Implementation of Different Architecture RNN and Train & Test Them on COCO Caption Dataset for Image Caption Task

Enze Ma
Department of Computer Science
University of California San Diego
e1ma@ucsd.edu

Yixin Jiang
Department of Computer Science
University of California San Diego
yij011@ucsd.edu

Fangqi Yuan
Department of Computer Science
University of California San Diego
fayuan@ucsd.edu

Abstract

For this project, we total created 3 different architectures of RNN. These three models' encode is the same which is used resnet50 model to encode the image and remove the last fc layer them replace it with our 2048-¿embedd size(which we decide) trainable fc layer. We will freeze all layers except the last fc layer we add. For the decoder part, the three modes have a similar architecture with one crucial difference. The first model we call the baseline model. It has three layers the first layer is the embedding layer that we use to project the word to vector. The second layer is the LSTM layer we use to learn the relation of the image and caption. The last one is and simple fc layer from hidden-size -¿ vocab-size. The second model is the vanilla RNN model, we use a regular vanilla RNN layer to replace the LSTM layer and keep others the same. The third model is Architecture 2(A2), which has the same architecture as LSTM but instead only uses the hidden state or feature as the input of the LSTM layer, we will combine the hidden state and feature as the input. At last, for the baseline model get 66 on bleu1, 8.2 on bleu4. RNN gets 65.7 on bleu1,7.85 on bleu4. A2 gets 67.5 on bleu1, 7.8 on bleu4. We can conclude that our model is successful.

1 Introduction

This project is trying to build a simple model to do the task we call image caption which is generating the description that can describe the main feature of the image, which is a combination of natural language processing and computer vision. We experiment with three different models, and through adjustment, we finally got the best results we could get under these three architectures. There is a big difference in the performance of the three models under bleu1 evaluation system, but the performance under bleu4 evaluation system is relatively low, which may reflect that our model can learn the main image features and transform them to word, but can't learn the grammar and the connection well. Therefore, we mainly use bleu1 as the basis for our evaluation. By comparing the three models, we found that A2 has the best performance. A2 uses the image feature as an input element at each step, and its better performance may indicate whether it is in the baseline or vanilla RNN we always lose some important feature during the LSTM or vanilla RNN layer. Adding these features at each step can help the model to better understand the image and draw more accurate

conclusions. Therefore, we think that the model architecture of A2 may be more suitable for the task of image caption.

2 Related Work

In this experiment, we use the teacher-forcing technique to train the model and simplify the loss case. Teacher forcing is a fast and efficient way to train a recurrent neural network model that uses the basic facts of previous time steps as input. It is a network training approach that is critical to developing a deep learning language model for machine translation, text summarization and image captioning, and many other applications (Rafi et al., 2014) and the teacher forcing technique can significantly improve the slow convergence and poor performance of recurrent neural networks. The existing image caption models usually use cross entropy (XE) loss and reinforcement learning (RL) to learn real words as hard targets. In order to solve the problem of inappropriate reward distribution in the training strategies adopted by the existing image caption models, Huang et al., (2020) proposed to construct a teacher model combined with real image attributes as a soft target to improve the training bias. The conclusion shows that this architecture with image embedding is suitable for the actual implementation of image caption model and improves the flexibility and actual performance of model training.

3 Methods

Note: embed-size is the size we can choose by ourselves. vocab-size is depending on the vocabulary list we get. hidden size is the size we choose by ourselves

3.1 Describe the architecture for the baseline(LSTM)

The baseline model is consist of two main parts, one is the encoder one is the decoder. For the encoder part, we use the pre-trained resnet50 model as the main part but replace it last fc layer with our fc layer which has embed-size output units. For the decoder part, we mainly have three layers, one is an embedding layer that projects each caption to a vector, so its input size should be vocab-size and its output size is just embed-size. The second layer is the LSTM layer with input size embed-size, the output size hidden-size, the num of layers inside the LSTM layer is 2. The third layer is the fc layer which has embed-size input units and vocab-size.

3.2 Describe the vanilla RNN model

The vanilla RNN model is consist of two main parts and is almost the same as the baseline, one is the encoder one is the decoder. For the encoder part, we use the pre-trained resnet50 model as the main part but replace it last fc layer with our fc layer which has embed-size output units. For the decoder part, we mainly have three layers, one is an embedding layer which input size should be vocab-size and its output size is just embed-size we can choose freely. The second layer is the most important change we did from baseline, we use the vanilla RNN layer to replace LSTM layer with the input size embed-size, the output size hidden-size, the num of layers inside the vanilla RNN layer is 2. The third layer is the fc layer which has embed-size input units and vocab-size.

3.3 Describe the model A2

The A2 model is consist of two main parts, one is the encoder one is the decoder. For the encoder part, we use the pre-trained resnet50 model as the main part but replace it last fc layer with our fc layer which has embed-size output units. For the decoder part, we mainly have three layers, one is an embedding layer which input size should be vocab-size and its output size is just embed-size. The second layer is still the LSTM layer but now we need different input sizes, because each timestep we need concat the word prediction and the image feature so we need to change the input size of this layer, it should have input size embed-size*2, output size hidden-size, the num of layers inside the LSTM layer is still 2. The third layer is the fc layer which has embed-size input units and vocab-size.

3.4 Describe how you sample outputs from the decoder and how your model obtains word embeddings.

deterministic: we choose the index which has the maximum value
scholastic: we choose the index by using the softmax(output) probability.
For baseline model, it's pretty simple to sample, the first time step we just input the image feature into the LSTM layer and use its output to make a prediction based on deterministic or scholastic add it to the result list, then embed the prediction and use it as our next time input, then we keep the same process until we hit the word ¡end¿ or reach the max-sequence we set. Then we can finally get the caption we want by removing unnecessary words like ¡start¿.
For vanilla RNN model, it's almost the same as the baseline model but instead input into the LSTM layer we input it into the vanilla RNN layer and everything else is the same process.
For A2 model, it's pretty similar to baseline, the first time step we just input the concatenation of padding embedding and feature into the LSTM layer and use its output to make a prediction based on deterministic or scholastic add it to the result list, then embed the prediction and concat it with the image feature then use it as our next time input, then we keep the same process until we hit the word ¡end¿ or reach the max-sequence we set. Then we can finally get the caption we want by removing.

3.5 Describe how you conducted your hyperparameter search (e.g., what range you sampled from for each hyperparameter, etc.).

For LSTM, first we tune the three parameters in a larger range, then we find that if the learning rate is larger that we have used 5e-4, our BLEU score decreases and we get some sharp curve jitter but the model performance does not improve significantly and then we lower the learning rate and there is a little improvement in performance but not too much. Then we finally decide the optimal range is about 1e-4. For hidden size and embedding size, the intuition we got in the process of experimenting is that both the too big size or small size will work worse, and we keep test the size with an interval change of 200 then we get about the same performance in between 500 to 1000 for hidden size. Likewise, we use the same method and notice that 450 is the best value for embedding size. We finally decide 800 as our best hidden size and 450 as our embedding size for the baseline model.
Since LSTM, RNN and A2 models are common in parameter selection, our parameter adjustment steps are roughly the same and our best hyperparameter records are listed below.

4 Results

4.1 Report the best hyperparameters you found for each model

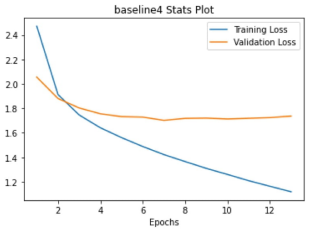
Baseline(LSTM):
Hidden size: 800
Embedding size: 450
Epochs = 20
Learning rate = 1e-4

RNN:
Hidden size: 1200
Embedding size: 700
Epochs = 20
Learning rate = 1e-4

Architecture 2:
Hidden size: 512
Embedding size: 300
Epochs = 20
Learning rate = 5e-4

4.2 Provide plots of your training loss and validation loss vs number of epochs for each of the three models (using the best hyperparameters for each). For each of your best models, also report the cross entropy loss on the test set.

4.2.1 Baseline(LSTM)

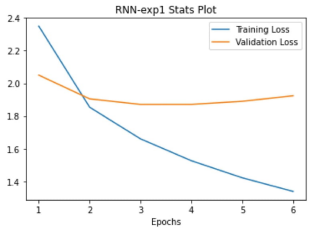



```
Epoch: 13, Train Loss: 1.1181931909864207, Val Loss: 1.7363970307717487, Took 0:04:19.624505, ETA: 0:30:17.371535


[W pthreadpool-cpp.cc:90] Warning: Leaking Caffe2 thread-pool after fork. (function pthreadpool)
[W pthreadpool-cpp.cc:90] Warning: Leaking Caffe2 thread-pool after fork. (function pthreadpool)
[W pthreadpool-cpp.cc:90] Warning: Leaking Caffe2 thread-pool after fork. (function pthreadpool)
[W pthreadpool-cpp.cc:90] Warning: Leaking Caffe2 thread-pool after fork. (function pthreadpool)

Test Performance: Loss: 1.7381923930922623, Bleu1: 66.22229390873864, Bleu4: 8.208453031941215
```

```
In [ ]:
```


Baseline(LSTM):
Cross entropy loss on the test set: 1.7381923930922623

4.2.2 RNN

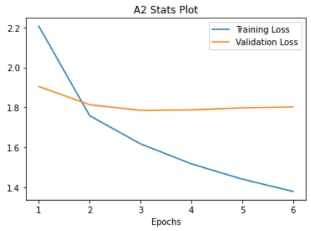



```
Epoch: 6, Train Loss: 1.3377565373883062, Val Loss: 1.9234546552299376, Took 0:04:40.011523, ETA: 1:05:20.161322


[W pthreadpool-cpp.cc:90] Warning: Leaking Caffe2 thread-pool after fork. (function pthreadpool)
[W pthreadpool-cpp.cc:90] Warning: Leaking Caffe2 thread-pool after fork. (function pthreadpool)
[W pthreadpool-cpp.cc:90] Warning: Leaking Caffe2 thread-pool after fork. (function pthreadpool)
[W pthreadpool-cpp.cc:90] Warning: Leaking Caffe2 thread-pool after fork. (function pthreadpool)

Test Performance: Loss: 1.9321151884451004, Bleu1: 65.74056110584493, Bleu4: 7.852915392314504
```

```
In [ ]:
```


RNN:
Cross entropy loss on the test set: 1.9321151884451004

4.2.3 Architecture 2




```
Epoch: 6, Train Loss: 1.3785542345641963, Val Loss: 1.8018053922040984, Took 0:07:21.905931, ETA: 1:43:06.683034


[W pthreadpool-cpp.cc:90] Warning: Leaking Caffe2 thread-pool after fork. (function pthreadpool)
[W pthreadpool-cpp.cc:90] Warning: Leaking Caffe2 thread-pool after fork. (function pthreadpool)
[W pthreadpool-cpp.cc:90] Warning: Leaking Caffe2 thread-pool after fork. (function pthreadpool)
[W pthreadpool-cpp.cc:90] Warning: Leaking Caffe2 thread-pool after fork. (function pthreadpool)

Test Performance: Loss: 1.7993824636068185, Bleu1: 67.53733119641801, Bleu4: 7.791744726516001
```

```
In [ ]:
```


A2:
Cross entropy loss on the test set: 1.7993824636068185

4.3 Report BLEU-1 and BLEU-4 scores for the three models on the test set. You may use any library function that implements BLEU scores. Helper functions for these are provided in the code, but you are free to use any other library function if you wish.

```
Test Performance: Loss: 1.7381923930922623, Bleu1: 66.22229390873864, Bleu4: 8.208453031941215
```

Figure 1: Baseline(LSTM) BLEU-1 and BLEU-4 scores on the test set

```
Test Performance: Loss: 1.9321151884451004, Bleu1: 65.74056110584493, Bleu4: 7.852915392314504
```

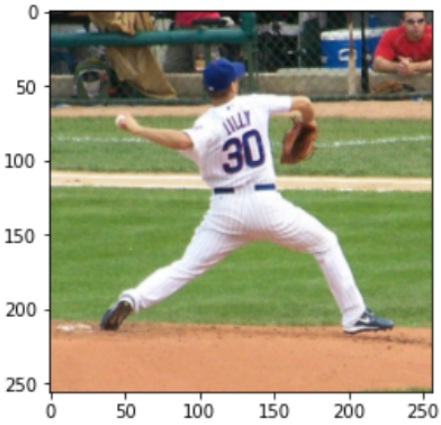
Figure 2: RNN BLEU-1 and BLEU-4 scores on the test set

```
Test Performance: Loss: 1.7993824636068185, Bleu1: 67.53733119641801, Bleu4: 7.791744726516001
```

Figure 3: Architecture 2 BLEU-1 and BLEU-4 scores on the test set

4.4 Provide at least three examples of images that resulted in good captions, and three images that resulted in bad captions for each of the three models.

4.4.1 Baseline(LSTM)




```
good  1
actual captions: ['A pitcher holds his arm far behind him during a pitch.', 'The baseball player is throwing a very intense pit
ch.', 'A baseball player pitching a ball on a field,', 'a professional pitcher on the mound getting ready to throw the ball',
'A baseball player throws a pitch while others watch from the dugout.']
predict captions: ['a', 'baseball', 'player', 'is', 'getting', 'ready', 'to', 'swing', 'at', 'a', 'pitch', '.']
Test Performance: Bleu1: 83.33333333333334, Bleu4: 1.1111111111111112
```

Figure 4: Baseline Good Example 1

```
345466  :  ['<start>', 'a', 'baseball', 'player', 'swinging', 'a', 'bat', 'at', 'a', 'ball', '<end>']
```

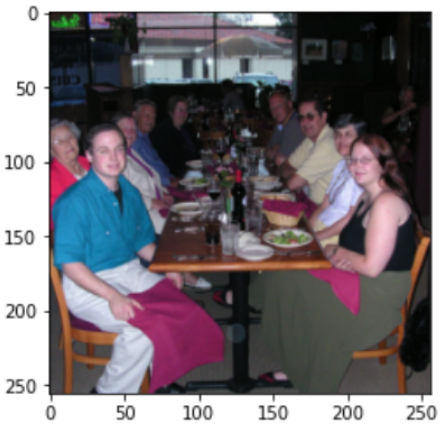
Figure 5: Baseline Good Example 1, Deterministic = true

```
345466  :  ['lufthansa', 'powers', 'standign', 'approaching', 'before', 'pasture', 'slope', 'grey', 'file', 'crates', 'rain-str
eaked', 'pig', 'deckered', 'who', 'shrubs', 'university', 'pictures', 'carelessly', 'crocheted', 'turn']
```

Figure 6: Baseline Good Example 1, Temperature = 5

```
345466  :  ['a', 'baseball', 'player', 'swinging', 'a', 'bat', 'at', 'a', 'ball']
```

Figure 7: Baseline Good Example 1, Temperature = 0.001




```
good  2
actual captions: ['A group of people sitting around a wooden table with food.', 'The nine people smile as they sit at a dinner
table.', 'A table full of people that are eating at a restaurant.', 'Co-workers often get together after a long day at work.',
'A group of people that are sitting around a table.']
predict captions: ['a', 'group', 'of', 'people', 'sitting', 'at', 'a', 'table', 'with', 'wine', 'glasses']
Test Performance: Bleu1: 81.81818181818183, Bleu4: 25.0
```

Figure 8: Baseline Good Example 2

```
277991  :  ['<start>', 'a', 'group', 'of', 'people', 'sitting', 'at', 'a', 'table', 'with', 'food', 'and', 'drinks', '.', '<end
>']
```

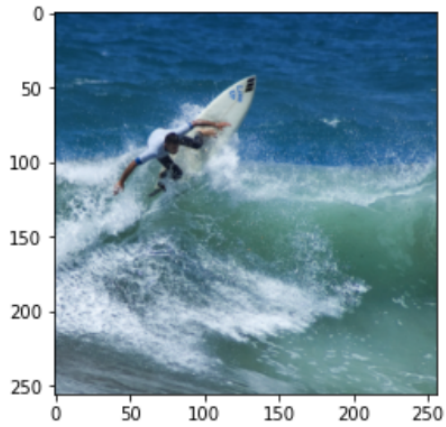
Figure 9: Baseline Good Example 2, Deterministic = true

```
277991  :  ['carves', 'tanding', 'town', 'told', 'striped', 'package', 'crown', 'panorama', 'purchasing', 'junction', 'sit', 't
asble', 'this', 'mouths', 'toward', 'desks', 'chickens', 'splash', 'wagging', 'sleepy']
```

Figure 10: Baseline Good Example 2, Temperature = 5

```
277991  :  ['a', 'group', 'of', 'people', 'sitting', 'at', 'a', 'table', 'with', 'food', 'and', 'drinks', '.']
```

Figure 11: Baseline Good Example 2, Temperature = 0.001




```
good  3
actual captions: ['A person that is surfing in the water.', 'A man on a surfboard surfing a wave in the ocean.', 'A man riding
a wave on a surfboard.', 'a surfer in a white shirt  surfing on a sunny day', 'A man on a surfboard riding an ocean wave.']
predict captions: ['a', 'surfer', 'is', 'riding', 'a', 'wave', 'on', 'a', 'board', '.']
Test Performance: Bleu1: 90.0, Bleu4: 28.57142857142857
```

Figure 12: Baseline Good Example 3

```
82715  :  ['<start>', 'a', 'surfer', 'riding', 'a', 'wave', 'in', 'the', 'ocean', '.', '<end>']
```

Figure 13: Baseline Good Example 3, Deterministic = true

```
82715  :  ['rare', 'propensity', 'riding', 'girafes', 'terminals', 'whit', 'provided', 'theater', 'meal', 'prints', 'prints',
'household', 'runaway', 'beverages', 'stopped', 'whippet', 'model', 'reeds', 'returned', 'clad']
```

Figure 14: Baseline Good Example 3, Temperature = 5

```
82715  :  ['a', 'surfer', 'riding', 'a', 'wave', 'in', 'the', 'ocean', '.']
```

Figure 15: Baseline Good Example 3, Temperature = 0.001


```
bad  1
actual captions: ['A wooden table topped with fruits and vegetables.', 'A pile of assorted vegetables are piled onto a wooden t
able.', 'Fruit and vegetables are on a wooden board.', 'A table contains an assortment of fruit and vegetables.', 'A table full
of various vegetables and legumes.']
predict captions: ['a', 'bowl', 'of', 'mixed', 'fruit', 'including', 'oranges', 'and', 'apples']
Test Performance: Bleu1: 44.44444444444444, Bleu4: 1.666666666666667
```

Figure 16: Baseline Bad Example 1

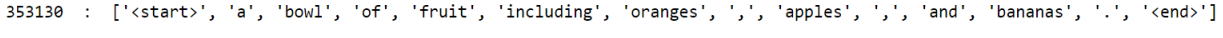


Figure 17: Baseline Bad Example 1, Deterministic = true

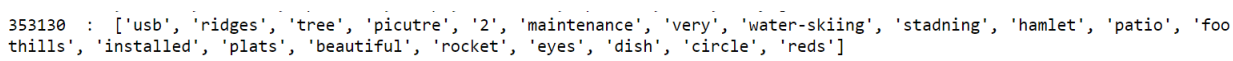


Figure 18: Baseline Bad Example 1, Temperature = 5

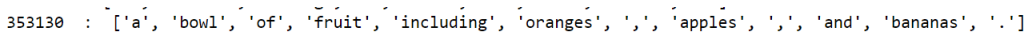
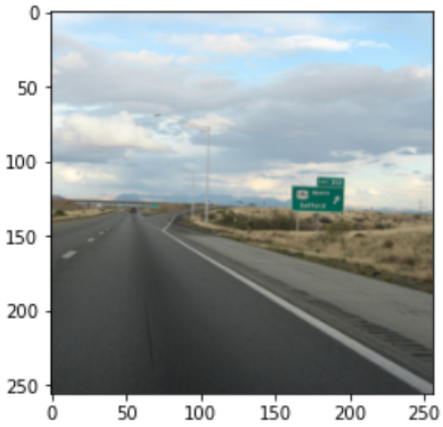


Figure 19: Baseline Bad Example 1, Temperature = 0.001




```
bad  2
actual captions: ['A highway sign sitting on the side of a highway.', 'THERE IS A HIGHWAY WITH SIGNS ON THE SIDE OF IT ', 'A qu
iet highway with a street sign up ahead.', 'US eastbound Interstate 10 highway at Exit 352', 'Driving on the highway towards an
exit ramp with brush on both sides.']
predict captions: ['a', 'road', 'with', 'a', 'sign', 'saying', '``', 'no', "'", 'written', 'on', 'a', 'highway', 'sign', 'sayi
ng', '``', '<unk>', "'", '.']
Test Performance: Bleu1: 36.84210526315789, Bleu4: 0.6250000000000002
```

Figure 20: Baseline Bad Example 2

```
65567  :  ['<start>', 'a', 'road', 'with', 'a', 'sign', 'and', 'a', 'truck', 'on', 'a', 'road', '.', '<end>']
```

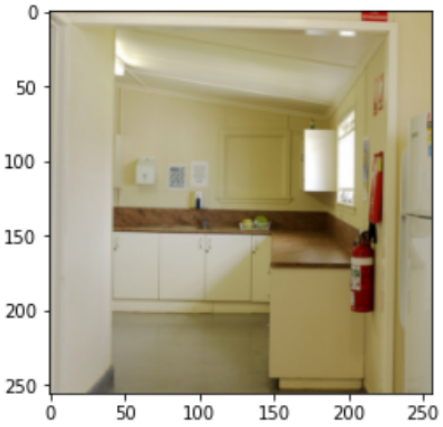
Figure 21: Baseline Bad Example 2, Deterministic = true

```
65567  :  ['knotted', 'lines', 'gazelle', 'planted', 'manual', 'parasols', 'duval', 'passes', 'dye', 'teal', 'loking', 'depict
s', 'warms', 'bear', 'deckered', 'sidelines', 'ominous', 'impersonators', 'irritated', 'taco']
```

Figure 22: Baseline Bad Example 2, Temperature = 5

```
65567  :  ['a', 'road', 'with', 'a', 'sign', 'and', 'a', 'truck', 'on', 'a', 'road', '.']
```

Figure 23: Baseline Bad Example 2, Temperature = 0.001




```
bad  3
actual captions: ['A room filled with different types of items all around.\n', 'All white kitchen with brown counter tops and r
ed fire extinguisher. ', 'A kitchen with cream colored walls and brown counters.', 'A modern kitchen has an abundance of counte
r space.', 'THIS IS A PICTURE OF A LARGE CLEAN KITCHEN']
predict captions: ['a', 'bathroom', 'with', 'a', 'sink', 'and', 'mirror', 'and', 'a', 'mirror']
Test Performance: Bleu1: 40.0, Bleu4: 1.4285714285714295
```

Figure 24: Baseline Bad Example 3

```
385103  :  ['<start>', 'a', 'bathroom', 'with', 'a', 'white', 'sink', 'and', 'a', 'toilet', 'in', 'it', '<end>']
```

Figure 25: Baseline Bad Example 3, Deterministic = true

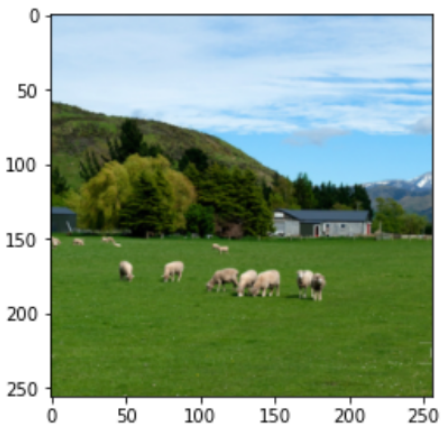
```
385103  :  ['secured', 'handcuffs', 'safe', 'league', 'ugly', 'builing', 'clay', 'purchase', 'believing', 'barreling', 'laid',
'continuum', 'menu', 'front', 'portion', 'dip', 'bloomed', 'phases', 'os', 'hay']
```

Figure 26: Baseline Bad Example 3, Temperature = 5

```
385103  :  ['a', 'bathroom', 'with', 'a', 'white', 'sink', 'and', 'a', 'toilet', 'in', 'it']
```

Figure 27: Baseline Bad Example 3, Temperature = 0.001

4.4.2 RNN




```
good  1
actual captions: ['Lots of sheep graze on a grass field.', 'A herd of sheep that are grazing on some grass.', 'A group of sheep
on a grass field.', 'A herd of sheep standing on top of a lush green field.', 'A group of sheep graze in a lush grass field.']
predict captions: ['a', 'herd', 'of', 'sheep', 'standing', 'on', 'a', 'lush', 'green', 'field', '.']
Test Performance: Bleu1: 100.0, Bleu4: 62.5
```

Figure 28: RNN Good Example 1

Figure 29: RNN Good Example 1, Deterministic = true

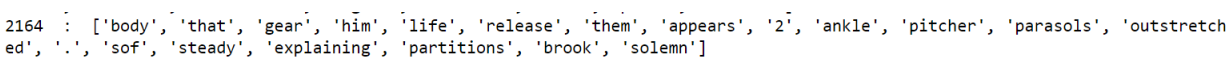
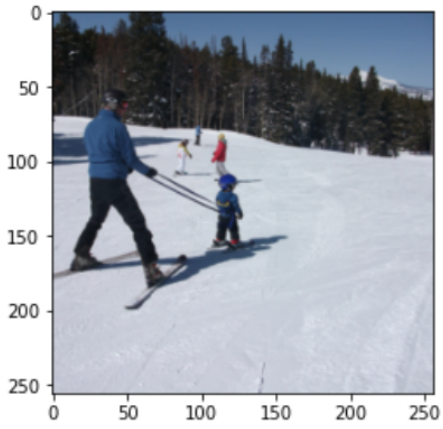


Figure 30: RNN Good Example 1, Temperature = 5

Figure 31: RNN Good Example 1, Temperature = 0.001




```
good  2
actual captions: ['A small boy on a leash skiing with an adult.', 'Some skiers skiing outside in the snow on a slop', 'A man ri
ding down a ski slope with a young boy.', 'A man and a young child on snow skis going down a hill.', 'The man and the little ch
ild ski down the slope.']
predict captions: ['a', 'person', 'on', 'skis', 'going', 'down', 'a', 'snow', 'covered', 'slope', '.']
Test Performance: Bleu1: 81.81818181818183, Bleu4: 12.500000000000004
```

Figure 32: RNN Good Example 2

Figure 33: RNN Good Example 2, Deterministic = true

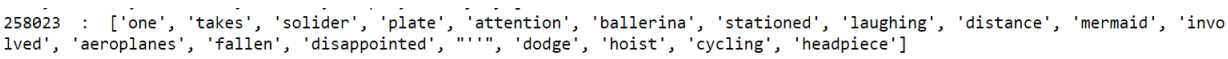


Figure 34: RNN Good Example 2, Temperature = 5

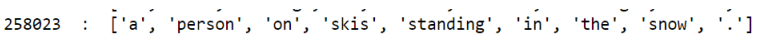
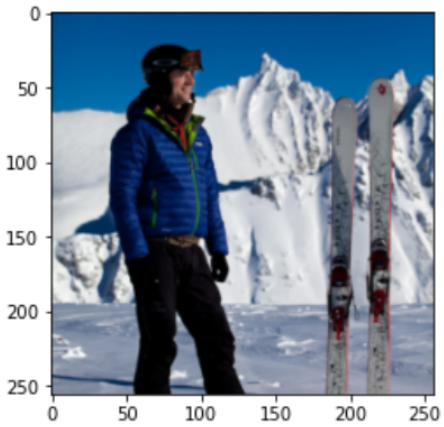


Figure 35: RNN Good Example 2, Temperature = 0.001




```
good  3
actual captions: ['A skier stands next to skis stuck into the snow.', 'A man standing on the ski slope with his pair of skis.',
'A man standing on top of snow covered ground.', 'A man that is standing in the ice.', 'A man standing in the snow beside his s
kis']
predict captions: ['a', 'person', 'on', 'skis', 'standing', 'in', 'the', 'snow', '.']
Test Performance: Bleu1: 88.88888888888889, Bleu4: 16.666666666666668
```

Figure 36: RNN Good Example 3

```
14874  :  ['<start>', 'a', 'person', 'on', 'skis', 'standing', 'in', 'the', 'snow', '.', '<end>']
```

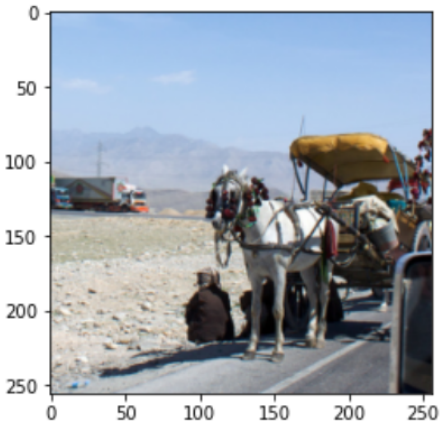
Figure 37: RNN Good Example 3, Deterministic = true

```
14874  :  ['ar', 'country', 'water', 'lit', 'lillies', 'been', 'scarf', 'off', 'has', 'evergreen', 'have', 'lake', 'taxiing',
'am', '<unk>', 'colored', 'areas', 'cops', 'entryway']
```

Figure 38: RNN Good Example 3, Temperature = 5

```
14874  :  ['a', 'person', 'on', 'skis', 'standing', 'in', 'the', 'snow', '.']
```

Figure 39: RNN Good Example 3, Temperature = 0.001




```
bad  1
actual captions: ['There is a horse drawn carriage on the side of the road.', 'A carriage attached to a horse on the side of th
e road while cars go by. ', 'A horse pulling a wagon on the side of the road.', 'a horse pulls a cart next to a bridge ', 'A wh
ite horse is on the side of a road with a carraige. ']
predict captions: ['a', 'herd', 'of', 'cattle', 'walk', 'down', 'a', 'street', '.']
Test Performance: Bleu1: 44.44444444444444, Bleu4: 1.666666666666667
```

Figure 40: RNN Bad Example 1

```
559902  :  ['<start>', 'a', 'man', 'riding', 'a', 'horse', 'drawn', 'carriage', 'down', 'a', 'street', '.', '<end>']
```

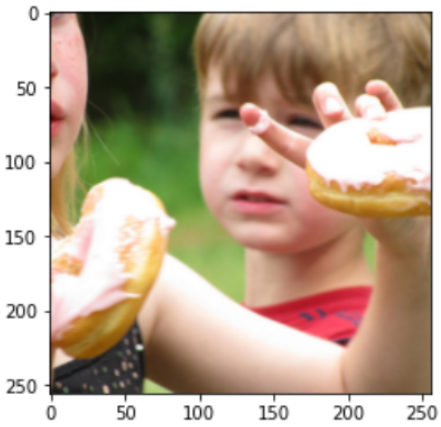
Figure 41: RNN Bad Example 1, Deterministic = true

```
559902  :  ['dog', 'wearing', 'graffiti', 'unhealthy', 'pets', 'ribbons', 'llittle', 'mounds', 'bros', 'come', 'alien', 'soap
s', 'skateboard', 'showing', 'umbrealla', 'types', 'crowded', 'ninth', 'divides', '/']
```

Figure 42: RNN Bad Example 1, Temperature = 5

```
559902  :  ['a', 'man', 'riding', 'a', 'horse', 'drawn', 'carriage', 'down', 'a', 'street', '.']
```

Figure 43: RNN Bad Example 1, Temperature = 0.001




```
bad  2
actual captions: ['A little girl holding twp donuts in her hands.', 'Two kids with donuts and frosting on their fingers.', 'A g
irl holds two donuts as a boy watches', 'A girl eats doughnuts in front of a boy', 'a boy and a girl and the girl is holding tw
o pink donuts']
predict captions: ['a', 'man', 'with', 'a', 'hot', 'dog', 'in', 'a', 'bun', 'eating', 'a', 'piece', 'of', 'pizza', '.']
Test Performance: Bleu1: 40.0, Bleu4: 0.8333333333333335
```

Figure 44: RNN Bad Example 2

```
334686  :  ['<start>', 'a', 'man', 'with', 'a', 'hot', 'dog', 'in', 'his', 'hand', 'and', 'eating', 'a', 'hot', 'dog', '.', '<e
nd>']
```

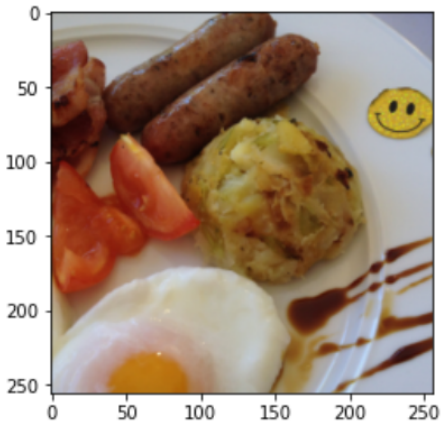
Figure 45: RNN Bad Example 2, Deterministic = true

```
334686  :  ['arms', 'their', 'electric', 'still', 'handmade', 'him', 'cakes', 'sumptuous', 'upward', 'stretching', 'people', 'h
older', 'tubular', 'furnitures', 'figure', 'research', 'move', 'place', 'outlook']
```

Figure 46: RNN Bad Example 2, Temperature = 5

```
334686  :  ['a', 'man', 'with', 'a', 'hot', 'dog', 'in', 'his', 'hand', 'and', 'eating', 'a', 'hot', 'dog', '.']
```

Figure 47: RNN Bad Example 2, Temperature = 0.001




```
bad  3
actual captions: ['A white plate topped with different breakfast foods.', 'A breakfast plate with bacon, eggs, sausage and mor
e.', 'On the plate is eggs,tomatoes sausage, and some bacon.', 'This breakfast has an egg, two sausage, tomatoes, potatoes, bac
on, and ham.', 'a plate an egg tomato and sausage on it ']
predict captions: ['a', 'plate', 'with', 'a', 'sandwich', 'cut', 'in', 'half', 'and', 'a', 'knife', '.']
Test Performance: Bleu1: 41.66666666666667, Bleu4: 1.1111111111111112
```

Figure 48: RNN Bad Example 3

```
402867  :  ['<start>', 'a', 'plate', 'of', 'food', 'with', 'a', 'fork', 'and', 'a', 'fork', '.', '<end>']
```

Figure 49: RNN Bad Example 3, Deterministic = true

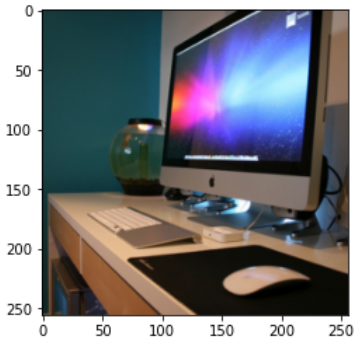
```
402867  :  ['toast', 'wooden', 'ready', 'sneaks', 'decorations', 'baked', 'chilies', 'courtesy', 'hands', 'each', 'in', 'balloo
ns', 'shoebox', 'written', 'where', 'term', 'pottery', 'photo', 'pods']
```

Figure 50: RNN Bad Example 3, Temperature = 5

```
402867  :  ['a', 'plate', 'of', 'food', 'with', 'a', 'fork', 'and', 'a', 'fork', '.']
```

Figure 51: RNN Bad Example 3, Temperature = 0.001

4.4.3 Architecture 2




```
good  1
actual captions: ['A flat screen monitor shows a brightly colored image on the screen.', 'A desktop computer monitor sitting on top of a des
k next to a mouse.', 'Computer setup with a multiple colored screen on it. ', 'An Apple computer monitor sits on a desk.', 'A computer scree
n has many colors, with a mouse and speakers next to it on the table.']
predict captions: ['a', 'computer', 'screen', 'and', 'a', 'keyboard', 'on', 'a', 'table', '.']
Test Performance: Bleu1: 90.0, Bleu4: 1.4285714285714295
```

Figure 52: Architecture 2 Good Example 1

```
186282  :  ['<start>', 'a', 'computer', 'monitor', 'and', 'a', 'keyboard', 'on', 'a', 'desk', '.', '<end>']
```

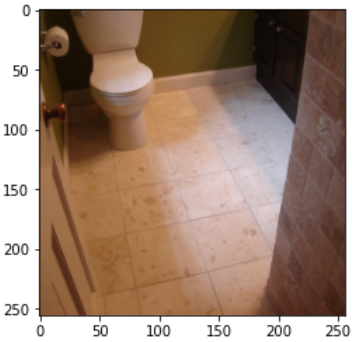
Figure 53: Architecture 2 Good Example 1, Deterministic = true

```
186282  :  ['peripheral', 'insides', 'pie', 'design', 'blob', 'think', 'people', 'much', 'placed', 'shop', 'scissors', 'slate', 'uses', 'mon
key', 'shells', 'showing', 'rad', 'like', 'names']
```

Figure 54: Architecture 2 Good Example 1, Temperature = 5

```
186282  :  ['a', 'computer', 'monitor', 'and', 'a', 'keyboard', 'on', 'a', 'desk', '.']
```

Figure 55: Architecture 2 Good Example 1, Temperature = 0.001




```
good  2
actual captions: ['A small bathroom with a white toilet and tan tile.', 'An open door reveals a white toilet and black cabinet.', 'A  toile
t, toilet paper and tile floor and walls. ', 'A toilet and toilet paper in a bathroom.', 'looking through an open door at a bathroom with a
toilet']
predict captions: ['a', 'bathroom', 'with', 'a', 'toilet', ',', 'sink', 'and', 'a', 'toilet', '.']
Test Performance: Bleu1: 81.81818181818183, Bleu4: 25.0
```

Figure 56: Architecture 2 Good Example 2

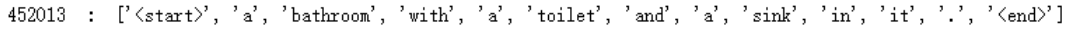


Figure 57: Architecture 2 Good Example 2, Deterministic = true

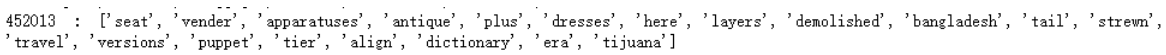


Figure 58: Architecture 2 Good Example 2, Temperature = 5

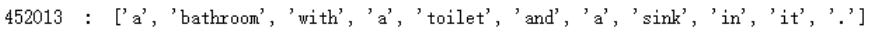
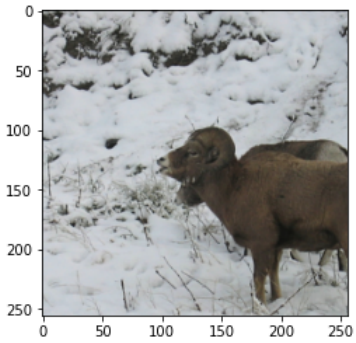


Figure 59: Architecture 2 Good Example 2, Temperature = 0.001




```
good  3
actual captions: ['A pair of rams graze through the snow on the side of a hill.', 'Two sheep stand in the snow and eat grass. ', 'A ram and
another sheep standing in the snow.', 'Two rams stand in the snow up to their ankles', 'Two rams standing together in front of a snowy emban
kment.']
predict captions: ['a', 'herd', 'of', 'sheep', 'standing', 'in', 'the', 'middle', 'of', 'the', 'snow', '.']
Test Performance: Bleu1: 83.33333333333334, Bleu4: 11.111111111111109
```

Figure 60: Architecture 2 Good Example 3

```
65239  :  ['<start>', 'a', 'herd', 'of', 'sheep', 'standing', 'in', 'the', 'grass', '.', '<end>']
```

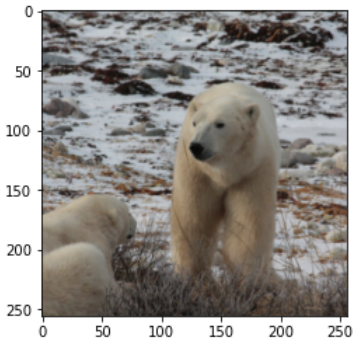
Figure 61: Architecture 2 Good Example 3, Deterministic = true

```
65239  :  ['sleepily', 'spectacular', 'pick', 'very', 'swim', 'murdered', 'wading', 'letters', 'pair', 'closely', 'filtred', 'case', 'too',
'hairdryer', 'tusks', 'baggage', 'heard', 'audience', 'indians', 's']
```

Figure 62: Architecture 2 Good Example 3, Temperature = 5

```
65239  :  ['a', 'herd', 'of', 'sheep', 'standing', 'in', 'the', 'grass', '.']
```

Figure 63: Architecture 2 Good Example 3, Temperature = 0.001




```
bad  1
actual captions: ['there are two polar bears standing near each other', 'A couple of white bears in the snow.', 'Two cute polar bears togeth
er in a snowy field.', 'A pair of polar bears gazing at each other while in a field of snow.', 'This polar bear is looking at the polar bear
that is laying on the ground.']
predict captions: ['a', 'herd', 'of', 'sheep', 'on', 'a', 'rocky', 'surface']
Test Performance: Bleu1: 44.124845129229776, Bleu4: 1.764993805169191
```

Figure 64: Architecture 2 Bad Example 1

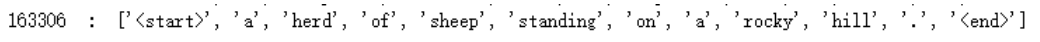


Figure 65: Architecture 2 Bad Example 1, Deterministic = true

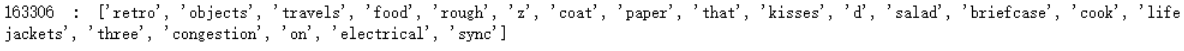


Figure 66: Architecture 2 Bad Example 1, Temperature = 5

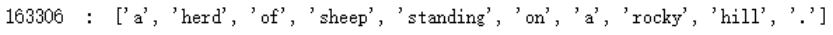
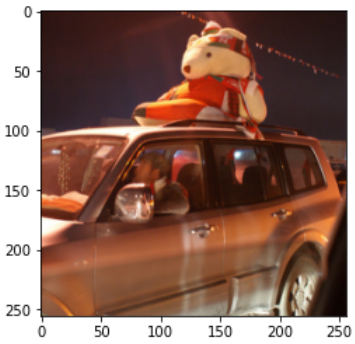


Figure 67: Architecture 2 Bad Example 1, Temperature = 0.001




```
bad  2
actual captions: ['A large white teddy bear sitting on top of an SUV.', 'A christmas bear on top of the roof of a truck', 'A stuffed animal
is sitting on the top of a vehicle.', 'A jeep with a giant christmas teddy bear strapped to the stop.', 'A large teddy bear attached to the
roof of an SUV.']
predict captions: ['a', 'close', 'up', 'of', 'a', 'car', 'with', 'a', 'dog']
Test Performance: Bleu1: 35.588329018524796, Bleu4: 1.33456233819468
```

Figure 68: Architecture 2 Bad Example 2

```
245013  :  ['<start>', 'a', 'close', 'up', 'of', 'a', 'car', 'with', 'a', 'dog', 'in', 'the', 'background', '<end>']
```

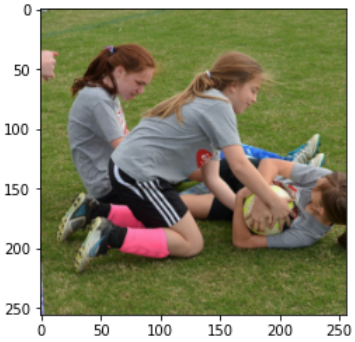
Figure 69: Architecture 2 Bad Example 2, Deterministic = true

```
245013  :  ['garages', 'desert', 'bot', 'lit', 'ferrari', 'vinyl', 'fire', 'zoo', 'used', 'tomatoes', 'wonderland', 'set', 'holder', 'forwar
ds', 'weight', 'arranges', 'stumbling', 'arranges', 'return', 'promote']
```

Figure 70: Architecture 2 Bad Example 2, Temperature = 5

```
245013  :  ['a', 'close', 'up', 'of', 'a', 'car', 'with', 'a', 'dog', 'in', 'the', 'background']
```

Figure 71: Architecture 2 Bad Example 2, Temperature = 0.001




```
bad  3
actual captions: ['A group of pretty young ladies playing a game of soccer.', 'team of girls trying to get the soccer ball', 'THREE YOUNG GI
RLS PLAYING BALL, WITH ONE OF THE GIRLS TRYING TO TAKE THE BALL AWAY', 'A lot of kids that are having some fun.', 'One girl trying to rip ou
t the soccer ball while others watch.']
predict captions: ['a', 'woman', 'sitting', 'on', 'a', 'grass', 'covered', 'field', 'with', 'a', 'frisbee', '.']
Test Performance: Bleu1: 33.33333333333333, Bleu4: 1.1111111111111112
```

Figure 72: Architecture 2 Bad Example 3

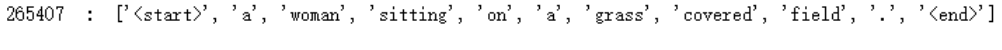


Figure 73: Architecture 2 Bad Example 3, Deterministic = true

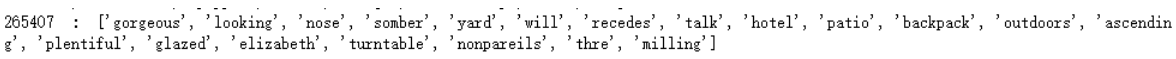


Figure 74: Architecture 2 Bad Example 3, Temperature = 5

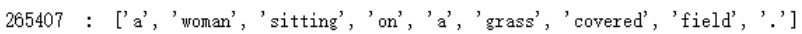


Figure 75: Architecture 2 Bad Example 3, Temperature = 0.001

5 Discussion

5.1 Discuss your results. Why do you think the results were the way they were?

First, we can find that for each image when the temperature is super low then the then it will produce the same result as deterministic, this because when is super low high value will be the dominant element with almost 100% to be the one be chosen. When the temperature is supper high then we just choose the word uniform randomly which just produces a sentence with the random word and makes no sense.
Overall We found that even though the architecture of the models is different, the performance is different. But they have similarities in the image of good and poor performance which means they trend to performance bad or good on the similar kind of image. We found that most of the images where they performed well were that the image features were more obvious and easy to distinguish, such as very bright and unique colors or unique shapes or both, which made them easier to associate image features with specific words and distinguish them. The bad performance images are generally more deceptive, confusing objects of a similar color or shape so they will misidentify them, or the subject of the image that the model pays attention to is different from what humans focus on, so they got a bad score. But overall A2 has the best performance, followed by baseline, and vanilla RNN is the worst. Stochastic is better than deterministic.

5.2 Why do you think the deterministic approach does not work well?

I think the reason why the performance of the deterministic approach is worse is mostly that the way that depends on how the model generates the caption, our output each time depends on our last prediction, we seem to be making the best choice at every step, but in fact, this choice method has a high probability can't reach the overall optimal solution. For example, if I still have 3 words to predict, predicting the optimal only depending on the last caption will indeed achieve the local optimal solution, but if we predict other possible words depending on the possibility according to the input, it may allow us to achieve a better overall result which means better captions, and this method has a high probability of getting us to an overall better solution than deterministic. Therefore, in most cases, the performance of using deterministic will be worse.

6 Team contributions

1) Enze Ma works on part of experiment.py, model_factory.py, modify the main.py, default.json and write the report
2) Yixin Jiang works on part of experiment.py, model_factory.py, modify the main.py, default.json and write the report
3) Fangqi Yuan works on part of experiment.py, model_factory.py, modify the main.py, default.json and write the report

7 Reference

[1] Rafi, S. & Das, R. (2021). RNN Encoder And Decoder With Teacher Forcing Attention Mechanism for Abstractive Summarization. In *2021 IEEE 18th India Council International Conference (INDICON)*, pp. 1-7, doi: 10.1109/INDICON52576.2021.9691681.

[2] Huang Y, Chen J. (2020). Teacher-critical training strategies for image captioning. arXiv:2009.14405.