3 Attention model

In video captioning system, for a long video with lots of objects and attributes it becomes difficult for vanilla encoder-decoder model to capture all the information. One approach to solve this problem is to initially go through the complete video but then focus to different temporal parts(clips) in the video while describing the video. This feature is known as attention model. Attention network is a model which selects a part of the information from total information at every time step. Since attention network avoids the need to represent input sequence with a single vector, it gives the model an opportunity to use intermediate encoder hidden states instead of using only the final encoder output as shown in Figure 4. They have been proved successful for problems in which there is an implicit relation between input and output sequence.

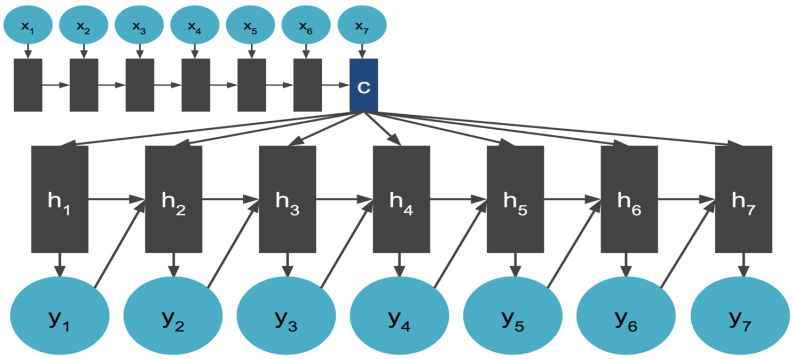

Figure 5: RNN Encoder-Decoder.

They have shown significant improvement in the task of machine translation[9], image captioning[3,6,7], video captioning[31,37] and speech recognition. For example in machine translation, an encoder, at every time step encodes the information of the sentence with focus on current word in its hidden state. A decoder then searches for a set of relevant positions in the source sentence using attention mechanism. This system is then jointly trained to maximize the probability of a correct translation given a source sentence. In comparison to utilizing only the output of the encoder to predict the new word, the decoder in this model utilizes all encoder hidden states using attention mechanism. This improves the encoder capacity to express long input sentences.
Typically, an attention model is a single layer neural network with $x_0, x_1, ..., x_{TI}$ or its hidden representation $Ih_0, Ih_1, ..., Ih_{TI}$ as input and context vector c_i as output. At each time step i, the attention model finds locations $L_{i,j}$ in the input that are relevant to the previous hidden state h_{i-1} in decoder RNN, as shown below, where j is location index and i is time step, g is a similarity function with decoder previous hidden state (h_{i-1}) and encoder feature vector for jth index (Ih_j) as inputs. Hence, the attention mechanism finds the similarity between previous hidden state h_{i-1} and encoder j^{th} hidden state (Ih_j). **why we are using tanh function to find similarity instead of dot product**

$$\begin{aligned} a_{ij} &= p(L_{i,j}/h_{i-1}) = g(h_i, Ih_j) \\ a_{ij} &= softmax(v * tanh(W^{ah} h_{i-1} + W^{ae} Ih_j)) \end{aligned} \tag{11}$$

The context vector ci is computed as a weighted sum of the encodings of x_i.

$$c_i = \sum_{j=1}^{TI} a_{ij} Ih_j \tag{12}$$

$$\alpha_{ij} = g(t_{i-1}, h_j, c_{i-1,j}) \tag{13}$$

$$C_{i,j} = f(C_{i-1,j}, \alpha_{i,j}, h_j, t_{i-1}) \tag{14}$$

In the attention based encoder–decoder model, conditional probability in equation 7 is defined below. Unlike the conventional encoder–decoder approach here the probability is conditioned on a distinct context vector ci for each target word yi.

$$p(y_i/y_{<i}, x_0, x_1, ..., x_{TI}, Q) = g_Q(h_i, y_{i-1}, c_i) \tag{15}$$

4.4 DCCA

Canonical Correlation Analysis (CCA) is a technique of learning a latent subspace (R^{dz}) from multiple views (R^{Dx}, R^{Dy}) of the same underlying signal [1]. In this subspace, projection of each view is maximally correlated with each other. Assuming noise in one view is uncorrelated with the noise in other view, multiview learning helps in learning a representation of each view which uses complementary information of another view. Given N pair of data points of two views, $(x_i, y_i)_{i=1}^N \in (R^{Dx}, R^{Dy})$, CCA learns a pair of linear projections (U,V) that maximizes the correlation of two

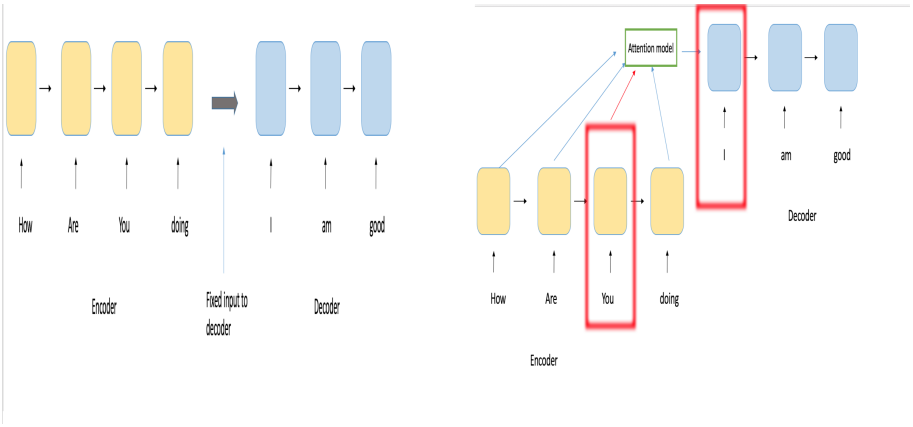

Figure 6: Attention based Encoder-decoder framework.

views. This objective function and constraints are shown in the equation 12, where X,Y are datasets such that $X = [x_1, x_2, ..., x_N]$ and $Y = [y_1, y_2, ..., y_N]$, $(U, V) \in (R^{Dx \times dz}, R^{Dy \times dz})$.

$$\begin{aligned} \max_{U,V} \quad & \frac{1}{N} tr(U^T X Y^T V) \\ s.t. : & U^T U = I_{dz},\ V^T V = I_{dz} \end{aligned} \tag{16}$$

DCCA is an extension of CCA in which data $(x_i, y_i) \in (R^{Dx}, R^{Dy})$ is non-linearly transformed to a new space (R^{dx}, R^{dy}) through deep neural networks. The weights of the neural networks are jointly trained with the objective to maximize the correlation between projected representations of both views. equation 13 shows DCCA formulation

$$\begin{aligned} \max_{W_f, W_g, U, V} \quad & \frac{1}{N} tr(U^T F G^T V) \\ s.t. : & U^T U = I_{dz},\ V^T V = I_{dz} \end{aligned} \tag{17}$$

where $F = [f(x_1), f(x_2), ..., f(x_N)]$ and $G = [g(y_1), g(y_2), ..., g(y_N)]$,(U,V) $\in (R^{dx \times dz}, R^{dy \times dz})$ and (f,g) are non-linear neural network based transformation functions with weights W_f, W_g respectively. The solution of equation 13 is shown below in equation 14.

$$(U, V) = C_{FF}^{-1/2} U^{hat}, C_{GG}^{-1/2} V^{hat}, \tag{18}$$

where U^{hat} and V^{hat} are first k left and right singular vectors of $C_{FF}^{-1/2} C_{FG} C_{GG}^{-1/2}$. This solution can also be obtained via stochastic gradient descent as done in [27,25].

Deep Canonically Correlated Autoencoders (DCCAE) is an extension of DCCA where the representations outputted by the neural networks are feed forward through auto-encoders [2]. Here, the weights of all four networks are jointly trained with the objective being a trade-off between maximizing the correlation of the projected representations and minimizing the reconstruction error of the autoencoder outputs and the input data.

4.5 Variational CCA

DCCA and DCCAE have shown remarkable success in learning representations based on multiple views of data. However, a disadvantage of DCCA is that it cannot be used as a model for generating samples from the latent space. Although, DCCAE model can reconstruct second view from first view, in practice, the inputs are not reconstructed well [28]. Variational CCA extends probabilistic latent variable model generative interpretation of CCA to non linear models. The probabilistic linear latent variable model defined the joint distribution of data pair (x,y) as shown in equation 15, where z is the latent variable.

$$\begin{aligned} p(x, y, z) &= p(z)p(x|z)p(y|z) \\ p(x, y) &= \frac{1}{N} \sum_{i=0}^{N} p(x, y, z_i) \end{aligned} \tag{19}$$

We have access to the joint distribution $(x_i, y_i)_{i=1}^n$ through n observations, which comprise our training data. For non-linear latent variable model the conditional distribution is modeled by DNNs as $p_\theta(x|z;\theta_x)$ and $p_\theta(y|z;\theta_y)$, where (θ_x, θ_y) are DNN parameters. The joint distribution $p_\theta(x, y)$ and the inference problem $p_\theta(z|x)$ are intractable in this form. To obtain the joint distribution $logp_\theta(x, y)$, derive a variational lower bound $L(x, y; \theta, \phi)$ as shown below.

$$L(x, y; \theta, \phi) = -D_{KL}(q_\phi(z|x)||p(z)) + E_{q(z|x)}[logp_\theta(x|z) + logp_\theta(y|z)] \tag{20}$$

where $q_\phi(z|x;\phi_z)$ is the approximate posterior distribution of the latent variable $p(z/X)$. It is also modeled by a DNN, for example assuming a Gaussian distribution, for a given input (x_i, y_i) the posterior distribution $q_\phi(z|x_i)$ can be obtained as an output of DNN as shown in equation 17, where $[\mu_i, \Sigma_i] = f(x_i;\phi_z)$ are deterministic nonlinear functions of x_i and network parameter ϕ_z.

$$log\, q_\phi(z_i|x_i) = logN(z_i; \mu_i, \Sigma_i), \tag{21}$$

For a given x_i, we can obtain the distribution $log\, q_\phi(z_i|x_i)$ and from monte-carlo sampling we can draw L samples of z_i.

$$\begin{aligned} z_i^l &= \mu + \phi * \epsilon^l \\ \epsilon^l &\in N(0, I), for\, l = 1, ..., L, \end{aligned} \tag{22}$$

hence ,

$$E_{q_\phi(z|x)}[logp_\theta(x|z) + logp_\theta(y|z)] \approx \frac{1}{L}\sum_{l=0}^{L}[logp_\theta(x_i|z_i^l) + logp_\theta(y_i|z_i^l)] \tag{23}$$

$logp_\theta(x_i|z_i^l)$ and $logp_\theta(y_i|z_i^l)$ measure the reconstruction errors of x_i and y_i from input x_i.
VCCA maximizes this variational lower bound on the data likelihood on the training set, where assuming gaussian conditional probability $p_\theta(x_i|z_i^l)$ with parameters $[\mu_{xi}, \Sigma_{xi}] = g_x(z_i;\theta_x)$ is function of z_i and DNN parameters θ_x and $[\mu_{yi}, \Sigma_{yi}] = g_y(z_i;\theta_y)$ is function of z_i and another DNN parameters θ_y.

$$\max_{\theta,\phi} \quad \frac{1}{N}\sum_{i=0}^{N} L(x, y; \theta, \phi) = \frac{1}{N} - D_{KL}(q_\phi(z_i|x_i)||p(z_i)) + \frac{1}{NL}\sum_{i=0}^{N}\sum_{l=0}^{L}[logp_\theta(x_i|z_i^l) + logp_\theta(y_i|z_i^l)] \tag{24}$$

Three DNNs with 4 hidden layers of 1024 rectified linear units each is used to parameterize the distributions: $q\phi(z|x)$, $p\theta(x|z)$, $p\theta(y|z)$ in VCCA. The reconstruction networks $p\theta(x|z)$ model each pixel of x as an independent Bernoulli variable and parameterize its mean (using a sigmoid activation).

5 Related Work

5.1 Deep captioning with multimodal RNN (M-RNN) (2015)

Traditional image classification algorithms, classify images into one of 'k' classes but an image can belong to more than one class. These approaches fails to capture any relationship between class labels by constraining the category of an image to a single label. One of the alternatives to this method is to jointly train the model for image recognition with a semantic model of its captions[17], [8]. This allows the model to represent the images into a better semantic space as compared to the one defined by unit labels as before. It also allows to leverage the relationship between labels which was not possible before.

Training data is a set of image-caption pairs. Model contains a language model part, a vision part and a multimodal part. Model is jointly trained on the images and the captions using log-likelihood cost function. The images are trained on a CNN and represented as the top-most layer and the text is trained on an LSTM network where the representation of a sentence is the hidden state of LSTM at the particular timestep. Once the model is trained it is tested on images to generate captions. This is similar to the scenario of finding the missing modality (text) given the observed modality (image). M-RNN architecture is shown is figure 5.

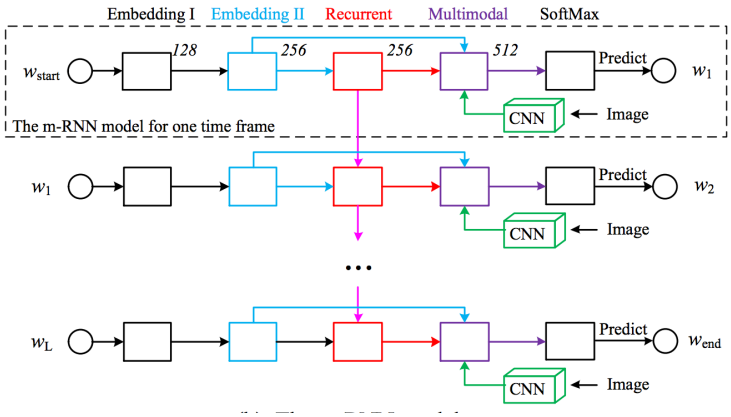

Figure 7: The inputs to the model are an image and its corresponding sentence descriptions. w1, w2, ..., wL represents the words in a sentence. The model estimates the probability distribution of the next word given previous words and the image. It consists of five layers (i.e. two word embedding layers, a recurrent layer, a multimodal layer and a softmax layer) and a deep CNN in each time frame. The number above each layer indicates the dimension of the layer. source [8]

code written in python is available at: https://github.com/mjhucla/mRNN-CR. Required libraries: numpy, scipy, nltk.

5.2 Deep correlation for matching image and text (2015)

Generally, hand crafted objective functions are applied to match image and text in a latent space. However, these objective functions are difficult to generalize in different domains and datasets. In this work, an end-to-end deep cca based framework is proposed for matching images and text in a joint latent space [25]. The architecture of the proposed network is illustrated in Figure 7. The images are trained on a CNN and represented as the top-most layer of CNN, and caption is represented as a TF-IDF feature vector. The text features were passed as input to n-layer DNN. Both image CNN and text neural networks were jointly trained with CCA objective function. The model was evaluated on ranking score on flicker8k and flicker30k dataset. Training process of this architecture is described below.

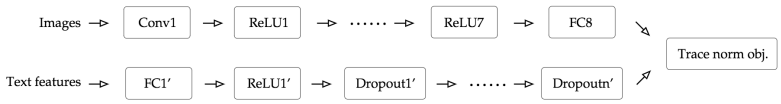

Figure 8: Architecture of the proposed model. source [25]

Let (X,Y) $\in$ ($R^{dx\times m}$,$R^{dy\times m}$) be the image and text representation at the output of final layer of neural network, where (dx,dy) are number of output neuron at the final layer of neural network. As shown in section 4.3 CCA learns a pair of linear projections (U,V) that maximizes the correlation between two view. The objective function and optimal value of U,V is shown in equation 16 and 17.

$$\begin{aligned} &\max_{U,V} \quad \frac{1}{N} tr(U^T X Y^T V) \\ &s.t. : U^T U = I_k,\ V^T V = I_k \end{aligned} \tag{25}$$

$$(U, V) = C_{FF}^{-1/2} U^{hat}, C_{GG}^{-1/2} V^{hat}, \tag{26}$$

where U^{hat} and V^{hat} are first k left and right singular vectors of T = $C_{XX}^{-1/2} C_{XY} C_{YY}^{-1/2}$. Let the singular value decomposition of T be T = UDV^T, then the gradient of total correlation with respect to X is given by:

$$\frac{\delta corr(X,Y)}{\delta X} = \frac{1}{m-1}(2\Delta_{xx}X + \Delta_{xy}Y) \tag{27}$$

where

$$\Delta_{xx} = \frac{-1}{2} C_{XX}^{-1/2} U D U^T C_{XX}^{-1/2} \tag{28}$$

$$\Delta_{xy} = C_{XX}^{-1/2} U V^T C_{YY}^{-1/2} \tag{29}$$

A GPU with 6GB of memory limits the batch size m to 100 when last layer dimension d is set to 4096. Overall each iteration for a batch of size m = 100 took approximately 26.5 seconds to complete.

Unsupervised learning of acoustic features: last layer dimension = L=112 batch size = 1000. Both have similar training batch data size.

5.3 Show and tell (2015)

Show and tell is an encoder-decoder based architecture[4]. Here Inception V3 model(encoder) is applied to input image to produce encodings that are good for object detection. RNN is used as an decoder where the vector representation of image and all previously generated words are fed as input to predict the current word. During training, log likelihood of target caption is maximized as shown in Equation 1, here S is target sentence, I is input image and Q is unknown but deterministic model parameters. Fine tuning is performed on this captioning system by jointly training its encoder and decoder components on human generated captions. This fine tuning helps encoder to learn natural description like the color of the leaves in addition to learning the objects like giraffe, tree,leaves and ground. The model architecture is explained in figure 6.

$$agrmaxQ : J(S/I;Q) = \sum_{i=1}^{T} logp(w_i/I;Q) \tag{30}$$

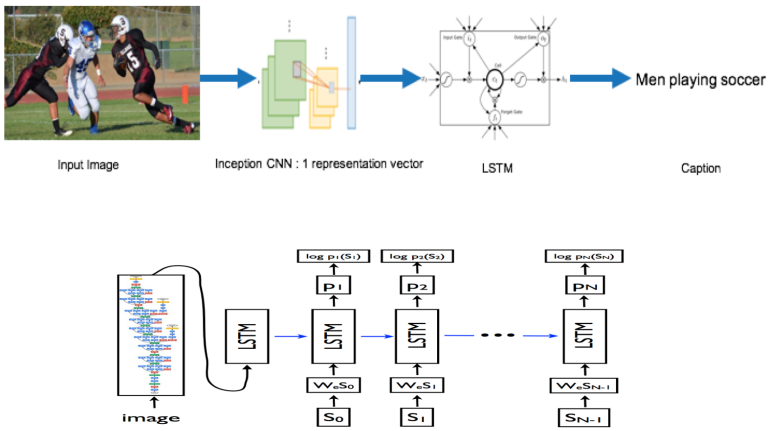

Figure 9: LSTM(decoder) model combined with a CNN image encoder. The decoder generates one word at each time step conditioned on previous hidden states and previous generated words. (below) The unrolled connections between the LSTM memories are in blue.

Results for this model were reproduced with a pretrained weights and are shown in Figure 7.

code written in chainer is available at: https://github.com/apple2373/chainer_caption_generation.

Figure 10: Result Show and Tell.

5.4 Show, attend and tell (preprint 2015)

In practice when we generate a caption for an image we also pays attention to different parts of the image. Attention mechanism is introduced in encoder-decoder framework to achieve this feature. Now, the model generates a mapping from set of features encoded by VGGnet network to hidden state of RNN at each time step. This gives the decoder ability to focus on relevant details. The attention mechanism finds a subset of the encoding and the LSTM generates a word from this encoding and previously generated words. As compared to previous approach where image is represented as a single vector, here CNN is applied at image patch instead of whole image, this allows model generate more descriptive feature vectors. A set of these feature vectors is used at each word generation using the attention mechanism. The model architecture is explained in Figure 8.

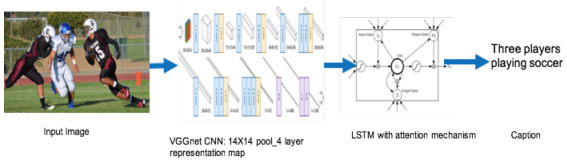

Figure 11: Features from forth(middle) layer of VGGnet model is used as image encoding. At each generation step decoder pays attention to specific set of features. It can be interpreted as probability that certain pixels are the place for producing next word. Decoder generates one word at each time step conditioned on the attention vector, the previous hidden state and previous generated words.

Results for this model were reproduced with a small dataset and are shown in figure 9.

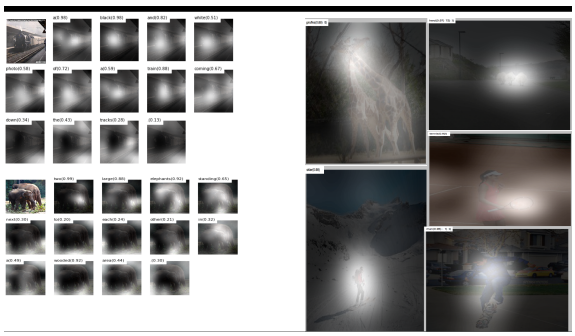

Figure 12: (Left) Caption and corresponding attention location (Right) Our model shows deeper understanding to words, for example it generates skier instead of man, herd instead of sheep, giraffes instead of giraffe.

code written in tensorflow is available at: https://github.com/yunjey/show-attend-and-tell-tensorflow.

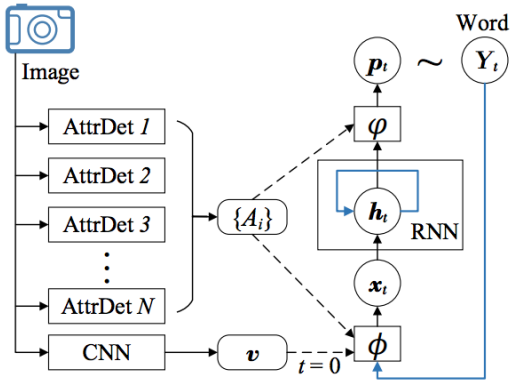

Figure 13: Visual features of CNN responses v and attribute detections Ai are injected into RNN (dashed arrows) and get fused together through a feedback loop (blue arrows). source [26]

5.5 Image Captioning with Semantic Attention (March, 2016):

Top-down approach for image captioning, starts from an image representation (feature representation) and converts it into words, whereas or bottom-up, extracts various descriptive information of an image and then combine them. In this work, a new algorithm is proposed that combines both approaches through semantic attention [26]. A global visual representation was extracted from classification CNN, in addition set of attributes were obtained from the image (descriptive information). Both these information is combined using a semantic attention model and is fed to LSTM network. This model is shown in figure 10.

5.6 Review Network for caption generation (Oct, 2016):

In a conventional attention based encoder-decoder network, at each time step the decoder pays attention to a subset of encoder hidden states. This work proposes an extension to the this framework [7]. At each time step, an attention based network reviews encoder hidden state to generate thought vectors which are more compact and global representation than encoder hidden state. The thought vector is fed as input to the attention network in decoder to generate image captions. This architecture is shown below in figure 11.

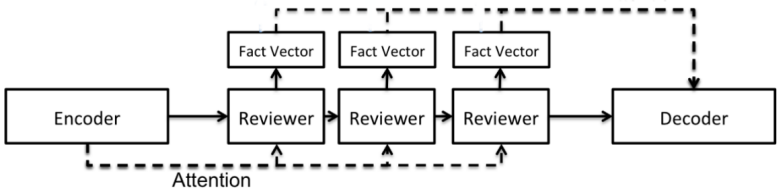

Figure 14: Review model with attention network at both encoder and decoder. source [7]

code written in Torch is available at: https://github.com/kimiyoung/review_net.

5.7 Knowing when to look (Dec, 2016)

In a traditional attention based network, generation of target word (y_i) depends on both source context(c_i) and target context (h_{i-1}, y_{i-1}). Attention based encoder-decoder frameworks for image caption generation use visual attention models (finding a subset of features from all image features), which is active for every generated word. However, for non visual words like 'of' , 'the' these visual attention features provides no information. On the contrary, it add noise in decoder input, leads to adverse effects in model performance. A novel adaptive attention model is proposed in this paper, it controls the ratio at which source and target context are used to generate a new word. It

provides a mechanism to decide when and where to look for image features [24]. To incorporate this feature in the model, vanilla LSTM architecture is extended which in addition to the hidden state, produces a visual sentinel vector. Visual sentinel is a latent representation of what the decoder already knows. This architecture is illustrated in figure 12.

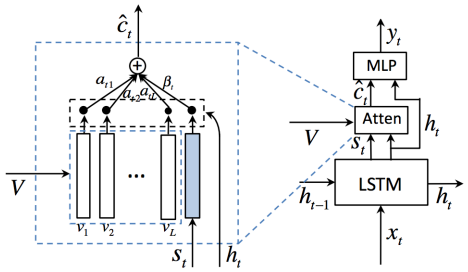

Figure 15: Proposed spatial attention model generating the t-th target word y_t given the image. Here s_t is sentinel vector, $v_1, v_2, v_3, ..., v_L$ are image features, h_t is LSTM decoder hidden state at time step t and c_t is new adaptive context vector. source [24]

Model	Dataset	Evaluation metric	Accuracy	
LCRN	MS-COCO	BLEU-4	0.306	
M-RNN	MS-COCO/flicker8K	BLEU-4/R@1	0.299/11.5	9
Show and Tell	MS-COCO	BLEU-4	0.309	
Show, attend and tell	MS-COCO	BLEU-4	0.277	
Review Network	MS-COCO	BLEU-4	0.313	
When to look	MS-COCO	BLEU-4	0.335	
Matching image and text	flicker8K	R@1	13.6	

Table 3: Performance of different models on Flickr8k and COCO test splits

6 Project Ideas

- When there is clutter in the scene it becomes very difficult for a system to generate a complex description of the image. we can learn better image features via multiview learning approach using DCCA to extract features. captions provide information about the image content, and the noise in the two views is uncorrelated and therefore filtered out by such a technique. Append the extracted features with raw data/CNN features.
- Introducing CCA based cost function in LSTMs, instead of log likelihood cost function.
- **Transfer learning:**
  - Instead of training a randomly initialized language model, building a language model from other relevant text sources like movie captions, DVS script.
  - Extracting two sets of features from images, one using CNN trained on classification dataset and other using CNN trained on video dataset.
- Fine tuning the network to retrain all encoder decoder weights.

7 Image/Video caption challenges

7.1 Large Scale Movie Description and Understanding Challenge (2016), at ECCV.

https://sites.google.com/site/describingmovies

Movie description: LSMDC16, contains short (4-5 seconds) movie clips with associated sentence descriptions. All the sentences have been manually aligned to the video. Task is for a given movie clip, generate a single sentence that describes this clip.

Multiple-Choice Test: Given a video query and 5 captions, find the correct caption for the video among 5 possible choices.

Movie Retrieval: Evaluation based on Recall@1, Recall@5, Recall@10 on 1000 video/sentence from original captions from variety of human activity categories from LSMDC2016 public-test data.

Movie Fill-in-the-Blank: Given a video clip and a sentence with a blank in it, the task is to fill in the blank with the correct word. There are almost 300k training examples, from 100k clips. In addition, a separate non-overlapping validation set with 21k examples (from 7500 clips) is provided, and evaluation will be on a test set of 30k examples (from 10k clips).

7.2 Microsoft Video to Language Challenge (2016):

http://ms-multimedia-challenge.com/challenge

Video description: Given an input video clip, generate a natural sentence to describe video content, ideally encapsulating its most informative dynamics. The dataset contains 41.2 hours and 200K clip-sentence pairs in total, covering the most comprehensive categories and diverse visual content.

7.3 Microsoft Common Objects in Context challenge at ECCV (2015):

http://mscoco.org/dataset/#overview

Caption generation: Given an image, generate a single sentence that describes this image. Dataset comprises 82783 training images, 40504 validation images and 40775 test images. Each image is accompanied by at least five captions of varying length.

	M1	M2	Total	Ranking
Google	5	4	9	1st
MSR	4	5	9	1st
Montreal/Toronto	3	2	5	3rd
MSR Captivator	2	3	5	3rd
Berkeley LRCN	1	1	2	5th

Table 4: MSCOCO image captioning challenge leader board

8 Evaluation Metric

Below are various matrics for automatic or human evaluation.

Human evaluation metric:

M1: Percentage of captions that are evaluated as better or equal to human caption.
M2: Percentage of captions that pass the Turing Test.
M3: Average correctness of the captions on a scale 1-5 (incorrect - correct).
M4: Average amount of detail of the captions on a scale 1-5 (lack of details - very detailed).
M5: Percentage of captions that are similar to human description.

Automatic evaluation metric:

BLEU [16] is a popular machine translation metric that analyzes the co-occurrences of n-grams between the candidate and reference sentences. BLEU has shown good performance for corpus level comparisons over which a high number of ngram matches exist. However, at a sentence-level the n-gram matches for higher n rarely occur. As a result, BLEU performs poorly when comparing individual sentences.

METEOR evaluation metric [15]. Compared to other n-gram-based metrics such as BLEU, METEOR is more appropriate to evaluate sequential predictions. METEOR scores the predictions by aligning them to more than one reference sentences, which are based on exact, stem, synonym, and paraphrase matches between words and phrases. Therefore METEOR takes more linguistic and semantic information into consideration.

9 References

[1] Andrew, G., Arora, R., Bilmes, J.A. & Livescu, K., 2013, May. Deep Canonical Correlation Analysis.ICML(9):(1247-1255).
[2] Wang, W., Arora, R., Livescu, K. & Bilmes, J., 2015. On Deep Multi-View Representation Learning: Objectives and Optimization. Int. Conf. Machine Learning (ICML)(10):1083–1092.
[3] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R.S. & Bengio, Y. (2015) Show, attend and tell: Neural image caption generation with visual attention. International Conference on Machine Learning
[4] Vinyals, O., Toshev, A., Bengio, S. & Erhan, D., 2015. Show and tell: A neural image caption generator. IEEE Conference on Computer Vision and Pattern Recognition (9):3156-3164
[5] Bahdanau, D., Cho, K. & Bengio, Y., 2015. Neural machine translation by jointly learning to align and translate. ICLR.
[6] Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J. & Mikolov, T., 2013. Devise: A deep visual-semantic embedding model. Advances in neural information processing systems(9):2121-2129.
[7] Yang, Z., Yuan, Y., Wu, Y., Salakhutdinov, R. & Cohen, W.W., 2016. Encode, Review, and Decode: Reviewer Module for Caption Generation. arXiv preprint arXiv(9):1605.07912.
[8] Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z. & Yuille, A., 2014. Deep captioning with multimodal recurrent neural networks (m-rnn).ICLR.
[9] He, Wanjia, Weiran Wang, & Karen Livescu. 2016 Multi-view Recurrent Neural Acoustic Word Embeddings. arXiv preprint arXiv:1611.04496.
[10] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. & Zitnick, C.L., 2014. Microsoft coco: Common objects in context. In European Conference on Computer Vision (16):740-755.
[11] Rohrbach, A., Rohrbach, M., Tandon, N. & Schiele, B., 2015. A dataset for movie description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (13):3202-3212.
[12] http://www.mpi-inf.mpg.de/departments/computer-vision-and-multimodal-computing/research/vision-and-language/mpii-movie-description-dataset/
[13] https://mila.umontreal.ca/en/publications/public-datasets/m-vad/
[14] Torabi, A., Pal, C., Larochelle, H. & Courville, A. 2015: Using descriptive video services to create a large data source for video annotation research. arXiv preprintarXiv:
[15] Denkowski, M. & Lavie, A., 2014: Meteor universal: Language specific translation evaluation for any target language. In: Proceedings of the EACL 2014 Workshop on Statistical Machine Translation.
[16] Papineni, K., Roukos, S., Ward, T., Zhu & W.J., 2002: Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics, Association for Computational Linguistics (8):311–318
[17] Ryan Kiros, Ruslan Salakhutdinov & Richard S. Zemel, 2014. Unifying Visual-Semantic Embeddings with Multi- modal Neural Language Models.arXiv preprint arXiv:1411.2539.
[18] Chen, D.L. & Dolan, W.B., 2011. Collecting highly parallel data for paraphrase evaluation. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 (11):(190-200).
[19] Soomro, K., Zamir, A.R. & Shah, M., 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402.
[20] Wang, F. and Tax & D.M., 2016. Survey on the attention based RNN model and its applications in computer vision. arXiv preprint arXiv:1601.06823.
[21] Karpathy, A., Johnson, J. & Fei-Fei, L., 2015. Visualizing and understanding recurrent networks. arXiv preprint arXiv:1506.02078.
[22] Sutskever, I., Vinyals, O. and Le, Q.V., 2014. Sequence to sequence learning with neural networks. In Advances in neural information processing systems (9):(3104-3112).

[23] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K. & Darrell, T., 2015. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (10):(2625-2634).
[24] Lu, J., Xiong, C., Parikh, D. & Socher, R., 2016. Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning. arXiv preprint arXiv:1612.01887.
[25] Yan, F. & Mikolajczyk, K., 2015. Deep correlation for matching images and text. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (10):(3441-3450).
[26] You, Q., Jin, H., Wang, Z., Fang, C. & & Luo, J., 2016. Image captioning with semantic attention. arXiv preprint arXiv:1603.03925.
[27] Wang, W., Arora, R., Livescu, K. and Bilmes, J.A., 2015. Unsupervised learning of acoustic features via deep canonical correlation analysis. In IEEE International Conference on Acoustics, Speech and Signal Processing. (5):4590-4594.
[28] Wang, W., Lee, H. and Livescu, K., 2016. Deep Variational Canonical Correlation Analysis. arXiv preprint arXiv:1610.03454.
[29] He, W., Wang, W. and Livescu, K., 2016. Multi-view Recurrent Neural Acoustic Word Embeddings. arXiv preprint arXiv:1611.04496.
[30] Klaus Greff; Rupesh Kumar Srivastava; Jan Koutnk; Bas R. Steunebrink; Jrgen Schmidhuber 2015. "LSTM: A Search Space Odyssey".
[31] Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H. and Courville, A., 2015. Describing videos by exploiting temporal structure. In Proceedings of the IEEE international conference on computer vision.(9):(4507-4515).
[32] Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T. and Saenko, K., 2015. Sequence to sequence-video to text. In Proceedings of the IEEE International Conference on Computer Vision.(9):(4534-4542).
[33] Zhang, C. and Tian, Y., 2016, October. Automatic Video Captioning via Multi-channel Sequential Encoding. In European Conference on Computer Vision.(16):(146-161).
[34] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K. and Darrell, T., 2015. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE conference on computer vision and pattern recognition . (10):(2625-2634).
[35] Yu, H., Wang, J., Huang, Z., Yang, Y. and Xu, W., 2016. Video paragraph captioning using hierarchical recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (10):(4584-4593).
[36] Bin, Y., Yang, Y., Shen, F., Xu, X. and Shen, H.T., 2016, October. Bidirectional long-short term memory for video description. In Proceedings of the 2016 ACM on Multimedia Conference. (9):(436-440).
[37] Pan, P., Xu, Z., Yang, Y., Wu, F. and Zhuang, Y., 2016. Hierarchical recurrent neural encoder for video representation with application to captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (10):(1029-1038).
[38] Yu, H., Wang, J., Huang, Z., Yang, Y. and Xu, W., 2016. Video paragraph captioning using hierarchical recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (10):(4584-4593).
[39] Kaufman, D., Levi, G., Hassner, T. and Wolf, L., 2016. Temporal Tessellation for Video Annotation and Summarization. arXiv preprint.
[40] Peris, Á., Bolaños, M., Radeva, P. and Casacuberta, F., 2016, September. Video description using bidirectional recurrent neural networks. In International Conference on Artificial Neural Networks. (9):(3-11).
[41] Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R. and Saenko, K., 2014. Translating videos to natural language using deep recurrent neural networks. arXiv preprint.
[42] Choi, Y.Y.H.K.J. and Kim, G., End-to-end Concept Word Detection for Video Captioning, Retrieval, and Question Answering.2016. arXiv preprint.
[43] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. 2016. Jointly modeling embedding and translation to bridge video and language. CVPR.