Project Description

Problem Statement:-

The objective is to develop and evaluate an automated translation model capable of accurately translating text between English and German.

Source of Data:-

Datasets provided by the ACL2014 Ninth Workshop on Statistical Machine Translation. These datasets encompass a variety of texts, including parliamentary proceedings (Europarl v7), web-crawled content (Common Crawl corpus), and news commentary (News Commentary).

Pre-processing Data:-

- Cleaning: Removing any unwanted characters, symbols, or formatting issues.
- Tokenization: Splitting sentences into words or subwords.
- Lowercasing: Converting all text to lowercase to ensure uniformity.
- Handling Punctuation: Removing or standardizing punctuation.
- Removing Stopwords: Optionally removing common words that may not be useful for certain tasks.
- Alignment Check: Ensuring that both files have the same number of lines and are correctly aligned.
- Term Frequency-Inverse Document Frequency (TF-IDF): Calculating the importance of words within the texts to highlight significant terms and aid in feature selection.

Model Selection

- RNN (Recurrent Neural Network):
  - Standard RNN: For our machine translation task, an RNN can be used as the foundational architecture. The RNN consists of an encoder and a decoder. The encoder processes the input sequence (German text), while the decoder generates the corresponding output sequence (translated English text). Standard RNNs, while useful for sequential data, often face challenges such as the vanishing gradient problem, which can impede their ability to capture long-range dependencies within the text. Despite these challenges, RNNs can serve as a baseline model for understanding the dynamics of sequence-to-sequence translation tasks.
- LSTM (Long Short-Term Memory):
  - LSTM: For our machine translation task, we adopt a sequence-to-sequence architecture using LSTM networks due to their ability to handle long-range dependencies more effectively than standard RNNs. The architecture consists of an encoder and a decoder. The encoder processes the input sequence (German text), creating a context vector that encapsulates the information from the input sequence. The decoder then takes this context vector and generates the corresponding output sequence (translated English text). LSTM networks are

chosen as the building blocks for both the encoder and decoder because of their internal gating mechanisms, which help manage the vanishing gradient problem and retain information over longer sequences. This makes them particularly effective for the task of machine translation, where capturing the context over long sentences is crucial.

- Transformer:
    - Transformer: Implementing a Transformer model which leverages self-attention mechanisms to capture dependencies in sequences without the recurrence-based architecture. Transformers offer several advantages, such as parallelization and potentially better performance for long-range dependencies. The Transformer's encoder-decoder structure will handle the translation task by allowing the model to pay varying degrees of attention to different words in the input sentence while generating each word in the output sentence. This self-attention mechanism enables the model to understand the context more effectively and translate text with higher accuracy and efficiency.

Training the Model:-

- Process:
    - Splitting data into training, validation, and test sets.
    - Initializing and running training algorithms to optimize model parameters.
    - Employing techniques such as early stopping and learning rate scheduling to enhance training efficiency and model performance.

Prediction:-

Utilizing the trained model to generate translations for unseen German text samples.

Evaluation:-

Evaluating the quality of translations using metrics such as BLEU score, which measures the similarity between the machine-generated translation and human-generated reference translations.

Capturing:-

Logging key metrics, such as training loss and validation accuracy, to track the performance of the model during training.

Documentation:-

Documenting the entire process, including data preprocessing steps, model architecture, training procedure, and evaluation metrics, to facilitate reproducibility and knowledge sharing.

Closure:-

Concluding the project by summarizing the findings, discussing limitations, suggesting potential improvements, and acknowledging contributions. Additionally, finalizing any remaining tasks, such as optimizing the model or deploying the model for practical use.

GIT Repo:- https://github.com/PrathamAgarwal52/machine-translation-model

Schedule:-

PROJECT SUB-TASK	TIME
Understanding and processing of data	Till 15 June
Building ML model	Till 20 July
Using ML model to predict new data	Till 27 July
Evaluation	Till 3rd August
Capturing the result	Till 10th August
Documentation and Closure of Project	Till 17 August

SUBMITTED TO:-	**SUPERVISOR:-**	**SUBMITTED BY:-**
Dr. Kumar Ratnakar	Guru Charan Bulusu	Pratham Agarwal
Mr. Arjun Verma		Pritam Burnwal