Danial Zohourian Image Captioning Report

I implemented four distinct models for an Image Captioning task on the Flickr30k dataset. The evaluation metrics used are BLEU and ROUGE scores, which we will talk about later. All of the models followed the same approach: a pre-trained CNN (ResNet 50) for image feature extraction and an RNN for caption generation. In the first model (**Vanilla**), a simple RNN was used. In the second model (**LSTM**), the simple RNN was turned into an LSTM. In the third model (**Attention**), Attention mechanism was added and finally in the fourth method (**GloVe**), pre-trained word embeddings were implemented.

In this report, I will first explain the mentioned evaluation metrics, Explain the model structure, and then, compare the results of each model.

Metrics

1. **BLEU (Bilingual Evaluation Understudy):**
    - **Purpose:** BLEU is primarily used to evaluate the quality of machine-generated text in tasks like machine translation.
    - **Calculation:** BLEU measures the precision of n-grams (contiguous sequences of n items, usually words) in the generated text compared to a set of reference texts. It computes a score between 0 and 1, where 1 indicates perfect match.
    - **Advantages:** Simple and easy to compute; widely used in machine translation evaluations.
    - **Limitations:** Insensitive to word order and can favor shorter sentences.
2. **METEOR (Metric for Evaluation of Translation with Explicit ORdering):**
    - **Purpose:** METEOR is designed to overcome some of the limitations of BLEU by considering synonyms and stemming.
    - **Calculation:** METEOR computes precision, recall, and F1-score based on unigram matches, stemming, synonymy, and word order.
    - **Advantages:** Incorporates more linguistic aspects, including synonyms and stemming; relatively robust.
    - **Limitations:** Can be sensitive to parameter tuning.
3. **CIDEr (Consensus-based Image Description Evaluation):**
    - **Purpose:** CIDEr is commonly used in the evaluation of image captions and text summarization tasks.
    - **Calculation:** CIDEr evaluates the consensus among multiple reference texts by measuring n-gram similarity. It gives higher scores to diverse and semantically rich outputs.
    - **Advantages:** Focuses on capturing the consensus and relevance of generated text; considers semantic content.
    - **Limitations:** May be influenced by the number of reference texts.

4. **ROUGE (Recall-Oriented Understudy for Gisting Evaluation):**
    - **Purpose:** ROUGE is widely used for summarization and text generation tasks.
    - **Calculation:** ROUGE measures the overlap of n-grams and word sequences between the generated text and reference texts. It includes various versions like ROUGE-N (unigrams, bigrams, etc.), ROUGE-L (longest common subsequence), and ROUGE-W (word overlap).
    - **Advantages:** Captures the recall of important content; flexible with different n-gram lengths.
    - **Limitations:** Limited in measuring semantic similarity and can be sensitive to text length.

In this project, I have used BLEU, ROUGE-1, ROUGE-2 and ROUGE-L to evaluate the model. I have also conducted self-evaluations by checking the outcome of each model.

Models

In every model, I have done the following procedure:

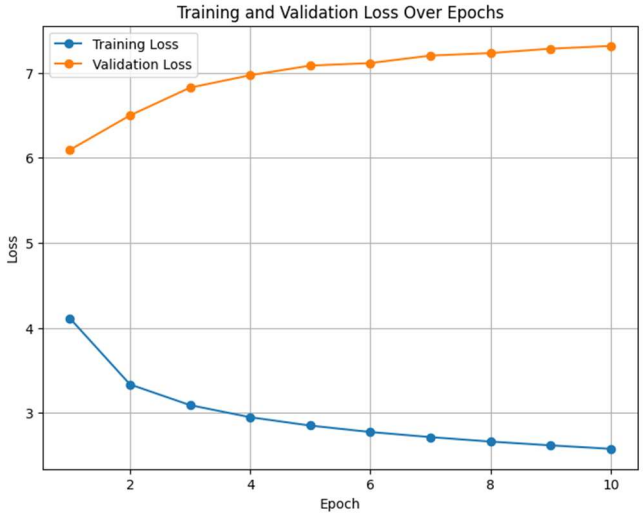
1. Reading the dataset from the added Flickr30k dataset in Kaggle.
2. Creating a Vocabulary function to build a vocabulary for a given set of sentences, tokenize text, and convert text into numericalized form using the created vocabulary. In the **GloVe** model, pre-trained word embeddings were also added to this function.
3. Splitting the data into train, validation, and test sets.
4. Creating the dataloaders to feed the models.
5. Defining the corresponding model, with a ResNet50 CNN **Encoder**, An RNN based **Decoder** and a **EncoderDecoder** class to define the whole model.
6. Initializing the model with the same hyperparameters. In the **Attention** model, the batch size was lower because the Kaggle GPU couldn't handle it.
7. Training the model for 10 epochs and writing the metrics in each one.
8. Testing the model on the test data.
9. Plotting the metrics.
10. Inference. The model parameters are saved in the glove_model.pth file and can be used in the GloVe.ipynb file to check how the model works.

The models were trained on GPU P100 in a Kaggle environment. Each epoch took 17-25m with respect to the model.

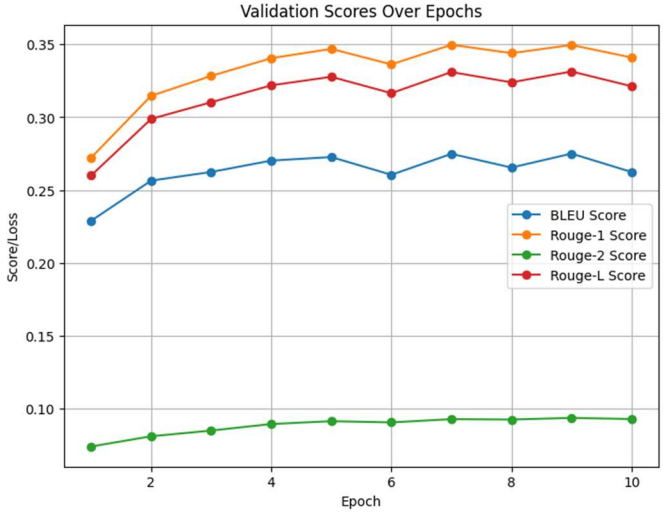
Following, are the results and comparisons for each of the models, **Vanilla**, **LSTM**, **Attention** and **GloVe**.

1. Vanilla

This is the baseline model: a pre-trained CNN (ResNet 50) for image feature extraction and a simple RNN for caption generation.



We can see that the validation loss increases over each epoch, which is usually not a good sign. But after experimenting I saw that the model is, in fact, getting better at captioning the images. The first instinct that comes to mind to be the reason of this, is overfitting, but I think that might not be the case here because it is increasing from the very beginning. I think with increasing the epoch count, we can observe the validation loss to decrease eventually. Also, using regularization techniques can also help this issue.

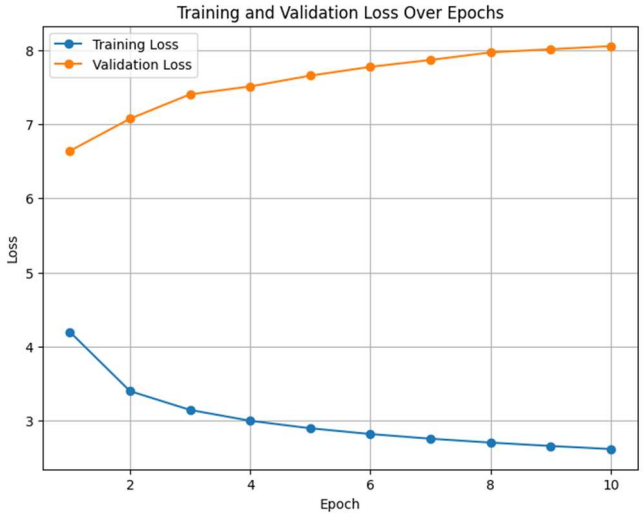


We can see that the scores tend to go up after each epoch which is a good sign. ROUGE scores over 40 and BLEU scores over 30 are considered acceptable. With more epochs, we may achieve that. The evaluations on the test set are as follows:

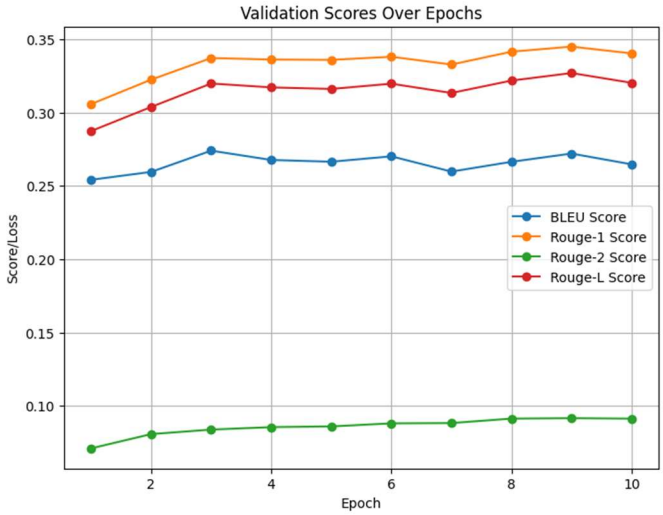
Loss: 9.1123, Rouge 1: 0.3412, Rouge 2: 0.0916, Rouge L: 0.3215, BLEU: 0.2627

2. LSTM

In this model, I turned the simple RNN into an LSTM cell.



This is the same as before. But with a lower overall loss value.



The scores also behave the same as before, but are a bit better. The evaluations on the test set are as follows:

Loss: 8.0071, Rouge 1: 0.3400, Rouge 2: 0.0915, Rouge L: 0.3205, BLEU: 0.2641

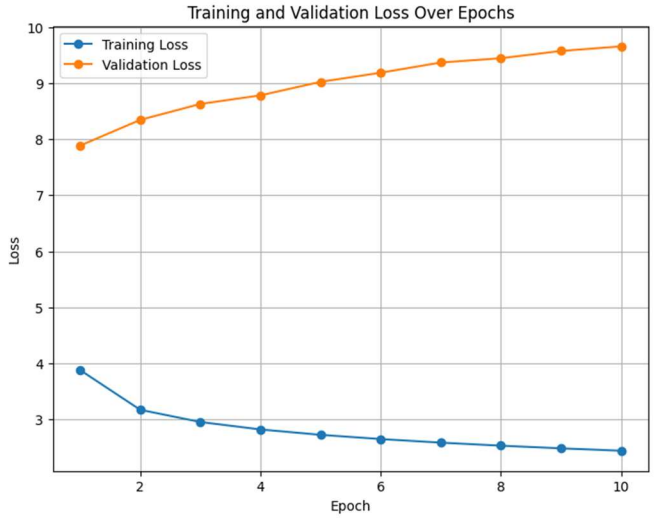
LSTM vs. Vanilla:

- Loss: LSTM (8.0071) < Vanilla (9.1123) - LSTM has a lower test loss.
- Rouge 1: LSTM (0.34) ~ Vanilla (0.3412) - Comparable Rouge 1 scores.
- Rouge 2: LSTM (0.0915) ~ Vanilla (0.0916) - Comparable Rouge 2 scores.
- Rouge L: LSTM (0.3205) ~ Vanilla (0.3215) - Comparable Rouge L scores.
- BLEU: LSTM (0.2641) > Vanilla (0.2627) - LSTM has a slightly higher BLEU score.

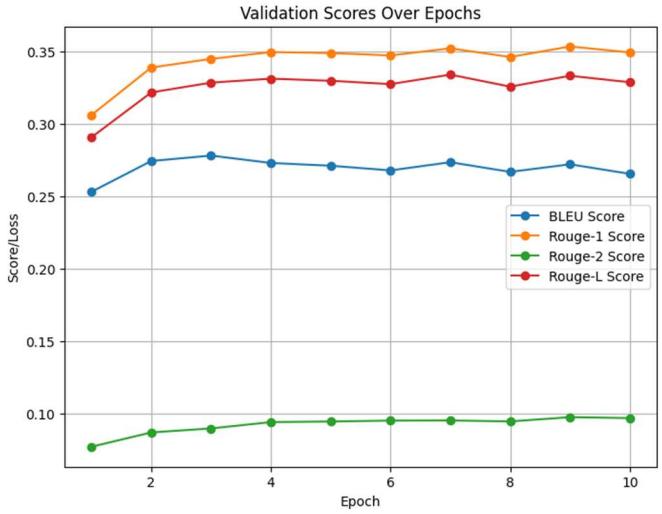
LSTM outperforms Vanilla in terms of test loss and BLEU score, but they have comparable Rouge scores.

3. Attention

In this model, an Attention mechanism as in the 'Attention is all you need' paper, Bahdanau, is added.



The same as before. Higher overall validation loss.



The trends in scores are as before, but higher overall. This shows that the attention mechanism has shown reasonable increase in the evaluation metrics. The evaluations on the test set are as follows:

Loss: 8.9839, Rouge 1: 0.3461, Rouge 2: 0.0945, Rouge L: 0.3258, BLEU: 0.2649

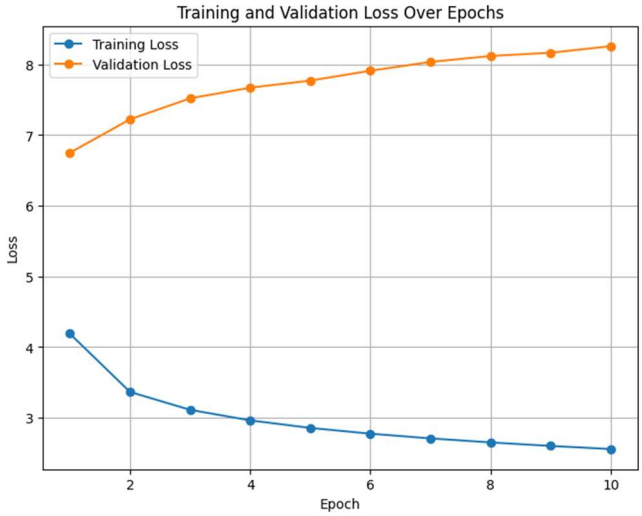
Attention vs. LSTM:

- Test Loss: Attention (8.9839) > LSTM (8.0071) - LSTM has a lower test loss.
- Rouge 1: Attention (0.3461) > LSTM (0.34) - Attention has a higher Rouge 1 score.
- Rouge 2: Attention (0.0945) > LSTM (0.0915) - Attention has a higher Rouge 2 score.
- Rouge L: Attention (0.3258) > LSTM (0.3205) - Attention has a higher Rouge L score.
- BLEU: Attention (0.2649) > LSTM (0.2641) - Attention has a higher BLEU score.

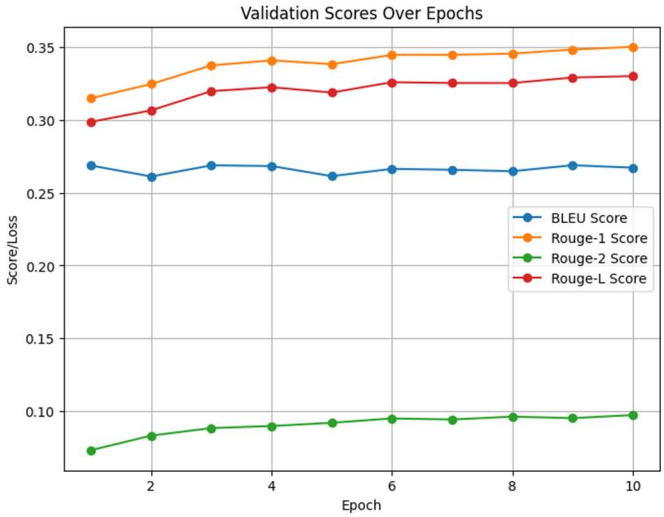
Attention outperforms LSTM in terms of Rouge scores and BLEU score, but LSTM has a lower test loss.

4. GloVe

In this model, pre-trained GloVe word embeddings are implemented to the last model.



The behavior is still the same, but the overall loss is lower.



The scores are increasing more steadily, especially the ROUGE scores. With higher epochs, this model will probably perform better. The evaluations on the test set are as follows:

Loss: 7.1381, Rouge 1: 0.3520, Rouge 2: 0.0975, Rouge L: 0.3318, BLEU: 0.2679

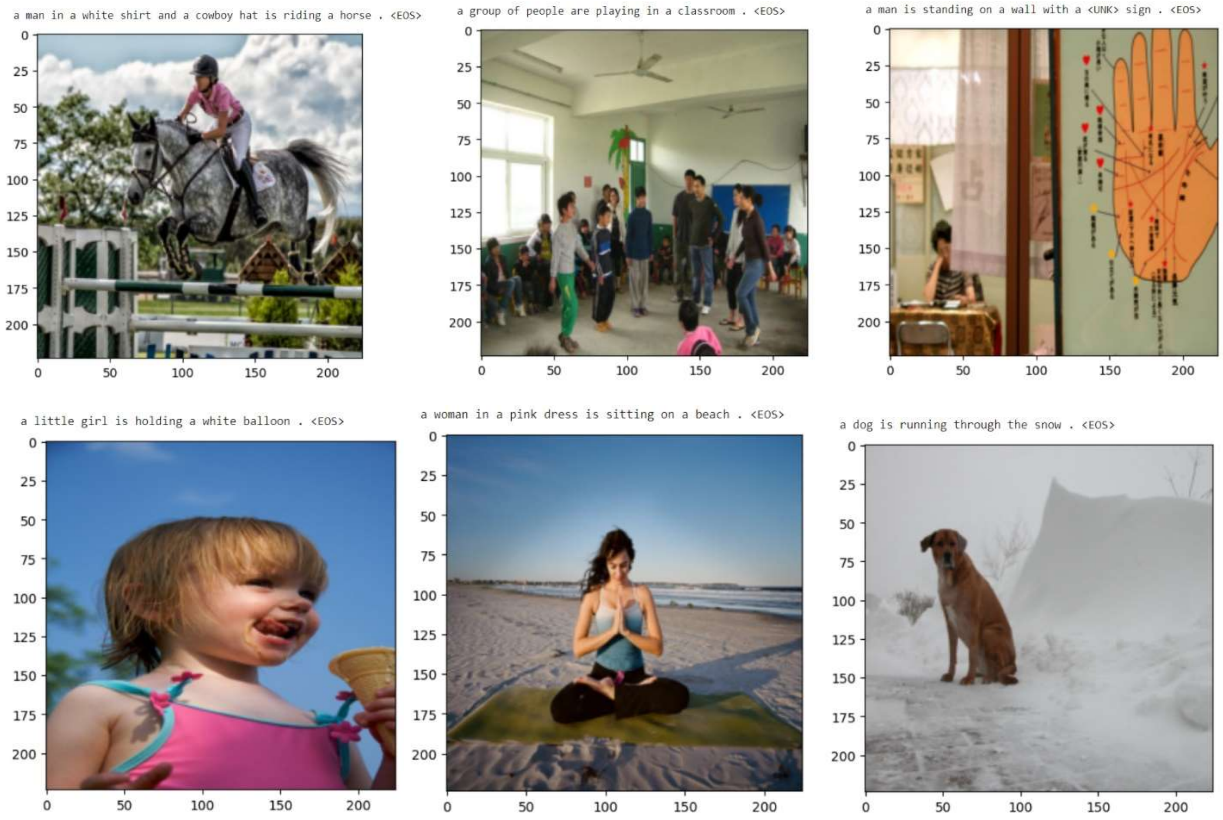
GloVe vs. Attention:

- Test Loss: GloVe (7.1381) < Attention (8.9839) - GloVe has a lower test loss.
- Rouge 1: GloVe (0.3520) > Attention (0.3461) - GloVe has a higher Rouge 1 score.
- Rouge 2: GloVe (0.0975) > Attention (0.0945) - GloVe has a higher Rouge 2 score.
- Rouge L: GloVe (0.3318) > Attention (0.3258) - GloVe has a higher Rouge L score.
- BLEU: GloVe (0.2679) > Attention (0.2649) - GloVe has a higher BLEU score.

GloVe outperforms Attention in terms of test loss, Rouge scores, and BLEU score.

Inference

These are some of the outputs of the final GloVe model, on the test set:



We can see that the model is:

- Capable of producing understandable English.
- Differentiate children from adults.
- Good at identifying objects and animals.
- Weak at identifying colors.
- Good at identifying environments.