Image Captioning using LSTM and GRU

Farah Aymen
Faculty of Informatics and Computer Science- AI Major
The British University in Egypt
Cairo, Egypt
farah194233@bue.edu.eg

Ashraf Adel
Faculty of Informatics and Computer Science- AI Major
The British University in Egypt
Cairo, Egypt
ashraf196280@bue.edu.eg

Jacinta Samir
Faculty of Informatics and Computer Science- AI Major
The British University in Egypt
Cairo, Egypt
jacinta206562@bue.edu.eg

Mohamed Negm
Faculty of Informatics and Computer Science- AI Major
The British University in Egypt
Cairo, Egypt
mohamed206069@bue.edu.eg

***Abstract*— Image captioning is a challenging job that requires the generation of natural language descriptions for photographs. Long Short-Term Memory (LSTM) and GRU models have demonstrated outstanding performance in a variety of natural language processing tasks, including machine translation and speech recognition. In this study, we offer a novel strategy for image captioning that makes use of the power of LSTM and GRU. The LSTM outperformed the GRU but not by a big gap.**

Keywords—Long Short-Term Memory (LSTM), GRU, Image Captioning

I. Introduction

Image captioning, the task of generating natural language descriptions for images, has gained significant attention in the field of computer vision and natural language processing. It has diverse applications, including content creation, image retrieval, and accessibility for visually impaired individuals. Generating accurate and meaningful captions for images is a challenging problem due to the inherent complexity of visual understanding and language generation. To tackle this challenge, researchers have developed various deep learning-based approaches, including Long Short-Term Memory (LSTM) and Attention models, which have shown remarkable success in a wide range of natural language processing tasks. LSTMs are a type of recurrent neural network (RNN) that can capture long-term dependencies in sequential data, making them suitable for language generation tasks. In recent years, there has been growing interest in using LSTM for image captioning. The LSTM-based decoder can effectively generate sequential language descriptions, while the Attention mechanism allows the model to dynamically attend to relevant visual features, mimicking human visual attention. This approach has shown promising results in generating more coherent and contextually relevant image captions. This research paper proposes a novel approach that utilizes LSTM and GRU mechanisms for image captioning. Our model employs an encoder-decoder architecture, where the encoder uses Convolutional Neural Networks (CNNs) to extract visual features from the input image, and the decoder utilizes LSTMs/GRUs to generate captions. We conduct extensive experiments on benchmark datasets and compare our two models to showcase their effectiveness in generating accurate and descriptive image captions. We also conduct ablation studies to analyze the impact of different components of our model and investigate the sensitivity of our model to hyperparameters.

II. Related Work

A. Long-short Term Memory (LSTM)

LSTMs, introduced by Hochreiter and Schmidhuber in 1997 [1], are specifically designed to handle long-term dependencies in sequential data, making them well-suited for language modeling, sequence prediction, and other similar tasks. They have a more complex structure compared to traditional RNNs, with specialized gates that control the flow of information, allowing them to capture and retain important information from past time steps, while selectively updating and forgetting irrelevant information.

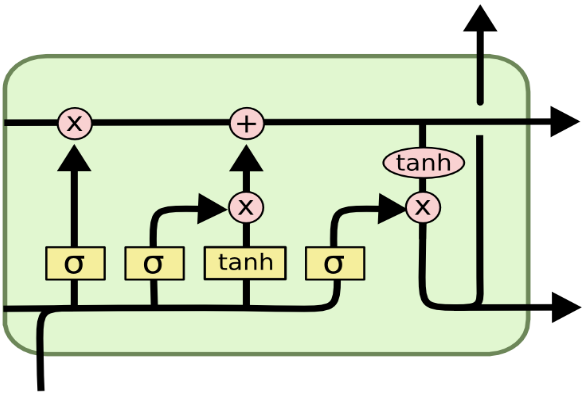


Figure 1 LSTM Architecture

The key components of an LSTM cell are the input gate, the output gate, and the forget gate. The input gate determines which parts of the current input should be stored in the cell state, the forget gate determines which parts of the previous cell state should be forgotten, and the output gate determines which parts of the current cell state should be outputted. These gates are controlled by learned parameters and are updated during training to optimize the model's performance. Its architecture allows for the modelling of long-range dependencies, making it particularly useful for tasks that require capturing contextual information from a large input sequence. In the context of image captioning, LSTMs can be used as a decoder to generate sequential language descriptions for images. The encoder part of the image captioning model extracts visual features from the input image, while the LSTM decoder generates the captions word-by-word, considering the contextual information captured by the LSTM's hidden states. Overall, LSTM is a powerful and widely used architecture for modelling sequential data, with the ability to capture long-term dependencies, making it suitable for tasks such as language generation, including image captioning [2].

B. Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) architecture that is similar to Long Short-Term Memory (LSTM) and is designed to address the issue of vanishing gradients in traditional RNNs during training. RNNs are neural networks that are designed to process sequential data, such as time series or sequences of text, by maintaining a hidden state that captures information from previous time steps. GRUs, introduced by Cho et al. in 2014 [3], are a type of RNN that simplifies the LSTM architecture by combining the hidden state and cell state into a single hidden state, making it computationally more efficient.

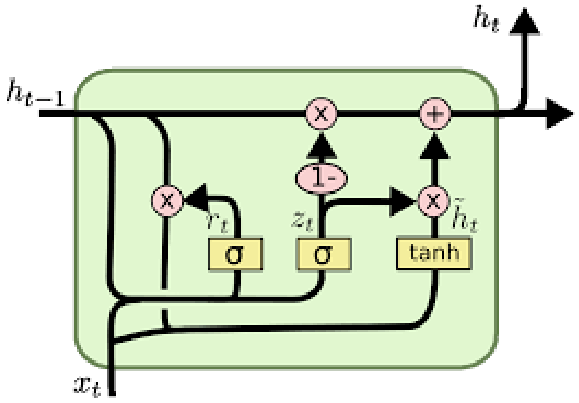


Figure 2 GRU Architecture

GRUs have a similar structure to LSTMs, with the addition of two gates: the reset gate and the update gate. The reset gate determines how much of the previous hidden state should be forgotten, and the update gate determines how much of the new hidden state should be updated with the current input. These gates are controlled by learned parameters and are updated during training to optimize the model's performance. The GRU architecture allows for the modeling of short- and long-term dependencies, making it suitable for tasks that require capturing contextual information from input sequences. In the context of language modeling, sequence prediction, and other similar tasks, GRUs can be used as an alternative to LSTMs for processing sequential data. GRUs have been shown to perform well in various natural language processing (NLP) tasks, such as machine translation, speech recognition, and sentiment analysis. GRUs can be used as a powerful tool for modeling sequential data, particularly in tasks that require capturing contextual information from input sequences. GRUs are known for their computational efficiency compared to LSTMs, making them suitable for large-scale applications where computational resources are limited. GRUs can be used to experiment with different architectures and hyperparameters to optimize model performance and to compare the performance of GRUs with other RNN architectures in their specific research domain. Overall, the GRU is a popular and effective RNN architecture for modeling sequential data, with the ability to capture short- and long-term dependencies, making it suitable for various scientific research applications, including language modeling, sequence prediction, and other similar tasks.

C. Image Captioning

In 2015, a research was conducted by Google on generating image captions where it first input the image on a deep Convolutional Neural Network (CNN) to process the image, then a language generating RNN that generates input images. The model is trained to maximize the likelihood to produce a sequence of words (description) based on the image it received [4].

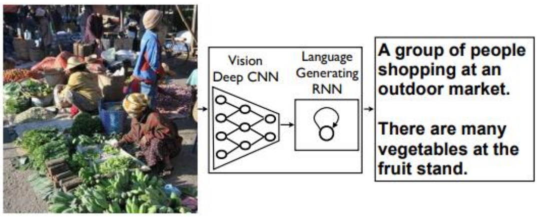

Vision
Deep CNN
Language
Generating
RNN
A group of people shopping at an outdoor market.
There are many vegetables at the fruit stand.

Figure 3 Google's Image Caption Generator's Concept

The result of their model reached a BLEU-1 score of 27.1 which outperformed other models they compared it with. Another research was done in 2022 where it introduces an expansion mechanisim that During the forward phase, the input data is transformed into a new one with a variable sequence length, and the opposite operation is performed in the backward pass to allow the network to process the input without restriction. the number of elements. The Block Static Expansion allows you to perform these operations on a collection of arbitrary and diverse lengths all at once [5]. The model is composed of the standard encoder decoder structure implemented on top of the Swin-Transformer.

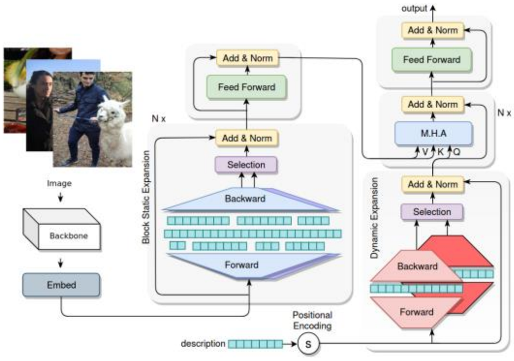


Figure 4 ExpansionNet V2's architecture

III. Methodology

Two models were developed for this study an LSTM and a GRU. The main idea behind the architecture of both models is to feed the caption to the LSTM/GRU and encode the image, then feed both to a feedforward network to predict the next word of the generated caption.

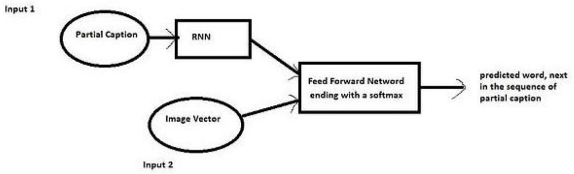
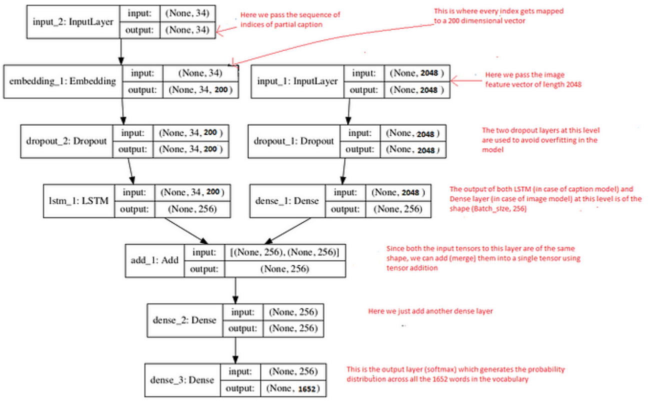

Input 1
Partial Caption
RNN
Feed Forward Netword ending with a softmax
predicted word, next in the sequence of partial caption
Image Vector
Input 2

Figure 5 Our paper's concept

The dataset that has been used in Flickr8K which contains 8k images where each image has 5 captions. The data was split into a dataset of 6000 images, validation set of 1000 images and a test set with 1000 images. The first step taken was the data preprocessing. To clean the data, we first removed the stop words from the captions then selected all the unique words out of them to create the corpus. This left us with 8763 unique words across the 40,000 captions. Looking at the words, we will see that there are some that didn't occur quite frequently enough, so we set a threshold of the frequency of the words to filter them from the less frequent ones. The threshold was at frequency equal to 10. This means any word that appeared less than 10 times should be removed. This left us with 1652 unique words. To mark the beginning and the ending of the caption, each caption was concatenated with 'startseq' at the beginning of the caption and 'endseq' at the end of the caption. Regarding the images, they were converted into vectors. These vectors represent the features of each image. To extract the features, Google's Inception model was used to extract a vector of length 2048 for each image. Looking back at the captions, each caption was encoded to a vector of a fixed size so it would be appropriate to feed to the model. To determine this fixed size, we calculated the length of the longest caption to set it as the vector size which turned out to be 34. Next two dictionaries were created to assign and map each word in the corpus to an index. These dictionaries are used later to map the words after getting the probability of the predictions. Next a generator was created to feed the input into batches to the model. The next step is creating the word embeddings. As discussed in the previous phase, the CBOW and skipgram weren't sufficient so we used a pre-trained model to generate the word embeddings which is Google's Glove.



Here we pass the sequence of indices of partial caption

This is where every index gets mapped to a 200 dimensional vector

Here we pass the image feature vector of length 2048

The two dropout layers at this level are used to avoid overfitting in the model

The output of both LSTM (in case of caption model) and Dense layer (in case of image model) at this level is of the shape (Batch_size, 256)

Since both the input tensors to this layer are of the same shape, we can add (merge) them into a single tensor using tensor addition

Here we just add another dense layer

This is the output layer (softmax) which generates the probability distribution across all the 1652 words in the vocabulary

Figure 6 LSTM Model Summary

Finally, we compiled the two models using the same hyperparameters so that we can properly compare them. We trained both on 30 epochs, used initial learning rate 0.001, 3 pictures per batch, used Adam optimizer and used ROUGE and BLEU metrics to evaluate them.

IV. RESULTS

The LSTM Model got a loss of 3.3268 while the GRU Model got loss of 3.225. Using the ROUGE and BLEU scores, the average ROUGE score of the LSTM was 0.19144 and the average BLEU score is 0.01695. Regarding the GRU, it had an average score for the ROUGE of 0.17871 and an average BLEU Score of 0.01569.

Table 1 Comparative analysis between the paper's LSTM and GRU

Model	Loss	BLEU	ROUGE
LSTM	3.3268	**0.01695**	**0.19144**
GRU	**3.225**	0.01569	0.17871

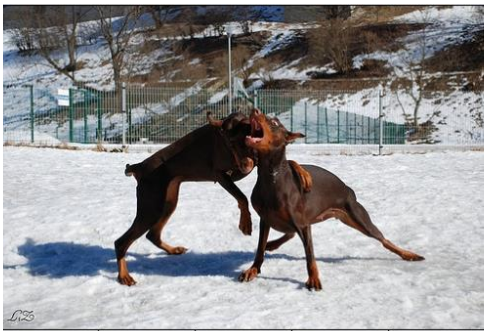


Figure 7 Test Image its captions were: 'The dogs are in the snow in front of a fence', 'The dogs play on the snow', 'Two brown dogs playfully fight in the snow', 'Two brown dogs wrestle in the snow', 'Two dogs playing in the snow' while the predicted cation for the LSTM was: 'brown dog is running on the grass'

Expiramenting on the model, figure 7 was tested on the LSTM, it predicted 'brown dog is running on the grass '. The resulted prediction seems to be adequate giving a good pridiction to the image. While the GRU's prediction was completely off the track as the predicted caption was 'outstretched directions campfire part striped masks vertical members almost masks'. This proved although the metrics' resulsts of the two models were close, the LSTM did a much better job than the GRU.

V. CONCLUSION

In Conclusion, the LSTM slightly performed better than the GRU, adding an attention layer to the model can improve its results and make it predict more accuratly. The GRU wasn't behind the LSTM by much as their architecture does not differ completely so it was expected to have close results between them.

REFERENCES

[1] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
[2] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks," Sep. 2019, [Online]. Available: http://arxiv.org/abs/1909.09586
[3] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.3555
[4] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and Tell: A Neural Image Caption Generator," Nov. 2014, [Online]. Available: http://arxiv.org/abs/1411.4555
[5] J. C. Hu, R. Cavicchioli, and A. Capotondi, "ExpansionNet v2: Block Static Expansion in fast end to end training for Image Captioning," Aug.

2022, [Online]. Available:
http://arxiv.org/abs/2208.06551