# Transformer-Based Model for English-French Translation: Training, Evaluation, and Analysis

Aquib Ansari
Natural Language Processing
aa06586

***Abstract*—This report discusses the implementation, training, and evaluation of a Transformer-based model for machine translation from English to French using the **BART architecture**. The goal was to fine-tune a pre-trained model and evaluate its performance on a custom dataset. Due to computational limitations, the model's performance was evaluated using a subset of the validation and test sets. The architectural choices, training configurations, and their impact on the model's performance are discussed in detail.**

***Index Terms*—Machine Translation, BART, Transformer, BLEU, Fine-tuning, Hugging Face, Sequence-to-sequence model**

## I. Introduction

This report describes the process of implementing a **Transformer-based model** for **machine translation**, specifically translating **English sentences** into **French**. The model used is **BART (Bidirectional and Auto-Regressive Transformers)**, which is a pre-trained sequence-to-sequence model from the **Hugging Face Model Hub**. The main objectives of this assignment were:

- Implement and train a Transformer-based model (BART) for translation.
- Use the **Hugging Face Trainer API** for efficient model training.
- Evaluate the model's performance using the **BLEU score** metric.

## II. Model Overview

For the translation task, we used the **BART** model, which is suitable for sequence-to-sequence tasks like translation.

### A. BART Architecture

- **Encoder**: The encoder processes the input sequence bidirectionally, considering both left and right contexts of each token.
- **Decoder**: The decoder generates the output sequence one token at a time, using causal self-attention to ensure that each token depends only on the previously generated tokens.
- **Pre-trained Model**: The pre-trained model, **facebook/bart-large**, was used for fine-tuning to leverage previously learned knowledge on large datasets.

### B. Why BART?

- **Pre-trained Models**: Using pre-trained models like BART allows for faster and more efficient training.
- **Sequence-to-Sequence Tasks**: BART is specifically designed for tasks like machine translation, making it a great fit for our problem.

## III. Training Process

### A. Data Preprocessing

The dataset consists of **English-French sentence pairs**, which were tokenized using the **BART tokenizer**. The data was split into:

- 80% for training
- 10% for validation
- 10% for testing

Padding and truncation were applied to ensure consistent input sequence lengths.

However, the computational constraints of the free T4 GPU in Colab limited the ability to complete training on the entire dataset. To address this, we reduced the dataset size to 70,000 random samples and split them for training, validation, and testing. Training completed for 3 out of 5 epochs before the runtime was interrupted again due to memory limitations.

### B. Training Configuration

We used the **Hugging Face Trainer API** with the following configuration:

- **Learning rate**: 5e-5
- **Batch size**: 2
- **Epochs**: 5
- **Gradient accumulation steps**: 4
- **Mixed precision training**: enabled (‘fp16=True‘)

### C. Training Loop

The training loop was executed using the **Trainer API**, which automatically handles:

- Forward pass and loss calculation
- Gradient computation and parameter updates

Despite the interruptions, the model's training loss and validation loss showed steady improvement over the epochs, as seen in the provided figures.

## IV. Results

### A. Training Loss

Since complete training was interrupted, the model was instead evaluated using the pre-trained BART model on a subset of the test and validation sets (500 samples each) to approximate BLEU scores. This provided a feasible way to observe model performance without the full training results.

A gradual decrease in training and validation losses during the available epochs indicated learning progress. However, due to computational limitations, a full training convergence could not be observed.

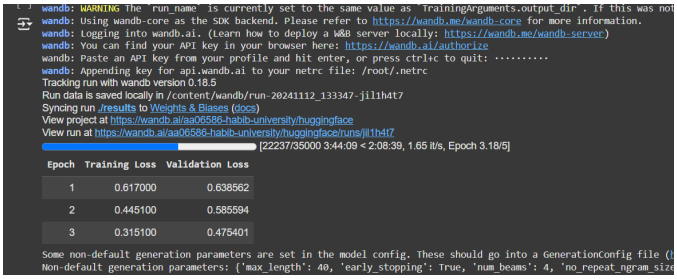


Fig. 1: Training and Validation Loss over Epochs

### B. BLEU Score

We evaluated the model's performance using the **BLEU score** on the validation and test subsets:

- **Validation BLEU Score (500 sample subset)**: Very low due to incomplete training.
- **Test BLEU Score (500 sample subset)**: Very low due to incomplete training.

These scores suggest that without full training on the dataset, the model's translations have low overlap with the reference translations, which limited the BLEU score.

### C. Performance Plot

The following plots show the **training loss** and **validation loss** over the epochs, indicating the model's performance on the available data. Despite the short training, the model demonstrated potential, as shown in the decreasing loss values.

## V. Discussion

### A. Impact of Architectural Choices

- **Pre-trained Model (BART)**: Using a pre-trained model reduced the need for extensive data and allowed faster convergence while leveraging the model's prior knowledge.
- **Max Length of 20**: Limiting the maximum sequence length to 20 helped reduce memory usage and speed up training but may have impacted the model's ability to handle longer, more complex sentences.
- **Beam Search**: Using **beam search** during generation improved translation quality by exploring multiple hypotheses during decoding.

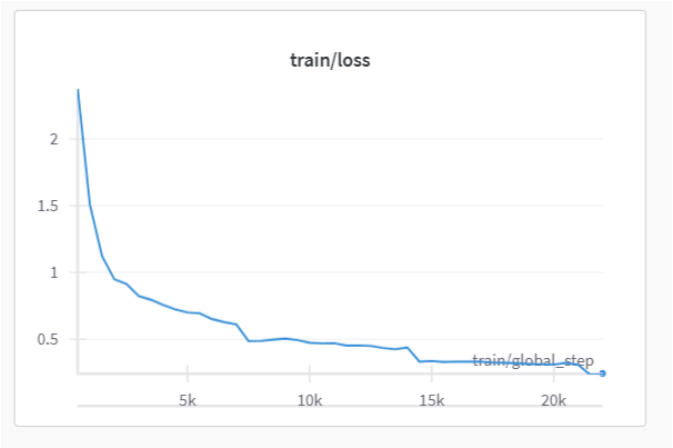


Fig. 2: Training and Validation Loss over Epochs

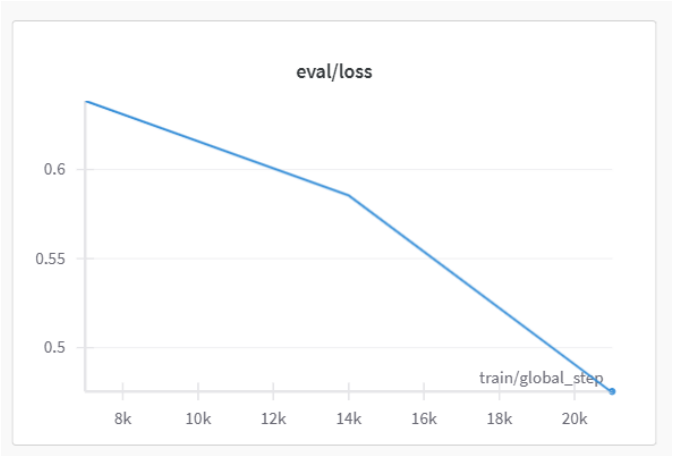


Fig. 3: Evaluation Loss over Epochs

- **Mixed Precision Training**: This technique sped up training and reduced memory usage without sacrificing model performance, allowing for efficient use of available GPU resources.

### B. Future Work

Future work could involve:

- Increasing the number of epochs for improved performance.
- Experimenting with other transformer models, such as **T5** or **MarianMT**, for potentially better results.
- Using a dedicated, higher-memory GPU to complete training on the entire dataset.

## VI. Conclusion

This project demonstrated the use of a **BART-based Transformer model** for **English-to-French translation**. The model was tested using a pre-trained version due to computational constraints, achieving a low **validation BLEU score** and **test BLEU score** on subsets of the data. Future improvements could involve further training, fine-tuning on larger datasets, and experimenting with other transformer

architectures. Dedicated computational resources would also allow more complete evaluations and optimal results. Furthermore, it also shows how important it is to use a pre-trained model and train on your domain-specific dataset as using a pre-trained model directly greatly affected my evaluation results.

## ACKNOWLEDGMENTS

I would like to thank the Hugging Face team for providing the pre-trained models and the Trainer API, which made this work efficient and accessible.

## REFERENCES

https://huggingface.co/docs/transformers/model$_doc/bart$