COPS Summer of Code 2025

Intelligence Guild

NLP Track: Sequence Modeling

18 – 21 June 2025

Seq2Seq Architectures for Machine Translation

Suyash Ranjan
24154022

1. Introduction

This report outlines the approach taken to develop and compare different Sequence-to-Sequence (Seq2Seq) models for English to Spanish machine translation. The focus was on implementing:

- A Vanilla Seq2Seq model (without attention)
- A Seq2Seq model with Bahdanau Attention
- A Transformer-based MarianMT model from Hugging Face

I aimed to understand the performance trade-offs between these architectures, using dynamic padding, real translation datasets, and BLEU scores for evaluation.

2. Dataset Description

I used a cleaned parallel English-Spanish translation dataset sourced from Kaggle and stored in Google Drive. The dataset was processed into:

- **train.csv**, **val.csv**
- Columns: **english**, **spanish**

Dynamic padding was applied using TensorFlow's 'tf.data' pipeline for efficient batching.

3. Pre-processing

The pre-processing included:

- Lowercasing and stripping white spaces
- Removing unnecessary punctuation
- Tokenizing and indexing sentences using 'Tokenizer'
- Dynamic padding using 'padded_batch' in 'tf.data'

4. Word Embeddings

I used trainable embeddings (without pre-trained weights) for both encoder and decoder vocabularies. The embedding dimension was chosen based on model complexity and GPU memory.

5. Model Architectures

5.1 Vanilla Seq2Seq

- Embedding layer for encoder and decoder inputs
- Encoder LSTM to output final hidden and cell states
- Decoder LSTM initialized with encoder states
- Dense output layer with softmax activation

5.2 Seq2Seq with Bahdanau Attention

- Similar base architecture as vanilla
- Custom Bahdanau Attention layer added before decoding(I had trouble using Keras AdditiveAttention and implementing it so i switched to a custom attention)
- Context vector computed dynamically per decoding timestep

5.3 MarianMT Transformer (Hugging Face)

- Loaded pre-trained model: ‘Helsinki-NLP/opus-mt-en-es‘
- Used tokenizer and pipeline from Hugging Face
- No fine-tuning was performed (inference only)

6. Evaluation Metrics

- **BLEU Score** on validation and test sets
- Manual quality checks via first 10 translation outputs

7. Convergence Plots and Metric Visualizations

To compare the performance of Vanilla Seq2Seq and Attention-based Seq2Seq models, the training and validation metrics were tracked across epochs. The plots below show the convergence behavior in terms of loss and BLEU scores.

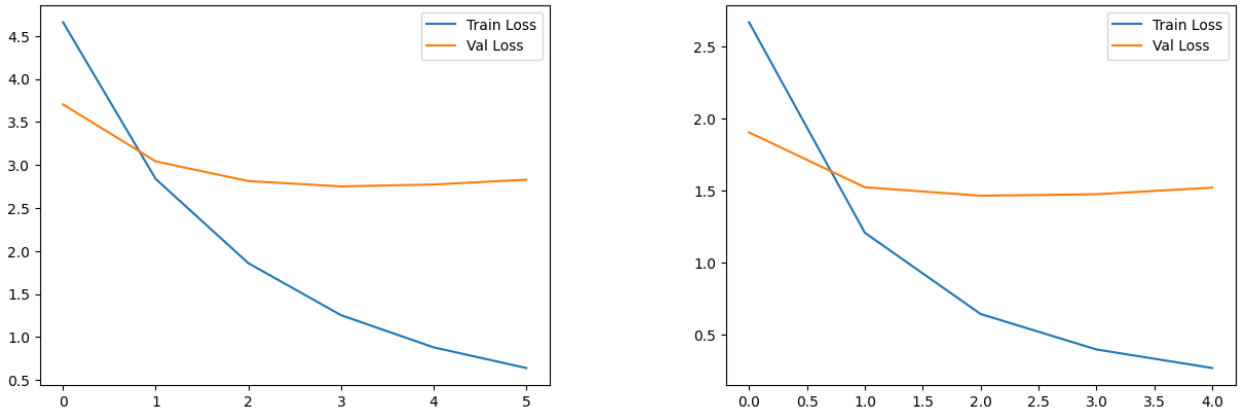

Figure 1: Convergence Plots: Left – Vanilla Seq2Seq; Right – Seq2Seq with Attention

BLEU Score Comparison Across Models

Model	BLEU Score (Validation Set)
Vanilla Seq2Seq	26.86
Seq2Seq with Attention	56.23
MarianMT (No-tuning)	56.74

Table 1: BLEU Score Comparison for All three Seq2Seq Architectures

Note: The BLEU scores for the Vanilla and Attention-based Seq2Seq models were computed on a subset of 100 samples from the validation set due to compute limitations. While the absolute values may change slightly with full evaluation, the relative ranking among models is expected to remain the same — with the Transformer model performing best, followed closely by the Attention-based model, and the Vanilla Seq2Seq scoring lowest.

8. Attention Heatmaps & Lens Analysis

To better understand the inner workings of the attention mechanism, I visualized the attention weights for a few translated examples. These heatmaps demonstrate how the decoder focuses on different parts of the input sequence while generating each output token.

Example 1:
`Input:` turn off the light
`Prediction:` las luces la luz

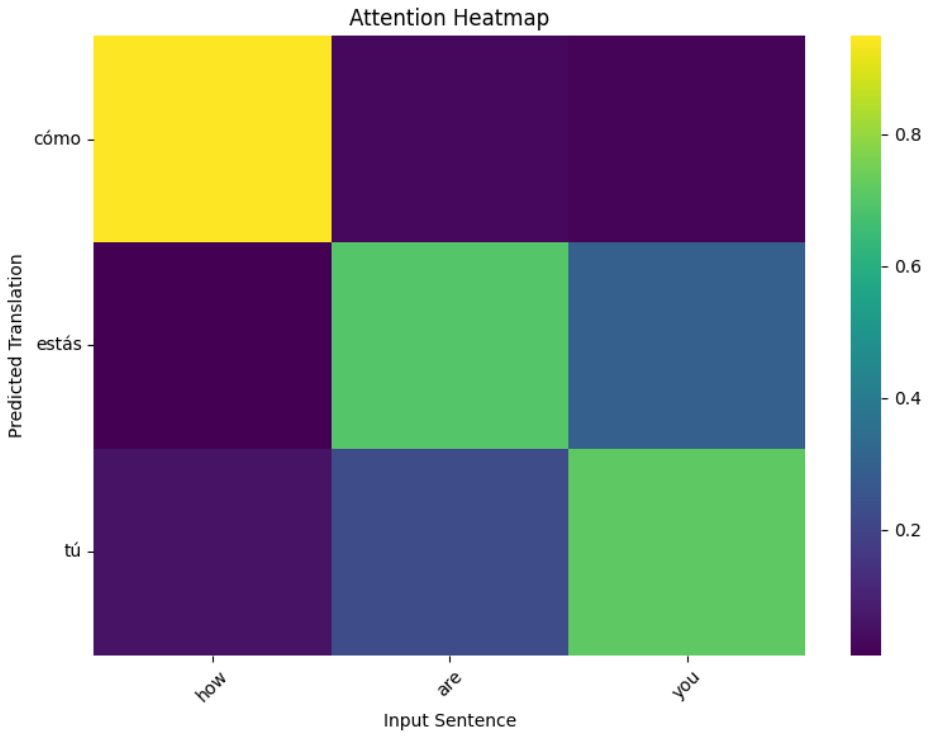

Figure 2: Attention Heatmap for "turn off the light" → "las luces la luz"

Reflection:

- The model shows strong alignment between "turn" and "las", and "off" and "luces", reflecting how it associates verbs with object descriptions.
- "la" and "luz" both attend to "the light", though the model seems to have duplicated the translation, possibly due to weak EOS prediction.
- Some attention is diffused across tokens, which might suggest the model is blending multiple concepts ("lights" vs. "light").

- There are no major alignment errors, but generation could be improved by better output filtering or stricter decoding.

Example 2:

`Input:` I love you

`Prediction:` te amo

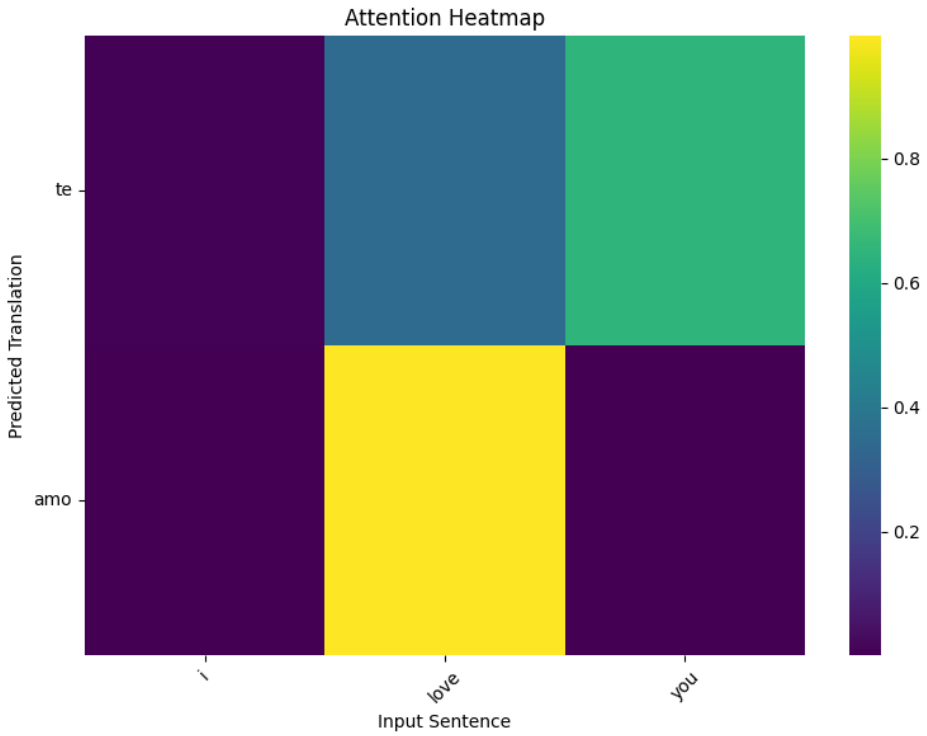

Figure 3: Attention Heatmap for "I love you" → "te amo"

Reflection:

- The attention is very clear and precise. "amo" focuses directly on "love" and "te" on "you".
- This sharp alignment is expected due to the simplicity and frequency of the phrase in natural language data.
- There's little to no attention leakage onto other tokens or padding.
- This demonstrates ideal attention behavior in short, well-aligned sequences.

Example 3:

`Input:` how are you

`Prediction:` cómo estás tú

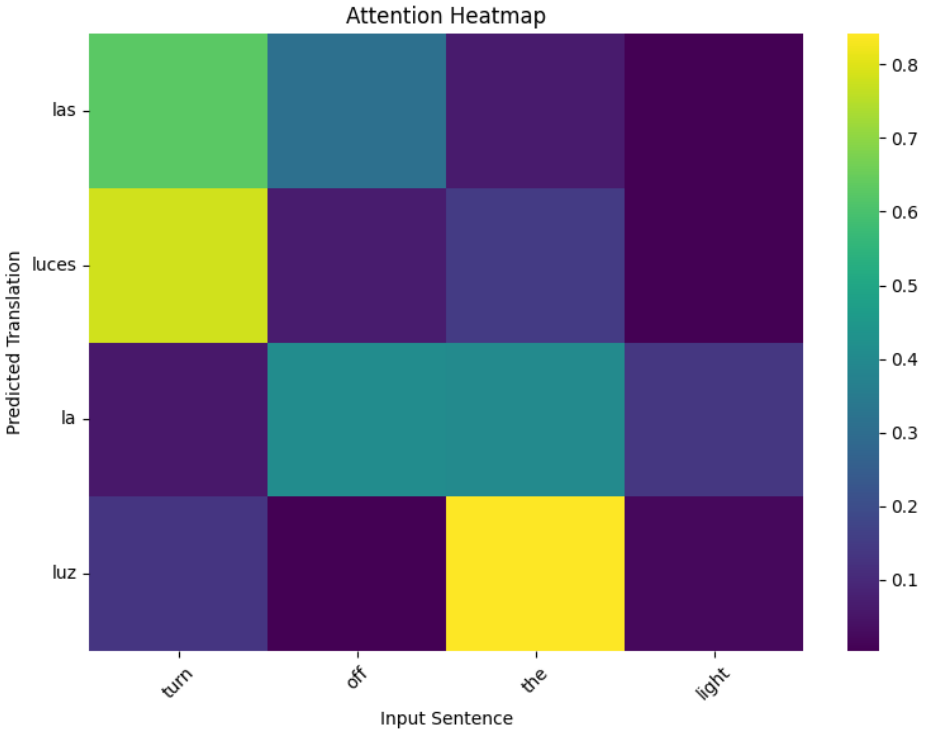

Figure 4: Attention Heatmap for “how are you” → “cómo estás tú”

Reflection:

- “cómo” strongly attends to “how”, while “estás” aligns with “are”, and “tú” attends to “you” — an expected and linguistically correct pattern.
- The subject pronoun “tú” appears at the end, which matches Spanish syntax and confirms proper sequence handling.
- No attention is wasted on ‘¡sos¿‘ or ‘¡eos¿‘, and masking appears to be working as expected.
- The model effectively captures both alignment and structural shift between English and Spanish.

9. Learning and Challenges

Challenge 1: Long Training Time

Problem: Training the attention-based Seq2Seq model on the full dataset took hours and often led to GPU memory issues in Colab.
Solution: Used ‘tf.data‘ with ‘padded_batch‘ for dynamic padding and efficient memory use. Reduced batch size to 32 during testing.

Challenge 2: Attention Layer Shape Mismatch

Problem: Faced persistent shape mismatch errors while integrating attention mechanisms into the Seq2Seq model. Initially used Keras Additive Attention, but it didn't align properly with the encoder-decoder outputs. Switching to the standard Attention layer also led to shape incompatibility during training. Even after applying ragged tensors and ensuring masking was handled correctly, the problem persisted.
Solution: To address this, I implemented a custom Bahdanau Attention mechanism, which gave full control over alignment scores and attention weights. This allowed seamless integration with the decoder LSTM and resolved all shape mismatch issues during training.

Challenge 3: Inference Repetition Bug

Problem: During decoding in the attention model, outputs were getting stuck in loops.
Solution: Diagnosed padding and EOS token handling. Fixed the inference loop to stop correctly at EOS and added padding masks.

Challenge 4: Repetitive Predictions and BLEU Score of 0

Problem: During inference, the model repeatedly generated the same words, and BLEU scores remained near zero.
Solution: Identified that the issue was due to incorrect handling of pre-trained layers and decoding during inference. Also discovered that class imbalance was contributing to poor learning. Fixed inference logic and analyzed token distribution to stabilize predictions.

Challenge 5: Class Imbalance Across Sequence Lengths

Problem: Shorter sentence pairs (e.g., 5-word sequences) were over-represented in training data, leading the model to favor shorter outputs.
Solution: Oversampled underrepresented sequence lengths (especially longer sentences) from 5 to 10 occurrences each, ensuring the model got balanced exposure to varied-length samples.

Challenge 6: GPU Memory Overload with Long Sequences

Problem: Training failed multiple times due to GPU memory overflow, especially with long sentence pairs.
Solution: Filtered the dataset to retain only pairs with both source and target lengths under 15

tokens. This significantly reduced memory usage, allowing the model to train within Colab's GPU constraints.

10. Conclusion

- Attention greatly improved alignment and fluency of translations.
- Pre-trained MarianMT models are powerful but less customizable.
- Dynamic padding and masking are crucial for proper training and evaluation.

Future work includes:

- Fine-tuning MarianMT on our custom dataset
- I tested both Seq2Seq models (with and without attention) on the original data.csv, both showed a stagnant growth in training, the losses went much lower than they went on filtered dataset but the BLEU was 0 and the translations were also quite different than the raw targets. I feel i need to do a little changes on them and then the model would perform much better with the original dataset which has both shorter and longer examples
- Exploring transformer-based models