# MLP Coursework 4: Image Captioning With Neural Networks

s1574336 s1619123

## Abstract

For this project, we explored performance of neural networks on image captioning. The two main comparisons include (1) LSTM vs. GRU gating mechanisms; and (2) non-attention RNNs vs attention-based RNNs.

In this final report, we built a non-attention GRU model, an attention LSTM model and an attention GRU model in comparison of our baseline—a non-attention LSTM model. The experiments were conducted on the current benchmark dataset—Microsoft COCO Caption and the models are evaluated through the MSCOCO evaluation system.

We observed that GRU gating models have similar final results compared to the LSTM gating model but show a trend for early convergence. Moreover, We can also see a boost in performance from using attention models. After analysing the results quantitatively and qualitatively, we tuned the hyperparameters for three new models to find the their settings. The best model is the attention LSTM model with 2 hidden layers, 400 hidden dimensions and 300 hidden attention dimensions. This achieved 28.3 (+0.3), 24.1 (+0.3), 90.8 (+1.4), 51.9 (+0.4) for BLEU-4, METEOR, CIDEr and ROUGE_L scores respectively on the test set compared to our baseline model.

## 1. Introduction

Automatic interpretation of images in a textual format is a challenge in the fields of Natural Language Processing and Computer Vision in Artificial Intelligence (Bernardi et al., 2016). A good image captioning generator is able to produce representative text from any given image. Meanwhile, it also ensures the generated output is understandable and easy to interpret. Image captioning generators can be applied to various areas such as image retrieval in news articles (Feng & Lapata, 2013). Thus, we decided to explore performance of neural networks on image captioning for our project.

In the previous work, we built a baseline based on the architecture from the Show and Tell project (Vinyals et al., 2014) which uses CNNs for feature extraction and LSTM RNNs for predictions with the best hyperparameters—0.75 dropout keep probability and 400 hidden units. For our baseline, we achieved 28.0 points, 23.8 points, 89.4 points, 51.5 points for BLEU-4, METEOR, CIDEr and ROGUE_L scores respectively.

In this report, we will list research questions and objectives in Section 1. The methodologies and experiments will be presented in Section 2 and 3 respectively. In Section 4, we will review current published work for further understanding of this topic. The report will be concluded by a brief summary of final outcomes regarding our research questions in Section 5.

**Research questions:** The goal of this project is to investigate the performance of different neural network structures on image captioning. In the interim report, we covered a few research questions such as how the number of hidden units and dropout keep probability affect the performance of our models. In this report, we will construct models with different neural architecture (GRU models and attention-based RNN models) to strike a comparison of performance between the new models and our baseline. Based on this, the following research questions will be investigated.

1. Will GRU, a relatively new gating mechanism, perform better compared with our baseline LSTM models?
2. How much improvement can the attention mechanics achieve compared to normal LSTM/GRU?
3. What are the preferred hyperparameters for the new models?
4. With their best settings, how do the new models perform on the test set?
5. What are the limitations of our attention-based models, and what potential solutions are there for further investigation?

**Objectives:** The overall workflow for our project is to build neural network models, tune the models with different hyperparameters according to CIDEr scores on the validation set, and finally analyse the results with regard to MSCOCO automated metrics. To specifically address the research questions for this report, our objectives are as follows:

- Use Inception-V3, a type of convolutional neural network, to extract features, and use GloVe representation as word embedding for caption preprocessing. Note that extracting features for attention models is different from that for non-attention models.

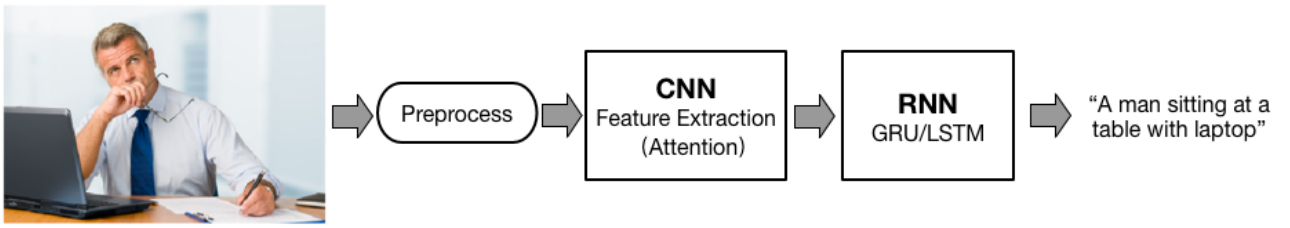



*Figure 1.* Overall pipeline of our image captioning models for this project

- Build a basic GRU RNN model, a basic attention-based LSTM model and a basic attention-based GRU model, and compare their performances.

- Tune the number of hidden layers in the set of {1, 2, 3} and the number of hidden units (also attention hidden units for attention based models) in the set of {300, 400, 500} on the validation set.

- Compute BLEU, METEOR, CIDEr and ROUGE_L scores of the best of all new models (GRU, attenion LSTM, attention GRU) in comparison with our baseline LSTM model on the test set.

## 2. Methodology

In this section, we will present the technical methodologies for achieving our goals. The work flow is shown in Figure 1. As can be seen, we start with feature extraction from the prepossessed images, then we input the features into the recursive neural networks. The performance will be evaluated by the MSCOCO automatic evaluation system on the test set.

Our standard LSTM and GRU models are modified based on the architecture from the Show and Tell project (Vinyals et al., 2014) as their architecture had received great success at the COCO 2015 Captioning Challenge. The implementation of the attention mechanism is inspired from the Show, Attend and Tell project (Xu et al., 2015).

### 2.1. Data preprossessing

As in coursework 3, the dataset was split into 113287, 5000, 5000 subsets to give the training, validation and test sets. The captions were tokenized in the same way after filtering out non-alphanumeric characters and converting all letters to lowercase. However, as it has been noticed in coursework 3 that the training time is proportional to the length of the longest sentence per update, we used a trick by building a dictionary which maps the length of each sentence for all corresponding captions (Xu et al., 2015). Therefore, we can randomly sample a length and retrieve a batch of that size during training. This accelerates the training procedure without noteworthy influence on performance.

### 2.2. Feature Extraction: Inception-V3

The first part of feature extraction is the same as in coursework 3. As Convolutional Neural Networks (CNNs) are considered as the state-of-the-art approach in image classification tasks, we continue to use them to represent images. We continued to apply Inception V3 (Szegedy et al., 2014) as it has lower computation cost compared withVGG and we do not want the performance to be influenced by different CNN architectures.

However, as we will feed the image vector into LSTM at every step to build attention models, we extracted the image with dimensions $64 \times 2048$ from the third to the last layer. For the standard GRU model, we still extracted the image with dimensions $1 \times 2048$ from the second to the last layer. The corresponding parameters were obtained with the code snippet in Listing 1.

**Listing 1** The code snippet to get the tensors in Inception-V3 CNN for feature extraction

```
# from the second to the last layer
sess.graph.get_tensor_by_name("pool_3:0")
# from the third to the last layer
sess.graph.get_tensor_by_name("mixed_10/join:0")
```

### 2.3. Word Embedding: GloVe representation

As in coursework 3, we use GloVe 300 dimensional representation (Pennington et al., 2014) to encode the information of image captions and project them into the vector space. The cost function of the model can be expressed by Equation 1:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - logX_{ij})^2 \qquad (1)$$

where $X$ is the word co-occurrence matrix, and $X_{ij}$ is the element which stands for the co-occurrence times of word $i$ and word $j$. $w_i, w_j$ stand for the vectors of word $i$ and word $j$. $b_i$ and $\tilde{b}_j$ are the biases defined by the author. $f(x)$ is the weighting function, and $V$ is the vocabulary size.

### 2.4. Recursive Neural Networks

We use Recurrent Neural Networks (RNNs) as the main component in our models. We built our baseline with LSTM gates in coursework 3, whereas here we add the model with GRU gates in comparison with our baseline. The basic structure of RNNs is illustrated in Figure 2.

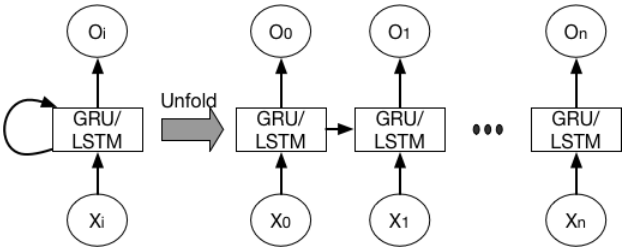



*Figure 2.* Basic RNN structure where $x_i$ and $o_i$ are the input and output at time step $i$, n is the number of inputs in sequences. The gating mechanism is either GRU or LSTM for our models.

**Long Short-Term Memory:** As mentioned in coursework 3, the use of LSTMs enables models to establish a long-term dependency in sentences, which allows the model to narrow down choices of the next predicted word. The long-term memory is achieved by tracking all previous predictions that are generated so far while predicting the next best word. On the other hand, LSTMs also consider the previous word during the prediction process. This is referred to as short-term memory. The equations for forward passing are presented in Equation 2.

$$\begin{aligned} forget_t &= \sigma_g(W_f[h_{t-1}, x_t] + b_f) \\ input_t &= \sigma_g(W_i[h_{t-1}, x_t] + b_i) \\ output_t &= \sigma_g(W_o[h_{t-1}, x_t] + b_o) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c[h_{t-1}, x_t] + b_c) \\ h_t &= o_t \circ \sigma_h(c_t) \end{aligned} \tag{2}$$

where at each time step t, $forget$, $input$ and $output$ are vectors for forget, input and output gates. $c$ and $h$ are the cell state and output values of the LSTM unit. $W$ and $b$ are the weight metrics and bias parameters. $\sigma_g$ is a sigmoid function, whereas $\sigma_c$ and $\sigma_h$ are hyperbolic tangent functions. The operator $\circ$ represents an element-wise multiplication.

**Gated Recurrent Units:** Similar to LSTM, Gated Recurrent Units (GRU), which were introduced in 2014 (Cho et al., 2014), also has a gate mechanism. Compared to LSTMs, GRUs have fewer parameters and appear to have similar performance on polyphonic music modelling and speech signal modelling (Chung et al., 2014). In other words, GRUs use a simpler structure and thus is more efficient. The equations for forward passing are presented in Equation 3.

$$\begin{aligned} z_t &= \sigma_g(W_z[h_{t-1}, x_t] + b_z) \\ r_t &= \sigma_g(W_r[h_{t-1}, x_t] + b_r) \\ h_t &= z_t \circ h_{t-1} + (1 - z_t) \circ \sigma_h(W_h[r_t \circ h_{t-1}, x_t] + b_h) \end{aligned} \tag{3}$$

where at each time step t, $z$ and $r$ represent update gate and reset gate values. $x$ and $h$ are the input and output. $W$ and $b$ are the weight metrics and bias parameters. $\sigma_g$ is a sigmoid function, whereas $\sigma_h$ is a hyperbolic tangent function. The operator $\circ$ represents an element-wise multiplication.

**Attention Mechanics:** Researchers (Bahdanau et al., 2014) argued that extracting a fixed-length vector from a various-length input may be a bottleneck, and they proposed the attention mechanism which was initially used in machine translation. The core idea behind the attention mechanism is that this algorithm takes inputs of a certain related context as vector representations. In this way, the network focuses on a relevant context of inputs and thus becomes more efficient. A similar concept is also applied to our models where we extract partial feature vectors of the image and generate a sentence accordingly.

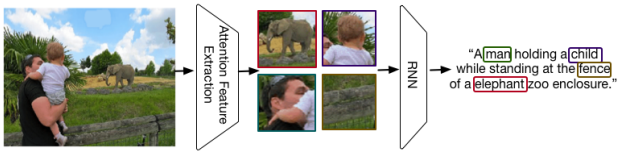



*Figure 3.* An example of our attention flow chart using a random image from MSCOCO.

An example of our attention flow chart is shown in Figure 3. We used the attention mechanism on the input image on our CNN layer, the attended feature vectors are then passed to RNN layers. We implemented attention models that utilise multiple subsections of images, which was described in (Xu et al., 2015). We computed the attention distribution at each time step $t$ according to the Equation 4.

$$\begin{aligned} \hat{I}_k &= \sigma_g(W_I I_k + b_I) \\ a_t^k &= W_a \tanh_g(W_{aI}\hat{I}_k + W_{ah}h_{t-1} + b_a) \\ att_t &= \text{softmax}(a_t + b_\alpha) \end{aligned} \tag{4}$$

where at each time step t, $\hat{I}_k$ is computed during the pre-processing stage using $W_I$ and $b_I$ from the best normal LSTM/GRU model. $x$ and $h$ are the input and output. $W$ and $b$ are the weight metrics and bias parameters. $att$ is the attention distribution.

Once the attention distribution was computed, we compute the weighted sum over 64 subsections, and concatenate the value with the embedded vector of the current word in the caption. i.e. the current word $s$ is concatenated with the image feature vector (a sum of images' 64 feature vectors scaled by the attention distribution $att$) at each time step $t$. The output is computed according to the Equation 5.

$$\begin{aligned} x_t &= (\sum_{n=1}^{64} att_t^k \hat{I}_k) \odot W s_t^I \\ h_t &= f_{RNN}(x_t, h_{t-1}) \\ output_{t+1} &= \text{softmax}(W h_t + b) \end{aligned} \tag{5}$$

where ⊙ represents vector concatenation. $x$ and $h$ are the input and output. $W$ and $b$ are the weight metrics and bias parameters. $att$ is the computed attention distribution. $s_t^I$ of each image $I$ represents the caption and $s$ is the word at time step $t$.

### 2.5. Training

Same as before, we used the tensorflow framework to train our model and cells. The relevant functions are shown in List 2.

**Optimiser and loss** According to previous coursework, Adam optimisation with default hyperparameter values (learning rate = 0.001, beta1=0.9, beta2 = 0.999, epsilon=1e-8) always performed well, therefore we kept it consistent throughout our experiments. Cross-entropy softmax error was used to compute loss for each word in the batch.

**Initialisation** Xavier initialisation (Glorot & Bengio, 2010) was used for weight, and constant initialisation was used for bias of all layers.

**Regularisation** Overfitting is always a problem in machine learning. In the previous coursework, dropout has been investigated and we noticed it could reduce the overfitting efficiently. Therefore, we kept it consistent and the dropout rate is 0.25 as optimised in coursework 3.

**Listing 2** The code snippet for training procedures

```
#LSTM cell
tf.nn.rnn_cell.BasicLSTMCell
#GRU cell
tf.nn.rnn_cell.GRUCell
#Adam optimiser
tf.train.AdamOptimizer
#loss
tf.nn.sparse_softmax_cross_entropy_with_logits
#weight initialization
tf.contrib.layers.xavier_initializer
#bias initializer
tf.constant_initializer(0)
# dropout
tf.rnn_cell.DropoutWrapper
```

### 2.6. Evaluation Metrics

We will continue use the same metrics used for our baseline to evaluate the quality of generated captions using our models: BLEU, METEOR, CIDEr and ROGUE_L scores. We will also consider changes in training loss to analyse the phenomenona which occur in the training process, such as convergence.

## 3. Experiments

We conducted the following experiments to accomplish our two main goals for this project: (1) investigate the impacts of using GRU model compared to our baseline LSTM model; (2) investigate the impacts of using attention mechanisms. In this section, we present the detailed implementation and settings, as well as the hyperparameters that were used for model tuning. The rest of the section covers the quantitative and qualitative analyses done on the test set using our four best final models.

### 3.1. Implementation

The implementation of our vanilla GRU model is similar to that in coursework 3. We first used the MSCOCO application programming interface (API) to access the MSCOCO data and extracted the features with Inception V3, then fed them into the GRU network, followed by the word vectors. For the attention- based models, we fed the image features at each time step so that the model can learn to attend to different regions. As stated in Section 3, the Adam optimiser was used to minimise loss during training. However, instead of directly calculating loss on the validation set, we evaluated the captions generated on the validation set to apply early stopping. As previous studies indicated that CIDEr is the preferred metric when optimising the hyperparameters of the model (Rennie et al., 2016), we chose the best model according to the CIDEr score on the validation set.

The optimized parameters obtained from coursework 3 are shown in Table 1. The same settings will be used for our GRU and attention-based models to compare their performances. For attention hidden dimension, we used 500 as a start point, as it is close to 512 which is the default value used in the original research (Xu et al., 2015). After investigating the influence of the attention mechanism, we also explored the best settings for each model and reported their metrics scores on the test set. The hyperparameters we tried are shown in Table 2.

| Layers | Learning rate | Batch batch | Hidden dimension | Dropout rate |
|---|---|---|---|---|
| 1 | 0.001 | 256 | 400 | 0.25 |

*Table 1.* Optimised settings for the baseline model

| Number of layers | Hidden dimension | Attention dimension |
|---|---|---|
| 1,2,3 | 300,400,500 | 300,400,500 |

*Table 2.* Hyperparameters we have deduced to find the optimised settings

[1]See https://github.com/kelvinxu/arctic-captions.git

### 3.2. Quantitative Analysis

For the quantitative analysis, we chose the four best models according to their CIDEr scores on the validation set and evaluated the models on the test set. We mainly consider the best final scores on the test set, along with the fluctuations in training loss, and CIDEr scores on validation sets when analysing the training process.

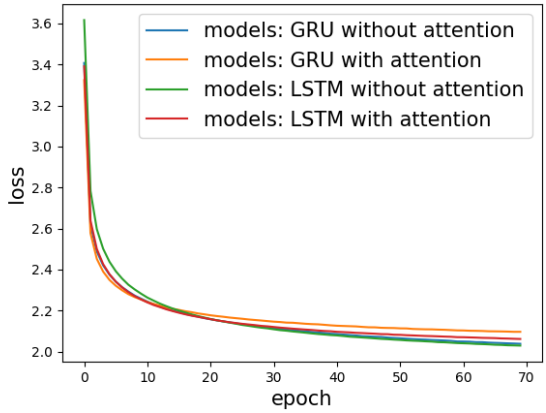



*Figure 4.* The training loss of the non-attention LSTM/GRU model and attention LSTM/GRU model over 70 epochs.

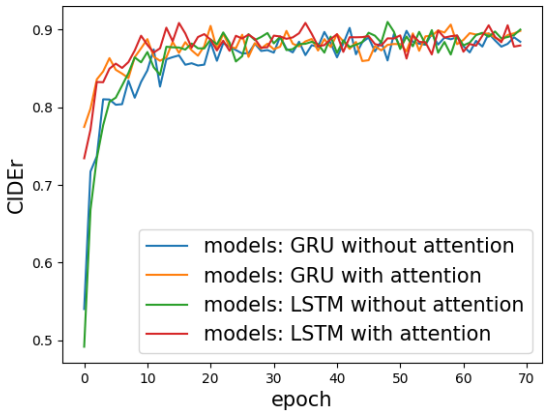



*Figure 5.* The CIDEr scores of the non-attention LSTM/GRU model and attention LSTM/GRU model over 70 epochs.

The changes in training loss of our models are shown in Figure 4. The attention models converged slightly faster for the first 20 epochs. It was also noted that the training loss for our normal GRU model also converged faster than the normal LSTM model: This is consistent with conclusion from empirical evaluation (Chung et al., 2014), which argued that GRU is a computationally simpler model compared to LSTM gating. On the other hand, we also measured the training time for normal LSTM and GRU models. The results indicated an insignificant difference in the training time for an epoch, both requiring approximately 8 minutes for non-attention models and 28 minutes for attention models.

With regard to the changes in CIDEr scores shown in Figure 5, it can be observed that the attention models achieved higher scores from the early stages, which is also reflected in the changes in training loss. However, with the increase in number of epochs, the differences in CIDEr score became diminished.

Before determining our final four models, we adjusted the number of layers and the number of attention hidden dimensions to generate the best model among the categories. We applied early stopping according to CIDEr scores to avoid overfitting. The settings yielding the highest CIDEr scores were then used on our test set. The hyperparameters for our final four models are shown in the Table 3. We will use these settings to evaluate the performance on our test set.

In Table 4, we reported our best final scores (BLEU-4, CIDEr, METEOR and ROUGE_L) on the test sets that were achieved by our best four models. The final scores of standard LSTM and GRU gating models are similar.

In addition, we could also observe a boost in performance from using attention models. We tuned the number of layers and attention dimensions for our attention models, and got the best hyperparameter settings for our best models on the test sets. Our best model is the attention LSTM model with 2 hidden layers, 400 hidden dimensions and 300 hidden attention dimensions. This best final model achieved 28.3 (+0.3), 24.1 (+0.3), 90.8 (+1.4), 51.9 (+0.4) for BLEU-4, METEOR, CIDEr and ROUGE_L scores respectively compared to our baseline.

| Models | Layers | Hidd Dim. | Att Dim. | CIDEr(epoch) |
|---|---|---|---|---|
| GRU | 1 | 400 | n/a | 90.22 (42) |
| LSTM-att | 2 | 400 | 300 | 91.60 (64) |
| GRU-att | 1 | 400 | 500 | 90.64 (58) |

*Table 3.* The best settings from justifying the hyperparameters for new models. Note that LSTM is omitted as its values were obtained in coursework 3. It can be found in Table 1.

| Models | BLEU-4 | CIDEr | METEOR | ROUGE_L |
|---|---|---|---|---|
| Baseline | 28.0 | 89.4 | 23.8 | 51.5 |
| GRU | 27.2 | 86.9 | 23.6 | 51.1 |
| LSTM-att | **28.3** | **90.8** | **24.1** | **51.9** |
| GRU-att | 27.5 | 88.6 | 23.8 | 51.4 |

*Table 4.* The final BLEU-4, CIDEr, METEOR and ROUGE_L scores achieved for all models with the settings listed in Table 3.


| Ground Truth |
|---|
| There is a map in the street of the city. |
| a bus stop map in a city near a water fountain. |
| The subway stop Square Victoria entrance and the map of the neighborhood. |
| A map and street sign with building in background. |
| A map of the town in the middle of the street with buildings in the background. |

*Figure 6.* Example image in the test dataset with its five original captions.

| Model | Generated caption |
|---|---|
| LSTM | a sign that is on a pole on a street |
| GRU | a sign on a building with a sign on it |
| attention based LSTM | a street sign on a pole in front of a building |
| attention based GRU | a street sign on a city street with buildings in the background |

*Table 5.* Generated captions for the example image with different models but at their best settings.

### 3.3. Qualitative Analysis

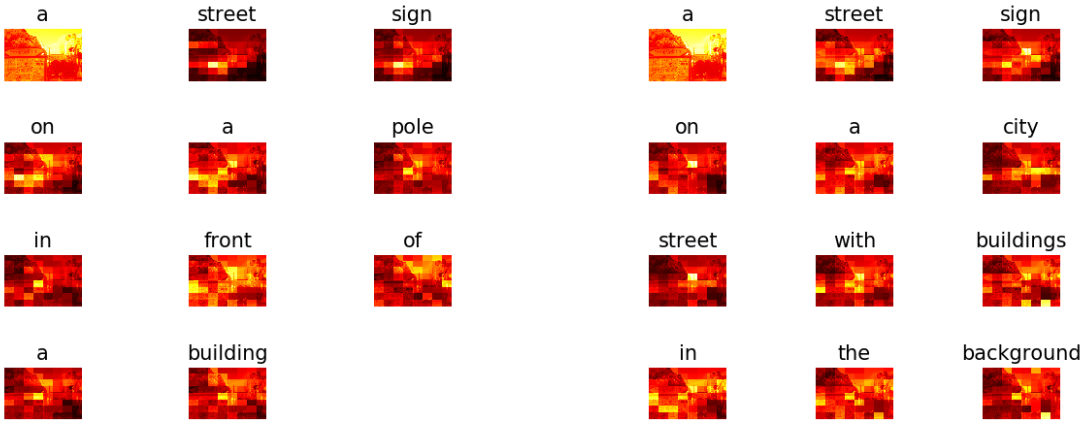
In order to present our results more visually, we generated random samples on the test set and compared the performances of different models qualitatively. Figure 6 shows the image content of an example sample and its corresponding five ground-truth captions in MSCOCO. The captions generated by our four models with their best settings are shown in Table 5. When given this sample, all models were able to capture the sign in the image. Compared with attention models, LSTM and GRU both try to describe the position of the sign: LSTM detects a street while GRU detects a building. However, neither of them understand the relationship between the building and the sign, and the syntax is quite unnatural. In contrast, both attention based models are more consistent with human intuition, with more detailed descriptions of the surroundings and relatively natural syntax. The attention probability distribution over the image is also shown in Figure 7. This presents the variance of the distribution when each word in the resultant caption is predicted, where bright rectangles indicate the attended regions. The attention maps are fuzzy as we only divided the image into 8 * 8 regions, which is much coarser compared to the original research (Xu et al., 2015) to save computational cost. The Gaussian filter was also omitted as it didn't help much with our 64 spatial regions. When predicting words such as sign, pole, buildings, the relevant regions in the image are brighter than others. There are more examples shown in Figure 8 to show how the model attends to objects in an image. It can be noticed that although some generated captions cannot describe the image contents entirely, and might misrepresent the actions (see the left in Figure 8), the attention mechanism still attends to the correct objects and it also gives us some intuition into what the model has seen and why it has generated that caption.

One drawback of our attention model is that some words with little lexical meaning (which only contributes to the syntax of the sentence) do not have corresponding image contents. For example, the attention maps for words such as "a","in", "of" in Figure 7 do not make much sense, as the attention mechanism might ask the language model to correlate these words with "sign" or "pole". Lu et al. argued that previous publications have demonstrated that gradients from non-visual words could deteriorate the overall effectiveness of attention mechanism (Lu et al., 2016). They also suggested that a possible solution is to apply the attention when it is necessary, referred to as adaptive attention.

## 4. Related Work

Image captioning is challenging because it not only requires the model to understand the contents of the image, but also deduces the salient parts and summarise them into a novel sentence through commonsense knowledge (Fang et al., 2014). There are two well-established approaches for this problem: the retrieval of existing captions, and the generation of novel captions (Fang et al., 2014).

**Retrieval-based model:** The main idea of a retrieval-based model is to project images and captions to a vector space, from which we can retrieve one with the other. This approach has been studied in a few recent papers (Socher et al., 2014; Ordonez et al., 2011; Farhadi et al., 2010). Farhadi et al. represented the space of the captions by triplets of <object, action, scene>. The model can then generate captions for the query image by retrieving whole image description via this meaning space obtained from some dataset (e.g. the UIUC Pascal Sentence data set) (Farhadi et al.,




(a) LSTM-attention (b) GRU-attention

*Figure 7.* Visualising attention map of the image when each word is generated. (bright indicates the attended regions, and corresponding word is located on the top)


(a) Left (b) Right

*Figure 8.* Examples of attending to the correct object. (bright indicates the attended regions, and corresponding word is coloured in the sentences.)

2010) . Its advantage is that generated captions have natural syntax, although on occasion might result miss out some salient contents of the image. Ordonez et al. attributed this problem to sparsity (Ordonez et al., 2011), and improved the model further by extending the object categories and enlarging the image content aspects with a much larger dataset.

RNN models are introduced in recent years to improve the representation of longer phrases. The Bag of Words approach is the most common way to represent sentences from word vectors, which simply just averages all the vectors. This might result in bad performance for some tasks, as it ignores the order of the words and gives them the same weight. After RNN models are developed to combine these word vectors and it has been noticed that they can capture a lot of syntactic structure of the sentence (Pollack, 1990; Socher et al., 2010). Based on that, Socher et al. provided a novel model for image captioning, which focuses on recognising the actions and agents that help to distinguish active and passive constructions (Socher et al., 2014).

However, although the retrieval-based approach can return well-written captions, it cannot create novel captions but can only modify similar captions that were retrieved (Xu et al., 2015). Hence, with the use of neural networks in the caption generation, the former approach falls behind, and also motivated us to investigate the performance of neural networks in image captioning.

**Novel caption generation model:** Kiros et al. firstly provided a multi-modal language model which can generate captions without templates, structured prediction or syntactic trees (Kiros et al., 2014a). They then explored further and proposed a new encoder-decoder model for caption generation (Kiros et al., 2014b). The encoder-decoder framework was inspired by several successes in the field of neural machine translation (Kalchbrenner & Blunsom, 2013). The idea behind this framework is that inputs which are in the source language are encoded in order to generate vector representations of the sequence. The representations are then decoded to produce an output in the target language by predicting the next word according to the vector and all previously predicted words. Image captioning can also be considered as a translation task: a translation of visual to

verbal information. Hence similarly, the encoder transforms the input image into feature representations and passes the vectors to the decoder. The decoder then generates a textual representation describing the image feature vectors. Kiros et al. used a CNN-LSTM encoder to learn a joint image-sentence embedding (Kiros et al., 2014a), then decoded them with a structure-content neural language model to generate captions (Kiros et al., 2014b).

On the other hand, Vinyals et al.only kept the CNN as the encoder, and represented the captions with embedding vectors (each word in the caption is associated with a fixed-length vector during training) (Vinyals et al., 2014). LSTM was used as the decoder to transform the representation to the description. Based on this show and tell project, Xu et al. introduced an attention-based model which learns the important image regions for caption generation (Xu et al., 2015). Their model also obtained the state-of-the-art performance on three benchmark datasets: Flickr8k, Flickr30k and MSCOCO. However, this attention-based model attends to the image at every time step, while not all words have corresponding signals, such as functional words "of" and "that". Consquently, the adaptive attention model was proposed (Lu et al., 2016). Instead of attending to the image for every word, the adaptive attention mechanism can decide whether to trust the visual signals or just the language model.

Our standard LSTM and GRU models are modified based on Vinyals's work (Vinyals et al., 2014), as this architecture has received great success at the COCO 2015 Captioning Challenge. We replaced the gating mechanism in the RNN decoder by GRU for better comparison with our LSTM baseline, while keeping the CNN encoder the same as in the paper. As the dataset was split randomly into fixed sizes, it is almost impossible to get the same image distribution and we can not directly compare the results quantitatively, but by visualising and judging from the captions generated, it is reasonable to conclude that our model works well on the task as discussed in Section 3.

Our attention-based LSTM and GRU models are inspired by the Show, Attend and Tell project (Xu et al., 2015) due to their straightforward implementation. As their attention-based model was trained for almost three days with more advanced resources compared to this piece of work, we had to reduce the dimensions of the image features due to the time and resource constraints. However, the results indicate improved performance compared to our non-attention models, and the attention mechanisms do attend to the correct objects as we discussed in Section 3.

As discussed above, there are some limitations about our attention mechanism, and the adaptive attention mechanism is a reasonable solution that can be further investigated. Besides, a recently proposed unconventional framework named Generative Adversarial Net (Goodfellow et al., 2014) is also considered promising for image captioning task. The GAN framework consists of two main components: a generator network and a discriminator network. The main idea behinds GANs is that the generator produces a caption when given an input image, the fake caption is passed to the discriminator along with the real caption. The discriminator then gives a fake/real score. The goal of the training process is to make generated captions and real captions indistinguishable by the discriminator. Shetty et al. applied this framework to the image captioning task and achieved comparable performance to the state-of-the-art in terms of the correctness of the captions (Shetty et al., 2017). On the other hand, Dai et al. pointed out that a major failure case using GAN can be the inclusion of incorrect details, such as counts (three/four people), which may be caused by insufficient sample data (Dai et al., 2017). Moreover, the focus on diversity and overall quality may also cause the generator to ignore the noise input and include more details for incorrect predictions. As the time and resources are constrained we did not try GANs for this assignment, but it is a promising potential aspect that can be explored in the future.

## 5. Conclusion

With the aim of exploring the impacts of neural networks on image captioning, we built four models to investigate the performance of neural network models: standard GRU/LSTM models and attention-based GRU/LSTM models.

Based on our results, we observed that the final scores of the standard LSTM and GRU gating models are similar. However, GRU gating and attention models show a trend for early convergence compared to our baseline model. Attention mechanisms also give a boost in performance compared with standard LSTM/GRU models. After tuning the hyperparameters, the best model we obtained is the attention LSTM model with 2 hidden layers, 400 hidden dimensions and 300 hidden attention dimensions, which achieved most improvement in metric scores compared to the baseline model. In addition to quantitative analysis, we also visualised the attention map along a time series when each word is generated in the caption. The mechanism did attend to the correct objects, and the captions generated contain more details when compared with non-attention models.

On the other hand, attention models do have its limitations. It was noticed that some words with little lexical meaning (which only contributes to the syntax of the sentence) do not have corresponding image contents. To address this, the adaptive attention mechanism, for example, can be an area of further research. Further work for improving the performance in image captioning may also include the Generative Adversarial Net, a newly proposed framework in the field.

## References


Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. URL http://arxiv.org/abs/1409.0473.

Bernardi, Raffaella, Çakici, Ruket, Elliott, Desmond, Erdem, Aykut, Erdem, Erkut, Ikizler-Cinbis, Nazli, Keller, Frank, Muscat, Adrian, and Plank, Barbara. Automatic description generation from images: A survey of models, datasets, and evaluation measures. *CoRR*, abs/1601.03896, 2016. URL http://arxiv.org/abs/1601.03896.

Cho, Kyunghyun, van Merrienboer, Bart, Gülçehre, Çaglar, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. URL http://arxiv.org/abs/1406.1078.

Chung, Junyoung, Gülçehre, Çaglar, Cho, KyungHyun, and Bengio, Yoshua. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. URL http://arxiv.org/abs/1412.3555.

Dai, Bo, Lin, Dahua, Urtasun, Raquel, and Fidler, Sanja. Towards diverse and natural image descriptions via a conditional GAN. *CoRR*, abs/1703.06029, 2017. URL http://arxiv.org/abs/1703.06029.

Fang, Hao, Gupta, Saurabh, Iandola, Forrest N., Srivastava, Rupesh Kumar, Deng, Li, Dollár, Piotr, Gao, Jianfeng, He, Xiaodong, Mitchell, Margaret, Platt, John C., Zitnick, C. Lawrence, and Zweig, Geoffrey. From captions to visual concepts and back. *CoRR*, abs/1411.4952, 2014. URL http://arxiv.org/abs/1411.4952.

Farhadi, Ali, Hejrati, Mohsen, Sadeghi, Mohammad Amin, Young, Peter, Rashtchian, Cyrus, Hockenmaier, Julia, and Forsyth, David. Every picture tells a story: Generating sentences from images. In Daniilidis, Kostas, Maragos, Petros, and Paragios, Nikos (eds.), *Computer Vision – ECCV 2010*, pp. 15–29, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15561-1.

Feng, Y. and Lapata, M. Automatic caption generation for news images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(4):797–812, April 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.118.

Glorot, Xavier and Bengio, Yoshua. Understanding the difficulty of training deep feedforward neural networks. In Teh, Yee Whye and Titterington, Mike (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL http://proceedings.mlr.press/v9/glorot10a.html.

Goodfellow, Ian J., Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron C., and Bengio, Yoshua. Generative adversarial networks. *CoRR*, abs/1406.2661, 2014. URL http://arxiv.org/abs/1406.2661.

Kalchbrenner, Nal and Blunsom, Phil. Recurrent continuous translation models. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, October 2013.

Kiros, Ryan, Salakhutdinov, Ruslan, and Zemel, Rich. Multimodal neural language models. In Xing, Eric P. and Jebara, Tony (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 595–603, Bejing, China, 22–24 Jun 2014a. PMLR. URL http://proceedings.mlr.press/v32/kiros14.html.

Kiros, Ryan, Salakhutdinov, Ruslan, and Zemel, Richard S. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539, 2014b. URL http://arxiv.org/abs/1411.2539.

Lu, Jiasen, Xiong, Caiming, Parikh, Devi, and Socher, Richard. Knowing when to look: Adaptive attention via A visual sentinel for image captioning. *CoRR*, abs/1612.01887, 2016. URL http://arxiv.org/abs/1612.01887.

Ordonez, Vicente, Kulkarni, Girish, and Berg, Tamara L. Im2text: Describing images using 1 million captioned photographs. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, pp. 1143–1151, USA, 2011. Curran Associates Inc. ISBN 978-1-61839-599-3. URL http://dl.acm.org/citation.cfm?id=2986459.2986587.

Pennington, Jeffrey, Socher, Richard, and Manning, Christopher D. Glove: Global vectors for word representation. In *In EMNLP*, 2014.

Pollack, J. B. Recursive distributed representations. *Artif. Intell.*, 46(1-2):77–105, November 1990. ISSN 0004-3702. doi: 10.1016/0004-3702(90)90005-K. URL http://dx.doi.org/10.1016/0004-3702(90)90005-K.

Rennie, Steven J., Marcheret, Etienne, Mroueh, Youssef, Ross, Jarret, and Goel, Vaibhava. Self-critical sequence training for image captioning. *CoRR*, abs/1612.00563, 2016. URL http://arxiv.org/abs/1612.00563.

Shetty, Rakshith, Rohrbach, Marcus, Hendricks, Lisa Anne, Fritz, Mario, and Schiele, Bernt. Speaking the same language: Matching machine to human captions by adversarial training. *CoRR*, abs/1703.10476, 2017. URL http://arxiv.org/abs/1703.10476.

Socher, Richard, Manning, Christopher D., and Ng, Andrew Y. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *In Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, 2010.

Socher, Richard, Karpathy, Andrej, Le, Quoc V., Manning, Christopher D., and Ng, Andrew Y. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014. ISSN 2307-387X. URL https://transacl.org/ojs/index.php/tacl/article/view/325.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott E., Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. URL http://arxiv.org/abs/1409.4842.

Vinyals, Oriol, Toshev, Alexander, Bengio, Samy, and Erhan, Dumitru. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555, 2014. URL http://arxiv.org/abs/1411.4555.

Xu, Kelvin, Ba, Jimmy, Kiros, Ryan, Cho, Kyunghyun, Courville, Aaron C., Salakhutdinov, Ruslan, Zemel, Richard S., and Bengio, Yoshua. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, abs/1502.03044, 2015. URL http://arxiv.org/abs/1502.03044.