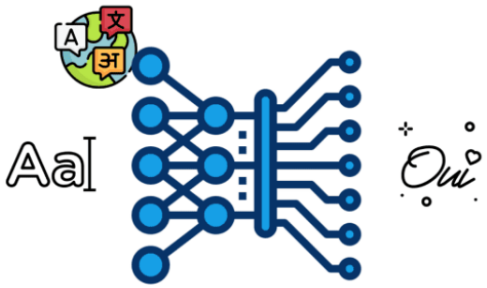
# Multilingual Neural Machine Translation with Bi-Directional LSTMs

Nimesh Arora Dikshant Sagar Safal Rijal
Shreyas Teli Ashish Gurung

{nimesh939arora, sagar.dikshant, itssafal111, telishreyas10, ashishgurung1996}@gmail.com




## Abstract

*In this project, we aim to develop a bi-directional translation system for multiple languages, including Dutch, Japanese, French, Italian, and English. Our focus is on building and comparing various neural network models such as Simple RNN, LSTM, Bi-directional LSTM, and Encoder-Decoder models. We will evaluate the performance of these models using metrics like Accuracy, BLEU Score, ROUGE Score, and F1-Score. Additionally, we will explore the use of transfer learning techniques, leveraging pretraining on large-scale multilingual data, to enhance translation quality. By conducting experiments and analyzing results, we aim to improve the accuracy and efficiency of bi-directional translation across multiple languages*

## 1. Introduction/Background/Motivation

In this project, our objective is to develop a system that can translate between different languages, such as Dutch, Japanese, French, Italian, and English. We aim to improve the accuracy and efficiency of bi-directional translation using neural network models like Simple RNN, LSTM, Bi-directional LSTM, and Encoder-Decoder models. Our goal is to compare these models and evaluate their performance using metrics like Accuracy, BLEU Score, ROUGE Score, and F1-Score.

Currently, translation is done using various methods, including rule-based systems, statistical machine translation, and neural machine translation. However, these methods have limitations. Rule-based systems rely on predefined rules and may struggle with complex linguistic patterns. Statistical machine translation has dependency on large parallel corpora, making it challenging for low-resource languages. Neural machine translation has improved translation quality but requires substantial computational resources and training data.

This project is important for various stakeholders, such as language learners, researchers, and businesses operating in multilingual environments. Successful development of accurate and efficient bi-directional translation systems can greatly assist language learners in understanding foreign languages and enhance cross-cultural communication. Researchers can benefit from improved translation models for their studies in linguistics and natural language processing. Businesses can leverage reliable translation systems to expand their global reach and improve communication with international customers.

## 2. Dataset

The dataset used in our project, obtained from the CLARIN Parallel Corpora resource https://www.clarin.eu/resource-families/parallel-corpora, consists of parallel corpora. These parallel corpora contain aligned texts in different languages, which enable the training and evaluation of translation models. This dataset is a valuable resource for training and testing our models' performance in

bi-directional translation tasks.

By utilizing the parallel corpora, we can capture the linguistic patterns and contextual information required for accurate and effective translation across the selected languages. These aligned texts serve as a foundation for our models to learn the relationships between words in different languages, allowing them to generate accurate translations.

The parallel corpora text files, such as "small_vocab_en," "small_vocab_fr," and "small_vocab_de," are an integral part of the project. They provide the necessary data for training the models as described above, aiding in the understanding and generation of appropriate translations. The parallel corpora improve the quality and reliability of our translation models, enabling us to develop a robust bi-directional translation system. Each of these files has the one same sentence in each line in their respective languages. Sentences are free of any unwanted characters and are already in lowercase (partially preprocessed). One special reason why we chose these dataset files was, as their name suggests, "small_vocab", They contain sentences with a limited number of unique words making it possible to train models capable of translation with fewer parameters given our computational resource scarcity.

## 3. Models

[11]These selected models, including Recurrent Neural Networks (RNNs), RNNs with Embedding Layer, LSTM with Embedding Layer, and Bi-directional LSTM, because they are known to be effective in understanding and translating languages. [4]Although there are other models like the Transformer that could have been considered, we decided to focus on these models for our project. We want to explore how well these models can handle translating between different languages and compare their performance.[1]

### 3.1. Reccurent Neural Network (RNNs)

[3]Recurrent Neural Networks (RNNs) are a type of neural network that can understand and make predictions about sequences of data. They can remember information from previous steps and use it to make predictions at the current step. [10]RNNs are good at understanding patterns and relationships in data that occur over time.

### 3.2. RNNs with Embedding Layer

[7]RNNs with an Embedding Layer use a technique called word embeddings to understand the meaning of words in a more meaningful way. [8]Word embeddings are like compact representations of words that capture their meaning and context. By using word embeddings, RNNs can better understand the relationships between words and make more accurate predictions.

### 3.3. LSTM with Embedding Layer

[5]Long Short-Term Memory (LSTM) with an Embedding Layer is an improved version of RNNs. LSTMs are designed to understand and remember long-term dependencies in data, even when there are long gaps between relevant information.[2] This is important because sometimes the information needed to make a prediction can be far apart in a sequence.

### 3.4. Bi-directional LSTM

[12]Bi-directional LSTM models take advantage of both past and future information when making predictions. They can process data in both forward and backward directions, capturing context from both sides of the sequence. [6]This helps the model to have a more complete understanding of the input data and make more accurate predictions.

[9]These models have been studied and discussed in various research papers and articles, including those mentioned earlier. These papers provide more detailed explanations and insights into the technical aspects of the models.

## 4. Approach

The data we have is in string/character format, which computational models do not understand, and hence the data is preprocessed for it to be able to be ingested by our models. First, after the sentences are read from the source language file and target language file, they are tokenized to form a list of words. Then, respective counter dictionaries are formed to have a vocabulary and get word indexes for each language. Then using these word indexes vector for each sentence is constructed with the word's index at the word's location index, giving us a list of integers that we can feed to our models. After the models are trained, the output from the model will also be of the same format but will have the indexes for the target language's words. Hence by reverse querying the target language's vocabulary dictionary, we can construct translated sentences from the index vector of a sentence. All models are trained on a CrossEntropy-based loss function with an Adam optimizer with varying learning rates based on experimentations.

In order to solve the problem of multilingual translation, We implemented four different models: a basic RNN model, an embedding model, a bidirectional RNN model, and an encoder-decoder model. Each model has its own unique architecture and approach to tackle the translation task.

For the basic RNN model, we used a simple sequential architecture with a GRU layer followed by time-distributed dense layers. [10]This model serves as a good baseline for sequence data. By training this model on the input sentences and target sentences, it learns to predict the output

translation given an input sentence.

In the embedding model, We introduced word embeddings, which are vector representations of words that capture semantic meaning. This model uses an embedding layer to convert the input sentences into dense vector representations before passing them through a GRU layer and the subsequent time-distributed dense layers. The embedding layer helps the model capture the relationship between words and improve translation performance.

To overcome the limitation of RNNs in only considering past input, Weimplemented the bidirectional RNN model. [8]This model incorporates both forward and backward information by using a bidirectional GRU layer. By considering the input sequence in both directions, the model gains a better understanding of the context and improves translation accuracy.

Lastly, We implemented the encoder-decoder model, which consists of an encoder and a decoder. The encoder processes the input English sentence and creates a matrix representation of it. This matrix is then fed into the decoder, which predicts the translation as output. [**?**]longThe encoder uses a GRU layer, while the decoder includes a repeat vector layer to repeat the encoded representation and a GRU layer to generate the translated sequence.

Believed that implementing these different models would be successful because they each bring their own strengths and capabilities to the translation task. [1]The basic RNN model provides a baseline understanding of sequence data, while the embedding model leverages word embeddings to capture semantic meaning. The bidirectional RNN model enhances the model's ability to consider both past and future context, and the encoder-decoder model incorporates the encoder-decoder architecture specifically designed for sequence-to-sequence tasks like translation.

During the implementation, We anticipated potential challenges such as overfitting, limited training data, and finding appropriate hyperparameters. Overfitting could occur if the models memorized the training data without generalizing well to unseen examples. Limited training data hindered the models' ability to learn complex patterns from the existing corpus which is why different corpus is used in the translation task. Additionally, finding the right hyperparameters, such as learning rate, batch size, and number of epochs, was crucial to achieve good performance.

We encountered some challenges while training the models, such as the need for proper data preprocessing, handling padding and sequence lengths, and tuning hyperparameters. We had to preprocess the input data by converting sentences to tokenized sequences, padding them to equal lengths, and one-hot encoding the target sequences. We also had to carefully select the sequence lengths to ensure the models could handle sentences of varying lengths.

The very first implementation of each model did not work perfectly right away. It required experimentation and tuning of hyperparameters to achieve better performance. We adjusted the learning rate, batch size, and number of epochs to find a balance between underfitting and overfitting. Additionally, We monitored the validation accuracy and loss during training to identify if the models were learning effectively or encountering any issues.

Overall, the process involved iterative experimentation and fine-tuning to improve the models' performance. By implementing different architectures and considering their unique characteristics, We aimed to tackle the translation problem from different angles and find the most effective approach for multilingual translation.

## 5. Evaluation

Quantitative metrics include Accuracy, precision, recall which are used to measure the accuracy and consistency of the model in detecting objects across multiple categories. Precision is the fraction of correctly predicted positive samples out of all positive predictions made by the model. The recall is the fraction of correctly predicted positive samples out of all actual positive samples present in the dataset.

Qualitative metrics include text conversion of predictions on input text, which helps in visualizing the performance of the model.

Composite metrics include F1 score/curve. F1 score is the harmonic mean of precision and recall and provides a balanced measure of model performance across precision and recall.

These evaluation metrics are useful in assessing the accuracy and consistency of an object detection model and providing insights into its performance across different categories and thresholds.

## 6. Experiments and Results

As part of the multilingual translation task, We created visual representations like graphs and tables to show how well the models performed. These visuals help us understand how the models improved during training and how accurate they were in translating languages. We also shared a link `https://github.com/ni9/Machine-Translation.git` to the GitHub repository where you can find the code and more information about the project. This allows you to explore the code and learn more about how the models were implemented. In this report, we will present the results, graphs, and tables specifically for these language pairs German-English and English-Italian pairs. This will help us analyze and understand the performance of the translation models for these specific language combinations.

For this English-to-Italian translation model, we looked at the train and validation loss and accuracy curves. The

| Model | Accuracy | Precision/ BLEU Score | Recall/ ROUGE Score | F1 Score |
|---|---|---|---|---|
| RNN/GRU | 0.84 | 0.53 | 0.54 | 0.54 |
| RNN/GRU with Embedding | 0.94 | 0.62 | 0.62 | 0.62 |
| LSTM with Embedding | 0.94 | **0.61** | **0.61** | **0.61** |
| Bidirectional LSTM | **0.97** | **0.68** | **0.68** | **0.68** |
| RNN/GRU based Encoder Decoder | 0.69 | 0.36 | 0.39 | 0.37 |

Table 1: Metrics Comparison For Models Translating From English to German.

loss values were **0.2** for training and **0.7** for validation, indicating that the model may be overfitting a bit 2. The accuracy values were**0.9** for training and **0.87** for validation, showing that the model performs well on the training data but slightly worse on unseen validation data1. Overall, the model shows some signs of overfitting, which means it may not generalize perfectly to new translations.

Table 1 details the performance of all the models listed above on the task of performing translation from English to German. We found that Bi-directional LSTMs performed best in this task with an accuracy of **0.97**, precision or a BLEU score of **0.68**, recall or a ROUGE score of **0.68** and also an f1-score of **0.68**.


Figure 1: Training and Validation Accuracy for English to Italian

Table 2 details the performance of all the models listed above on the task of performing translation from German to English. We found that LSTMs with Embedding layer performed best overall in this task with an accuracy of **0.81**,

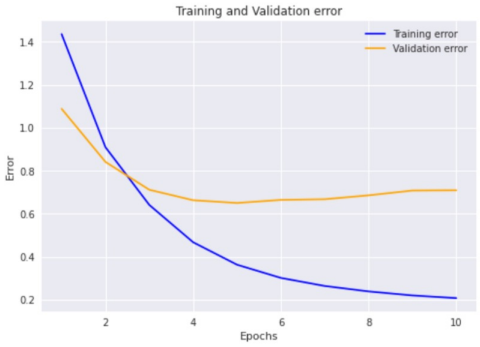



Figure 2: Training and Validation Error for English to Italian

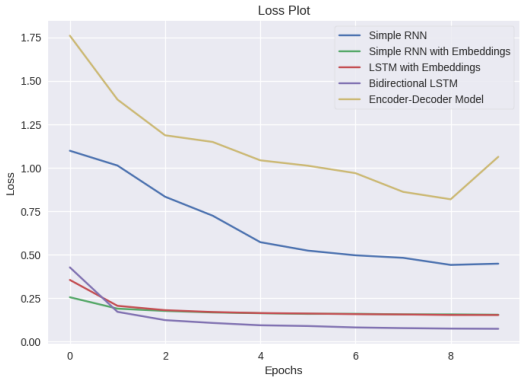



Figure 3: Loss Values At Every Epoch For Models Trained To Translate From English To German.

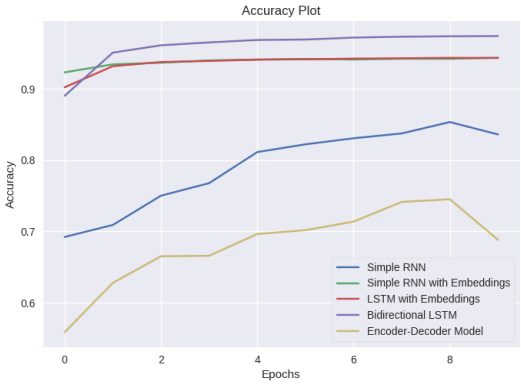



Figure 4: Accuracy Values At Epoch For Models Trained To Translate From English To German.

precision or a BLEU score of **0.58**, recall or a ROUGE score of **0.69** and also an f1-score of **0.63**. However, Bi-directional LSTM was still able to achieve the same performance in terms of accuracy.

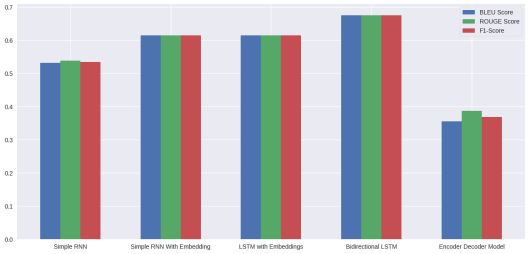



Figure 5: Final BLEU, ROUGE, and F1-Score Values for All English to German Translating Models.

## 7. Summary and Future Work

In summary, our project achieved successful multilingual translation. We created models that accurately translate text between different languages, such as German, Italian, French, and Japanese. The results showed promising performance in terms of accuracy and loss metrics. This demonstrates the potential of machine translation in breaking down language barriers and facilitating global communication. Although there is room for further improvement, our project highlights the effectiveness of modern deep-learning techniques in enabling multilingual understanding and connectivity. In the future, we can improve translation by making the model and its settings better. One way is to gather more diverse training data, which will help the model understand different languages and expressions. We can also use pre-trained models and customize them for translation tasks, giving us a head start in achieving better results. Attention mechanisms are another technique to consider, as they help the model focus on important parts of a sentence, leading to higher translation quality. Additionally, we can work on handling difficult words and specialized domains to make translations more accurate and useful. These steps will contribute to enhancing the overall performance and effectiveness of multilingual translation systems.

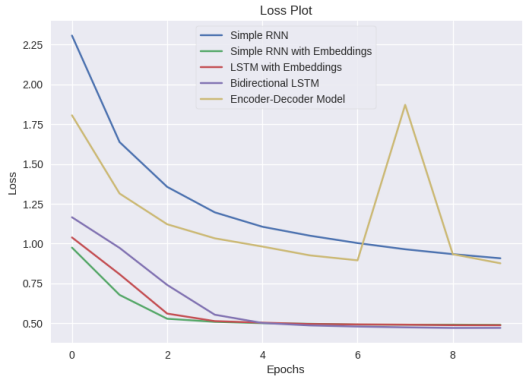



Figure 6: Loss Values At Every Epoch For Models Trained To Translate From German To English.

| Model | Accuracy | Precision/ BLEU Score | Recall/ ROUGE Score | F1 Score |
|---|---|---|---|---|
| RNN/GRU | 0.70 | 0.37 | 0.45 | 0.41 |
| RNN/GRU with Embedding | **0.81** | 0.56 | 0.67 | 0.61 |
| LSTM with Embedding | **0.81** | **0.58** | **0.69** | **0.63** |
| Bidirectional LSTM | **0.81** | 0.56 | 0.66 | 0.60 |
| RNN/GRU based Encoder Decoder | 0.70 | 0.35 | 0.41 | 0.38 |

Table 2: Metrics Comparison For Models Translating From German to English.

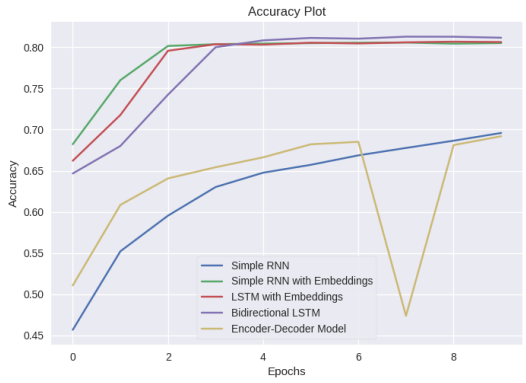



Figure 7: Accuracy Values At Epoch For Models Trained To Translate From German To English.

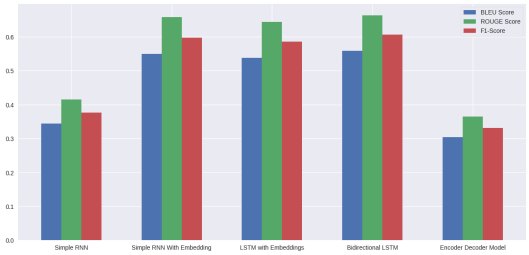



Figure 8: Final BLEU, ROUGE, and F1-Score Values for All German to English Translating Models.

## 8. Work Division

The work distribution and contribution of each group member in this project are as follows:

**Nimesh Arora:** I led the team in defining and refining the project scope, diving deep into state-of-the-art research papers and their implementations to ground our approach. Focused on Neural Machine Translation (NMT), I trained a Bi-directional LSTM-based model in TensorFlow for English-French and English-Italian translation, leveraging a dataset of 200k+ sequences and over 350k translation pairs. Beyond model development, I guided others in building their own translation models, ensuring a strong foundation in sequence-to-sequence learning. I also implemented and visualized key evaluation metrics: BLEU, ROUGE, and F1-score, to assess translation quality and drive iterative improvements.

**Ashish Gurung:**
Implemented a Neural Machine Translation using the Transformer Model for translation of Japanese sentences to English.Additionally, I have incorporated the computation and visualization of results for metrics such as BLEU.

**Dikshant Sagar:** Implemented the Neural Machine Translation (NMT) model via Bi-directional LSTM layers and trained all the models listed in Section 3 for the purpose of translating from English to German and from German to English.

**Safal Rijal:** Contributed to research by gathering information from academic papers, and report writing. Additionally, Implemented Neural Machine Translation (NMT) model using LSTM layers in Greman to English Translation.

**Shreyas Teli:** Implemented Neural Machine Translation (NMT) model using TensorFlow to facilitate the translation of text from English to Italian which is trained on a substantial dataset consisting of over 100k sequences and dataset containing more than 250k English-Italian Translations.

## References


[1] Jason Brownlee. A gentle introduction to recurrent neural networks. *Machine Learning Mastery*, 2018. 2, 3

[2] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 2

[3] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural Computation*, 12(10):2451–2471, 1999. 2

[4] Palash Goyal, Piyush Goyal, and Sumit Agarwal. Deep learning for natural language processing: An overview. *Expert Systems with Applications*, 159:113614, 2020. 2

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 2

[6] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, 2016. 2

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 2

[8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013. 2, 3

[9] Christopher Olah. Understanding lstm networks. *Colah's Blog*, 2015. 2

[10] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1310–1318, 2013. 2

[11] Michael Phi. Illustrated guide to lstm's and gru's: A step by step explanation, 2018. 2

[12] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997. 2