Image Caption Generation using Recurrent Neural Networks (RNN)

Krishna Daamini Ellendula
Department of Electrical & Computer Engineering
University of California, San Diego
kellendula@ucsd.edu

Nishanth Rachakonda
Department of Computer Science
University of California, San Diego
nrachakonda@ucsd.edu

Pranav Khanna
Department of Computer Science
University of California, San Diego
pkhanna@ucsd.edu

Soumya Ganguly
Department of Mathematics
University of California, San Diego
s1gangul@ucsd.edu

Abstract

This project introduces the use of deep learning for image captioning tasks. The method is applied to the COCO dataset which is a large scale object detection, segmentation, and captioning dataset. In this project we use different deep learning architectures like LSTM + CNN model architecture and Vanilla RNN for experimentation. As the best result, we get a BLEU-1 score of 67.31 and BLEU-4 score of 8 using RNN and LSTM. We also demonstrate the effect of different hyperparameters on efficiency of these models.

1 Introduction:

Deep learning models have proven to be effective in modeling complex real-world data and have found widespread use in a variety of applications especially its numerous applications in image processing. One of the most popular use of deep learning models is generating captions of images. The problem of image captioning is at the intersection of Computer Vision and Natural Language Processing. Given an image, the model needs to generate a sensible, accurate, and grammatically correct caption for the image. It requires both methods from computer vision to understand the content of the image and a language model from the field of natural language processing to turn the understanding of the image into words in the right order. The ability to generate coherent descriptions has utility in image classification and annotation for large sets of data, particularly with regards to image search engines. Additionally, image captioning can act as a stepping stone to further tasks such as video classification and image and text inference.

For this project, we will use recurrent neural networks to deal with data that has temporal structure. A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes can create a cycle, allowing output from some nodes to affect subsequent input to the same nodes. This allows it to exhibit temporal dynamic behavior. In order to generate captions, we use an encoder-decoder architecture for this assignment. We use a pre-trained convolutional network (convnet) as the encoder (ResNet 50) and an RNN model as the decoder. We kept the convnet weights fixed, and trained an embedding layer to map into the RNN. The encoder takes the image as input and encodes it into a vector of feature values. This will then be passed through a linear layer for providing the input to the RNN . In the RNN, it will be trained to predict the next word at each step. We used both LSTM (Long Short Term Memory network) and a vanilla RNN for our model in the decoder part. The training uses images and several captions for each image generated by humans with the training set being Common Objects in Context (COCO) dataset. After training, the network

is run in generative mode to generate captions on images it has never seen before. We will show how different architectures affect the accuracy and plausibility of our image captions measured by BLEU scores. As the best result we get a BLEU-1 score of 67.31 and BLEU-4 score of 8 using RNN and LSTM.

2 Background/Related Work: (remaining)

In the lecture slides of CSE 251B by Cottrell [5], [4], the essential theory behind convolutional neural nets is discussed which provides us with the base structure of the networks used in this project. By the same author, we have the slides of [2] to get idea about how to use RNNs for 'Generative modelling'. In the slides of [3], the former author discusses the fundamentals of LSTM (Long Short Term Memory network) which forms the basis of one of the models here.

We use 'Transfer Learning' for generating image features that can be captioned. The first introduction to transfer learning techniques were described by Stevo Bozinovski and Ante Fulgosi in 1976, details of which can be found in [1]. Some of the earliest works were explicitely 'Recurrent Neural networks' were mentioned are [9] and [7]. We also had immense help from Andrej Karpathy's blog on RNNs called 'The Unreasonable Effectiveness of Recurrent Neural Networks'. The link of the blog can be found here. LSTM was first introduced in the paper [6] by Sepp Hochreiter and Jürgen Schmidhuber in 1997. BLEU (bilingual evaluation understudy) is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. We use BLEU1 and BLEU4 score to calculate the effectiveness of our image captioning when compared to a human. Many useful resources for learning about BLEU scores can be found here, here and in [8].

Transfer learning [1] aids in generating image features without the need to learn these features separately. This helps in decreasing the size of the network, which decreases the number of trainable parameters. This aids the model in learning the image captions better.

3 Methods

3.1 Architecture for baseline(LSTM) model:

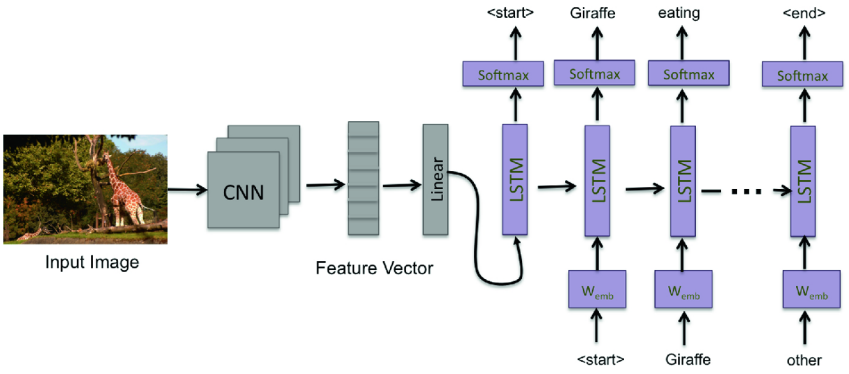


Figure 1: Architecture of the baseline(LSTM) model

Figure 1 shows the architecture of our LSTM based model. In the encoder, we used a pretrained convolutional network, ResNet-50, which uses a combination of convolutional layers, pooling layers, fully connected layers, and shortcut connections. The last layer was removed, and a trainable linear layer was added to adapt it to the given task. The encoder generates a feature vector of a fixed size for each image which is fed into the LSTM. A random square section of 256×256 for each image was cropped to be used as input to the encoder.

For LSTM decoder, first the hidden state and the cell state are initialized as zero. The feature vector of the image is initially fed into the LSTM. Then for each caption an embedding vector is

generated which is fed in the next time. Instead of using the output from the last LSTM cell, a technique called 'Teacher forcing' is being used here. Teacher forcing uses the teaching signal from the training dataset at the current time step, target(t), as input in the next time step, thus giving $x(t+1) =$ target(t), rather than the output $y(t)$ generated by the network. The outputs of LSTM will be fed to a linear layer to scale up the dimensions. In Pytorch, an `nn.LSTM` module is being implemented here.

3.2 Architecture for Vanilla RNN model:

In this project, we experimented with the Vanilla RNN model. A Recurrent Neural Network is a class of neural networks where the output from the previous time step is fed as input to the current time step. This results in cyclic connections between nodes, unlike traditional feed-forward networks, and the network uses Backpropagation through time(BPTT) to update the gradients. These models are generally used to train sequential data such as speech recognition, and image captioning.

In this architecture, we replaced the LSTM module in the baseline model with a simple RNN module. This module has 2 hidden layers with ReLU activations and takes two inputs at the current time step t: the current feature $x(t)$ and hidden state $h(t-1)$. We again follow 'Teacher Forcing' by using the teaching signal from the training dataset at the previous time step, $target(t-1)$, as input in the current time step i.e., $x(t) = target(t-1)$.

The same encoder from the baseline model is used for this method. The main difference is that the LSTM module takes hidden states $h(t-1)$ and cell states $c(t-1)$ as inputs from the previous step, whereas the Vanilla RNN module takes only hidden states $h(t-1)$ as inputs. Words from the captions are converted to embedded features and fed as inputs at each time step $x(t)$.

3.3 Architecture 2:

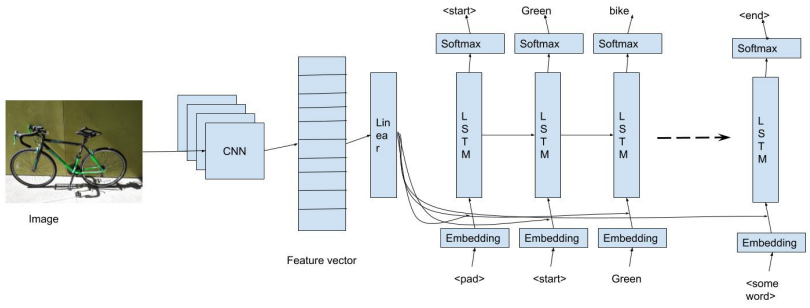


Figure 2: Architecture of the modified model

In this subsection of experiments we use the model architecture as illustrated above in Figure 2. For the encoder part again, we use the same Resnet-50 architecture as before. For the decoder part in this model architecture, we use the same LSTM architecture as in the baseline model, but with a crucial difference. This model generates captions by passing the input image and the input word token at the first timestep to the LSTM decoder to learn how to generate the "¡start¿" token. From the second timestep, the model inputs the output from the previous timestep, concatenated with the image features, to the LSTM decoder using 'Teacher Forcing' during training and sampling during generation. The output is then passed through the word embedding layer, LSTM layer, and a linear layer before being inputted into the softmax layer to generate the predicted words for the current timestep.

3.4 Method of sampling outputs from decoder:

For generating captions there are two options to find the next best word given the output of the network from the previous step.

- **Deterministic:** In the deterministic model, we find the vocabulary word with the highest probability given the last network output as our input. Thus, we are trying to find:

$$\text{next\_word}_{det} = \mathbf{argmax_y} \mathbf{P}(\text{next\_word} = \mathbf{y} | \text{input\_word = x})$$

- **Stochastic:** This involves creating a distribution based on the previous network output. Once this distribution is created, we then sample a vocabulary word from the distribution. Thus, we will not always get the same word as our sampled output (unless our parameter 'temperature' below, is too low). However, words with higher probability will be sampled more often. To get the distribution, we calculate the weighted softmax of the outputs: $y^j = exp(o^j/\tau)/\sum_n exp(o^n/\tau)$, where o^j is the output from last layer, n is the size of the vocabulary, and τ is a configurable hyperparameter called 'temperature', as described in class. The temperature controls how stochastic the sampling is: As the temperature approaches 0, the distribution is nearly deterministic, as it goes to infinity, the distribution is completely uniform. A typical value for the temperature is below 1, although 1 is used during training. The temperature is only used during generation here.

3.5 Method of obtaining word embeddings:

For training the RNN, we use teacher forcing where we feed the caption words from the training dataset at each time step to the LSTM module. The LSTM module takes fixed dimensional vectors as inputs. This mapping of all the words from the vocabulary to a vector space is called embedding and the resulting vector space is called embedding space. It mainly serves two purposes:

- Represent words as vectors so that they can be used for training the RNN.
- Learn the semantic relations of the words and place similar words closer in the embedding space.

Each word is first one-hot encoded based on the vocabulary and then converted to a lower dimensional embedding using a fully connected layer. The training data set contains a vocabulary of 14463 words and the decoder network's input size is 300. So, these words need to be converted to 300-dimensional feature vectors. For this purpose, we used Pytorch's `nn.Embedding` module with `num_embeddings` = 14463 and `embedding_dim` = 300 as inputs. The weights of the module of shape (`num_embeddings`,`embedding_dim`) are trained by Backpropagation.

3.5.1 Method of hyperparameter tuning:

For all the models, we have tuned on the hyperparamters like embedding size, LSTM hidden state size, image feature vector size and learning rate using random search algorithm. For embedding size, we tried 200, 250, 300, 350, 400. For LSTM hidden unit size, we tried 128, 256, 512. For image feature vector size we tried 200, 250, 300, 350, 400 and for learning rate we tried $5 \times 10^{-4}, 1 \times 10^{-3}$.

4 Results:

4.1 Best hyperparameters for each model:

As seen above, we implement a random search algorithm for hyperparamter tuning. The best results of hyperparamter tuning, according to BLEU-1 scores, among all the searches we conducted, are listed in the following table. In all these experiments we kept batchsize as 64, temperature as 0.1, used Adam optimizer and stochastic sampling for generating captions.

Table 1: Best set of hyperparameters for each model based on BLEU-1 score

Model	Embedding Size	LSTM Hidden state size	Image feature vector size	Learning rate
Baseline (LSTM) Model	400	512	400	5×10^{-4}
Vanilla RNN Model	400	512	400	5×10^{-4}
Architecture 2	200	512	300	1×10^{-3}

4.2 Loss Plots for all models:

In the following 3 plots, the training and validation loss plots with respect to number of epochs can be found. We give the plots for all the 3 different architectures/models considered here.

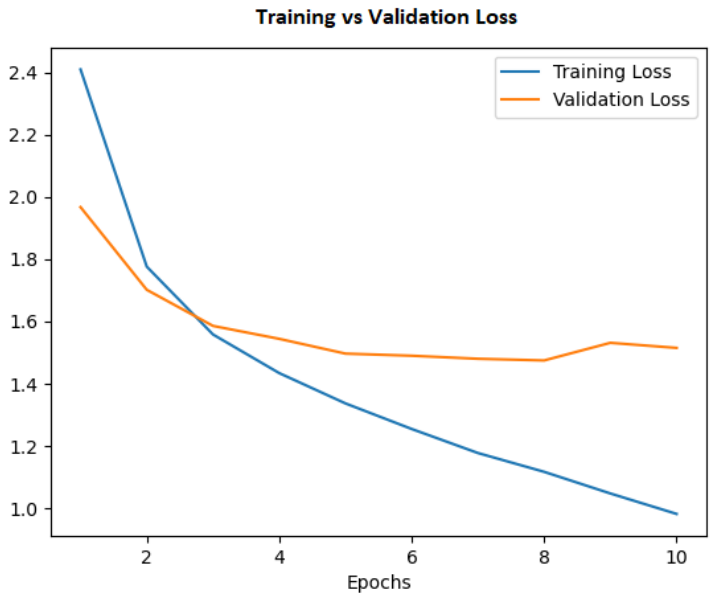


Figure 3: Training and Validation loss plots in 10 epochs with Baseline LSTM model for a. learning rate=0.0005, b. batch size=64, c. embedding size=400, d. hidden state size=512, e. image feature size = 400

4.3 BLEU-1 and BLEU-4 scores for all models:

The BLEU-1 and BLEU-4 scores for each of the models, using the best hyperparameter sets as described above, can be found in the following table.

Table 2: Best BLEU-1 and BLEU-4 scores for each model

Model	BLEU-1 Score	BLEU-4 Score
Baseline (LSTM) Model	66.05222	7.71613
Vanilla RNN Model	63.75161	6.73715
Architecture 2	67.31017	8.00097

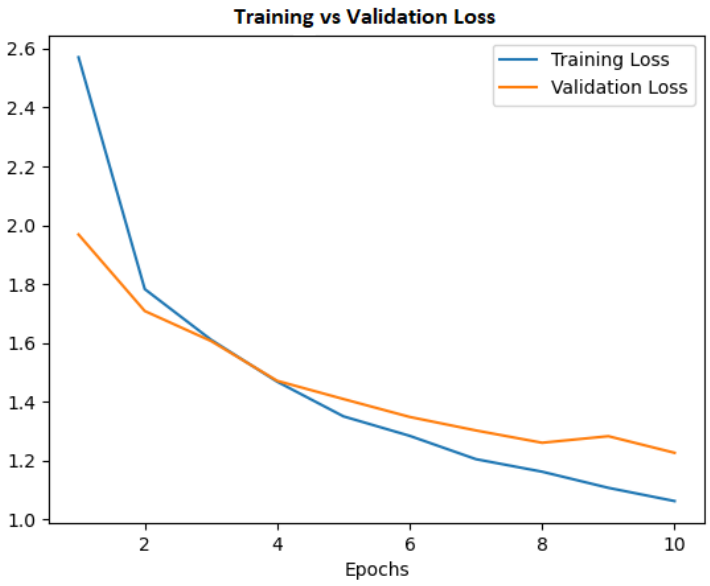


Figure 4: Training and Validation loss plots in 10 epochs with Vanilla RNN model for a. learning rate=0.0005, b. batch size=64, c. embedding size=400, d. hidden state size=512, e. image feature size = 400

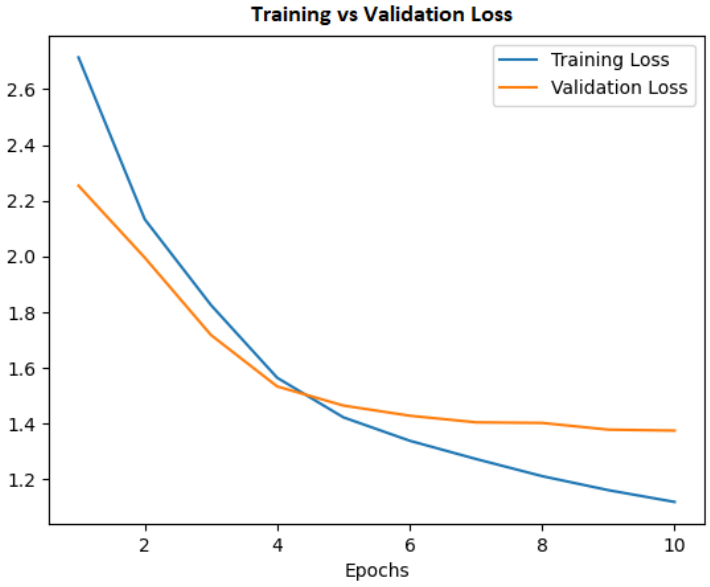


Figure 5: Training and Validation loss plots in 10 epochs with Architecture 2 model for a. learning rate=0.001, b. batch size=64, c. embedding size=200, d. hidden state size=512, e. image feature size = 300

4.4 Examples of captioned images:

We now provide three examples of images that resulted in 'good' captions, and three images that resulted in 'bad' captions for each of the three models. The properties 'good' and 'bad' are measured

with respect to the BLEU-1 score obtained for the images. For each image used here, we have chosen 'good' and 'bad' using a temperature of 0.4. We also provide captions generated on these images by using 1) the deterministic approach, and 2) using a very high and a very low temperature (i.e. 5 and 0.001).

Actual Captions

- a pair of skis next to a pair of ski boots.
- a pair of green , yellow and orange skis.
- skis and ski boots sit together on a tiled floor.
- a pair of green yellow and orange skis sit next to a pair of green ski boots.
- skis and ski boots sit next to each other on floor.

Predicted Captions

- a pair of snowboards and a pair of shoes are on a bed. (temp 0.001)
- a pair of snowboards and a pair of shoes are on a floor. (temp 0.4)
- magenta hydant fremont doll appearing room/ mature binoculars karmill garnished affixed run other evening yello mountains cleaner lassie latin designates sandwiches. (temp 5)

Figure 6: 'Image with 'bad' captions 1': using Baseline Model

Actual Captions

- a person on white surfboard riding a wave next to a cliff.
- a man riding a wave on top of a surfboard.
- a person is surfing in a on a wave.
- silhouette of a surfer catching a wave in the distance
- the person is in the water having fun.

Predicted Captions

- a man is surfing on a wave in the ocean. (determenistic)
- a man is surfing on a wave in the ocean. (temp 0.001)
- a man riding a surfboard on a wave in the ocean. (temp 0.4)
- cheer received ramp expensive unwanted names art reporter divider league toilette bowls by representation appliances visit scalpel loungers backsides situation. (temp 5)

Figure 7: 'Image with 'good' captions 1': using Baseline Model

Actual Captions

- an animal biting a yellow frisbee next to another man.
- otters investigating a frisbee thrown into their naturalistic enclosure.
- two dark colored animals with a yellow plastic disc.
- two otters that are playing with a frisbee.
- two small otters playing with a yellow frisbee.

Predicted Captions

- a large brown bear standing on top of a green field. (deterministic)
- a large brown bear standing on top of a green field. (temp 0.001)
- a squirrel standing next to a pile of sheep. (temp 0.4)
- deep represent falling creature eating buried eyes wildwood sideburns sunflower umping enclosed surrounding napkins flat-bread cliffs arrangement gauchos tilting long crepe. (temp 5)

Figure 8: 'Image with 'bad' captions 1': using Baseline Model

Actual Captions

- two people standing next to each other on top of the snow.
- two people standing in the snow with skies on the ground and standing in the snow.
- two people standing in the snow with ski equipment.
- two people with hats standing next to ski gear.
- skiers with equipment looking at snow covered rise in distance.

Predicted Captions

- a man is standing in the snow on skis. (determenistic)
- a man is standing in the snow on skis. (temp 0.001)
- a man is standing in the snow on skis. (temp 0.4)
- peter conveyor fuselage hips wizard rice playin peers lacquer anymore well sets seaside peepee figure ledge beet only door prepared treatment. (temp 5)

Figure 9: 'Image with 'bad' captions 1': using Baseline Model

Actual Captions

- a man riding a wave on top of a surfboard.
- the man is surfing on his surfboard in the ocean.
- a man in blue shirt surfing on a white surfboard.
- a man on a surfboard riding a wave in the ocean.
- a person riding a surfboard on an ocean wave

Predicted Caption

- a man riding a wave on top of a surfboard. (determenistic)
- a man riding a wave on top of a surfboard. (temp 0.001)
- a man riding a wave on top of a surfboard. (temp 0.4)
- stop figure trots motorocycles seattle lessons shaker platform streets towels forage interacting rotten keish standing some cartoons festivity drip missiles barricade. (temp 5)

Figure 10: 'Image with 'good' captions 1': using Baseline Model

Actual Captions

- a man wearing a wet suit riding the wave.
- a person riding a wave on top of a surfboard.
- a surfer rides a small wave in the ocean.
- a man riding a wave on a blue ocean.
- a young man ridding a small wave on a surfboard.

Predicted Captions

- a man riding a wave on top of a surfboard. (determenistic)
- a man riding a wave on top of a surfboard. (temp 0.001)
- a man riding a wave on top of a surfboard. (temp 0.4)
- omelette piles christmas fridgerator cutting shirt knives directly pasadena romaine blankets pants backed bites escorted traffic cloud jungle dvd bags pin. (temp 5)

Figure 11: 'Image with 'good' captions 1': using Baseline Model

Actual Captions

- a pair of skis next to a pair of ski boots.
- a pair of green , yellow and orange skis.
- skis and ski boots sit together on a tiled floor.
- a pair of green yellow and orange skis sit next to a pair of green ski boots.
- skis and ski boots sit next to each other on floor.

Predicted Captions

- a snow covered covered covered in white and white and white. (determenistic)
- a snow covered covered covered in white and white and white. (temp 0.001)
- a snow covered covered covered in white and white and white. (temp 0.4)
- venturing pets sailboats depicts balcony performs here parked next curly brimmed lawn seating color hover arching crow shiny incoming not sigh.(temp 5)

Figure 12: 'Image with 'bad' captions 1': using Vanilla RNN Model

Actual Captions

- a person on white surfboard riding a wave next to a cliff.
- a man riding a wave on top of a surfboard.
- a person is surfing in a on a wave.
- silhouette of a surfer catching a wave in the distance
- the person is in the water having fun.

Predicted Captions

- a man riding a wave on top of a surfboard. (deterministic)
- a man riding a wave on top of a surfboard. (temp 0.4)
- a man riidng a wave on top of a surfboard. (temp 0.001)
- maine leaved pull common jerseys junction fodder varnished liking restricted i milking lab softball s. briefcase stepladder nosing biting park disembarking (temp 5)

Figure 13: 'Image with 'good' captions 1': using Vanilla RNN Model

Actual Captions

- an animal biting a yellow frisbee next to another man.
- otters investigating a frisbee thrown into their naturalistic enclosure.
- two dark colored animals with a yellow plastic disc.
- two otters that are playing with a frisbee.
- two small otters playing with a yellow frisbee.

Predicted Captions

- a bear is sitting on a tree branch. (determenistic)
- a bear is sitting on a tree branch. (temp 0.001)
- a bear is standing in the grass near a tree. (temp 0.4)
- communal searches woth duty dawn playing wooly motercycles climbing ties guy annoyed shaves sparsely foliage torsos patch astroturf heated western world.(temp 5)

Figure 14: 'Image with 'good' captions 1': using Vanilla RNN Model

Actual Captions

- two people standing next to each other on top of the snow.
- two people standing in the snow with skies on the ground and standing in the snow.
- two people standing in the snow with ski equipment.
- two people with hats standing next to ski gear.
- skiers with equipment looking at snow covered rise in distance.

Predicted Captions

- a man is skiing down a snowy hill. (determenistic)
- a man in a white jacket and a snowboard on a snow covered slope. (temp 0.001)
- a man in a blue jacket is skiing down a hill. (temp 0.4)
- carnation anthers branches overlooks slap computing spots followed effect makeshift no fastball noses stoic direction a australian mounted maneuvering fireworks nathans.(temp 5)

Figure 15: 'Image with 'bad' captions 1': using Vanilla RNN Model

Actual Captions

- a man riding a wave on top of a surfboard.
- the man is surfing on his surfboard in the ocean.
- a man in blue shirt surfing on a white surfboard.
- a man on a surfboard riding a wave in the ocean.
- a person riding a surfboard on an ocean wave

Predicted Caption

- a man riding a wave on top of a surfboard.(determenistic)
- a man riding a wave on top of a surfboard.(temp 0.001)
- a man riding a wave on top of a surfboard.(temp 0.4)
- priced trio gathers no crowds bater piano towers wrangling advertises skylight parasail butchers saucer riderless dry exposed cocker microphones main powerful. (temp 5)

Figure 16: 'Image with 'good' captions 1': using Vanilla RNN Model

Actual Captions

- a man wearing a wet suit riding the wave.
- a person riding a wave on top of a surfboard.
- a surfer rides a small wave in the ocean.
- a man riding a wave on a blue ocean.
- a young man ridding a smal wave on a surfboard.

Predicted Caption

- a man riding a wave on top of a surfboard.(determenistic)
- a man riding a wave on top of a surfboard.(temp 0.001)
- a man riding a wave on top of a surfboard.(temp 0.4)
- vac toiled decline handicapped fill sunset ink masquerade brahma fondue knobs toa captivity elevating alcove bent battenburg amount ontop purple noodles. (temp 5)

Figure 17: 'Image with 'good' captions 1': using Vanilla RNN Model

Actual Captions

- a woman standing on a tennis court holding a racquet.
- a woman wearing a red tennis outfit holds her racket out to the side near a tennis ball on a tennis court.
- a tennis player hitting a ball on the tennis court.
- a woman hits a tennis ball at the end of the court.
- a woman preparing to hit a tennis ball.

Predicted Captions

- a woman swinging a tennis racquet on a tennis court. (determenistic)
- a woman swinging a tennis racquet on a tennis court. (temp 0.001)
- a woman holding a tennis racquet on a tennis court. (temp 0.4)
- scuba neon trying squatting rumpled perching coordinating bored sweatpants lopsided itself dental peripheral decoratively fogy are opens slope out than dead. (temp 5)

Figure 18: 'Image with 'good' captions 1': using Architecture 2

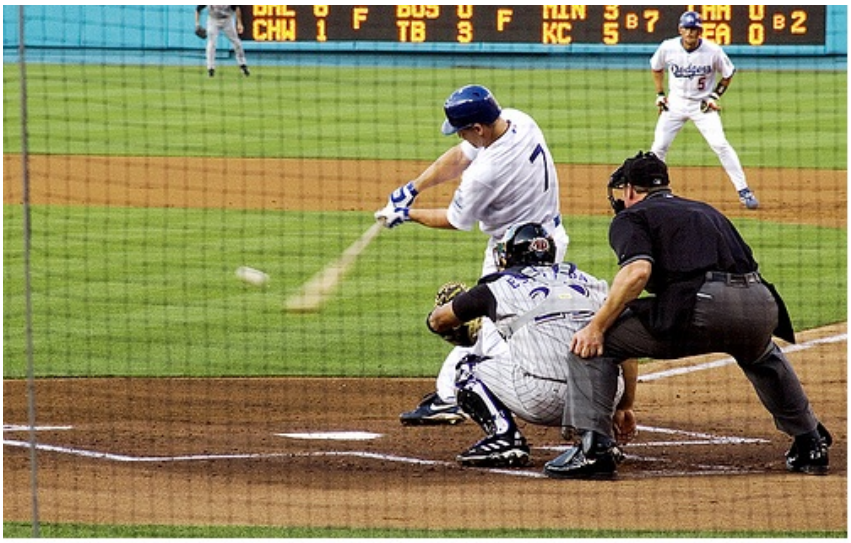
Actual Captions

- people at the baggage claim area of an airport.
- three people standing before airport counters below airport signs.
- people standing at counters of booths being served.
- the baggage delivery section of an air port.
- patrons are going to the shops of an airport.

Predicted Captions

- a man is standing in front of a large store. (determenistic)
- a man is standing in front of a large store. (temp 0.001)
- a person is standing in front of a cart of food. (temp 0.4)
- movable wiped somewhere outdoors grill sightseeing photoshoot assistive lays advertising asphalt swarm city oin vacuuming dock dying preen cell pushing pre. (temp 5)

Figure 19: 'Image with 'bad' captions 1': using Architecture 2



Actual Captions

- a left handed baseball player swinging a bat in front of a catcher and umpire.
- a baseball player swinging a bat at a ball.
- a man hitting a baseball in a professional baseball game.
- baseball player hitting a ball with a baseball bat.
- a baseball batter trying to hit a baseball.

Predicted Captions

- a baseball player swinging a bat at a ball. (determenistic)
- a baseball player swinging a bat at a ball. (temp 0.001)
- a baseball player is swinging a bat at a baseball game. (temp 0.4)
- plug busses dr smiling driveway birdcage turtles protests breed stickers visible tires counter-tops barclays tobacco crazy spout ship generic dragon crystal. (temp 5)

Figure 20: 'Image with 'good' captions 1': using Architecture 2

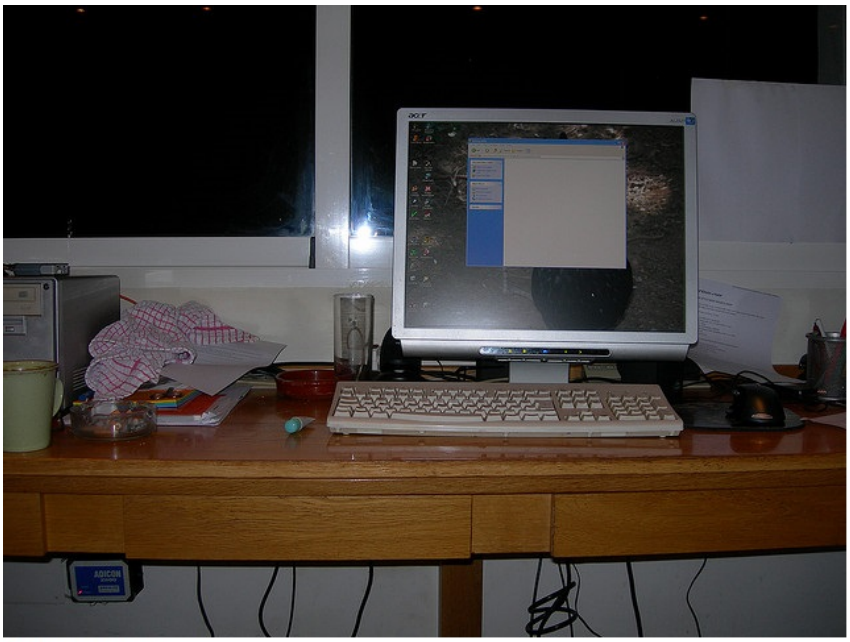
Actual Captions

- many fire hydrants made to look like cartoon characters.
- colorful fire hydrants are set on a sandy surface.
- many cartoon modeled objects are in the sand.
- six fire hydrants painted as cute cartoon characters.
- a group of whimsically yet creepy-looking fire hydrants.

Predicted Captions

- a woman sitting on a bench with a teddy bear. (determenistic)
- a woman sitting on a bench with a teddy bear. (temp 0.001)
- a woman sitting on a bench with a teddy bear. (temp 0.4)
- primitive cabinet allowed celebrating amused glances assembles shaker pine happiness connected bonzai grilled materials into plumage birdcage couches potty warning soon. (temp 5)

Figure 21: 'Image with 'bad' captions 1': using Architecture 2



Actual Captions

- a desktop computer sitting on top of a wooden desk.
- a computer is turned on on top of a desk.
- a desktop computer sits upon a cluttered desk.
- a desk containing a computer monitor and keyboard.
- computer station with one computer and many accessories.

Predicted Captions

- a computer monitor sitting on top of a desk. (determenistic)
- a computer monitor sitting on top of a desk. (temp 0.001)
- a desktop computer computer sitting on a desk. (temp 0.4)
- babies hoagie guiding colorless scratch festooned soon burning curtained costume greater rigged sst think somebody material pottery hours against shapes starting. (temp 5)

Figure 22: 'Image with 'good' captions 1': using Architecture 2

Actual Captions

- glass shelves on a display with tags on items.
- a glass case with shelves and various items displayed.
- a glass shelf with a clock flask wooden box and various other objects.
- a glass display case contains wooden boxes and trinkets like flasks and pocket knives.
- relics are positioned on glass shelves with a digital piece.

Predicted Captions

- a kitchen with a lot of items and a glass of tea. (determenistic)
- a kitchen with a lot of items and a glass of tea. (temp 0.001)
- a kitchen with a lot of doughnuts and a few bottles of coffee and a glass of coffee. (temp 0.4)
- pursing bundles shadows visible indoors mac fender thumbs oats spraying thermometer airstrip gestures spotlights surfboarding be steps specially violet struts lazing. (temp 5)

Figure 23: 'Image with 'bad' captions 1': using Architecture 2

5 Discussion:

5.1 Baseline LSTM Model

With the baseline model we experimented with different combinations of hyperparameters like varying the learning rate, number of hidden units in LSTM and the word embedding size. During our experimentation we observed that the baseline models performs best when the learning rate is set to 0.0005, the word embedding size is kept at 400 and the LSTM has 512 hidden units and gives a BLEU-1 score of 66.05 and BLEU-4 score of 7.71. We also notice in our experiments that as our word embedding size increases the performance of the baseline model gets better.

5.2 Vanilla RNN Model

We also conducted similar experiments with the Vanilla RNN model by altering hyperparameters like varying the learning rate, number of hidden units in LSTM and the word embedding size. During our experimentation we observed that the baseline models performs best when the learning

rate is set to 0.0005, the word embedding size is kept at 400 and the LSTM has 512 hidden units and gives a BLEU-1 score of 63.75 and BLEU-4 score of 6.73. We notice that the the validation loss is much less for the vanilla RNN model than when compared to the baseline model, however the baseline model has better BLEU scores than the vanilla RNN model.

5.3 Architecture 2

We also conducted similar experiments on Architecture 2. During our experimentation we observed that the baseline models perform best when the learning rate is set to 0.001, the word embedding size is kept at 200 and the LSTM has 512 hidden units and gives a BLEU-1 score of 67.31 and BLEU-4 score of 8.00.

References

[1] Stevo Bozinovski. Reminder of the first paper on transfer learning in neural networks, 1976. *Informatica*, 44, 09 2020.

[2] Gary Cottrell. Generative modeling with rnns. *CSE 251B Lecture Slides*, Fall 2022(1):1–36, 2022.

[3] Gary Cottrell. Modeling sequences(recurrent networks). *CSE 251B Lecture Slides*, Fall 2022(1):52–60, 2022.

[4] Gary Cottrell. Convolutional networks, part ii. *CSE 251B Lecture Slides*, Winter 2023(2):1–64, 2023.

[5] Gary Cottrell. Introduction to convolutional networks. *CSE 251B Lecture Slides*, Winter 2023(1):1–71, 2023.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[7] Michael I. Jordan. Chapter 25 - serial order: A parallel distributed processing approach. In John W. Donahoe and Vivian Packard Dorsel, editors, *Neural-Network Models of Cognition*, volume 121 of *Advances in Psychology*, pages 471–495. North-Holland, 1997.

[8] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

[9] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. 1986.

Team Contribution:

The project is a result of a team effort where both of us contributed as a group. Both ran the codes and provided necessary data for plotting and making tables in this report and both actively participated in hyperparameter tuning for the momentum method and finding optimal early stop epoch. However, roughly the contribution of each team member is listed below:

- Nishanth implemented train loop, caption generations, ResNet50, architecture-1, Vanilla RNN and architecture-2 models. Experimented on various hyper parameter models. Helped in writing discuss section part of the report.
- Soumya primarily worked on implementing the codes for resnet-50, architecture-1. Experimented on various hyperparameter models. Helped in writing the introduction, related work, results and methodologies of the report.
- Pranav worked in running the experiments for the baseline model and vanilla RNN model. He explored the effect of varying hyperparameters on model efficiency along with Nishant, Daamini and Soumya. In the report Pranav wrote the discussion of results section, abstract and description of Architecture 2.

- Daamini worked on generating captions, train loop and resnet-50. Experimented on various hyperparameter models. Helped in writing the introduction, related work, results and methodologies of the report.