Auto Translation System - Summary Report

# 1. Objective

The objective of this project is to build a neural machine translation system capable of translating text from a source language to a target language using a Transformer-based architecture. The system is designed to automatically detect the input language and provide translations in a user-selected target language.

# 2. Technologies Used

This system uses TensorFlow and TensorFlow Text for building and training the Transformer model. SubwordTextEncoder is employed for tokenizing text inputs into subword units. The translation quality is evaluated using BLEU scores, and inference is performed using greedy decoding techniques.

# 3. Steps Performed

The process begins with data handling, where parallel corpora are loaded and cleaned. Sentences are lowercased and stripped of extra whitespace. The dataset is then split into training and testing sets, and subword tokenizers are trained for each language.  
  
The Transformer architecture includes positional encoding to retain the order of tokens, multi-head attention mechanisms, feed-forward networks, and appropriate masking techniques for padding and decoder lookahead.  
  
During training, the model is compiled using Sparse Categorical Crossentropy as the loss function and optimized with the Adam optimizer. A custom learning rate scheduler is used to enhance training performance.  
  
For inference, the model tokenizes and encodes the input sentence, and the decoder generates output one token at a time until the <end> token is reached. The final translation is obtained by detokenizing the generated output.

# 4. Auto Language Detection

A standout feature of this translation system is its ability to automatically detect the source language using the `langdetect` library. If no source language is provided, the system uses language detection to infer it. It then maps the detected language code to the appropriate token used by the translation model. This feature enhances the user experience by removing the need to manually specify the input language, enabling a smoother translation workflow.

# 5. Language Support

The system currently supports translations among English, Hindi, French, and Spanish. It uses language-specific codes to work with multilingual models, and it can be extended to support additional languages by updating the tokenizer and model mappings.

# 6. Strengths and Limitations

This system is modular, easy to extend, and provides a clean implementation using TensorFlow. It effectively demonstrates key natural language processing concepts and offers accurate basic translations.  
  
However, the system currently uses greedy decoding, which may not yield the best possible translations compared to more advanced techniques like beam search. While suitable for small- to medium-scale applications, the system may require optimizations for large-scale deployment. Additionally, the language detection step, while helpful, may occasionally produce inaccurate results for very short or ambiguous inputs.