Assignment 3 Report



Name: David Naddall

Student Email: [659586@student.inholland.nl](mailto:659586@student.inholland.nl)

# Introduction

This report details the development and evaluation of two types of neural machine translation (NMT) models—RNN-based and Transformer-based—using the 'TED-Parallel-Corpus' dataset. The models are trained to translate text from Dutch to Spanish. This task is a significant component of natural language processing and aims to create systems that can translate text accurately and efficiently between languages.

# Preprocessing Methods

The dataset utilized, the 'TED-Parallel-Corpus,' consists of TED Talk translations into various languages. For this project, the focus is exclusively on Dutch and Spanish translations. The preprocessing steps are explained in the next section.

## Data Cleaning and Preparation

The comprehensive corpus was downloaded, and a filtering process was applied to extract only the sentences translated between Dutch and Spanish. Post-filtering, the Dutch-Spanish sentence pairs were written to a separate file dedicated to further processing.

With the isolated data, a parsing routine was implemented to organize the content into pairs where each Dutch sentence was aligned with its corresponding Spanish translation. This pairing is fundamental for training the models, as it allows them to learn the translation from source to target language directly from aligned sentence pairs.

Configuration and Standardization:

The vocabulary size was set at 15,000 words. This configuration limits the models to consider only the most frequent words, thus balancing the complexity of the model with computational efficiency.

The input sequences were standardized to a fixed length of 20 words. This uniformity is essential for batch processing in neural networks and ensures that all inputs are treated consistently during training. A custom standardization function was employed to convert all text data to lowercase and remove specific punctuation marks while retaining numeric brackets. This preprocessing step is critical for preparing the text in a format suitable for neural network processing, enhancing model performance by maintaining consistency in input data. There after these text pairs were randomly shuffled to prevent order-based biases during training. The majority of the data, 70%, was allocated to the training set. This is the dataset on which the neural network models would perform the majority of their learning, adjusting their internal parameters to minimize translation errors based on the provided inputs and corresponding outputs. The validation set, comprising 15% of the data, is used to evaluate the model during the training process. This subset helps in tuning the hyperparameters and provides a check against overfitting, ensuring that the model's performance generalizes well to new, unseen data.

The remaining 15% of the data serves as the test set. This segment is used strictly for evaluating the model's performance after the training process is complete. It provides a final metric that indicates how well the model is expected to perform in real-world translation tasks, reflecting its ability to handle completely new data.

## Models

### RNN Encoder-Decoder

#### Encoder

Bidirectional GRU: The encoder consists of a Bidirectional Gated Recurrent Unit (GRU) with a latent dimension of 1024. This setup allows the encoder to process the input text from both directions (forward and backward), capturing context from the entire sentence more effectively than a unidirectional approach. The GRU is chosen for its efficiency in modeling time series data like text, where dependencies between words can significantly impact translation accuracy.

Embedding Dimension: 256

Each word in the vocabulary is converted into a 256-dimensional vector. The embedding layer serves as the initial step in processing the input text, transforming discrete words into dense vectors that capture semantic meanings in a continuous vector space. This representation is crucial for the neural network to understand and manipulate linguistic information.

#### Decoder

The decoder also utilizes GRU layers but operates in a forward direction only. It takes the output from the encoder—a context-rich representation of the entire input sentence—and generates the translated text one word at a time. This sequential generation mimics the natural process of human translation, where the context of the entire sentence influences each word's choice in the output.

The decoder predicts the next word in the sequence based on the current state and the cumulative knowledge from the previous words. This step is repeated until the complete sentence is translated or a termination condition (like a maximum length or an end-of-sentence token) is met.

A dropout layer (50% rate) is included to prevent overfitting.

### Transformer Model

Embedding Dimension: 256

Each word in the input sequence is transformed into a 256-dimensional vector through the embedding layer. This dense representation captures semantic properties of the words, enabling the model to process and understand the textual input effectively.

The model uses custom TransformerEncoder and TransformerDecoder classes, incorporating multi-head attention and feed-forward networks.

The architecture includes custom TransformerEncoder and TransformerDecoder classes. These components are tailored to handle specific aspects of the language translation task, utilizing the inherent strengths of the Transformer model such as parallel processing and learning long-range dependencies.

Dense Dimension: Inside the Transformer's feed-forward networks, the dense dimension is set at 2048. This dimensionality provides the networks ample space to perform complex transformations, allowing the model to learn detailed and nuanced representations of the input data.

Number of Attention Heads: The model features 8 attention heads in its multi-head attention layers. This setup enables the model to attend to different parts of the input sequence simultaneously, providing a more comprehensive understanding of the context. Multi-head attention is pivotal in enhancing the model's ability to focus on relevant parts of the input when predicting each word in the translation.

Dense Dimension: 2048 in the feed-forward network

Number of Attention Heads: 8

Essential for the model's ability to parse various dependencies and nuances in the text. By dividing the attention process into multiple heads, the Transformer can maintain different subspaces of attention, which is crucial for understanding and translating complex sentence structures accurately.

Both the RNN and Transformer models use the sparse\_categorical\_crossentropy loss function. This choice is suitable for classification tasks with multiple classes, where each class represents a word in the vocabulary. It is computationally efficient as it allows the model to predict the probability of each word without the need to convert the output into a one-hot encoded format.

Optimizers: The Transformer model employs the RMSprop optimizer, while the RNN uses Adam. RMSprop is chosen for the Transformer due to its effectiveness in handling the non-stationary nature of text data by adapting the learning rate to the parameters. It helps in stabilizing the updates during training. Adam, used for the RNN, is known for its adaptive learning rate capabilities, which provide a balance between computational efficiency and robust learning.

# Results

### Rnn

The model training, across 30 epochs, shows a gradual decrease in loss and a slight increase in accuracy on the training dataset. However, the validation loss increases with each epoch, suggesting that the model is overfitting to the training data:

Training Loss and Accuracy: Starting at a loss of 0.6548 with an accuracy of 65.06%, the model's loss reduced to 0.5748 with a slight increase in accuracy to 68.07% by the end of 30 epochs. This progression indicates that the model is learning from the training data, adapting its parameters to minimize the training loss effectively.

Validation Performance: Conversely, the validation loss started at 1.4673 and worsened to 1.6458, with corresponding accuracy decreasing, highlighting a classic sign of overfitting. This suggests that while the model is getting better at predicting the training data, it is performing poorly on unseen data (validation set).

### Transformer

Accuracy: The model's accuracy on the training data starts at 67.95% and gradually increases to 77.83% by the 30th epoch. This indicates a consistent improvement in the model's ability to classify or predict correctly on the training dataset.

Validation Accuracy: Similarly, the validation accuracy starts at 69.44% and increases, reaching a peak of 73.67% at the 14th epoch, but it mostly stabilizes around 73% towards the later epochs. The validation accuracy is crucial as it reflects the model’s performance on unseen data.

Loss: Both training and validation loss decrease initially, which is expected as the model learns from the training data. However, the validation loss starts increasing slightly after around the 20th epoch, suggesting potential issues like overfitting.

### Translation Examples and Analysis

The translation outputs show varied levels of success and some typical issues with machine learning-based translation systems:

Both the RNN and Transformer models often resort to generic or repeated phrases, indicated by the use of [UNK] for unknown words, which appears frequently in the translations. This suggests a limitation in the vocabulary or the model's inability to handle less frequent words.

The translations often miss the mark in terms of context and detailed meaning. For example:

Dutch: "Hij loopt elke ochtend langs het kanaal."

RNN Translation: "[start] y se [UNK] cada mañana en el canal [end]"

Transformer Translation: "[start] está [UNK] cada mañana [end]"

Both translations struggle with some words, losing the subject of the sentence ('He') entirely in the Transformer's output.

Longer or more complex sentences see increased errors and odd phrasing:

Dutch: "De kinderen spelen buiten terwijl hun ouders het diner voorbereiden."

RNN Translation: "[start] los niños [UNK] a su familia y si le [UNK] a nosotros mismos [end]"

Transformer Translation: "[start] mientras los niños los [UNK] a su [UNK] a lo [UNK] [end]"

These translations are not only incorrect but also illustrate the model's struggles with sentence structure and logical coherence.

### BLEU

The BLEU (Bilingual Evaluation Understudy) scores provided for the RNN and Transformer translations offer a quantitative measure of the translations' quality compared to a reference translation. The extremely low scores across all sentences suggest significant issues with both translation models. Here's a detailed analysis:

Analysis of Translation Quality

Very Low BLEU Scores: BLEU scores close to zero across all translations indicate that the model-generated translations align very poorly with the reference translations. This is evident from scores like 5.7592918561109494e-155, which essentially round to zero, indicating almost no overlap with the reference sentences beyond perhaps some common stop words.

Both models struggle with vocabulary, as indicated by frequent [UNK] tokens, suggesting either a limitation in the models' vocabulary or an inability to handle certain words.

The translations often miss or incorrectly translate key verbs and subjects, leading to translations that are not just inaccurate but also nonsensical in the target language context.

Grammatical structure is often not preserved, and in many cases, the semantic meaning is completely lost.

The RNN model occasionally retains more structure from the source sentence but still fails to deliver accurate translations. The Transformer model, despite its advanced architecture, does not necessarily outperform the RNN model in terms of BLEU score, which could be due to issues with how it handles context or due to the complexity of the model not being fully leveraged during training.

# Conclusions

The analysis of translation examples from the neural network models trained on the 'TED-Parallel-Corpus' dataset highlights significant challenges and areas for improvement in machine learning-based translation systems. Both the RNN and Transformer models exhibit issues that compromise the quality and reliability of their outputs:

The frequent appearance of [UNK] tokens in translations indicates a limitation in the vocabulary management of both models. This results in generic or repeated phrases that do not convey the specific meaning intended in the original texts. This issue suggests that the models struggle to handle less frequent words, which is a critical aspect of robust translation systems.

The models often fail to accurately capture the context and detailed meanings of the sentences they translate. For instance, both models failed to translate the subject 'He' in the sentence from Dutch, indicating a significant loss of meaning in translation. Such inaccuracies point to a deficiency in the models' ability to understand and translate the grammatical structures and semantic nuances of the input language.

The models' translations of longer and more complex sentences show increased errors and odd phrasing. This is evident from their outputs, which not only misinterpret the original sentences but also disrupt logical coherence. These issues are indicative of the models' limited capability in handling complex linguistic structures and maintaining context over longer text spans.

The near-zero BLEU scores across all translations underscore the inadequacy of the current models in producing translations that align with the reference sentences. Such low scores reveal that the translations fail to match the reference translations even on basic overlaps such as common stopwords.

These findings clearly demonstrate a pressing need for significant revisions in model architecture, training procedures, and data preprocessing. To enhance the translation quality and make the models suitable for practical translation tasks, focused improvements in handling vocabulary diversity, contextual accuracy, and sentence complexity are essential. By addressing these critical areas, there is potential to substantially improve the performance of machine translation systems, making them more reliable and effective in real-world applications.

# Discussion

Future work could explore the integration of larger datasets and more complex preprocessing techniques, such as subword tokenization, to enhance translation accuracy. Additionally, experimenting with different model configurations and more advanced optimization algorithms might yield further improvements. During the training of the neural network model over 30 epochs, a notable observation was the absence of a callback mechanism, specifically for early stopping. This oversight meant that training continued even after the validation metrics ceased to improve, leading to a scenario where the validation loss began to increase—a classic sign of overfitting. Without a callback to halt training based on the validation performance, the model potentially over-learned the training data details, diminishing its generalization capabilities to new data. Furthermore, Continuing training without improvements in validation metrics leads to unnecessary computational expense and time. This made it difficult to make improvements or changes to the code as it would take quite long to train the model.

# References

Kulkarni, A. (n.d.). TED-Multilingual-Parallel-Corpus. GitHub repository for the TED Parallel Corpus dataset, which contains sentences from TED Talks translated into various languages. This dataset was crucial for training the neural network models for the task of translating text between Dutch and Spanish. Available at: TED-Multilingual-Parallel-Corpus.