Neil de la Fuente 1630223 Daniel Vidal 1634599 Joan Samper 1631430

LAB6 NLP

RNNs for TRANSLATION

Introduction

In our recent lab exercise, we embarked on the task of machine translation using deep learning techniques, specifically using PyTorch Recurrent Neural Network models. Our focus was to develop a system capable of translating sentences from English to Italian, leveraging the power of neural networks. This report outlines the process we followed, from data preparation to training and evaluating our model, all aimed at achieving accurate and efficient language translation.

Data Preparation

We started by preparing English and Italian sentence pairs. We have a dataset of 345244 translations in different languages, but we are going to use the translations between English and Italian, which were then subjected to several preprocessing steps to make them suitable for training. This included converting Unicode characters to ASCII, normalizing strings by converting them to lowercase and removing non-letter characters, and finally, organizing our data into pairs of sentences in the two languages.

An essential aspect of our data preparation involved creating a language class (Lang) to track vocabulary and convert sentences into numerical representations. This class facilitated the construction of dictionaries mapping words to indices and vice versa, crucial for our neural network to process text.

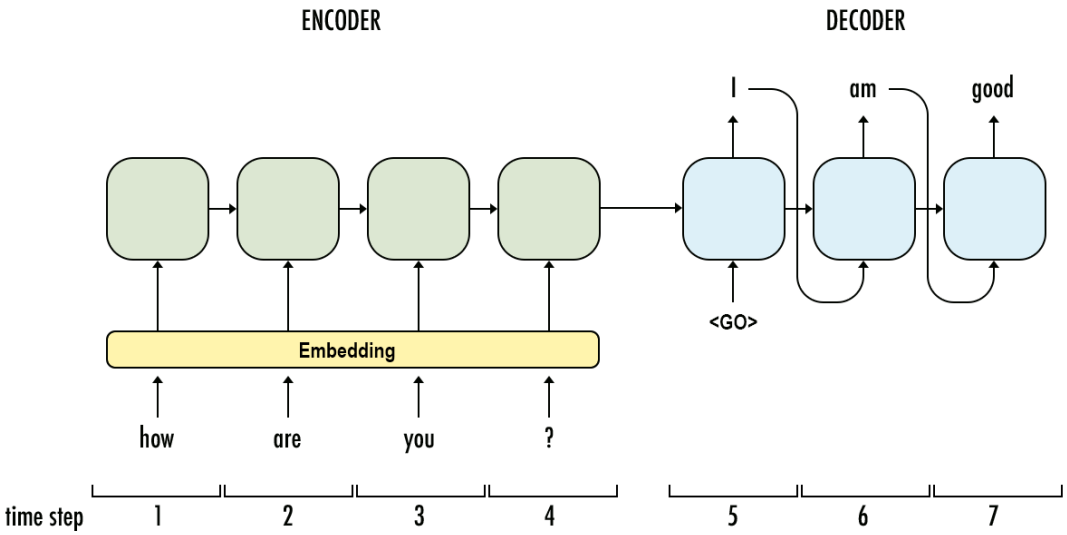
Counted words:

itaian words: 26170
english words: 13069

Translation Example : 'questo libro e piccolo come quello' → 'this book is as small as that one'

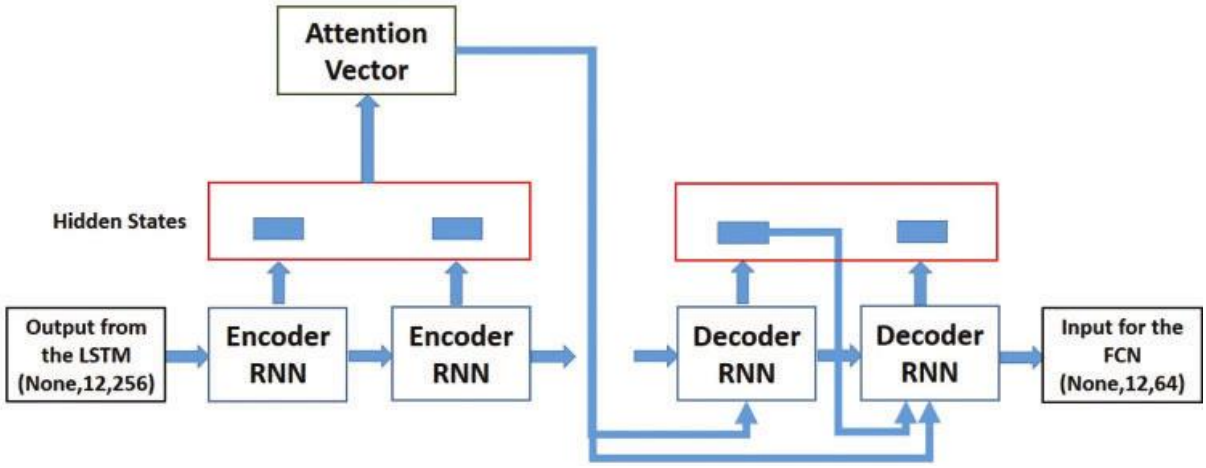
Building the Model

Our translation model consists of two main components: an encoder and a decoder, both implemented using Recurrent Neural Networks (RNNs) with Gated Recurrent Units (GRUs). The encoder reads the input sentence and compresses its information into a context vector, which the decoder then uses to generate the translation.



Sequence to sequence model architecture

To enhance our model's ability to focus on relevant parts of the input sentence during translation, we incorporated an attention mechanism. This technique allows the decoder to weigh different parts of the input sentence differently at each step of the translation, leading to more accurate and contextually relevant outputs.

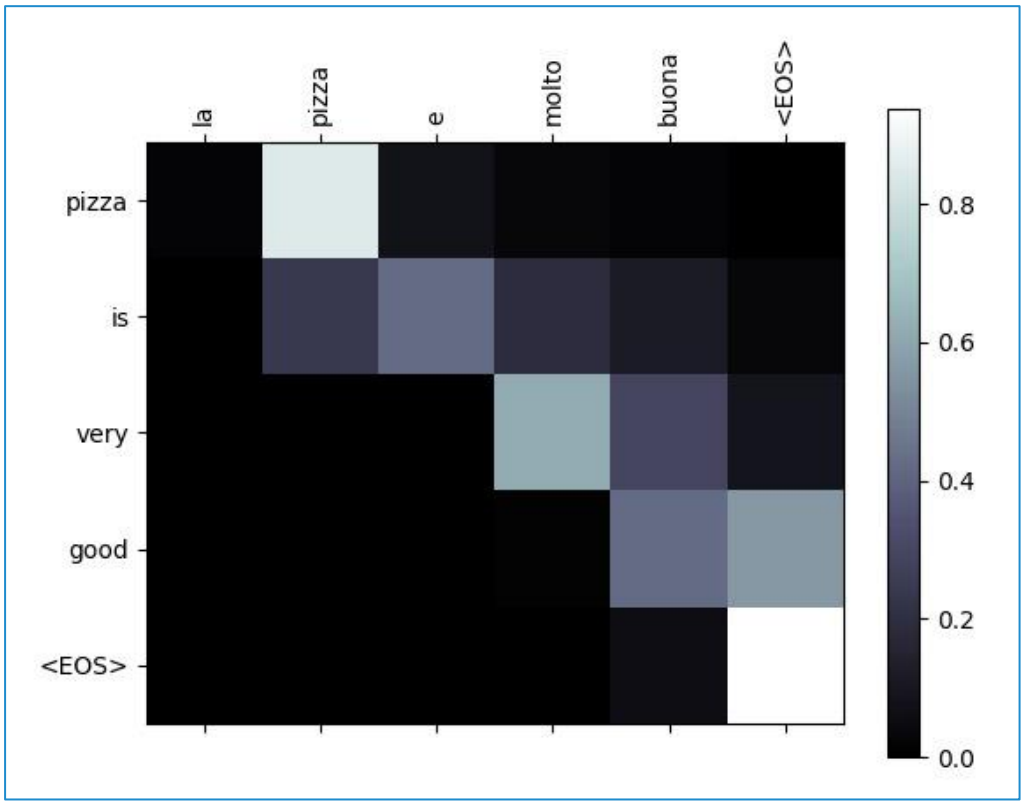


Bahdanau Attention Mechanism

In the encoder all the tokens of a sentence will be given to an embedding layer of size 256 to match every token to a trainable embedding that will represent the word. The embedding created by the embedding layer is then processed by a GRU layer of size 512, and the embeddings generated are directly passed to the decoder. To generate the next token teacher forcing is applied, so the last word will be the real word from the ground truth, and not the one generated by the model. This technique reduces the stochasticity of the model and makes the training faster.

The decoder will generate as many tokens as the maximum length of the strings in the dataset, and for every token to generate it receives the embedding generated by the encoder and a badhanau attention is applied to the embedding generated by the encoder, in order to attend to all the tokens of the sentence, and not only the last one. This enhances the performance and overcomes the problem of forgetting that RNNs have. After this attention layer, the resulting embedding is passed through a GRU layer at the decoder with hidden size of 128, generating another embedding.

This generated embedding is then given to a linear layer to generate a tensor of probabilities for every word in the vocabulary of the output language. The predicted word will be the word with maximum probability. The loss is computed by comparing the one hot tensor of targets with the probability tensor of outputs by using cross-entropy loss. Then this loss is backpropagated through the model to perform gradient descent.



This is an attention map where we can see the original sentence (italian) and its translation (english) with the attention scores between all tokens. Here we can also see that the the attention scores make sense since the works that have the same meaning have very high values and the ones that have no relation have a value of 0.

Training the Model

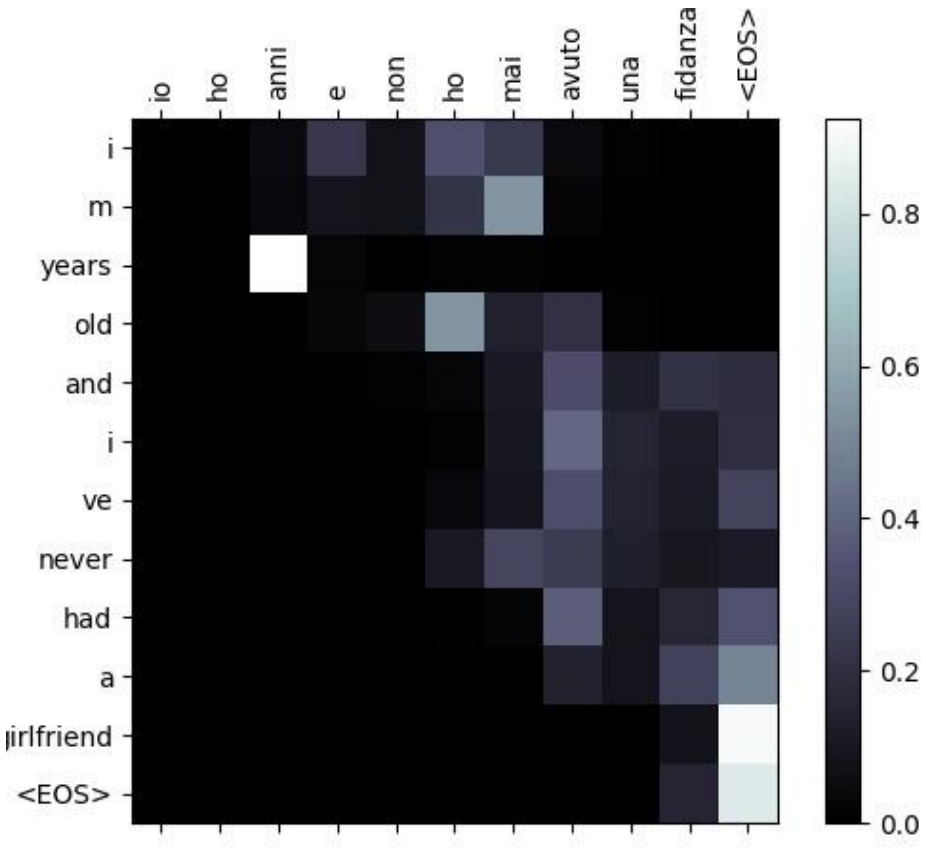
Training involved feeding our model with pairs of English and Italian sentences, allowing it to learn the intricate patterns of translation between these languages. We used a technique called teacher forcing to speed up the training process, where the correct output token at each step is provided to the decoder during the next step, rather than the decoder's own prediction.

To optimize our model, we utilized the Adam optimizer and calculated loss using the Negative Log-Likelihood Loss (NLLLoss), which measures the performance of our model across all the sentence pairs in our dataset.

Evaluating the Model

After training, we evaluated our model's performance by translating new sentences from English to Italian. This qualitative evaluation involved randomly selecting sentences from our dataset, translating them using our model, and comparing the generated translations to the actual Italian sentences. We also tried with new sentences that are not in the database.

Examples of qualitative evaluation:



Example 1.

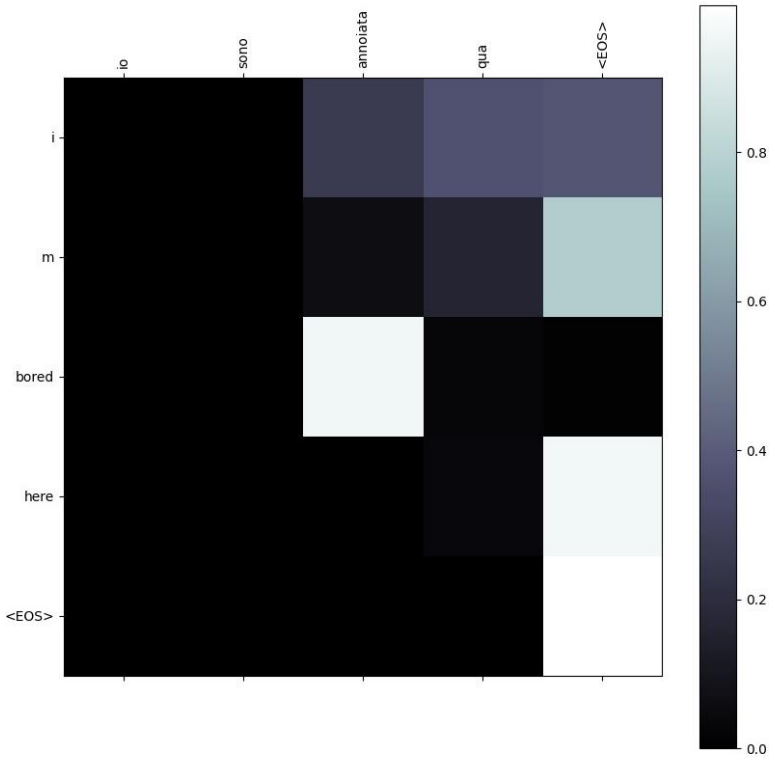


Figure 2.

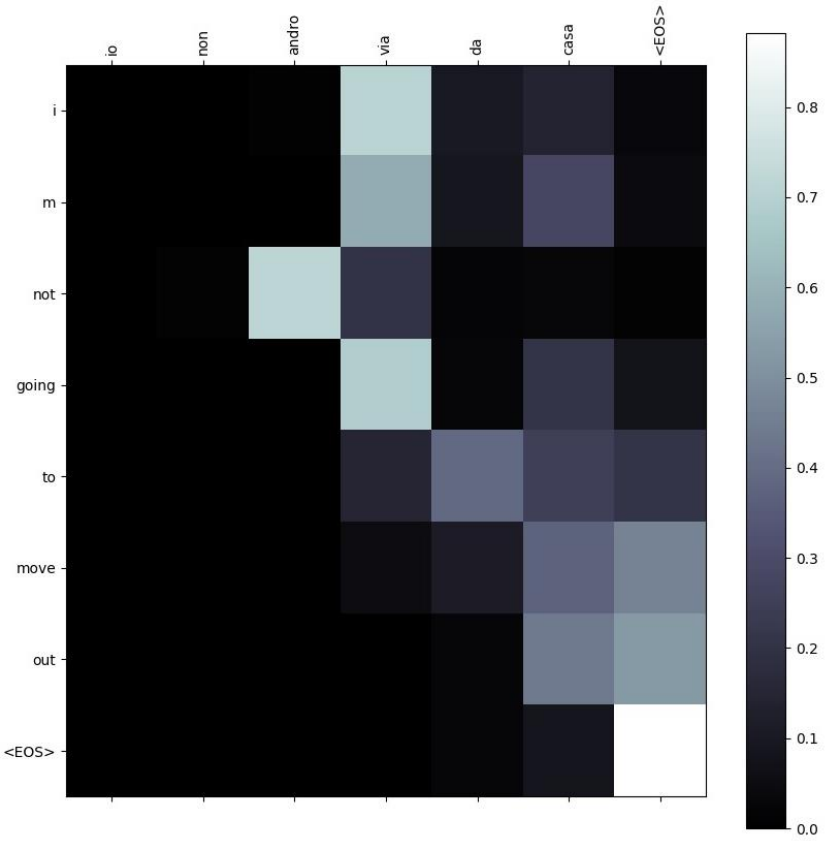


Figure 3.

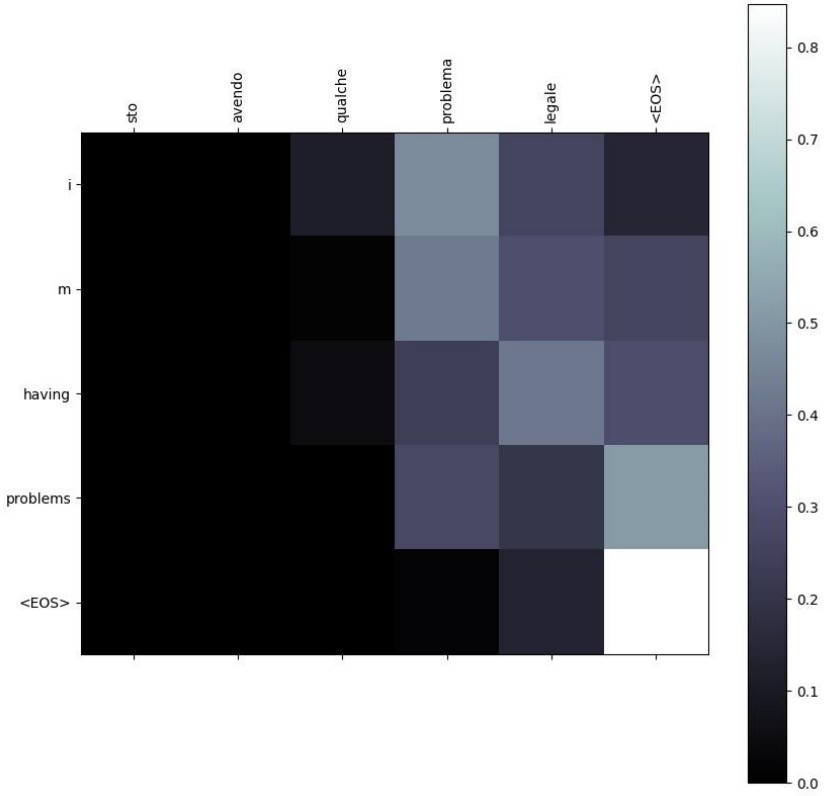


Figure 4.

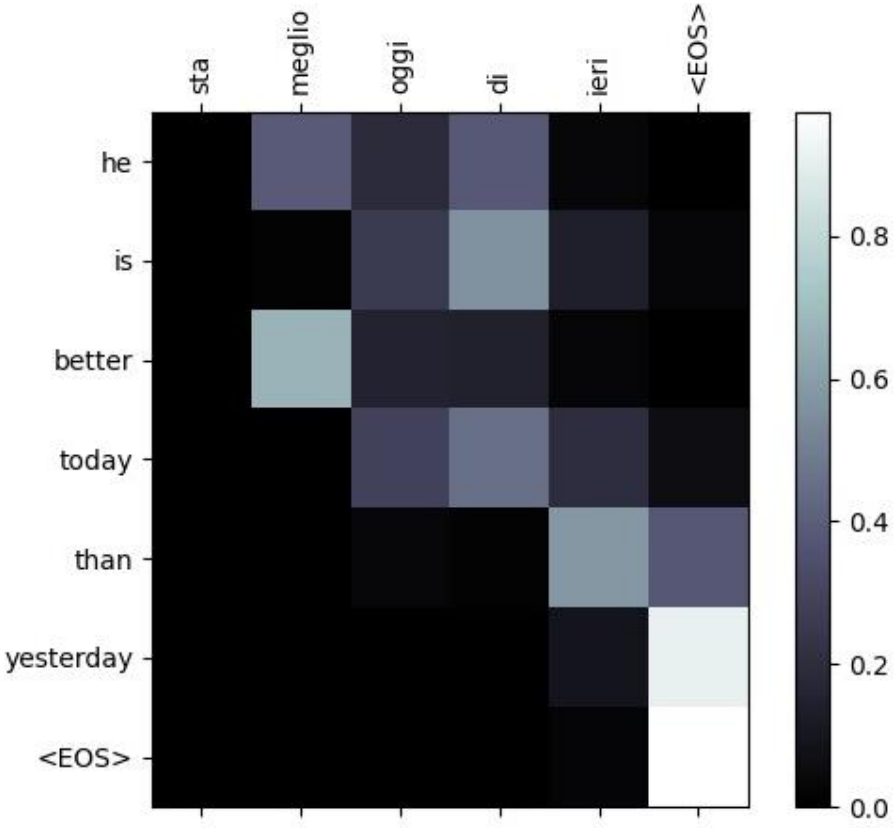


Figure 5.

In all previous examples we can see that the model is able to provide accurate translations for the given italian sentences although the attention scores do not seem to make a lot of sense for many pair of words but at least most of the important words in the sentences have good attention score values.

For a quantitative evaluation, we employed the BLEU score, a metric for evaluating the quality of text which has been machine-translated from one natural language to another. This allowed us to measure the accuracy of our translations systematically.

Conclusion

This exercise provided us with invaluable insights into the challenges and complexities of machine translation. Through iterative processes of data preparation, model building, and training, we developed a system capable of translating between English and Italian with promising results. The incorporation of an attention mechanism significantly enhanced the model's ability to produce contextually relevant translations, demonstrating the power of neural networks in overcoming the nuances of language translation.