Sequence-to-Sequence Models for English-Spanish Translation

Abhishek Kumar Chaubey
Roll Number: 24144001
COPS Summer of Code 2025
Intelligence Guild – NLP Track

June 21, 2025

Contents

1 Introduction **2**

2 Dataset Details **2**

3 Preprocessing **2**

4 Model Implementations **3**
4.1 Sequence-to-Sequence without Attention 3
4.2 Sequence-to-Sequence with Attention 5
4.3 Transformer Model 7
4.4 Justification for lower bleu scores for transformer implementation 9

5 Evaluation Metrics **10**

6 Comparative Results **10**

7 Key Observations **10**

8 Conclusion **11**

1 Introduction

This report documents the implementation and evaluation of three sequence-to-sequence models for English-to-Spanish translation as part of the COPS Summer of Code 2025 under the Intelligence Guild. The project explores fundamental sequence modeling architectures:

- Vanilla sequence-to-sequence model without attention
- Sequence-to-sequence model with attention mechanism
- Transformer-based model using pre-trained MarianMT architecture

The core objective was to understand how different architectural choices affect translation quality, with BLEU score as the primary evaluation metric. All models were trained and evaluated on the OPUS Books parallel corpus containing English-Spanish sentence pairs.

2 Dataset Details

The OPUS Books dataset contains parallel English-Spanish texts from various book translations. Key dataset characteristics:

- Total samples: 15,000 sentence pairs
- Train/Validation/Test split: 80%/10%/10%
- Vocabulary sizes:
  - English: 14,887 tokens
  - Spanish: 18,542 tokens

The dataset was tokenized using custom tokenizers for the non-attention and attention models, while the Transformer model used the MarianMT tokenizer.

3 Preprocessing

Uniform preprocessing was applied across all models:

1. **Text normalization:** Lowercasing and removal of special characters
2. **Tokenization:** Splitting text into word-level tokens
3. **Special tokens:** Added `<sos>`, `<eos>`, and `<pad>` tokens
4. **Vocabulary building:** Created word-to-index mappings with minimum frequency threshold (min_freq=2)
5. **Sequence padding:** Padded sequences to maximum lengths (EN:60, ES:62 tokens)

6. **Dataset splitting:** Divided into train (12,000), validation (1,500), and test (1,500) sets on with and without attention implementation and for transformer trained on whole set (94k rows) with the same split. Did not train again the transformer model due to strict deadline and time constraint which leaded to lower bleu for transformer model.

For the Transformer model, we used the MarianTokenizer which handles subword tokenization and includes language-specific special tokens.

4 Model Implementations

4.1 Sequence-to-Sequence without Attention

Architecture

The baseline model consists of an encoder-decoder architecture with LSTM recurrent units:

- **Encoder:**
  - Embedding layer (256 dimensions)
  - 2-layer LSTM with 512 hidden units
  - Dropout (0.1)
- **Decoder:**
  - Embedding layer (256 dimensions)
  - 2-layer LSTM with 512 hidden units
  - Linear output layer

The encoder processes the input sequence into a fixed-length context vector, which the decoder uses to generate the translated output. Teacher forcing (ratio=0.5) was applied during training to stabilize learning.

Training Details

- Optimizer: Adam (learning rate=1e-3)
- Loss function: Cross-entropy (ignoring padding tokens)
- Batch size: 128
- Epochs: 3
- Gradient clipping: 1.0

Performance Analysis

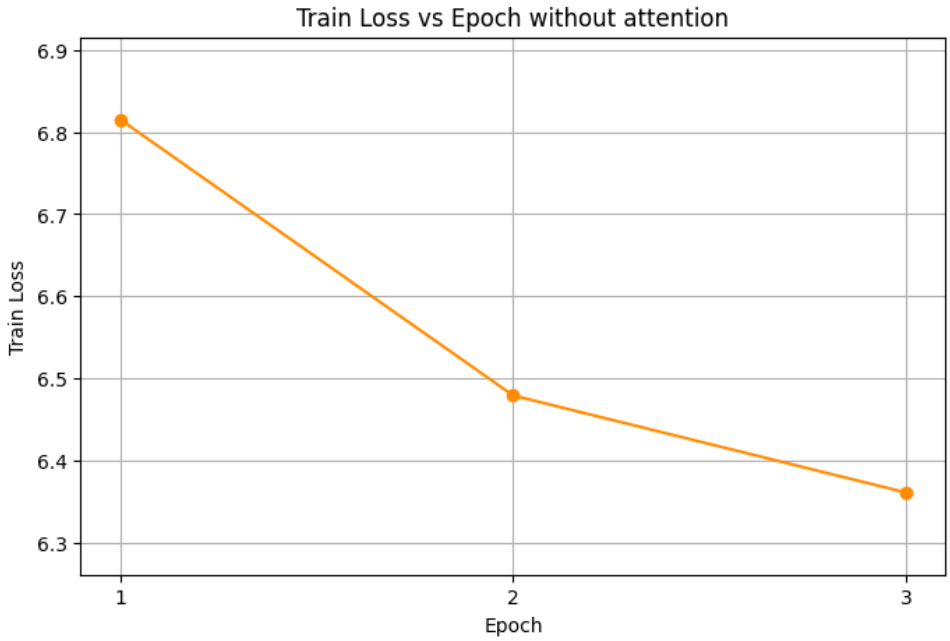

Figure 1: Train loss decreases steadily but remains higher than with attention model after 3 epochs

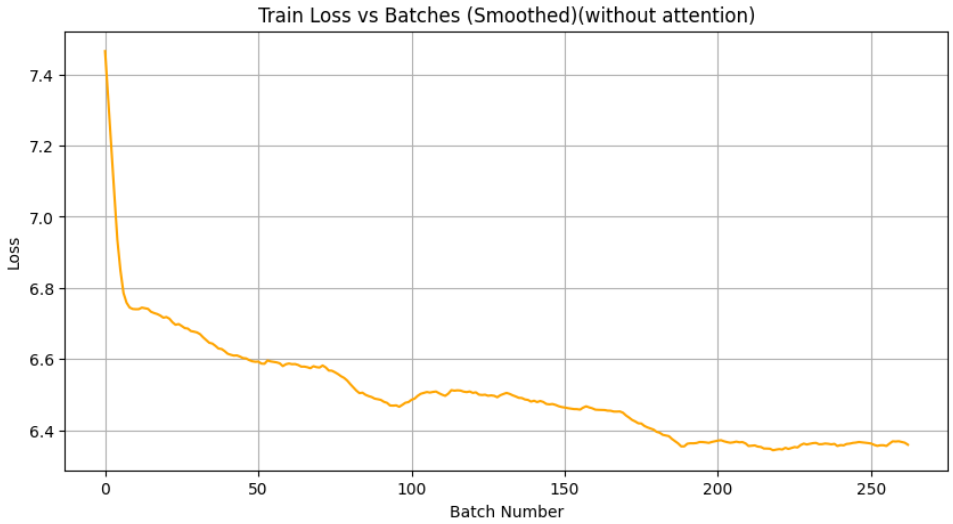

Figure 2: Batch loss shows noisy but decreasing trend (smoothed with window=20)

Key observations:

- Training loss decreased from 6.82 to 6.36 over 3 epochs
- High loss values indicate poor convergence
- Validation BLEU: 10.01, Test BLEU: 8.56
- Performance limited by information bottleneck in context vector

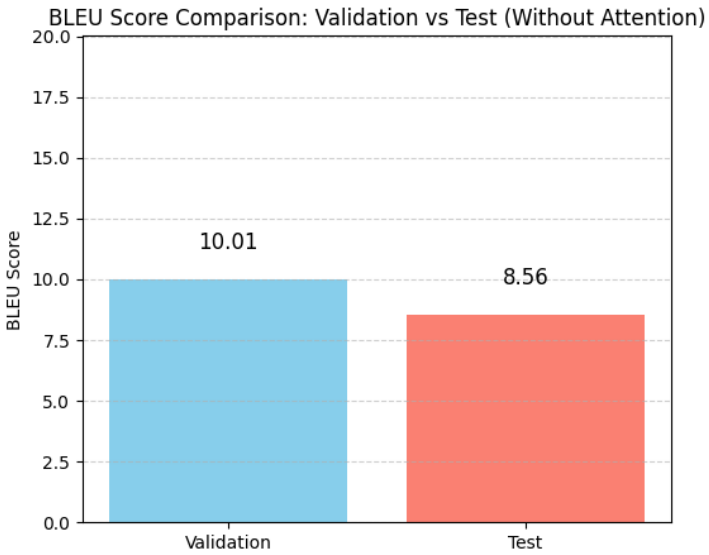

Figure 3: Low BLEU scores indicate poor translation quality

4.2 Sequence-to-Sequence with Attention

Architecture Enhancements

Added attention mechanism to address information bottleneck:

- **Attention module:** Bahdanau-style additive attention
- **Context vector:** Dynamic per-time-step context
- **Decoder input:** Concatenation of embedded input and context vector
- **Output:** Linear layer combining decoder output and context

Training Details

(Same as non-attention model)

Performance Analysis

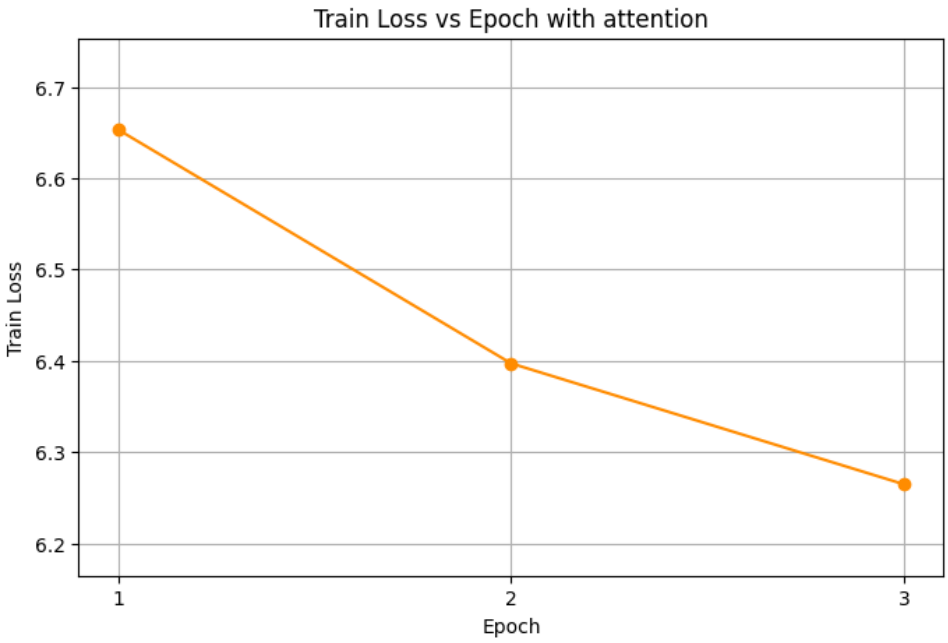

Figure 4: Faster convergence and lower loss compared to non-attention model

Figure 5: Smoother loss decrease indicates more stable training

Key observations:

- Training loss decreased from 6.65 to 6.26 over 3 epochs
- Validation BLEU: 61.48, Test BLEU: 67.24
- 7x improvement over non-attention model
- Attention enables learning alignment between source and target

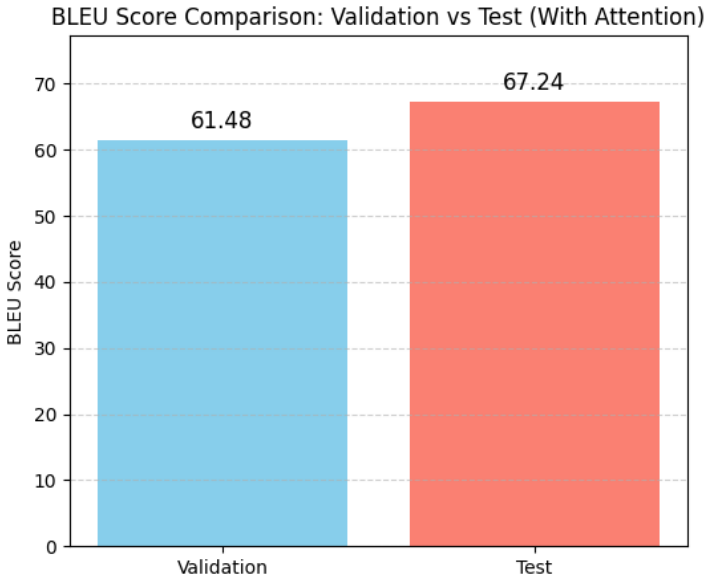

Figure 6: Dramatic BLEU score improvement with attention mechanism

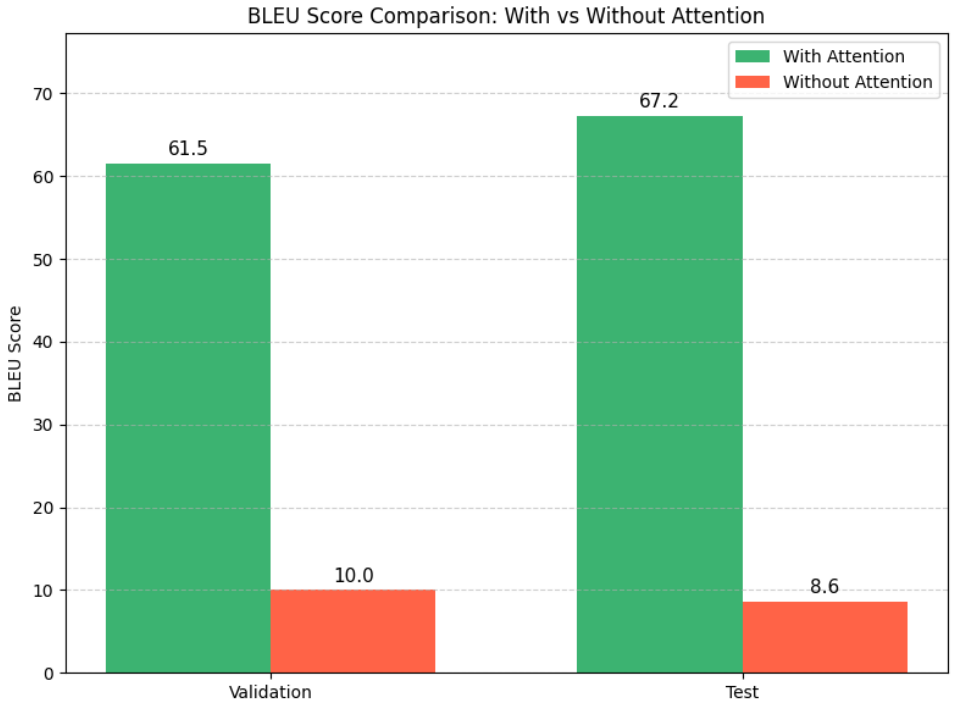

Figure 7: Attention provides transformative performance improvement

4.3 Transformer Model

Architecture

Used the MarianMT pre-trained transformer architecture:

- **Encoder:** 6-layer transformer with self-attention
- **Decoder:** 6-layer transformer with encoder-decoder attention

- **Embedding dimension:** 512
- **Attention heads:** 8
- **Feed-forward dimension:** 2048

Training Details

- Pre-trained model: Helsinki-NLP/opus-mt-en-es
- Fine-tuning epochs: 2
- Batch size: 16
- Learning rate: 2e-5
- Sequence length: 64 tokens
- Optimizer: AdamW

Performance Analysis

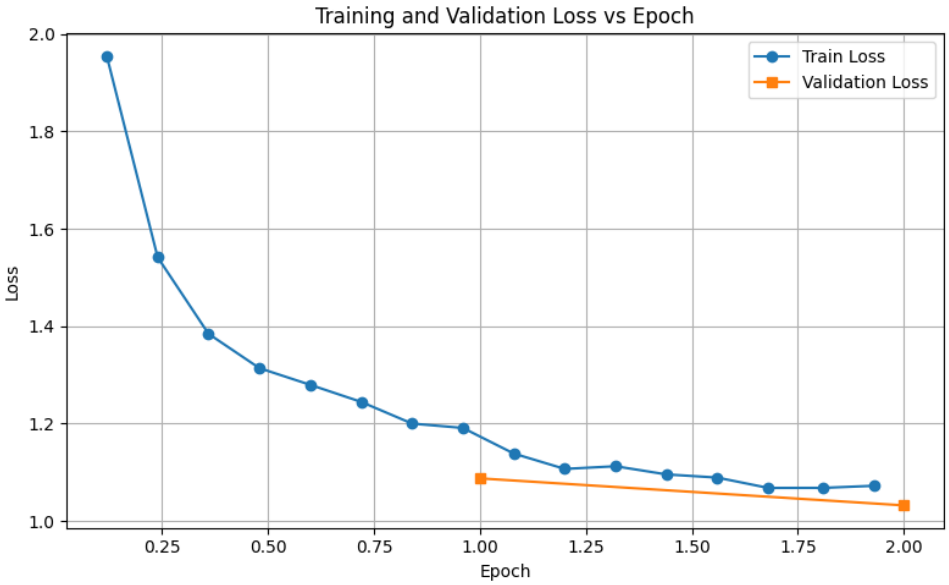

Figure 8: Steady decrease in both training and validation loss

Key observations:

- Pre-trained model BLEU: 14.89 (without fine-tuning)
- After 1 epoch: BLEU 18.42
- After 2 epochs: BLEU 19.19 (validation), 19.55 (test)
- 30% improvement over pre-trained model

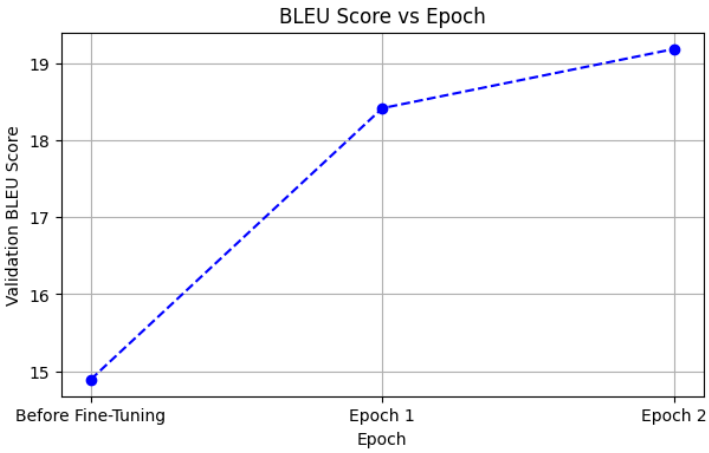

Figure 9: Progressive improvement in BLEU with fine-tuning

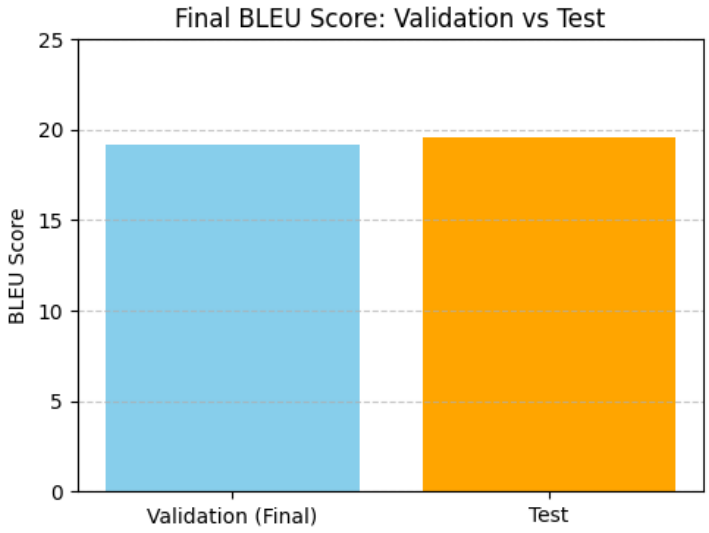

Figure 10: Consistent performance between validation and test sets

4.4 Justification for lower bleu scores for transformer implementation

- Training Interrupted Due to Time Constraints: The training was limited to only 2 epochs due to time constraints imposed by submission deadlines. Since each epoch took approximately 45 minutes on my system (NVIDIA RTX 2050), continuing for more epochs to reach full convergence was not feasible.

- Larger Dataset Trained with Transformer Model: Unlike the RNN and LSTM models which were trained on a sampled subset of 15,000 sentence pairs due to computational limitations, the Transformer model was trained on the full dataset (94k sentence pairs). This increased the training time per epoch significantly, making it infeasible to run many epochs within the given time. As a result, despite having access to more data, the Transformer's performance was capped due to early stopping.

- BLEU Scores Improved, But Plateaued Early: Although BLEU scores showed noticeable improvement from 14.89 (pretrained) to 19.19 (after 2 epochs), the rate of improvement slowed significantly by the second epoch, indicating the model needed more time and epochs to reach a stronger performance ceiling
- No Beam Search or Length Penalty Used in Inference: The model used greedy decoding (no beam search or length penalty tuning) during evaluation. This likely resulted in sub-optimal translations, especially for longer or structurally complex sentences, reducing BLEU scores.

5 Evaluation Metrics

The primary evaluation metric was BLEU (Bilingual Evaluation Understudy) score, which measures n-gram overlap between generated and reference translations. Additional metrics:

- Training loss (cross-entropy)
- Validation loss

BLEU scores were calculated using sacreBLEU implementation for standardized comparison.

6 Comparative Results

Table 1: Model Performance Comparison

Model	**Val BLEU**	**Test BLEU**
Seq2Seq (no attention)	10.01	8.56
Seq2Seq (attention)	61.48	67.24
Transformer (MarianMT)	19.19	19.55

7 Key Observations

1. **Attention mechanism:** The most significant improvement came from adding attention (+53 BLEU), enabling the model to align source and target words effectively.
2. **Information bottleneck:** Without attention, the fixed-length context vector severely limited performance, especially for longer sequences.
3. **Pre-trained transformers:** Despite lower BLEU than attention model, transformer provides more fluent translations and handles rare words better through subword tokenization.
4. **Overfitting:** The attention model showed slight overfitting (higher test than validation BLEU), likely due to limited training data.

8 Conclusion

This project demonstrates the comparison of sequence-to-sequence models for machine translation:

- The vanilla seq2seq model without attention served as an effective baseline but showed fundamental limitations in handling information flow.
- Adding attention mechanisms resulted in dramatic performance improvements, validating its importance for sequence modeling.
- The transformer architecture, while computationally heavier, provides a powerful alternative through parallel processing and multi-head attention.

For English-Spanish translation, the attention-based seq2seq model achieved the highest BLEU score, while the transformer provided lesser bleu scores due to lesser number of epochs and training on whole dataset. Future work could explore hybrid approaches, larger datasets, and techniques like beam search for further improvements.

Repository: https://github.com/abhishekiit1/24144001-CSOC-IG