M.K.S.S.S's Cummins College of Engineering for Women
Department of Electronics & Telecommunication Engineering

Course: Deep Learning
A.Y. (2023-24)
Open Ended Assignment

Enhancing Neural Machine Translation: From Encoders and Decoders to Attention Mechanisms

Presented by:

GROUP No. 1

Project Group Members:

101 - Aarushi Bose
116 - Nupur Deshpande
119 - Priyanka Dhawale
122 - Diksha Prakash

Guided by:
Dr. Ashwini M. Deshpande
E&TC Department

Problem statement

This project aims to enhance the accuracy and naturalness of English-to-Marathi neural machine translation by progressively integrating encoder-decoder architectures with deep LSTM layers and attention mechanisms.

Abstract

The project aims to enhance the accuracy and fluency of English to Marathi neural machine translation (NMT) systems through a series of architectural improvements. Initially, employing traditional encoder-decoder structures for sequence modeling, the system faced limitations in capturing nuanced linguistic features, leading to suboptimal translation quality. To address this, the model was upgraded to incorporate deep LSTM architectures, enabling the learning of hierarchical representations for better understanding and translation of complex sentence structures. Additionally, attention mechanisms were introduced to improve alignment and focus during the translation process, facilitating more accurate word mapping between English and Marathi languages. By iteratively refining the system architecture through these advancements, the goal is to achieve higher translation accuracy, semantic coherence, and naturalness in Marathi translations of English text.

Literature Survey

S. No.	Title of Paper	Name of Author	Year of Publication	Summary and Results
1.	"Sequence to Sequence Learning with Neural Networks"	Sutskever et al.	2014	seq2seq model architecture, employing recurrent neural networks (RNNs) for both encoder and decoder
2.	"Neural Machine Translation by Jointly Learning to Align and Translate"	Bahdanau et al.	2015	Address the issue of information compression in the fixed-length context vector of traditional seq2seq models
3.	"Effective Approaches to Attention-based Neural Machine Translation"	Luong et al.	2015	Explored various attention mechanisms and their impact on translation quality and computational efficiency.
4.	"Attention is All You Need"	Vaswani et al.	2017	Eliminates recurrence entirely in favor of self-attention mechanisms
5.	"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"	Devlin et al.	2018	A pre-trained Transformer-based model for various natural language understanding tasks

Software used

Google Colab is a free cloud-based platform that offers access to GPUs and TPUs for running Python code in interactive notebooks.The program was implemented using TensorFlow and Keras libraries for deep learning tasks. It was executed on a T4 GPU provided by Google Colab for efficient computation.

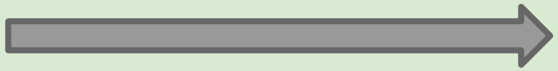
Dataset Information

The dataset comprises tab-delimited bilingual sentence pairs extracted from the Tatoeba Project, a collaborative online database of sentences and translations in multiple languages. The sentences are sourced from various contributors, ensuring a diverse range of language usage and contexts. With over 6 million sentence pairs available, the dataset offers a rich resource for natural language processing tasks such as translation, language modeling, and sentiment analysis. For this specific project, a subset of approximately 38,695 sentence pairs has been selected for analysis and model training. This subset is sufficiently large to train robust machine learning models while being manageable in terms of computational resources and processing time.

```
with open('/content/mar.txt','r') as f:
  data = f.read()
```

```
len(data)
```

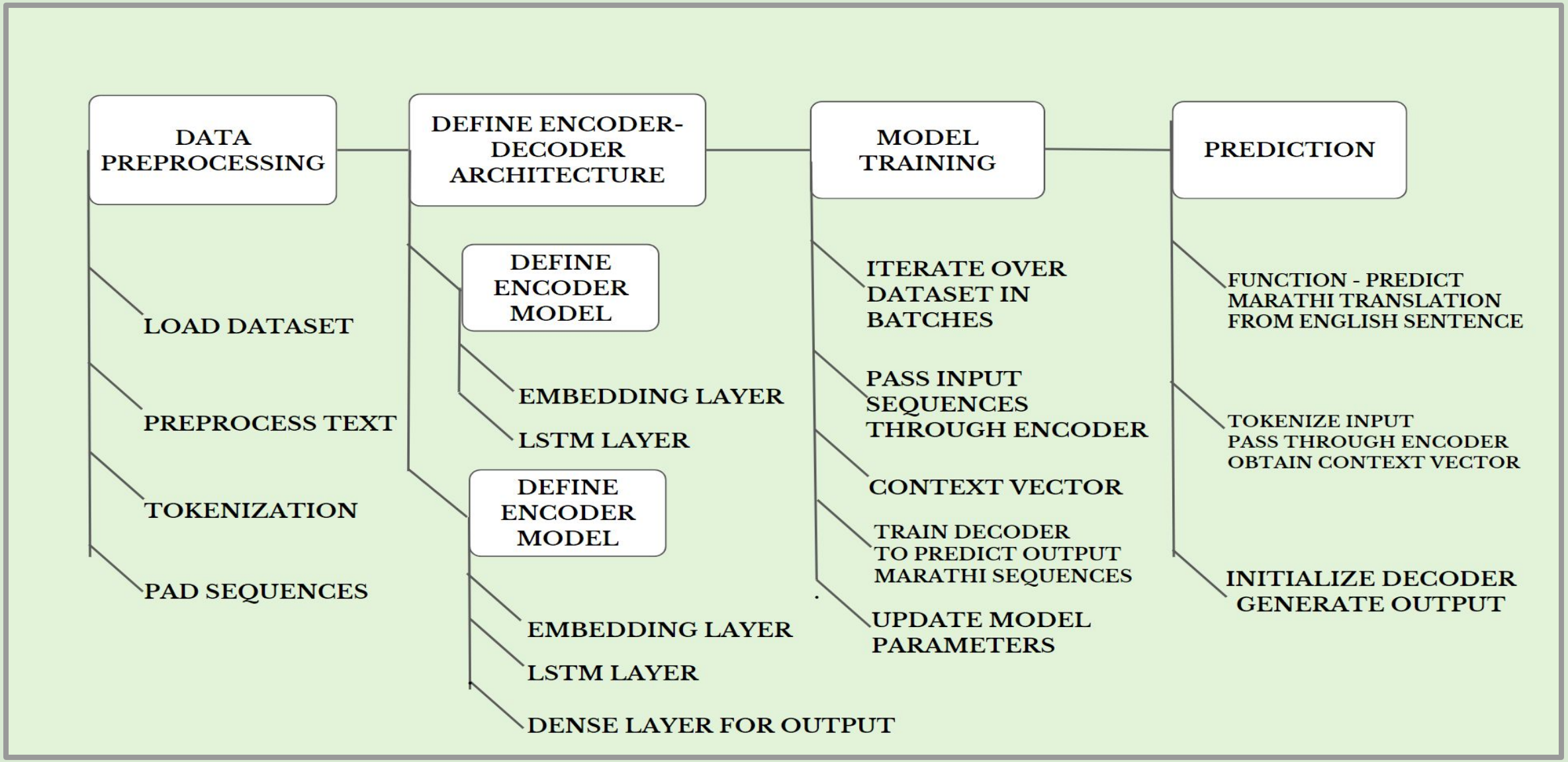
```
6355807
```



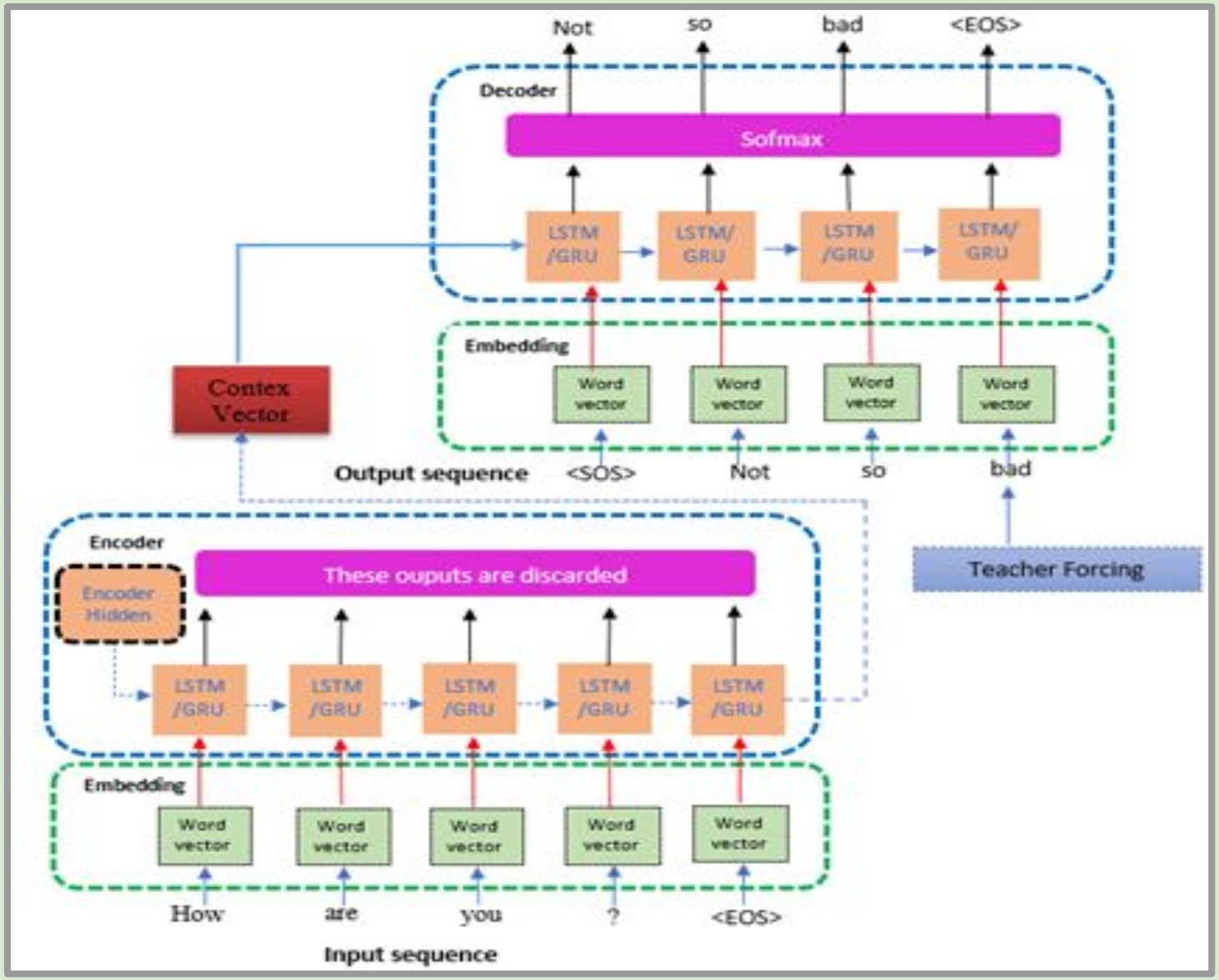
```
language_data.shape
```

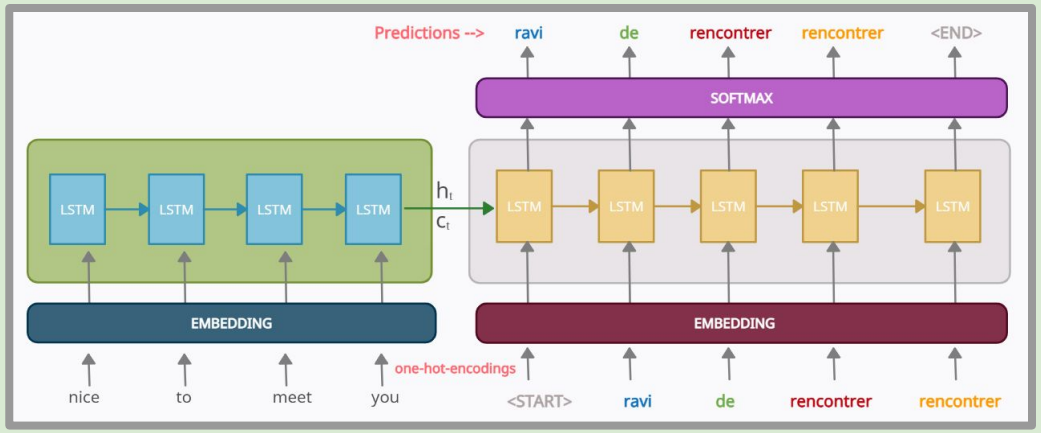
```
(38695, 2)
```

Detailed Block Diagram



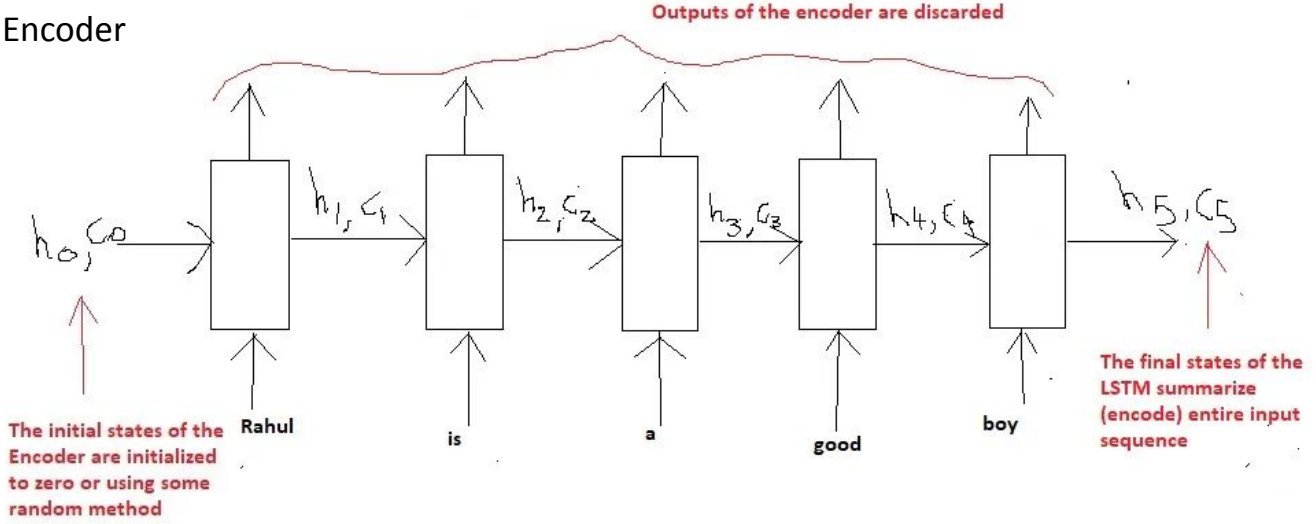
Model Architecture (Encoder - Decoder)

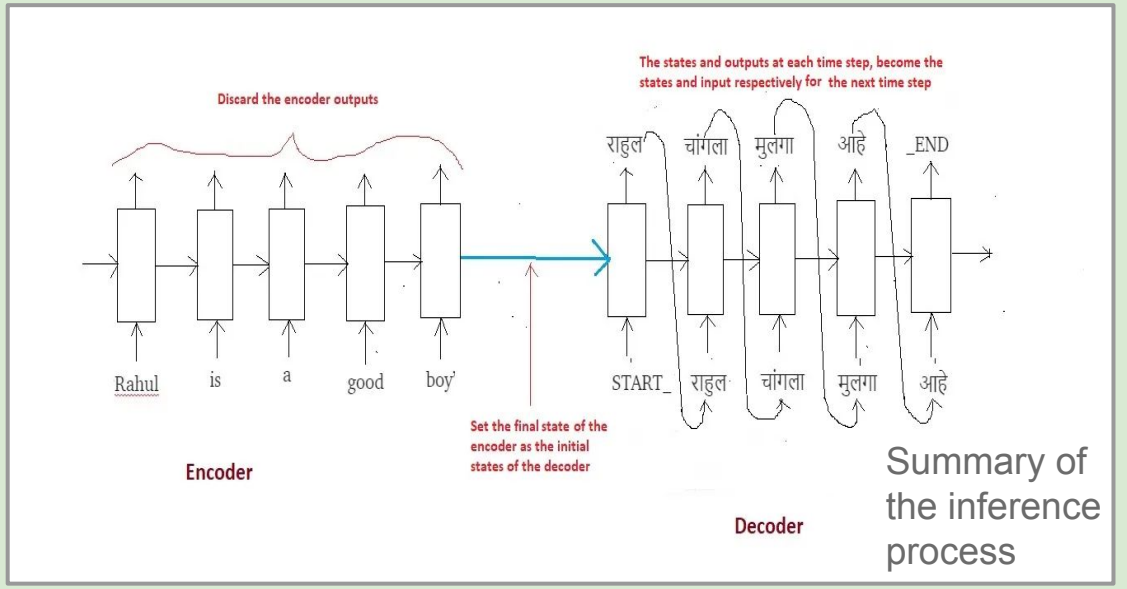




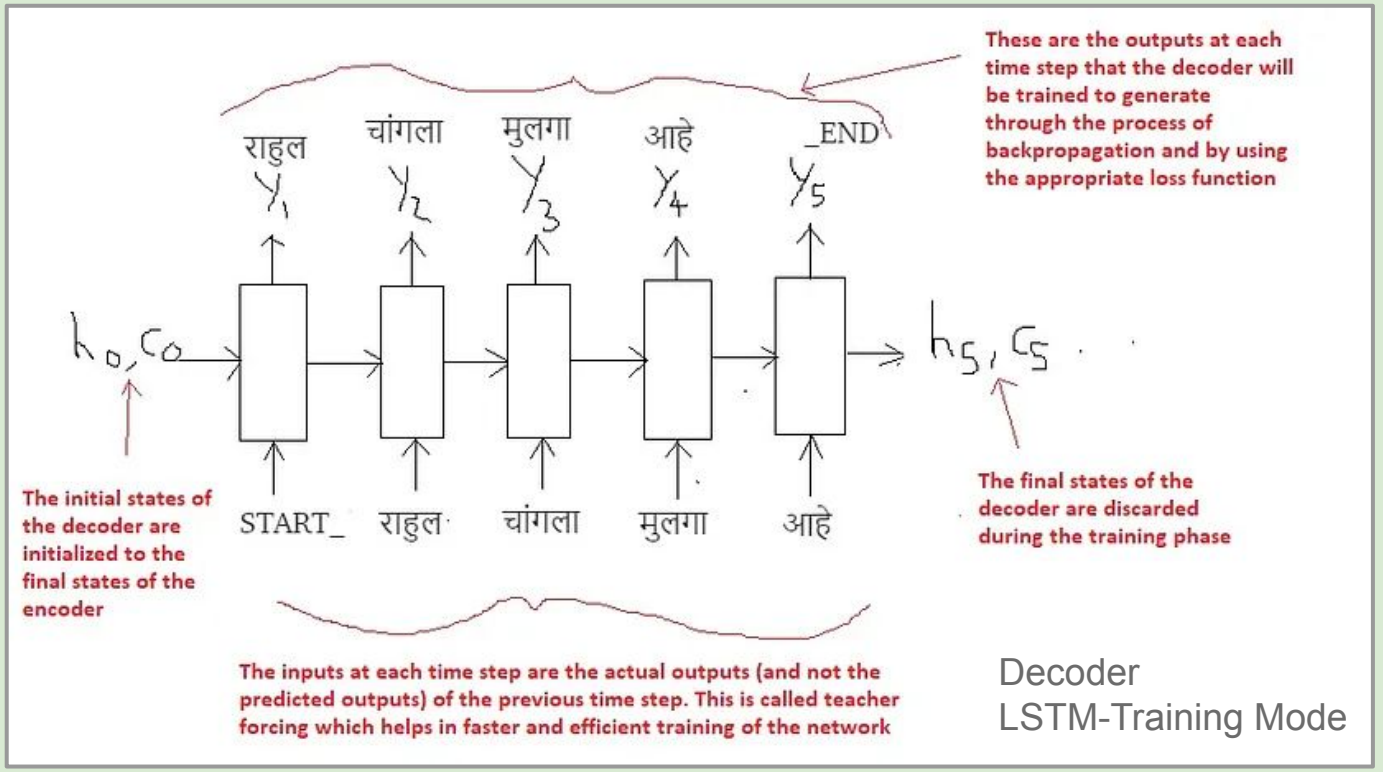
- The encoder processes the input sequence and produces a context vector, which captures the semantic information of the input.
- The decoder takes this **context vector** as input and generates the output sequence token by token.
- At each step of decoding, the decoder utilizes the context vector along with its internal states to predict the next token in the output sequence, conditioned on the input and previously generated tokens.

Encoder



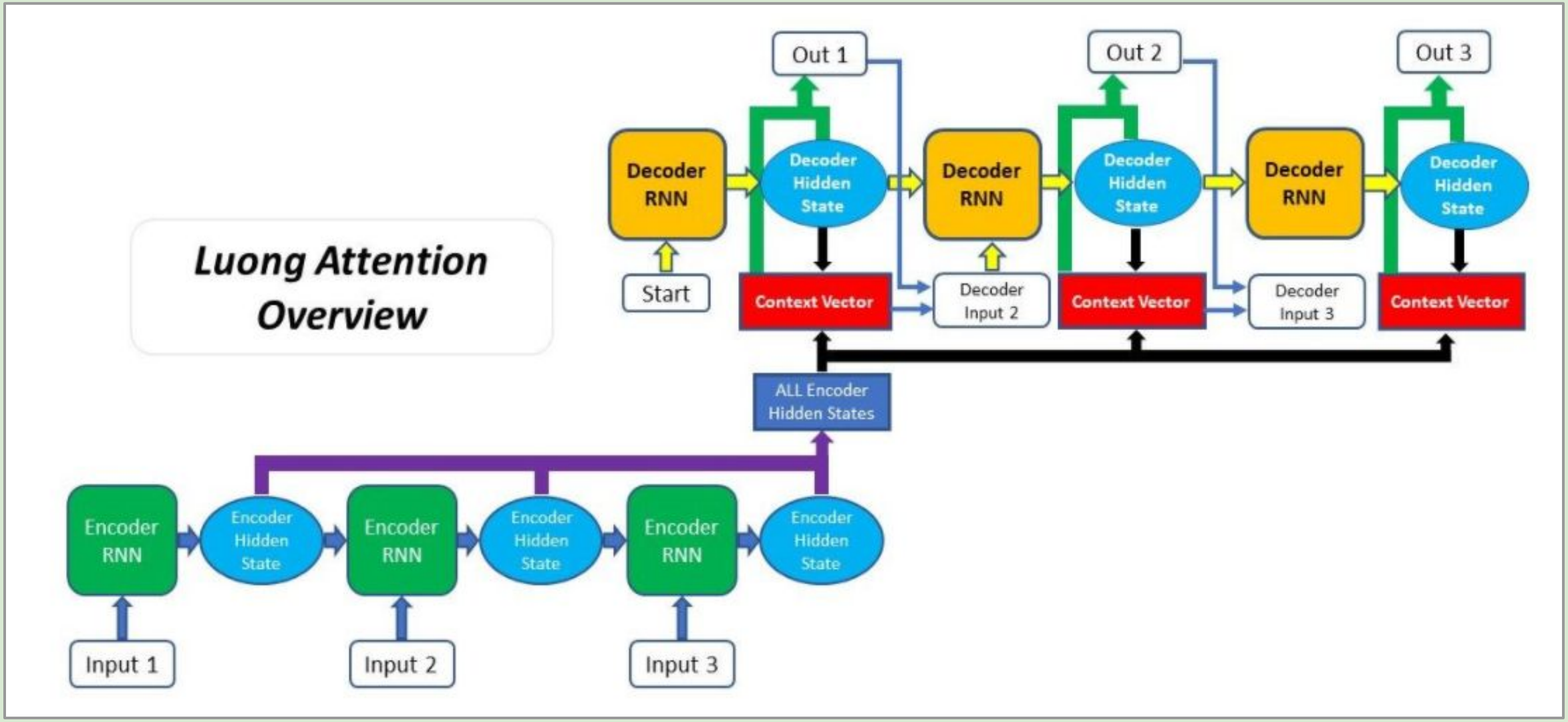


Summary of the inference process



Decoder
LSTM-Training Mode

Model Architecture (Encoder - Decoder with Luong attention)



TRAINING

The model is trained using the prepared dataset. We define the loss function as the Sparse Categorical Cross Entropy, suitable for sequence prediction tasks.

Training Loop: Iterate through batches of data, compute loss, and optimize the model parameters using backpropagation. Monitor accuracy to ensure the model is learning the translation task effectively.

Algorithm

1. Read data from file '/content/mar.txt'.
2. Split the data into lines and store it in the variable `uncleaned_data_list`.
3. Limit the size of `uncleaned_data_list` to 38695.
4. Extract English and Marathi words from `uncleaned_data_list` and store them in `english_word` and `marathi_word` respectively.
5. Create a DataFrame named `language_data` with columns 'English' and 'Marathi' and populate it with `english_word` and `marathi_word` respectively.
6. Save `language_data` to a CSV file named 'language_data.csv'.
7. Tokenize the English and Marathi sentences using the `Tokenizer` class and store the tokenizers in `eng_tokenizer` and `mar_tokenizer` respectively.
8. Obtain word index dictionaries for English and Marathi from the tokenizers and store them in `eng_word_index` and `mar_word_index` respectively.
9. Calculate the vocabulary sizes for English and Marathi and store them in `ENG_VOCAB_SIZE` and `MAR_VOCAB_SIZE` respectively.

10. Determine the maximum length of English and Marathi sentences and store them in `max_eng_len` and `max_mar_len` respectively.
11. Pad the English and Marathi sequences to have the same length using `pad_sequences` and store the padded sequences in `eng_padded` and `mar_padded` respectively.
12. Split the data into training and testing sets using `train_test_split` and store the sets in `X_train`, `X_test`, `y_train`, and `y_test`.
13. Define hyperparameters including `EPOCHS`, `BUFFER_SIZE`, `BATCH_SIZE`, `steps_per_epoch`, `embedding_dim`, `units`, and `hidden_dim`.
14. Create a TensorFlow Dataset from the training data.
15. Define the Luong Attention mechanism, Encoder, and Decoder models.
16. Create an Adam optimizer and define a checkpoint for model saving.
17. Train the model using the training dataset for the specified number of epochs, updating the model parameters with the optimizer.
18. Define a function `predict_seq2seq_att` to predict output sequences using the trained model.
19. Use the function `predict_seq2seq_att` to predict an output sequence for a given input sentence.

Program

Encoder-decoder model :
https://colab.research.google.com/drive/1-Dyd-o0G1MnlxekNlJR-SoRP-Sk6uW_B

Encoder-decoder model with Luong attention :
https://colab.research.google.com/drive/1SL35kOnZz94-KkJmGrMK9-OmVQOXl3zs#scrollTo=wMFZouh1WUjv

Output

Input Sentence: i want to drink water
Predicted Output Sequence: मला पाणी प्यायचं आहे

Input Sentence: i do not like my dress
Predicted Output Sequence: मला माझा ड्रेस आवडला नाही

Input Sentence: i want to sleep
Predicted Output Sequence: मला झोपायचं आहे

Input Sentence: i am very tired
Predicted Output Sequence: मी खूप थकलेय

Model Comparison

Encoder Decoder Model

```
Epoch 25 Batch 0 Loss 0.1236 Accuracy 0.9244
Epoch 25 Batch 100 Loss 0.1480 Accuracy 0.9231
Epoch 25 Batch 200 Loss 0.1400 Accuracy 0.9102
Epoch 25 Batch 300 Loss 0.1326 Accuracy 0.9099
Epoch 25 Batch 400 Loss 0.1730 Accuracy 0.9000
Epoch 25 Batch 500 Loss 0.2185 Accuracy 0.8647
Epoch 25 Loss 0.2119 Accuracy 0.8916
Time taken for 1 epoch 19.51 sec
```

Encoder Decoder Model with attention

```
Epoch 25 Batch 0 Loss 0.1583 Accuracy 0.8918
Epoch 25 Batch 100 Loss 0.1417 Accuracy 0.9136
Epoch 25 Batch 200 Loss 0.1297 Accuracy 0.9273
Epoch 25 Batch 300 Loss 0.1651 Accuracy 0.8988
Epoch 25 Batch 400 Loss 0.2150 Accuracy 0.8899
Epoch 25 Batch 500 Loss 0.1556 Accuracy 0.9069
Epoch 25 Loss 0.1808 Accuracy 0.8931
Time taken for 1 epoch 41.17 sec
```

1. Accuracy Difference: The model with attention slightly outperforms the one without, showing higher accuracy scores across batches and overall accuracy (0.8931 vs. 0.8916).

2. Training Time: However, training with attention takes longer per epoch (41.17 sec) compared to training without attention (19.51 sec) due to the additional computations for attention mechanisms.

3. Performance Benefit: Despite the longer training time, the attention mechanism offers a small but noticeable improvement in accuracy, making it a valuable addition for tasks where accuracy is critical.

4. Consideration: When choosing between the two, it's essential to consider the trade-off between improved performance and increased computational cost, based on the specific requirements of the task and available resources.

5. Overall Impact: The slight accuracy gain from using attention suggests its effectiveness in enhancing model performance, especially in tasks demanding precise sequence prediction.

Applications

Language Learning: It can aid language learners by providing translations of English text into Marathi, allowing them to understand English content better and improve their Marathi comprehension skills.

Cross-Lingual Communication: It facilitates communication between English speakers and Marathi speakers who may not be proficient in each other's languages, enabling them to exchange information and ideas effectively.

Document Translation: It can be used to translate various types of documents, such as articles, reports, and presentations, from English to Marathi, catering to the needs of Marathi-speaking users.

Website Translation: Website owners can use the converter to translate their websites from English to Marathi, providing a localized user experience for Marathi-speaking visitors and potentially expanding their audience base.

Multilingual Information Retrieval: It enables users to retrieve information in Marathi from English-language sources, enhancing accessibility to information for Marathi-speaking individuals.

Conclusion

This model presents a machine translation system for converting English sentences into Marathi using a Seq2Seq model with LSTM units. We also used attention mechanism to get more accurate results of language translation with accuracy of 0.9. Through data preprocessing, model training, and evaluation, we have demonstrated the system's ability to facilitate cross-lingual communication. Future work includes enhancing translation quality and exploring multilingual capabilities.

References

[1] Ilya Sutskever, Oriol Vinyals, Quoc V. Le “Sequence to Sequence Learning with Neural Networks”- 2014 · Cited by 25576

[2] Y. Zhi, T. Kaynak, and W. Zhang, “Modified maximum power point tracking technique for wind energy application,” *IEEE Trans. Power Electron.*, vol. 27, no. 7, pp. 3023-3027, Jul 2012.

[3] F. Zhang, “A new high power factor AC-AC inverter,” *IEEE Power Electron. Lett.*, vol. 1, no. 5, pp. 10–13, Mar. 2000.