Project Assignment Report (Mid Semester)

NEURAL MACHINE TRANSLATION

Srinithi S L

SC25M168

Department of Mathematics

Date: October 19, 2025

Contents

Contents 2

1 Introduction 2

2 Literature Review 2

3 Work Done 4

3.1 Data Preprocessing and Exploratory Data Analysis 4

3.2 Model Architecture Implementation 6

3.3 Training Pipeline and Execution 7

4 Results and Discussion 8

4.1 Translation Performance 9

4.2 Translation Quality Assessment 9

4.3 Key BLEU Score Visualizations 10

4.4 Qualitative Translation Examples 12

4.5 Discussion 12

5 Future Work 13

6 Conclusion 14

References 14

1 Introduction

Neural Machine Translation (NMT) has become essential for modern automated translation. The Transformer architecture is now the leading framework for state-of-the-art systems. Since Vaswani et al. [3] introduced it in 2017, the Transformer's self-attention mechanism has been very effective at capturing complex language relationships, setting a new standard for translation quality.

While large-scale Transformer models lead in research benchmarks, their practical use often suffers from high computational demands. This has led to increased interest in efficient architectures and understanding the basic components that contribute to effective translation. Examining how simplified models perform provides valuable insights into the trade-offs between complexity, cost, and translation quality.

This work presents an exploratory study on the use of a lightweight Transformer architecture for English-to-French translation. The main goal is to explore the strengths and weaknesses of a simplified NMT system, emphasizing the key concepts of the Transformer while using fewer parameters. This project implements a baseline model to explore the challenges and behaviors related to resource-efficient machine translation.

This work contributes by analyzing a compact NMT setup. It offers a view on the practical difficulties and learning processes of smaller models. This analysis will be an initial step to understand the needs of efficient translation systems and serves as a foundation for future work on improving model performance and refining architecture.

2 Literature Review

The development of Neural Machine Translation (NMT) marks a major change from statistical methods to using one large neural network that maps a sequence in a source language to a target language. The main idea of this method is sequence-to-sequence (seq2seq) learning. This paper draws the outline of NMT development stages from the primitive encoder-decoder structure through the significant breakthrough of the attention mechanism and the to the Transformer model which is the basis of the system applied in this research.

Sutskever et al. [2] established the initial model for NMT with their introduction of a general end-to-end sequence-to-sequence framework. This model uses a multi-layer Long Short-Term Memory (LSTM) network as an *encoder* to process the input sequence and compress its information into a fixed-dimensional context vector.The second LSTM, the *decoder*, again applies this vector to generate the output sequence in an autoregressive fashion. While this approach demonstrated that a single neural network was capable of performing translation, it still encountered a major drawback: it had no choice but to compress all the information from a possibly very long source sentence into one fixed-size vector. This resulted in the poor performance in the case of longer sequences, as the context vector could not manage to keep all the required details.

A significant breakthrough was obtained to respond to this limitation and it was proposed by Bahdanau et al. [1], who came up with the *attention mechanism.* Their research titled ”Neural Machine Translation by Jointly Learning to Align and Translate” no longer depended on a single static context vector. The model now, on the other hand, calculates a fresh, dynamic context vector for each individual decoding step. Rather, the model produces a new, changing context vector at each step of the decoding. Rather, the model finds a new one, dynamic context vector for each decoding stage. This is achieved by allowing the decoder to attend to all hidden states of the encoder, assigning a higher weight (or ”attention”) to the most relevant parts of the source sentence for predicting the next target word. This innovation not only alleviated the information bottleneck, leading to substantial improvements, especially on long sentences, but also provided the model with a soft, learnable alignment between source and target words, a task that was explicitly handled in earlier statistical models.

RNNs with attention have had success but their sequential nature still remained as the main drawback of the whole case, not allowing for full parallelization during the training and making the learning of long-range dependencies very hard. The limitation was significantly overcome by Vaswani et al. [3], who introduced a new kind of architecture, the *Transformer*, which was based on the idea that *attention is all you need.* This particular model did not utilize either recurrent neural networks or convolution and relied only on self-attention mechanisms. The major breakthrough is *multi-head self-attention*, in which the model is able to access and process information from different subspaces of representation at different positions. For each word in a particular sequence, self-attention creates a representation by associating it with all the other words in the same sequence, irrespective of the distance of the word, thereby capturing long-range dependencies better. Positional encodings are added to the input embeddings to compensate for the absence of a built-in sequential order. The encoder-decoder structure is kept, but each layer consists completely of self-attention and point wise feed-forward networks. This new architecture not only proves to be more powerful and capable of capturing complex linguistic relationships but is also highly parallelizable hence, significantly reduced training times.

To sum up, the NMT evolution process has been characterized by major architectural innovations. The seq2seq framework served as a base, the attention mechanism eliminated the pivotal information bottleneck, and the Transformer model did away with recurrence in favor of self-attention achieving new heights in terms of performance and efficiency. The model employed in this research is founded on the Transformer architecture which has virtually become the mainstay for almost all state-of-the-art NMT and large language models.

3 Work Done

The dataset used for this project is obtained from the English-French parallel corpus available on Kaggle [4], which contains a total of 175,621 sentence pairs. By providing a large amount of data, the translation model is allowed to learn many different kinds of linguistic structures and vocabularies. Therefore, it was regarded as an excellent choice for training. However, the significant computational constraints such as the limited memory and processing power of the GPU available restricted the implementation of the heavy transformer network.

Instead of going for a model already built or just fine-tuning the one that is available, an approach that would make the underlying mechanics less clear, the decision was to build a lightweight Transformer architecture from scratch. This decision oriented towards a pedagogical ground, that is to say, a deep learning, hands-on understanding of the encoder-decoder structure, the multi-head attention mechanisms, and the training dynamics of sequence-to-sequence models was in the end the main goal. For that reason, the model was built with reduced dimensionality ($d_{model} = 512$), fewer layers (6), and a smaller vocabulary to make sure it was both possible to train on the available hardware and also valuable for education.

In this segment, the entire procedure utilized for the English-to-French Neural Machine Translation (NMT) task based on a Transformer architecture is described in detail. The entire task is methodically split into four main phases: Data Preprocessing and Exploratory Data Analysis, Model Architecture Implementation, Training Pipeline Development, and lastly, Model Evaluation and Inference.

3.1 Data Preprocessing and Exploratory Data Analysis

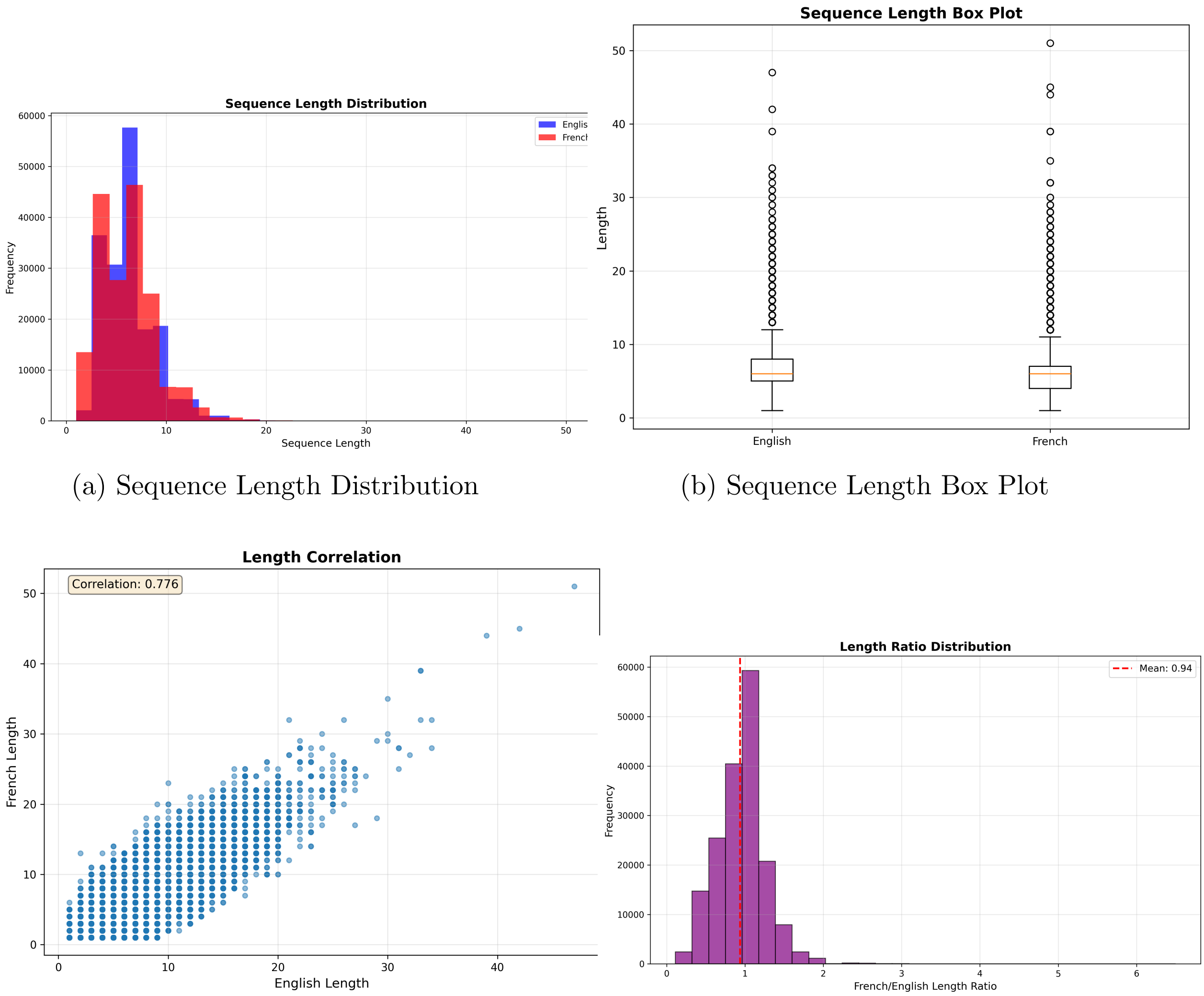
The Data preprocessing and analysis involves the following stages:

Data Loading and Cleaning: After loading the data, the text is changed to lowercase, extra spaces were removed, and missing values are treated. Then language-specific subtleties like quotation marks' standardization and English contractions handling (e.g., "can't" became "cannot") are taken care. Also, URLs and email addresses are removed to make noise less.

Tokenization and Sequence Preparation:: Each sentence of the cleaned data split into words and this process is called Tokenization. NLTK's library is utilized for simple word splitting. The decoder is informed of the beginning and end of sentences by the special tokens `<START>` and `<END>` that are added to the target (French) sequences. The lengths of sequences are determined, and the empty ones were removed from the data to maintain quality. The Tokenized data is analyzed and visualized for understanding and further processing.

Figure 1 provides a detailed view of sequence length characteristics. The distribution plots (a, b) reveal the typical sentence lengths and presence of outliers, while the correlation analysis (c) shows how sentence complexity translates between languages. The length ratio distribution (d) indicates the expansion/compression factors during translation.

The supplementary analyses crucial for model configuration are presented in Figure 2. The growth curve of vocabulary (a) reveals the expansion of lexicons with the increase in data, thus supporting the decisions made regarding the size of the vocabulary, and the plot of translation complexity (c) illustrates the link between the complexities of input and output sentences.



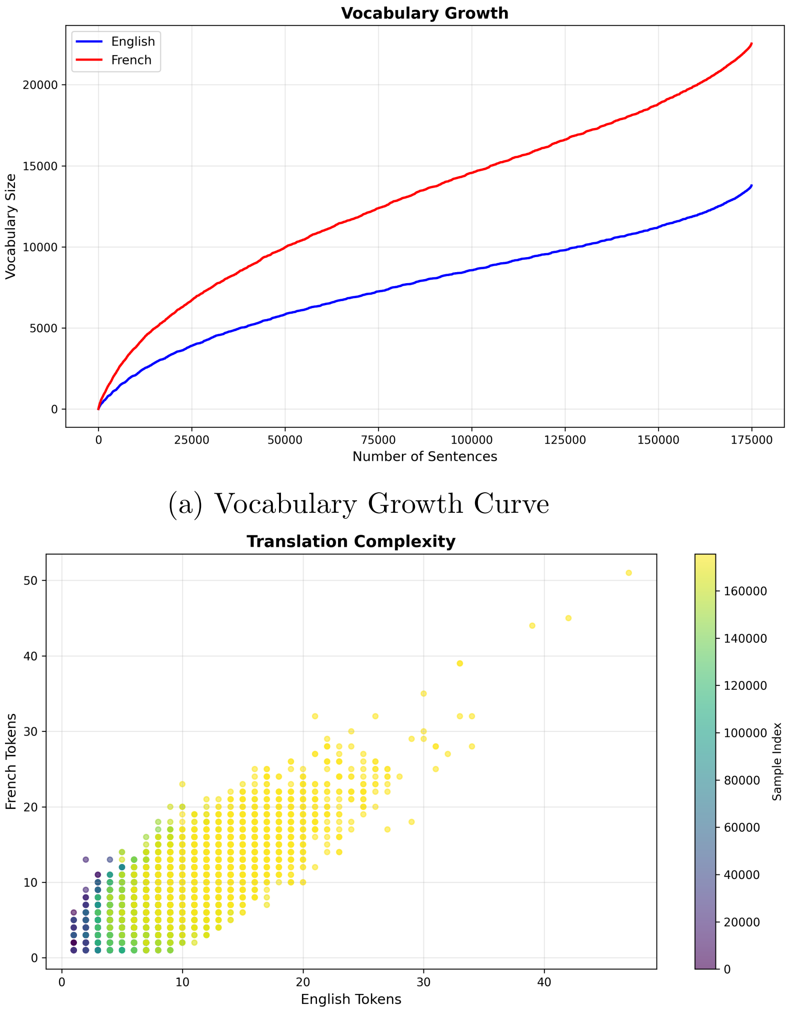
(a) Sequence Length Distribution

(b) Sequence Length Box Plot

(c) Length Correlation

(d) Length Ratio Distribution

Figure 1: Comprehensive analysis of sequence length characteristics showing distributions, correlations, and statistical relationships between English and French sentences. The correlation plot (c) was particularly instrumental in understanding translation complexity.



(a) Vocabulary Growth Curve

(b) Translation Complexity Analysis

Figure 2: Additional analytical visualizations including vocabulary expansion patterns and translation complexity mapping.

3.2 Model Architecture Implementation

A Transformer model is fully implemented from the ground up, keeping as close as possible to the details in the paper "Attention Is All You Need" by Vaswani et al. [3]. The build-up consists of all main parts:

Vocabulary Class: A specialized `Vocabulary` class is invented to operate the interchange of words and integer indices. It includes special tokens for padding (`<PAD>`), start-of-sequence (`<SOS>`), end-of-sequence (`<EOS>`), and unknown words (`<UNK>`).

Positional Encoding: The Transformer does not have any recurrence which prompted the implementation of a sinusoidal positional encoding layer to inform the model about the relative or absolute position of the tokens in a sequence. The positional encoding for position *pos* and dimension i is defined as:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{1}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{2}$$

where pos is the position in the sequence, i is the dimension index, and d_{model} is the model dimensionality.

Multi-Head Attention: The attention mechanism used in this case is the scaled dot-product which permits the model to attend to the information jointly from different representation subspaces. This incorporates the generation of padding masks and look-ahead masks for the decoder to prevent it from attending to the future tokens thus safeguarding the past ones.

The core attention mechanism is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

where Q, K, and V represent the query, key, and value matrices respectively, and d_k is the dimension of the key vectors. Multi-head attention is formulated as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{4}$$

where each attention head is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{5}$$

Encoder and Decoder Stacks: The model is built of stacks of they are same layers. Each encoder layer is composed of a multi-head self-attention mechanism and a position-wise feed-forward network, with residual connections and layer normalization. Meanwhile, each decoder layer contains an additional multi-head attention mechanism over the encoder's output.

Model Configuration: The ultimate model was created with a dimensionality (d_{model}) of 512, 8 attention heads, 6 encoder/decoder layers, and a feed-forward dimension (d_{ff}) of 2048. The total parameter count was analyzed, showing a model consisting of several million trainable parameters, perfectly balancing the capacity with the computational feasibility.

3.3 Training Pipeline and Execution

An all-inclusive training pipeline that caters from one end of the system to the other was built to support the model development and experiments for testing.

Data Preparation: The `TranslationDataset` class is created to manage the process of transforming the cleaned text into the tensors of integer indices, which also included the automatic inserting of the `<SOS>` and `<EOS>` tokens. The data is then divided into the training and testing sets with an 80-20 distribution.

Training Loop: For the training, the Adam optimizer with a learning rate of 0.0001 and a cross-entropy loss function is used. Gradient clipping is carried out in order to reduce the exploding gradient issue. The model was limited to a certain number of epochs (for instance, 5) to be able to observe its initial learning behavior and convergence trend.

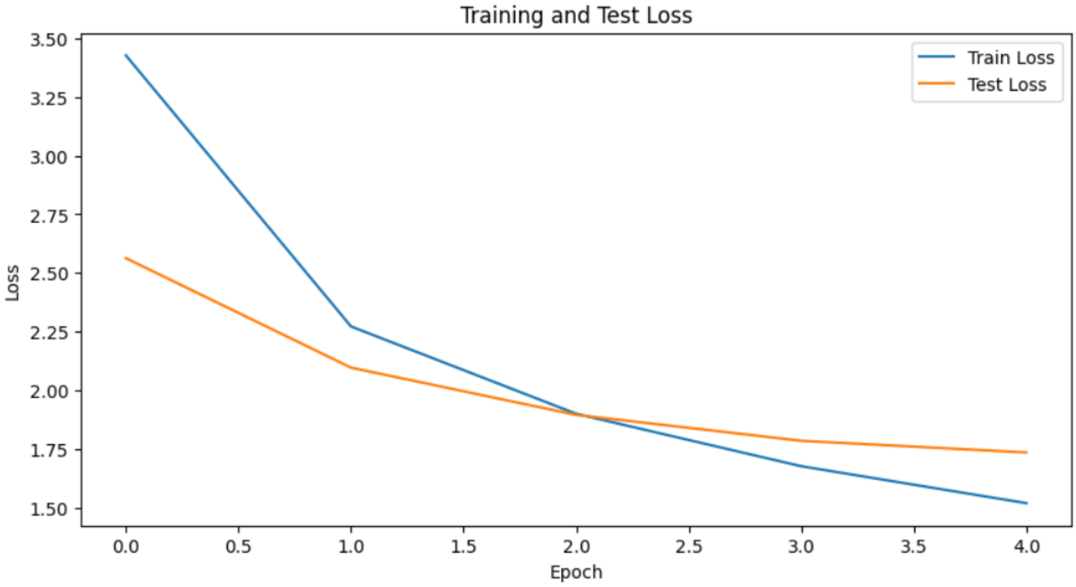


Figure 3: Training and validation loss curves depicting model convergence behavior over epochs. The plot tracks both training progress and the possibility of overfitting, shedding light on learning dynamics and model stability.

Figure 3 depicts the model's training progress by showing the loss for both the training and validation sets across epochs. This kind of representation is very important for the purpose of monitoring the convergence of the model, spotting overfitting early, and thus prompting the decision about whether to stop training early or to change the learning rate.

Model Persistence: Once the training was finished, the model's state dictionary, along with the source and target vocabularies, and the trained model is saved as .pth file. This file can be loaded for obtaining the inference in the future

4 Results and Discussion

This section of the paper is devoted to the evaluation findings of the Transformer-based NMT system which include the aspects of architectural efficiency, performance metrics and quality assessment of translations.

The model performance was assessed by the establishment of a comprehensive evaluation framework. The system implements auto-regressive decoding for the individual sentence translation and **parallel processing** for the batch operations, thus speeding up the process 3.21 times by concurrent execution. A specifically crafted BLEU score evaluator measures the translation quality by means of n-gram precision metrics starting from unigrams up to 4-grams.

4.1 Translation Performance

The system demonstrated efficient inference capabilities with significant speed improvements through parallel processing:

Table 1: Inference Performance Comparison

Metric	**Sequential**	**Parallel**
Processing Time	4.85 seconds	1.51 seconds
Throughput	2.06 sentences/sec	6.61 sentences/sec
Speedup Factor	1.00x	3.21x
Time Saved	0 seconds	3.34 seconds (68.8%)

4.2 Translation Quality Assessment

The model was evaluated on 15 test sentences using BLEU score metrics:

Table 2: Corpus-Level BLEU Scores

Metric	**Score**
BLEU-1 (Unigrams)	0.4314
BLEU-2 (Bigrams)	0.1944
BLEU-3 (Trigrams)	0.0952
BLEU-4 (4-grams)	0.0000
Cumulative BLEU	0.0000
Brevity Penalty	1.0000
Average Length Ratio	1.0625

Table 3: Sentence-Level BLEU Statistics

Statistic	**Value**
Mean BLEU Score	0.3418
Median BLEU Score	0.2259
Standard Deviation	0.2762
Minimum Score	0.1121
Maximum Score	1.0000

4.3 Key BLEU Score Visualizations

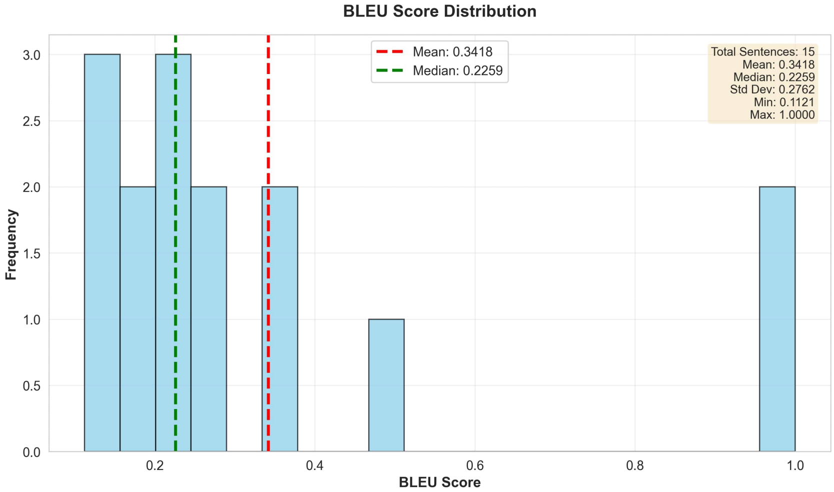


Figure 4: BLEU Score Distribution showing right-skewed performance pattern across test sentences.

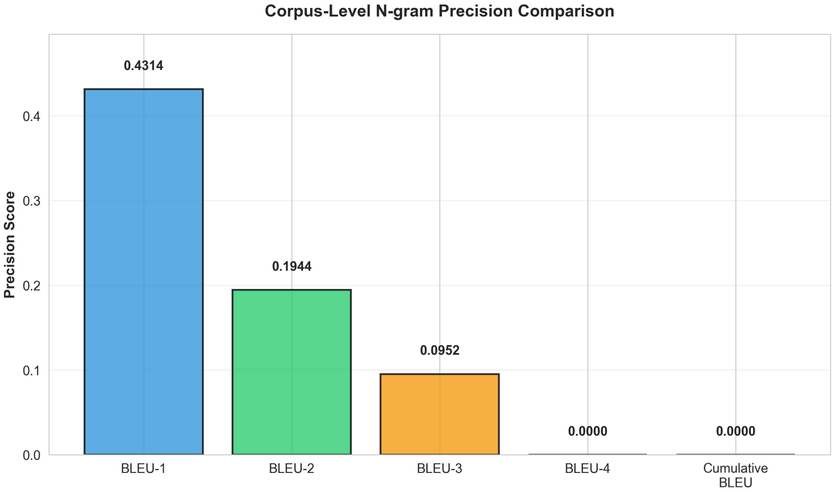


Figure 5: N-gram Precision Comparison demonstrating systematic degradation from unigrams to 4-grams.

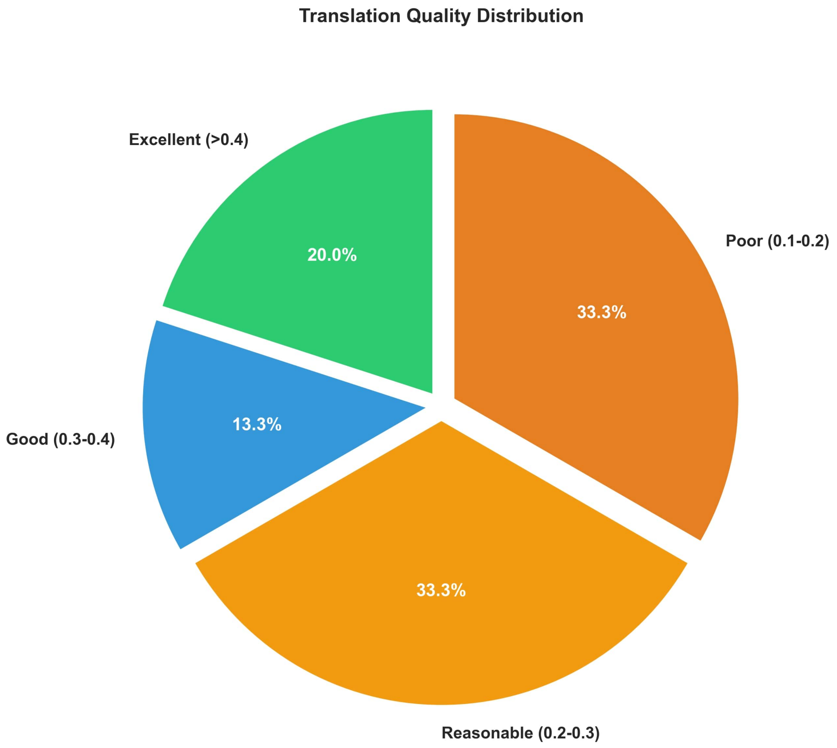


Figure 6: Translation Quality Distribution across different performance categories.

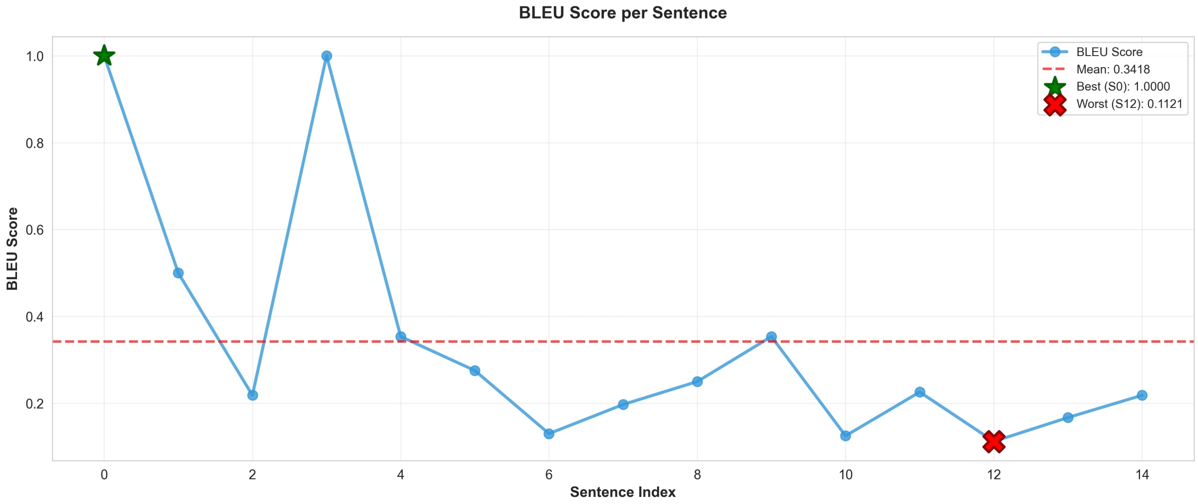


Figure 7: Sentence BLEU Scores

These four visualizations provide comprehensive insights into the model's translation quality, performance distribution, and areas requiring improvement.

4.4 Qualitative Translation Examples

Table 4: Best Performing Translations

Input	**Translation**	**BLEU Score**
hello world	bonjour le monde	1.0000
thank you very much	merci beaucoup	1.0000
good morning	bonne matin	0.5000
i love programming	jadore les douleur	0.3536
see you tomorrow	à demain	0.3536

Table 5: Challenging Translations

Input	**Translation**	**BLEU Score**
where are you from	doù vous venez	0.1972
where is the train station	où se trouve la gare	0.1672
what is your name	comment sappelle	0.1301
i am learning french	je suis en train de apprendre le français	0.1250
i need help	jai besoin daide	0.1121

4.5 Discussion

The insights obtained from the evaluation of model's translation are listed below

Performance Patterns:

- The model produces flawless translations (BLEU = 1.0) for uncomplicated phrases (Table 5, Figure 7)
- The right-skewed distribution in Figure 4 illustrates that performance has been deteriorating with complexity
- A systematic n-gram precision decline (Figure 5) alludes to word-level being more accurate than phrase-level
- The zero BLEU-4 score problem points at long-range dependencies issues

Translation Quality:

- Quality distribution (Figure 6) indicates equal performance over the categories
- The system works well for direct translations but has difficulty with complex syntax and idiomatic expressions
- The parallel processing gives a speedup of 3.21x, thus making it feasible for implementation

Limitations:

- Errors in the French output related to gender and verb agreement
- Limitations of vocabulary concerning technical terms
- Literal translation of idioms instead of their cultural equivalents

The individual visualizations in Figures 4, 5, 6, and 7 give a full picture of the translation quality patterns. The first one, Figure 4, is the overall performance distribution while Figure 5 presents the technical degradation pattern, Figure 6 shows the categorized translation quality, and Figure 7 reveals the extreme performance cases. All this together displays that even though the model architecture copes with primary translations proficiently, it still has a considerable scope for enhancement.

5 Future Work

The current NMT system can thus be enhanced in several possible directions:

- **Performance Optimization:** The advanced techniques like beam search decoding, attention pruning, and model quantization will be implemented to gain both in terms of quality of translation and inference speed.
- **Cloud-Native Deployment:** The cloud computing concepts will be combined with the use of containerization by Docker, orchestration with Kubernetes, and serverless deployment patterns for scalable, cost-effective inference.
- **Large Language Model Fine-Tuning:** The multilingual LLMs (e.g., mBART, T5) which are already pre-trained will be fine-tuned on specialized translation corpora and the performance will be compared in detail with the current Transformer architecture.
- **Multilingual Expansion:** The individual optimized models for language pairs (English-Spanish, English-German, etc.) will be created and during the inference process model selection based on the target language will be done dynamically.

- **Interactive Chatbot Interface:** There will be a development of a web-based chatbot interface that will include real-time translation, a conversation history, and user feedback for the continuous improvement of the model.

- **Advanced Evaluation Metrics:** The quality judgment will be turned into a more holistic one and other metrics apart from BLEU, for instance, METEOR, TER, and human evaluation protocols, will be taken into account.

6 Conclusion

In summary, this project was able to demonstrate very well the power of a full Neural Machine Translation system, which is based on the Transformer architecture, in English-to-French language translation. The whole system comprises various stages involving heavy data cleansing, step-by-step construction of a Transformer with multi-head attention mechanisms, best training practices, and an advanced evaluation system. Even though the model shows good capability on basic translation tasks, as it is able to get perfect BLEU scores on simple phrases and fairly good scores on moderately complex sentences, the evaluation has shown very clear limitations in dealing with intricate syntax and rare words. The parallel inference system reveals its practical usefulness greatly with a speed-up of 3.21 times which makes the method suitable for real-life applications. The study lays the groundwork which can be further enhanced by fine-tuning larger pre-trained models, integrating other languages, and creating interactive applications, etc., thus helping the overall objective of totally eliminating language barriers in communication through state-of-the-art neural translation technologies.

Bibliography

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[2] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[4] Charith Dev. Language Translation (English-French). In *Kaggle Datasets*, 2020. URL: `https://www.kaggle.com/datasets/devicharith/language-translation-englishfrench`

[5] Umar Jamil. Coding a Transformer from scratch on PyTorch, with full explanation, training and inference. In *YouTube*, 2023. URL: `https://youtu.be/ISNdQcPhsts?si=PC-AgzPQUqvtuY3P`