CS7.501: Advanced NLP

Project Name: Code-Mixed Machine Translation

Project Team: LexFlow

November 15, 2024

Team

Name	RollNumber
Shiva Shankar Gande	2023202005
Akshay Kohad	2023202007

1. Introduction

1.1 Problem Statement

The CodeMix project addresses the problem of translating code-mixed Standard English text into Hinglish. Code-mixing, prevalent in informal communication platforms like social media, combines words and syntax from multiple languages in a single sentence, which presents challenges for traditional machine translation models. This project aims to bridge the language gap by enabling smoother communication for those unfamiliar with both languages in the mix, providing them with a standard English translation.

The main challenge addressed in this project is the translation of code-mixed English-Hinglish text into coherent, understandable Hinglish. Code-mixed languages are a blend of two or more languages within a single conversation or text, often making it difficult for monolingual speakers to understand. With limited research and models dedicated to translating code-mixed languages effectively, there is a need for more effective solutions to bridge the communication gap.

1.2 Project Scope:

The objective of this project is to develop a machine translation system that translates code-mixed language, standard English into specifically Hinglish (a mix of Hindi and English). The project will aim to improve upon baseline performance using advanced techniques and models.

Goals:

1. **Data Processing**:
    - Use a dataset containing English-to-Hinglish parallel sentences for training.
2. **Model Selection**:
    - Utilize Custom Transformer as the baseline model, which has achieved a BLEU score of 20.58.

- Explore various translation models and techniques (e.g., transformer models, T5-small,MBart etc.) to improve the baseline performance.

3. **Model Optimizatio**n:
   - Experiment with pre-processing techniques such as language-specific tokenization or transliteration.
   - Experiment with post-processing techniques like fine-tuning to enhance translation accuracy.
4. **Evaluation**:
   - Use BLEU to evaluate the quality of the translations.
   - Conduct both automatic and manual evaluations to ensure the model's performance on code-mixed inputs.
5. **Challenges and Solutions**:
   - Address the unique challenges posed by code-mixed language, such as inconsistent grammar and word order between the two languages.
   - Explore alternate models, hybrid approaches, and custom solutions to improve performance over the baseline model.
6. **Deliverables**:
   - A functional code-mixed English-to-Hinglish translation model with performance improvements over the baseline.
   - A comprehensive report detailing model architectures, training procedures, evaluation metrics, and final results

1.3 Importance of the Problem

With the rise of digital communication, code-mixed language use has become common, especially among multilingual users on social platforms. However, it creates challenges for those who understand only one language involved in the mix. This problem is particularly relevant in multilingual countries where communication barriers due to code-mixing can limit information access and create misunderstandings. Addressing code-mixed translation contributes to broader inclusion and accessibility in digital communication.

1.4 Expected Outcome

The Project improves the result by training on various models on Advanced Dataset i.e findnitai/english-to-hinglish which has size of 1,89,102 sentences.The scope of this project includes building a machine translation model capable of handling code-mixed English-Hinglish text. Starting with a baseline using Transformer, which achieved a BLEU score of 20.58, the project progressively explores more models to improve translation quality. This includes potential pre-processing, post-processing, and experimentation with models such as T5,Mbart,bart,Helsinki to enhance translation accuracy and overall performance.

2. Literature Review / Background Study

Related Work

Title: "CoMeT: Towards Code-Mixed Translation Using Parallel Monolingual Sentences"

The key ideas of this paper are as follows:

- The paper proposes a system for translating English to Hinglish using mBART, a pre-trained multilingual sequence-to-sequence model, by transliterating Roman Hindi words to Devanagari script and utilizing parallel monolingual sentences.
- The system achieves a BLEU score of 12.22 on the test set and includes a detailed error analysis to explore the limitations of the provided dataset and metrics.
- Code-mixed languages like Hinglish are prevalent in multilingual societies but lack robust resources due to their informal nature, making it challenging to collect data.
- The authors suggest using a normalized BLEU score metric to better account for spelling variations in code-mixed sentences and emphasize the importance of evaluating code-mixing quality in translations.
- The research highlights the growing need for code-mixed language processing due to increased use on social media and messaging platforms, and aims to augment datasets for various Hinglish tasks by translating valuable datasets from high-resource languages.

Title: "Adapting Multilingual Models for Code-Mixed Translation"

The key ideas of this paper are as follows:

- The paper introduces a two-stage back-translation method called Back-to-Back Translation (B2BT) for adapting multilingual models to code-mixed translation, eliminating the need for external resources and showing significant improvements in translation quality.

- B2BT achieves substantial gains in BLEU scores, outperforming existing methods by up to +3.8 BLEU on code-mixed Hi→En, Mr→En, and Bn→En tasks, and achieving the highest score on the LinCE Machine Translation leaderboard for code-mixed Es→En.
- The scarcity of parallel data for code-mixed to pure language translation is addressed by B2BT through synthetic parallel data generation without relying on language-specific tools, making it scalable and simpler to implement.

- The approach involves training a base multilingual model on parallel matrix language to English corpus and non-parallel data, followed by two stages of back-translation-based fine-tuning to improve translation accuracy.

- The methodology is validated through both human evaluation and its impact on downstream models, with plans to release a new dataset and the code publicly.

Title:"Exploring Text-to-Text Transformers for English to Hinglish Machine Translation with Synthetic Code-Mixing"

The key ideas of paper as follows:

- The paper addresses the challenge of translating from English to Hinglish (a mix of Hindi and English), focusing on machine translation (MT) for code-mixed languages.
- The authors fine-tune the mBART, a pre-trained multilingual sequence-to-sequence model, for English-to-Hinglish translation. By leveraging mBART's pre-training on Hindi in Devanagari script, the model improves performance on Hinglish translations through the transliteration of Romanized Hindi words into Devanagari.
- The paper explores how translating English sentences into both Hindi and Hinglish, then using these parallel sentences as input for the model, helps enhance the translation quality.
- The authors propose a normalized BLEU score to handle the spelling variations in Romanized Hindi words, improving the fairness of translation evaluations. In addition, the Code-Mixing Index (CMI) is introduced as a metric to evaluate the level of code-mixing in the generated translations, ensuring linguistic diversity is captured.
- The authors suggest augmenting the dataset with more code-mixed sentence pairs, improving evaluation methods for code-mixed translation, and incorporating more linguistic features into translation systems for better results.

3.Selection of Baselines

- ## Custom Transformer:

  This model is typically tailored to specific needs by adjusting the transformer architecture (like layers, attention heads, or embedding dimensions) to optimize for particular tasks or languages. It's trained from scratch or fine-tuned on targeted data, allowing for greater flexibility in areas such as language translation, text classification, or summarization.
  Test Bleu Score Achieved : 20.58

- ## T5-small:

  Developed by Google, T5 (Text-To-Text Transfer Transformer) frames every NLP task as a text-to-text problem, enabling it to be highly adaptable. The T5-small variant is a lightweight version with fewer parameters (60 million), making it suitable for tasks requiring reduced computational resources while still delivering good performance in tasks like translation, summarization, and question answering.
  Test Bleu Score Achieved : 26.98

- ## Mbart:

  A multilingual version of BART, MBart is designed for translation and other multilingual tasks. It's pretrained on multiple languages, enabling it to perform zero-shot translations across language pairs. MBart is particularly useful in translation tasks where parallel data might be sparse, as it can leverage cross-lingual transfer.
  Fine tuned using lora on english to hinglish dataset
  Test Bleu Score Achieved : 43.23

- ## Helsinki-NLP/opus-mt-en-ROMANCE:

  This is part of the OPUS-MT project, focused on developing open machine translation models for many language pairs. The "en-ROMANCE" model translates English to Romance languages (e.g., Spanish, Italian, French). It's a fine-tuned transformer model trained on OPUS parallel corpora, providing reasonably high accuracy for translating to Romance languages without extensive resource requirements.
  The above model is finetuned using lora on english to hinglish dataset
  Test Bleu Score Achieved : 32.22

- ## Bart-Base:

  BART is a transformer encoder-decoder (seq2seq) model with a bidirectional (BERT-like) encoder and an autoregressive (GPT-like) decoder. BART is pre-trained by (corrupting text with an arbitrary noising function, and learning a model to reconstruct the original text.BART is particularly effective when fine-tuned for text generation (e.g.

summarization, translation) but also works well for comprehension tasks (e.g. text classification, question answering).BART model pre-trained on English language.

4. Dataset

Dataset: findnitai/english-to-hinglish

Dataset Description

This dataset comprises approximately 189,000 sentence pairs, each containing an English sentence and its corresponding Hinglish translation. The translations are sourced from both human annotations and synthetic generation methods, providing a diverse range of linguistic expressions.
It is an English to Hinglish Dataset aggregated from publicly available data sources.

Sources:

1. Hinglish TOP Dataset
2. CMU English Dog
3. HinGE
4. PHINC

Data Characteristics

Languages: The dataset focuses on English and Hinglish, where Hinglish combines Hindi vocabulary with English syntax and grammar.
Format: Each entry is a JSON object with the following structure:

- "en": The English sentence.
- "hi_ng": The Hinglish translation.
- "source": Indicates the origin of the translation—1 for human-annotated and 0 for synthetically generated.

Size: The dataset contains 189,102 rows, with a total size of approximately 27.1 MB.

Sample Entries

Here are some examples from the dataset:

```
{
 "en": "What's the name of the movie",
 "hi_ng": "film ka kya naam hai",
 "source": 1
}
{
 "en": "Hi, the rotten tomatoes score is great but the meta critic score seems a little low a movie of this quality.",
 "hi_ng": "namaste, sada hua tomatoes score mahaan hai, lekin meta critic score is gunavatta kee philm se thoda kam lagata hai.",
 "source": 1
}
{
 "en": "Do you think you will like the movie",
 "hi_ng": "kya aapako lagata hai ki aapako film pasand aaegee",
 "source": 1
}
{
 "en": "What kind of movie is it",
 "hi_ng": "yah kis tarah kee philm hai",
 "source": 1
}
```


5. Methodology

1. Model: Custom Transformer

Approach Overview:

This project implements a custom **Transformer model** for translating English sentences to Hinglish. The model is built from scratch, incorporating standard Transformer components, and trained on the English-to-Hinglish dataset. It uses various optimization techniques such as tokenization, custom embedding layers, and positional encodings to handle the complexities of bilingual sentence translation.

Model Architecture:

The implemented Transformer is an encoder-decoder model with configurable parameters such as embedding size (d_model = 512), number of attention heads (h = 8), and feed-forward hidden size (d_ff = 2048). It follows the standard Transformer architecture introduced by Vaswani et al., including multi-head self-attention, feed-forward layers, and residual connections. The model is designed to tokenize and project input and output sequences efficiently.

Implementation Details:

1. **Dataset:** The English-to-Hinglish dataset contains 50% of the original dataset (~94,500 sentence pairs) after filtering, with splits for training (~75,600 pairs), validation (~9,450 pairs), and testing (~9,450 pairs).
2. **Preprocessing:**
    - Custom tokenizers are trained for English and Hinglish, with special tokens ([SOS], [EOS], [PAD]) added to manage sentence boundaries and padding.
    - Sentences longer than 60 words are excluded to ensure consistent sequence lengths.
    - Inputs and labels are padded to a fixed sequence length of 128.
3. **Training Setup:**
    - The model is trained for 10 epochs with a batch size of 16.
    - Cross-entropy loss is used, with label smoothing (0.1) and padding ignored during loss computation.
    - The model is saved at the end of every epoch for checkpointing.

Optimization Techniques:

1. **Custom Architecture:**
    - Layer normalization and residual connections are used to stabilize training and avoid vanishing gradients.
    - Multi-head attention blocks enhance the model's ability to capture long-range dependencies.
2. **Tokenization and Positional Encodings:**
    - Custom word-level tokenizers ensure compatibility with the dataset.
    - Positional encodings help the model understand token positions in sequences.
3. **Greedy Decoding:** During inference, the model generates translations using a greedy decoding approach, adding tokens one at a time until an [EOS] token is reached or the sequence length is exceeded.
4. **Learning Rate Scheduling:**
    - A constant learning rate of 10−410^{-4}10−4 is used with Adam optimizer and weight initialization via Xavier initialization for stable convergence.

Results

- **BLEU Score**:
    - Achieved a **BLEU score of 20.58** on the test dataset.
- **Performance Visualization**:
    - Training loss and BLEU scores are visualized after each epoch.
    - The BLEU score distribution for individual test samples is plotted to analyze model performance variability.

2. Model: T5-Small

Approach Overview:

This project fine-tunes the T5-small model to translate English sentences to Hinglish. The model leverages mixed precision training, gradient accumulation, and multi-GPU support to optimize for both memory usage and performance.

Model Architecture:

The chosen model, T5-small, is an encoder-decoder model with about 60 million parameters, optimized for sequence-to-sequence tasks like translation. Its architecture is well-suited for handling language pair complexities, mapping English to Hinglish.

Implementation Details:

1. **Dataset**: The English-to-Hinglish dataset contains 189,000 sentence pairs, split into training, validation (evaluation), and test sets. The split is 80-10-10, giving approximately 151,200 pairs for training, 18,900 for validation, and 18,900 for testing.
2. **Preprocessing**: Each English input is prefixed with “translate English to Hinglish:” to standardize task format. Both source and target texts are tokenized up to 128 tokens and padded. Padding tokens in labels are masked with -100 to ignore them during loss computation.
3. **Training Setup**: Training is set to 5 epochs, using a batch size of 16 for training and 32 for validation and testing. Gradient accumulation occurs every 2 steps, allowing a simulated larger batch size. DataParallel wraps the model for multi-GPU support if multiple GPUs are available.

Optimization Techniques:

1. **Gradient Accumulation**: With accumulation every 2 steps, memory demands are reduced, allowing effective updates for larger batches.

2. **Learning Rate Scheduler**: A linear schedule with a 10% warmup period dynamically adjusts the learning rate, ensuring smooth ramp-up and stable training.
3. **Mixed Precision Training**: autocast and GradScaler are used for mixed precision, enhancing GPU efficiency and reducing memory usage.
4. **Beam Search Decoding**: A beam size of 4 is applied during decoding to improve translation quality by exploring multiple sentence structures.

Results

- **BLEU Score**:
  - Achieved a **BLEU score of 26.98** on the test dataset.
- **Performance Visualization**:
  - Training loss and BLEU scores are visualized after each epoch.
  - The BLEU score distribution for individual test samples is plotted to analyze model performance variability.

3. Model: Helsinki-NLP/opus-mt-en-ROMANCE

Approach Overview

The model leverages the Helsinki-NLP's opus-mt-en-ROMANCE transformer for English-to-Hinglish translation. Utilizing a Seq2Seq approach, we apply the model to translate source English text to Hinglish using mixed precision and gradient accumulation to optimize GPU usage.

Model Architecture

The model is a Transformer-based sequence-to-sequence model with an encoder-decoder architecture, pre-trained on multilingual translation tasks. It includes tokenization, positional embeddings, and self-attention mechanisms to capture language nuances across languages.

Implementation Details

1. **Dataset**: A custom English-to-Hinglish dataset with a split of 90% for training and 10% for evaluation.
2. **Data Preprocessing**: Source and target sequences are truncated or padded to a maximum length of 128 tokens, ensuring consistent input sizes.
3. **Batch Size**: Training with a batch size of 32, adjusted to GPU memory capacity.
4. **Epochs**: 3 epochs are used, with gradient accumulation steps set to 2 to mimic larger batch training.

Optimization Techniques

1. **Optimizer**: AdamW with a learning rate of 3e-5.
2. **Learning Rate Scheduler**: A linear scheduler with a 10% warmup phase.
3. **Mixed Precision**: Gradient scaling with autocast to reduce memory load and accelerate computations.
4. **Evaluation Metric**: BLEU score is used for evaluation to measure translation quality.

Results

- **BLEU Score**:
  - Achieved a **BLEU score of 32.22** on the test dataset.
- **Performance Visualization**:
  - Training loss and BLEU scores are visualized after each epoch.
  - The BLEU score distribution for individual test samples is plotted to analyze model performance variability.

4.Model: mbart-large-50-many-to-many-mmt

Approach Overview:

This approach uses the MBART model with a LoRA fine-tuning layer to improve English-to-Hinglish translation. The LoRA configuration targets specific modules (q_proj, k_proj, v_proj, out_proj) and uses rank r=8, scaling factor α=32, and dropout 0.1 to maintain memory efficiency while enhancing translation accuracy.

Model Architecture:

The architecture is based on the facebook/mbart-large-50-many-to-many-mmt model, fine-tuned with LoRA layers for efficient adaptation to English-to-Hinglish translation. MBART's pre-trained multilingual structure helps with sequence-to-sequence tasks, and its tokenizer supports 50 languages, making it suitable for this translation task.

Implementation Details:

1. **Dataset**: findnitai/english-to-hinglish, reduced by 50% for memory efficiency.
2. **Preprocessing**: English sentences are tokenized as input_ids, and target Hinglish sentences are tokenized as labels with padding.

3. **Data Collation**: A DataCollatorForSeq2Seq is used with batch size 16 and pin_memory=True for faster data loading.
4. **LoRA Setup**: LoRA layers target projection modules to enhance adaptation without fully training all model parameters.
5. **Training**: Mixed-precision with gradient accumulation (2 steps) and three training epochs. AdamW optimizer and linear learning rate scheduler handle weight updates.
6. **Evaluation**: SacreBLEU metric is used for validation and testing after each epoch. Test BLEU distribution is visualized.

Optimization Techniques:

1. **Gradient Accumulation**: Gradients accumulate over two steps to handle large batches effectively.
2. **Mixed-Precision**: autocast and GradScaler reduce memory usage and speed up training.
3. **Learning Rate Scheduler**: A linear scheduler improves stability across epochs.
4. **DataParallel**: Used for multi-GPU training, where torch.nn.DataParallel wraps the model across GPUs.

Results

- **BLEU Score**:
  - Achieved a **BLEU score of 43.23** on the test dataset.
- **Performance Visualization**:
  - Training loss and BLEU scores are visualized after each epoch.
  - The BLEU score distribution for individual test samples is plotted to analyze model performance variability.

5.Model: BART-Base

Approach Overview

This approach utilizes the **BART-Base model** for translating English sentences into Hinglish (Roman script). The pre-trained sequence-to-sequence architecture of BART is fine-tuned for the English-to-Hinglish task. By leveraging its robust denoising autoencoder capabilities, BART effectively captures the nuances of translation.

Model Architecture

The architecture is based on the **facebook/bart-base** model, which is a transformer-based sequence-to-sequence model pre-trained for tasks such as translation, summarization, and text generation. The model is fine-tuned for English-to-Hinglish translation with:

- **Source**: English sentences tokenized as `input_ids`.
- **Target**: Hinglish sentences tokenized as `labels`.

Implementation Details

1. **Dataset**: findnitai/english-to-hinglish.
    - The dataset includes English sentences paired with their Hinglish translations in the Roman script.
    - The full dataset is used for training.
2. **Preprocessing**:
    - English sentences are tokenized as input_ids, while Hinglish sentences are tokenized as labels.
    - Padding is applied with a maximum sequence length of 128 tokens for both inputs and outputs.
    - Padding tokens in labels are replaced with -100 to ensure they are ignored during loss computation.
3. **Data Collation**:
    - A DataCollatorForSeq2Seq is used to dynamically pad input and output sequences for each batch.
    - Batch sizes:
        - Training: 32 sentences per batch.
        - Evaluation: 16 sentences per batch.
4. **Training**:
    - **Optimizer**: AdamW with a learning rate of 5e-5.
    - **Scheduler**: Linear learning rate scheduler with warm-up.
    - **Epochs**: Four epochs for fine-tuning.
    - **Mixed-Precision Training**: Utilized for memory efficiency and faster computation.
    - **Multi-GPU Training**: The model is wrapped with torch.nn.DataParallel for multi-GPU setups.
5. **Evaluation**:
    - Metric: **SacreBLEU** is used to evaluate the quality of translations.
    - Predictions are post-processed to remove unnecessary spaces and special tokens before computing BLEU scores.
    - BLEU score distribution on the test set is visualized to assess translation performance across samples.

Optimization Techniques

1. **Mixed-Precision Training**:
    - torch.cuda.amp.autocast and GradScaler are used to reduce memory consumption and speed up computations.
2. **Gradient Accumulation**:
    - Not utilized in this setup since the GPU memory allows for training with a batch size of 32.
3. **Learning Rate Scheduler**:
    - A linear learning rate scheduler ensures smooth training and avoids large fluctuations.
4. **DataParallel**:
    - For multi-GPU environments, torch.nn.DataParallel wraps the model for efficient training across GPUs.

Results

- **BLEU Score**:
    - Achieved a **BLEU score of 33.06** on the test dataset.
- **Performance Visualization**:
    - Training loss and BLEU scores are visualized after each epoch.
    - The BLEU score distribution for individual test samples is plotted to analyze model performance variability.

Evaluation Process :

1. **Validation (Evaluation) Set**: At the end of each epoch, each model is evaluated on the validation set using BLEU scores. This ongoing evaluation tracks the model's learning progress and provides feedback for tuning.
2. **Test Set Evaluation**: After training is complete, the model's performance is assessed on the held-out test set, again using BLEU scores. This provides an objective measure of generalization to unseen data.
3. **BLEU Score Analysis**: Individual BLEU scores are calculated for test samples and plotted to assess the model's performance distribution.

4. **Sample Translations**: A few sample translations from the test set are displayed alongside their references, providing qualitative insights into translation fluency and accuracy in Hinglish.

6.Metrics

BLEU Score:

BLEU (**Bi**Lingual **E**valuation **U**nderstudy) is a metric for automatically evaluating machine-translated text. The BLEU score is a number between zero and one that measures the similarity of the machine-translated text to a set of high quality reference translations. A value of 0 means that the machine-translated output has no overlap with the reference translation (which indicates a lower quality) while a value of 1 means there is perfect overlap with the reference translations (which indicates a higher quality).

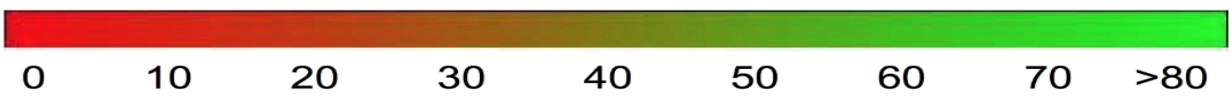
It has been shown that BLEU scores correlate well with human judgment of translation quality. Note that even human translators do not achieve a perfect score of 1.0.

Interpretation

Trying to compare BLEU scores across different corpora and languages is strongly discouraged. Even comparing BLEU scores for the same corpus but with different numbers of reference translations can be highly misleading.However, as a rough guideline, the following interpretation of BLEU scores (expressed as percentages rather than decimals) might be helpful.

BLEU Score	Interpretation
< 10	Almost useless
10 - 19	Hard to get the gist
20 - 29	The gist is clear, but has significant grammatical errors
30 - 40	Understandable to good translations
40 - 50	High quality translations
50 - 60	Very high quality, adequate, and fluent translations
> 60	Quality often better than human

The following color gradient can be used as a general scale interpretation of the BLEU score:



The mathematical details

Mathematically, the BLEU score is defined as:

$$\text{BLEU} = \underbrace{\min\left(1, \exp\left(1 - \frac{\text{reference-length}}{\text{output-length}}\right)\right)}_{\text{brevity penalty}} \underbrace{\left(\prod_{i=1}^{4} precision_i\right)^{1/4}}_{\text{n-gram overlap}}$$

with

$$precision_i = \frac{\sum_{\text{snt} \in \text{Cand-Corpus}} \sum_{i \in \text{snt}} \min(m^i_{cand}, m^i_{ref})}{w^i_t = \sum_{\text{snt'} \in \text{Cand-Corpus}} \sum_{i' \in \text{snt'}} m^{i'}_{cand}}$$

where

- m^i_{cand} is the count of i-gram in candidate matching the reference translation
- m^i_{ref} is the count of i-gram in the reference translation
- w^i_t is the total number of i-grams in candidate translation

The formula consists of two parts: the brevity penalty and the n-gram overlap.

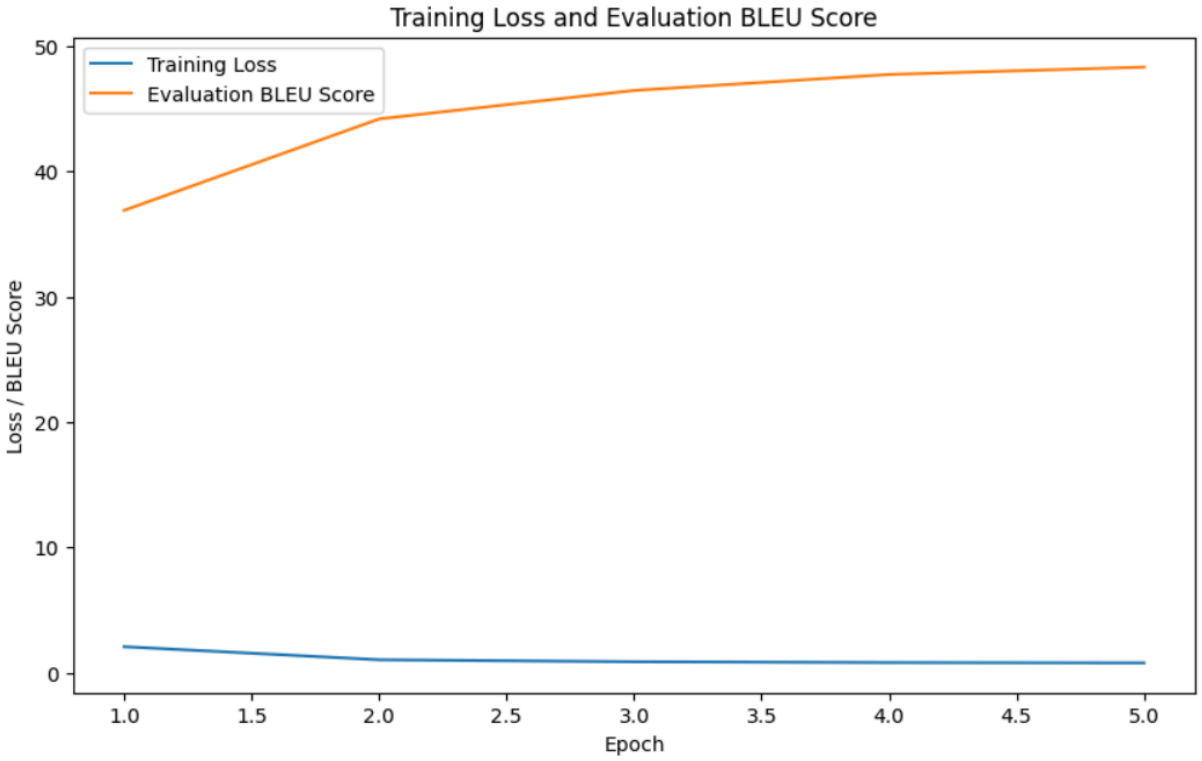
- **Brevity Penalty**
  The brevity penalty penalizes generated translations that are too short compared to the closest reference length with an exponential decay. The brevity penalty compensates for the fact that the BLEU score has no recall term.
- **N-Gram Overlap**
  The n-gram overlap counts how many unigrams, bigrams, trigrams, and four-grams (i=1,...,4) match their n-gram counterpart in the reference translations. This term acts as a precision metric. Unigrams account for adequacy while longer n-grams account for fluency of the translation. To avoid overcounting, the n-gram counts are clipped to the maximal n-gram count occurring in the reference (*mrefn*).

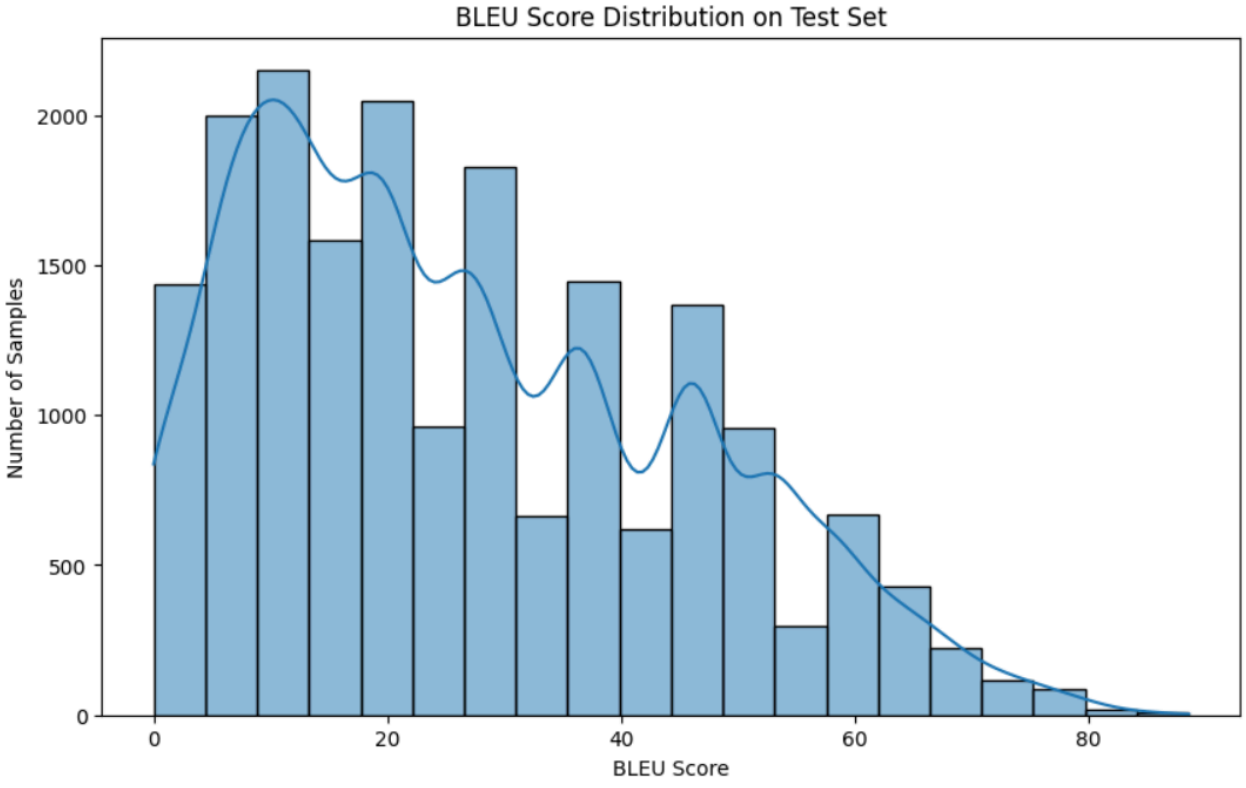
7. Results

Quantitative Analysis and Qualitative Analysis

T5-small

```
Epoch [1/5], Step [0/10637], Loss: 4.9462
Epoch [1/5], Step [10600/10637], Loss: 2.0790
Epoch [1/5], Step [10600/10637], Loss: 2.0790
Epoch [1/5], Evaluation BLEU Score: 36.91
Epoch [2/5], Step [0/10637], Loss: 1.2476
Epoch [2/5], Step [10600/10637], Loss: 1.0344
Epoch [2/5], Evaluation BLEU Score: 44.20
Epoch [3/5], Step [0/10637], Loss: 0.7940
Epoch [3/5], Step [10600/10637], Loss: 0.8752
Epoch [3/5], Evaluation BLEU Score: 46.48
Epoch [4/5], Step [0/10637], Loss: 0.9770
Epoch [4/5], Step [10600/10637], Loss: 0.8109
Epoch [4/5], Evaluation BLEU Score: 47.75
Epoch [5/5], Step [0/10637], Loss: 1.0989
Epoch [5/5], Step [10600/10637], Loss: 0.7833
Epoch [5/5], Evaluation BLEU Score: 48.34
Training completed.
```


Training Loss and Evaluation BLEU Score
Training Loss
Evaluation BLEU Score
Loss / BLEU Score
Epoch
0
10
20
30
40
50
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


BLEU Score Distribution on Test Set
Number of Samples
BLEU Score
0
500
1000
1500
2000
0
20
40
60
80

Average BLEU score on test set: 26.98

Sample translations from the test set:

Source: delete all recurring alarms
Reference: sarey recurring alarms delete karo
Prediction: sabhi recurring alarms delete kare

Source: what is the best route to the racetrack from my sister's house, if i left around 2 pm?
Reference: agar mai 2 pm ke aas paas nikal jaoon to meri sister ke ghar se racetrack tak pahunchne ke liye sabse best route konsa hai?
Prediction: agar mai 2 pm ko nikalta hoon toh mere sister ke ghar se racetrack tak sabse acha route kya hai?

Source: play the top 40 songs right now
Reference: top 40 songs ko abhi play kare
Prediction: abhi top 40 songs bajao

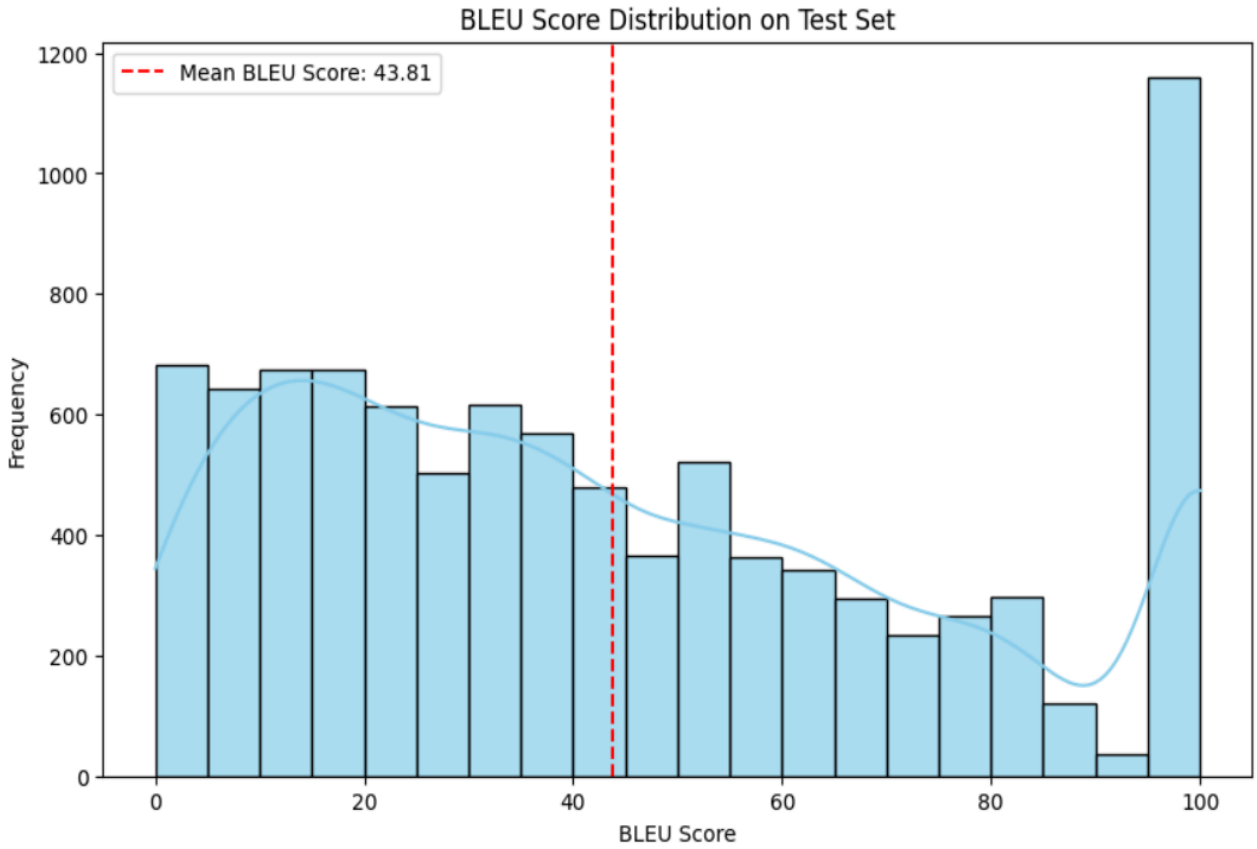
Source: i want to cancel that alarm
Reference: mai wo alarm cancel karna chahta hoon
Prediction: mai ye alarm cancel karna chahta hoon

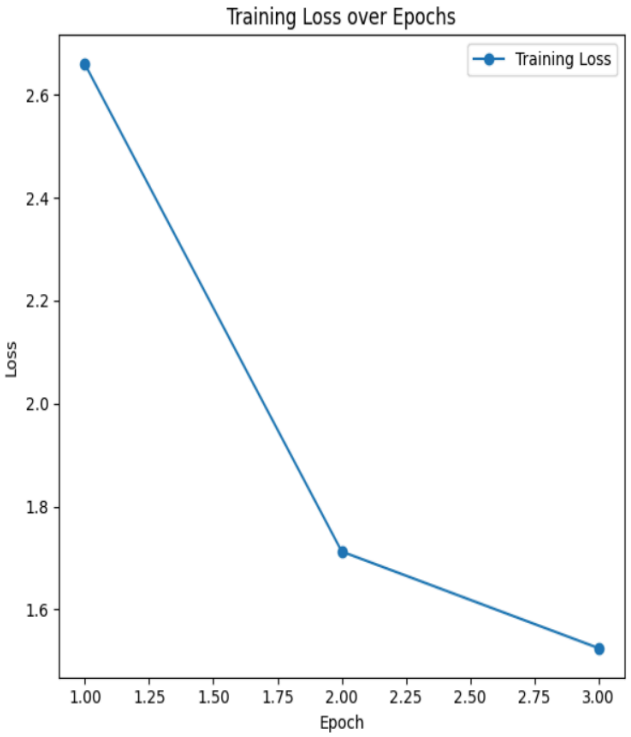
Source: alert me at 6 tomorrow instead of 7 am.
Reference: mujhe kal 7 am ke bajaye 6 ko alert kare
Prediction: mujhe kal subah 6 baje subah 7 baje ke liye alert kare

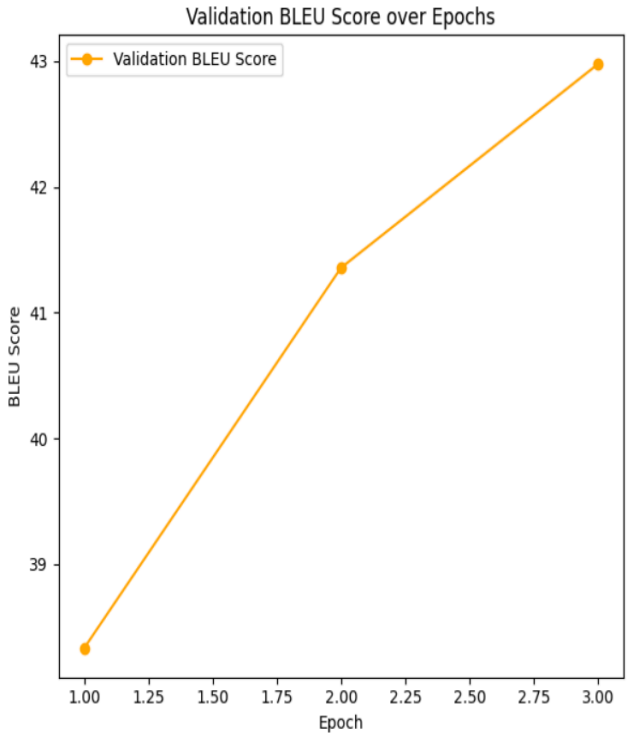
Custom Translation:
Input: I was waiting for my bag
Translated Output: mujhe mere bag ke liye waiting karne ke liye

Mbart

Test Bleu Score: 43.23







Custom Translation:
Input: play the top 40 songs right now
Translated Output: abhi top 40 songs bajao

Custom Translation:
Input: i want to cancel that alarm
Translated Output: mai wo alarm cancel karna chahta hoon

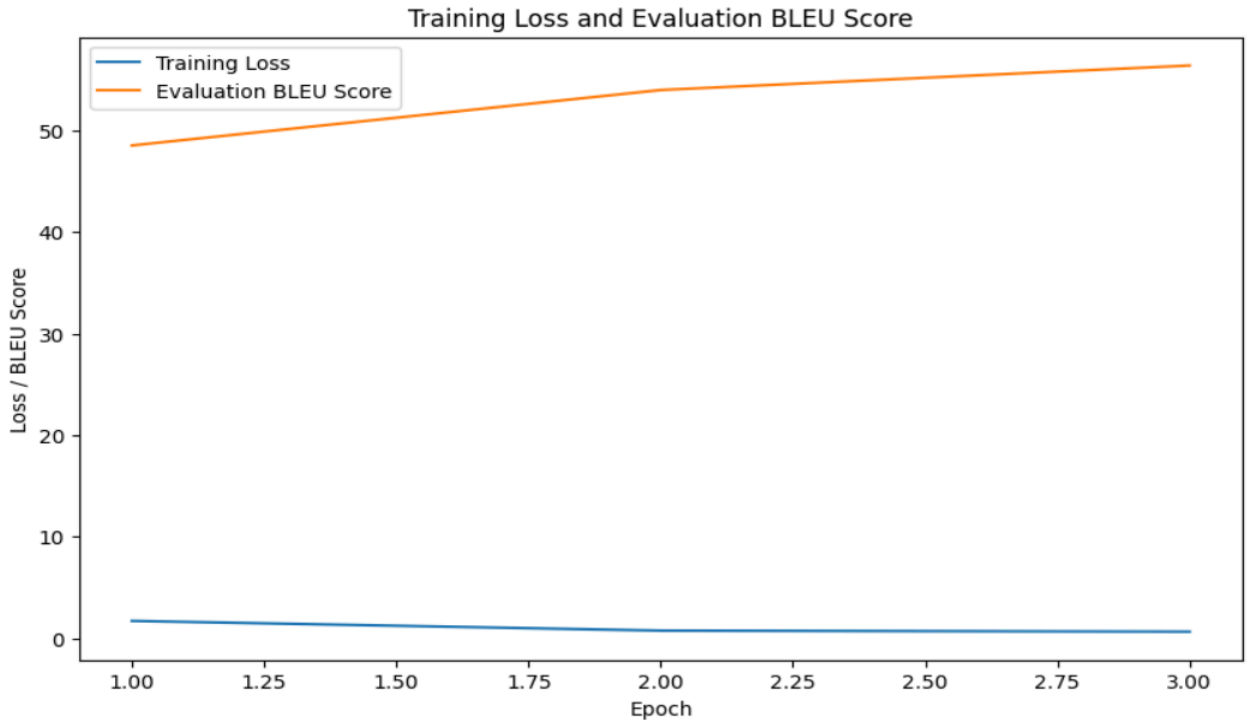
Custom Translation:
Input: what is the best route to the racetrack from my sister ' s house , if i left around 2 pm ?
Translated Output: mere sister ke ghar se racetrack tak sabse acha rasta kya hai

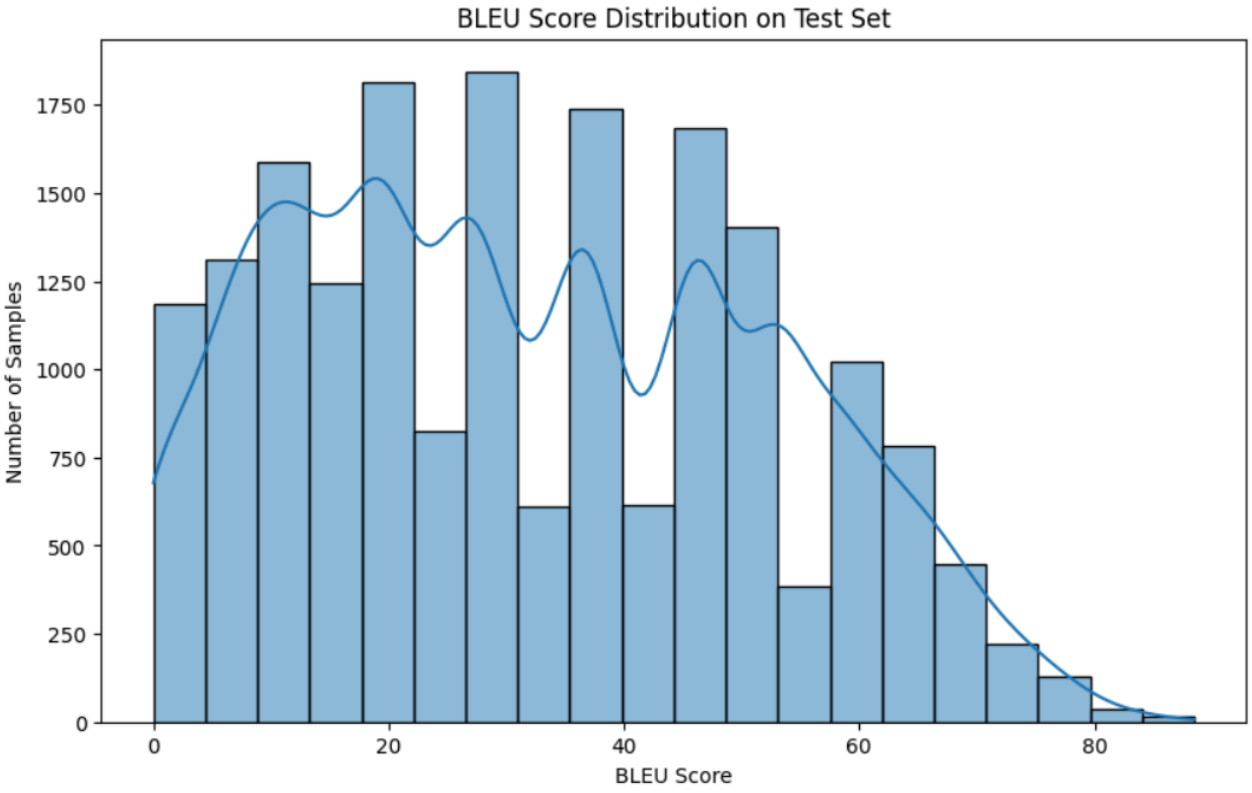
Custom Translation:
Input: delete all recurring alarms
Translated Output: sabhi recurring alarms ko delete kare

Custom Translation:
Input: I was waiting for my bag
Translated Output: mai mere bag ke liye wait karna chahta hoon

Hellenski

Test Bleu Score: 32.22





Sample translations from the test set:

Source: delete all recurring alarms
Prediction: sabhi recurring alarms delete karo

Source: what is the best route to the racetrack from my sister ' s house , if i left around 2 pm ?
Prediction: agar mai 2 pm ke aas paas nikal jaoon to meri sister ke ghar se racetrack tak sabse acha rasta kaunsa hai ?

Source: play the top 40 songs right now
Prediction: top 40 songs abhi bajao

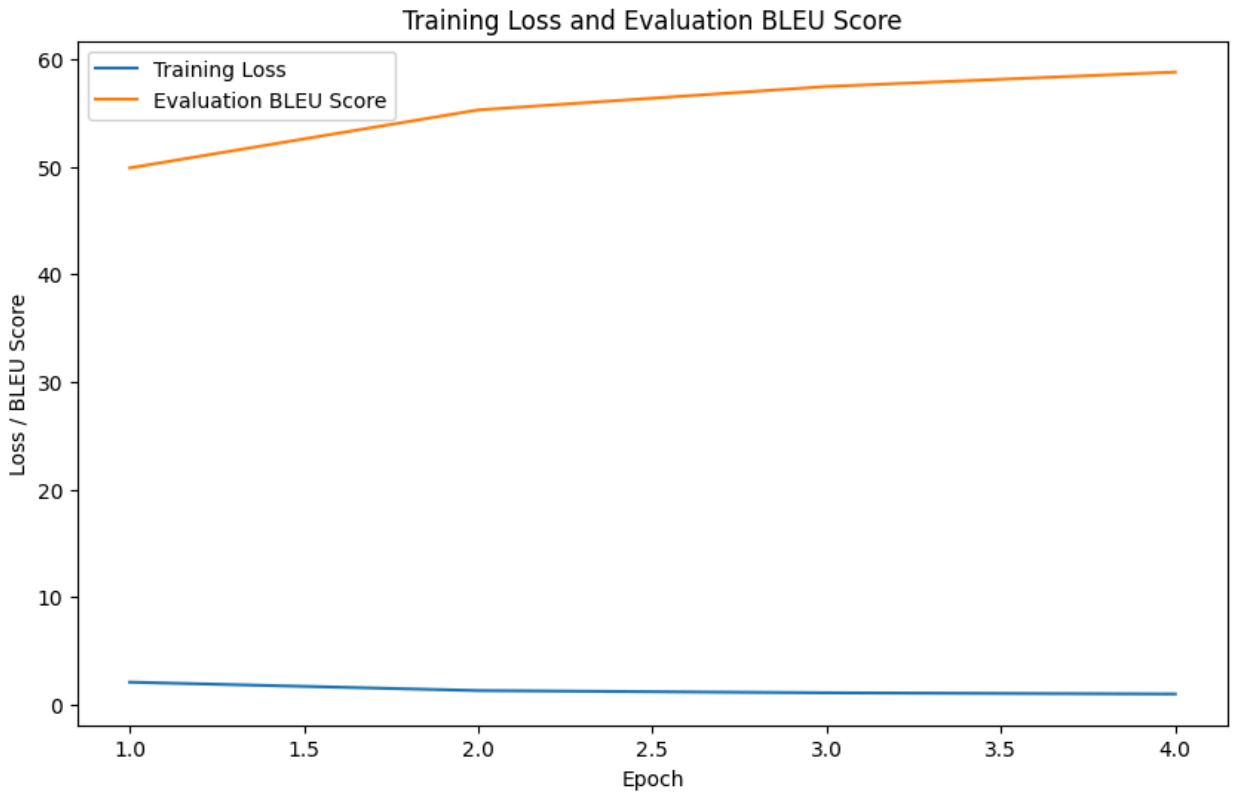
Source: i want to cancel that alarm
Prediction: mai wo alarm cancel karna chahta hoon

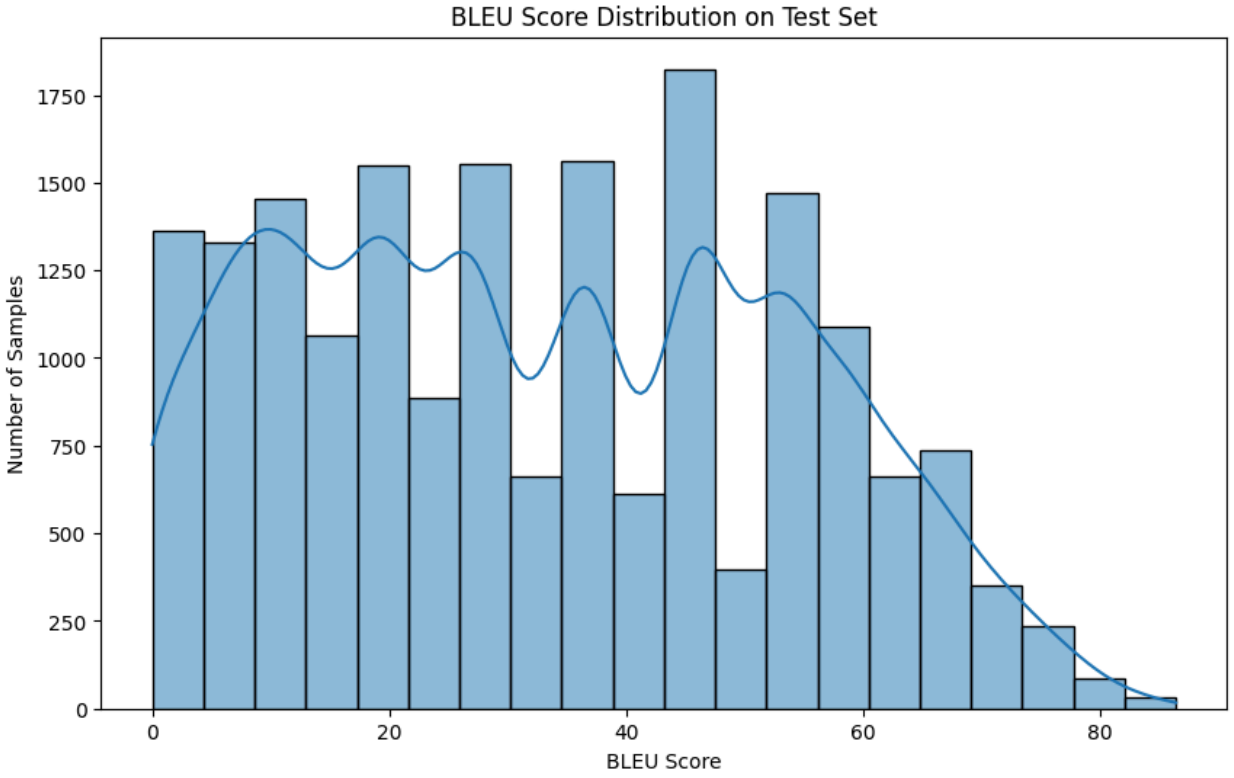
Source: alert me at 6 tomorrow instead of 7 am .

```
Prediction: mujhe kal subah 7 baje ke bajaye 6 baje alert karen .
Custom Translation: Input: I was waiting for my bag
Translated Output: main apne bag ko wait kar raha tha
```

Bart-base

```
Epoch [1/4], Step [0/5319], Loss: 8.7758
Epoch [1/4], Step [1000/5319], Loss: 3.2906
Epoch [1/4], Step [2000/5319], Loss: 2.7521
Epoch [1/4], Step [3000/5319], Loss: 2.4591
Epoch [1/4], Step [4000/5319], Loss: 2.2671
Epoch [1/4], Step [5000/5319], Loss: 2.1316
Epoch [1/4], Evaluation BLEU Score: 49.90
Epoch [2/4], Step [0/5319], Loss: 1.5474
Epoch [2/4], Step [1000/5319], Loss: 1.4240
Epoch [2/4], Step [2000/5319], Loss: 1.3788
Epoch [2/4], Step [3000/5319], Loss: 1.3612
Epoch [2/4], Step [4000/5319], Loss: 1.3383
Epoch [2/4], Step [5000/5319], Loss: 1.3202
Epoch [2/4], Evaluation BLEU Score: 55.29
Epoch [3/4], Step [0/5319], Loss: 1.0143
Epoch [3/4], Step [1000/5319], Loss: 1.1406
Epoch [3/4], Step [2000/5319], Loss: 1.1318
Epoch [3/4], Step [3000/5319], Loss: 1.1204
Epoch [3/4], Step [4000/5319], Loss: 1.1158
Epoch [3/4], Step [5000/5319], Loss: 1.1053
Epoch [3/4], Evaluation BLEU Score: 57.46
Epoch [4/4], Step [0/5319], Loss: 1.0129
Epoch [4/4], Step [1000/5319], Loss: 1.0277
Epoch [4/4], Step [2000/5319], Loss: 1.0142
Epoch [4/4], Step [3000/5319], Loss: 1.0053
Epoch [4/4], Step [4000/5319], Loss: 0.9966
Epoch [4/4], Step [5000/5319], Loss: 0.9956
Epoch [4/4], Evaluation BLEU Score: 58.81
Training completed.
```





Average BLEU score on test set: 33.06

```
Sample translations from the test set:

Source: how cold will it be today ?
Reference: aj kitni thandi hone wali hai ?
Prediction: aj kitni thandi hone wali hai ?

Source: remind me to pick up prescription
Reference: prescription lene ke liye mujhe yad dilaen
Prediction: mujhe prescription lene ke liye yaad dilaye

Source: Change 9 am alarm to 9 : 30 am
Reference: 9 am ke alarm ko 9 : 30 am me badle
Prediction: 9 am ke alarm ko 9 : 30 am me badle

Source: Would I be able to take the kids to the park tomorrow ?
Reference: kya mai kal bacchon ko park me le jaunga ?
Prediction: kya mai kal park me bacchon ko le jaunga ?

Source: timer start
Reference: timer ko start karo
Prediction: timer ko start karo

Custom Translation:
Input: I was waiting for my bag
Translated Output: mai mere bag ke liye waiting kar raha hoon
```

8. Analysis and Discussion

Model Comparison

All the models mentioned below performed better than the baseline model mentioned in the paper due to quality and quantity(1,89002 parallel sentences) of dataset which was formed on 23rd november 2023 on HuggingFace. Even though Bleu Score is not a good metric for evaluating code-mix for the comparison of models.

Model	Dataset	Epochs	BLEU Score	Training Time	GPUs
T5 Small	findnitai/english-to-hinglish	5	26.98	5 hours	2x Tesla T4
T5 Small (30% Dataset)	findnitai/english-to-hinglish	3	14.16	45 minutes	2x Tesla T4
Custom Transformer	English-to-Hinglish (Devanagari Script)	10	7.98	2 hours	P100
Custom Transformer	findnitai/english-to-hinglish	10	20.58	3 hours	P100
BART-Base	findnitai/english-to-hinglish	4	33.06	4 hours	2x Tesla T4
mBART-Large-50	findnitai/english-to-hinglish	3	43.23	3.36 hours	2x Tesla T4
Helsinki-NLP/opus-mt-en-ROMANCE	findnitai/english-to-hinglish	5	32.22	4 hours	P100

9.Future Work

Incorporating Additional Evaluation Metrics

- **Code-Mixing Index (CMI)**:
  - Hinglish is inherently a code-mixed language, and CMI provides a quantitative measure of the extent of language switching in the translations.
  - Future work will integrate CMI alongside BLEU to evaluate how naturally the models generate code-mixed text.
  - A detailed analysis will investigate the relationship between BLEU scores and optimal CMI ranges to ensure balanced and realistic Hinglish translations.
- **Semantic Metrics**:
  - Metrics such as **BERTScore** and **METEOR** will be explored to measure semantic similarity between predictions and references, as BLEU may overlook contextual and paraphrased accuracy.
- **Human Evaluation**:
  - Collect feedback from native Hinglish speakers to validate whether the generated translations align with human expectations in terms of fluency and cultural nuances.

10. References

- **Hugging Face**: A platform providing tools and resources for natural language processing, including pre-trained models and datasets.
  Hugging Face
- **Google Cloud Translation - Evaluate Models**: This resource offers insights into evaluating translation models, including the interpretation of BLEU scores.
  Google Cloud
- **CoMeT: Towards Code-Mixed Translation Using Parallel Monolingual Sentences**: A study exploring machine translation for code-mixed languages using parallel monolingual data.