Image Captioning through generative LSTM/RNN

Du Xiang
Halıcıoğlu Data Science Institute
University of California
La Jolla, CA 92093
dxiang@ucsd.edu

Arthur Wang
Computer Science Department
University of California
La Jolla, CA 92093
tiw032@ucsd.edu

Qiwen Zhang
Halıcıoğlu Data Science Institute
University of California
La Jolla, CA 92093
q2zhang@ucsd.edu

Judy Yang
Halıcıoğlu Data Science Institute
University of California
La Jolla, CA 92093
s7yang@ucsd.edu

Abstract

In this work, we focused our efforts on generating natural language captions from an image as input by using an encoder-decoder framework. As for encoder, we used a pre-trained Convolutional Neural Network ResNet50 model [4] using the Torchvision module from PyTorch [11]. The encoder will take images from the COCO dataset [8] and feed into the decoder, which will later serve as the generator for the captions. The decoders we have tried include two architectures of the Long Term Short Term memory network (LSTM) [5] and a Recurrent Neural Network (RNN). As a result, our best model for the first LSTM architecture achieved a BLEU-1 score of 67.77, BLEU-4 score of 8.97. Our best model for the second LSTM architecture achieved a BLEU-1 score of 69.05, BLEU-4 score of 8.74. The RNN architecture achieved a BLEU-1 of 65.22, BLEU-4 of 7.05. We found that LSTM as the decoder performs better than vanilla RNN which shows how LSTM doesn't suffer from gradient diminishing problem which in turn performs better in tasks like text generation than vanilla RNNs. Meanwhile, we notice that the second architecture of LSTM which feeds the image information at every time step performs better than the first LSTM which only feeds in the image embeddings at the first time step.

1 Introduction

Recently, as the AI and Machine Learning field have grown rapidly, natural language generative models are becoming more and more popular. Specifically, image captioning task is gaining more attention and becomes a good benchmark that combines both computer vision and natural language processing skills. In the actual application perspective, image captioning benefits the visually impaired people, and it also serves an important role for developing virtual assistants, transforming image information to text information, etc. Given the importance and interesting application of the task, we are motivated to tackle the problem using the COCO dataset[8].

In order to accomplish the generation of natural language captions from images, we must find a way to connect the information of the image to the captions or words. We know previously that Convolutional Neural Network can effectively extract discriminating information from the image pixels, and LSTM is really good at capturing the previous information (good for language inference in this case). Therefore, we will use a encoder-decoder framework where the encoder of the network is a CNN and the decoder of the network is a RNN or LSTM.

2 Related Work

While image captioning requires the corporation of image recognition and image textual description generating, techniques of both natural language processing and computer vision are needed. Research regarding image captioning methodology can be roughly categorized into three categories, which are template-based approaches, retrieval-based approaches, and novel image caption generation approaches.

The template-based approach is considered as the simplest method for image captioning. It has the pre-determined templates with several blanks to fill in [3]. Conditional random field (CRF) was used to identify objects, attributes and prepositions of image content and make predictions about the best label[3].The drawback of this method is that a pre-determined template is needed for every image, and only a few words can be varied each time, so there is a lack of flexibility for tasks. Meanwhile, the retrieval-based approach chooses semantically similar sentences from a pool of sentences or refers to similar images recognized by the algorithm [13] This method also requires a pre-genearted sentence pool and cannot produce accurate description for the images if no sufficient and accurate semantic resources are given. Apart from these two, our project makes use of the novel caption generation approaches, through which we apply deep learning strategy and machine learning to visually recognize the images and generate captions for them. The common way using this methodology to interpret visual content can be achieved through using CNN (convolutional neural network)as the encoder for image classification task and RNN (recurrent neural network) or LSTM (long short-term memory) as the decoder for caption sentence generating [15]. There are a wide range of choices for the specific encoders and decoders. Kiro et al. first introduces the multimodal neural language model to implement image captioning tasks with neural network [6]. Then they also designed an endoder-decoder pipeline so that sentences can be encodesd by LSTM and decoded by SC-NLM (structure-content neural language model) [7]. After that, DT-RNN (dependency tree-recursive neural network) was used to embed sentence into a vector space so as to retrieve, and then m-RNN was introduced to displace the feed-forward neural language model[12, 9]. With the popularity of this field increases, recently more mechanisms are proposed. Fang et al. introduced the multi-instance learning and traditional maximum-entropy language model to generate the descriptions, and Chen et al. introduced the visual representation learning with RNN [2, 1]. Attention mechanism was also introduced. For each layer of decoders, attention towards the image was re-adjusted so the model weigh different areas of the image differently [16]. Our project would focuses on the basic CNN-RNN and CNN-LSTM approach.

3 Methods

In this project, we developed three groups of models, which are ResNet50-LSTM model, ResNet50-RNN model, and ResNet50-LSTM with image input of each decoder layer. We developed our models to serve for the image captioning purpose. The details of each model are explained below.

3.1 Baseline LSTM

For our baseline model, we first employ ResNet50, which is a convolutional neural network, to encode the input image to a linear vector of its features and then apply this linear feature vector into our two layers LSTM decoder.

The basic structure of the LSTM model can be represented by these listed equations. Here, x_t represents each layer of LSTM, and it returns p_{t+1}. The output of each layer is in the format of a tuple (m_t, c_t), and it is passed as the current hidden state to the next hidden state.

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1} + b_f) \tag{2}$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1} + b_o) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{cx}x_t + W_{cm}m_{t-1}) \tag{4}$$

$$m_t = o_t \odot c_t \tag{5}$$
$$p_{t+1} = Softmax(m_t) \tag{6}$$

There are three types of gate to control information flow in LSTM: the forget gate, input gate, and the output gate. The gates use hyperbolic tangent and sigmoid activation functions. The forget gate is displayed as equation 2, it decides what information needs attention and what can be ignored. The information is passed from current input x_t and the hidden state or the input m_{t-1}. The Sigmoid function generates values between 0 and 1, and multiply with the cell state data, in order to select the information necessary for the next step. The input gate processes the information regarding the current state x_t and the hidden state or the input m_{t-1}, and that the same information of the hidden state and current state is being passed through the $tanh$ function, which produces the result of c_t to be between -1 and 1, and then this result is used for the point-by-point multiplication. The output gate determines the value of the next hidden state. As sigmoid funtion is appliedto process the current state and previous hidden state, and the new information produced by the current cell state is passed to the tanh function. Together these three gates build up the basic LSTM architecture.

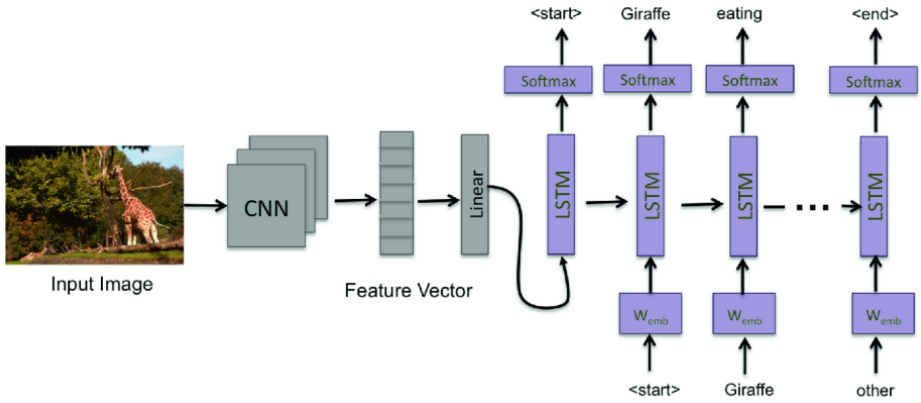

Figure 1: LSTM: Architecture 1 [14]

While the CNN model ResNet 50 extracts the features from the input image, the feature vector is linearly transformed to have the same dimension as the input dimension of the LSTM network. The network is then trained to be the language model on our feature vector. In our LSTM model, to implement the training process, we defined the label and target text, with labels containing the extra "$< start >$" word token and the target containg the extra "$< end >$" word token. They indicate the start and the end of the sentence, respectively.

During training, we used the method of "Teacher Forcing" to train the network. Teacher forcing works by using the teaching signal from the training dataset at the current time step, target(t), as input in the next time step x(t+ 1) = target(t), rather than the output y(t) generated by the network. For example, if the image description is "A group of elephants walking in muddy water", the source sequence is a list containing [Image embedding, '$< start >$', 'A', 'group', 'of', 'elephants', 'walking', 'in', 'muddy', 'water'] and the target sequence is a list containing ['$< start >$', 'A', 'group', 'of', 'elephants', 'walking', 'in', 'muddy', 'water', '$< end >$']. For each LSTM output, we would utilize softmax approach to determine what is the vocabulary generated at every time step. Once the "$< end >$" is hit, the sentence generating process will stop and the output would be given. Moreover, padding is used to make the generated captions in the fixed same length so that they could be stored in a mini-batch. The specifics of the output sampling are introduced in the section 3.4.

To gain the best combination of the hyperparameters, we sampled different parameters, combined them together, and observed the model performance. For the baseline LSTM model, we tried out learning rate 1e-5, 5e-4, and 1e-4, hidden size 256, 512, 768, and embedding size 250, 200.

3.2 Vanilla RNN

We also conducted experiment to replace LSTM to a regular two-layer Vanilla RNN model for model variations. The ouput of vanilla RNN for each timestamp could be represented as:

$$O = HW_{hq} + bq$$

, where $H = ReLU(X_t W_{xh} + H_{t-1} W_{hh} + b_h)$, given that the input and hidden layers are in shape of $X_t \in \mathbb{R}^{n*d}, H_t \in \mathbb{R}^{n*h}$. In other words, a Vanilla RNN is a LSTM without gates, where the output of the current layer of a RNN is depended on the result combining input at current time multiplied with its corresponding weights and the previous hidden states after applying the activation ReLU function.

3.3 LSTM with Image Encoding At Each Timestamp

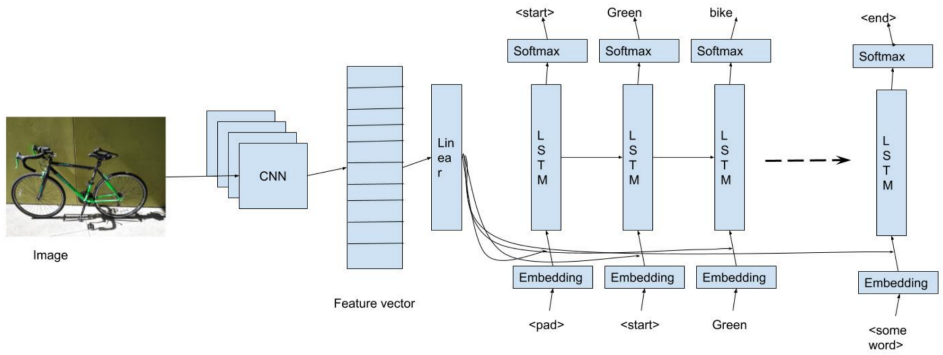

Figure 2: LSTM: Architecture 2 [14]

This model is very similar to the baseline LSTM model in terms of the architecture. Hence, calculation in detail will be the same as the baseline LSTM in the previous section. The input difference will be discussed in detail to explain what is special for LSTM with Image Encoding At Each Timestamp. The difference is that in the baseline LSTM model, image embedding was sent into the model as the input for the first timestamp and the following timestamp inputs are all word embedding. LSTM with Image Encoding At Each Timestamp, on the other hand, pass in a combination of word embedding and image embedding for each timestamp. In this case image embedding is fixed, a concatenation of word embedding at each timestamp and image embedding is the input for the corresponding timestamp. A pseudo input <pad> is used as the first word embedding input instead of using pure image embedding in baseline LSTM to predict <start>.

3.4 Output Sampling and Word Embedding

For output sampling, images to be sampled are first passed into the encoder, which is the resnet50 model. The outputs from this encoder are the image embedding for the images. A loop would start for the predictions of tokens. At this point, the method would differ from LSTM/RNN and LSTM architecture 2.

For LSTM/RNN, this image embedding would be sent to be trained decoder. The outputs from this decoder include both the prediction array and the updated hidden and cell state. A deterministic or stochastic approach would be applied to find the best matched token.

For the deterministic approach, we would output the word predicted at each time step, which has the maximum value of possibility. And for the stochastic approach, we need to apply τ as temperature for each unit in the output by:

$$y^j = exp(o^j/\tau) / \sum_n exp(o^n/\tau)$$

, where o^j represents one output from the output layer and n is the size of vocabulary. After calculating all y^j, we will output the one with the maximum value.

Then, this predicted token would be used as the input to be sent to the decoder again with the updated hidden and cell state. Then this prediction loop will continue until it predicts a <end> token or reached the max prediction length. For LSTM architecture 2, a word embedding for <pad> would be firstly created. An concatenation of the image embedding from the CNN network and word embedding of <pad> would be sent to be trained decoder. Similar to LSTM/RNN, the outputs would be the predicted token and updated versions of hidden and cell states. At this point, before passing in the predicted token like in LSTM/RNN, the predicted would be concatenated with the image embedding and then this concatenated result would be used as the input to be sent to the decoder again with the updated hidden and cell state. The prediction loop will continue until it predicts an <end> token or reached the max prediction length.

For word embedding, nn.Embedding() was used to create the embedding layer with weights initialized from N(0,1). By passing different words/indexs arrays, the corresponding vector would be generated as the output of this embedding layer. Hence, it acts as a lookup table that given an unique index, an unique vector with lower dimension would be generated to represent the index. In our case, each input word would be turned into an index and feed into the embedding layer. The embedding layer then will transform the index to a N-dimensional vector where N is the embedding size defined in the experiement.

3.5 Hyperparameter Searching and Tuning

Learning rate, hidden state size and embedding size are tuned. Search space detail can be found in the table below:

Table 1: Hyperparameter search space

	Learning Rate	Hidden Size	Embedding Dimension
ResNet50-LSTM	1e-5, 5e-4, 1e-4	256, 512, 768	150, 200, 250, 300, 350
ResNet50-LSTM with Architecture 2	5e-4, 1e-4	256, 512, 768	200, 250, 300, 350, 400
ResNet50-RNN	1e-5, 5e-4, 1e-4	N/A	N/A

In the current project, search space is determined with a selective search approach, which we adjust the search space of models based on the outcome from the grid search from previous models. This is why we are seeing different search spaces for different models. Firstly, we tuned the ResNet50-LSTM model. We found that a learning rate of 1e-5 is too slow, making the model unable to converge after 10 epochs and resulting in low bleu1 and bleu4 scores. Hence, we removed 1e-5 from our search space when tuning the hyperparameters for architecture 2. Also, during tuning ResNet50-LSTM model, we found a small embedding size usually result in bad loss and bleu scores. Hence, all embedding dimensions were added 50 in the search space for the LSTM architecture 2. For ResNet50-RNN, only the learning rate needs to be tuned according to the write up. Hence, no grid search was applied on this model for hidden state size and embedding dimension.

4 Result

We have summarized the best model with their hyperparameters in the table listed below:

Table 2: Model Hyperparameters

Hyperparameters \ model	ResNet50-LSTM	ResNet50-RNN	ResNet50-LSTM with Architecture 2
learning_rate	0.0001	0.0005	0.0001
hidden_size	768	512	768
embedding_size	350	300	200

4.1 Baseline LSTM

For our baseline LSTM model, we test our best model with hyperparameters of learning rate as 0.0001, the number of hidden state size as 768, and the embedding size as 350, which could also be seen from Table 2,and the plot of training and validation loss for this model is at Figure 3

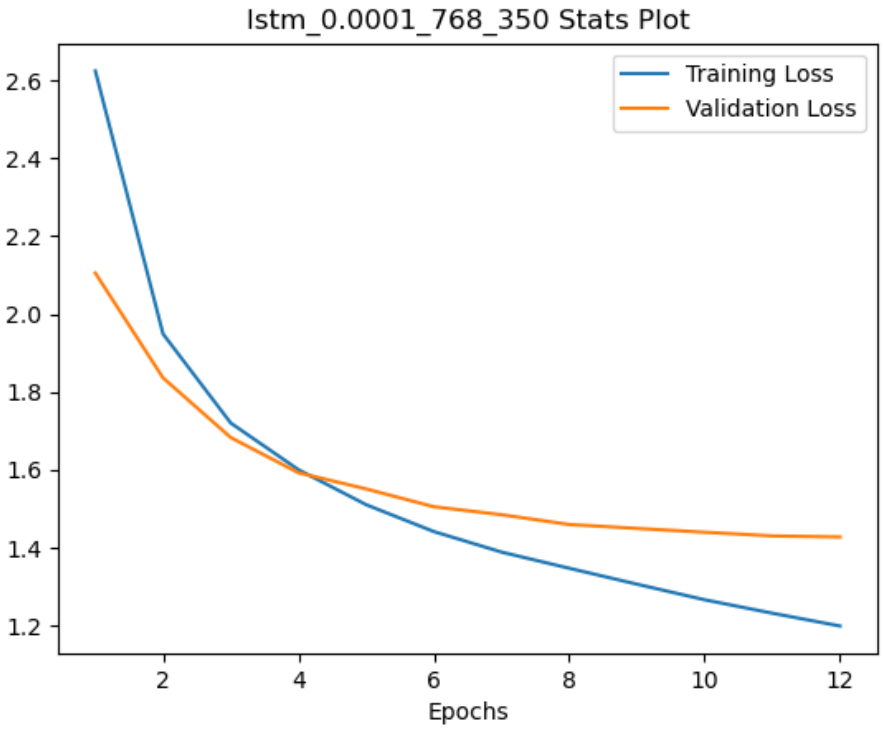

Figure 3: Baseline LSTM training and validation loss plot

We will summarize the performance of this model on the test set in the table below:

Table 3: Baseline LSTM test performance

Cross Entropy Loss	BLUE-1 score	BLUE-4 score
1.4344153363653953	67.77495522450724	8.967362508454396

4.1.1 Sample Results

We will list a couple of images with their predicted captions that are generated by our baseline LSTM model.

Actual captions:
a group of young boys playing a game of soccer .
a group of children playing soccer on a field
various boys in maroon and white shirts playing soccer .
a group of young children on a field playing soccer
a group of boys playing soccer on a field .

Predicted caption:
a group of young men playing a game of soccer .

Predicted caption with 0.001 temperature:
a group of people playing soccer on a field

Predicted caption with 5 temperature:
particularly look nordic majestic pitch nesting homeless employees bookcases played napkins spins drain leaning pancakes conventional sofas isles fountain sleeps

Predicted caption with deterministic:
a group of people playing soccer on a field

Figure 4: Baseline LSTM Caption good result 1

Actual captions:
a man power sliding on a long board
a young man sitting on his skateboard touching the ground
a man riding a skateboard down a street .
young man with skateboard appearing like he just fell down .
a man doing a trick on a skateboard in the middle of the street .

Predicted caption:
a man riding a skateboard in the middle of a street .

Predicted caption with 0.001 temperature:
a man riding a skateboard down a street .

Predicted caption with 5 temperature:
had patch beagle layer mcdonalds buiildings taxis comical accommodate motorcade allowed volley boats carrot valentines robinson napkin sanding pa reading

Predicted caption with deterministic:
a man riding a skateboard down a street .

Figure 5: Baseline LSTM Caption good result 2

Actual captions:
a man standing on a skateboard riding it down a sidewalk .
a man wearing a helmet rides a skateboard
a man on a skate board wearing a helmet and no shirt .
an old man riding a skateboard down a street .
a man in a helmet is riding a surfboard on the road .

Predicted caption:
a man in a helmet is skateboarding down a street .

Predicted caption with 0.001 temperature:
a woman in a white shirt and a black shirt and a skateboard

Predicted caption with 5 temperature:
boxes pack organize point sectional woodpeckers designed thomas partial dangled hostess demonstrating football socks valentines pursuit only microscope christmas actions

Predicted caption with deterministic:
a woman in a white shirt and a black shirt and a skateboard

Figure 6: Baseline LSTM Caption good result 3

Actual captions:
a herd of elephants walking through a lake filled with water .
a family of elephants washing up at a watering hole .
a group of elephants walking in muddy water .
group of elephants walking in muddy water today .
the elephants are standing beside each other near the water .

Predicted caption:
an elephant standing next to a black and white photo .

Predicted caption with 0.001 temperature:
a baby elephant walking in the dirt with a baby elephant .

Predicted caption with 5 temperature:
hardcover stationed cocktail canopy several saying easily utilities unpaved co photos childs trio bench breath dried liddle nose curl play

Predicted caption with deterministic:
a baby elephant walking in the dirt with a baby elephant .

Figure 7: Baseline LSTM Caption bad result 1

Actual captions:
this is an open box containing four cucumbers .
an open food container box with four unknown food items .
a small box filled with four green vegetables .
an opened box of four chocolate bananas .
an open box contains an unknown , purple object

Predicted caption:
a white toothbrush in a black and white photo .

Predicted caption with 0.001 temperature:
a pair of scissors are on a bed in a room .

Predicted caption with 5 temperature:
witha calm watermelon vehicle assistive fencing toys damaged above brocolli nature sources ceilings external advertises udders enjoyable adorably loader fridge

Predicted caption with deterministic:
a pair of scissors are on a bed in a room .

Figure 8: Baseline LSTM Caption bad result 2

Actual captions:
a pot of water has carrots in it .
appears to be carrots boiling in a pot of water
some cut up carrots are simmering in a pot on the stove .
some carrots are in a pot of boiling water .
several cut up carrots boiling in a pot of water .

Predicted caption:
a person holding a knife and cutting a knife .

Predicted caption with 0.001 temperature:
a person holding a knife and knife in a bowl .

Predicted caption with 5 temperature:
berries metallic euro normal pastry this vehicles this slopes woth 6th sunday events handicap !
traverses put outfits functional feel

Predicted caption with deterministic:
a person holding a knife and knife in a bowl .

Figure 9: Baseline LSTM Caption bad result 3

4.2 Res50-RNN

In addition to the LSTM model, we also tested how replacing LSTM to RNN would affect the performance of our model, and the below figure represents the training and validation loss for this model.

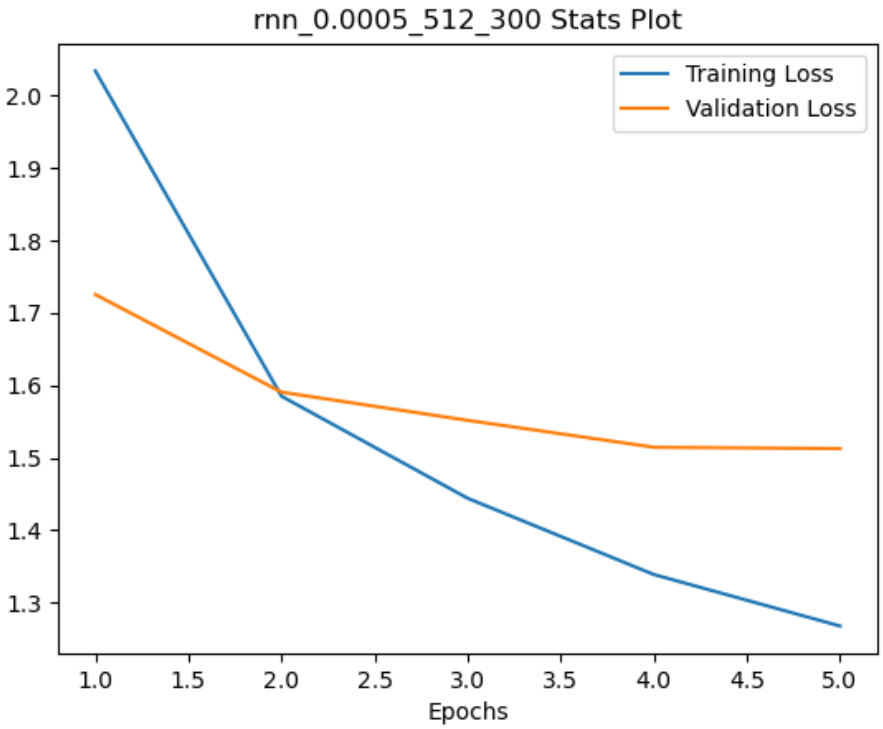

Figure 10: RNN training and validation loss

The performance of this model on the test set is shown in the table below:

Table 4: RNN test performance

Cross Entropy Loss	BLUE-1 score	BLUE-4 score
1.5607510373947469	65.22791534456205	7.053109833571947

4.2.1 Sample Results

We will list a couple of images with their predicted captions that are generated by this RNN model.

Actual captions:
a black and white dog is catching an orange frisbee .
a dog in the air with a frisbee in its mouth .
a black and white dog running with a frisbee in its mouth
a dog leaps through the air as it catches a frisbe
a border collie dog jumps and catches a frisbee in a park .

Predicted caption:
a dog is playing with a frisbee in a park .

Predicted caption with 0.001 temperature:
a dog is playing with a frisbee in a park .

Predicted caption with 5 temperature:
roam cream petting younger emerging wildlife calm joysticks during counter link consoles skewers oxygen split bending cabbages mogul

Predicted caption with deterministic:
a dog is playing with a frisbee in a park .

Figure 11: RNN Caption good result 1

Actual captions:
a man standing on a tennis court holding a tennis racquet .
male tennis play in backhand position , fixing to hit low ball which is not shown .
a young man wearing glasses hitting a tennis ball
a tennis ball is about to be hit back over the net .
a person hitting a tennis ball with a racket .

Predicted caption:
a woman standing on a tennis court holding a tennis racket .

Predicted caption with 0.001 temperature:
a woman is playing tennis on a tennis court .

Predicted caption with 5 temperature:
woman payer trawler dives lizard sits tricky tying elephants readying turns ago flanked holdign cakes passangers filling into claremont

Predicted caption with deterministic:
a woman is playing tennis on a tennis court .

Figure 12: RNN Caption good result 2

Actual captions:
a man is surfing on a crashing wave .
a surf boarder takes a turn on a rough wave .
a man surfing on his surf board against the waves
a man riding a white surfboard on a wave in the ocean .
a surfer in a wetsuit rides on a wave .

Predicted caption:
a man is surfing on a wave in the ocean .

Predicted caption with 0.001 temperature:
a man in a wetsuit riding a wave on a surfboard .

Predicted caption with 5 temperature:
goggles backwards " adjacent carried boy reagan chair on handle buttoned pilaf classic multicolored catch oxen phrases achievement 1950

Predicted caption with deterministic:
a man in a wetsuit riding a wave on a surfboard .

Figure 13: RNN Caption good result 3

Actual captions:
a trio of clocks are attached to the semi-circular frame by wires .
an arched wire sign sculpture has three clocks .
an art sculpture with three street clocks combined together under and arch and held together in the air .
three clocks set to different times suspended from an arch .
a group of three clocks handing from wires .

Predicted caption:
a clock is mounted on a metal pole .

Predicted caption with 0.001 temperature:
a clock on a pole with a clock on top .

Predicted caption with 5 temperature:
christmas redheaded miss honoring on curtain tables made en asia lo surgeons make interactive lines extended pawn page items arms

Predicted caption with deterministic:
a clock on a pole with a clock on top .

Figure 14: RNN Caption bad result 1

Actual captions:
three double decker busses are parked in front of a building .
three red double decker buses sitting partially behind a fence
red busses are parked behind a tall white fence .
a set of three red double decker buses parked next to each other .
three red double decker buses behind a gray fence .

Predicted caption:
a bus driving down a street with a car parked on the side of the road .

Predicted caption with 0.001 temperature:
a bus driving down a street with a bus .

Predicted caption with 5 temperature:
biplane adrift albert suv mantel various personal lifts there sizing noses travels kneeling over advertising croissant attentively splashing apartment specialized

Predicted caption with deterministic:
a bus driving down a street with a bus .

Figure 15: RNN Caption bad result 2

Actual captions:
a snow boarder riding down a snow covered summit .
a person struggling to get through deep snow .
a man is struggling to climb up a snow covered mountain .
a person slides down a steep , snowy mountain .
a person climbing up a snow covered mountain .

Predicted caption:
a man in a black jacket and black pants and black pants and black pants

Predicted caption with 0.001 temperature:
a person on a snowboard in the snow .

Predicted caption with 5 temperature:
youre pavilion sleeping games powdered oman shoreline smallest sunglasses impending crap buldings performance models sweeps ware beamed heads cattle anniversary

Predicted caption with deterministic:
a person on a snowboard in the snow .

Figure 16: RNN Caption bad result 3

4.3 LSTM with Image Encoding at Each Timestamp

We then conducted an experiment that combining the feature vector of images and word embedding before we pass to the LSTM. And the below table shows the metrics regarding this model, while the figure represents the training and validation loss for this model.

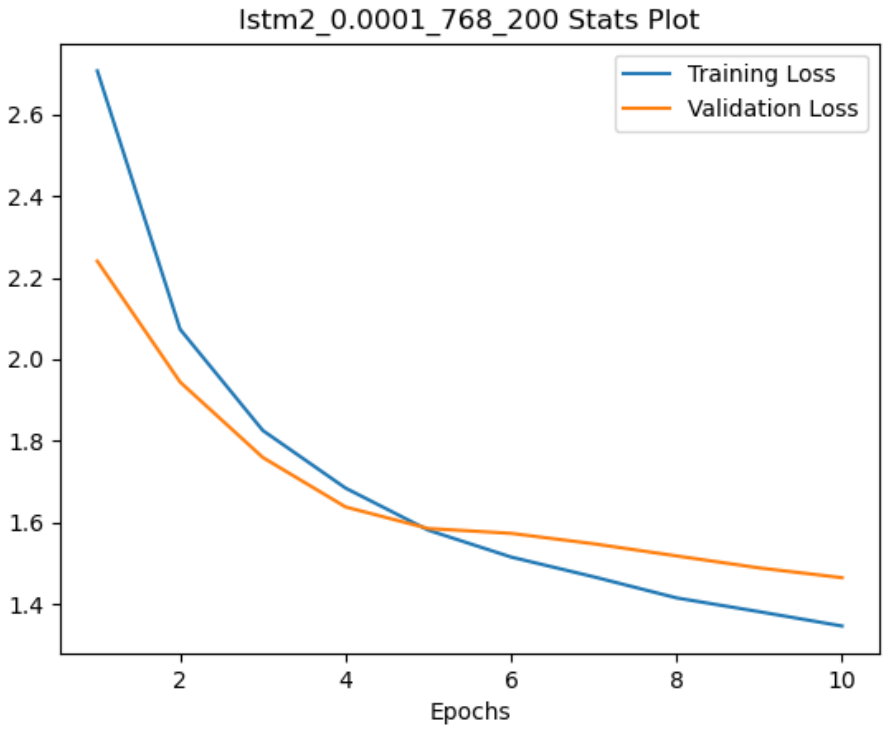

Figure 17: LSTM with Image Encoding at Each Timestamp Training and Validation Loss

The performance of this model on the test set is shown in the table below:

Table 5: LSTM with image encoding at each timestamp test performance

Cross Entropy Loss	BLUE-1 score	BLUE-4 score
1.466473819347138	69.05194122741226	8.742979007892718

4.3.1 Sample Result

We will list a couple of images with their predicted captions that are generated by this model.

Actual captions:
a white bus driving down a street next to power line .
a passenger bus with a billboard on the side of it
a bus is parked and waiting on passengers
large city bus stopped on the side of the road
a city bus that has advertising on the side and it is parked on the side of a street .

Predicted caption:
a bus is parked on the side of a road .

Predicted caption with 0.001 temperature:
a double decker bus is parked on a street .

Predicted caption with 5 temperature:
asia dear music connection fogy lapt often sprinkling werid directions cleaner newscast maneuvers pleasant vagina shooting phrases teat listen deodorant

Predicted caption with deterministic:
a double decker bus is parked on a street .

Figure 18: LSTM with Image Encoding Caption good result 1

Actual captions:
a large group of people running after a frisbee
a group of people playing a game of frisbee .
this is people running in a grassy field
a group of people playing frisbee on a field .
men and women running on a grassy field .

Predicted caption:
a group of people playing a game of frisbee .

Predicted caption with 0.001 temperature:
a group of people playing a game of frisbee .

Predicted caption with 5 temperature:
eyeball telephone paints gliding region thriving secret shrubby socialize hard factory playground cloths . lanyard skying diners changing wildebeests toothbrushes

Predicted caption with deterministic:
a group of people playing a game of frisbee .

Figure 19: LSTM with Image Encoding Caption good result 2

Actual captions:
dog sticking its head out of a car window
a dog is looking out the window of a car .
there is a dog with his head out of the window
a dog with it 's mouth open and head next to the window of a car , looking out of the vehicle .
a dog looking out the window of a vehicle

Predicted caption:
a dog is looking out the window of a vehicle .

Predicted caption with 0.001 temperature:
a dog is standing on a bench in the water .

Predicted caption with 5 temperature:
front knapsack interface duval reception gutter salad lg bullying harley-davidson control oft raining polaroid serengeti shacks opposing mayors easter ahead

Predicted caption with deterministic:
a dog is standing on a bench in the water .

Figure 20: LSTM with Image Encoding Caption good result 3

Actual captions:
two men standing over a table eating food .
two guys wearing suits are eating at a party .
two men in black dress clothing eating food off of a table .
two men in suits are grabbing food from a table .
two men eating vegetables in front of a window .

Predicted caption:
a man and a woman posing for a picture .

Predicted caption with 0.001 temperature:
a man and a woman standing next to each other .

Predicted caption with 5 temperature:
depiction stage slice 381 dressing homeplate foggy freckled insects splash girlfriend ahead stood cabinetry neck magazines sled lunchtime patio monk

Predicted caption with deterministic:
a man and a woman standing next to each other .

Figure 21: LSTM with Image Encoding Caption bad result 1

Actual captions:
there is a garbage truck surrounded by people on a road
a machine is digging as a group of people look on .
a crowd of men stand beside heavy equipment on a road .
many men in turbans stand around a dump truck as it works on the side of the road .
people gathered around a construction truck in the street .

Predicted caption:
a man and a woman standing next to a plane .

Predicted caption with 0.001 temperature:
a man is standing next to a large truck .

Predicted caption with 5 temperature:
tiolet burger old batteries structures flops basset tippy machines ball allow perch mein flooded shaped cold but planks profile footage

Predicted caption with deterministic:
a man is standing next to a large truck .

Figure 22: LSTM with Image Encoding Caption bad result 2

Actual captions:
an advertisement from a christmas wonderland store with several ge products .
a page in a general electric 's catalog advertising christmas ideas .
electronics advertised on a paper for christmas .
an ad for general electric with clocks and appliances on it
a various items such clocks and kitchen appliances are advertized .

Predicted caption:
a display of a variety of different types of doughnuts .
Predicted caption with 0.001 temperature:
a display of a variety of different items on it .

Predicted caption with 5 temperature:
supports technicians landed juveniles lend balanced little floor fame spatulas traipsing worst butts blending designating verizon wind reads circular cage

Predicted caption with deterministic:
a display of a variety of different items on it .

Figure 23: LSTM with Image Encoding Caption bad result 3

5 Discussion

Based on our results, the three model structures we implement have different level of image captioning ability. The first model, which is the one has CNN-LSTM structure with image input only right before the first decoder layer, has the lowest cross entropy loss. In terms of the BLEU-1 score, all three models have achieved values higher than 60, which is quite impressive as this suggests a great quality that can be potentially better than humans. However, the BLEU-4 scores for all three models are quite low and below 10, which can be considered as useless. While BLEU-4 is the culumative score from 1-gram to 4-gram comparison, this suggests all these models we developed have a good performance when conducting token-to-token comparison, but much worse

in the comparison with larger token groups [10].

LSTM with architecture 2 (LSTM2) has the best BLEU1 score while LSTM with architecture 1 (LSTM1) has the best BLEU4 score. This result is reasonable. For BLEU1, it measures the average proportion of matched 1-grams between the prediction caption and reference captions. LSTM with architecture 2 has the better BLEU1 score than LSTM because image embedding was passed in at each timestamp in LSTM2, hence increasing the impact of image features on the next token predictions and increasing the possibility of containing the matching tokens in the predicted outcome. However, it has a slightly lower score of BLEU4. Since BLEU4 measures the average proportion of matched 4-grams, correct order of predicted tokens was much more emphasized than BLEU1. Since image embedding was only passed in once at the very first timestamp in LSTM1, the impact of image feature is less compared to LSTM2. This makes the model use a larger proportion of information from word embedding and cell state to predict the next token. Cell state serves like memory in LSTM, when a larger proportion of information from memory was used to predict the next token, tokens in the correct order will be more likely to be predicted. These are the reason why LSTM1 has the better BLEU4 score while LSTM2 has the better BLEU4 score. RNN, on the other hand, has the worst BLEU1 and BLEU 4 score. This is due to architectural difference between RNN and LSTM. With the use of gates, LSTM is able to avoid the problems of vanishing gradient and exploding gradient. Hence, the weights that RNN learned are likely to be worse than LSTM's weights in terms of the ability to predict correct captions.

Deterministic approach to sample from predictions usually results a worse result. This might be due to that only taking the maximum output is so extreme that the model's ability at generalizing to unseen data is worse. A stochastic approach allows the model to sample from a distribution hence enable the model to be more flexible when making predictions. This might increase bias a bit but would highly reduce variance and make the model generalize better to unseen data. Furthermore, temperature plays an important role at predicting captions using the stochastic approach. When temperature is very low, the stochastic approach would be similar to the deterministic approach. This is because that when sampling from a list of probabilities, temperature is the denominator and prediction probability is the numerator. When temperature is low, the small predicted probability would be smaller and the large predicted probability would be larger. On the other hand, when the temperature is too high, the prediction distribution to be sampled from will turn into a uniform distribution. Since the temperature is the denominator, a high number of temperature will make every prediction to be same likely. Hence, a high temperature will lead to outcomes that make no sense.

6 Team contributions

Judy Yang: Implemented the baseline model with Du. Implemented part of the pipeline of the code in the experiment.py. Helped implement the RNN mode as well as the LSTM architecture. Discussion with the whole team to write the discussion part. Wrote related work.

Du Xiang: Implemented the baseline model with Judy. Worked with Qiwen for the sample function. Helped implement the RNN model. Produce captions for LSTM2. Wrote results section with Arthur. Wrote introduction and abstract section. Proofread and made revisions on the model part. Discussion with the whole team to write the discussion part.

Qiwen Zhang: Helped implement the baseline model and skeleton of the code. Worked on caption selection with Arthur. Helped implement the RNN mode as well as the LSTM architecture. Produce captions for LSTM2. Discussion with the whole team to write the discussion part. Wrote model section with Arthur.

Arthur Wang : Helped implement the baseline model and skeleton of the code. .Helped implement the RNN mode as well as the LSTM architecture. Produce captions for RNN. Wrote results section with Du. Wrote model section with Qiwen. Discussion with the whole team to write the discussion part.

References

[1] Xinlei Chen and C Lawrence Zitnick. "Learning a recurrent visual representation for image caption generation". In: *arXiv preprint arXiv:1411.5654* (2014).

[2] Hao Fang et al. "From captions to visual concepts and back". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1473–1482.

[3] Ali Farhadi et al. "Every picture tells a story: Generating sentences from images". In: *European conference on computer vision*. Springer. 2010, pp. 15–29.

[4] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[5] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[6] Ryan Kiros, Ruslan Salakhutdinov, and Rich Zemel. "Multimodal neural language models". In: *International conference on machine learning*. PMLR. 2014, pp. 595–603.

[7] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. "Unifying visual-semantic embeddings with multimodal neural language models". In: *arXiv preprint arXiv:1411.2539* (2014).

[8] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: 1405.0312 [cs.CV].

[9] Junhua Mao et al. "Deep captioning with multimodal recurrent neural networks (m-rnn)". In: *arXiv preprint arXiv:1412.6632* (2014).

[10] Kishore Papineni et al. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[11] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[12] Richard Socher et al. "Grounded compositional semantics for finding and describing images with sentences". In: *Transactions of the Association for Computational Linguistics* 2 (2014), pp. 207–218.

[13] Chen Sun, Chuang Gan, and Ram Nevatia. "Automatic concept discovery from parallel text and visual corpora". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2596–2604.

[14] Anurag Tripathi, Siddharth Srivastava, and Ravi Kothari. "Deep Neural Network Based Image Captioning". In: *Big Data Analytics*. Ed. by Anirban Mondal et al. Cham: Springer International Publishing, 2018, pp. 335–347. ISBN: 978-3-030-04780-1.

[15] Oriol Vinyals et al. "Show and tell: A neural image caption generator". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3156–3164.

[16] Kelvin Xu et al. "Show, attend and tell: Neural image caption generation with visual attention". In: *International conference on machine learning*. PMLR. 2015, pp. 2048–2057.