# Image captioning using Show, Attend and Tell

**Tanaya Kolankari**
Electrical and Computer Engineering
University of California San Diego
A53265700
tkolanka@ucsd.edu

**Varad Joshi**
Electrical and Computer Engineering
University of California San Diego
A53272494
v1joshi@ucsd.edu

**Sneh Shah**
Electrical and Computer Engineering
University of California San Diego
A53264450
s5shah@ucsd.edu

Team Name: Team SVT
Github Repo: https://github.com/tkolanka/ece285_mlip_projectA

## 1 Introduction

Image Captioning is the process of generating textual description of an image. It uses both Natural Language Processing and Computer Vision to generate the captions. The most recent works in this domain have been around using Deep Learning models with the encoder-decoder framework. The generated caption should be able to not only describe the contents of an image but also capture the relationship between objects in natural language. Image caption generation involves two main tasks, generating a vectorial representation of images, and decoding those representations into natural language sentences. Image caption generation has long been viewed as a very difficult problem. This is mainly because it tries to imitate the human ability to compress a large amount of visual features into natural language. The "Attend" feature of our model tries to imitate the human behaviour of focusing on specific portions of the image to identify the relationship between them. The most trivial point of conflict occurs in evaluating the correctness of the caption. Another possible challenge is the dataset bias with the trained models over-fitting the common words on the trained data.

This project focuses on "Show, Attend and Tell", one of the Deep Learning models used for achieving the designated task of Image Captioning. It is one of the well known models, building on the Show and Tell methodology. The report will start with explanation of the model architecture in section 2 followed by the algorithm and the necessary equations. Section 3 will delve in to the Experimental setting for our specific task and the different modifications made to the original algorithm.

## 2 Show, Attend and Tell

The model uses deterministic soft attention along with varying intensities of teacher forcing. Moreover, we explored three different neural networks for encoding - vgg19, ResNet50 and ResNet101.

## 2.1 Encoder Architecture

The model takes in a raw image and generates a caption as a sequence of one-hot encoded words.

$$y = \{y_1, ..., y_C\},\ y_i \in \mathbb{R}^K$$

where C is the length of the caption and K is the size of the vocabulary.

A Convolutional Neural Network (CNN) is used to extract feature vectors from the image. The CNN generates *L* vectors of *D* dimensions, each of which corresponds to a different part of the image.

$$a = \{a_1, ..., a_L\},\ a_i \in \mathbb{R}^D$$

To obtain a correspondence between a feature vector and a portion of the image, the features are extracted from a lower convolutional layer. This allows the decoder to selectively focus on certain parts of the image by selecting a subset of the feature vectors.

## 2.2 Decoder Architecture

The model uses a long short-term memory (LSTM) cell to produce a caption, one word at each time step. The input to the LSTM cell is a context vector $\hat{z}$, the previous hidden state and the previously generated word. The context vector is the focused representation of the input image at that time step. We use a function $\phi$ that computes the context vector $\hat{z}$ from the feature vectors $a_i, i = 1, ..., L$ corresponding to features at different image locations. For each location *i*, the mechanism generates a weight $\alpha_i$ corresponding to the feature vector $a_i$. This weight is computed by an *attention model* $f_{att}$ for which we use a fully connected linear layer conditioned on the previous hidden layer. This is further discussed in the next section.

We trained multiple LSTMs with varying intensities of teacher forcing. With teacher forcing, the input to the next iteration of the LSTM takes the word from the correct caption rather than the word generated by the previous iteration. Through this, we expect to negate propagating the error from the previous iteration to the next iteration.

## 2.3 Deterministic Soft Attention

Here, we take the expectation of the context vector $\hat{z_t}$ directly,

$$\mathbb{E}_{p(s_t|a)}[\hat{z_t}] \ = \ \sum_{i=1}^{L} \alpha_{t,i} a_i$$

and formulate a deterministic attention model by computing a soft attention weighted feature vector $\phi(\{a_i\}, \{\alpha_i\}) \ = \ \sum_i^L \alpha_i a_i$ as per [5]. This makes the model smooth and differentiable so that end-to-end learning happens using standard back-propagation.

The normalized weighted geometric mean (NWGM) for the softmax $k^{th}$ word prediction is given as follows:

$$NWGM[p(y_t = k \mid a)] \ = \ \frac{\prod_i exp(n_{t,k,i})^{p(s_{t,i}=1|a)}}{\sum_j \prod_i exp(n_{t,j,i})^{p(s_{t,i}=1|a)}} \ = \ \frac{exp(\mathbb{E}_{p(s_t|a)}[n_{t,k}])}{\sum_j exp(\mathbb{E}_{p(s_t|a)}[n_{t,j}])}$$

The above equation shows that NWGM of a softmax unit is obtained by applying softmax to the expectations of the underlying linear projections.

# 3 Experimental Setting

## 3.1 Dataset and Training Parameters

The dataset used for running the network is obtained from MS COCO (Common Objects in Context dataset). Each image in the dataset is associated with 5 captions which correctly describe the corresponding image.

For the LSTM decoder, there are two sets of weights and biases associated with each layer. One set corresponds to the current weighted image and the other corresponds to the previous hidden state. For the first hidden layer, there needs to be an input which needs 2 more linear layers. These 2 layers add 4 more parameters (weights and biases for each layer).

For the Attention module, there are 3 linear layers where each layer has its corresponding weights and biases. Along with this, the embedding also has its own weights which add to the training parameters list.

## 3.2 Configuration Setting

**[1]** uses the CNN encoder and the decoder LSTM in the model. Building on this, the aim of our project was to change the parameters of the experiment and try to better the performance. The measure of the performance is given by the BLEU score for each epoch. The different results are obtained by varying the convolutional layer in the encoder, modifying the regularization constant $(\alpha)$. The different settings for the experiments conducted were:
Experiment 1: VGG19 encoder, without Teacher Forcing
Experiment 2: VGG19 Encoder, with Teacher Forcing
Experiment 3: VGG19 Encoder, with Scheduled Sampling for Teacher Forcing
Experiment 4: VGG19 Encoder, with Teacher Forcing and Higher Regularization
Experiment 5: Resnet101 Encoder, with Teacher Forcing
Experiment 6: Resnet50 Encoder, with Teacher Forcing

## 3.3 Evolution of Loss over Training

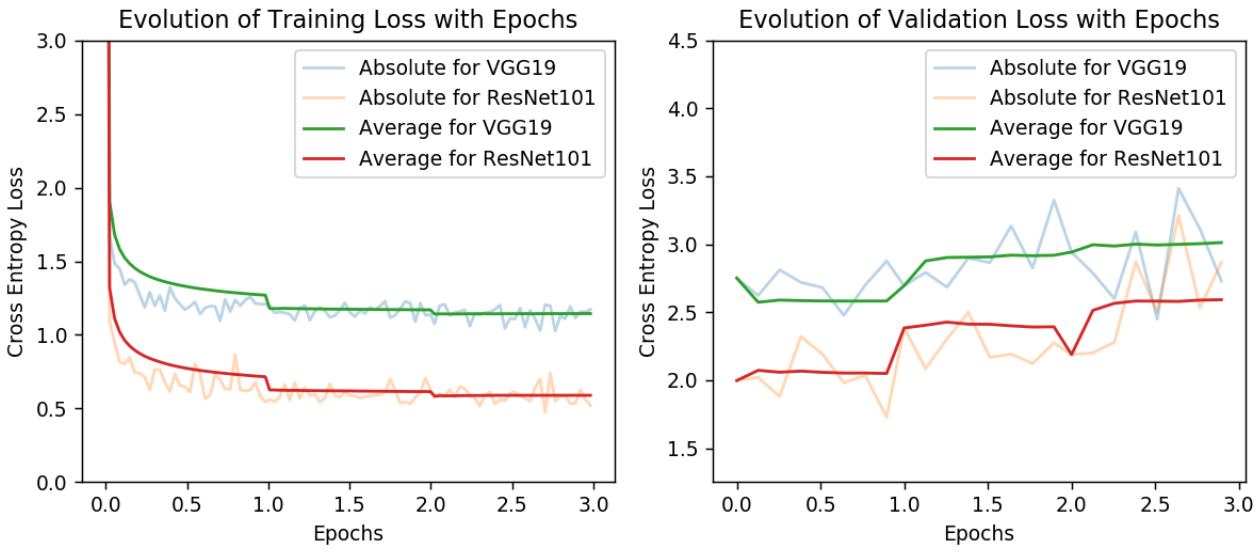


Figure 1: Evolution of Loss for different encoders

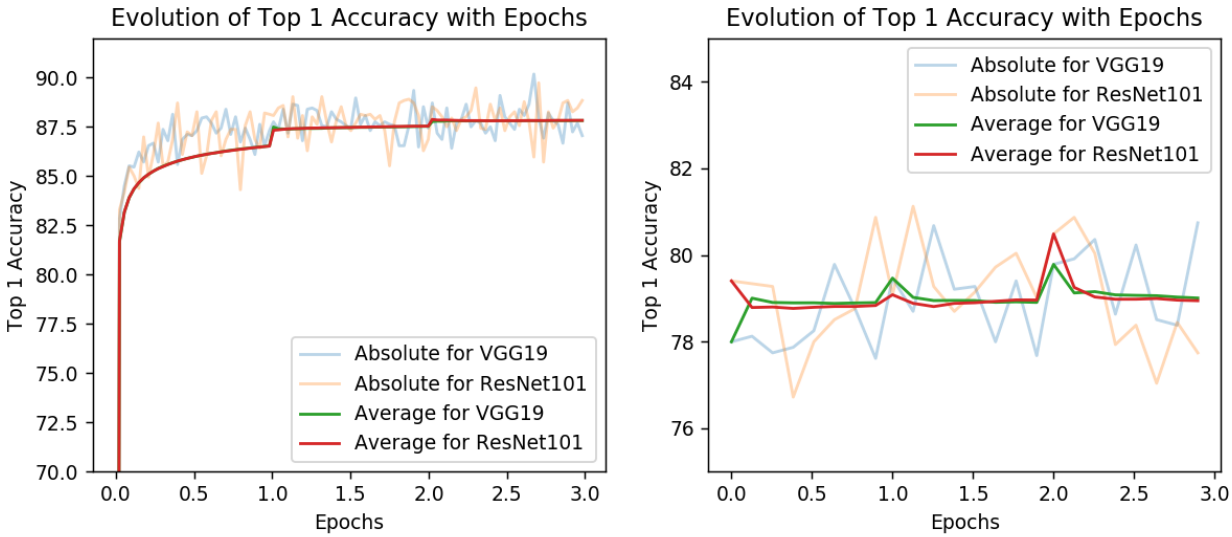


Figure 2: Evolution of Top 1 Accuracy for different encoders

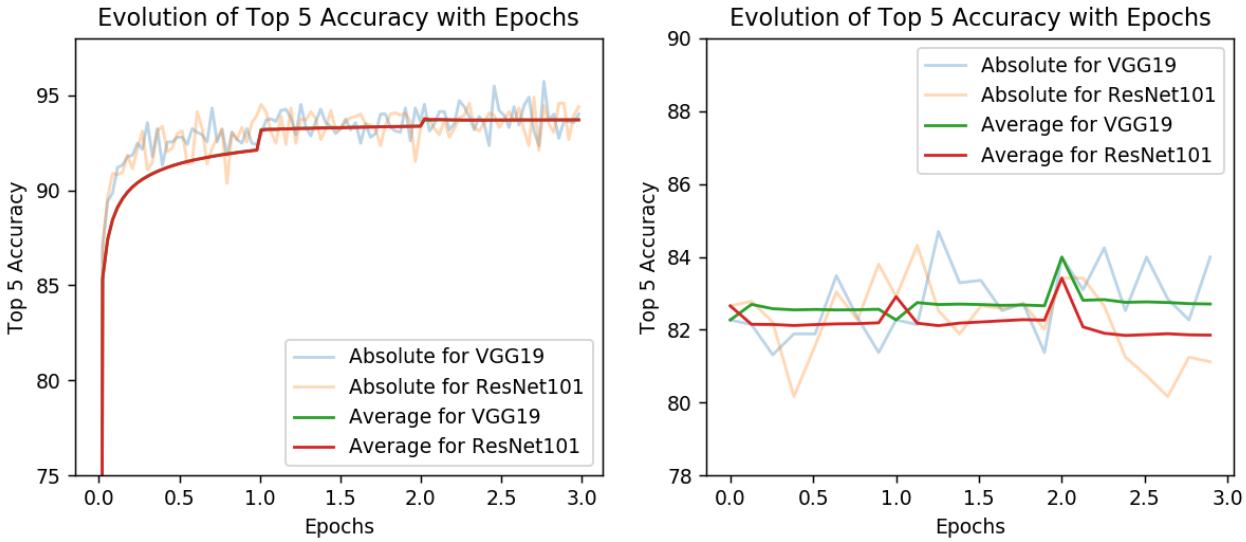


Figure 3: Evolution of Top 5 Accuracy for different encoders

## 3.4 Validation and Testing Procedure

One epoch is assumed to be complete only after both training and validation are complete. The validation for an epoch is done immediately after all the batches are trained. We are using a **batch size of 32** and **learning rate of 4e-4**. Using Andrej Karapathy's data splits for the validation data, we used the inbuilt library in the Python NLTK to calculate the BLEU scores.

# 4 Results

## 4.1 Effect of Teacher Forcing (TF)

As can be seen from the fig 4, enabling Teacher forcing on the network always or with a probability is better than not using TF.

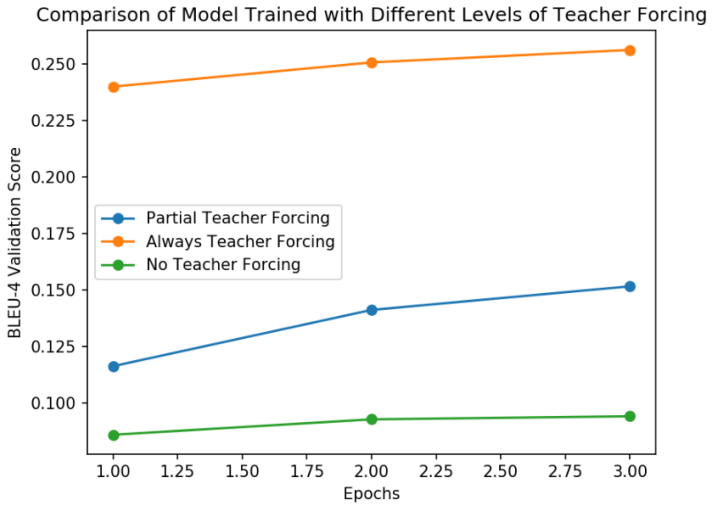


Figure 4: Effect of Teacher Forcing in VGG19 encoder


Figure 5: Exp 1 (Left); Exp 3 (Center); Exp 2 (Right)


Figure 6: Exp 1 (Left); Exp 3 (Center); Exp 2 (Right)

Disabling TF trains the model poorly and hence the words repeat or the caption fumbles. Enabling a TF Ratio of 0.5 (Scheduled Sampling) captions the image much better. In Fig.2 and Fig.3, Exp 1 is the leftmost which fails miserably with the captioning as well as English. Exp 3 with partial TF captions good enough to understand what the image has. Exp 2 with TF always true describes the image much better with better adjectives and analyses it better than the other two variants.

## 4.2 Effect of Regularization Constant

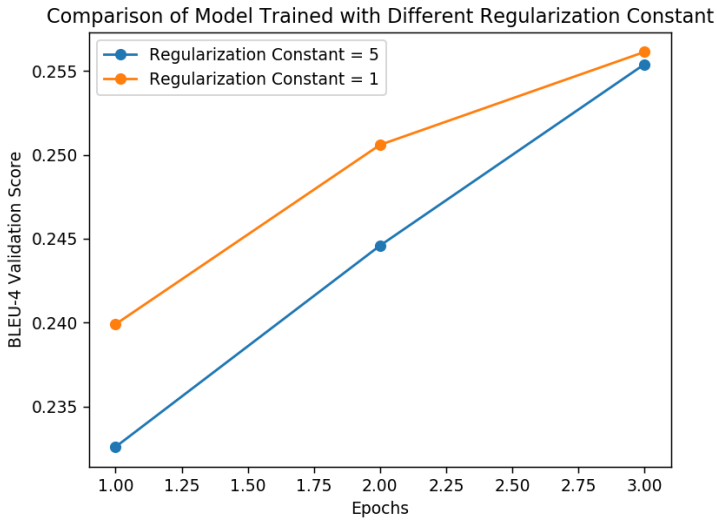


Figure 7: Effect of Regularization Constant

The regularization $\alpha_c$ incurs a higher penalty on the attention weights. By penalizing the weights of the attention, there is a possibility that the network will restrict the attention in order to minimize the overall loss function. After running 3 epochs, the results obtained are not conclusive enough to decide the effect of $\alpha_c$ on the overall loss. Training the network for more epochs might be one solution to validate our understanding.


Figure 8: $\alpha_c$ = 1 (Left); $\alpha_c$ = 5 (Right)

In the two figures(Fig 5 and Fig 6), both the captions describe the image in accordance to what the image actually is. From the BLEU scores, it is evident that increasing $\alpha_c$ did not serve its purpose for the limited number of epochs. These two sample images and their corresponding captions also add to the ambiguity. For Fig 5, $\alpha_c$ = 5 has a more descriptive caption and the trend is reversed for Fig 6.

**A cutting board topped with a variety of fruits and vegetables on a table.**

**There are many different types of vegetables on the table.**

Figure 9: $\alpha_c$ = 1 (Left); $\alpha_c$ = 5 (Right)

## 4.3 Effect of Different Encoders

As can be seen from the fig 10, using different encoders can significantly vary performance. In our case, switching from VGG19 to ResNet101 gave us lower loss but the top-1 and top-5 accuracy remained constant. This can be observed in fig 1, fig 2 and fig 3.

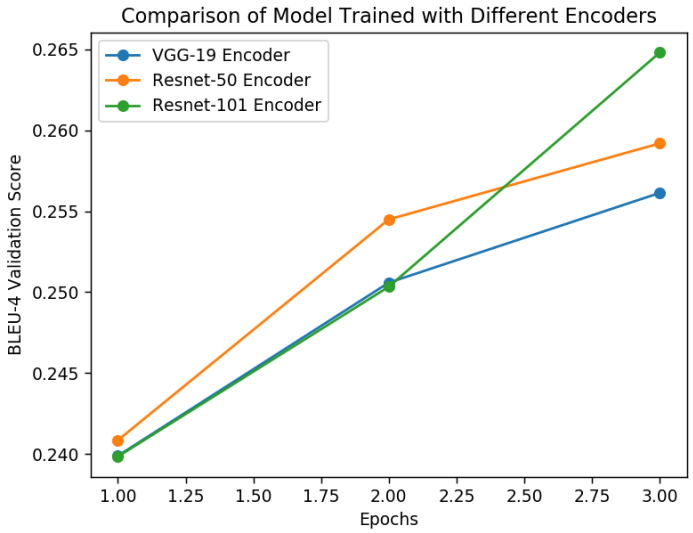


Figure 10: Effect of Different Encoders

In the two figures Fig 10 and Fig 11, one can observe that all the 3 models caption the images equally good. The description will pass the human evaluation test. Comparing the BLEU scores hints that Resnet101 outperforms the other 2 networks. The summary of the BLEU-1, BLEU-2, BLEU-3 and BLEU-4 are summarized in the table below.

Figure 11: VGG 19 (Left); Resnet50 (Center); Resnet101 (Right)

A group of people are sitting at a dining room table.

A group of young people sitting on a couch playing a video game

A group of people sitting at a table in a living room.

| Exp. No. | Experiment | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 |
|---|---|---|---|---|---|
| 1 | VGG19 encoder, without Teacher Forcing | 0.559 | 0.306 | 0.172 | 0.094 |
| 2 | VGG19 Encoder, with Teacher Forcing | 0.714 | 0.513 | 0.365 | 0.256 |
| 3 | VGG19 Encoder, with Scheduled Sampling for Teacher Forcing | 0.605 | 0.370 | 0.239 | 0.152 |
| 4 | VGG19 Encoder, with Teacher Forcing and Higher Regularization | 0.710 | 0.509 | 0.364 | 0.255 |
| 5 | Resnet101 Encoder, with Teacher Forcing | 0.726 | 0.526 | 0.377 | 0.265 |
| 6 | Resnet50 Encoder, with Teacher Forcing | 0.720 | 0.518 | 0.370 | 0.259 |

# 5 Poor Captioning Examples

In fig 12, the captions of all of the three models clearly fail the human evaluation test for the given image. The blurred background is being mapped to different words by the three models. The model could not extract the features of the image effectively. Hence, to caption the image, since it did not have enough information from the features, it used the most commonly occurring words to create the caption.

Figure 12: VGG 19 (Left); Resnet50 (Center); Resnet101 (Right)

A person holding an umbrella in front of a group of people.

A woman in a blue shirt is holding a cell phone to her ear.

A person on a motor bike in front of a car.

Figure 13: Image Captioning for bus on Resnet101

A double decker bus driving down the street.

A double decker bus driving down the street in a residential area.

A double decker bus driving down the street.

Fig 13 is another example of the model performing poorly. Our analysis is that this flaw is built inside the model itself. The model is correct in captioning the leftmost image which actually contains the double decker bus. As discussed earlier, the model is over-fitting to the training data. Hence, it identifies all of the buses as double decker buses. At the same, the words in the caption that follow the word 'double decker bus' are also the same in all of the 3 captions. This dataset bias will always exist any model we train on.

# 6 Discussion

In our experimentation, we modified the convolutional layer for the encoding part which gave us better results on Resnet compared to VGG. On a simplistic scale, as the number of convolutional layers increased, the network performed better on the BLEU-4 evaluation. When using Teacher Forcing, we anticipated that the model will perform poorly because it is spoon fed the captions. But that wasnt the case. There was a certain level of Teacher Forcing required to overcome the initial bias and generate the correct caption.

A commonly occurring problem of dataset bias was also seen in our model. In some of the images, it tried to fit the more commonly occurring objects in images where they were present, out of their natural context. We tried to read up more on the issue and found about a possible solution to overcome this problem. The solution proposed by **[4]** creates a test diagnostic dataset where the common object occur in unusual places and try to train the network to remove this bias.

One improvement which would definitely help as seen in the reference papers is running for larger number of epochs. The analysis is highly biased towards our settings and hence cannot be generalised. The other but more time consuming practice would be to perform an end-to-end training of both the decoder and the encoder. Currently, since we are using pre-trained models for the encoder, they are not customised for this specific task.

# References


[1] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, Yoshua Bengio. *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*

[2] https://github.com/AaronCCWong/Show-Attend-and-Tell *A PyTorch implementation of Show, Attend and Tell*

[3] https://machinelearningmastery.com/teacher-forcing-for-recurrent-neural-networks/ *Teacher Forcing for Recurrent Neural Networks*

[4] Pierre L. Dogin, Igor Melnyk, Youssef Mroueh, Jarret Ross, Tom Sercu. *Adversarial Semantic Alignment for Improved Image Captions*

[5] Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua. *Neural machine translation by jointly learning to align and translate.*