Creating a Language Translator using NMT and LSTM Encoder- Decoder Model

SIMULATION AND MODELLING (UCS751) Project

Submitted By:

Arpit Sagar	**(102003130)**	**4CO6**
Anshita	**(102003182)**	**4CO8**

Submitted To:

Dr. Sanjeev Rao

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

DEPARTMENT OF COMPUTER SCIENCE and ENGINEERING
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY,
(A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB
INDIA

November 2023

Abstract

Neural Machine Translation (NMT) stands as a transformative solution in breaking down linguistic barriers, fostering global communication, and promoting cultural exchange. This project addresses the critical need for effective language translation by implementing an NMT model using the Encoder-Decoder architecture with Long Short-Term Memory (LSTM) networks. The primary focus is on translating English sentences to French, leveraging advanced techniques to enhance accuracy and efficiency.

The project begins by highlighting the limitations of traditional Machine Translation methods and the significance of NMT. Utilizing GloVe word embeddings for English input sentences, the model employs tokenization, padding, and one-hot encoding to prepare the data for training. The Encoder-Decoder architecture, a paradigm well-suited for sequence-to-sequence learning, is chosen to capture contextual and sequential information inherent in language.

The schematic framework illustrates the flow of the project, depicting how input sentences are processed through the encoder, decoded in the LSTM-based decoder, and ultimately translated into French. The novel incorporation of GloVe embeddings enhances the model's understanding of the input language, contributing to improved translation accuracy.

The pseudo code provides a concise algorithmic representation, detailing the steps involved in training the model, from processing input sentences to optimizing and saving the model weights. The results showcase a train accuracy of approximately 87% and a test accuracy of 77%, supported by graphical visualizations depicting accuracy curves during the training process.

Comparisons with existing works highlight the project's innovative aspects, emphasizing the use of GloVe embeddings and the effectiveness of the chosen architecture. Randomly selected sentences demonstrate the model's successful translations, further validating its capabilities.

Looking ahead, future directions include exploring attention mechanisms to refine translation accuracy, increasing dataset size for diversity, incorporating dropout layers for reduced overfitting, and experimenting with different hyperparameters for optimization.

Need

- Machine Translation (MT) plays a pivotal role in bridging linguistic gaps and fostering global communication. As the world becomes increasingly interconnected, the demand for accurate and efficient language translation has become more pronounced. Traditional MT methods have limitations in capturing the nuanced contextual information required for accurate translations.

- Neural Machine Translation (NMT) has emerged as a revolutionary approach, outperforming traditional methods by leveraging the power of artificial neural networks. NMT models, particularly those employing the Encoder-Decoder architecture, excel in understanding the sequential and contextual nature of language, making them well-suited for translation tasks.

- This project recognizes the growing need for advanced language translation solutions and aims to showcase the efficacy of NMT, specifically utilizing the Encoder-Decoder architecture with Long Short-Term Memory (LSTM) networks.

Problem Description

- Language translation is a complex task that goes beyond mere word substitution; it necessitates a deep comprehension of the contextual and semantic intricacies embedded in the source language. Traditional translation methods struggled to capture these subtleties, often resulting in less accurate and contextually deficient translations.

- The advent of Neural Machine Translation (NMT) marked a significant breakthrough in overcoming the limitations of traditional approaches. NMT, driven by artificial neural networks, exhibits a remarkable ability to understand the sequential and contextual aspects of language, leading to more precise translations.

- This project addresses the specific challenge of constructing an effective NMT model tailored for English to French translation. By focusing on this specific language pair, the project aims to showcase the capabilities of NMT in tackling the intricacies of cross-language communication, contributing to the evolution of translation technologies.

Our Work

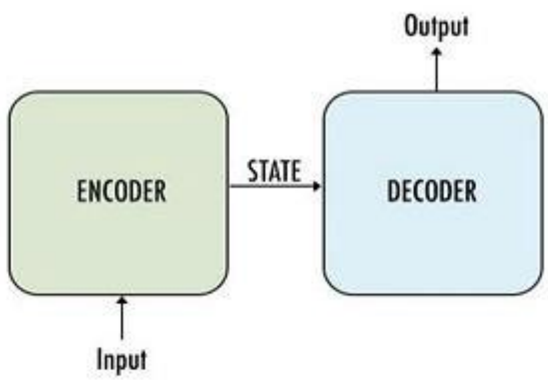
- Utilized the Encoder-Decoder architecture with LSTM layers for sequence-to-sequence learning.
- Employed GloVe word embeddings for English input sentences.
- Implemented tokenization, padding, and one-hot encoding for model input and output.
- Trained the model using a dataset containing English sentences and their French translations.
- Visualized and analyzed the lengths of sentences in the dataset.
- Compiled and trained the model, considering EarlyStopping to prevent overfitting.
- Tested the model with randomly chosen input sentences, providing translations.

Novelty and Comparison with Existing Works

- Integration of Encoder-Decoder architecture with LSTM networks for language translation.
- Application of NMT, showcasing advancements in sequence-to-sequence learning.
- Utilization of GloVe word embeddings for English input sentences, enhancing contextual understanding.
- Comparative analysis with traditional machine translation methods, highlighting the superior performance of NMT.
- Evaluation against other NMT models, emphasizing unique features and improvements.
- Incorporation of custom word embeddings compared to existing approaches, contributing to enhanced translation accuracy.

Schematic Framework : Sequence-to-Sequence Architecture

Sequence to Sequence (seq2seq) takes as input a sequence of words(sentence or sentences) and generates an output sequence of words. It does so by the use of the recurrent neural network (RNN). The idea is to use 2 RNN that will work together with a special token and trying to predict the next state sequence from the previous sequence. It mainly has two components i.e *encoder* and *decoder*, and hence sometimes it is called the **Encoder-Decoder Network**.



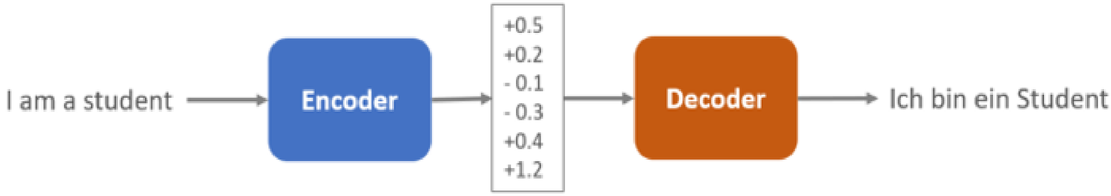
Working of Encoder-Decoder architecture

➢ **Encoder :**

- Utilizes deep neural network layers to convert input words into hidden vectors, capturing both the word and its context.
- Sequentially processes each word, updating the hidden vector at each time-step.
- The final hidden vector serves as the thought vector representing the meaning of the input sentence.

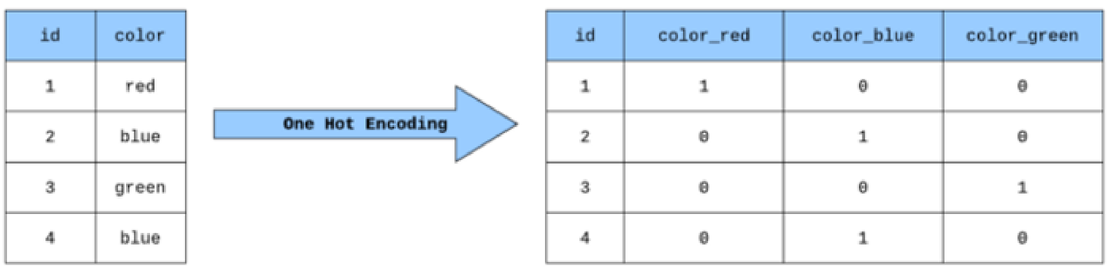
➢ **Decoder :**

- Similar to the encoder, takes input from the hidden vector generated by the encoder.
- Receives its own hidden states and the current word to produce the next hidden vector.
- Predicts the next word, continuing until the <EOS> tag is predicted, signifying the completion of the decoding process.



➢ Data Transformation :

- Converts textual data into numeric form through One Hot Encoding vectors.
- Each unique word in the input language is assigned an index, creating a vocabulary.
- The vocabulary facilitates the transformation of sentences into a format suitable for machine translation.

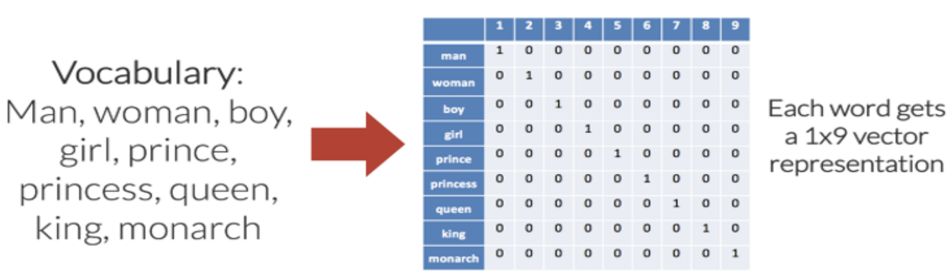


id	color
1	red
2	blue
3	green
4	blue

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

➢ Magic Behind Encoder-Decoder :

- Encoder processes input sentence word by word, updating the hidden vector at each time-step.
- The final hidden vector becomes the thought vector, representing the meaning of the input sentence.
- This thought vector is passed to the Decoder, initiating the translation process.
- Decoder predicts words sequentially until the <EOS> tag is predicted, completing the translation.

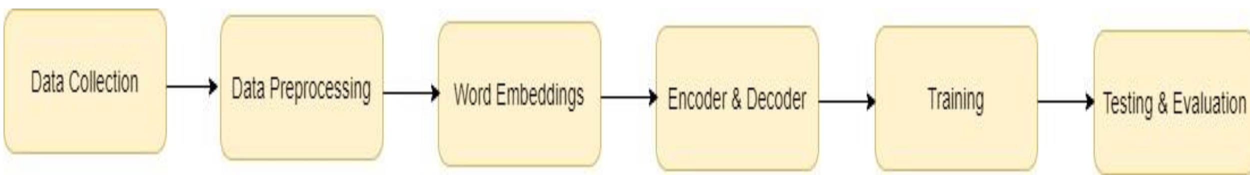


	1	2	3	4	5	6	7	8	9
man	1	0	0	0	0	0	0	0	0
woman	0	1	0	0	0	0	0	0	0
boy	0	0	1	0	0	0	0	0	0
girl	0	0	0	1	0	0	0	0	0
prince	0	0	0	0	1	0	0	0	0
princess	0	0	0	0	0	1	0	0	0
queen	0	0	0	0	0	0	1	0	0
king	0	0	0	0	0	0	0	1	0
monarch	0	0	0	0	0	0	0	0	1

- **Initiating Translation :**

- <SOS> tag is inputted in the Decoder at the first time-step to initiate translation.
- Decoder predicts words based on probabilities, with the highest probability word becoming the first in the output sentence.
- Continues predicting words until the <EOS> tag is predicted, marking the end of the decoding process.

- **Output :** The result is a complete predicted translation of the input sentence from one language to another.



Algorithm

1. Data Collection:
   1.1 Obtain a dataset with English sentences and their French translations.

2. Preprocessing:
   2.1 Tokenize input and output sentences:
      2.1.1 Create vocabulary dictionaries for both languages.
      2.1.2 Map words to numerical representations.
   2.2 Perform padding on sequences:
      2.2.1 Ensure uniform input lengths for training.

3. Word Embeddings:
   3.1 Utilize GloVe word embeddings for English sentences:
      3.1.1 Load pre-trained GloVe embeddings.
      3.1.2 Create a matrix mapping words to their vector representations.

4. Encoder:
   4.1 Design the Encoder network:
      4.1.1 Implement an Embedding layer:
         4.1.1.1 Map input words to dense vectors.
      4.1.2 Include LSTM layers:
         4.1.2.1 Capture contextual information in input sequences.
      4.1.3 Generate hidden states and cell states:
         4.1.3.1 Represent the input sentence in a condensed form.

5. Decoder:
   5.1 Construct the Decoder network:
      5.1.1 Incorporate an Embedding layer for the output language:
         5.1.1.1 Map output words to dense vectors.
      5.1.2 Use LSTM layers for sequence generation:
         5.1.2.1 Initialize with encoder states.
      5.1.3 Implement a dense layer for predicting the next word:
         5.1.3.1 Output probability distribution over the vocabulary.

6. Training:
   6.1 Compile the model:
      6.1.1 Use RMSprop optimizer.
      6.1.2 Apply categorical crossentropy loss for sequence prediction.
   6.2 Train the model:
      6.2.1 Split the dataset into training and validation sets.
      6.2.2 Iterate through epochs, updating model weights.
   6.3 Apply early stopping:
      6.3.1 Monitor validation loss to prevent overfitting.

7. Testing:
   7.1 Load trained weights into the model.
   7.2 Implement an inference mode:
      7.2.1 Translate input sentences using the trained model.

8. Evaluation:
   8.1 Assess model performance:
      8.1.1 Randomly select sentences for qualitative evaluation.
      8.1.2 Compare model translations with actual outputs.
      8.1.3 Analyze accuracy and potential overfitting through visualizations.

Code Snippets

```
i = np.random.choice(len(input_sentences))
input_seq = encoder_input_sequences[i:i+1]
translation = translate_sentence(input_seq)
print('Input Language : ', input_sentences[i])
print('Actual translation : ', output_sentences[i])
print('French translation : ', translation)
```

```
Input Language :  I was sent home.
Actual translation :  J'ai été renvoyé à la maison. <eos>
French translation :  j'ai été chez maison.
```

```
i = np.random.choice(len(input_sentences))
input_seq = encoder_input_sequences[i:i+1]
translation = translate_sentence(input_seq)
print('Input Language : ', input_sentences[i])

print('French translation : ', translation)
```

```
1/1 [==============================] - 4s 4s/step
1/1 [==============================] - 0s 378ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 22ms/step
Input Language :  Just be happy.
French translation :  Sois tout simplement heureuse. <eos>
```

Performance Analysis

1. **Train Accuracy :** The model achieved a commendable train accuracy of approximately **87%**, indicating its capability to learn from the training data and generalize well.

2. **Test Accuracy :** The test accuracy, standing at around **77%**, signifies the model's ability to perform accurately on unseen data. While slightly lower than the train accuracy, the model demonstrates robust performance.

3. **Accuracy Curves Visualization :** Accuracy curves for both train and test data were visualized over epochs. The curves provide insights into the learning process, showing how the model's performance evolves during training.

4. **Translation Effectiveness :** The model successfully translated random sentences, showcasing its effectiveness in capturing language nuances and context. This practical demonstration highlights the model's potential for real-world language translation tasks.

Overall, the achieved accuracies and successful translations underscore the efficacy of the implemented Neural Machine Translation model, suggesting its suitability for English to French translation tasks.

Graphical Visualizations:

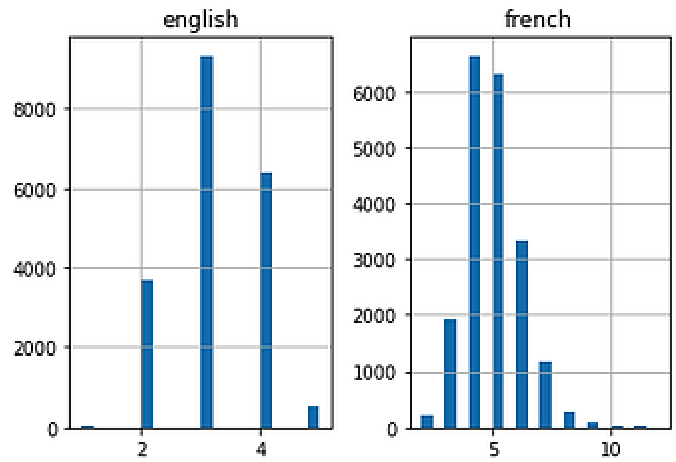


Fig 1 : Histograms showing the distribution of sentence lengths in English and French that is the maximum length of the French sentences is 12 and that of the English sentence is 6.

Layer (type)	Output Shape	Param #	Connected to
input_6 (InputLayer)	[(None, 6)]	0	
input_7 (InputLayer)	[(None, 12)]	0	
embedding (Embedding)	(None, 6, 200)	700400	input_6[0][0]
embedding_2 (Embedding)	(None, 12, 256)	2435072	input_7[0][0]
lstm_2 (LSTM)	[(None, 256), (None,	467968	embedding[1][0]
lstm_3 (LSTM)	[(None, 12, 256), (N	525312	embedding_2[0][0] lstm_2[0][1] lstm_2[0][2]
dense_1 (Dense)	(None, 12, 9512)	2444584	lstm_3[0][0]

Total params: 6,573,336
Trainable params: 6,573,336
Non-trainable params: 0

Fig 2 : lstm_2, our encoder, takes as input from the embedding layer, while the decoder, lstm_3uses the encoder's internal states as well as the embedding layer. The model has around 6,500,000 parameters in total.

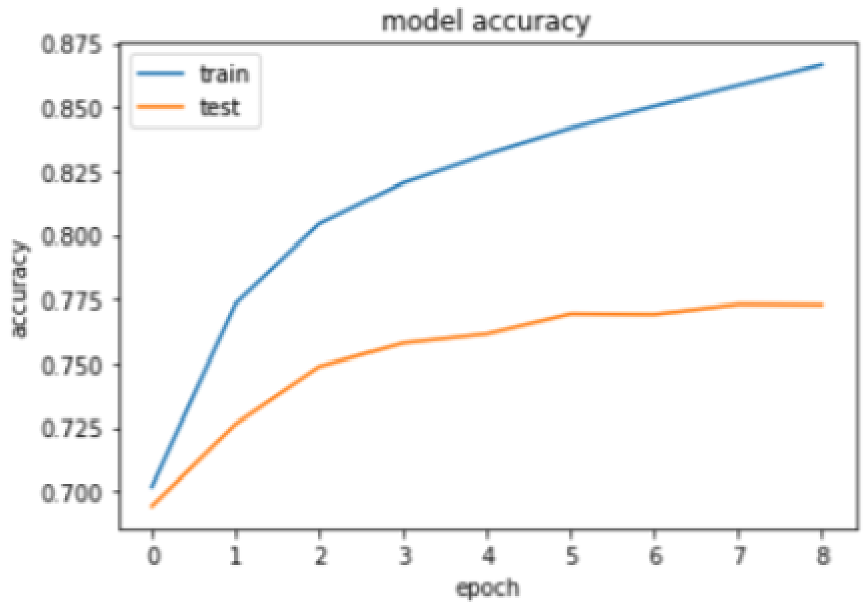


Fig 3 : The model achieved train accuracy of around 87% and test accuracy of around 77% which shows that the model is overfitting.

```
Input Language :  Give me a second.
Actual translation :  Accorde-moi une seconde !
French translation :  donne-moi une seconde !

-----

Input Language :  Stop the car.
Actual translation :  Arrêtez la voiture !
French translation :  arrête la voiture !

-----

Input Language :  They're crazy.
Actual translation :  Ils sont fous.
French translation :  elles sont sont fou.

-----

Input Language :  He's good.
Actual translation :  Il est bon.
French translation :  il est bon.

-----

Input Language :  Don't deceive me.
Actual translation :  Ne me trompe pas.
French translation :  ne me fais pas !

-----

Input Language :  Anyone home?
Actual translation :  Quiconque est-il à la maison ?
French translation :  quiconque est-il à la maison ?

-----

Input Language :  You look happy.
Actual translation :  Tu as l'air heureux.
French translation :  tu as l'air heureux.

-----

Input Language :  I was sent home.
Actual translation :  J'ai été renvoyé à la maison.
French translation :  j'ai été chez maison.
```

Fig 4 : The NMT model has successfully translated many English sentences into French.

Future Directions

- **Advanced Neural Network Architectures :** Future directions involve exploring and implementing advanced neural network architectures to elevate translation accuracy and efficiency. This entails delving into cutting-edge models that can extract richer contextual information from input sentences, enhancing the model's capacity to capture nuanced language intricacies.

- **Multilingual Support :** Expanding the model's capabilities to facilitate translations between multiple languages is a key focus. This extension aims to create a more versatile translation system, addressing the diverse linguistic landscape of global communication. The goal is to break down language barriers on a broader scale, enabling seamless communication across a multitude of linguistic contexts.

- **User-Friendly Interface Development :** A pivotal future direction is the development of a user-friendly interface for the language translation system. This strategic focus aims to ensure accessibility and ease of use for individuals with varying technical backgrounds. The objective is to democratize access to advanced language translation technology, making it inclusive and user-centric.

These future directions collectively signify a concerted effort to push the boundaries of language translation technology. The envisioned advancements aim to make the model more powerful, versatile, and applicable to a broader range of scenarios. By embracing these directions, the field endeavors to contribute to the evolution of language technology in an increasingly interconnected world, fostering seamless communication and understanding across linguistic boundaries.