# Objective: Translation Model Fine-tuning and Deployment

Problem Statement: Translation models often fail to account for cultural nuances, idioms, and community-specific expressions.

* For example, translating रसी जल गयी, बल नही गया using a generic model like Google Translate may yield incorrect results. The accurate translation is Even after one hit’s rock bottom, their arrogance remains unchanged.
* **Requirements**:

1. **Model Selection**: Choose a pre-trained translation model (e.g., MarianMT, T5, M2M-100).
2. **Dataset**: Use a small, labelled dataset with examples of culturally nuanced idioms and their translations.
   * Examples can be sourced from:
     + Tatobbaidioms Dataset
     + OpenSubtitles Dataset
3. **Tasks**:
   * Fine-tune the translation model on this dataset using frameworks like Hugging Face Transformers.
   * Deploy the fine-tuned model on a cloud platform (AWS, Azure, or GCP).
   * Build a REST API using frameworks like FastAPI or Flask to expose the model for predictions.
4. **Evaluation**:
   * Use BLEU or METEOR as evaluation metrics to measure translation accuracy.
   * Demonstrate performance improvements by comparing pre- and post-fine-tuning metrics.

* **Evaluation** **Criteria**:
* **Technical Skills:** Leveraging frameworks like Hugging Face, LangChain, and deploying on cloud platforms.
* **Problem-Solving:** Handling challenges related to fine-tuning and deployment.
* **Code Quality:** Clean, efficient, and well-documented.
* **API Design:** User-friendly documentation of REST API.
* **Deployment:** Scalable deployment ensuring reliability.

# **Model Selection**: Choose a pre-trained translation model (e.g., MarianMT, T5, M2M-100).

model selection step is crucial to ensure we deploy the most effective and efficient model for our specific requirements.

Before selecting one to deploy, I will be experimenting with two different models **(T5 & M2M-100)**, employing techniques such as **LoRA** and **4-bit quantisation.** After rigorously compare their performance across key metrics, such as accuracy, speed, and resource utilisation.

1. **Model selection: Facebook /**[**M2M100 418M**](https://huggingface.co/facebook/m2m100_418M)

* M2M100 418 a state-of-the-art multilingual translation model developed by Facebook AI (now Meta).
* It is designed to handle direct translation between any pair of 100 languages without relying on intermediate language pivots (e.g., translating from Hindi to English via an intermediate language like French).
* This direct translation approach makes it particularly suitable for fine-tuning to capture cultural nuances, as the model learns the specific linguistic and cultural patterns of each language pair directly.
* it was introduced in Beyond English [paper](https://arxiv.org/abs/2010.11125) and first released in [this](https://github.com/pytorch/fairseq/tree/master/examples/m2m_100) repository.
  + Transformers architecture (source: Vaswani et al., 2017)
    **Architecture**: M2M-100 is based on the **Transformer** architecture, which is backbone of most modern NLP models. Transformer uses self-attention mechanisms to capture contextual relationships between words in a sequence, making it highly effective for tasks like translation.

Figure 1. Transformer architecture

* + **Encoder-Decoder Structure**: An encoder processes the source text and a decoder that generates the target text. The encoder captures the meaning of the input text, while the decoder generates the translated output in the target language.
  + **Cross-Attention**: Decoder uses cross-attention to focus on relevant parts of the source **text while generating the target text, ensuring that the translation is contextually** accurate.
  + **Language-Specific Embeddings**: M2M-100 incorporates **language-specific embeddings** for each of the 100 supported languages (including Hindi and English). These embeddings help in distinguish between languages and improve translation quality.
* **Suitability for Task**

**M2M-100** is highly suitable for the task of fine-tuning a translation model to handle cultural

nuances, idioms, and community-specific expressions for the following reasons:

* + Language-Specific Embeddings
  + Scalability
  + Fine-Tuning Flexibility

1. **Model selection: Google /**[**T5-Small**](https://huggingface.co/google-t5/t5-small)

* **T5-Small** is a smaller variant of the Text-To-Text Transfer Transformer (T5) model developed by Google AI.
* Developed by Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu on [Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer](https://d.docs.live.net/df56beb8e7ea691b/Documents/Exploring%20the%20Limits%20of%20Transfer%20Learning%20with%20a%20Unified%20%20Text-to-Text%20Transformer) and released in [this](https://github.com/google-research/text-to-text-transfer-transformer#released-model-checkpoints) repository.
  + **Architecture**: T5 is also based on the Transformer architecture but is designed as a general-purpose text-to-text model. It treats every NLP task as a text-to-text problem, including translation.
* **Suitability for Task**

While M2M-100 and T5 Small share many desirable characteristics for fine-tuning on tasks involving cultural nuances, it has one key advantage that sets it apart:

* + **Smaller Model Size:** T5 Small is a smaller version of the original T5 model, making it more suitable for resource-constrained environments and faster inference.

# **Dataset:**

1. **Dataset Description:** [WikiMatrix Overview](https://opus.nlpl.eu/WikiMatrix/hi&en/v1/WikiMatrix)

WikiMatrix is a large-scale parallel corpus compiled by Facebook Research, designed to facilitate multilingual machine translation (MT) and other natural language processing (NLP) tasks. The dataset is derived from Wikipedia and contains parallel sentences in 1,620 language pairs across 86 languages.

* **Dataset Statistics:**
  1. Languages: **86**
  2. Language Pairs: **1,620**
  3. Parallel Sentences: **135 million**
  4. Total Tokens: **6.06 billion**
  5. Sentence Fragments: **300.27 million**

**Challenges**

* Domain-Specific Data: The dataset is derived from Wikipedia, which may not cover all domains equally. For domain-specific translation tasks, additional data may be needed.
* As my master's thesis was around this topic, I do have a dataset that might be helpful for this use case. However, I am not sure if I am allowed to use it here for this specific task. Therefore, I will stick to WikiMatrix.

1. **Data Preprocessing: (***Data Manipulation.ipynb***)**

* **Data Cleaning:** Data cleaning is a crucial step before preprocessing for Transformer models. Here's why:
  1. **A computer screen shot of a black screen

     Description automatically generatedNoise Reduction**: Cleaning removes inconsistencies, errors, and irrelevant information (noise) from your data. This improves the quality of your data and prevents the model from learning spurious patterns or being misled by erroneous information.

Figure 2 Non-English Characters

As per our use case, there is no use for non-English and non-Hindi data. Therefore, we will drop them as they will negatively impact our model's performance.

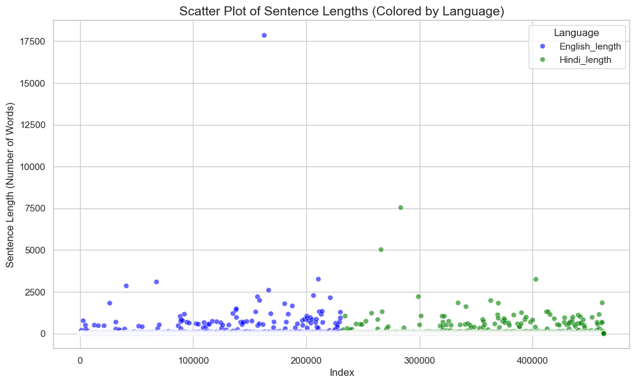
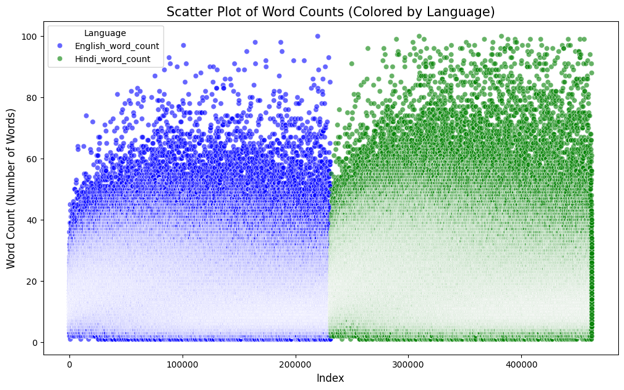
* 1. **Outliers:** data points that significantly deviate from the general pattern or trend observed in the rest of the dataset.

Figure 3. Scatter Plot to check for outliers

After further inspection, it was found that many of these sentences contained irrelevant patterns, empty spaces, and mixed languages, which are not needed.



To address this, we will keep sentences with a length threshold of 100 characters and drop the remaining ones.

* **Why it is important to drop outliers!!**  
  Outliers can negatively affect the performance of machine learning models, as they can cause the model to overfit or underfit the data.

Even after dropping outliers and removing noise we are we are still left with 2,30,225 rows

Figure 4 Density after Dropping.

* **Why it is threshold of 100 or 50 ?**A graph showing a number of words

  Description automatically generated
* Overall, the word count distributions provide valuable insights into the characteristics of the English and Hindi sentences in the dataset. By analysing these distributions, we can gain a better understanding of the data and make informed decisions about data preprocessing, Parameter selections, and evaluation.

Figure 5. KDE plot for Word Count

* Shape of the Distribution: Both English and Hindi word count distributions exhibit a similar shape, resembling a right-skewed or positively skewed distribution. This is typical for many natural language datasets, where shorter sentences are more common than longer ones.

At the end of Data cleaning, we are left with 2,24,194 saving it as processed data which is still a lot, will either only be using 1000 or 2000 of it for finetuning considering that we have cleaned our data enhancing the quality and inference should be better

In addition to other techniques, I explored vocabulary size reduction. Lowercasing the text can indeed reduce the vocabulary size, potentially leading to faster training and inference times.

**At the conclusion of the data cleaning phase, we were left with a dataset of 224,194 samples. While this constitutes a substantial dataset, we will likely utilize only 1,000 to 2,000 samples for fine-tuning. This approach is justified considering the thorough data cleaning process, which should enhance training quality and subsequently improve model inference performance**

**Model Selection & Fine Tuning (*Model Selection\_FT\_QT.ipynb*)**

1. **Facebook /**[**M2M100 418M**](https://huggingface.co/facebook/m2m100_418M) **:** it is a sequence-to-sequence (seq2seq) model. In essence, seq2seq models are designed to handle tasks that involve transforming one sequence of elements into another, here 100 is number of languages and 418 refers to the paraments or learnable weights in model neural network.

In the context of natural language processing, this often translates to tasks like:

* Machine Translation: Translating text from one language to another.
* Text Summarisation: Condensing large text into shorter summaries.
* Dialogue Generation: Human like conversational responses.
* Before fine-tuning a pretrained model, it is essential to preprocess the data, as different transformer models expect input in specific formats. Regardless of the data type—text, images, or audio—it must be converted and assembled into batches of tensors.
* Hugging Face provides powerful tools to simplify this preprocessing, with its dataset’s module being particularly helpful for managing and preparing datasets.